



Northeastern University

---

Department of Civil and Environmental  
Engineering Reports

Department of Civil and Environmental  
Engineering

---

September 26, 2014

# Laser-based structural sensing and surface damage detection

Burcu Guldur  
*Northeastern University*

Jerome F. Hajjar  
*Northeastern University*

**Report No. NEU-CEE-2014-03**

---

## Recommended Citation

Guldur, Burcu and Hajjar, Jerome F., "Laser-based structural sensing and surface damage detection" (2014). *Department of Civil and Environmental Engineering Reports*. Report No. NEU-CEE-2014-03. Department of Civil and Environmental Engineering, Northeastern University, Boston, Massachusetts. <http://hdl.handle.net/2047/d20015559>

This work is available open access, hosted by Northeastern University.

Northeastern University was founded in 1898, as a private research university. Northeastern University is a leader in worldwide experiential learning, urban engagement, and interdisciplinary research that meets global and societal needs. Department of Civil and Environmental Engineering has over 100 years of history and tradition in research, teaching and service to the community, making important contributions to the development of our civil infrastructure and the environment, both nationally and internationally.

Contact:

Department of Civil & Environmental Engineering

400 Snell Engineering Center

Northeastern University

360 Huntington Avenue

Boston, MA 02115

(617) 373-2444

(617) 373-4419 (fax)

*The authors thank M. L. Wang, D. P. Bernal, M. MacNeil, M. Clifford at DGT Survey Group, and Faro Technologies for their contributions to this research. This material is based upon work supported by the National Science Foundation under Grant No. IIS-1328816 and Northeastern University. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.*

Northeastern University

Boston, Massachusetts

September 2014

## **Abstract**

Damage due to age or accumulated damage from hazards on existing structures poses a worldwide problem. In order to evaluate the current status of aging, deteriorating and damaged structures, it is vital to accurately assess the present conditions. It is possible to capture the in situ condition of structures by using laser scanners that create dense three-dimensional point clouds. This research investigates the use of high resolution three-dimensional terrestrial laser scanners with image capturing abilities as tools to capture geometric range data of complex scenes for structural engineering applications. Laser scanning technology is continuously improving, with commonly available scanners now capturing over 1,000,000 texture-mapped points per second with an accuracy of  $\sim 2$  mm. However, automatically extracting meaningful information from point clouds remains a challenge, and the current state-of-the-art requires significant user interaction. The first objective of this research is to use widely accepted point cloud processing steps such as registration, feature extraction, segmentation, surface fitting and object detection to divide laser scanner data into meaningful object clusters and then apply several damage detection methods to these clusters. This required establishing a process for extracting important information from raw laser-scanned data sets such as the location, orientation and size of objects in a scanned region, and location of damaged regions on a structure. For this purpose, first a methodology for processing range data to identify objects in a scene is presented and then, once the objects from model library are correctly detected and fitted into the captured point cloud, these fitted objects are compared with the as-is point cloud of the investigated object to locate defects on the structure. The algorithms are demonstrated on synthetic scenes and validated on range

data collected from test specimens and test-bed bridges. The second objective of this research is to combine useful information extracted from laser scanner data with color information, which provides information in the fourth dimension that enables detection of damage types such as cracks, corrosion, and related surface defects that are generally difficult to detect using only laser scanner data; moreover, the color information also helps to track volumetric changes on structures such as spalling. Although using images with varying resolution to detect cracks is an extensively researched topic, damage detection using laser scanners with and without color images is a new research area that holds many opportunities for enhancing the current practice of visual inspections. The aim is to combine the best features of laser scans and images to create an automatic and effective surface damage detection method, which will reduce the need for skilled labor during visual inspections and allow automatic documentation of related information. This work enables developing surface damage detection strategies that integrate existing condition rating criteria for a wide range damage types that are collected under three main categories: small deformations already existing on the structure (cracks); damage types that induce larger deformations, but where the initial topology of the structure has not changed appreciably (e.g., bent members); and large deformations where localized changes in the topology of the structure have occurred (e.g., rupture, discontinuities and spalling). The effectiveness of the developed damage detection algorithms are validated by comparing the detection results with the measurements taken from test specimens and test-bed bridges.

## Table of Contents

<b>Abstract</b> .....	<b>i</b>
<b>Acknowledgements</b> .....	Error! Bookmark not defined.
<b>Table of Contents</b> .....	<b>iii</b>
<b>Figures</b> .....	<b>viii</b>
<b>Tables</b> .....	<b>xiv</b>
<b>List of the Common Terms</b> .....	<b>xviii</b>
<b>1. Introduction</b> .....	<b>1</b>
1.1    Objectives and Scope .....	6
1.1.1    Small Deformations .....	6
1.1.2    Large Deformations with No Change in Topology .....	7
1.1.3    Large Deformations with Localized Change in Topology .....	7
1.2    Organization.....	10
<b>2. Background</b> .....	<b>13</b>
2.1    3D Laser Scanners .....	14
2.2    Key Specifications and Sources of Error for Time-of-flight and Triangulation-based 3D Laser Scanners .....	18
2.2.1    Key Specifications .....	18
2.2.2    Sources of Errors .....	19
2.2.3    Incomplete data.....	21
2.3    Laser Scanning and Image Processing Applications in Civil Engineering.	22
2.4    Laser Scan Processing for Modeling and Damage Detection.....	22
2.4.1    Registration.....	23
2.4.2    Feature Detection.....	24
2.4.3    Segmentation .....	25
2.4.3.1    Region Growing for Segmentation .....	26
2.4.4    Object Detection .....	27
2.5    Laser-based Modeling.....	30
2.6    Laser-based Damage Detection .....	31
2.7    Image Processing for Damage Detection.....	36
2.8    Combined Laser and Image Processing for Damage Detection .....	38
2.9    Current Inspection Strategies for Bridges.....	39
<b>3. Research Methodology</b> .....	<b>43</b>

3.1	Summary of Detectable Damage Types .....	43
3.2	Research Strategies .....	46
3.2.1	Methods for Small Deformations .....	47
3.2.2	Methods for Large Deformations with No Change in Topology .....	48
3.2.3	Methods for Large Deformations with Localized Change in Topology . .....	48
<b>4.</b>	<b>Equipment Specifications.....</b>	<b>50</b>
4.1	Available Datasets .....	53
<b>5.</b>	<b>Laser Data Processing towards Object Detection.....</b>	<b>56</b>
5.1	Registration .....	56
5.2	Neighborhood Size Selection.....	58
5.3	Outlier Point Detection .....	60
5.3.1	Radius-based Outlier Removal.....	61
5.3.2	Statistical Outlier Removal.....	62
5.3.3	Comparison Between Radius-based and Statistical Outlier Removal Methods .....	64
5.4	Surface Normal Estimation.....	67
5.5	Curvature Estimates .....	70
5.6	Extraneous Point Detection.....	71
5.7	Feature Detection .....	74
5.7.1	Gauss Map Clustering .....	75
5.8	Region Growing for Segmentation .....	78
<b>6.</b>	<b>Object Detection for Modeling and Damage Detection.....</b>	<b>82</b>
6.1	Point Feature Histograms.....	84
6.2	Surface Fitting.....	86
6.3	Global Feature-based Object Detection .....	88
6.4	Graph-based Object Detection.....	91
6.4.1	Voxelization.....	92
6.4.2	Skeletonization .....	96
6.4.3	Expanded Object Library.....	98
6.4.4	Object Detection and Model Fitting .....	101
<b>7.</b>	<b>Camera Integrated Laser-based Surface Damage Detection.....</b>	<b>104</b>
7.1	Noise Removal.....	105
7.1.1	Outlier Removal .....	105
7.1.2	Extraneous Point Removal .....	108

7.2	Graph-based Damage Detection .....	111
7.2.1	Surface Patch Investigation for Voxelization .....	113
7.2.2	Alignment .....	117
7.2.3	Rupture / Spalling.....	120
7.3	Surface Normal-based Damage Detection.....	124
7.3.1	Surface Patch Investigation for Computing Reference Normals.....	127
7.3.2	Normal Comparison for Plain Point Clouds.....	129
7.3.3	Normal Comparison for Texture-mapped Point Clouds.....	132
7.3.4	Effect of the Neighborhood Size on Normal Comparison .....	137
7.3.5	Clustering and Evaluation .....	139
7.3.5.1	Cluster Thresholding .....	140
7.3.5.2	Hierarchical Clustering .....	141
7.3.5.3	Cluster Evaluation .....	143
7.3.6	Damaged Area and Volume .....	145
7.3.6.1	Convex Hull for Damage Quantification .....	146
7.3.6.2	Mesh Grid for Damage Quantification.....	151
7.4	Parameters and corresponding ranges for investigated damage types.....	154
<b>8. Implementation of Neural Network Classifier for Crack Detection .....</b>		<b>157</b>
8.1	Clustering Tuning for Small Defects .....	159
8.2	Crack Dimension Extraction.....	161
8.3	Crack Dimension Validation.....	164
8.4	Spalling Dimension Validation.....	166
8.5	Classification through Artificial Neural Networks.....	171
8.5.1	Feature Extraction.....	173
8.5.2	Classification .....	175
<b>9. Combining Current Visual Bridge Inspection Strategies with Camera Integrated Laser-based Surface Damage Detection .....</b>		<b>179</b>
9.1	Current Condition of State Bridges.....	181
9.2	Bridge Condition Ratings .....	184
9.3	Bridge Load Ratings .....	193
<b>10. Damage Detection Applications and Results.....</b>		<b>197</b>
10.1	Available Datasets .....	198
10.1.1	Synthetic Point Clouds and Model Libraries.....	199
10.1.2	Steel Testing Frame .....	200
10.1.3	Concrete Testing Frame.....	201

10.1.4	DeKalb County Bridge .....	204
10.1.5	Bowker Overpass.....	207
10.2	Validation and Testing.....	212
10.3	Object Detection .....	213
10.3.1	Global-feature based Object Detection.....	214
10.3.1.1	Validation .....	214
10.3.1.1.1	DeKalb County Bridge.....	214
10.3.2	Graph-based Object Detection.....	216
10.3.2.1	Validation .....	217
10.3.2.1.1	Steel Testing Frame .....	217
10.3.2.1.2	Concrete Testing Frame .....	218
10.3.2.1.3	Bowker Overpass .....	221
10.4	Surface Damage Detection .....	223
10.4.1	Cracks .....	224
10.4.1.1	Validation .....	225
10.4.1.1.1	Bowker Overpass .....	225
10.4.2	Bent Members/ Alignment Issues / Points of Discontinuity .....	229
10.4.2.1	Validation .....	230
10.4.2.1.1	Concrete Testing Frame .....	230
10.4.2.2	Testing.....	232
10.4.2.2.1	Steel Testing Frame .....	232
10.4.2.2.2	Concrete Testing Frame .....	233
10.4.2.2.3	DeKalb County Bridge.....	236
10.4.3	Rupture / Cross-section Change / Spalling / Steel Delamination .....	236
10.4.3.1	Validation .....	238
10.4.3.1.1	Bowker Overpass .....	238
10.4.3.2	Testing.....	254
10.4.3.2.1	Concrete Testing Frame .....	254
10.4.3.2.2	DeKalb County Bridge.....	256
10.4.4	Limitations for Surface Damage Detection .....	258
10.5	Condition Rating.....	261
10.5.1	Concrete Testing Frame.....	262
10.5.2	DeKalb County Bridge .....	264

10.5.3	Bowker Overpass.....	265
<b>11.</b>	<b>Summary and Conclusion .....</b>	<b>267</b>
11.1	Conclusions.....	271
11.2	Future Work .....	272
	<b>References .....</b>	<b>275</b>
	<b>Appendices.....</b>	<b>293</b>
<b>A.</b>	<b>Appendix A - Results of Surface Normal Variation Study .....</b>	<b>294</b>
<b>B.</b>	<b>Appendix B - Representation of Mesh Grid Results.....</b>	<b>295</b>
<b>C.</b>	<b>Appendix C - Concrete Testing Frame Damage Measurements .....</b>	<b>296</b>
<b>D.</b>	<b>Appendix D – Neural Network Training Set Information .....</b>	<b>349</b>
<b>E.</b>	<b>Appendix E - Initial Site Investigation and Laser Scanning of Bowker Overpass ..</b>	<b>353</b>
E.1	Inspection and Field Report.....	353
E.1.1	4FH .....	353
E.1.2	4FJ .....	358
E.2	Laser Scan Locations for Field Scanning .....	361
E.2.1	4FH – Span 7 – Top and Bottom Deck .....	361
E.2.2	4FJ - Joint .....	370
E.3	Captured Point Clouds .....	374

## Figures

Figure 2.1: Representation of operating principles for pulse-based time-of-flight 3D laser scanners (Janos, July 2014).....	16
Figure 2.2: Representation of operating principles for phase-shift time-of-flight 3D laser scanners (Janos, July 2014).....	16
Figure 2.3: Representation of operating principles for triangulation-based 3D laser scanners (Geomatic 3D Systems Inc., May 2014).....	17
Figure 3.1: Research map. ....	44
Figure 5.1: The demonstration of radius-based outlier removal algorithm after O'Artista and Beintum (2010). ....	62
Figure 5.2: Results of statistical outlier removal example.....	63
Figure 5.3: (a) The complete scan of the collapsed test setup; and (b) the raw point cloud data used in comparison example. ....	64
Figure 5.4: (a) Raw point cloud; (b) resulting data set after radius-based outlier removal; and (c) the resulting data set after statistical outlier removal. ....	65
Figure 5.5: Close-up view for (a) raw point cloud; (b) resulting data set after radius-based outlier removal; and (c) resulting data set after statistical outlier removal.....	66
Figure 5.6: (a) Resulting point cloud after statistical outlier removal for raw data extracted from testing frame (Figure 5.3); and (b) the example neighborhood representation for a selected query point $p_q$ .....	69
Figure 5.7: (a) Plane fitting result for the selected query point $p_q$ from the previous figure; and (b) representation of the computed normal vector. ....	69
Figure 5.8: (a) Complete point cloud of the test setup at the MUST-SIM facility; and (b) a part of the point cloud of the test setup that is used for outlier and extraneous point detection.....	72
Figure 5.9: (a) Results of the radius-based outlier removal; and (b) results of extraneous point removal applied on a portion of the testing frame given in Figure 5.8. ....	73
Figure 5.10: Results of the extraneous point removal where the bolts and clamp is shown with rectangles and (b) side view of the clamp points detected by using the changes in surface variation.....	73
Figure 5.11: Example of gauss map clustering of an edge point represented by red dot where the green points represent the neighborhood points that participated in triangulation. ....	76
Figure 5.12: Gauss map clustering results for (a) a point on a flat surface; and (b) an edge point. ....	77
Figure 5.13: Edge points detected on synthetic point cloud. ....	78
Figure 5.14: Segmentation results for the synthetic point cloud. ....	81

Figure 6.1: Example of point feature histograms for points lying on simple 3D geometric surfaces after Rusu (2009). .....	86
Figure 6.2: (a) Scan data of DeKalb County Bridge with section utilized for data processing; and (b) DeKalb County Bridge point cloud with four piers detected as cylinders and pier cap surfaces detected as planes. ....	90
Figure 6.3: (a) A sample point set, (b) surface-based voxelization (also called surface meshing) result, and (c) point-based voxelization result. ....	92
Figure 6.4: Representation of (a) a sample voxel grid and (b) a single voxel. ....	94
Figure 6.5: (a) 3D voxel grid of a C-section; and (b) 2D binary image at $z=4$ ". ....	96
Figure 6.6: (a) Example C-section with the direction of height function, (b) 2D skeleton representation, (c) 3D skeleton representation. ....	98
Figure 6.7: An example of a model library representation of C-sections. ....	100
Figure 6.8: (a) Detected C-section $M_0$ ; (b) best match $B_0$ ; and (c) example section $CS_i$ . ....	102
Figure 7.1: Processed portion of DeKalb County Bridge, where red rectangle shows the part of the deck used for sensitivity analysis. ....	105
Figure 7.2: (a) Graph of $k$ versus number of outliers; and (b) graph of $k$ versus number of remaining points. ....	106
Figure 7.3: (a) Graph of $n$ versus number of outliers; and (b) graph of $n$ versus number of remaining points. ....	107
Figure 7.4: Image of concrete testing frame; and zoomed-in image of the investigated point area that has bolts attached to the concrete surface. ....	109
Figure 7.5: (a) Raw point cloud coming from area shown in Figure 7.4 that has 13,021 points; (b) resulting data set after statistical outlier removal with 12,630 points; and (c) resulting data set after extraneous point removal with 12,067 points. ....	110
Figure 7.6: (a) Graph of $n$ versus number of outliers and (b) graph of $n$ versus number of remaining points. ....	111
Figure 7.7: (a) Skeleton of a bent member; (b) skeleton of a discontinuous member; and (c) modal and point cloud section cuts of a C-section. ....	112
Figure 7.8: Steps of graph-based damage detection. ....	113
Figure 7.9: Beam and column legends of concrete testing frame. ....	115
Figure 7.10: Complete skeleton of concrete testing frame; and zoomed-in representation of one of the base columns. ....	119
Figure 7.11: (a) Drawing of one of the C-sections of steel testing frame; and (b) a representative cross-section with possible processing directions. ....	122
Figure 7.12: Reference normal representations: (a) from surface patches; (b) from object skeleton; and (c) from a reference point. ....	124

Figure 7.13: (a) Normal orientations of an undamaged surface; and (b) normal orientations of a damaged surface.....	126
Figure 7.14: Steps of surface normal-based damage detection.....	127
Figure 7.15: Representations of reference normals and normals from defect areas: (a) from surface data; (b) from object with a skeleton; and (c) from a surface with a reference point. ....	129
Figure 7.16: (a) Reference normal and one of the surface normals from a representative damaged area, shown in red; and (b) comparison angle between these two normals. ...	130
Figure 7.17: (a) Original point cloud remaining after outlier removal for a portion of the bottom deck of DeKalb County Bridge; and (b) defect locations after the application of surface normal-based damage detection. ....	131
Figure 7.18: Images of (a) BC-1-1 and (b) CD-3-1.....	133
Figure 7.19: Normal deviations for (a) BC-1-1 and (b) CD-3-1 for $k = 6$ . ....	133
Figure 7.20: Labeled cracks for (a) BC-1-1 and (b) CD-3-1. ....	134
Figure 7.21: Damage detection results for $k = 3$ , where (a) BC-1-1( $T_{int} = 0.63$ ) and (b) CD-3-1 ( $T_{int} = 0.78$ ). ....	135
Figure 7.22: Damage detection results for $k = 6$ , where (a) BC-1-1( $T_{int} = 0.63$ ) and (b) CD-3-1 ( $T_{int} = 0.78$ ). ....	135
Figure 7.23: Surface normal variations for several nearest neighborhood sizes. ....	138
Figure 7.24: (a) Defect locations for the investigated portion of the bottom deck of DeKalb County Bridge before cluster thresholding and (b) after cluster thresholding. .	141
Figure 7.25: Cluster evaluation results for the dataset from the bottom deck of Dekalb County Bridge given in Figure 7.24. ....	145
Figure 7.26: Rubber band analogy for convex hull. ....	146
Figure 7.27: (a) Convex hull representation of a sample defect dataset and (b) actual geometric representation of the defect.....	148
Figure 7.28: (a) Defect detection results for the investigated portion of the bottom deck of DeKalb County Bridge; and (b) 3D representation of the points of defect D1. ....	148
Figure 7.29: Labels of detected damaged locations for investigated portion of the bottom deck of DeKalb County Bridge.....	149
Figure 7.30: (a) 3D representation of the points of defect D1; and (b) <i>convhulln</i> result for the points of D1. ....	150
Figure 7.31: (a) 3D representation of the points of defect D1; and (b) <i>griddata</i> result for the points of D1.....	152
Figure 7.32: Representation of the points of D1 projected on the reference surface for area computation, by using the mesh grid approach.....	153
Figure 8.1: Results for both (a) initial and (b) improved clustering methods for BC-1-1. ....	160

Figure 8.2: Bounding box representation for BC-1-1.....	161
Figure 8.3: Presentation of a sample bounding box, and the approach for crack thickness computation.....	163
Figure 8.4: Reference measurement drawings for spalled concrete regions. ....	166
Figure 8.5: Sample false positive patterns: (a) small holes existing on the surface and (b) lines with significant intensity variation. ....	172
Figure 8.6: (a) Image of a portion of the concrete testing frame, and (b) defect detection results shown on corresponding portion of the point cloud.....	178
Figure 9.1: 2013 State bridge condition ratings (Governing., June 2013).....	181
Figure 9.2: (a) Model drawing of a C-section and (b) damaged cross-section representation extracted from laser scan.....	196
Figure 10.1: (a) Surface model and (b) point cloud representations of lab frame. ....	200
Figure 10.2: Dimensions of the test setup (Sagiroglu, 2012). ....	201
Figure 10.3: Beam and column legends of the testing frame. ....	202
Figure 10.4: (a) Testing frame with dynamic loading and (b) under pull down loading.....	202
Figure 10.5: Point cloud representation of the concrete testing frame. ....	203
Figure 10.6: (a) Picture and (b) point cloud of DeKalb County Bridge. ....	205
Figure 10.7: Point cloud of a portion of DeKalb County Bridge.....	206
Figure 10.8: Location and aerial view of Bowker Overpass (Google Maps., July 2014; Aaccessmaps., May 2014). ....	207
Figure 10.9: Location plan on the aerial view of Bowker Overpass (Mercuri and Mirliss, 2013). ....	208
Figure 10.10: Plan of the Bowker Overpass with the focus on Bridge 4FH and Bridge 4FJ (MassDOT., 1963). ....	209
Figure 10.11: Entire point cloud. ....	210
Figure 10.12: Point cloud of 4FH, Span 7. ....	211
Figure 10.13: Point cloud of 4FJ, corner. ....	211
Figure 10.14: (a) Scan data of DeKalb County Bridge with section utilized for data processing, and (b) point cloud of the same section with four piers detected as cylinders and pier cap surface detected as planes.....	215
Figure 10.15: Detected objects' properties exported to a computer aided design software (CAD): (a) object representations and (b) solid surfaces.....	215
Figure 10.16: Revit model of the steel testing frame.....	218
Figure 10.17: (a) The complete scan of the collapsed test setup and (b) Example section cut a-a at the specified location.....	219

Figure 10.18: (a) Skeleton of the test setup and (b) Resulting voxel representation of the section cut a-a at the specified location associated with a certain node of the skeleton graph. ....	219
Figure 10.19: (a) Portion of Bowker Overpass used for graph-based object detection, and (b) extracted cross-section. ....	221
Figure 10.20: Typical W-section dimensions. ....	222
Figure 10.21: Image of the processed portion of the Bowker Overpass for crack detection. ....	227
Figure 10.22: Image of the processed portion of the Bowker Overpass for crack detection. ....	227
Figure 10.23: Result of the crack detection algorithms for the region shown in Figure 10.21. ....	228
Figure 10.24: Result of the crack detection algorithms for the region shown in Figure 10.22. ....	228
Figure 10.25: Skeleton of the concrete testing frame with computed column alignment values. ....	232
Figure 10.26: Skeleton of the concrete testing frame with computed column alignment values and points of discontinuity. ....	235
Figure 10.27: Image of first section loss region, S1. ....	240
Figure 10.28: Results for (a) overall damage detection results and (b) section loss. ....	240
Figure 10.29: Image of second section loss region, S2. ....	242
Figure 10.30: Results for (a) overall damage detection and (b) section loss for S2. ....	242
Figure 10.31: Locations for the extracted section loss patches (Mercuri and Mirliss, 2013). ....	243
Figure 10.32: Image of CS1. ....	244
Figure 10.33: Damage Detection results for CS1: (a) surface representation and (b) 3D representation. ....	245
Figure 10.34: Image of CS2. ....	245
Figure 10.35: Damage Detection results for CS2: (a) surface representation and (b) 3D representation. ....	246
Figure 10.36: Image of CS3. ....	246
Figure 10.37: Damage Detection results for CS3. ....	247
Figure 10.38: Image of DL1. ....	249
Figure 10.39: Damage Detection results for DL1: (b) 3D representation and (b) close-up view with marked interest locations. ....	249
Figure 10.40: Image of DL2. ....	250

Figure 10.41: Damage Detection results for DL2: (b) 3D representation and (b) close-up view with marked interest locations. ....	250
Figure 10.42: Span 7, underside of deck condition plan (Mercuri and Mirliss, 2013)...	253
Figure 10.43: Point cloud of DeKalb County Bridge. ....	257
Figure 10.44: Minimum detectable damage dimensions versus point density for small deformations and large deformations with no change in topology. ....	260
Figure 10.45: Minimum detectable damage dimensions versus point density for large deformations with localized change in topology. ....	260
Figure C.1: Reference measurement drawings for spalled concrete regions. ....	296
Figure E.1: Side view of Bridge 4FH, Span 7 from east. ....	354
Figure E.2: Underside of Bridge 4FH, Span 7 and delamination on the south end. ....	354
Figure E.3: Section loss on east-side exterior fascia girder on Bridge 4FH, Span 7. ....	355
Figure E.4: Corrosion on the south side of the joint of Bridge 4FH, Span 7. ....	355
Figure E.5: Corrosion and delamination on the north side of the joint of Bridge 4FH, Span 7. ....	356
Figure E.6: Concrete spalling and cracking on the underside of Bridge 4FH, Span 7 deck. ....	356
Figure E.7: Cracks and corrosion on the underside of Bridge 4FH, Span 7 deck. ....	357
Figure E.8: Concrete spalling and exposed reinforcement on the underside of Bridge 4FH, Span 7 deck. ....	357
Figure E.9: Side view of the joint from the east on Bridge 4FJ with concrete spalling, delamination, etc. ....	358
Figure E.10: Section loss, concrete spalling and exposed reinforcement from the exterior view of the joint of Bridge 4FJ. ....	359
Figure E.11: Concrete spalling under the expansion joint of Bridge 4FJ. ....	359
Figure E.12: Corrosion on the north side of the joint of Bridge 4FJ. ....	360
Figure E.13: Corrosion on the south side of the joint of Bridge 4FJ. ....	360
Figure E.14: Laser scanner locations for bottom deck of Bridge 4FH, Span 7. ....	361
Figure E.15: Laser scanner locations for top deck of Bridge 4FH, Span 7. ....	367
Figure E.16: Laser scan locations for joint between Span 23 and Span 24. ....	370
Figure E.17: Entire point cloud. ....	374
Figure E.18: Point cloud of 4FH, Span 7. ....	375
Figure E.19: Point cloud of 4FJ, corner. ....	375

## Tables

Table 2.1: Summary of condition rating guidelines for deck, superstructure and substructure (Ohio Department of Transportation., 2010).....	42
Table 3.1: Detectable damage types and methods for detection.....	45
Table 4.1: List of the available laser scanners. ....	50
Table 4.2: Equipment specifications. ....	52
Table 4.3: List of the available laser data sets. ....	55
Table 5.1: List of laser scanners and corresponding registration software.....	57
Table 5.2: Statistical outlier removal example. ....	63
Table 5.3: Comparison between radius-based and statistical outlier removal methods. ..	65
Table 5.4: Parameters and corresponding ranges for given outlier removal methods.....	66
Table 5.5: Parameters and corresponding ranges for given surface normal estimation. ..	67
Table 5.6: Parameters and corresponding ranges for given curvature estimation. ....	71
Table 5.7: Parameters and corresponding ranges for given extraneous point removal. ...	74
Table 5.8: Parameters and corresponding ranges for gauss map clustering. ....	78
Table 5.9: Parameters and corresponding ranges for region growing segmentation.....	81
Table 6.1: Parameters and corresponding ranges for point feature histogram. ....	85
Table 6.2: Parameters and corresponding ranges for global feature-based object detection. ....	90
Table 6.3: Parameters and corresponding ranges for voxelization. ....	96
Table 6.4: Parameters and corresponding ranges for the graph-based object detection and model fitting.....	103
Table 7.1: Point densities from varying surface patches. ....	116
Table 7.2: Voxel grid dimensions.....	116
Table 7.3: Results of alignment algorithm for the investigated column.....	120
Table 7.4: Parameters and corresponding ranges for alignment comparison. ....	120
Table 7.5: Percentage change in cross-sectional area calculated at varying elevations. ....	123
Table 7.6: Parameters and corresponding ranges for area computation. ....	123
Table 7.7: Parameters and corresponding ranges for normal comparison for plain point clouds.....	131
Table 7.8: Parameters and corresponding ranges for normal comparison for texture-mapped point clouds. ....	132
Table 7.9: Hand measured and calculated thickness (largest crack width) and the length values for the damage locations labeled in Figure 5 for $k = 3$ . ....	136

Table 7.10: Hand measured and calculated thickness (largest crack width) and the length values for the damage locations labeled in Figure 5 for $k = 6$ .	136
Table 7.11: Percentage error values for both $k = 3$ and $k = 6$ .	136
Table 7.12: Parameters and corresponding values for hierarchical clustering.	143
Table 7.13: Parameters and corresponding values (or ranges) for cluster evaluation.	144
Table 7.14: The area and volume results for the convex hull application for the defects shown in Figure 7.29.	150
Table 7.15: Area and volume results for mesh grid application for defects shown in Figure 7.29.	154
Table 7.16: Parameters and corresponding ranges for alignment check.	154
Table 7.17: Parameters and corresponding ranges for investigated damage types.	156
Table 8.1: Results of crack validation study.	165
Table 8.2: Measured dimensions for concrete spalling validation.	168
Table 8.3: Computed dimensions for concrete spalling validation.	169
Table 8.4: Results of concrete spalling validation.	170
Table 8.5: Performance results for the neural network classifier.	177
Table 9.1: Changes in states' structurally deficient bridge inspection ratings between 2007 and 2013 (FHWA, July 2014).	182
Table 9.2: Changes in states' structurally deficient bridge inspection ratings between 2007 and 2013, cont'd (FHWA, July 2014).	183
Table 9.3: Massachusetts bridge statistics (FHWA, July 2014).	183
Table 9.4: Massachusetts bridges compared to U.S (FHWA, July 2014).	183
Table 9.5: Condition rating and deficiency reporting guides for Massachusetts Department of Transportation (O'Artista and Beintum, 2010).	185
Table 9.6: Criteria for alignment check (Ohio Department of Transportation., 2010)...	187
Table 9.7 : Condition rating criteria for steel beams/girders/slab (Ohio Department of Transportation., 2010).	188
Table 9.8: Percentage steel section loss values with respect to depth of section loss (Ohio Department of Transportation., 2010).	189
Table 9.9: Condition rating criteria for concrete beams/girders/slab (Ohio Department of Transportation., 2010).	190
Table 9.10: Condition state definitions for reinforced concrete items (AASHTO., 2010).	191
Table 9.11: Element definitions for condition states given in Table 9.10 (AASHTO., 2010).	191
Table 9.12: Condition rating criteria for substructure that contains timber piers.	192

Table 9.13: Condition factors (FHWA., 2012).....	195
Table 10.1: Libraries and validation datasets for object detection methods.....	213
Table 10.2: Validation and testing datasets associated with different damage types. ....	213
Table 10.3: Parameters and corresponding values for global feature-based object detection.....	214
Table 10.4: Parameters and corresponding ranges for the voxelization, model fitting and graph-based object detection.....	217
Table 10.5: Results of graph-based object detection for steel testing frame. ....	218
Table 10.6: Column detection results of graph-based object detection for concrete testing frame. ....	220
Table 10.7: Beam detection results of graph-based object detection for concrete testing frame. ....	220
Table 10.8: Detection results and actual girder dimensions. ....	222
Table 10.9: Nominal girder dimensions for W36 sections. ....	223
Table 10.10: Parameters and corresponding ranges for crack detection. ....	225
Table 10.11: Comparison between the field measurements and computed crack dimensions for patch one. ....	229
Table 10.12: Comparison between the field measurements and computed crack dimensions for patch two. ....	229
Table 10.13: Parameters and corresponding ranges for voxelization/skeletonization performed for graph-based damage detection.....	230
Table 10.14: Alignment results for columns of the concrete testing frame.....	231
Table 10.15: Alignment values computed for the steel testing frame. ....	233
Table 10.16: Alignment results for beams of the concrete testing frame. ....	233
Table 10.17: Points of discontinuity locations for the concrete testing frame.....	235
Table 10.18: Alignment results for detected timber piles of the DeKalb County Bridge. ....	236
Table 10.19: Parameters and corresponding ranges for surface-based damage detection. ....	238
Table 10.20: Comparison between the field measurements and computed section loss dimensions for S1. ....	240
Table 10.21: Comparison between the field measurements and computed section loss dimensions for S2. ....	242
Table 10.22: Comparison between the inspection report values and computed section loss dimensions. ....	243

Table 10.23: Comparison between the field measurements and computed concrete spalling for investigated patches. ....	247
Table 10.24: Comparison between the field measurements and computed delamination heights for investigated patches. ....	251
Table 10.25: Computed area and volume results for the detected damage regions. ....	251
Table 10.26: Computed area and volume results for the detected damage regions on the columns of the concrete testing frame. ....	255
Table 10.27: Computed area and volume results for the detected damage regions on the beams of the concrete testing frame. ....	256
Table 10.28: The results of the surface damage detection algorithms for individual items. ....	257
Table 10.29: Condition rating results for columns of the concrete testing frame. ....	263
Table 10.30: Condition rating results for beams of the concrete testing frame. ....	264
Table 10.31: Condition rating results for a portion of the DeKalb County Bridge. ....	265
Table 10.32: Condition rating results for two locations on the Bowker Overpass. ....	266
Table A.1: Surface normal variation results for a selected neighborhood. ....	294
Table B.1: Mesh grid results for defects D2-D7. ....	295
Table C.1: Measurements for labelled crack spalling locations for concrete testing frame. ....	297
Table C.2: Images of labelled crack and spalling locations for concrete testing frame. ....	304
Table D.1: Information on neural network training patches. ....	349
Table E.1: Images of the scanner locations captured during field scanning of Bridge 4FH, Span 7, bottom deck. ....	362
Table E.2: Images of the scanner locations captured during field scanning of Bridge 4FH, Span 7, top deck. ....	368
Table E.3: Images of the scanner locations captured during field scanning of Bridge 4FJ, corner. ....	371

## List of the Common Terms

The list below contains the definitions of common terms that are used most frequently throughout this work.

<i>Graph</i>	Representation of the skeleton of an object, consisting of vertices or nodes and lines or edges that connect them.
<i>Intensity value</i>	Number varying from 0 to 1 that is stored for each pixel (2D) or voxel (3D) of a grayscale digital image.
<i>Neighborhood points</i>	Set of $k$ nearest points ( $k$ nearest neighbors) determined based upon a distance criterion for a given query point.
<i>Point cloud</i>	Set of data points in a given coordinate system representing the external surfaces in a scene.
<i>Point cluster</i>	Segmented portion of a point cloud.
<i>Range data</i>	Distances to points in a scene from a given point, generally associated with the sensor device.
<i>Skeleton</i>	Thin version of a shape that is equidistant to the boundaries of the cross section of the shape and that represents geometrical and topological properties of the shape, such as its connectivity, topology, length, direction, and width.
<i>Skeletonization</i>	Process of extracting the skeleton of a shape.
<i>Surface patch</i>	Extracted portion of a point cluster used to compute the local properties.

*Voxel*

Single sample or data point on a regularly spaced, three-dimensional grid.

*Voxelization*

Process of converting geometric objects from a continuous geometric representation into a set of voxels that approximates the continuous object.

## **1. Introduction**

In 2013, the American Society of Civil Engineers (ASCE) released the most recent ASCE Report Card for America's infrastructure; this report depicts the current condition and performance of the nation's infrastructure (ASCE., July 2014). In this report, the average grade for all infrastructure types was a D+, and the grade given for bridges specifically was a C+. At this time, a significant percentage of the nation's bridges are either functionally obsolete, meaning those bridges were built using outdated standards, or structurally deficient, meaning those bridges are not safe because one or more of their major components have deteriorated. Even though this percentage has decreased over the last decade due to the increasing efforts of states, still 25% of the nation's bridges are in either deficiency category. This demonstrates the importance of accurately assessing the current status of aging, deteriorating and damaged structures and taking necessary precautions based on these up-to-date assessment results. This research investigates the use of high resolution three-dimensional terrestrial laser scanners with image capturing abilities as tools to capture geometric range data of complex scenes for structural engineering applications.

In recent years, applications of improved non-destructive testing methods for assessing the current conditions of structures have become more frequent. The traditional methods along with the new techniques have been increasingly used for determining and tracking structural integrity and assessing the nature of damage in a structure; some of the common methods include tap tests, impact-echo, ultrasonic measurements, acoustic emissions, gamma-ray radiography, ground penetrating radar, etc. (Chang et al., 2003; Sohn et al., 2004).

Even though non-destructive evaluation technologies for structures have improved significantly, visual inspection is still the primary tool to assess conditions of structures such as bridges and transportation infrastructure, power generation and transmission systems, above-ground pipeline systems, rail systems, dams and levee systems, and other exposed infrastructure. These inspections are important to track any changes that occur on structures during two successive inspections and also to ensure that the structures satisfy all applicable serviceability requirements. However, the results obtained through visual inspections may lack consistency that is essential to assess the current condition of structures effectively; moreover, such inspections are time consuming and even dangerous in some cases. There is a significant potential for variability in condition ratings, element-level inspection results, inspection notes, and photographs that are obtained during routine inspections. Since the allocations of construction, maintenance, and rehabilitation resources are often done based on the recorded condition ratings, this variation in these ratings is especially important. At the same time, these recorded condition ratings are also important parameters in load rating calculations. Depending on the severity of structures' current conditions, load ratings are used for determining further actions.

The Federal Highway Administration's Nondestructive Evaluation Validation Center (NDEVC) performed a study to evaluate the accuracy and reliability of visual bridge inspection in 2000, and the findings of this study are discussed in Phares et al. (2004). The study was conducted on two in-service and five decommissioned bridges by 49 inspectors from 25 states. For this study, the standard condition rating system given in the National Bridge Inventory (NBI), which was prepared by the Federal Highway

Administration (FHWA), is used. The condition ratings vary from 0 to 9, where 0 represents the failed condition and 9 the excellent condition. The results of the study showed that for individual bridge components, 95% of primary element condition ratings vary from the average with a standard deviation of two rating points; and only 68% of the condition ratings have a standard deviation of one rating point. Although the mentioned study was completed over a decade ago, the routine inspections for bridges still suffer from similar issues. Thus, it is important to develop new inspection strategies, which benefit from evolving data capturing technologies such as laser scanning to record and access the current conditions of structures effectively.

In this research, in order to capture the current condition of structures and to perform automated condition assessment, new camera-integrated laser scanner based inspection strategies have been developed. These strategies use camera-integrated laser scanners as an inspection tool that performs damage detection and characterization. Since the camera-integrated laser scanners are capable of capturing point clouds that provide information on the entire structure, in contrast to the current strategies, it is possible to develop a quantitative and systematic inspection strategy that involves no human/computer interaction. This approach allows the inspection information to be retained for future investigation, and it provides opportunity for comparative investigation over time.

Laser scanning capabilities have advanced significantly in recent years and have gained more recognition as a tool for applications in numerous fields. In the civil engineering domain, laser scanning technology has been used for several applications that include health monitoring, damage detection, etc. Some of the key laser scanner-based applications include monitoring civil infrastructure systems such as long span bridges to

obtain direct measurements of absolute displacement time histories at predefined locations (Chen et al., 2011), using laser scanners with wireless leave-in-place sensor nodes for in-situ detection and for monitoring metal fatigue (Buckner et al., 2008), comparing displacements measurements obtained by laser scanners with other static deformations sensors such as linear variable displacement transducers (LVDTs), electric strain gages and long gage fiber optic sensors (Park et al., 2007), using terrestrial laser scanner datasets to perform concrete surface damage recognition using mean and Gaussian curvatures of the surface of structures (Teza et al., 2009), locating and quantifying the damaged areas by using reference planes, which represent the intact condition of the surface of the investigated structure (Liu et al., 2011), using laser scanning for bridge under-clearance measurements to assess damage and to help engineers to develop bridge improvement planning (Liu et al., 2011). However, using laser scanners for inspecting existing structures and performing condition assessments is a relatively new and expanding research area. Possible future applications may involve rapid inspection of existing structures by using laser scanner mounted unmanned aerial vehicles

As mentioned previously, laser scanning technology has been evolving notably in recent years. Current commercial laser scanners are more accurate, more compact and a lot cheaper than they were a decade ago. It is now possible to collect millions of texture mapped data points that are accurate to within millimeters (FARO Technologies Inc., July 2014). However, one limiting factor preventing laser scanners from being widely utilized in more fields is the challenge of extracting useful feature information from large 3D datasets. These extracted features may then be tied to more complex structural models

or procedures, particularly of damaged structures. A laser scanner is capable of recording the location of a large quantity of points in space; however, reducing the data to useful information about the scene, including identification of objects without the use of pre-established object markers, is difficult. There is currently no comprehensive software available to take range data and produce useful models for structural engineering applications through automated methods.

In this research, in order to extract meaningful clusters from laser point clouds, widely accepted point cloud processing algorithms are implemented. The processing steps of these algorithms are registration, neighborhood size selection, outlier point detection, curvature estimation, extraneous point detection, feature detection, and object detection. These steps are used to extract the location, orientation, and sizes of objects in a scene. Once the objects are identified, either three-dimensional information or surface information of the detected objects is used to locate defects and quantify their magnitude. Two separate damage detection strategies are developed to perform damage detection. First, a graph-based damage detection method, which uses both global and local properties of detected objects, is developed. This method is used to compare the as-is conditions of individual objects with either the predicted or user-defined properties of the same object. Second, a surface normal-based damage detection method, which uses the local surface properties instead of using the global object properties, is established. This method is used to detect localized damage, which can be extracted by using the underlying surface properties along with the texture-mapped images.

## 1.1 Objectives and Scope

The main objective of this research is to establish enhanced inspection strategies for large-scale structures and infrastructure systems including bridges and transportation infrastructure, power generation and transmission systems, above-ground pipeline systems, rail systems, dams and levee systems, and other exposed infrastructure, and to propose a new automated strategy for damage quantification and documentation that will enhance the current practice for infrastructure inspection, where trained inspectors examine each structural component separately. This research aims to provide solutions for different damage types and states. The damage states are organized into three categories: small deformations, large deformations with no change in topology, and large deformations with localized change in topology. This research focuses on enhancing current visual inspection methods and developing automated geometry reconstruction and damage detection strategies to locate and quantify the surface defects that fall under the listed damage categories.

### 1.1.1 Small Deformations

For loadings that cause only small deformations, the most important damage types to capture are cracks and corrosion. For these types of deformations, using only laser scanners for damage detection may not be adequate. Thus, texture-mapped 3D point clouds, which are obtained by ubiquitous mapping of registered photographic images onto a 3D point cloud, should be used for enhanced visual inspection.

This research aims to improve the current visual inspection methods and to focus especially on structures with difficult access, or where frequent inspection is needed.

High-resolution images mapped on point clouds are used for automatically specifying locations of cracks, spalling, etc. with point cloud-based damage detection; and for developing strategies for storing and viewing this information.

### 1.1.2 Large Deformations with No Change in Topology

For deformations that are large enough to be documented, it is required to record alignment issues at important points of the structure for identifying damaged structural elements in order to track the changes on structures during their lifetime successfully. This will be accomplished by detecting and classifying individual members of a structure and comparing these members with their pre-defined library representations.

The focus of this research is to develop a sufficient member inspection technique that uses lasers and/or cameras as sensors that can access the structure at all stages and develop methods to assess geometry before and after deformation.

### 1.1.3 Large Deformations with Localized Change in Topology

In a heavily damaged structure, some portions of the structure may have been severed while the majority of the topology of the structure has remained intact, and this may result in localized changes in the structure's topology. In this case, it is critical to investigate the fracture locations and take precautions before further damage occurs. The damage types investigated under this category are ruptures, concrete spalling, steel delamination, and points of discontinuity.

This research aims to create an automated damage detection and localization system which detects separate objects from raw 3D point clouds, and compares the current

condition of an object with the best-match from a pre-built object library. This method will be effective in determining the location of damaged members as well as quantifying the severity of damage.

To achieve these described objectives, the following new methods and algorithms have been developed in this work for texture-mapped point clouds:

- A surface-normal based damage detection method that only uses the 3D coordinate information for locating rupture, spalling, delaminations (Section 7.3.2).
- An improved surface-normal based damage detection method that also uses the intensity information along with 3D coordinate information for locating small deformations such as cracks, corrosion (Section 7.3.3).
- A graph-based damage detection method for detecting alignment issues, points of discontinuity (Section 7.2.2).
- Automated defect clustering method for processed point cloud regions that have more than one defect (Section 7.3.5).
- A mesh grid-based defect area and volume computation method (Section 7.3.6.2).
- An improved clustering method for segmenting the detected cracks (Section 8.1).
- Automated crack length and width extraction method (Section 8.2).
- A decision-making system based on detected defects for automated condition rating assignment to investigated items of the structure. (Section 10.5).

At the same time, several algorithms and/or methods from the literature have been adapted and implemented for the purposes of this research; these include:

- A global feature-object detection method to perform structural sensing. (Section 6.3). Common point cloud processing algorithms that include registration, outlier removal, extraneous point removal, feature extraction, region growing are implemented. The segmented surfaces are then used for detection several defined objects from the investigated point clouds.
- A graph-based object detection method that generates skeletons from cross-section cuts of a voxelized cluster through skeletonization in order to detect common structural members (Section 6.4). Available skeletonization algorithms are used to extract the skeleton from the cross-section cuts. These then form the basis for a new algorithm for object detection that is developed based on the use of a library generated from common structural shapes.
- A convex hull-based damage area and volume quantification method for point clouds (Section 7.3.6.1).
- A method that converts cross-section voxel representation automatically into a polygon for computing the changes in the cross-section through area calculation and determining the total volume change on the investigated member (Section 7.2.3). Available polygon generating algorithms are used specifically for extracted member cross-sections, and the obtained results are then used for computing cross-section changes to document associated damage.

- An artificial neural network classifier for extracting true positives from detected crack clusters from point clouds (Section 8.5).

Several datasets are used as the proof-of-concept for all the damage states listed in Sections 1.1.1-1.1.3.

## **1.2 Organization**

Chapter 1 of this dissertation presents limitations of the current routine visual inspection methods and discusses the use of laser scanning technology to improve the current practice. Chapter 2 follows by describing previous research that has been completed regarding point cloud and image processing, which consist of steps to process the three-dimensional datasets collected via laser scanners, and also regarding laser- and image-based surface damage detection methods. This chapter also discusses the current visual inspection methods for bridges. Even though the developed algorithms are general and applicable to any structure that has visible surfaces, the primary scope of this research is on bridges. Thus, while discussing the current visual inspection strategies and condition rating criteria, only the strategies for bridges are included. Chapter 3 describes the research methodologies that have been developed as part of this project. Chapter 4 lays out the equipment specifications for several sensors that are used in this research.

Starting with Chapter 5 and continuing in Chapter 6, both readily available and developed point cloud processing algorithms are discussed. Chapter 5 focuses on the point cloud processing algorithms that are implemented for this research; these algorithms include registration, neighbourhood size selection, outlier point detection, surface normal estimation, curvature estimation, extraneous point detection, and feature

detection. Chapter 6 begins with discussing point feature histograms, which are used to identify the underlying surfaces. Once the surface type is determined, it is followed by object detection. This chapter describes two object detection methods, which are used in different situations. First, a global-feature based object detection method that operates on the global properties of individual objects is introduced. Then, a graph-based object detection method, which is more suitable for localized detection is discussed. The applications of both object detection methods are shown for several datasets.

Chapter 7 begins by discussing the developed surface damage detection methods. In this chapter, two surface damage detection methods are described. The graph-based damage detection method uses three-dimensional information associated with each object to localize and quantify element damage. The surface normal-based damage detection method is dependent on the extracted surface information. This chapter discusses both methods in detail and provides several application examples. Chapter 8 continues discussing the damage detection methods by presenting a strategy for improving crack detection through implementation of a cluster optimization algorithm, which is specifically developed for cracks. This chapter also discusses the implementation of artificial neural network for improving the accuracy of the developed crack detection algorithms.

Chapter 9 focuses on combining current visual inspection methods given for bridges with camera integrated laser-based surface damage detection. The visual inspection strategies from several states are presented. This chapter also discusses the importance of repeatability and accuracy of condition rating determination for load rating calculations on bridges.

Chapter 10 presents the results obtained from the investigated datasets by using the methods described for object detection, damage detection, and condition rating. In each section of this chapter, first, the description of a dataset is given and then, the results obtained for that dataset are included. Finally, Chapter 11 ends the dissertation with summary and conclusions and several proposals for future work.

## **2. Background**

There are thousands of structures such as large dams and bridges that have been now operating for more than five decades and have age related problems. Assessing the current condition of structures such as these has always been important due to their impact on the landscape where they are built. The main aim of routine inspection activities is to assess structural safety condition of these structures, and it is a significant concern for authorities (Committee on the Safety of Existing Dams., 1983).

This research investigates the use of camera-integrated laser scanners as a tool for improving the current visual inspection strategies. In order to develop new strategies that involve laser scanners, it is important to understand the available scanner options and limitations associated with them. Thus, this chapter starts by explaining laser scanning technology, available options, limitations, and sources of errors. This discussion is followed by a description of how laser scanning and image processing are currently used for civil engineering applications.

Once the defect scope is determined, the next step is to develop several strategies that involve processing texture-mapped laser scanner datasets for locating and quantifying visible damage. However, raw laser scanner datasets are not suitable to be directly processed for damage detection. Thus, readily available point cloud processing algorithms, which are commonly used on laser scanner datasets, are investigated. These algorithms are used to reduce point clouds into meaningful clusters, which either represent objects or surfaces. An extensive literature review is performed on this topic to determine the most suitable algorithms for the applications in this work. In the section

that follows point cloud processing literature review, several studies on laser-based modeling are represented.

Next, the previous research on laser- and/or image-based damage detection is investigated. First, the studies that involve laser scanner-based damage detection are described. Second, the image-based damage detection applications are discussed. Finally, the studies, for which a combination of laser scanners and cameras are used, are laid out. This chapter concludes with a description of the current visual inspection strategies for bridges, which are later used for assessing the condition of individual elements automatically based on detected surface damage.

## **2.1 3D Laser Scanners**

3D laser scanning can be described as the controlled steering of laser beams that is used to get a distance measurement at every point on the surface of investigated shapes. The scanned shapes can be objects, buildings, bridges, landscapes, and etc. 3D laser scanners capture data from the surface of shapes. The collected datasets can either consist of only the geographic coordinates of each point on the surface, or they can have color information in addition to the geographic coordinates.

There are two well-established categories that divide all 3D laser scanners into two types: contact and non-contact laser scanners. Non-contact laser scanners are then divided into two main categories: active and passive scanners (Curless, 1999; Bernardini and Rushmeier, 2002).

Contact 3D scanners require physical touch with the surface of investigated shapes. Shapes are positioned and held in place by a fixture on a precision surface plate, which

has a specific maximum surface roughness. The contact scanners either consist of a carriage system with rigid arms that are perpendicular to each other, where each axis glides along a track, or an articulated arm with rigid bones and high precision angular sensors.

Non-contact active 3D scanners are used to probe an object or environment by emitting either radiation or light. Radiation passes through objects and light is reflected from the surface. Some possible emission types can be listed as ultrasound, x-ray and light. There are several types of non-contact active laser-scanners; however, they can be grouped under two categories: surface-based scanners that are classified as time-of-flight, triangulation-based, conoscopic, hand-held, structured light, and modulated light; and volumetric scanners that include computed tomography (CT), magnetic resonance imaging (MRI), industrial computed tomography, and micro-tomography. The 3D scanners that are used in this research are all non-contact active 3D scanners, and they fall under either the time-of-flight scanners category or the triangulation-based scanners category.

Time-of-flight 3D laser scanners use laser light to collect geographic location information from the surface of structures by using a time-of-flight laser rangefinder. The round-trip time of a pulse of light is timed by the rangefinder. This recorded round-trip time is used to determine the travel distance. If the recorded travel time is  $t$ , then the distance to the object can be represented as  $\frac{c \cdot t}{2}$ , where  $c$  is the speed of light. Since travel distance represents twice the distance between the scanner and the surface, only half of the travel distance is used to calculate the distance of a surface point from the scanner. Time-of-flight laser scanners are divided into two categories as well: pulse-based or

phase-shift systems. Pulse-based systems use straight laser beams with constant power; however, phase-shift systems modulate the power of the laser beams. For phase-shift systems, the scanner compares the phase of the sent laser beam with the phase of the received laser beam (Pfeifer and Briese, 2007). Time-of-flight 3D scanners are classified as mid- to long-range scanners with a focal distance larger than 2 meters. The representation of the operating principles for pulse-based time-of-flight 3D laser scanners is shown in Figure 2.1, and in Figure 2.2, the operating principles for phase-shift time-of-flight 3D laser scanners are represented.

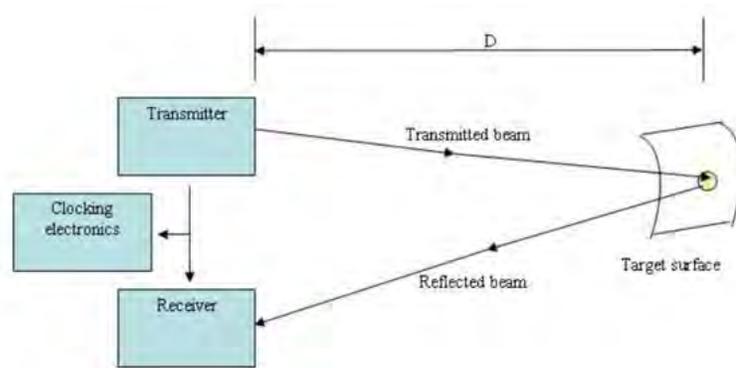


Figure 2.1: Representation of operating principles for pulse-based time-of-flight 3D laser scanners (Janos, July 2014).

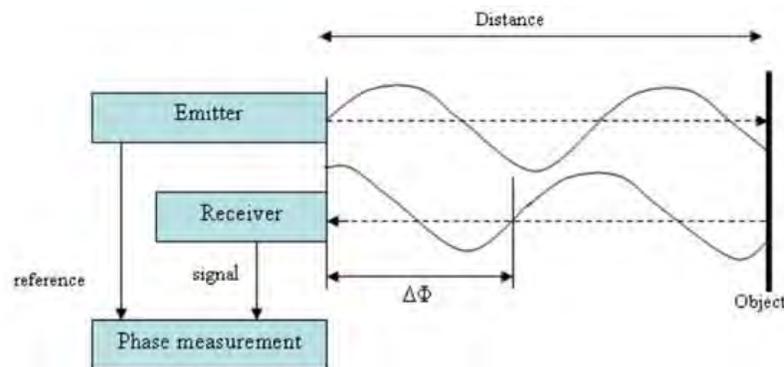


Figure 2.2: Representation of operating principles for phase-shift time-of-flight 3D laser scanners (Janos, July 2014).

Similar to time-of-flight 3D laser scanners, triangulation-based 3D laser scanners are also non-contact active scanners that use laser light to collect surface point locations. However, instead of a rangefinder, this scanner system uses cameras to exploit the location of the laser dot, which shines on the surface. These scanners are called triangulation-based 3D scanners because the laser dot, the camera and the laser emitter form a triangle. Since the distance between the camera and the laser emitter and their angle with each other and the third corner defined by the laser dot are known, it is possible to compute the size of the triangle that gives the location of the laser dot (Pfeifer and Briese, 2007). Triangulation-based 3D laser scanners are classified as short range laser scanners with a focal distance smaller than 1 meter. The representation of the operating principles for triangulation-based 3D laser scanners is shown in Figure 2.3.

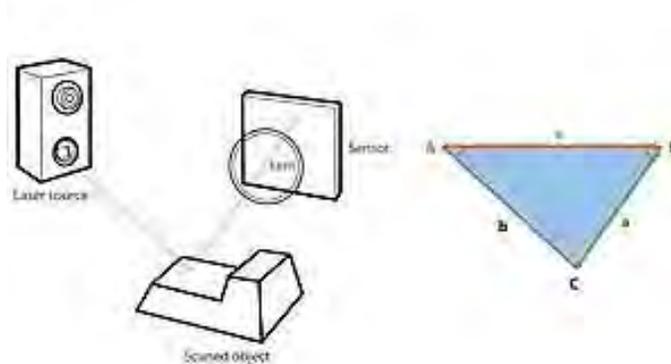


Figure 2.3: Representation of operating principles for triangulation-based 3D laser scanners (Geomatic 3D Systems Inc., May 2014).

The second group of non-contact types of laser scanners consists of non-contact passive 3D laser scanners. Non-contact passive 3D scanners, which are the second major non-contact type scanners, detect reflected ambient radiation; however, they do not emit any type of radiation. Since visible light is already an ambient radiation, it can be detected by non-contact passive 3D scanners. These types of scanners are generally less

expensive than the non-active 3D scanners since they do not need sophisticated hardware. Some of the available systems can be listed as stereoscopic and photometric.

## **2.2 Key Specifications and Sources of Error for Time-of-flight and Triangulation-based 3D Laser Scanners**

As mentioned, the two major types of laser scanners used in this research are time-of-flight laser scanners, which have a longer range, and triangulation-based laser scanners, which have a shorter range. Both types have their own advantages and challenges relative to their suitability for different applications. Since the main focus is on time-of-flight and triangulation-based 3D laser scanners, only the key specifications and sources of error associated with these types of 3D laser scanners are investigated.

### **2.2.1 Key Specifications**

There are three key specifications that define the performance characteristics of 3D laser scanners: accuracy, precision, and resolution. Error sources that affect these performance characteristics need to be investigated in order to understand the limitations of 3D laser scanners.

Accuracy is defined as the degree of agreement between a measurement and the conventional true value of the quantity being measured. Triangulation-based laser scanners are generally more accurate compared to the time-of-flight scanners. Timing the round-trip of a light beam is difficult due to the high speed of light; thus, the accuracy of the distance measurement is relatively low (Reshetyuk, 2006).

Precision, on the other hand, is the degree of repeatability in a measurement. Accuracy is generally confused with precision in the system specifications. Precision can be estimated if all the systematic errors influencing the 3D laser scanner measurements have been corrected, and the standard errors are taken into account. Precision is related to the hardware properties instead of the used system; thus, it is not possible to give a comparison between time-of-flight and triangulation-based scanners on a precision basis (Gordon and Lichti, 2004).

Finally, resolution is defined as the size of the smallest discernable feature that can be detected by the scanner. It is important to differentiate range resolution and angular resolution. Range resolution is the minimum detectable range change, whereas angular resolution is the smallest discernable feature size on a homogeneous surface. Angular resolution is directly dependent on the beam divergence angle and the sampling interval. Angular resolution is especially important for damage quantification. Low-resolution data is typically not detailed enough to extract important features. On the other hand, high-resolution data is harder to process and increases the computational cost. As computational power is improving rapidly, the ability to use higher-resolution lasers with real-time processing of data will increase (Reshetyuk, 2006).

### 2.2.2 Sources of Errors

The sources of errors that cause problems on the described key specifications can be divided into two main groups: internal (instrumentation) and external (object-related, environmental, and methodological errors) (Gordon and Lichti, 2004). A more distinctive

separation can be given as instrument, object-related, environmental and methodological errors (Staiger, 2005; Reshetyuk, 2006).

Instrument errors are dependent on the technical specifications of 3D laser scanners, where these technical specifications are directly affected from the scanner design. Instrument errors can either be fundamental, or they might be specific to the scanner hardware. Fundamental errors, which are related to the laser rangefinder or beam deflection unit, represent the natural limitations of laser scanning. However, errors specific to the scanner hardware, which are related to the laser rangefinder, beam deflection unit, and axes errors, can either be removed or minimized through better system design or calibration. This category contains an error type that is called “mixed-pixels”. This type of error is especially important since it might affect the results of the algorithms implemented in this research. As the laser beam hits the edge of an object, there is an inevitable loss in accuracy since a light beam splits into two separate pulses with different trajectories after it hits an edge on the structure. This results in recording two data points from two different locations for only one laser pulse. These inaccurate points are later removed from the point cloud before further processing; however, the object dimensions extracted from the resulting point cloud has accuracy issues (Tang et al., 2009).

Object-related errors are associated with the scanned surfaces. The most important cause of the error is reflectance. Since the 3D laser scanners receive the emitted laser beams that are reflected from the scanned surfaces, the reflectance of the underlying surface is an important parameter. Reflectance can be taken as the ratio between the reflected and incident power (Ingensand et al., 2003). Reflectance is dependent on the

following factors: material properties that include electric permittivity, magnetic permeability and conductivity; surface color; wavelength of laser; incidence angle of the laser beam; surface roughness; polarization; temperature; moisture of the surface (Nayar et al., 1991; Jelalian, 1992; Curless, 1999; Ingensand et al., 2003).

Some of the environmental errors can be listed as ambient temperature, pressure, relative humidity, illumination, vibration, etc. Understanding the influence of these factors on the measurements is important since they result in measurement errors, and they are difficult to control.

Finally, methodological errors are results of the followed surveying strategy. The three main sources of the methodological errors are the density of the laser spot that depends on the sampling resolution, the range to the object and the chosen approach for geo-referencing the points (Staiger, 2005).

### 2.2.3 Incomplete data

Another source of error, which causes problems for further processing steps, is incomplete datasets. Incomplete datasets can be a result of either incomplete laser scanning due to occlusion, or insufficient access to all parts of the structure. In this case, point cloud and image processing algorithms may result in false positives or can fail to detect important features that are mandatory for further processing. Thus, it is important to know the properties of a candidate point cloud so as to avoid any unexpected results, and also to develop methodologies that will overcome occlusion.

### **2.3 Laser Scanning and Image Processing Applications in Civil Engineering**

This section investigates the possible application areas for laser scanning and imaging in civil engineering. Current approaches used for routine inspections involve collecting information with a visual inspection carried out by expert personnel (Chang et al., 2003; Sohn et al., 2004). Thus, most methodologies are subjective and non-repeatable, and the documentation of inspections is difficult.

Laser scanners do not require direct access to the object, and this resolves some of the limitations of conventional sensors that have to be installed on structures (Vežočník et al., 2009). In recent years, usage of laser scanners and/or cameras for various engineering surveying applications such as monitoring field displacement, deformation monitoring, damage detection, and damage documentation has increased rapidly (Monserrat and Crosetto, 2008; Lindenbergh et al., 2009; Olsen et al., 2009; Rabah et al., 2013; Torok et al., 2013; Adhikari et al., 2014). In this context, cameras are mainly used for applications such as crack, corrosion and change detection etc. (Lee and Park, 2011).

### **2.4 Laser Scan Processing for Modeling and Damage Detection**

As mentioned, in order to perform damage detection on point clouds, first the investigated point cloud has to be divided into meaningful clusters. This is achieved by implementing several point cloud processing algorithms for object and/or surface extraction. This section explores previous research that focuses on point cloud processing.

Prior to point cloud processing, recorded point clouds are just 3D datasets that give the geographic location of each point on the surface of a structure. Objects and other

features in a scene and their relationship with each other can be deduced by implementing well-recognized point cloud processing steps. This information can then be used to determine possible damage locations on the structure, which is explained in Chapter 7 in detail.

The main steps of point cloud processing can be listed as registration, sharp feature detection, segmentation, and object detection. For this work, literature regarding data processing methods for all steps was explored, and some algorithmic extensions were developed and implemented.

#### 2.4.1 Registration

3D point clouds are generated through capturing multiple scans of a scene since a laser scanner's field of view does not cover the whole structure. In order to have a complete 3D point cloud of a structure, it is necessary to change the location of the laser scanner after each scan and then register all recorded scans with each other. Registration is a mandatory pre-processing step required when multiple scans are taken to capture a scene (Besl and McKay, 1992; Yu et al., 2001; Gelfand et al., 2005; Barnea and Filin, 2008).

Automatically computing a three-dimensional rigid transformation that aligns two point clouds is a problem that frequently arises in three-dimensional modeling of large-scale environments and in structural change detection (Smith et al., 2008). Researchers generally present robust algorithms that can estimate and verify this transformation even in the presence of widely-differing scanning viewpoints and substantial structural changes in the environment between scans.

The most common registration and refining method in literature is called “Iterative Closest Point (ICP)” (Menq et al., 1992; Yang and Medioni, 1992). ICP starts from an initial transformation estimate, and uses this estimate to map points from one scan onto the second scan in order to establish temporary correspondences between points in the two scans. These correspondences are then used to refine the estimate. The process iterates until a convergence criterion is met (Besl and McKay, 1992).

#### 2.4.2 Feature Detection

Detecting feature points is one of the most important steps of point cloud processing. Feature points are detected through either mesh or point based feature detection methods. A feature point can either be a part of a corner where three or more surfaces connect, or a line or edge that connects two surfaces. These features can be detected in a point cloud by searching through the points looking for those whose normal vectors vary greatly from their neighbors (Rabbani et al., 2006; Demarsin et al., 2007). High normal vector variation between neighboring points represents an abrupt change of surface orientation. Once the feature points are detected, the most distinctive points are labeled as key-points and these key-points are used for monitoring and surface reconstruction purposes. The feature extraction problem is especially related to surface reconstruction, which has important applications in laser range scanning.

There are two general approaches for processing raw point clouds for feature detection: point-based methods and mesh-based methods. The point-based methods begin processing a data set by directly utilizing the points themselves (Pauly et al., 2002; Daniels et al., 2007; Demarsin et al., 2007). However, mesh based methods begin by

creating a triangulation or surface reconstruction of the cloud (Benkő et al., 2001; Huang and Menq, 2002). Point based methods have the advantage of using the point set exactly as is, without introducing any smoothing or loss of sharp features. Also, when working from a point set directly, the step of generating a consistent mesh, which can be very difficult and time consuming for irregular or noisy point sets, is avoided (Vančo and Brunnett, 2007). There has been less work done on point-based methods compared to mesh-based methods, even though the point-based methodology in lieu of a mesh-based methodology allows processing of the point cloud without the difficulty of generating a mesh and introducing smoothing (Vosselman et al., 2004).

### 2.4.3 Segmentation

Numerous approaches have been explored for geometric surface modeling from unstructured point clouds. Surface modeling requires several of the mentioned preceding steps: registration, feature detection, and segmentation. In this section the focus will be on segmentation, the last step before surface reconstruction, which is performed by aggregating points with similar attributes together (Dorninger and Nothegger, 2007). A point cloud is segmented by labeling each point in a point cloud, so that the points belonging to the same surface or region are given the same label (Zhan et al., 2010).

There are mainly three categories of segmentation: edge-based segmentation, surface-based segmentation and hybrid segmentation. The edge-based segmentation algorithms use features including normal vector, gradients, principle curvatures, etc. in order to extract the points belonging to surface edges. Since each separate surface edge expresses different features, once the edge points are determined, they are tracked to

obtain surface borders. The points within borders are grouped together (Fan et al., 1987; Sappa and Devy, 2001; Meyer and Marin, 2004). The region-growing methods use continuous surfaces that have homogeneity or similar geometrical properties for segmentation (Besl and Jain, 1988; Pu and Vosselman, 2006; Rabbani et al., 2006). Finally, the hybrid segmentation uses more than one property to process uncategorized points in a scan. Those properties may be geo-metrical, image-metrical, or texture-metrical (Lucieer and Stein, 2005; Liu and Xiong, 2008).

Segmentation simplifies decision making and data analysis by representing compound objects, which are defined by multiple points, by segments. Further steps such as object detection, recognition or reconstruction also become easier once the complex objects are represented by segments extracted from point clouds. Segmentation also reduces the data volume significantly; it is especially effective when handling the individual segments only, instead of the original point cloud for post-processing.

#### *2.4.3.1 Region Growing for Segmentation*

The main segmentation method implemented for this research is “region growing”. This section provides background information on some of the key region growing-based segmentation studies. The purpose of region growing algorithms is to group the points that share similar properties such as smoothness constraint. Thus, the output of these algorithms is the set of clusters, where each cluster consists of a group of points that are considered to be a part of the same smooth surface.

Surface extraction through region growing is based on the assumption of surface coherence (Besl and Jain, 1988; Vieira and Shimada, 2005). This indicates that almost

every point sampled from an object's surface will be geometrically related to its nearby points despite the presence of noise and they will all lie near a single, smooth surface.

The concept of region growing, which is used to merge points in a scan into groups that can be approximated by surfaces, was first introduced for images by Besl (1988) and Besl and Jain (1988); thereafter adapted for gridded height data in Sapidis and Besl (1995). This approach was first implemented for segmenting dense, unstructured meshes (Vieira and Shimada, 2004; Vieira and Shimada, 2005), and then it was further extended into a method for automatic surface extraction from point clouds (Vieira and Shimada, 2009).

#### 2.4.4 Object Detection

Shape searching and object retrieval is the last and the most important step of point cloud processing for both laser-based modeling and automated damage detection from 3D range data. There are many approaches that have been investigating shape-based retrieval of 3D data; and they can be listed as computer vision (Pope, 1994), mechanical engineering (Kriegel et al., 2003), artifact searching (Rowe et al., 2001), molecular biology (Kastenmüller et al., 1998), and chemistry (Bruno et al., 1997). Research in vision and computer graphics has been dominating the 3D shape searching area so far, where researchers have specifically focused on the 'shape matching' problem. However, computer-aided design (CAD) and engineering applications of 3D shape searching require many different considerations other than the shape matching problem. In addition to shape matching, advanced clustering and automated classification methods are required (Iyer et al., 2005).

Many of the 3D shape searching techniques related with this research can be classified based on shape representations: global feature-based, graph-based, histogram-based and 3D object recognition-based.

Global feature-based methods use global properties of the 3D model such as moments, invariants, Fourier descriptors, and geometry ratios. In Zhang and Chen (2001) calculating these features effectively from the mesh representation of an object were demonstrated. The limitations of global feature-based methods are that they fail to discriminate among locally dissimilar shapes, cannot capture the specific details of a shape and are not very robust (Iyer et al., 2005).

An important shape characteristic of 3D models is the topology. Graph-based methods use topology for shape searching and retrieval. Generally, relational data structure such as graphs and trees are used to represent topology. Subsequently, a graph or tree comparison can be used to estimate the similarity between two shape representations. Tree comparison is faster and easier compared to graph comparison; however, most engineering components cannot be represented as trees. On the other hand, with the increase in graph size, graph comparison costs increase proportionally. Other applications include exact or inexact matching techniques applied on topological graphs. Exact matching methods work best in the absence of noise, whereas inexact matching methods compute a measure of similarity even in the presence of noise. Topological graphs allow representation of 3D models at multiple levels of detail and facilitate matching of local geometry (Iyer et al., 2005). Some 3D shape matching approaches use boundary representation (B-Rep) for determining similarity, B-rep is represented as a graph. Shape matching is achieved by determining similarity between

corresponding B-Rep graphs (Tangelder and Veltkamp, 2004). However, there are many other approaches that convert the surface representation of a 3D model into a simpler topology preserving representation such as Reeb graphs (Reeb, 1946), shock graphs (Cyr and Kimia, 2001) and skeletal graphs (Sundar et al., 2003). Even though these approaches are faster than B-Rep comparisons, they often result in the oversimplification of shape that causes major problems in large 3D databases. There are also some other approaches that use B-Rep graphs for deriving graph invariants such as number of nodes and edges, degrees of nodes, and eigenvalues directly for fast shape comparison.

Histogram-based techniques use sampled points on the surface of the 3D model to extract characteristics. Histograms or distributions are created based on the frequency of occurrence of these characteristics. A distance function, which compares histograms, is used to determine similarity. The number of sampled points is especially important in determining the accuracy and effectiveness of histogram-based techniques, higher accuracy can be achieved by using a larger number of sampled points. However, the robustness of the method is inversely related to the number of sampled points. Histogram-based methods can be collected under two main categories: shape histograms (Ankerst et al., 1999) and shape distributions (Osada et al., 2002).

Finally, context-based object detection approach consists of 3D object recognition-based methods. The computer vision community has studied 3D object recognition techniques extensively. There are many available methods that have been developed for 3D object recognition. Some of them are based on aspect graphs (Koenderink and Van Doorn, 1976), extended Gaussian images (Horn, 1984), superquadrics (Solina and Bajcsy, 1990), spin images (Johnson and Hebert, 1999), and

geometric hashing (Lamdan and Wolfson, 1988). Some applications use multiple images of each object are used to obtain the features of all surfaces and combined for model construction, thus the specific viewing direction is not affecting the object recognition (Yi and Chelberg, 1998; Lee and Park, 2011).

## **2.5 Laser-based Modeling**

Point cloud processing is the first step for understanding and deducing meaningful information from 3D range data. This process forms a solid foundation for developing methods that are useful for civil engineering applications since extracting important features from point clouds is vital.

Laser-based modeling, which is one of the major civil engineering applications for laser scanning, includes incorporation of laser-scan data with the generation or validation of both computer-aided design (CAD) models and building information models (BIMs). CAD models generally represent structures with sets of independent planer surfaces, whereas BIMs, which provides adjacency relationships between connected elements in a model, represent facilities in a semantically rich manner. The modeling of a BIM from a given point cloud involves three tasks, which are modeling the geometry of the components, assigning material properties and an object category to a component and finally, establishing relationships between components (Huber et al., 2011). Some key examples of laser-based modeling from literature are discussed in the following paragraph.

In a prior study, laser scanners have been used to augment measurements from other sensors to determine the geometry of existing structures (Gielsdorf et al., 2004). In

another application, Vosselman et al. (2004) reviewed several techniques that can be used for recognizing specific geometric shapes in order to implement an automatic processing of the point clouds. Industrial components are modeled from point clouds into surfaces without using triangulation but by using a feature-based strategy and surface feature-based strategy (Ke et al., 2006). Accurate 2D plan models of building interiors have also been created by an automated method in which 3D point cloud data was used as input (Okorn et al., 2010). Adan et al. (2011) discussed a method that converts raw 3D point data to a semantic model automatically. This method identifies objects in an indoor environment including walls, floors, ceilings, windows and doorways even in the presence of significant clutter and occlusion. Ip and Gupta (2007) used a partial 3D point cloud of an artifact for retrieving the CAD model consists of polygonal meshes for segmenting the point cloud and searched for a potential model match.

Arayici (2008) summarized the research on creating BIMs of existing structures using point clouds captured by 3D laser scanners. This study highlights that this process requires adapting automated data processing and pattern recognition techniques. Another extensive review for extracting as-built building information models from laser point cloud is given in Tang et al. (2010).

## **2.6 Laser-based Damage Detection**

The second major aspect relevant to civil engineering application, which follows laser-based modeling and is also the core focus of this research, is laser-based damage detection. In the past two decades, researchers developed several methodologies for using laser scanning technology for both monitoring structures and detecting damage.

The common characteristic of most of the current approaches used for monitoring is to measure with high accuracy displacements, strains, pressures, or related quantities of a small number of points or collect information with visual inspection carried out by expert personnel (Chang et al., 2003). Thus, most methodologies are based on a discrete sample instead of sampling the complete surface. Currently, available instruments such as linear variable differential transformers (LVDTs), cable-extension position transducers, accelerometers and other optical systems that are commonly used in laboratory or field testing of structures similarly capture position and displacement. They record measurements often with greater accuracy than common laser scanners, but these measurements are feasible only for pre-selected discrete points. However, laser scanners provide opportunity to develop new health monitoring methods, which do not require sensors mounted on structures (Mosalam et al., 2014).

The most common application is tracking user-defined key-points on structures over a time period. Key-points can either be reflectors, which are mounted on the surface of structures, or distinctive feature points that can be successfully located and monitored in each successive 3D data set. This application could be developed further for use in health monitoring of existing structures. Recently, work has been done exploring the use of lasers for tracking in-situ deformation of undamaged structural components, such as beams and columns (Hawarey and Falk, 2004; Gordon and Lichti, 2007; Park et al., 2007). In another study, in order to monitor civil infrastructure systems such as long span bridges and to obtain direct measurements of absolute displacement time history at predefined locations (laser tracking references), a new vision-based approach, which consists of high-resolution depth cameras, has been developed (Wahbeh et al., 2003).

Laser scanners are also used for applications such as long term deflection monitoring, gage-free stress and strain estimation, etc. Another application is long-term monitoring where deformations are measured at historical buildings to derive changes at different epochs. Commercial software is used for forming 3D models of buildings and this model is used for computation of deformations (Sternberg, 2006). Buckner et al. (2008) used laser scanners with wireless leave-in-place sensor motes for in-situ detection and for monitoring metal fatigue. Fatigue related changes in the surface were detected by the device. Experiments showed that this device can track early stages of fatigue development at monitored locations, when cracks are smaller than 50 $\mu$ m in length. It is also very important to test the accuracy of the measurements recorded with laser scanners and in order to test laser accuracy, a study is performed to investigate the capabilities of laser scanners for certain civil engineering applications. Two detailed laboratory measurements and some field measurements were performed for confirming the accuracy of stated by manufacturer and for examining the effect of different materials, colors and incident angle; and found that laser scanners limited accuracy may problems for engineering surveying applications (Berenyi et al., 2010). Another study was performed on a simply supported steel beam to compare displacements measurements obtained by laser scanners with other static deformations sensors such as linear variable displacement transducers (LVDTs), electric strain gages and long gage fiber optic sensors. The results show that maximum deflections estimated by laser scanners are less than 1mm and within 1.6% of the ones measured by LVDTs. This study also showed that the maximum stress value estimated by laser scanners are close to the directly measured value with long gage fiber optic fibers (Park et al., 2007). In addition, advanced lasers are able to complete on

the order of thirty complete scans per second, thus enabling these lasers to be used to record the movement of a structure during many types of dynamic experimental testing (Siringoringo and Fujino, 2009).

Another commonly investigated application for laser scanners is to track changes, mostly during construction, by comparing two successive scans that are recorded throughout the process. In some studies, the algorithms attempt to locate objects in the as-built geometry based on the BIMs of the as-designed geometry. Tracking of components within a point cloud during construction has been investigated in Bosche and Haas (2008) and Chi et al. (2009). In another study, laser scanners are used to conduct quality assurance. Modeled and the as-is conditions of the structure are compared to identify potential errors in the model (Anil et al., 2013). Bosché (2010) displayed a method for detecting 3D CAD model objects automatically in point clouds and for performing dimensional compliance control. Kim et al. (2013) performed an automated construction process measurement in three phases: alignment of the as-built data with the as-planned model, matching of the as-built data to information in the BIM, and revision of the as-built status. Son and Kim (2010) discussed a method that uses laser scanner data and images for 3D structural component recognition, in order to monitor the construction progress. Shih and Wang (2004) use point clouds in order to compare the work process at a construction site with the original construction schedule. Turkan et al. (2012) mentioned that even though the laser scanning technology has been investigated in construction industry, it couldn't reach to its full potential possibly due of the complexity of the available commercial 3D data processing software packages. Thus, their study

proposes a system that combines 3D object recognition with schedule information for tracking the construction progress.

A more recently developed application is to capture in situ damage or collapse of a structure by using laser scanners, in which individual laser scans of a scene may be captured from different viewpoints to permit the creation of a complete 3D record of a damaged structure (Olsen et al., 2009). These types of applications enable localization and quantification of surface damage in order to enhance the current visual inspection strategies. In Anil et al. (2013), laser scanners are used to represent crack information on the surface of structures using a BIM approach. Teza et al. (2009) used terrestrial laser scanner datasets to perform concrete surface damage recognition using mean and Gaussian curvatures of the surface of structures. Liu et al. (2011) used reference planes, which represent the intact condition of the surface of the investigated structure, in order to locate and to quantify the damaged areas. Laser scanning is used for bridge under-clearance measurements to assess damage and to help engineers to develop bridge improvement planning (Liu et al., 2011). Some other applications such as post blast assessment and bridge deflection measurements are also discussed in the extension of the same study (Chen, July 2014). There have been also some studies that investigate the performance of laser scanners for detecting the investigated damage types. In a study, the performance of laser scanners for detecting thin cracks for damage assessment of reinforced concrete frames is characterized (Anil et al., 2013). The limits for the terrestrial laser scanner-based crack detection were discussed in Laefer et al. (2014). The performance of several surface flatness detection algorithms that operate on concrete

surfaces and a set of laser scanners used to collect surface scans are evaluated in Tang et al. (2010).

Laser scanning technology is also used for condition assessment of wide areas as well. The terrestrial-LIDAR technologies are used to visualize the surface and structural deformations in Kayen et al. (2006). Olsen and Kayen (2014) discussed the challenges and benefits of using 3D laser scanning on post-disaster reconnaissance efforts. Kashani et al. (2014) used the terrestrial laser scanners to perform damage assessment on the structures affected from tornados and to use this information to estimate the wind speeds. Long-range Doppler wind light detection and ranging (LIDAR) are used to collect measurements at wind farms in order to perform applications such as mapping wind turbulence and wakes, analyzing turbine-to-turbine interaction, and etc. (Bingöl et al., 2010; Käsler et al., 2010; Trujillo et al., 2011; Hirth et al., 2014). In another application, terrestrial laser scanners are used to perform damage assessment after the 2010 Chile earthquake and tsunami (Olsen et al., 2012).

## **2.7 Image Processing for Damage Detection**

Although laser scanners provide dense data sets that cover the surface of structures, without image information it is not possible to detect some damage types directly from raw point clouds. These damage types include minor cracks and corrosion, which can be detected from images by using self-learning algorithms such as artificial neural networks (Kaseko and Ritchie, 1993), fuzzy-logic algorithms (Cheng et al., 1999) etc. Image processing for extracting feature information is a very well-researched area and its contents are extensive. Some of the possible feature extraction and damage detection

methods for images can be listed as geometric feature extraction, spectral feature extraction and textural feature extraction. It will be out of scope to discuss all of these methods in detail so only the main category names are listed. Additional examples of research conducted in this area are presented in the following paragraph.

Jahanshahi et al. (2011) have developed an integrated inspection software tool, which consists of inexpensive digital cameras for evaluating defect evolution in structures. Visual assessment of structure's condition is done with the cameras mounted on the bridge that are remotely controlled by an inspector. In order to supplement current bridge visual inspection, a new automated remote bridge-inspection technique called spatial integrated small-format aerial photography is proposed. This system offers a low-cost solution for bridge surface imaging. Images taken from airplanes, at an elevation 1000 ft. from the ground, are used for detecting cracks and joint openings on bridge decks and highway pavements (Chen et al., 2011). Abdel-Qader et al. (2003) provided a comparative study on four image-based crack-detection techniques performed on concrete images that can be listed as fast Haar transform (Haar, 1910), fast Fourier transform (Cooley and Tukey, 1965), Sobel (Sobel and Feldman, 1968) and Canny (Canny, 1986). Higgins and Turan (2011) used close-range photogrammetry to extract gusset plate geometry, in order to ensure that the drawings actually reflect the as-built conditions. An unsupervised principle component analysis (PCA) based algorithm is used to determine the crack locations in concrete bridge decks in order to automate the inspection strategies (Abdel-Qader et al., 2006). A wider area application is discussed in Chen and Hutchinson (2010), where satellite images are used to classify the urban structural damage.

Even though the major form of the input data is 2D images, 3D images are also used for automated damage detection. Instead of 2D image descriptors, 3D volume descriptors of the generated object model are used for crack detection (Torok et al., 2012; Torok et al., 2013).

## **2.8 Combined Laser and Image Processing for Damage Detection**

New developments in laser scanning technology now enable capturing 3D point clouds with color information, which can also be referred to as texture-mapped point clouds. This new development opens up new research possibilities, which have not been investigated much so far. Even if the number of examples is limited; there is research, where laser scanners coupled with cameras, are used for creating photo-realistic models of structures. This is especially common in the area of historical preservation. 3D laser scans and images are gathered together to create exact replicas of existing structures in digital environment for preserving cultural heritage.

Virtual geometric models from real buildings, terrains, or infrastructure systems are obtained by benefiting from the recent advances in laser scanning technology and using related 3D processing algorithms (Varady et al., 1997; Huber et al., 2011). Moreover, real texture mapped 3D models are obtained by mapping photographic images onto geometric models, which enables combined usage of lasers along with high-resolution cameras (Zalama et al., 2011). Li-Chee-Ming et al. (2009) managed to create these models by using image data automatic 3D point cloud registration, automatic target recognition used for geo-referencing, automatic plane detection algorithm used for surface modeling, and texture mapping. Li et al. (2008) used the LiDAR data and

imagery for performing post-earthquake assessment, in order to determine the building damage degree. General overview of remote sensing and GIS applications for damage assessment are discussed in Yamazaki (2001). An extended summary of the optical techniques, which include satellite images and LiDAR technology, is given in Olsen et al. (2013).

## **2.9 Current Inspection Strategies for Bridges**

The most recent ASCE Report Card for America's infrastructure, which depicts the condition and performance of the nation's infrastructure, was released in 2013 (ASCE., July 2014). In this report, the average grade for all infrastructure types was given as D+, which demonstrates the importance of assessing the current condition of our infrastructure and taking necessary precautions based on these up-to-date assessment results. Even though this health assessment is crucial for large structures such as bridges, power transmission lines or dams, which were graded as C+, D+ and D, respectively, such inspections are generally conducted by using the data collected from discrete sensor locations and/or visual inspections that are carried out by trained experts.

Both of the inspection methods described above has challenges. First, even if the sensors mounted on structures record accurate data for any measured quantity, they do not represent the complete surface behavior of the investigated structure since they are discretely located. Second, visual inspections often involve shutting down a portion of the structure and sending out trained experts for executing the inspections. This method is dangerous, time consuming, expensive, and subjective, since the results of each inspection depend on the personal judgment of the inspector.

Since the visual inspection of each structure type has different criteria, it is important to select a certain structure with well-defined inspection strategies as a baseline for categorizing particular damage types detected from laser scan data. Even though the developed damage detection algorithms can perform on different types of structures, for this research, visual inspection strategies for bridges are adopted since they consist of many exposed members..

The Federal Highway Administration (FHWA) requires that structure inventory and appraisal data should be entered into the State or Federal agency inventory within 90 days of the date of inspection for State or Federal agency bridges and within 180 days of the date of inspection for all other bridges. This data entry includes routine, in-depth, fracture critical, underwater, damaged and special inspections. It also includes existing bridge modifications that alter previously recorded data for new bridges and changes in load restrictions or closure status. The following Structure Inventory and Appraisal (SI&A) items receive an overall condition rating: Deck, superstructure, substructure, channels and culverts.

Inventory items pertain to the characteristics of a bridge. For the most part, these items are permanent characteristics that only change when the bridge is altered in some way, such as reconstruction or load restriction. Inventory items include identification, structure type and material, age and service status, geometric data, classification, load rating and posting status, proposed improvements, and inspection history. On the other hand, appraisal items are a judgment of a bridge component condition in comparison to current standards. Appraisal items are used to evaluate the structure based on the level of service it provides on the highway system. Appraisal rating items include condition rating

items, structural evaluation, deck geometry, vertical and horizontal under-clearances, waterway adequacy, approach roadway alignment, traffic safety features, and scour critical bridge status.

This research mainly concentrates on condition rating items, which compares the current physical state of the structure to what it was the day it opened, and structural evaluation, which gives the overall condition of the structure based on all major deficiencies, including its ability to carry the required loads. Sample condition rating guidelines from different states are adapted in this work to classify the damage severity and to assign labels to the detected damage in a well-known format. As an example, Table 2.1, which is taken from Manual Bridge Inspection Manual of Ohio, gives the summary of condition rating guidelines for the deck, superstructure and substructure. More detailed damage descriptions and condition rating criteria for several items are described in Chapter 9.

Table 2.1: Summary of condition rating guidelines for deck, superstructure and substructure (Ohio Department of Transportation., 2010).

<b>1-4 Worst Span</b>	<b>NBIS 9-0 – Deck, Superstructure and Substructure SUMMARY Items TOTAL BRIDGE</b>		<b>Inspector Guidelines</b>
<b>1</b>	9 - Excellent		Brief comments as appropriate
	8 - Very Good	No problems noted	
	7 - Good	Some minor problems	
<b>2</b>	6 - Satisfactory	Structural elements show some minor deterioration	Comments as appropriate
	5 - Fair	All primary structural elements are sound but have minor section loss, deterioration, spalling or scour	Document deficiencies quantitatively with descriptive comments.
<b>3</b>	4 - Poor	Advanced section loss, deterioration, spalling or scour	Candidate to establish monitoring parameters with specific locations to track the deficiencies rate-of-change at the next inspection. In addition to quantitative documentation take photos, make sketches and/or establish monitoring control points.
	3 - Serious	Loss of section, deterioration, spalling or scour has seriously affected primary structural components. Local failures are possible. Fatigue cracks in steel or shear cracks in concrete may be present	Document deficiencies quantitatively with descriptive comments; establish monitoring benchmarks. Discuss with Program Manager, structure may be prone to localized failures.
<b>4</b>	2 - Critical	Advanced deterioration of primary structural elements. Fatigue cracks in steel or shear cracks in concrete may be present or scour may have removed substructure support. Unless closely monitored it may be necessary to close the bridge until corrective action is taken	Document deficiencies quantitatively with descriptive comments; establish monitoring benchmarks. Contact Program Manager; candidate to dispatch repairs and or road closure. Confirm in writing.
	1 - Imminent Failure	Major deterioration or section loss present in critical structural components or obvious vertical or horizontal movement affecting structure stability. Bridge is closed to traffic but corrective action may put bridge back into light service	Dispatch personnel for immediate closure. Notify Program Manager. Confirm in writing.
	0 - Failed	Out of service - beyond corrective action	

### **3. Research Methodology**

This chapter discusses the research methodologies that have been implemented in this research. Figure 3.1 lays out the general research map. The steps for developing and implementing algorithms for damage detection by using laser scanners with and without integrated high-resolution cameras are discussed respectively in the following chapters entitled “Laser Data Processing towards Object Detection” (Chapter 5), “Object Detection for Modeling and Damage Detection” (Chapter 6), and “Camera Integrated Laser-based Surface Damage Detection” (Chapter 7). The developed methods are calibrated and validated using several point cloud dataset that are collected with laser scanners. Introductory information on the investigated datasets is given in Section 4.1.

In this chapter, first the detectable damage types, which can be located and quantified by using laser scanning and/or imaging, are discussed in Section 3.1. Then, the research strategies are laid out in Section 3.2.

#### **3.1 Summary of Detectable Damage Types**

Although laser scanner and camera technologies are improving rapidly and high-end solutions are getting less expensive, these types of sensors can only record surface data, thus restricting the detectable types of damage.

This research mainly focuses on detecting visible damage on concrete and steel structures and quantifying this damage in an automated fashion. Quantification of visible damage is important because extensive damage at a critical location or member in large structures can cause large changes in stresses at other locations. Possible concrete damage types that can be captured with laser scanners include cracking, spalling and

cross-section changes; and steel damage types include corrosion, cracking, bent members, ruptures, points of discontinuity and cross-sectional changes. Table 3.1 outlines the possible damage types that can be detected with laser scanners and/or cameras along with the point cloud processing and image processing steps that should/could be followed for each input data type for detecting a certain type of damage.

In this research, we are focused on developing algorithms for detecting damage types that are associated with significant volumetric change and/or the damage types that can be differentiated by using texture information. Investigated damage types include large cracks (cracks with widths smaller than approximately 5 mm, depending on the laser sensor used, are not detectable), corrosion, bent members and alignment issues, points of discontinuity, ruptures, and spalling.

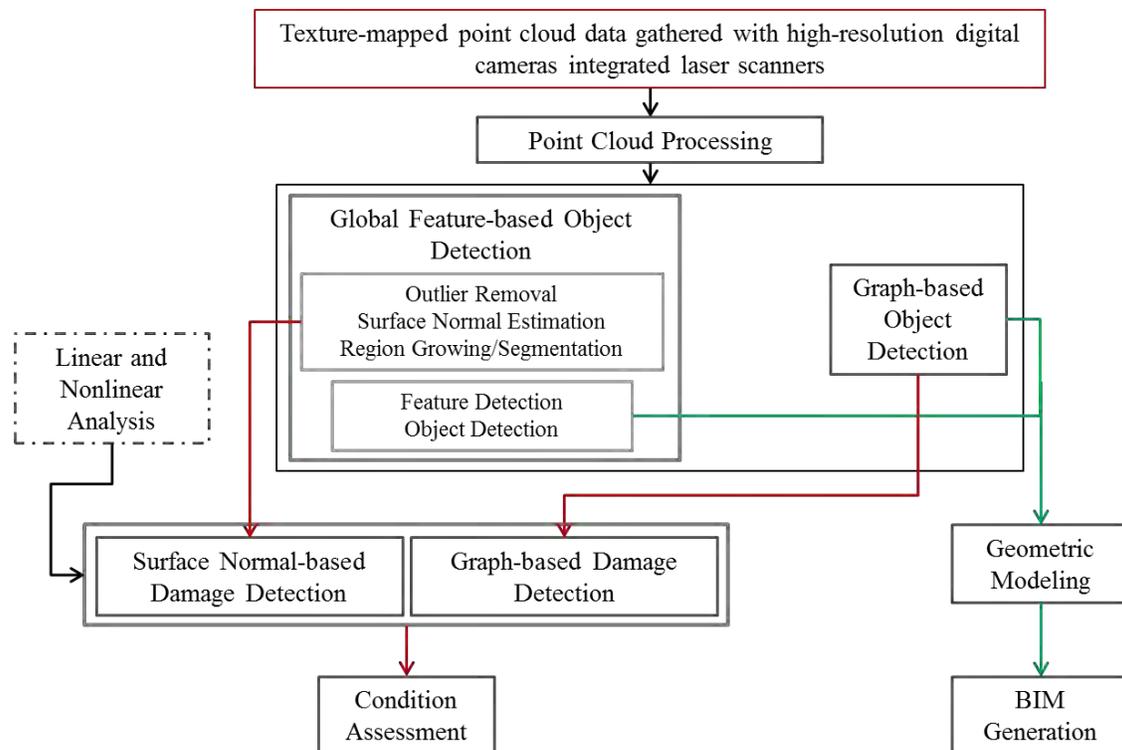


Figure 3.1: Research map.

Table 3.1: Detectable damage types and methods for detection.

Type of Visible Damage	Laser Scanners		Images	Point Cloud Processing	Image Processing	Optional Pre-Data	Required Pre-Data	
	Close Range	Long Range				Point Clouds	Point Clouds	Images
Cracks and tears	Yes	Depends on the resolution	Camera Distance, Camera Orientation and Resolution	Depth Images	Region Based Segmantation, Chain Codes, Boundary Following, Boundary Descriptors, Regional Descriptors, Patterns and Pattern Classes	N/A	N/A	Training images
Corrosion	Yes	Depends on the resolution	Camera Distance, Camera Orientation and Resolution	Depth Images	Region Based Segmantation, Chain Codes, Boundary Following, Boundary Descriptors, Regional Descriptors, Patterns and Pattern Classes	N/A	N/A	Training images
Bent members and alignment issues	Yes	Yes	N/A	Registration, Outlier Removal Keypoint Detection, Feature Detection, Segmentation for Region Growing, Object Detection	N/A	BIM/FEM	Object Library	N/A
Points of discontinuity or restraint	Yes	Yes	N/A	Registration, Outlier Removal Keypoint Detection, Feature Detection, Segmentation for Region Growing, Object Detection	N/A	BIM/FEM	Object Library	N/A
Ruptures	Yes	Depends on the resolution	Camera Distance, Camera Orientation and Resolution	Registration, Outlier Removal Keypoint Detection, Feature Detection, Segmentation for Region Growing, Object Detection	Region Based Segmantation, Chain Codes, Boundary Following, Boundary Descriptors, Regional Descriptors, Patterns and Pattern Classes	BIM/FEM	Object Library	Training images
Spalling	Yes	Yes	N/A	Registration, Outlier Removal Keypoint Detection, Feature Detection, Segmentation for Region Growing, Object Detection	N/A	BIM/FEM	Object Library	N/A

### **3.2 Research Strategies**

In this research, the aim is to develop new methods coupled with existing algorithms for detecting different types of damage on structures. This requires laying out algorithmic steps that are needed for each damage type separately.

As represented in Figure 3.1, regardless of the damage type, it is necessary to go through point cloud processing steps to extract important features from 3D point clouds. Thus, the first aim is to detect objects and/or surfaces from raw 3D point clouds by implementing algorithms for each step of point cloud processing, which includes registration, neighborhood size selection, outlier removal, curvature estimation, extraneous point removal, feature detection, region growing for segmentation, and object detection. Detected object properties can either be used for developing strategies for damage detection or for geometric modeling.

As stated in Section 1.1, detection of damage is addressed in three different categories related to the level of deformation and damage seen in the structure. These three categories are small deformations (enhanced visual inspection), large deformations with no change in topology (automated damage detection), and large deformations with localized change in topology (automated damage detection and geometry reconstruction). Table 3.1 represents the damage types that are in the scope of this research, along with the point cloud processing and image processing steps that should/could be followed for detecting a certain type of damage. The following sections briefly summarize the research methodologies that are developed for each of the three damage levels.

It is also possible to use the as-is representations of individual elements or the surfaces, which are extracted from point clouds, to create a new finite element model or

alter an existing finite element model based on the detected surface damage. These models can be analyzed under both static and dynamic loading to assess the current condition of the investigated structures in detail. Finite element model generation is not in the scope of this research; however, it is discussed in the “Future Work” chapter, Chapter 11.

### 3.2.1 Methods for Small Deformations

Some of the small damage types can be listed as cracks, corrosion, etc. These damage types can be detected by processing texture-mapped point clouds, which include color information as well as the geographic locations of points.

The locations of the defect points are extracted using the damage detection strategies discussed in Chapters 7 and 8. The defect points are clustered, and each cluster, which represents a defect, is quantified automatically.

Either one or a combination of soft computing methods, such as unsupervised/supervised classification, minimum distance classification, maximum likelihood classification, fuzzy-logic algorithms, artificial neural networks, or genetic algorithms, can be used for locating and quantifying the cracks. Since the main goal is to achieve automated detection, this research is focused on unsupervised and self-learning algorithms. In this research, an artificial neural network is implemented for automated crack detection, and the details are given in Chapter 8.

This research focuses on detecting large cracks and corroded areas, where change in surface topology is observed.

### 3.2.2 Methods for Large Deformations with No Change in Topology

It is possible to detect large deformations with no change in topology by first individually detecting every element from the processed laser scan. This can be achieved by extracting feature information from the point clouds. These features are used for clustering the investigated point cloud into discernible elements. These clusters are matched with the objects from a pre-built library using context-based shape matching and object retrieval methods. Detected objects are gathered together to create a geometric model of the structure. Each point cluster is compared with the corresponding intact object, in order to locate the large deformations.

The comparison between detected clusters and the objects from the library is completed by converting their point cloud representations into voxel grids. Each voxel grid will be treated as a stack of 2D binary images. In this context, each binary image represents the perimeter of the cluster or the object at a certain distance along the length of the member. The perimeter values obtained from the clusters, and the library objects will be compared with each other, and this will be used to document both the location and the severity of the damage.

The most significant example of this damage type is bent members. Our focus is on determining any misalignment in each individual member.

### 3.2.3 Methods for Large Deformations with Localized Change in Topology

As mentioned in Section 1.1.3, it is possible to observe localized changes in the structure's topology even though the majority of the topology of the structure has

remained intact. It is important to locate and quantify this damage, in order to take the necessary precautions before further damage occurs.

Several surface- and component-based damage detection strategies are developed for defect detection, localization, and quantification. These strategies are discussed in Chapters 7 and 8 in detail. For surface-based damage detection, the locations of the defects are extracted by using the estimated and/or initial surface properties. The points that deviate from the expected processing results are labelled and then, these damaged points are automatically clustered and quantified. For element-based damage detection, cross-sectional properties extracted from the investigated point clusters are used to determine the object type. Once the object is recognized, the cross-section information is used to automatically compute the local defects that exist on objects. The damage detection is performed at the cross-section level and then, the findings are extracted to compute the overall damage in 3D.

This last section covers the final three damage types that are included in our scope: ruptures, points of discontinuity, and spalling.

## 4. Equipment Specifications

3D scanning technology as discussed in Chapter 1 is used to capture 3D data clouds that represent the whole surface of the structure. This captured data can be used to create a digital model for reverse engineering, quality assurance, inspection, CAD-to-part comparison, factory planning, investigation, and automatic object and damage recognition in modelling. This chapter focuses on the specifications of the 3D scanners utilized for this research and the resulting properties of the collected point clouds, which are presented in Section 4.1.

In this research, three different 3D scanners were used to capture point clouds from varying specimens. The list of the 3D scanners used in this work is given in Table 4.1.

Table 4.1: List of the available laser scanners.

<b>Laser Scanner</b>
FARO Photon 80
FARO Focus 3D
Microsoft Kinect

Both terrestrial laser scanners manufactured by FARO Technologies Inc., including the FARO Photon 80 and the FARO Focus 3D, use phase-shift measurement technology. The principle of this method is to compare the phases of the emitted and the received signals. The distance is measured by analyzing the shift in the wavelength of the return beam. These types of terrestrial laser scanners emit a periodical signal of moderate intensity.

The FARO Photon 80 was launched by FARO Technologies Inc. at the end of 2008. The specifications of this terrestrial laser scanner are given in Table 4.2. The range values given in Table 4.2 depend on ambient light for this scanner. At moderate ambient lightning, a full range is observed on 90% matte white surfaces. Tests also showed that

bright sunshine shortens the actual scanner range. Another important point is that the systematic error given in Table 4.2 is measured on stationary orthogonal 90% reflectivity reference paper in averaging mode (FARO Technologies Inc., July 2014).

The second laser scanner is the FARO Focus 3D, which was released by FARO Technologies Inc. at the end of 2011. At the time of its release, this terrestrial laser scanner was the smallest and the most advanced laser scanner on the market, with a built in multi-sensor hardware feature, which is a combination a height sensor with a compass to complement the device's existing dual axis compensator. This new sensor improves the automatic registration and reduces the post-processing work since it defines the height and the orientation against a set of points for each scan. Similar to its predecessor, the range values listed in Table 4.2 are dependent on ambient light and this might act as a source of noise. Bright ambient light (e.g., sunshine) may shorten the actual range of the scanner to lesser distances. The range can even be more than 120 m for normal incidence on high-reflective surfaces in low ambient light. The ranging error is defined as the systematic measurement error around 10 m and 20 m (FARO Technologies Inc., July 2014).

Microsoft Kinect uses an infrared (IR) projector and sensor system (one non-RGB and one RGB camera), which is a common combination in both industrial manufacturing and inspection applications. The IR projector throws out infrared light so that the other non-RGB camera captures the depth information in 3D space. There is a specific angle between the emitter and the sensor in order to recover the depth from triangulation. Some of the key specifications of the Microsoft Kinect are given in Table 4.2.

Table 4.2: Equipment specifications.

	<b>Properties</b>	<b>FARO Photon 80</b>	<b>FARO Focus 3D</b>	<b>Kinect</b>
Ranging Unit	Minimum Range (m)	0.6	0.6	0.8
	Maximum Range (m)	76	120	3.5
	Measurement Speed	120000 (points/sec)	976000 (points/sec)	30 (frames/sec)
	System Distance Error	±2mm at 25m	±2mm	±1 cm at 2m
Color Unit	Resolution (pixel)	Optional	Up to 70x10 <sup>6</sup>	640 x 480
	Dynamic Color Feature	N/A	Automatic adaptation of brightness	N/A
Deflection Unit	Vertical Field of View (°)	320	305	40
	Horizontal Field of View (°)	360	360	58
	Vertical Resolution (°)	0.009	0.009	N/A
	Horizontal Resolution (°)	0.00076	0.009	N/A
	Angular Resolution (horizontal/vertical)	±0.009	N/A	N/A
	Maximum Vertical Scan Speed (rpm)	2880	5820	N/A
Laser (Optical Transmitter)	Laser Power (CW average, mV)	20	20	N/A
	Laser Class	3R	3R	1
	Wavelength (nm)	785	905	750 - 900
	Beam Divergence (°)	0.009	0.011	N/A
	Beam Diameter at Exit (mm)	3.3, circular	3.0, circular	N/A
Multi-sensor	Inclination Sensor Accuracy (°)	0.1	0.015	N/A
	Inclination Sensor Resolution (°)	0.001	N/A	N/A
	Inclination Sensor Range	±15	±5	±27
Hardware Specifications	Ambient Temperature (°C)	5-40	5-40	0-40 (indoor)
	Humidity	Non condensing	Non condensing	N/A
	Weight (kg)	14.5	5	1.4
	Power Supply	Battery – 6 hours	Battery – 5 hours	USB 2.0

## 4.1 Available Datasets

In order to test the accuracy and the efficiency of the developed algorithms for both object and damage detection, several datasets have been collected and processed. The complete list of the datasets and their properties such as the total number of points, location, 3D scanner used for data collection, and color data availability are given in Table 4.3.

The first data set is a synthetic point cloud that is used for validating the effectiveness of the point cloud processing algorithms. This data set was used to validate the effectiveness of the point cloud processing algorithms described in Chapter 5 and also the global feature-based object detection method given in Section 6.3. The results are used to explain the algorithms in Chapter 5 and Section 6.3.

The second dataset is from a small steel frame experimental test specimen that consists of three C-sections. This dataset was collected with the Microsoft Kinect, so the accuracy of the data is worse than the ones collected via terrestrial laser scanners. This point cloud was mainly used to calibrate the graph-based object detection method described in Section 6.4. It was also used to validate the efficiency the graph-based damage detection algorithms given in Section 7.2 to show that the algorithms are working on the steel testing frame. The results are given in Chapter 10.

The third dataset is coming from the rocking frame experimental test specimen that was tested at Multi-axial Full-scale Substructure Testing and Simulation (MUST-SIM) facility at the University of Illinois at Urbana-Champaign (Eatherton et al., 2014). Only a portion of this dataset was used to validate the extraneous point removal algorithms

described in Section 5.6 and to test the graph-based object detection method given in Section 6.4. The results are given in Chapter 10.

The fourth dataset is collected from a concrete frame experimental test specimen that was used for research on progressive collapse of reinforced concrete structures (Sasani et al., 2007). This data set was used to calibrate and validate the damage detection algorithms described in Chapter 7. The details of the test and the results of the damage detection algorithms are given in Chapter 10.

The fifth point cloud data is collected from a collapsed bridge that was located in Dekalb County, IL (Borello et al., 2009; Borello et al., 2010). A portion of this dataset was used to test the global feature-based object detection method described in Section 6.3 and also the damage detection algorithms given in Chapter 7. The details on the bridge and the results of the damage detection algorithms are included in Chapter 10.

The final dataset is recently collected from one span of the Bowker Overpass, which is located in Boston, MA. This dataset was processed both for object detection given in Chapter 6 and for damage detection described in Chapter 7. All the necessary information on the overpass and the results are included in Chapter 10.

Table 4.3: List of the available laser data sets.

<b>Name</b>	<b>Total Number of Points</b>	<b>Location</b>	<b>Device Used for Scanning</b>	<b>Color Data</b>
Synthetic Point Cloud	27,370	N/A	N/A	N/A
Steel Testing Frame	893,937	Northeastern University, Boston, MA	Microsoft Kinect	N/A
Rocking Frame	1,223,416	University of Illinois, Urbana-Champaign, Urbana, IL	Faro Photon 80	N/A
Concrete Testing Frame	1,215,389	Northeastern University, Boston, MA	Faro Focus 3D	Available
DeKalb County Collapsed Bridge	6,131,734	DeKalb County, IL	Faro Photon 80	N/A
Bowker Overpass	Over 40 billion	Boston, MA	Faro Focus 3D	Available

## **5. Laser Data Processing towards Object Detection**

Point cloud processing begins with extracting information from each individual point of the complete dataset and then employing relationships between these points to derive properties of the underlying surface and object. In order to achieve this, it is required to pick appropriate methodologies for each of the associated point cloud processing steps, including: registration, neighborhood size selection, outlier point removal, curvature estimation, extraneous point removal, feature extraction, region growing and segmentation, surface fitting and object detection. This chapter discusses the details of the listed point cloud processing step towards object detection. The object detection methods are later discussed in Chapter 6.

For initial algorithmic development in this chapter, three of the datasets listed in Table 4.3 are used. The most frequently used data set is the synthetic point cloud. This synthetic point cloud consists of simple surfaces and it serves as an example to check basic functionality of algorithms. In addition to the synthetic point cloud, the point clouds of both rocking frame and concrete testing frame were processed to determine and address the challenges of actual laser scan data.

### **5.1 Registration**

Registration, which aligns and combines multiple data sets into a single set of range data, is a pre-processing step required if multiple scans are taken to capture a scene. Scan registration is now a tool that is commonly available as part of the laser manufacturer software programs.

For this research, three separate 3D scanners are used. Detailed explanation of the used hardware and their specifications are given in Chapter 3. Each 3D scanner has its own registration software that uses different point cloud registration strategies. Thus, the details of the registration software for each scanner are described below. The list of the laser scanners and their corresponding registration software is given in Table 5.1.

Table 5.1: List of laser scanners and corresponding registration software.

<b>Laser Scanner</b>	<b>Registration Software</b>
Faro Photon 80	Scene (2009 version)
Faro Focus 3D	Scene 3D
Microsoft Kinect	ReconstructMe

Both Faro laser scanners use registration software called Scene, which is software provided by FARO Technologies Inc. This software provides automatic search for reference spheres and black and white reference targets, which are mounted several locations around the investigated structure. These targets are used as the keypoints for the registration software and successive scans are automatically combined together. The new versions of the software also provide targetless scan replacement by automatic identification of several sharp features. However, due to accuracy concerns, this method is not utilized for the point clouds used in this research. The new version of the software also provides automatic coloring of the scans with high-resolution color photographs coming with FARO color option (FARO Technologies Inc., July 2014).

The software used with the Microsoft Kinect sensor is called ReconstructMe (PROFACTOR., July 2014). It is a 3D real-time scanning system developed by PROFACTOR. ReconstructMe has the same principles of an ordinary video camera; 3D scan is constructed by moving the object to be modeled. An average rate for scanning with compatible scanners is usually 20-30 frames per second. This software is also

capable of capturing and processing the color information of the object being scanned, if the sensor provides the necessary color stream. However, the best scanning results are obtained for non-planar and non-symmetric objects, which make it an inefficient solution for scanning the symmetric objects that are common in civil engineering structures. Also, it is not compatible with terrestrial laser scanners, and it is mainly developed for indoor usage.

The provided registration processes identified above were sufficiently accurate; therefore, additional work on registration was not warranted.

## 5.2 Neighborhood Size Selection

The next step after registration, in point cloud processing, is to select a local neighborhood of points around each point in order to extract local properties. This method is the basis of all the following processing steps, including outlier removal, feature extraction and object detection, since it provides information on local properties of each point in an unorganized point cloud.

In this work, the ‘ $k$  nearest neighbors’ ( $KNN$ ) method is used (Hoppe et al., 1992; Rabbani et al., 2006; Vančo and Brunnett, 2007). For a given data set of  $P$  of  $P_n$  points, a  $KNN$  search establishes the  $k$  closest points within  $P$  to a query point  $p_q$  or set of points.

For this research, a special case of the Minkowski metric, shown in Equation 5.1 below, is used as the distance metric  $d_{st}$  to find the neighborhood points for each  $p_q$ , for a given  $k$ . This special case is also denoted as the ‘Euclidian distance’ and is established in Equation 5.2 by using  $p = 2$  in Equation 5.1. Given an  $m \times n$  data matrix of a point cloud  $P_x$ , which is examined as  $m \times (1 \text{ by } n)$  row vectors  $x_1, x_2, \dots, x_{mx}$ , and  $m \text{ by } n$

data matrix of a point cloud  $P_y$ , which is examined as  $my$  (1 by  $n$ ) row vectors  $y_1, y_2, \dots, y_{my}$ , the various distances between the vector  $x_s$  and  $y_t$  are calculated as follows:

$$d_{st} = \sqrt[p]{\sum_{j=1}^n |x_{sj} - y_{tj}|^p} \quad 5.1$$

$$d_{st}^2 = (x_s - y_t) \cdot (x_s - y_t)' \quad 5.2$$

The neighborhood is defined as the  $k$  closest points;  $k$  can either be defined by the user or determined adaptively. The local surface properties related with each investigated point are dependent upon the size of the neighborhood  $k$ . The size of the neighborhood was initially set as a minimum value. Later, this neighborhood size is incrementally increased to determine the optimum value that would provide adequate robustness to measurement noise while still reflecting local surface properties (details are given in Section 7.1.1). The size of the neighborhood  $k$  has to be sensitive to the level of detail required by the application. For example, if the aim is to detect cracks from the surface data, then  $k$  needs to be small enough to capture those.

Deciding the neighborhood size  $k$  that would reflect the characteristics of the query point  $p_q \in P$ , for a certain feature representation effectively, is a common issue for the computer graphics research community. Pauly et al. (2003) proposes a solution that computes several feature values over multiple-scales and then, track the jumps in the feature value curve. For example, if the surface normals are used as the feature representation, the jumps in the surface curvature estimate  $\gamma_p$ , which is described in Section 5.5, can be used to look for the locations where strong deviations in the normal

direction occur. This information can then be used to determine the critical size of the neighborhood.

For this research, the neighborhood size is automatically computed for each defect type by using the surface data resolution. It should be noted that the point cloud resolution poses a threshold for the minimum detectable damage size. The computation process for the  $k$  values for varying defect types, and their effect on the obtained results are discussed in Chapter 7.

### **5.3 Outlier Point Detection**

Point clouds that are captured by laser scanners often have varying point densities. The overall accuracy of the captured point clouds are affected by several factors such as the sparse outliers due to measurement errors, outliers due to the splitting of laser beams at the edge of an object (a phenomenon known as jump edges, depth discontinuities, or occlusion boundaries), etc. Since these outlier points could affect the estimated local point cloud characteristics such as surface normals or curvature changes, leading to errors in later processing steps, it is critical to identify and remove them early in the process.

For this work, two outlier methods are implemented, including a radius-based outlier algorithm and a statistical outlier removal algorithm. Details of these two methods are discussed below.

### 5.3.1 Radius-based Outlier Removal

The radius-based outlier removal algorithm by Rusu et al. (2008) is the first method that is implemented. For radius-based outlier removal, after the local neighborhood size  $k$  is determined, the Euclidean distance to each point in the neighborhood  $d_n$  is computed for each individual point  $p_q \in P$  and stored.  $P$  represents the unfiltered point cloud. A search radius  $r$  for outlier removal is then defined by the user. This  $r$  value is used to create a virtual sphere around the point of interest. The distance  $d_n$  for each neighborhood point is checked to see if the entire neighborhood points for  $p_q$  are falling into the sphere defined by  $r$ . At the end, any point that fails to meet this criterion was removed from the original  $P$ .

In Figure 5.1, the process of radius-based outlier removal is demonstrated in 2D. For the example given in Figure 5.1, the Euclidian distance is labelled as  $d_n$  the search radius is shown as  $r$  and the neighborhood size  $k$  is equal to 4. Black, green and blue points are investigated. For the given  $r$  and  $k$ , neither the black nor the blue points are satisfying the criterion, which states that all the neighborhood points should fall inside the sphere defined by  $r$  and centered on the green point. In this case, both black and blue points are labelled as outliers and removed from  $P$ . Green point is the only point of interest that meets the given criterion.

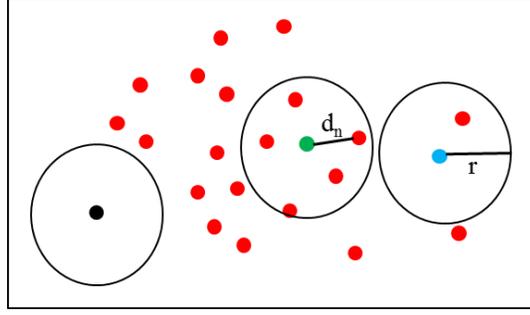


Figure 5.1: The demonstration of radius-based outlier removal algorithm after O'Artista and Beintum (2010).

### 5.3.2 Statistical Outlier Removal

The second method implemented for outlier removal is statistical outlier removal, which is also discussed in Rusu et al. (2008) in detail. This method uses the distribution of point-to-neighbors' distances in  $P$  in order to identify the outlier points. This distribution is assumed to be Gaussian with a mean and standard deviation. Thus, the mean Euclidean distance  $\bar{d}_n$  to each point in the neighborhood is computed for each point in the point cloud. The mean  $\mu_d$  and standard deviation  $\sigma_d$  of  $\bar{d}_n$  are then computed in order to determine significant characteristics of the entire point cloud. The main purpose is to retain all points whose  $\bar{d}_n$  to the  $k$  closest points is similar to the  $\mu_d$ , which is computed for the rest of the points. Thus, if  $\bar{d}_n > \mu_d + n \cdot \sigma_d$  or  $\bar{d}_n < \mu_d - n \cdot \sigma_d$ , where  $n$  is the standard deviation multiplier that restricts the density for any point, the point is removed from the data set. The remaining point cloud  $P'$  is given by Equation 5.3, where  $p_q$  represents the query point.

$$P' = \{p_q \in P | (\mu_d - n \cdot \sigma_d) \leq \bar{d}_n \leq (\mu_d + n \cdot \sigma_d)\} \quad 5.3$$

Figure 5.2 shows an illustrative example for the effects of the statistical outlier removal on a sample dataset. The statistical outlier removal is applied to a point cloud data of 18695 points  $P_n$ . The  $k$  for the example is taken as 16, with  $n = 1$ . The resulting outlier set is 496 points  $O_n$ , and  $P'$  has 18199 points. Figure 5.2 represents the plot for both raw and reduced data sets, where the vertical axis shows  $\bar{d}_n$  for each point and the horizontal axis is used to show the point index. Blue bars belong to the raw data set and red bars represent  $\bar{d}_n$  values for the resulting  $P'$ . It can be observed from the graph that the reduced point cloud  $P'$  has a uniform  $\bar{d}_n$  value for the remaining data set. Table 5.2 summarizes the information discussed in this paragraph.

Table 5.2: Statistical outlier removal example.

$P$	$k$	$n$	$O_n$	$P'$
18695	16	1	496	18199

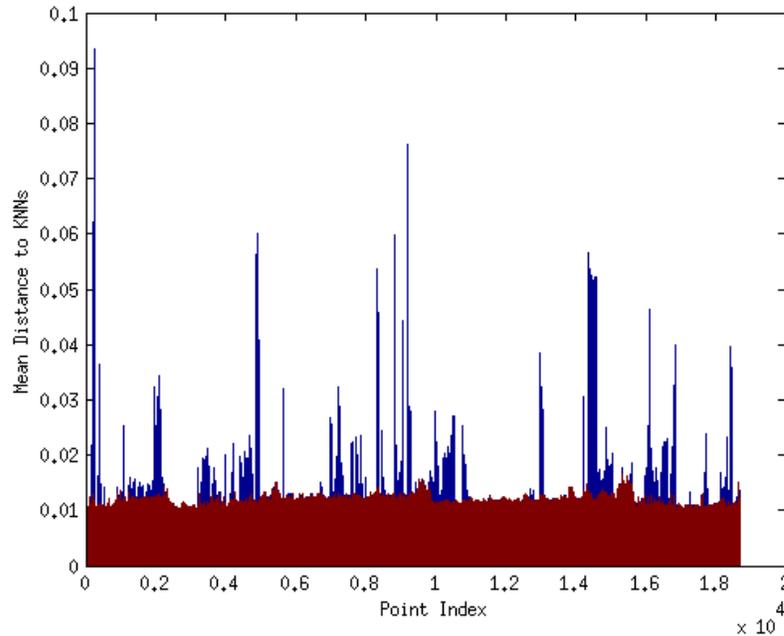


Figure 5.2: Results of statistical outlier removal example.

### 5.3.3 Comparison Between Radius-based and Statistical Outlier Removal Methods

The outlier removal is the second step after *KNN* for both the point cloud processing methods towards object detection and the developed damage detection algorithms. It is significantly important because the results of the further steps are directly affected by the results of the outlier removal.

In order to represent the differences between the two proposed methods; an example, which compares the results of the radius-based and statistical outlier removal methods, is discussed in this section. Figure 5.3 represents the data used for this example. Data is extracted from one of the columns of the point cloud of the concrete testing frame; the extraction location is shown with a red rectangle in Figure 5.3(a), and Figure 5.3(b) represents the point cloud. Figure 5.4 represents the results after the outlier removal, where black rectangles indicate the locations of the close-up views shown in Figure 5.5. The summary of the results are given in Table 5.3.

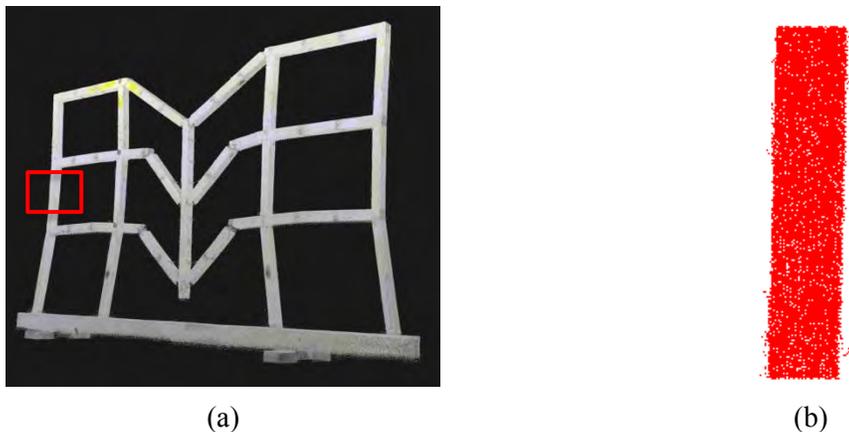


Figure 5.3: (a) The complete scan of the collapsed test setup; and (b) the raw point cloud data used in comparison example.

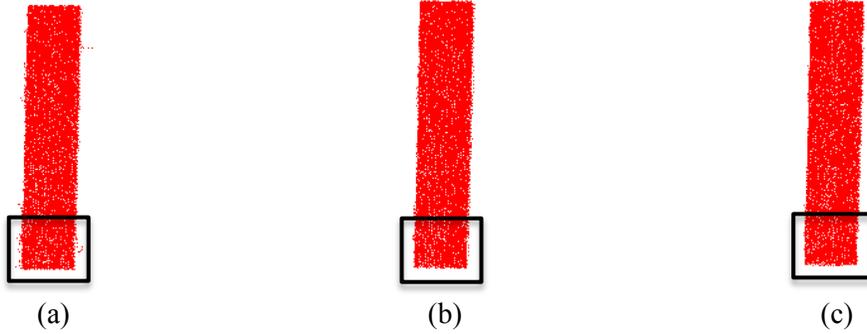


Figure 5.4: (a) Raw point cloud; (b) resulting data set after radius-based outlier removal; and (c) the resulting data set after statistical outlier removal.

Table 5.3: Comparison between radius-based and statistical outlier removal methods.

<b>Method</b>	<b><math>P</math></b>	<b><math>k</math></b>	<b><math>r</math> (in)</b>	<b><math>n</math></b>	<b><math>O_n</math></b>	<b><math>P'</math></b>
Radius-based Outlier Removal	18695	16	0.25	N/A	188	18507
Statistical Outlier Removal	18695	16	N/A	2	215	18480

The raw point cloud has a significant number of outlier points (Figure 5.5(a)), whereas the processed point clouds shown in Figure 5.5(b) and Figure 5.5(c) have a notably fewer number of outlier points. It should also be noted that statistical outlier removal has better results compared to radius-based outlier removal. Since the statistical outlier removal method fits a Gaussian distribution to the neighborhood points to compute the necessary parameters for point removal, this method automatically includes the effects of resolution changes on the surface of structures in the computations. However, the parameters of the radius-based outlier removal are user-defined and they are not dependent on the point resolution. Thus, even though the number of the outlier points  $O_n$  is similar for both of the described methods, it can be observed in Figure 5.5(b) and Figure 5.5(c) that the radius-based outlier removal method is not as effective as the statistical outlier removal method, since some of the outlier points are still present in the end result obtained by using radius-based outlier removal. The outlier locations are

marked with the black rectangles in Figure 5.5. The radius-based outlier removal method fails to remove some outlier points that do not belong to the surface.

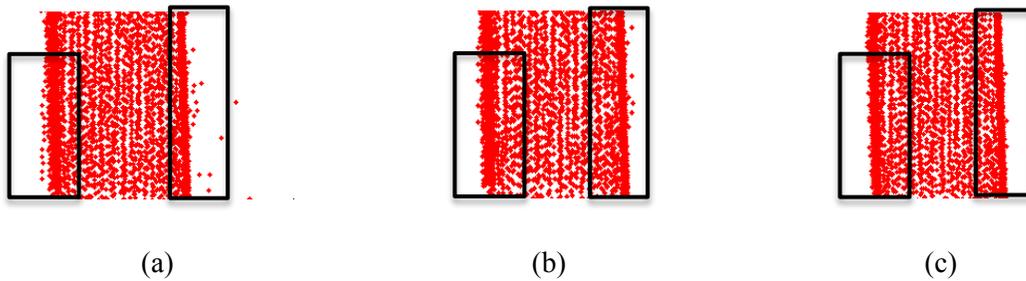


Figure 5.5: Close-up view for (a) raw point cloud; (b) resulting data set after radius-based outlier removal; and (c) resulting data set after statistical outlier removal.

Table 5.4 represents the summary of the parameters and corresponding ranges for both outlier removal methods. The process for determining the outlier removal parameters are discussed in Chapter 7.

Table 5.4: Parameters and corresponding ranges for given outlier removal methods.

<b>Method</b>	<b>Parameters</b>	<b>Range</b>
Radius-based Outlier Removal	Neighborhood size*, $k$	$>3$
	Search radius*, $r$	0.1 in. – 2 in.
Statistical Outlier Removal	Neighborhood size*, $k$	$> 3$
	Standard deviation multiplier, $n$	0.0 – 3.0

\* Either neighborhood size or search radius has to be defined for each run.

The statistical outlier removal method is used for the following steps of this research. This method both yields better results and also reduces the parameter number that has to be specified by the user to only a single value,  $k$ , when the standard deviation multiplier  $n$  is kept at a default value. In contrast, the radius-based outlier method always required two input parameters,  $k$  and  $r$ .

## 5.4 Surface Normal Estimation

The local feature representation of a query point  $p_q$  can be estimated by using the neighborhood points  $P_k$  once the neighborhood size  $k$  is determined. Surface normal estimation is essential because it captures the local properties of every query point  $p_q$  in a given point cloud  $P$ , and the obtained local feature representation can be used to capture the geometry of the underlying sampled surface around investigated  $p_q$ . It is especially important for our applications since the surface normal-based damage detection algorithm discussed in the Section 7.3 is a direct extension of this method. Table 5.5 lists the parameters and the ranges associated with them for surface normal estimation. The selection process for the surface normal estimation parameters for varying defect types are discussed in Chapter 7.

Table 5.5: Parameters and corresponding ranges for given surface normal estimation.

<b>Method</b>	<b>Parameters</b>	<b>Range</b>
Surface Normal Estimation	Neighborhood size, $k$	$>3$

There are several available surface normal estimation methods that exist as discussed in Klasing et al. (2009); for this research, a first order 3D plane fitting based method based on Berkmann and Caelli (1994) is implemented for computing the surface normal associated with each  $p_q$ . A least-squares plane fitting estimation in  $P_k$ , which was described in Shakarji (1998), is used to determine the normal to a point on the surface, by estimating the normal of a plane tangent to the fitted surface. A point  $p_x$  and a normal vector  $\vec{n}$  is used to represent the plane, and the distance from a point  $p_i \in P_k$  to the plane

is defined as  $d_i = (p_i - p_x) \cdot \vec{n}$ . A least-square approach is used to compute the values of  $p_x$  and  $\vec{n}$  such to achieve  $d_i = 0$ , where the centroid of  $P_k$  is computed by Equation 5.4:

$$p_x = \bar{p} = \frac{1}{k} \cdot \sum_{i=1}^k p_i \quad 5.4$$

Thus, the solution for estimating the surface normal  $\vec{n}$  is reduced to an analysis of the eigenvectors and eigenvalues of a covariance matrix created from the nearest neighbors of  $p_q$ . The covariance matrix  $\mathcal{C} \in \mathbb{R}^{3 \times 3}$  of  $P_k$  is expressed as in Equation 5.5 and 5.6:

$$\mathcal{C} = \frac{1}{k} \cdot \sum_{i=1}^k (p_i - \bar{p}) \cdot (p_i - \bar{p})^T \quad 5.5$$

$$\mathcal{C} \cdot \vec{v}_j = \lambda_j \cdot \vec{v}_j, j \in \{0,1,2\} \quad 5.6$$

where  $\lambda_j$  is the  $j$ -th eigenvalue of the covariance matrix, and  $\vec{v}_j$  the  $j$ -th eigenvector.  $\mathcal{C}$  is symmetric and positive semi-definite, and its eigenvalues are real numbers,  $\lambda_j \in \mathbb{R}$ . The eigenvectors  $\vec{v}_j$  form an orthogonal frame that corresponds to the principle components of  $P_k$ . These components are obtained by using Principle Component Analysis (PCA); the MatLab function, *princomp*, is used for this computation. The eigenvector  $\vec{v}_0$ , which corresponds to the smallest eigenvalue  $\lambda_0$ , where  $0 \leq \lambda_0 \leq \lambda_1 \leq \lambda_2$ , is therefore the approximation of  $+\vec{n} = \{n_x, n_y, n_z\}$  or  $-\vec{n}$ .

Figure 5.6(a) represents a portion of the concrete testing frame's point cloud that is filtered with the statistical outlier removal method; the complete point cloud is given in Figure 5.3. Figure 5.6(b) is showing the neighborhood example for a single query point

$p_q$  where  $k = 16$ . Figure 5.7(a) represents the result of the least squares plane fit for the query point  $p_q$  given in Figure 5.6(b), where the light and dark blue lines shows the shortest distance to the fitted plane along y axis from the points that fall on either side of the fitted plane. In Figure 5.7(b), the black arrow shows the result of the estimated normal for the given query point  $p_q$ , where  $p_q$  is the origin of the estimated normal vector.

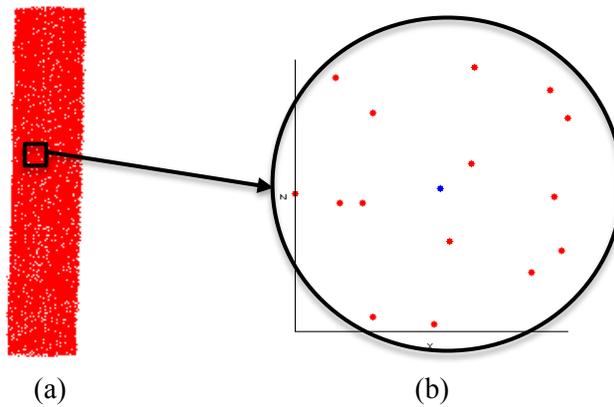


Figure 5.6: (a) Resulting point cloud after statistical outlier removal for raw data extracted from testing frame (Figure 5.3); and (b) the example neighborhood representation for a selected query point  $p_q$ .

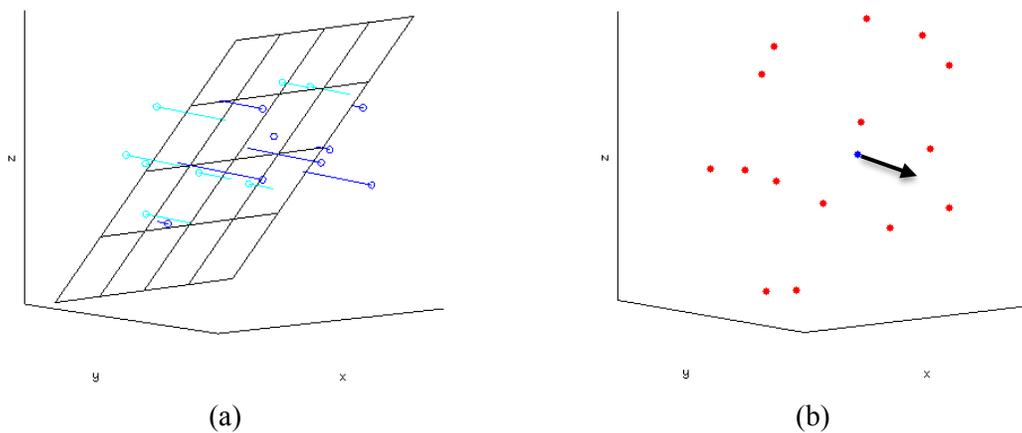


Figure 5.7: (a) Plane fitting result for the selected query point  $p_q$  from the previous figure; and (b) representation of the computed normal vector.

There is not a clear mathematical way to correctly estimate the orientation of the normal computed via Principle Component Analysis (PCA) since the computed normal vectors are inconsistently oriented over a point cloud dataset  $P$ . However, this problem has a trivial solution if the viewpoint  $p_v$  is known. The viewpoint is an additional input parameter that represents the location of the scanner with respect to the investigated surface points. In order to prove that all the computed normals are oriented towards the viewpoint  $p_v$  each computed normal  $\vec{n}_i$  should satisfy the Equation 5.7.

$$\vec{n}_i \cdot (p_v - p_i) > 0 \quad 5.7$$

## 5.5 Curvature Estimates

Another important property of the surface that can be deduced from  $p_q$  is the curvature estimate associated with each point. The curvature estimate both provides information about the properties of the underlying surface and also can be used to remove extraneous points. Similar to surface normal estimation, there are many methods to compute the surface curvature at a specific point. However, most of them require that the surface is already represented with a triangular mesh and are sensitive to noise (Koenderink and van Doorn, 1992; Dyn et al., 2001; Hetzel et al., 2001). Thus, these methods are not suitable for our applications that require curvature estimation directly from the surface points, not from triangular meshes.

The eigenanalysis (PCA) presented earlier can also be used to compute a similarity measure based on the surface curvature around  $p_q$  in addition to the surface normal estimates. The eigenvalues  $\lambda_j$  obtained from the covariance matrix  $\mathcal{C}$  are used to approximate the surface variations (Pauly et al., 2002). From the eigenvalues ( $\lambda_0 \leq \lambda_1 \leq$

$\lambda_2$ ) of the covariance matrix, the variation at  $p_q$  denoted by  $\gamma_p$  along the surface normal  $\vec{n}$  is estimated with Equation 5.8. Equation 5.8 shows that the change of the curvature in a neighborhood  $P_k$  centered around  $p_q$  can be approximated by the ratio between the minimum eigenvalue and the sum of the eigenvalues and the result will be invariant under scaling. As an example, it can be concluded that all points in  $P_k$  are on the plane tangent to the surface if variation  $\gamma_p$  has small values.

$$\gamma_p = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \quad 5.8$$

If all the points lie on a plane then  $\gamma_p = 0$  and the maximum surface variation  $\gamma_p = \frac{1}{3}$  is assumed for completely isotropically distributed points. The surface variation is not an intrinsic property since it is dependent on the neighborhood size  $k$  (Pauly et al., 2002). The parameters and the given range for them are listed in Table 5.6. The selection process for the curvature estimation parameters are discussed in Chapter 7.

Table 5.6: Parameters and corresponding ranges for given curvature estimation.

<b>Method</b>	<b>Parameters</b>	<b>Range</b>
Curvature Estimation	Neighborhood size, $k$	>3

## 5.6 Extraneous Point Detection

Extraneous objects that are not a part of the structure and will cause errors for processing the surrounding surfaces. Points detected for small, complex details exhibit high surface variation. To address this, the mean value of the computed variations  $\mu_\gamma$  and their standard deviation  $\sigma_\gamma$  are computed for the entire point cloud  $P'$  remaining after outlier removal. If  $\gamma_p > \mu_\gamma + n \cdot \sigma_\gamma$  or  $\gamma_p < \mu_\gamma - n \cdot \sigma_\gamma$ , where  $n$  is the standard deviation multiplier that restricts the density for any point, the point is removed from

$P'$ . The remaining point cloud after extraneous point removal  $P''$  is given as Equation 5.9, where  $p_q$  represents the each query point.

$$P'' = \{p_q \in P' | (\mu_\gamma - n \cdot \sigma_\gamma) \leq \gamma_p \leq (\mu_\gamma + n \cdot \sigma_\gamma)\} \quad 5.9$$

Figure 5.8(a) shows the complete point cloud of a laboratory specimen that was tested at Multi-axial Full-scale Substructure Testing and Simulation (MUST-SIM) facility at the University of Illinois at Urbana-Champaign (Eatherton et al., 2014). Figure 5.8(b) represents a part of the point cloud of the laboratory specimen which is occluded with details such as cables and clamps.

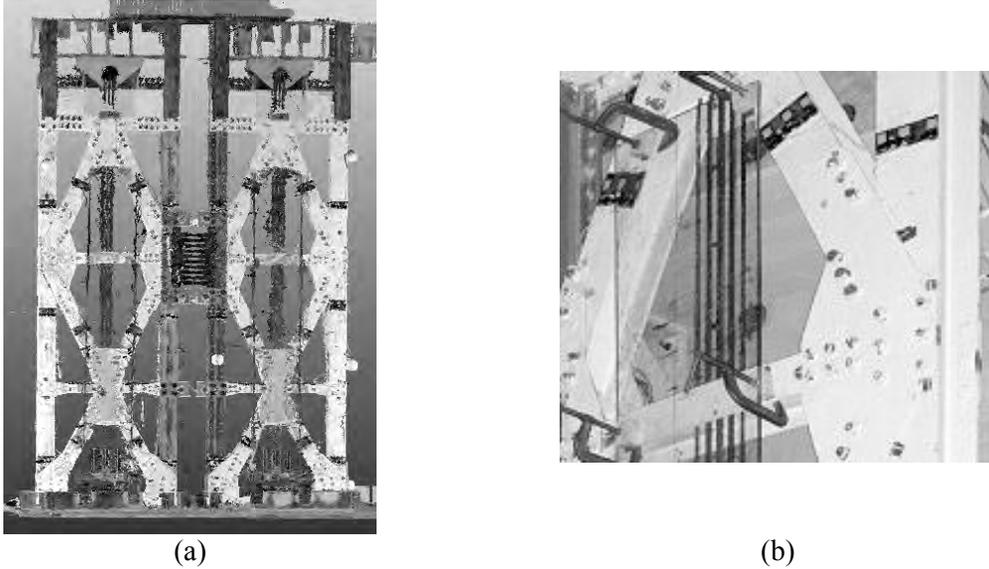


Figure 5.8: (a) Complete point cloud of the test setup at the MUST-SIM facility; and (b) a part of the point cloud of the test setup that is used for outlier and extraneous point detection.

Figure 5.9(a) represents the results of radius-based outlier removal method on the set of points shown in Figure 5.8(b) where the neighborhood size  $k$  is taken as 16 and radius  $r$  is 0.25". The surface points are shown in blue, and the detected outliers are shown in green. Figure 5.9(b) shows the results of the extraneous point removal, which is performed after the radius-based outlier removal. The extraneous points are shown in red.

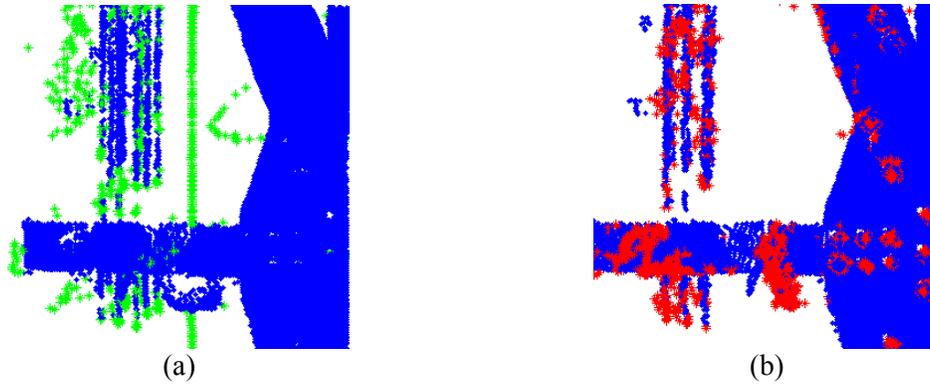


Figure 5.9: (a) Results of the radius-based outlier removal; and (b) results of extraneous point removal applied on a portion of the testing frame given in Figure 5.8.

It is evident from Figure 5.9(a) that the radius-based outlier removal method removes the points that are not meeting a certain radius  $r$  criteria; thus, the results are not dependent on the underlying surface properties. This method is not efficient in detecting the points representing the bolts or the clamps since the points associated with these features satisfy the  $r$  criteria. However, extraneous point removal uses the surface variations for point reduction, so the points associated with the bolts and clamps are detected by capturing the locations of high regional surface variations. The results of the extraneous point removal method for bolts and clamp is shown in Figure 5.10. The parameters and the corresponding ranges are listed in Table 5.7.

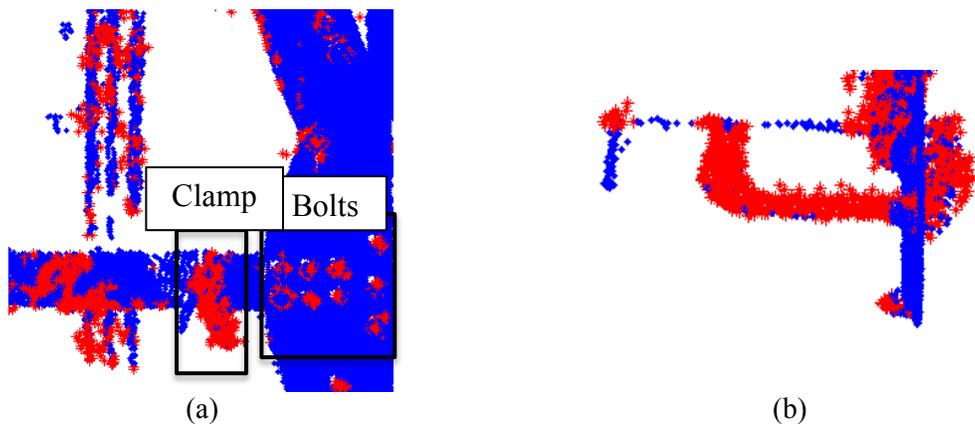


Figure 5.10: Results of the extraneous point removal where the bolts and clamp is shown with rectangles and (b) side view of the clamp points detected by using the changes in surface variation.

Table 5.7: Parameters and corresponding ranges for given extraneous point removal.

<b>Method</b>	<b>Parameters</b>	<b>Range</b>
Curvature Estimation	Neighborhood size, $k$	$>3$
	Standard deviation multiplier, $n$	0-3

## 5.7 Feature Detection

A sharp feature can either be a corner where three or more surfaces connect, or a line or edge that connects two surfaces. These sharp features can be detected in a point cloud by searching through the points looking for those whose normal vectors vary greatly from their neighbors (Rabbani et al., 2006). High normal vector variation between neighboring points represents an abrupt change of surface orientation. Since the point cloud datasets that we acquire represent a set of point samples on the real surface, there are two possibilities for determining a normal: obtaining the underlying surface from the acquired point cloud dataset, using surface meshing techniques, and then computing the surface normals from the mesh; or using approximations to infer the surface normals from the point cloud dataset directly. In our research, we implemented a method that addresses the latter. Once the surface normals are determined, these are used for classifying feature points in an unorganized point cloud.

The method implemented to perform feature point extraction from unorganized point clouds is building off the work of Weber et al. (2010). This method identifies the feature points by using Gauss map clustering on local neighborhoods without user interaction and with no surface reconstruction. Details of the method are discussed below.

### 5.7.1 Gauss Map Clustering

The method implemented for feature point extraction is based on Gauss map clustering. This method is specifically used for determining sharp features. After determining the local neighborhood  $P_k$  for each point, a Gauss map for each point is formed. First, all possible  $k(k - 1)$  triangulations of a query point  $p_q$  with its neighborhood points are produced, where  $k$  is the number of neighborhood points. The normal vector  $\vec{n}$  of each triangle is then calculated and mapped onto a unit sphere centered at  $p_q$ . Figure 5.11 represents an example of Gauss map clustering for a sharp feature point shown as a red dot. In Figure 5.11, the neighborhood points are shown with green dots and the surface points that do not satisfy the neighborhood criteria are shown with black dots. Once the Gauss map is formed, each point undergoes a flatness test to determine whether the investigated point belongs to a surface or to a feature (Weber et al., 2010).

Characteristics of Gauss maps differ depending on the type of surface (such as flat, curved etc.) on which the point is located. This property is used to differentiate a surface point from a feature point. In this work, all possible triangulations associated with one point and projected normal vectors were used in order to overcome normal vector orientations issues. Normal vectors are projected as points on a defined unit sphere centered at  $p_q$  to form clusters, and the half the number of clusters,  $\frac{c_{gm}}{2}$ , on the unit sphere is used to determine whether or not a point belongs to a feature. Half of the total number of the clusters is used since the projected normals double the obtained cluster number.

An angle measure is introduced for completing this clustering step. The angle between any two normal vectors  $\theta_{ij}$  is measured and this angle is compared with a threshold value  $\sigma_{sp}$  (sensitivity parameter). The sensitivity parameter is defined between  $0^\circ$  to  $90^\circ$ . If the calculated angle between two normals is smaller than the threshold value, these vectors are gathered in the same cluster. The clustering criteria is given by the Equation 5.10.

$$\|n_i \cdot n_j\| = \cos \theta_{ij} < \cos \sigma_{sp} \quad 5.10$$

This process is repeated until the minimum number of clusters, where angle measured between each cluster couple is greater than  $\sigma_{sp}$ , is achieved. Figure 5.12 shows the results of the gauss map clustering for two different surface types. In Figure 5.12(a), a flat surface with only one cluster is shown whereas in Figure 5.12(b), a sharp feature point that is located at the edge of two intersecting surfaces.

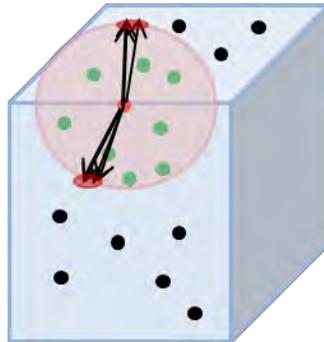


Figure 5.11: Example of gauss map clustering of an edge point represented by red dot where the green points represent the neighborhood points that participated in triangulation.



Figure 5.12: Gauss map clustering results for (a) a point on a flat surface; and (b) an edge point.

Once the clustering is finished, the total number of clusters,  $\frac{c_{gm}}{2}$ , at each point is investigated. If there is only one cluster left, the point is classified as belonging to a flat surface. However, if two, three or four clusters remain, then it is concluded that the point belongs to a sharp feature.

By using a least-squares approach, a curve is then fit to extracted feature points to represent each edge mathematically. Linear segments are used initially in this implementation, as it is a common form for most structural engineering shapes such as columns and beams. Figure 5.13 shows the detected edge points on the synthetic point cloud and Table 5.8 shows the typical parameters and respective ranges used in algorithms for gauss map clustering. In this research,  $k$  (or  $r$ ) is computed automatically for the outlier removal method, since it is observed that these calculated values are adequate for removing the outlier points in the available datasets, listed in Chapter 4, and the  $\sigma_{sp}$  is taken as  $10^9$ .

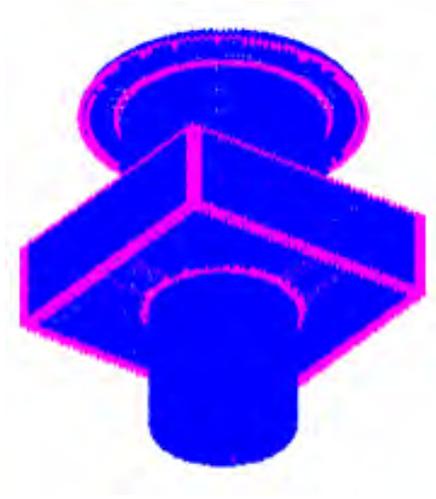


Figure 5.13: Edge points detected on synthetic point cloud.

Table 5.8: Parameters and corresponding ranges for gauss map clustering.

<b>Method</b>	<b>Parameters</b>	<b>Range</b>
Gauss Map Clustering	Neighborhood size*, $k$	>3
	Search radius*, $r$	0.1 in. – 2 in.
	Sensitivity parameter, $\sigma_{sp}$	0° - 10°

\* Either neighborhood size or search radius has to be defined for each run.

## 5.8 Region Growing for Segmentation

Generally, a set of planar surfaces can be used to describe the geometry of man-made structures and these separate segments can be detected after processing point cloud data. Once the feature points are extracted and removed from the point cloud, the remaining points  $P'''$  with similar properties are divided into segments.  $P'''$ , which is also referred as  $P_r$ , represents the remaining points from a point cloud dataset after the statistical outlier removal, extraneous outlier removal and feature point removal are applied respectively.

Region growing is a method of segmentation that begins with individual points and collects neighboring points based on defined criteria to build up segments. In this work, in order to achieve segmentation, a region growing algorithm with a smoothness constraint was implemented since the methods using only the curvature or higher derivatives for segmentation usually result in over-segmentation (Rabbani et al., 2006). The implemented algorithm groups the processed points together to create segments by using the estimates of the surface normals at each query point  $p_q$ . The method for computing the surface normals is discussed in Section 5.4 in detail.

The implemented method uses local surface normals and point connectivity to find smoothly connected areas in a point cloud. It begins by selecting a seed point  $p_s$  from  $P_r$ , which already excludes the sharp features, and then checking surrounding points to determine if they belong in the same segment  $S_j$ . Whether a point  $p_i$  should be added to the segment being grown is determined based upon the angle between its normal  $n_i$  and the normal of the current seed point  $n_s$ . If the angle  $T_\theta$  is less than a threshold value  $\theta_{th}$ , then the current point is added to the current region  $S_j$ . The process of region growing explained in the following steps:

1. Select a seed point from  $P_r$  for the region growing process based on the associated surface variation  $\gamma_p$  computed for each query point  $p_q$ . The process of curvature estimation is explained in Section 5.5. The point with the minimum  $\gamma_p$  is selected as the initial seed point  $p_s$ .
2. Select an specific angle  $T_\theta$  as the smoothness threshold for the angle between the normals of the current seed point  $n_s$  and its neighborhood points  $n_i$ , where  $n_i$  represents the normal estimation for  $p_i$  that is a member of neighborhood

points  $P_k$  of  $p_s$ .  $P_k$  is calculated for each  $p_q$  by using the *KNN* method described in Section 5.2. The region growing constraint can be represented with the Equation 5.11 if the smoothness angle threshold is expressed in radians. The absolute value of the dot product is taken in order to account for the  $180^\circ$  ambiguity. If the Equation 5.11 is satisfied for  $p_i$ , then this point is added to the current segment  $S_j$ .

$$\|n_i \cdot n_s\| > \cos T_\theta \quad 5.11$$

3. If all the points are segmented, then finalize the region growing process. If not, pick new seed point  $p_s$  based on the surface variation of the remaining points.
4. Apply the step 2 and step 3. If there are still remaining points, list them based on their surface variation  $\gamma_p$  and pick the point with the least  $\gamma_p$  as  $p_s$ .
5. Add the current region to the segmentation and repeat step 3.
6. Finalize the segmentation.

The result of region growing algorithm on the synthetic point cloud is shown in Figure 5.14; each color indicates an individual segment recognized by the algorithm. Table 5.9 lists the important parameters and corresponding ranges given for region growing segmentation. In this research, a fixed value for  $T_\theta$  is used,  $T_\theta = 5^\circ$ .

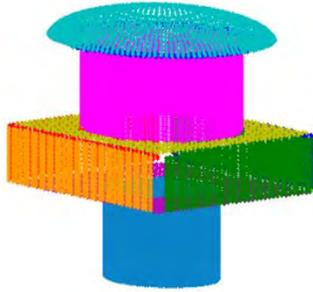


Figure 5.14: Segmentation results for the synthetic point cloud.

Table 5.9: Parameters and corresponding ranges for region growing segmentation.

<b>Method</b>	<b>Parameters</b>	<b>Range</b>
Region Growing Segmentation	Angle threshold, $T_\theta$	0°-10°

## 6. Object Detection for Modeling and Damage Detection

The final step of the point cloud processing prior to damage detection is object detection. This step is required to extract the underlying surface properties from point cloud data. Since the damage detection algorithms operate based on both local and global properties of the underlying objects, for specific damage detection algorithms it is essential to first identify the underlying object/surface properties from the segmented regions in order to perform the necessary post-processing steps for damage localization and quantification.

Generally, segmented regions do not represent an entire object but portions of it. This may be due to occlusion or just because how the objects are connected with each other in a scene. Thus, the segmented regions need to be classified geometrically based on their surface properties such as plane, cylinder, sphere, and etc. Once the surfaces are characterized, the appropriate quadratic surface equations are used for fitting. The coefficients of the quadratic equation computed via regression are used to store the geometric properties of the investigated segmented region. Then, the segmented regions are either left as-is or combined with the adjacent regions to create point clusters depending on their local properties. These point clusters are used to individually identify the objects in a scene. Later, the detected objects are used for damage localization and quantification. Either the entire 3D representation of an object is used to compare the original state of an item with its current conditions, or the surface properties of each region are used for the surface-based damage detection as described in Chapter 7.

First, a point feature histogram (*PFH*) based method is used for extracting the primitive geometric representations of the segmented regions. The details of this method

are described in Section 6.1. In Section 6.2, the general quadratic equation is given, and its application for surface property estimation is described in detail. Then, the related regions are combined to create point clusters. The objects in the scene are detected from point clusters as described in Sections 6.3 and 6.4.

Sections 6.3 and 6.4 summarize two different object detection methods that are implemented for this research: a global feature-based object detection method and a graph-based object detection method. The global feature-based method uses the 3D representation of an object for detection, and the graph-based method allows object detection at both the local and global scales. Both of these methods have advantages and disadvantages.

First, the global feature-based object detection is beneficial if the global properties of the investigated object are intact, in our case not damaged. Thus, it is commonly used for classifying objects in an environment for creating information models, e.g. BIM. However, local defects on a scanned object might result in distinctive alterations in the calculated global properties, and this will affect the accuracy of the object detection.

Second, the graph-based object detection method is more flexible since the object recognition is based on the local properties instead of the global ones. Therefore, even if a portion of the investigated object is damaged, it is still possible to use this method to determine the object type from the undamaged portion. On the other hand, since the objects are generally defined in 2-D, it is harder to extract point clusters for separate objects even if the object type is known.

In our research, our main focus is on damage detection. Thus, most of the damage detection methods, which are developed and explained in Chapter 7, are based on graph-

based object detection; however, both approaches are used depending on the specific type of damage.

## 6.1 Point Feature Histograms

In the introduction of this chapter, the necessity of identifying the underlying surface type before performing a surface fitting is mentioned. In order to perform the surface characterization, a point feature histogram based method is implemented. This method is used for both classification and for feature extraction of point clouds through a covariance-based procedure (Belton and Lichten, 2006; Rusu, 2009). The computation steps for both the covariance matrix and the surface normals are discussed in related sections of Chapter 5.

Point Feature Histograms (PFH), which are calculated based on surface normals, are informative pose-invariant local features that represent the underlying surface model properties at a point  $p_i$ . Their computation is based on the combination of certain geometrical relations between  $p_i$ 's nearest  $k$  neighbors. This method uses the interactions between the directions of the estimated normals to compute the surface variations. PFH's mainly incorporate 3D point coordinates and estimated surface normals, but at the same time they combine other properties such as curvature,  $2^{nd}$  order moment invariants, etc.

A PFH determined at a point  $p_i$  relies on the presence of 3D coordinates and estimated surface normals, and it is computed as follows: first, for each point  $p_i$ , all of  $p_i$ 's  $k$  nearest neighbors ( $KNN$ ) enclosed in the sphere with a given radius  $r$  are found; second, a natural moving Darboux  $uvn$  frame is constructed on the surface for every pair of points  $p_i$  and  $p_j$  ( $i \neq j$ ) in the  $k$ -neighborhood of  $p_i$  and their estimated normals  $\vec{n}_i$

and  $\vec{n}_j$  where  $u = \vec{n}_i$ ,  $\vec{v} = (p_j - p_i) \times \vec{u}$ , and  $\vec{w} = \vec{u} \times \vec{v}$ . In this context,  $p_i$  is the point with a smaller angle between its associated normal and the line connecting the points  $p_i$  and  $p_j$ . The angular variations of  $\vec{n}_i$  and  $\vec{n}_j$  are computed by using the geometric relations between the Darboux frame of both  $p_i$  and  $p_j$ .

The features can be separately computed and grouped for both convex and concave shapes since PFH computation is based on normal information. These features are informative enough to differentiate between points lying on different surfaces when the computation parameters are chosen carefully. Figure 6.1 presents *PFH* signatures for points lying on three different surfaces types, namely a plane, sphere with a radius of 5 in, and a cylinder with a radius of 5 in. For this application  $k$  is taken as 3 since a synthetic point cloud dataset with a low point density is used. In order to illustrate that the features are discriminative, a confusion matrix with gray values representing the distances between the mean histograms of different shapes is assembled. This confusion matrix is obtained by using the Histogram Intersection Kernel (Barla et al., 2003), and it is given at the top left part of the Figure 6.1. Table 6.1 represents the parameters and the typical corresponding ranges used for computing point feature histograms. The values for these parameters are kept the same with the values used in the statistical outlier removal method described in Section 5.3.2. *PFH* computations are done by using the implemented Point Cloud Libray (PCL) algoritihms (Rusu and Cousins, 2011).

Table 6.1: Parameters and corresponding ranges for point feature histogram.

<b>Method</b>	<b>Parameters</b>	<b>Range (or default value)</b>
FPFH based classification	Neighborhood size*, $k$	>3
	Search radius*, $r$	0.1 in.– 2 in.

\* Either neighborhood size or search radius has to be defined for each run.

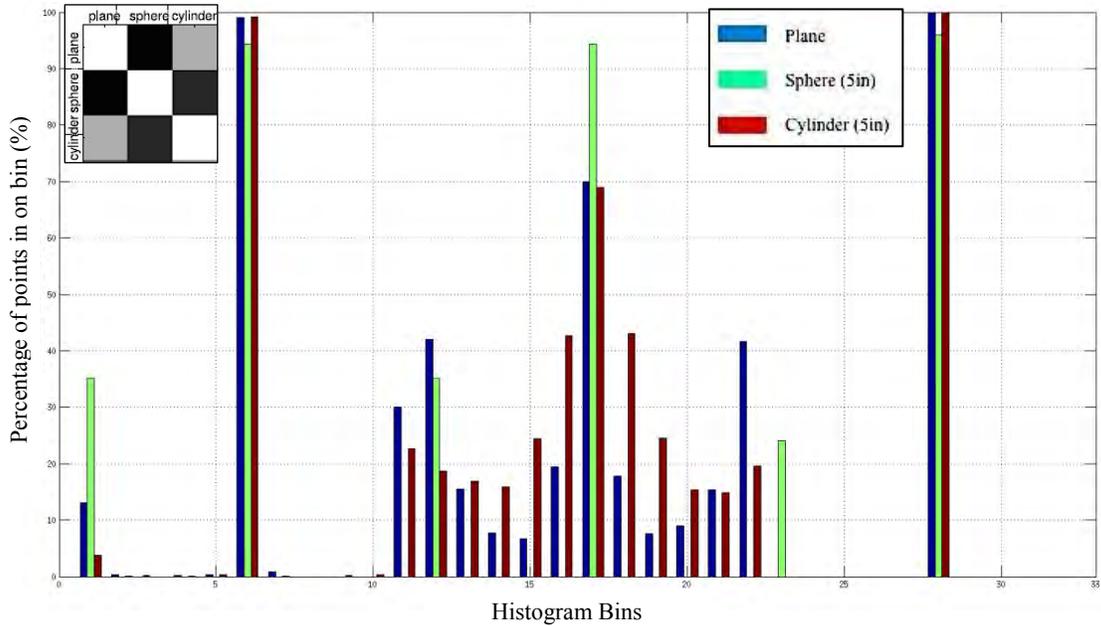


Figure 6.1: Example of point feature histograms for points lying on simple 3D geometric surfaces after Rusu (2009).

## 6.2 Surface Fitting

Once the surfaces that define the segmented regions are categorized, the corresponding mathematical representation of each surface is used for fitting that is used to compute the surface parameters. For each segment, surface fitting is performed by using the appropriate regression model (e.g., ordinary least squares) depending on the underlying surface type (Dai and Newman, 1998).

Simple surfaces (planes, cylinders, spheres, cones, etc.) can be represented by quadratic equations. The general equation for an arbitrary quadric surface in Cartesian space is given by Equation 6.1 (Breyer, 1987). Depending on the categorized surface type, the equation of the quadratic surface varies (Adams, 1999). The appropriate version of the general quadratic equation is then used to estimate the surface parameters of each region. It should be noted that, in this research, surface fitting is only performed on

simple surfaces, including (as listed earlier) planes, cylinders, spheres, cones, etc. since the investigated objects are composed of either only one of the listed surface types or a combination of several surface types.

$$Ax^2 + By^2 + Cz^2 + 2Fyz + 2Gxz + 2Hxy + 2Px + 2Qy + 2Rz - D = 0 \quad 6.1$$

where

$$e = \begin{bmatrix} A & H & G \\ H & B & F \\ G & F & C \end{bmatrix}, E = \begin{bmatrix} A & H & G & P \\ H & B & F & Q \\ G & F & C & R \\ P & Q & R & D \end{bmatrix}, \rho_3 = \text{rank}(e), \rho_4 = \text{rank}(E), \Delta = \det(E).$$

For a set of segmented points representing a surface, the coefficients are estimated by solving a group of  $m$  equations (Equations 6.2 - 6.4). This approach is implemented for each segment, which is identified during region growing described in Section 5.8. The detected surfaces and the computed parameters are later used for object detection.

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{b} \quad 6.2$$

where

$$\mathbf{A} = \begin{bmatrix} x_0^2 & y_0^2 & z_0^2 & x_0y_0 & y_0z_0 & z_0x_0 & x_0 & y_0 & z_0 \\ x_1^2 & y_1^2 & z_1^2 & x_1y_1 & y_1z_1 & z_1x_1 & x_1 & y_1 & z_1 \\ \vdots & \vdots \\ x_m^2 & y_m^2 & z_m^2 & x_my_m & y_mz_m & z_mx_m & x_m & y_m & z_m \end{bmatrix} \quad 6.3$$

and

$$\mathbf{X} = [A \ B \ C \ 2F \ 2G \ 2H \ 2P \ 2Q \ 2R]' \quad 6.4$$

### **6.3 Global Feature-based Object Detection**

The first object detection method builds on the surface fitting described in the previous section. Once the point cloud has been transformed into a collection of surfaces bounded by edges, and the surface properties are extracted, the next step is to determine how those surfaces combine to form objects. Identifying objects in a scene is an important part of point cloud processing leading towards structural engineering applications.

One way of defining an object is as a set of connected surfaces, where generally an input range data is used to create a symbolic description of the object in terms of the visible surface patches. An object definition using this approach is organized like a graph with the vertices representing surfaces, and the edges representing proximity and relative orientation. The object graph is matched with the model graph from a pre-defined model base for recognizing the objects in a scene. Various features of each surface of object models are stored for construction of a model base; these features are compared with the computed surface parameters in order to enhance the matching results. This step is performed to prevent false detections resulted due to the objects that have similar surface connectivity information but have varying surface properties.

When two or more surfaces are identified as neighbors, the angle between their average normals is computed and stored for checking potential model matches. Once the surfaces with common feature points are linked together, they are checked for compactness. This step is performed to avoid grouping of surfaces, which are parts of adjacent objects but sharing common feature points, within the same point cluster. The collection of surfaces and inter-surface relationships are matched against a predefined

object library. The model matching results are checked against a dissimilarity measure. Every feature stored in the object library is checked, and the best match is extracted for each detected object in a scene. The object detection algorithm in this work is looking for an overall similarity of minimum 85% for both global and local features in order to match the detected object with the corresponding model representation.

The objects in the library are generically defined based on geometric primitives and their relative relationship, allowing expansion for new objects. Each object definition contains information to identify the object (surface types and relative orientations) and describe the object for storage after detection. The description of the object contains methods to describe the location and orientation of the object in space and methods to describe its geometric properties, for example, length, width and height for a cuboid. The result is a description of each object in a scene in simple form that can then be used to track location, orientation, and size. These recorded properties could be scaled and used for detecting similar objects with different geometric properties. In this way, the developed object library can be used for different point clouds that contain similar objects with varying geometric properties. Table 6.2 shows the common parameters and defined ranges for these parameters for global feature based object detection. In this research, the maximum dissimilarity measure is defined as 15% for both local and global object matching.

To determine the effectiveness of the algorithms, examples from the point cloud of DeKalb County Bridge is used. Figure 6.2 is showing a portion of the test-bed bridge point cloud (a) and it is also representing sharp features and detected piers along with the

detected pier cap surfaces (b). A cylinder, which has varying length, was defined in the object library and matched with the pier surface (Walsh et al., 2013).

Table 6.2: Parameters and corresponding ranges for global feature-based object detection.

<b>Method</b>	<b>Parameters</b>	<b>Range</b>
Global Feature-based Object Matching	Local dissimilarity measure	0% - 15 %
	Global dissimilarity measure	0% - 15 %

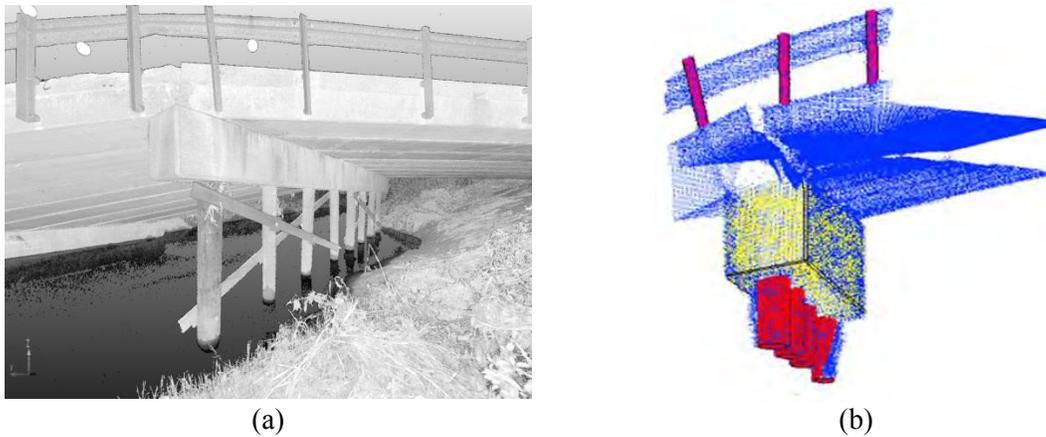


Figure 6.2: (a) Scan data of DeKalb County Bridge with section utilized for data processing; and (b) DeKalb County Bridge point cloud with four piers detected as cylinders and pier cap surfaces detected as planes.

This method is effective for modeling an undamaged structure by differentiating objects in a scene. However, since the object representations in the library consist of global properties for each object, variances in local properties reduce the effectiveness of the method. Thus, the method is suitable for detecting objects in a scene for which the geometries of the objects are either intact or have minor defects, but it is not successful for object detection where significant changes in the global properties of mentioned objects occurs.

In order to detect objects from geometrically altered situations, another object detection method, which is called graph-based object detection, is implemented. Graph-based object detection method is described in the following section.

#### **6.4 Graph-based Object Detection**

This section presents a skeleton-based approach for object detection. This method involves performing two main feature extraction steps on deduced point clusters obtained through grouping segmented regions. The connectivity information described in the previous section is used to find the surfaces that are part of the same object, and the points associated with each surface (segmented regions) are grouped together to form point clusters. First, the voxel model of the unorganized point cluster is extracted, and this model is used to create a 3D binary image. Second, this 3D binary image is divided into 2D binary images along the length of the cluster (cross-section cuts). Subsequently, these images are processed by thinning and/or distance transform to produce skeleton graphs, and they are also used to calculate 2D shape descriptors. 2D shape descriptors and skeleton graphs are matched with the objects from a pre-built object library in which the object information is stored in terms of shape descriptors and skeleton graphs.

The main advantage of this method over the global feature-based object detection method, is that it can be used on incomplete point clouds as well. The cross-sections, which store 2D shape information, are used for object detection. This qualifies the graph-based object detection as a suitable method to create skeletons for searching objects within point clouds composed of defected members.

### 6.4.1 Voxelization

The initial requirement of the graph-based object detection method is an effective skeletonization that will later be used for object recognition. This object recognition method is a more comprehensive version of the one described in the previous section: the global feature-based object detection.

Voxelization is the first step for creating a skeleton from a point cluster. It is the process of converting the geometric representation of a point cloud into a set of voxels that accurately represents the investigated point cloud within a discrete voxel space. It is an important tool to extract both 2D and 3D information from unorganized point clouds. Most of the existing voxelization methods that are used for operating on surface representations of objects (such as aerial laser scanner or terrestrial laser scanner data) could be divided into two sub-categories: surface-based voxelization methods (Gibson, 1995) and point-based voxelization methods (Nehab and Shilane, 2004). The representations of both surface-based and point-based methods are shown in Figure 6.3. The voxelization method used for this research is a point-based voxelization technique, which does not require an initial surface model of the object, and hence allows direct point cloud processing.

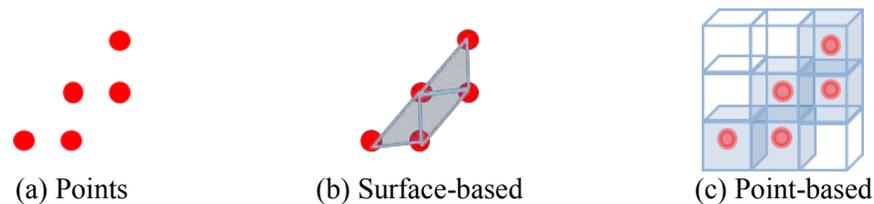


Figure 6.3: (a) A sample point set, (b) surface-based voxelization (also called surface meshing) result, and (c) point-based voxelization result.

The set of input points from each point cluster, which are rotated to fit a given axis by using the results of orthogonal linear regression (Jolliffe, 2002), are given as  $P = \{p_0, \dots, p_{n-1}\}$ , where  $n_p$  is the number of points, and each point  $p_i$  is in the form of  $p_i = (x_i, y_i, z_i)$ . Principle component analysis, which is used to compute the eigenvalues  $\lambda$  and the eigenvectors  $\vec{v}$  in Section 5.4, is used to determine the direction of the fitted line. The direction vector that defines the fitted line is represented by the coefficients of the first principle component. This direction vector  $\vec{v}_1$  and a reference rotation vector  $\vec{r}$ , which is taken as  $[0 \ 0 \ 1]$  ( $z$ -direction) of a reference coordinate system, are used for rotation. All the points in  $P$  are rotated to align with  $\vec{r}$  by using the Equation 6.5 and Equation 6.6. First, the rotation matrix  $R$  is computed by using the Rodrigues' rotation formula. The original  $P$  is then multiplied with  $R$  to get the rotated point cloud  $P_{rot}$ . This step is performed to enable the selection of a height function along the  $z$ -axis. The height function increment  $h_p$ , which also refers to the resolution in  $z$ -direction, is an input parameter defined by the user. Its value is determined based on the requirements of the investigated defect type.

$$[\vec{v}_1]_x \vec{r} = \vec{v}_1 \times \vec{r} = \begin{bmatrix} 0 & -\vec{v}_{13} & \vec{v}_{12} \\ \vec{v}_{13} & 0 & -\vec{v}_{11} \\ -\vec{v}_{12} & \vec{v}_{11} & 0 \end{bmatrix} \vec{r} \quad 6.5$$

$$R = I + \sin \beta [\vec{v}_1]_x + (1 - \cos \beta)(\vec{v}_1 \vec{v}_1' - I) \quad 6.6$$

where  $\beta$  is the angle between  $\vec{v}_1$  and  $\vec{r}$ .

The voxelization method implemented here is explained in (Hinks, 2011), and it consists of three main steps. These steps can be listed as construction of a voxel grid based on the laser point clouds, mapping of the given laser point cloud to the generated

voxel grid, and determination of active voxels based on the performed point mapping. The bounds of the voxel grid assure that every point  $p_i \in P$  has an accurate voxel mapping. These bounds are defined by the minimum and maximum values of all points in  $P$  along  $x$ ,  $y$ , and  $z$ -directions:  $(x_{min}, x_{max})$ ,  $(y_{min}, y_{max})$ , and  $(z_{min}, z_{max})$ . The voxel grid dimensions in  $x$  and  $y$ - directions are determined by using a targeted resolution value  $r_p$  ( $\Delta_x = \Delta_y$ ) provided by the user or computed based on the point cloud resolution. However, the voxel grid dimension in  $z$ -direction is defined as  $h_p = \Delta_z$ . The number of voxels along each axis  $X_n$ ,  $Y_n$  and  $Z_n$  is determined by using Equations 6.7 - 6.9. Figure 6.4 represents samples of a voxel grid and a single voxel.

$$X_n = \frac{|x_{max} - x_{min}|}{\Delta x} \quad 6.7$$

$$Y_n = \frac{|y_{max} - y_{min}|}{\Delta y} \quad 6.8$$

$$Z_n = \frac{|z_{max} - z_{min}|}{\Delta z} \quad 6.9$$

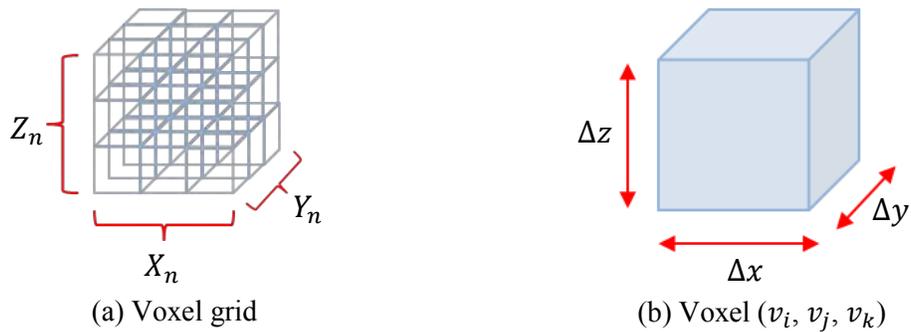


Figure 6.4: Representation of (a) a sample voxel grid and (b) a single voxel.

Once the suitable voxel grid is created, active voxels are determined based on the point mapping results. Mapping is performed by using Equations 6.10 - 6.12. Voxels that contain more points than a given threshold value  $T_n$  are marked as an active voxel. In this research the threshold value  $T_n$  is taken as 1, thus each voxel that has at least one point is marked as active. This process converts the 3D dataset into a 3D binary image. The parameters and the corresponding ranges for the described voxelization method are given in Table 6.3. The selection process for the given parameters is discussed in Chapter 7.

$$v_i = \left\lfloor (X_n - 1) \frac{(x_i - x_{min})}{(x_{max} - x_{min})} + 0.5 \right\rfloor \quad 6.10$$

$$v_j = \left\lfloor (Y_n - 1) \frac{(y_i - y_{min})}{(y_{max} - y_{min})} + 0.5 \right\rfloor \quad 6.11$$

$$v_k = \left\lfloor (Z_n - 1) \frac{(z_i - z_{min})}{(z_{max} - z_{min})} + 0.5 \right\rfloor \quad 6.12$$

where  $\lfloor x \rfloor$  operator rounds down the computed voxel coordinates to the closest integer value.

As explained above, the voxel grid is a volumetric representation, and for  $T_n = 1$  it can be treated as a 3D binary image that builds up from 2D binary images. 3D binary images can be separated into their building blocks by using a height function. Each z-layer of the 3D binary image represents a 2D binary image. Figure 6.5(a) represents the voxel representation of an example structural steel C-section, and Figure 6.5(b) shows one of the binary images that forms the voxel grid. The point cloud cluster used for Figure 6.5 is extracted from the steel lab frame dataset.

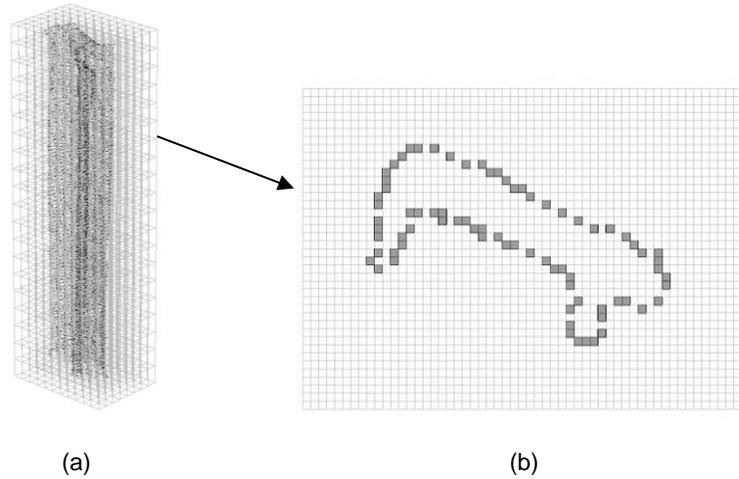


Figure 6.5: (a) 3D voxel grid of a C-section; and (b) 2D binary image at  $z=4$ ".

Table 6.3: Parameters and corresponding ranges for voxelization.

<b>Method</b>	<b>Parameters</b>	<b>Range</b>
Voxelization	Voxel grid dimension, $\Delta_x, \Delta_y$	0.03 in. – 0.5 in.
	Height function increment, $h_p$	0.1 in. – 2 in.

#### 6.4.2 Skeletonization

Skeletonization, which follows voxelization, is one of the most important steps in graph-based object detection since the topology is an essential shape characteristic for both 2D and 3D models. Graphs and trees are the two major relational data structures that are used for topology representations.

The skeleton of a model is computed by using skeletal graph-based techniques, and the shape descriptors of the model are obtained by converting this skeleton into a skeletal graph (Iyer et al., 2005). Some of the common skeletonization methods include distance transform (Niblack et al., 1992), thinning (Lee et al., 1994) and Voronoi-based methods (Hisada et al., 2001). The skeletonization methods used for this research include thinning and distance transform.

Both thinning and distance transform are used to create skeletons from 2D binary images. Thinning is used for creating skeletons of the objects in the object library where the objects are represented with filled cross-sections, whereas distance transform, is used for the 2D binary images that are extracted from voxel grids generated from the point cloud. In latter case, distance transform is chosen because the binary images deducted from the point clouds form only the perimeter of the cross-sections, and in some cases these are incomplete. Thinning cannot be applied on a binary image, which consists of an incomplete perimeter of a cross-section, since it is only applicable on closed-loops.

Thinning (or morphology) is used to transform a digital image into a simplified and topologically equivalent image. Thinning is an iterative pixel-deletion process. The criteria for the pixel-deletion are defined such that the connectivity of the image pattern is preserved. In this research, thinning is achieved by a MatLab function called *bwmorph* with the option of '*thin*', which is based on Lam et al. (1992).

The distance transform method computes the Euclidian distance transform of a binary image; this is achieved by assigning a distance value to each pixel, which is the distance between that pixel and the nearest non-zero pixel. Pixels with the highest Euclidian distance values are then connected together to create a skeleton of a given cross-section. The MatLab function called *bwdist* is used for this operation. The default distance measure used for the distance transform is the Euclidian distance (Maurer Jr et al., 2003). However, other options such as chessboard, cityblock or quasi-euclidian can also be used as distance measures (Rosenfeld and Pfaltz, 1966). Examples of 2D and 3D skeletons are shown for a C-section in Figure 6.6.

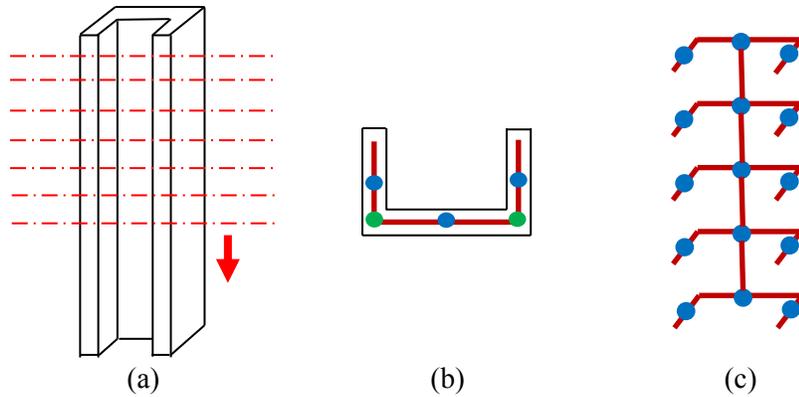


Figure 6.6: (a) Example C-section with the direction of height function, (b) 2D skeleton representation, (c) 3D skeleton representation.

### 6.4.3 Expanded Object Library

The object library is an essential part of object recognition process. In order to recognize objects clustered in the field, isolated objects are compared to objects in a object library. As mentioned in Section 6.3, the object library is composed of a variety of object descriptions starting with simple prismatic shapes such as rectangular prisms, cylinders, etc. More sophisticated objects are also included, which are created based on section properties taken from the AISC Steel Construction Manual (AISC., 2011) for W, M, S, HP, C, MC, WT, MT, ST shapes and angles. In order to detect objects by using skeletons as well, some additional representative information on the objects are stored in the object library. For each object, a list of entities that are representing the object's cross-section are added to the object library; this additional information includes a cross-section's filled binary image, perimeter binary image, 3D point cloud at  $z=0$ , 2D image descriptors associated with both binary images, and skeleton properties.

Two-dimensional shape descriptors, which are also called image moments, are particular weighted average (moment) of the image pixels' intensities, or a function of such moments. Once the cross-section of a cluster is extracted as a binary image,

calculated image moments are useful to describe and detect objects. In this research, the Hu set of invariant moments was used (Hu, 1962). This set consists of seven moment invariants, which are invariant under translation, changes in scale and rotation. Moment invariants were calculated and stored for each model section in the library. Hu's seven moment invariants are given by Equations 6.13 - 6.19.

$$I_1 = \eta_{20} + \eta_{02} \quad 6.13$$

$$I_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad 6.14$$

$$I_3 = (\eta_{30} - \eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad 6.15$$

$$I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad 6.16$$

$$I_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 \\ - (\eta_{21} + \eta_{03})^2] \quad 6.17$$

$$I_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ + 4\eta_{11}(\eta_{12} + \eta_{30})(\eta_{21} + \eta_{03}) \quad 6.18$$

$$I_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 \\ - (\eta_{21} + \eta_{03})^2] \quad 6.19$$

where

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{(1+\frac{i+j}{2})}} \quad 6.20$$

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad 6.21$$

$$\bar{x} = \frac{M_{10}}{M_{00}} \quad 6.22$$

$$\bar{y} = \frac{M_{01}}{M_{00}} \quad 6.23$$

where  $x$  and  $y$  represent the pixel coordinates.

An example of how a typical C-section was characterized in the library is shown in Figure 6.7. Figure 6.7 consists of the hollow and filled representations of the C-section, its skeleton obtained through skeletonization methods, key-points detected from the skeleton and image invariants. These properties are stored for every single section and later used for object detection and model fitting.

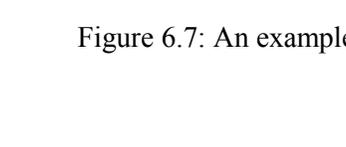
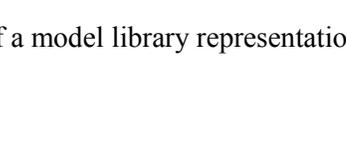
		$I_1$	9.1792
		$I_2$	55.8758
		$I_3$	66.2719
		$I_4$	13.5370
		$I_5$	38.8079
		$I_6$	-101.1878
		$I_7$	-9.6898
		Corners	2
		Key-points	6
		Parts	3

Figure 6.7: An example of a model library representation of C-sections.

#### 6.4.4 Object Detection and Model Fitting

Object detection was completed by using already extracted properties of the examined point cloud sections at varying elevations along z-axis  $CS_i$  where  $i$  represent the section number, and performing a search in the object library in order to find the best match. All available information, which includes moment invariants and skeleton properties, is compared with the entries in the object library, and the closest match  $M_0$  is found. Similar to global feature-based object detection, an overall similarity of minimum 85% for local features is enforced in order to match the detected object with the corresponding object representation.

Model fitting is performed by using the Iterative Closest Point (*ICP*) approach (Besl and McKay, 1992), which is a common method for point cloud registration. There are four common steps of *ICP* algorithms. First, the points in both  $CS_i$  and  $M_i$  are associated by the neighbor criteria where for each point in  $CS_i$  the closest point in  $M_i$  is found. These correspondences are weighed, and the outliers are removed from the  $CS_i$  dataset. Second, the transformation parameters rotation and translation are estimated by using a mean square cost function. Third, the points are transformed by using the estimated parameters. This transform aligns each point to its match found in the first step. Finally, the iterations are continued until a defined threshold criterion is met. Table 6.4 represents the parameter and their corresponding ranges used in algorithms during graph-based object detection. In this research, the maximum dissimilarity measure is defined as 15% for both local and global object matching. The maximum correspondence distance is kept constant at  $0.5in$ ; and the maximum number of iterations is defined as 100.

3D model point cloud at  $z = 0$  associated with  $M_0$ , which is a component already stored in the model library, was used to register each  $CS_i$  in order to find the section that minimizes the error function and the best match was labeled as  $B_0$ . The error is defined by the the sum of the squares of all errors between  $M_0$  and each  $CS_i$ ; the error value for each section in the point cloud data at different elevations was recorded as  $e_{0i}$ .  $B_0$  is assumed to represent the as-built condition of the section. Once the  $B_0$  section is chosen, ICP algorithm is repeated for each section extracted from point cloud section and new error values were recorded as  $e_{1i}$ . If  $\min e_{0i} < \min e_{1i}$ , then  $M_0$  is defined as the new  $S_0$  Section. The  $B_0$  section was then used for cross-sectional change detection. Figure 6.8 shows examples of detected  $M_0$  and  $S_0$  and  $CS_i$ .

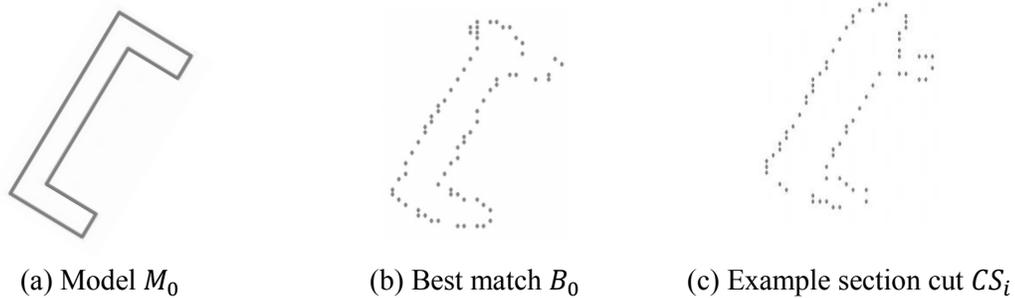


Figure 6.8: (a) Detected C-section  $M_0$ ; (b) best match  $B_0$ ; and (c) example section  $CS_i$ .

The graph-based object detection algorithm is also looking for an overall similarity of minimum 85% for both global and local features in order to match the detected object with the corresponding model representation.

Table 6.4: Parameters and corresponding ranges for the graph-based object detection and model fitting.

<b>Method</b>	<b>Parameters</b>	<b>Range (or default value)</b>
Iterative Closest Point	Maximum correspondence distance	0.4 in. – 2 in.
	Number of iterations	50 - 100
	Transformation Epsilon	1.00E-08
	Euclidian Fitness Epsilon	1.0
Graph-based Object Matching	Local dissimilarity measure	0 - 15 %
	Global dissimilarity measure	0 - 15 %

## **7. Camera Integrated Laser-based Surface Damage Detection**

The object representations can be utilized for obtaining either the geometric or finite element model of structures once the objects are detected accurately by using the strategies described in Chapter 6. The defect localization and quantification can be achieved by developing new strategies for comparing the captured data, which represents the as-is condition of the structure, with the correctly detected and fitted objects from model library. This general methodology allows to categorize and develop damage detection strategies for a wide range of damage types that fall under three main categories described as: small deformations already existing on the structure; damage types that induce larger deformations, where the initial topology of the structure has not changed appreciably; and large deformations, where localized change in the topology of the structure has occurred. A detailed list of investigated damage types includes bent members, misalignments, spalling, section loss, rupture, points of discontinuity, and cracks.

There are two major damage detection methodologies based on modal comparison implemented for this research: graph-based damage detection and surface normal-based damage detection. Both of these methods build on the methods described in Chapter 5, where laser data processing towards object detection is discussed. Chapter 9 describes how to use current visual inspection methods to interpret the obtained results and report them efficiently through combining current visual inspection methods with camera integrated laser-based damage detection.

## 7.1 Noise Removal

### 7.1.1 Outlier Removal

Similar to point cloud processing, the first step for both damage detection methods is noise removal, since the outlier points affect the obtained results. As was mentioned before, statistical outlier removal that is explained in Section 5.3.2 is used as the main method for noise removal from the investigated point cloud. Statistical outlier removal is dependent on two parameters. These parameters are the neighborhood size  $k$  and the standard deviation multiplier  $n$ . The parameters and the corresponding ranges for statistical outlier removal are given in Table 5.3.

A portion of the top deck data of the DeKalb County collapsed bridge is used to show the sensitivity of the statistical outlier removal algorithm to both  $k$  and  $n$ .

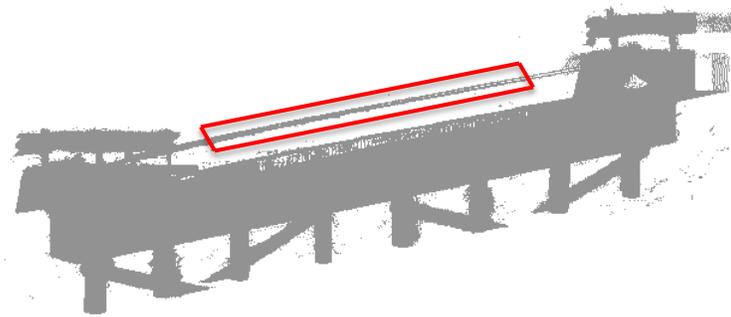


Figure 7.1: Processed portion of DeKalb County Bridge, where red rectangle shows the part of the deck used for sensitivity analysis.

Figure 7.2 shows the sensitivity of statistical outlier removal to the neighborhood size  $k$ . Figure 7.2(a) represents the results of  $k$  versus the number of detected outliers, whereas Figure 7.2(b) shows the results of  $k$  versus the remaining number of points, once the outliers are removed from the point cloud. During this process the standard deviation multiplier  $n$  was kept constant,  $n = 1$ . Several common neighborhood sizes are included

in the analysis. The results showed that at  $k = 16$  and  $n = 1$  removed the largest number of points from the point cloud.

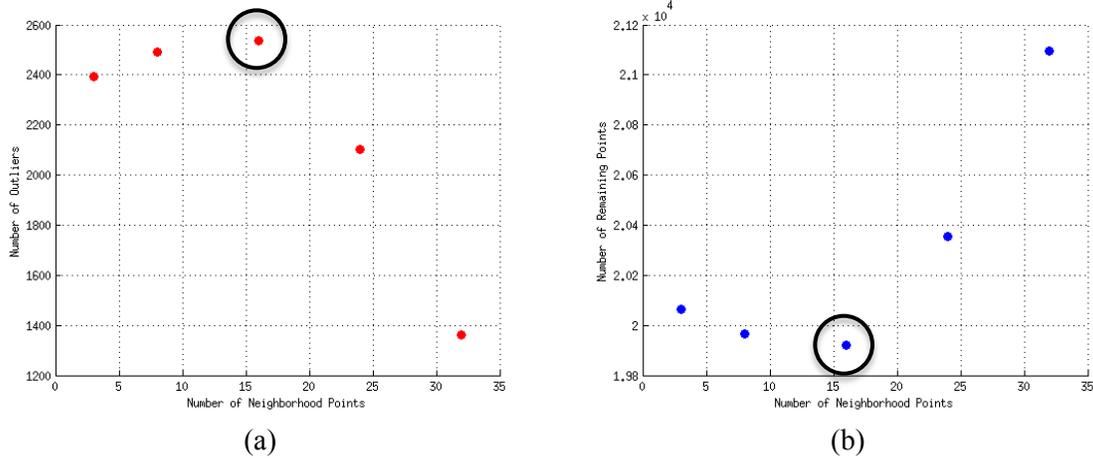


Figure 7.2: (a) Graph of  $k$  versus number of outliers; and (b) graph of  $k$  versus number of remaining points.

Figure 7.3 shows the sensitivity of the statistical outlier to the standard deviation multiplier  $n$ . Figure 7.3(a) represents the results of  $n$  versus the number of detected outliers. On the other hand, Figure 7.3(b) shows the results of  $n$  versus the number of points left in the statistical outlier removal method applied point cloud. During this process a constant value of neighborhood size  $k = 16$  is used. A range of  $n$  that varies between 0 and 3 are used for the analysis. The results for  $k = 16$  and  $n = 1$  are marked with black circles. It can be observed that in this case  $n = 1$  does not result in the largest number of outliers. However, this does not indicate that  $n = 0$ , which removed the most number of points from the dataset, gives better outlier removal results. However, since the criteria band is too short,  $n = 0$  causes misdetection and removes some of the surface points along with noise. The opposite occurs when  $n = 3$  is used; some of the outlier points are not detected since the criteria band is too wide.

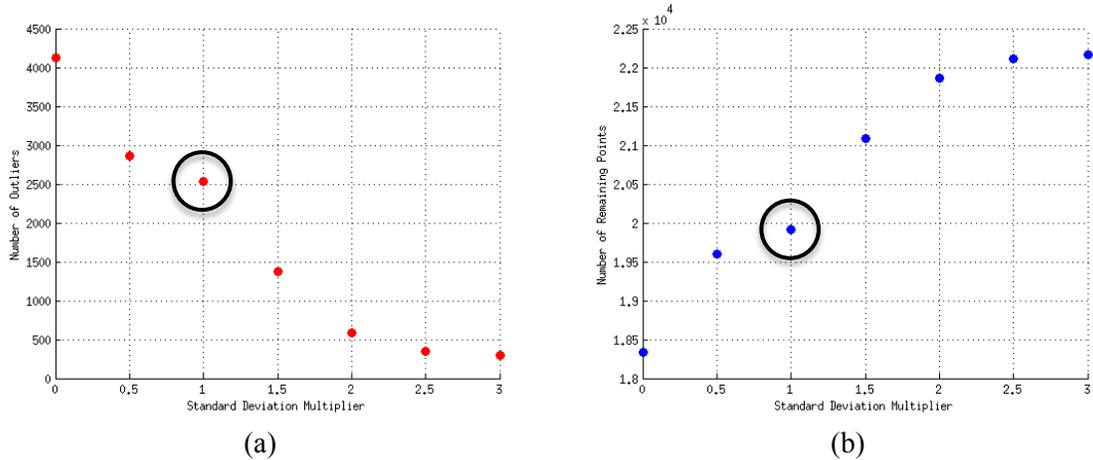


Figure 7.3: (a) Graph of  $n$  versus number of outliers; and (b) graph of  $n$  versus number of remaining points.

This analysis shows that the number of detected outlier points is dependent on the parameters  $k$  and  $n$ . It is important to determine the value used for these parameters carefully, since they affect the results of the damage detection algorithms.

For this research, the neighborhood sizes, which is the first required parameter for statistical outlier method, are computed automatically based on the resolution of the investigated point clouds, whereas the standard deviation multiplier, the second parameter, is kept constant at  $n = 1$  for the applications reported in Chapter 10. The neighborhood size selection is performed by following a similar methodology that has already been discussed in this section. For each investigated point cloud, the number of detected outliers for varying  $k$  values is computed. These  $k$  values start with 3, which is the minimum number of points required for plane fitting and then, they are followed by  $k = 8$ . The number of outlier points for every 8 point increment is then recorded. The results are investigated to find the peak number of detected outlier points and its associated  $k$  value, similar to the application shown in Figure 7.2. It should be noted that the computed neighborhood sizes vary significantly from point cloud to point cloud; as

an example, for the concrete testing frame and DeKalb County Bridge, the neighborhood size is computed as 16; however, for the Bowker Overpass, which has a very dense surface point distribution, the neighborhood size is determined as 296. The details of the statistical outlier method are discussed in Section 5.3.2.

#### 7.1.2 Extraneous Point Removal

The statistical outlier method removes noise from the surface data; however it is not very efficient for removing the points that belong to small extraneous objects that are not a part of the investigated surface.

Figure 7.4 shows the image of the concrete testing frame. Since this setup was used to test both dynamic and static loading effects after the sudden base column removal, several loading pins are attached to beams and columns of the frame in order to hang some loads. These loading pins are not a part of the original structure, so they should be removed before any further investigation. The zoomed in image in Figure 7.4 shows some of the loading pins attached to the middle column. The details of this test setup are discussed in Section 10.1.3.

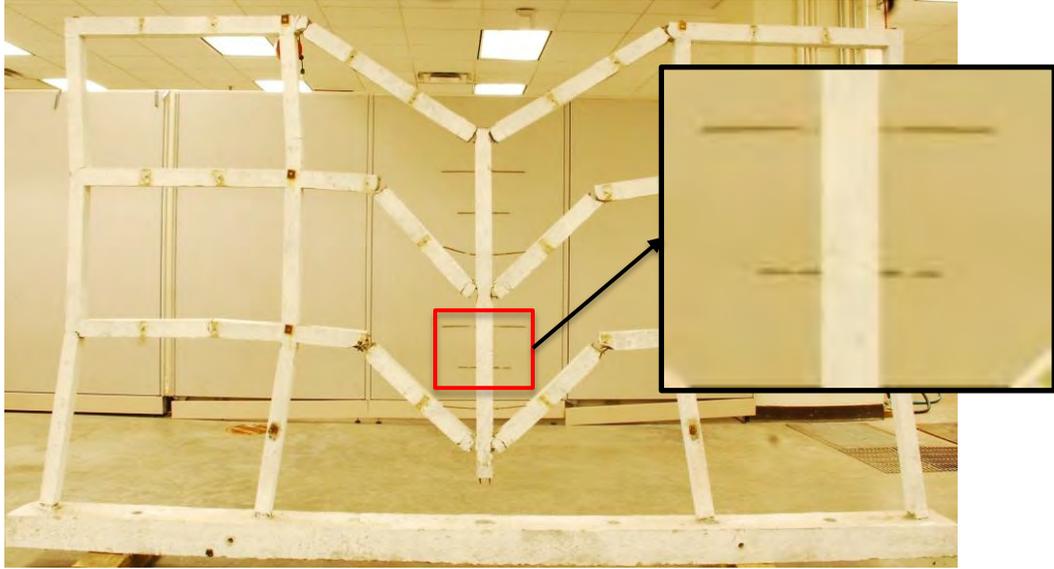


Figure 7.4: Image of concrete testing frame; and zoomed-in image of the investigated point area that has bolts attached to the concrete surface.

The raw point cloud of the portion shown in Figure 7.4, before the application of the noise removal methods, is given in Figure 7.5(a). Figure 7.5(b) represents the remaining point cloud after the statistical outlier removal method is applied. This method removes noise from the surface points and it also eliminates some of the points associated with the extraneous objects, which are loading pins in this case. However, some of the points belonging to the extraneous objects remain in the resulted point cloud. The next step is to apply the extraneous point removal in order to filter the remaining extraneous points. Figure 7.5(c) shows the result of the extraneous point removal. Since this method uses variations in the curvature of the points, it is more sensitive to local changes. The extraneous point detection method, described in Section 5.6, successfully eliminates the remaining outliers from the dataset.

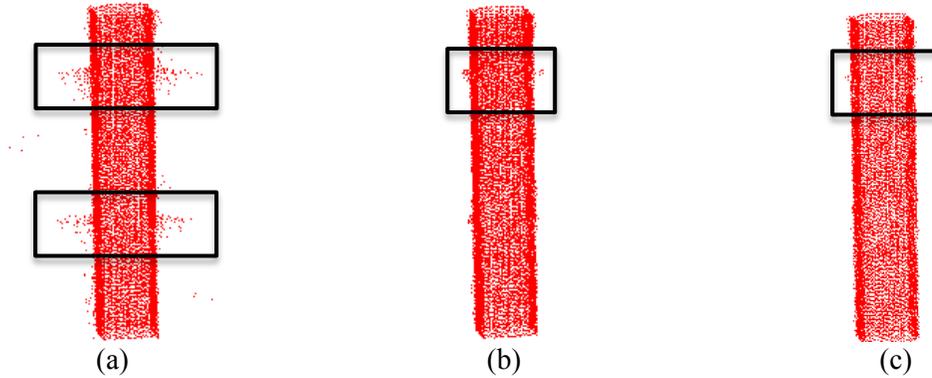


Figure 7.5: (a) Raw point cloud coming from area shown in Figure 7.4 that has 13,021 points; (b) resulting data set after statistical outlier removal with 12,630 points; and (c) resulting data set after extraneous point removal with 12,067 points.

Similar to the statistical outlier method, extraneous point removal is dependent on two parameters that are the neighborhood size  $k$  and the standard deviation multiplier  $n$ . The neighborhood size is kept constant at  $k$  value computed for the statistical outlier method, for the concrete testing frame  $k$  is equal to 16, in order to keep two methods consistent. However, when  $n = 1$  is used, this method resulted in excessive surface point loss along with the loss of extraneous points. Thus, a sensitivity study is performed to see this method's dependency on  $n$ . Figure 7.6 represents the results of the extraneous point removal with varying  $n$ ;  $n$  is ranging between 0 and 3 with 0.5 increments. The number of the outliers for  $n = 1$  is equal to 1888, which is nearly the 15% of the processed dataset. Thus,  $n = 1$  results in excessive surface data removal. In order to avoid this problem, a larger band for the removal criteria is used, where  $n = 2$ . The results for  $n = 2$  are highlighted in Figure 7.6.

This analysis proved that the total number of the removed extraneous points is affected by the variations in the parameters  $k$  and  $n$ . It is important to carefully pick the value used for these parameters for further analysis. For this research, the neighborhood sizes are kept consistent with the values computed for the statistical outlier method; on

the other hand, the standard deviation multiplier is defined as a constant  $n = 2$ , for the applications shown in Chapter 10.

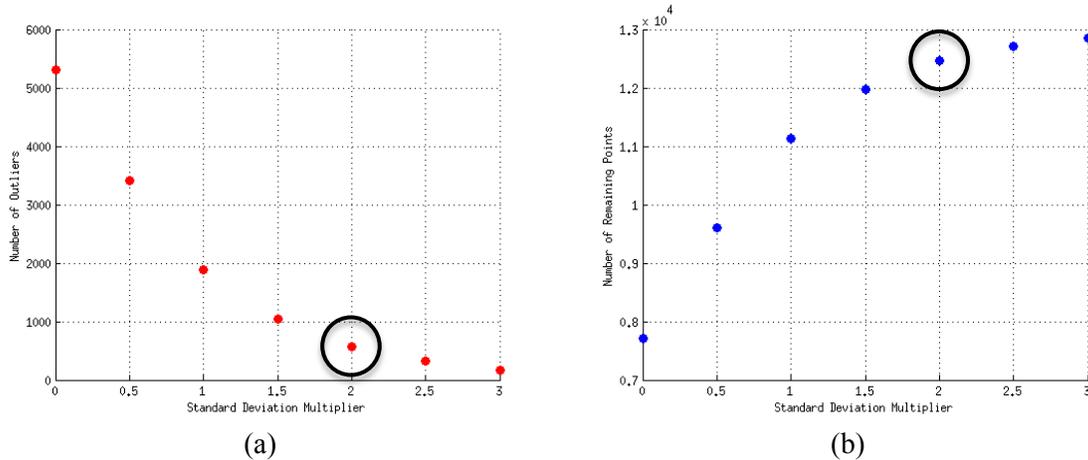


Figure 7.6: (a) Graph of  $n$  versus number of outliers and (b) graph of  $n$  versus number of remaining points.

## 7.2 Graph-based Damage Detection

Once the outliers and the extraneous points are removed from the system, and the point clusters are formed, the next step is to perform damage detection by using the developed damage detection algorithms. The first damage detection method developed for this research is called graph-based damage detection. This method is an extension of the graph-based object detection described in Section 6.4. The primary aim is to use the skeleton of a detected object for localizing and quantifying local defects. This is a new approach for detecting defects, due to local discrepancies through point cloud processing, by using object detection and model comparison.

First, the voxel model of the unorganized point cloud cluster is extracted and this model is used to create a 3D binary image. Second, this 3D binary image is divided into 2D binary images along the length of the cluster (cross-section cuts); subsequently, these

images are processed by thinning and/or distance transformation to produce skeleton-graphs and also used to calculate both 2D and 3D shape descriptors. Steps listed above are explained in Section 6.4 in detail.

Defects such as material loss, element discontinuity, or deformed locations are located and quantified by comparing as-is conditions of the structure with the detected object model. Cross-sectional cuts of sections are used to calculate the perimeter, and the area of the cross-section along the length of the member for each point cloud cluster. Then, the results are compared with the model properties at the same location. Figure 7.7 represents the example skeletons for bent and discontinuous members, and the cross-sectional cuts of modal and as-is condition of a damaged C-section. The general steps for the graph-based damage detection are shown as a flowchart in Figure 7.8.

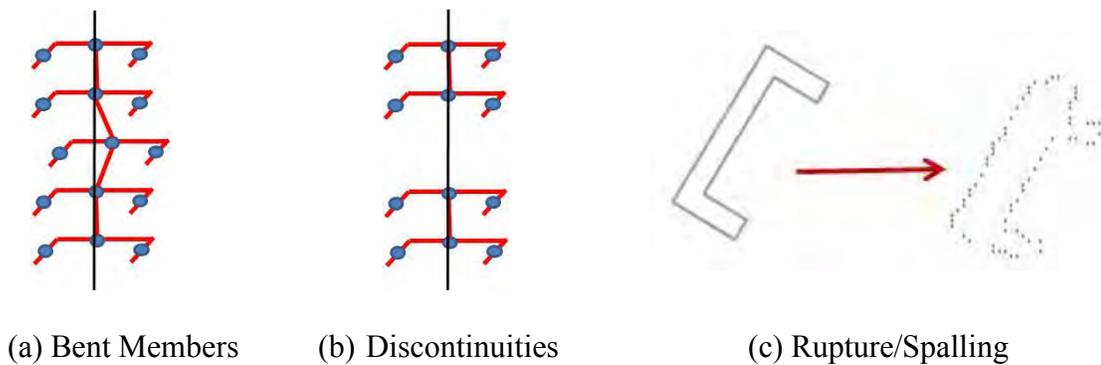


Figure 7.7: (a) Skeleton of a bent member; (b) skeleton of a discontinuous member; and (c) modal and point cloud section cuts of a C-section.

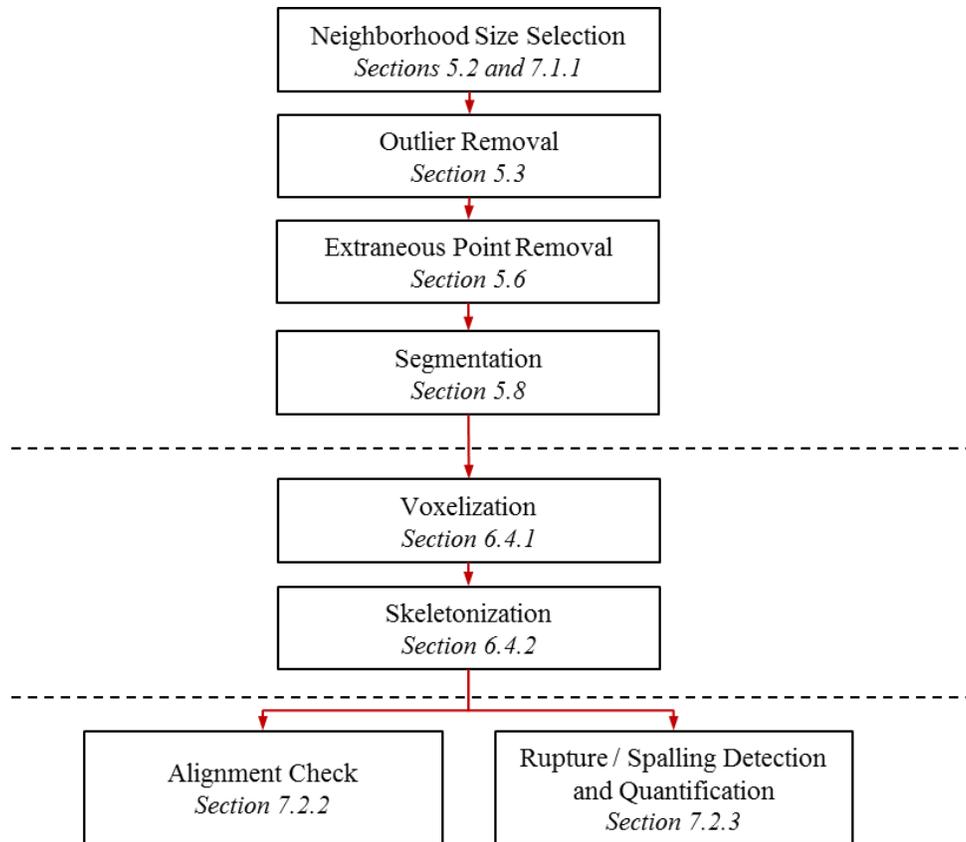


Figure 7.8: Steps of graph-based damage detection.

### 7.2.1 Surface Patch Investigation for Voxelization

Surface patch investigation is an important pre-processing step for both graph-based damage detection and surface normal-based damage detection methods, which are discussed in Sections 7.2 and 7.3 respectively.

Voxel size is an important parameter for the voxelization method for surface patch investigation. As was mentioned in Section 6.4, the selected voxel size affects the results of object detection, since it alters the volumetric representation extracted from point clouds. Volumetric representations are used to get the skeleton and the cross-section information at varying lengths. Both the skeleton and the cross-sections are later used for damage detection as explained in the following sections.

Voxel size is determined based on the resolution of point clouds. Laser scanners capture point clouds around the structure to form a complete 3D representation. However, even if all the scans provide coverage for the entire test specimen, they are not adequate to obtain a uniform point density at the surface across the entire structure, due to the limited number of scans. For most of the laser scanning applications, the main objective is to cover all surfaces of an investigated structure. Thus, it is difficult to obtain a uniform point density on the surface of structures.

In order to decide on the appropriate voxel size and to understand the smallest detectable damage size, it is necessary to get the point resolution at varying locations on the surface of structures. Surface patches, which are extracted from detected regions, of size 1.2 *in* by 1.2 *in* are used to compute the surface resolution in this research. Since the resolution study is performed on the concrete testing frame; the size of the patch is determined based upon the minimum dimension of the frame. The representative surface patch dimension is selected such that it would be larger than the half of the minimum dimension, which is 2"; thus, 1.2 *in*. is used as the patch dimension. The location of the patch is automatically selected by using the curvature information. The point with the least curvature is chosen to be middle point of the patch. The computation of curvature values is discussed in Section 5.5. If the point is close to one of the edges, and it is not possible to collect the patch data, then the next point with the lowest curvature value is checked.

To demonstrate the variation of the point densities over the surface, a study is conducted, in which surface resolutions at varying locations on the concrete testing frame are computed. The image of the concrete testing frame is given in Figure 7.4, and beam

and column legends are shown in Figure 7.9. The details for the concrete testing frame are discussed in Section 10.1.3.

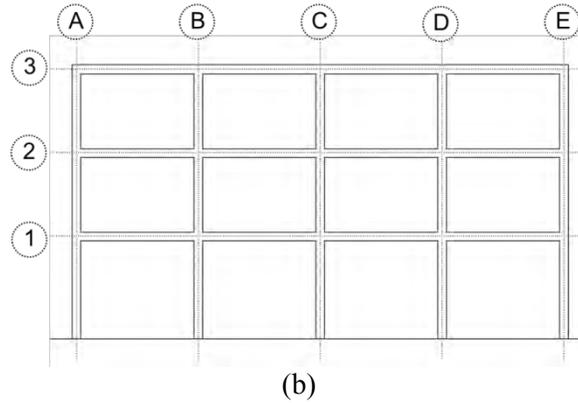


Figure 7.9: Beam and column legends of concrete testing frame.

Several surface patches are checked from different portions of the structure and point densities are calculated accordingly. Results are given in Table 7.1. As an example, BC-3-1 represents the top beam located between columns B and C; the last number indicates where from is the patch taken. If it is from the front, then the value is 1, and if it is from the back, this value is 4. It can be seen from Table 7.1 that point density values for the lower part of the test setup, BC-1, are lower than the ones calculated for BC-2 and BC-3. These point densities are subsequently used to determine voxel dimensions for different patches. Voxel grid dimensions, which are obtained separately for varying point densities on the surface, are determined by using a targeted resolution value ( $\Delta_x = \Delta_z$  and  $\Delta_y = 1$ ). Voxel grid dimensions for the investigated surface patches are given in Table 7.2.

Table 7.1: Point densities from varying surface patches.

	<b>Side Length (in)</b>	<b>Area (in<sup>2</sup>)</b>	<b># of points</b>	<b>Average Point Density (points/in<sup>2</sup>)</b>
<b>BC-1-1</b>	1.2	1.44	187	129.86
<b>BC-1-4</b>	1.2	1.44	215	149.31
<b>BC-2-1</b>	1.2	1.44	262	181.94
<b>BC-2-4</b>	1.2	1.44	276	191.67
<b>BC-3-1</b>	1.2	1.44	312	216.67
<b>BC-3-4</b>	1.2	1.44	323	224.31
<b>CD-2-1</b>	1.2	1.44	261	181.25
<b>CD-2-4</b>	1.2	1.44	420	291.67
<b>CD-3-1</b>	1.2	1.44	309	214.58
<b>CD-3-4</b>	1.2	1.44	545	378.47

Table 7.2: Voxel grid dimensions.

	<b>Side Length (in)</b>	<b># of points</b>	<b># of voxels per side</b>	<b>Voxel Dimensions (in)</b>
<b>BC-1-1</b>	1.2	187	14	0.09
<b>BC-1-4</b>	1.2	215	15	0.08
<b>BC-2-1</b>	1.2	262	17	0.07
<b>BC-2-4</b>	1.2	276	17	0.07
<b>BC-3-1</b>	1.2	312	18	0.07
<b>BC-3-4</b>	1.2	323	18	0.07
<b>CD-2-1</b>	1.2	261	17	0.07
<b>CD-2-4</b>	1.2	420	21	0.06
<b>CD-3-1</b>	1.2	309	18	0.07
<b>CD-3-4</b>	1.2	545	24	0.05

The results show that the upper beams, which were exposed to the scanner more than the lower beams, due to the location of the scanner during scanning, have higher point densities for the given unit area. The number of points per unit area, the point density, is affected from both the number of scans coinciding at certain location and the angle of the laser scanners with respect to the patch location. Examining the point cloud resolution at different locations is important to understand the limitations of the performed damage detection. Especially for cracks, the point cloud resolution plays an important role, if the point density is smaller than the crack width, then it is not possible

to locate this defect. Thus, an adaptive neighborhood size selection strategy is developed. The neighborhood parameter  $k$  is selected based on the point cloud resolution for the developed algorithms in order to effectively deal with this variation in the resolution.

### 7.2.2 Alignment

The next step, once the grid size is determined from resolution analysis, is to create a voxel grid of each investigated point cluster and then, use this voxel grid, which represents a 3D binary image, for graph-based damage detection.

Graph-based damage detection is used to check two types of defects. First, the alignment problems of individual members are checked and second, the cross-sectional cuts are used to compute the local changes and to investigate them in 3D.

In this section, alignment comparison is discussed. The skeleton of each object is created based on the steps described in Section 6.4. Skeletons provide the necessary topological information required for computing possible alignment problems related with individual objects. This application is especially beneficial for inspecting structures that have exposed members such as bridges and for monitoring construction process.

This method requires knowledge on the objects and their relative orientations with each other. As mentioned in Chapter 6, in this research, a predefined object library is used for storing information on several object types. However, these stored object descriptions do not provide orientation information. The orientation of each object has to be extracted from the formed skeleton. Some of the basic conceptual relationships such as the orientations of beams and columns can be automatically deduced from point clouds. On the other hand, for the cases where the skeleton is not sufficiently representative of

the initial object orientation, some additional user input is required. Ideally, Building Information Models (BIMs) are used to import this required information on object orientation.

In both cases, the as-is skeleton of each object is compared with either the extracted or the BIM defined object representation. The discrepancies are detected, and the necessary damage indices such as vertical sag and/or horizontal movement, are computed. The related indices are selected based on the available damage detection criteria provided by the Federal Highway Administration (AASHTO., 2011). From state to state, the indices and their limitations vary significantly. In this research, the Bridge Inspection Manual of Ohio State Department of Transportation (Ohio Department of Transportation., 2010) is primarily used, since it provides detailed defect descriptions and condition rating criteria for most of the individual components of bridges. The details of the implementation of current inspection methodologies into laser-based damage assessment are discussed in Chapter 9.

Once the skeleton of the as-is condition of an object is obtained by using the methods described in Section 6.4.2, the individual points of the skeleton are compared with their counterparts in the skeleton of the undamaged representation of the same object. The same process is repeated for all the detected objects in a scene, and the results are recorded. Figure 7.10 represents the skeleton of the concrete testing frame, which is discussed in Section 10.1.3, including a zoomed-in view of one of the base columns.

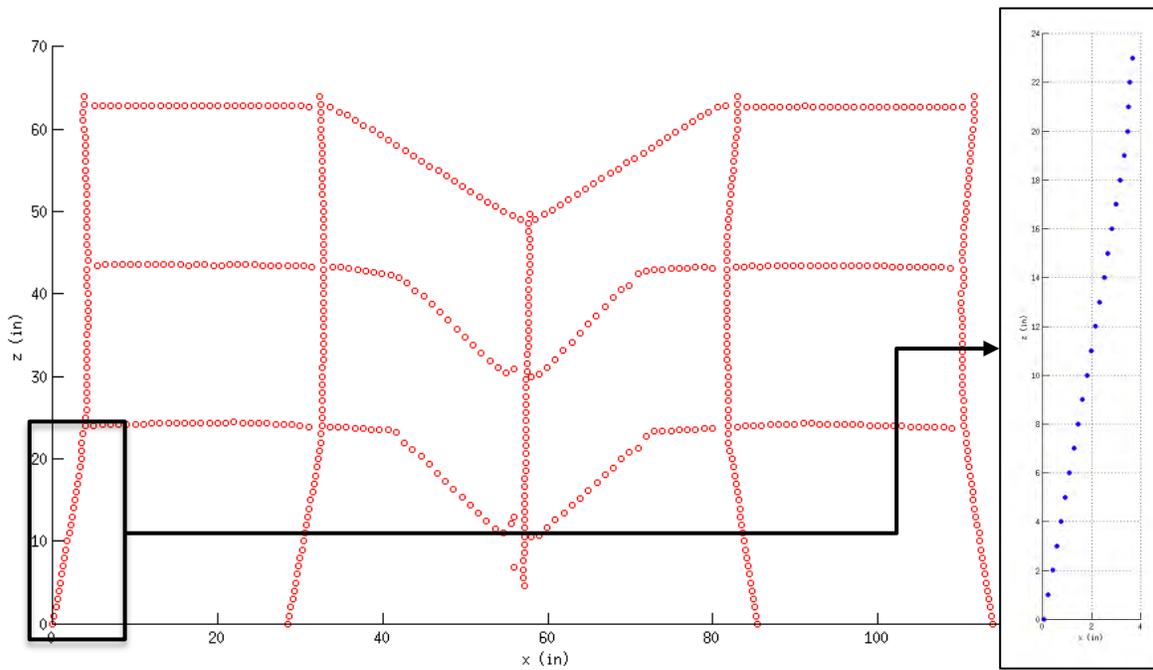


Figure 7.10: Complete skeleton of concrete testing frame; and zoomed-in representation of one of the base columns.

This test setup was restricted to deflect in the y-direction during loading. Thus, the base column was subjected to significant horizontal movement. The horizontal movement values are quantified along the length of the object by using the developed algorithms for alignment comparison. Each point that is defined at each height increment location is used to compute the value of the horizontal movement at that specific location. Table 7.3 shows the results of the alignment comparison algorithm for the column highlighted in Figure 7.10. The parameters and the corresponding ranges for the described alignment comparison method are given in Table 7.4.

Table 7.3: Results of alignment algorithm for the investigated column.

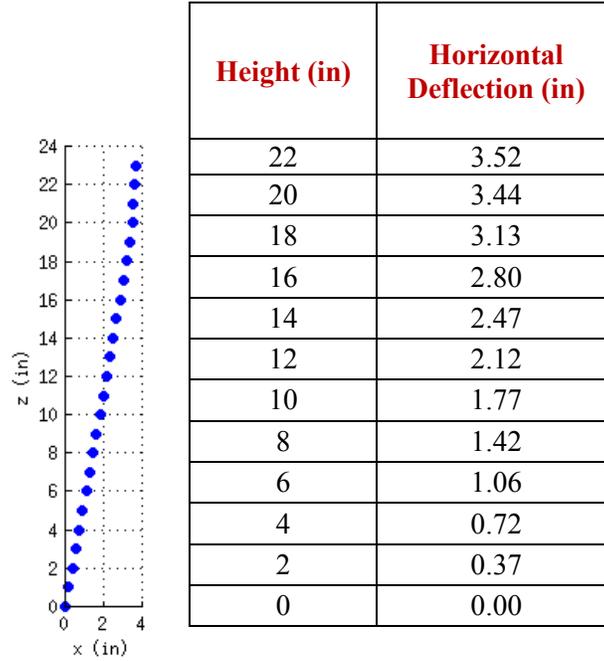


Table 7.4: Parameters and corresponding ranges for alignment comparison.

Method	Parameters	Range
Alignment comparison	Height function increment, $h_p$	0.1 in. – 2 in.

### 7.2.3 Rupture / Spalling

Cross-section comparison is the second main component of graph-based damage detection. Once the voxel grid is formed and the skeletons of individual objects are extracted, the skeletal point locations are used to obtain the cross-section cuts, at each height function increment  $h_p$ , from the investigated object. If the voxel grid of an object is treated as a 3D binary image, then, for the investigated object, each cut made in z-direction results in a 2D binary image of the cross-section,  $CS_i$ , at the corresponding z-elevation. The details of voxelization and skeletonization are given in Sections 6.4.1 and 6.4.2, respectively.

Each active pixel of the 2D binary image is used to create a matrix of points that is used to compute both perimeter and area of each investigated cross-section,  $CS_i$ . In order to calculate the area of each cross-section cut, a connectivity map of all the points that forms the point matrix of  $CS_i$  is created. The main principle of this method is to use the surface points to create a connectivity map that will represent the perimeter of the cross-section, when the points are plotted with an order. For a given list of 2D points, a singly connected nearest-neighbor path in either clockwise or counter-clockwise directions is constructed. The described algorithm is a specialized version of the *points2contour* algorithm developed by Ursell (2013) for MatLab.

First, a random point  $p_q$  is selected from the point matrix, and it is set as the first point of the connectivity map. Second, the algorithm starts looking for the nearest-neighbor in the specified direction, which is either clockwise or counter clockwise. Once the closest point is found, it is labeled as the second point of the connectivity map. The algorithm keeps iterating until all the points in the point matrix are processed. The outliers, which are detected based on radius-based outlier removal that is described in Section 5.3.1, are not included in the connectivity map. There is not a unique solution for mapping of points into a connected contour; especially, if there are more than two neighborhood points, or when the nearest neighbor matrix is not symmetric. In this case, selecting a different initial  $p_q$  might result in a different contour. However, it is observed that the mentioned alteration affects neither the perimeter nor the area computations significantly. The variations in the calculated perimeter values are within 4% range, whereas the variation in area computation is in 2% range. In Figure 7.11(a), a drawing of one of the C-sections that forms the steel testing frame and the height function direction

for cross-section extraction are shown. Figure 7.11(b) represents one of the C-section cuts taken from the investigated steel member, which is processed with the discussed rupture/spalling detection algorithms, and the two possible processing directions for the connectivity map construction are shown.

Once the connectivity map is generated, several readily integrated MatLab functions are used to compute both perimeter and area of the cross-section,  $CS_i$ . First, a water-tight polygon is created by using the connectivity map to get the perimeter and then, the *polyarea* function is used to compute the area. The *polyarea* function returns the area of any polygon that is specified by the vertices defined in the connectivity map. The obtained results are compared with the original state of the object to locate and quantify the discrepancies. The percentage change in cross-sectional area calculated at varying elevations for the C-section, which is shown in Figure 7.11, are given in Table 7.5.

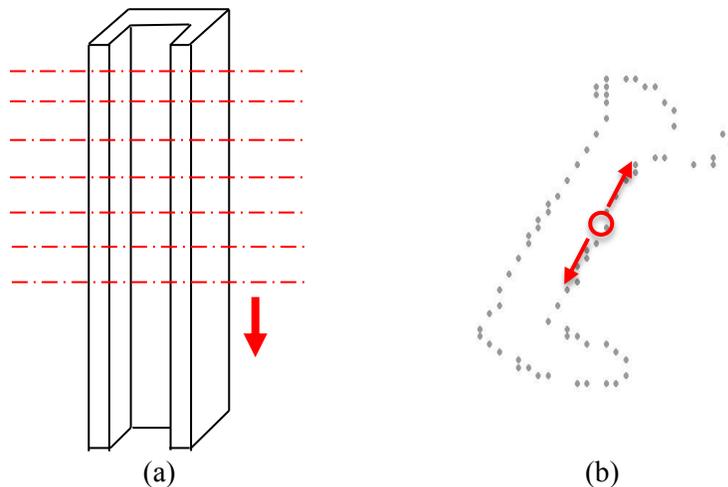


Figure 7.11: (a) Drawing of one of the C-sections of steel testing frame; and (b) a representative cross-section with possible processing directions.

The voxel grid representation is also used to compute the volumetric change throughout the length the investigated object. Each  $CS_i$ , which is represented by a 2D binary image, is taken as a single volumetric unit for which the height is taken as  $h_p$ . Later, the volume of the entire object at its as-is state is calculated by adding up the volume computations associated with each  $CS_i$  along the height of the object. This computed volume is then compared with the original volume of the object, in order to obtain the volumetric change (Guldur and Hajjar, 2013). The parameters and the corresponding options for the described area computation method are given in Table 7.6.

Table 7.5: Percentage change in cross-sectional area calculated at varying elevations.

<b>Elevation (in)</b>	<b>Area (in<sup>2</sup>)</b>	<b>Change in Cross-Sectional Area (%)</b>	<b>Change in Volume (in<sup>3</sup>)</b>	<b>Change in Volume (%)</b>
0	1.55	6.1	0.55	1.7
2	1.77	7.2		
4	1.57	5.2		
6	1.67	0.9		
8	1.70	2.7		
10	1.66	0.6		
12	1.66	0.2		
14	1.66	0.4		
16	1.65	0.1		
18	1.65	0.1		
20	1.57	4.7		

Table 7.6: Parameters and corresponding ranges for area computation.

<b>Method</b>	<b>Parameters</b>	<b>Options</b>
Area computation	Direction for connectivity map	Clockwise and counter clockwise

### 7.3 Surface Normal-based Damage Detection

The second method for camera integrated laser-based damage detection relies on the modal properties of the detected surfaces and/or objects. The relative orientation of the estimated surface normal with respect to a reference normal is used to locate the defected areas on the surface of structures. The reference normal can be a surface normal computed via surface patches; the normal representing the skeleton of the detected object; or the normal vector between a reference point and the current query point. The listed reference normal vector representations are shown in Figure 7.12. In Figure 7.12, the dashed arrows represent the reference normal and the solid arrows show the computed surface normals at  $p_q$ .

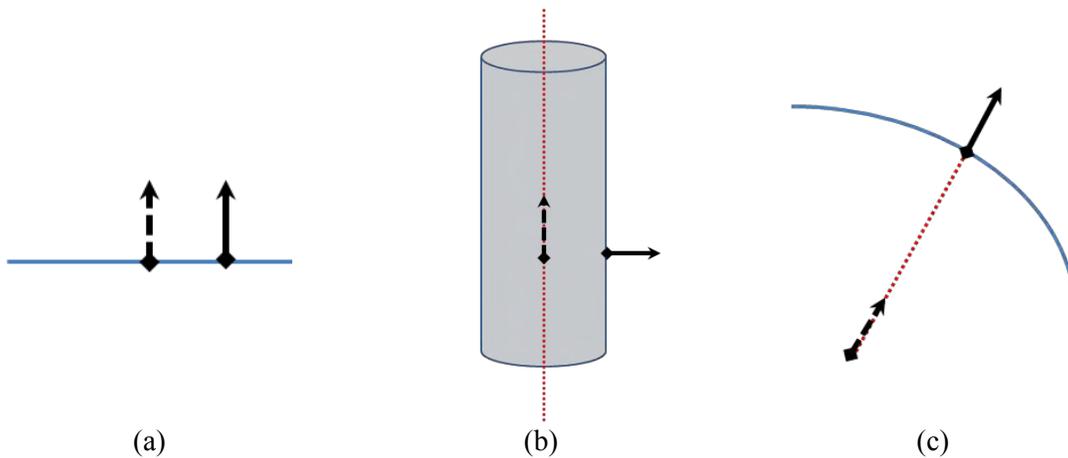


Figure 7.12: Reference normal representations: (a) from surface patches; (b) from object skeleton; and (c) from a reference point.

The surface-normal based damage detection method described in this section expands the methodology discussed in Torok et al. (2013). Torok et al. (2013) introduced the idea that every structural component has a centroid and that the normal vector of any surface triangle, which was obtained by connecting adjacent pixels on the surface, would

be perpendicular to this centroid. Any triangle that has a normal vector, which falls out of a certain angle range, is marked as a crack. Torok et al. (2013) is using images instead of point clouds in order to create 3D representation of the investigated locations.

For this research, instead of the one centroid assumption for individual objects (in this work, centroid is referred as skeleton), a larger domain of possible normal comparison schemes are introduced as shown in Figure 7.12. The reference normals are compared with the estimated surface normals, in order to detect locations where unexpected normal orientations are observed. The details of the method are given in Section 7.3.2. Both the method developed in this work and that presented in Torok et al. (2013) are more suitable for detecting defects that are large enough to cause significant variations on surface normals. Figure 7.13 represents examples of normal orientations for both undamaged and damaged columns.

The method developed in this work uses varying numbers of  $k$  neighborhood points while determining the surface normal associated with an investigated  $p_q$ . When large  $k$  values are used, the fitted surface for the normal estimation eliminates some of the small local changes. Thus, smaller defects such as cracks, which have widths close to the resolution of the surface points, cannot be detected. It is required to use different values of  $k$  for different defect types. The effect of  $k$  in normal comparison is discussed in Section 7.3.4.

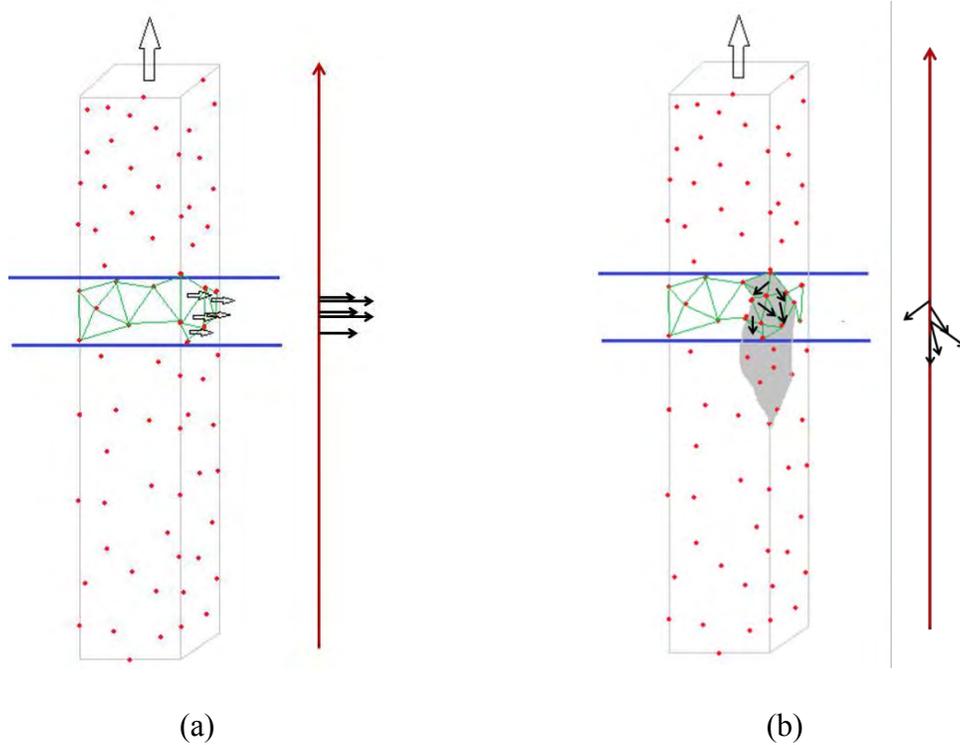


Figure 7.13: (a) Normal orientations of an undamaged surface; and (b) normal orientations of a damaged surface.

As was mentioned above, using only the normal variations on the surface would be sufficient for detecting larger defects. However, another parameter should be introduced if certain defects with sizes close to the resolution of the scanner are to be detected. Thus, the pixel information (intensity) obtained from texture-mapped point clouds is used for enhancing the detection capabilities for smaller defects; the damage detection method developed for texture-mapped point clouds is discussed in Section 7.3.3.

The general steps of the surface normal-based damage detection are listed in the flowchart given in Figure 7.14.

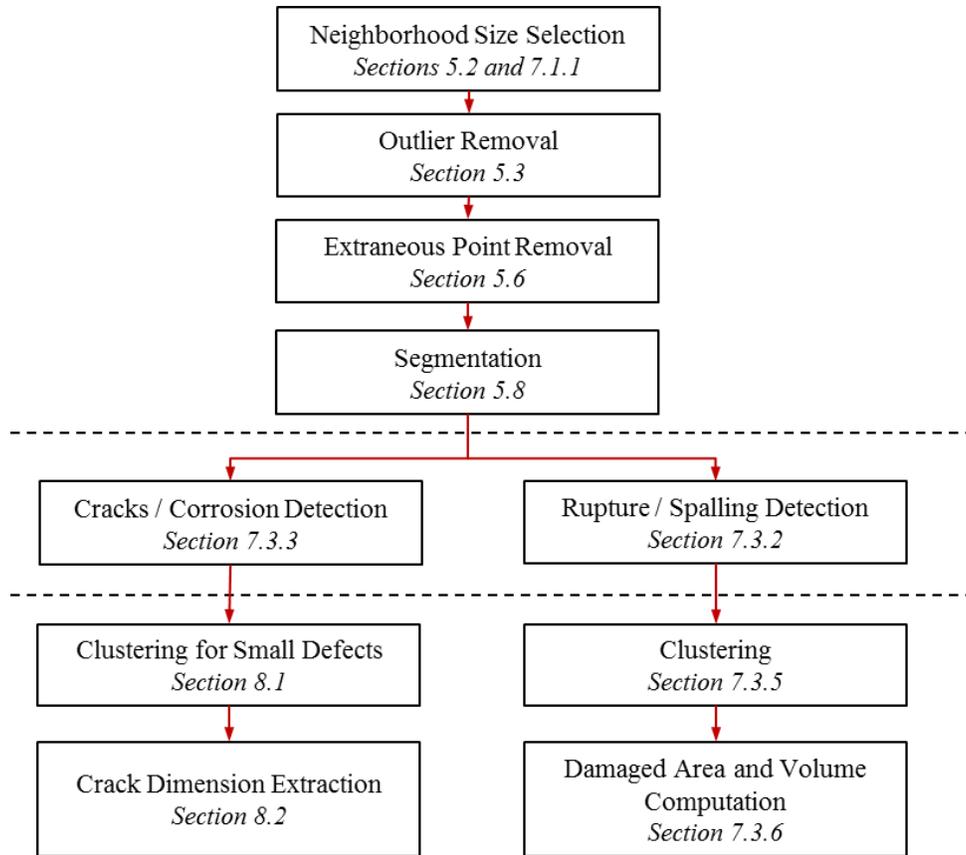


Figure 7.14: Steps of surface normal-based damage detection.

### 7.3.1 Surface Patch Investigation for Computing Reference Normals

Surface patch investigation is a crucial step in the pre-processing phase of the surface normal-based damage detection, if the object information that reserves properties such as the skeleton or the reference point is not available. In order to perform the surface normal-based damage detection on the surfaces shown in Figure 7.12(a), it is required to extract the normal orientation from the undamaged portion of the underlying surface. This normal will be compared to the surface normal associated with each point,  $p_i$  on a segmented surface,  $S_j$ . The methodology for segmentation via region growing is discussed in Section 5.8.

The normal comparison is trivial when the skeleton or the reference point information is already available. The normal of query point  $p_q$  is computed by using neighborhood points and then compared with either the normal representing the skeleton, or the normal between the reference point and the investigated  $p_q$ . However, in some cases only the surface data is available and it is not possible to use a skeleton or a reference point. In this case, a couple of patches extracted from undamaged portions of the surface are used to get an average value for the reference surface normal.

The patches are selected from the undamaged locations automatically. This is ensured by using the curvature information. The point with the least curvature for a neighborhood size of  $k_p = 8 \cdot k$  is selected as the seed point;  $k$  has already been computed for the statistical outlier removal, the details are discussed in Section 7.1.1.  $k_p = 8 \cdot k$  instead of  $k$  is used in order to capture the local changes in a larger area. Once the first seed point is selected, the point and  $k_p$  are used to estimate the surface normal. The second point is selected from the remaining data set, once the point with the least curvature and its neighborhood points are removed from the dataset. The point with the least curvature after the point removal is selected as the second seed point and the normal is computed. The same steps are finally repeated for the third time and the average of the three normals is calculated. This average normal is used as the reference normal in further steps of surface-normal based damage detection. The following two sections describes how the relationship between the reference normal and computed surface normals are used for performing damage detection.

### 7.3.2 Normal Comparison for Plain Point Clouds

The basic principles of surface normal-based damage detection are discussed at the beginning of Section 7.3. In this section, the parameters and the ranges for these parameters are discussed. This section focuses on point cloud data that lacks color information (intensity); the normal comparison methodology for the point clouds that include color information, which is an extension of the method described in this section, is later in this chapter, in Section 7.3.3. For plain point clouds, only the geographic locations of the surface points are available for point cloud processing. The unavailability of the color information prevents this type of dataset from being used for detecting defects such as cracks or corrosion. Cracks and corrosion result in slight local changes which in most cases are smoothed out during the surface normal estimation; thus, they are commonly not detectable without color information.

Figure 7.15 represents the examples for surface damage and the normals on the damaged area for the three possible schemes shown in Figure 7.12; the reference normals are shown with dashed lines. In Figure 7.16(a), the reference normal  $n_r$  and one of the surface normals,  $n_i$ , from the damaged area are shown in red and in Figure 7.16(b), the comparison angle between these two normals is given as  $\alpha_{ri}$ .

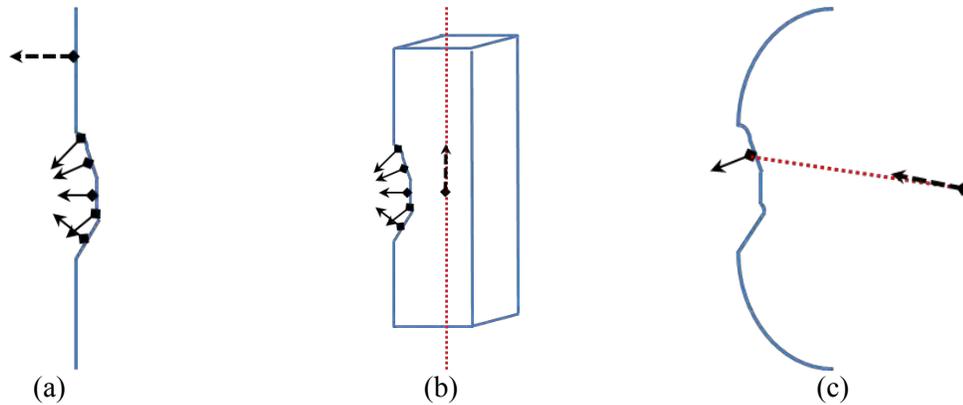


Figure 7.15: Representations of reference normals and normals from defect areas: (a) from surface data; (b) from object with a skeleton; and (c) from a surface with a reference point.

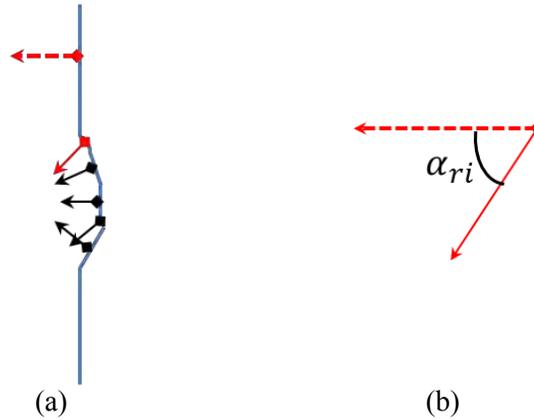


Figure 7.16: (a) Reference normal and one of the surface normals from a representative damaged area, shown in red; and (b) comparison angle between these two normals.

Once  $\alpha_{ri}$  is computed for all the points in a dataset, the computed angle values are compared with a threshold value. This angle threshold  $T_\alpha$  is used to differentiate undamaged locations on the surface from the damaged locations. For this work, varying  $T_\alpha$  are used for extracting the defect locations depending on the quality of the point cloud. The range of  $T_\alpha$  is between  $0^\circ$  to  $10^\circ$  or  $170^\circ$  to  $180^\circ$ . Since it is not possible to determine the direction of a normal unless the viewpoint of the sensor is known, the reciprocal angle threshold ( $170^\circ$  to  $180^\circ$ ) is always checked. For denser point clouds, generally  $T_\alpha = 10^\circ$  is used. However,  $T_\alpha = 5^\circ$  is used for a less crowded dataset. It is possible to detect local changes more effectively from denser datasets. The variation in computed  $\alpha_{ri}$  is significantly larger for smaller  $k$  values. The effect of  $k$  in  $\alpha_{ri}$  is discussed in Section 7.3.4. Since the variation is higher, a larger value for  $T_\alpha$  must be used in dense point clouds.  $T_\alpha$  and the corresponding range is given in Table 7.7. Table 7.7 only includes the parameters introduced in this section; any prior dependencies are given in the related sections. For example, parameter  $k$  and its range are discussed in the surface normal estimation section, Section 5.4.

Table 7.7: Parameters and corresponding ranges for normal comparison for plain point clouds.

<b>Method</b>	<b>Parameters</b>	<b>Range</b>
Normal Comparison for Plain Point Clouds	Angle threshold, $T_\alpha$	0°-10°

Figure 7.17(a) is representing a portion of the point cloud extracted from the bottom of the deck of the DeKalb County Bridge. Figure 7.17(b) shows the defect detection results for  $k = 16$  and  $T_\alpha = 10^\circ$ , the points from the defect locations are shown in blue. The principles of the automated clustering and the area and volume calculations for the detection results are discussed in Sections 7.3.5 and 7.3.6.

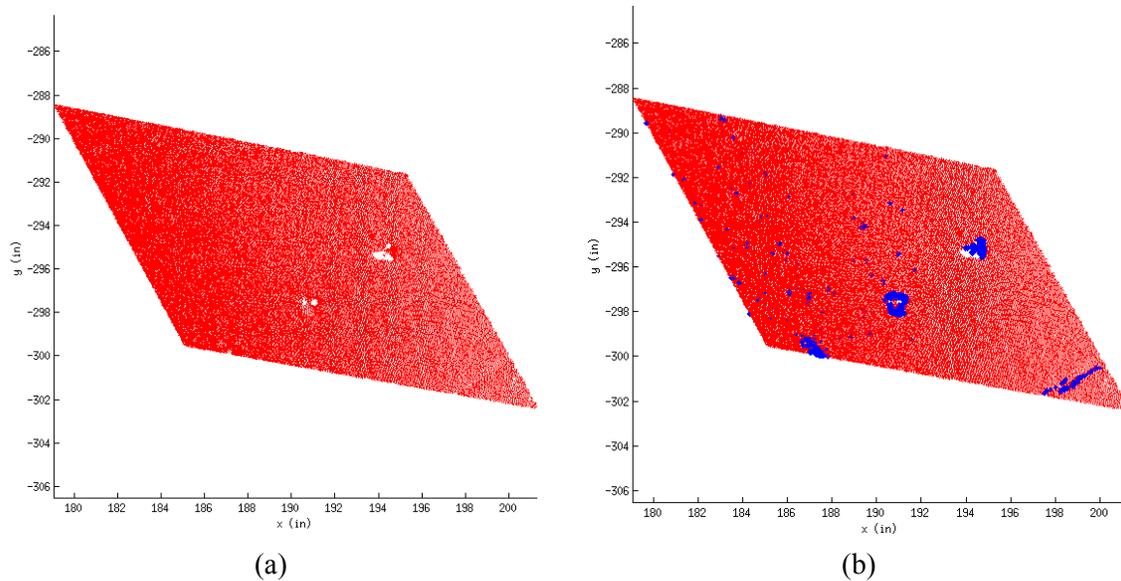


Figure 7.17: (a) Original point cloud remaining after outlier removal for a portion of the bottom deck of DeKalb County Bridge; and (b) defect locations after the application of surface normal-based damage detection.

### 7.3.3 Normal Comparison for Texture-mapped Point Clouds

The normal comparison for texture-mapped point clouds uses the same principles described in Section 7.3.2. The only difference is that this method benefits from the color information associated with each point, in order to complete the defect detection.

The adverse effect of the increase in neighborhood size  $k$  for surface normal estimation on detecting smaller defects such as cracks was mentioned before. In order to overcome this, another threshold is added to the developed algorithms. Whenever the color information is available and a small  $k$  value is used, a threshold value represented by  $T_{int}$  is used to reprocess the elected candidate defect points. The points that remained after this second thresholding, represent the defect locations on the surface. The threshold is defined for the intensity values instead of the RGB values since the color output of the used laser scanners are in grey-scale. The range for the  $T_{int}$  is given in Table 7.8.

Table 7.8: Parameters and corresponding ranges for normal comparison for texture-mapped point clouds.

<b>Method</b>	<b>Parameters</b>	<b>Range</b>
Normal Comparison for Texture-mapped Point Clouds	Intensity threshold, $T_{int}$	0-1

The rest the section focuses on the results of the proposed method obtained by using two  $k$  values and two  $T_{int}$  values. In order to achieve this purpose, two locations on the concrete testing frame are investigated; images related to each location are shown in Figure 7.18. The beam and column legends for the concrete testing frame shown in Figure 7.4 are given in Figure 7.9.

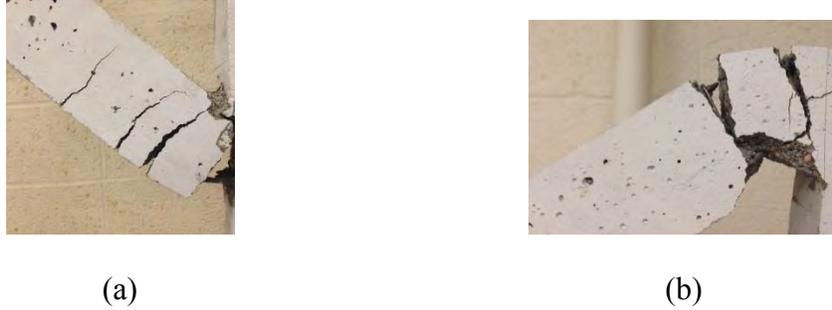


Figure 7.18: Images of (a) BC-1-1 and (b) CD-3-1.

As in the previous section, normals are calculated on surface of the investigated point cluster, in order to extract local properties. The angle  $\alpha_{ri}$  between each normal vector  $n_i$ , which is determined for individual points, and the reference surface normal  $n_r$ , which is calculated by using patches from undamaged portions of the structure, are then calculated. If the deviation from the  $n_r$  is between 10 and 170 degrees,  $10^\circ < \alpha_{ri} < 170^\circ$ , then the point is labeled with a red dot and listed as a possible defect location. Figure 7.19 represents the deviated normals from the surfaces BC-1-1 and CD-3-1 for a neighborhood size of 6; red dots represent the surface normals that satisfy the condition:  $10^\circ < \alpha < 170^\circ$ .



Figure 7.19: Normal deviations for (a) BC-1-1 and (b) CD-3-1 for  $k = 6$ .

The results are dependent on the intensity threshold  $T_{int}$ , which is automatically calculated based on the intensity distribution among the investigated surface patches and the neighborhood size  $k$ , that is defined by the user for the displayed application. Figure 7.20 represents the labels given for the investigated defect locations for both BC-1-1 and CD-3-1. Figure 7.21 represents the results for  $k = 3$ . The intensity threshold calculated for BC-1-1 is 0.63, shown in Figure 7.21(a), and it is equal to 0.78 for CD-3-1, shown in Figure 7.21(b).  $T_{int}$  is adjusted so that no more than 20% of the intensity values are lower than the threshold.



Figure 7.20: Labeled cracks for (a) BC-1-1 and (b) CD-3-1.

In order to investigate the effect of neighborhood size on the results, the detection process was repeated for  $k = 6$  and the results are shown in Figure 7.22. Intensity thresholds are kept constant for both BC-1-1 and CD-3-1.

It can be observed from Figure 7.21(a) that a smaller neighborhood size,  $k = 3$ , yields better results for BC-1-1 compared to  $k = 6$  for which the algorithm failed to detect one of the cracks, BC1. However, for CD-3-1 even if decreasing the neighborhood size slightly improved the results at CD1 and CD2, it also resulted in false positives at the bottom, left bottom of Figure 7.21(b).



Figure 7.21: Damage detection results for  $k = 3$ , where (a) BC-1-1 ( $T_{int} = 0.63$ ) and (b) CD-3-1 ( $T_{int} = 0.78$ ).



Figure 7.22: Damage detection results for  $k = 6$ , where (a) BC-1-1 ( $T_{int} = 0.63$ ) and (b) CD-3-1 ( $T_{int} = 0.78$ ).

Table 7.9 and Table 7.10 represent the thicknesses (largest crack width) and the lengths for the damaged locations labeled in Figure 7.20. Both measurements taken with a micrometer and the values calculated from damage detection results are given in Table 7.9 and in Table 7.10. Table 7.11 represents the percentage error values calculated based on the comparison between calculated and measured damage thickness and length values.

Table 7.9: Hand measured and calculated thickness (largest crack width) and the length values for the damage locations labeled in Figure 5 for  $k = 3$ .

		<b>Micrometer Measurement</b>		<b>Texture-mapped Point Cloud Processing</b>	
		<b>Thickness (<i>in</i>)</b>	<b>Length (<i>in</i>)</b>	<b>Thickness (<i>in</i>)</b>	<b>Length (<i>in</i>)</b>
<b>BC-1-1</b>	<b>BC1</b>	0.06	1.02	0.08	0.94
	<b>BC2</b>	0.12	1.40	0.15	1.41
	<b>BC3</b>	0.28	1.91	0.31	1.92
<b>CD-3-1</b>	<b>CD1</b>	0.68	1.21	0.56	1.33
	<b>CD2</b>	0.31	1.50	0.37	1.56

Table 7.10: Hand measured and calculated thickness (largest crack width) and the length values for the damage locations labeled in Figure 5 for  $k = 6$ .

		<b>Micrometer Measurement</b>		<b>Texture-mapped Point Cloud Processing</b>	
		<b>Thickness (<i>in</i>)</b>	<b>Length (<i>in</i>)</b>	<b>Thickness (<i>in</i>)</b>	<b>Length (<i>in</i>)</b>
<b>BC-1-1</b>	<b>BC1</b>	0.06	1.02	N/A	N/A
	<b>BC2</b>	0.12	1.40	0.19	1.06
	<b>BC3</b>	0.28	1.91	0.45	1.82
<b>CD-3-1</b>	<b>CD1</b>	0.68	1.21	0.55	1.15
	<b>CD2</b>	0.31	1.50	0.40	1.56

Table 7.11: Percentage error values for both  $k = 3$  and  $k = 6$ .

		<b>Error (%) for k=3</b>		<b>Error (%) for k=6</b>	
		<b>Thickness</b>	<b>Length</b>	<b>Thickness</b>	<b>Length</b>
<b>BC-1-1</b>	<b>BC1</b>	33.3	-7.8	N/A	N/A
	<b>BC2</b>	25.0	0.7	58.3	-24.3
	<b>BC3</b>	10.7	0.5	60.7	-4.7
<b>CD-3-1</b>	<b>CD1</b>	-17.6	9.9	-19.1	-5.0
	<b>CD2</b>	19.4	4.0	29.0	4.0

It can be observed from Table 7.11 that the error values associated with the detected thicknesses are high compared to the error values associated with the length measurements. This shows that even if the location and the length of the crack could be effectively detected with the proposed method, it is less efficient in detecting the crack thicknesses. Automated length and thickness measuring strategies are developed and discussed in Section 8.2. For the application presented in this section, the parameters are determined by the user (Guldur and Hajjar, 2014).

#### 7.3.4 Effect of the Neighborhood Size on Normal Comparison

The comparison angle  $\alpha_{ri}$ , which is individually computed for every point on a surface patch, is heavily affected by the selected neighborhood size  $k$ . As it is mentioned at the beginning of Section 7.3, the candidate points for damaged areas are extracted based on the angle between the computed reference surface normal and the normal of the query point  $p_q$ . The latter is computed by using the points within the  $p_q$ 's neighborhood point set that are extracted based on the selected  $k$  value determined by *KNN*, which is described in Section 5.2.

The value of  $k$  that is used for surface normal estimation explained in Section 5.4 affects the results of the  $\alpha_{ri}$  computation. Figure 7.23 shows the results obtained from a dataset of 33 points for varying  $k$  values; the investigated dataset is extracted from the point cloud shown in Figure 7.5(c). The investigated  $k$  values are 3, 6, 8, 16, 24 and 32. Figure 7.23 shows the complete range of computed  $\alpha_{ri}$  values of each point in the 33 point dataset for the listed  $k$  values (details are in Appendix A).

It is important to determine the suitable  $k$  for varying defect types as well as varying point resolutions on the investigated surface. The larger  $k$  values use a larger set of points so the plane fitting smooths the local variations. In this case, it is not possible to capture defects that create only small local changes like cracks. The range of the angles for  $k = 32$  is less than  $1^\circ$ . However, if a small value of  $k$  such as three is used, the variation in the computed  $\alpha_{ri}$  values range from  $5^\circ$  to  $75^\circ$ . It is observed that as the  $k$  value decreases, the sensitivity to local changes increases.

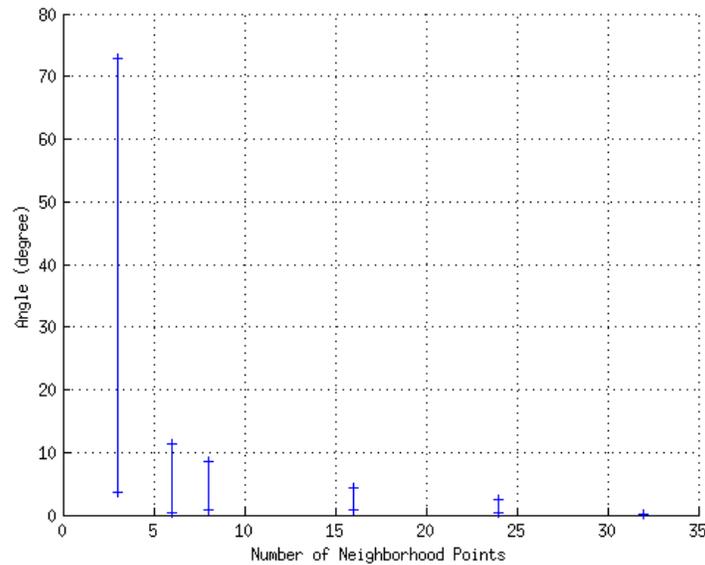


Figure 7.23: Surface normal variations for several nearest neighborhood sizes.

On the other hand, it should be noted that using small  $k$  values to capture small local variations is not always beneficial, especially when the color information is unavailable. When most of the  $\alpha_{ri}$  values are higher than the assigned threshold  $T_\alpha$ , it is not possible to extract defect locations since most of the points are labeled as candidate points for defect locations. This makes it difficult to detect the damaged location only by using the surface normal. An example is shown in Figure 7.19.

In this research, for large defects, an adaptive and automated  $k$  computation method is followed. In order to compute  $k$ , the patches that were extracted for determining the reference normals in the previous section, Section 7.3.3, are used. To these patches, the same sensitivity analysis that was performed in the beginning of this section is applied. For each investigated patch, starting with a small  $k$  value ( $k = 3$ ),  $\alpha_{ri}$  values are computed for varying  $k$  values. The processed  $k$  value is constantly increased at each run until all the computed  $\alpha_{ri}$  values within a neighborhood are less than  $T_\alpha = 10^\circ$  ensuring that the selected neighborhood size is large enough for avoiding excessive local sensitivity. It should be noted that, as discussed earlier, the computed neighborhood sizes vary significantly from point cloud to point cloud. For example, for the concrete testing frame and DeKalb County Bridge, the neighborhood sizes that are used for surface normal-based damage detection are computed as 16 (similar to the statistical outlier method); however, for the Bowker Overpass, which has a very dense surface point distribution, the neighborhood size is determined as 288 (a little less than the  $k$  computed for the statistical outlier removal). For small cracks and for corrosion,  $k/2$  is used assuming color information is available; the main aim of this reduction in the  $k$  value used is to capture the comparatively small local changes and then use the additional color information to extract defect locations..

### 7.3.5 Clustering and Evaluation

The next step after the extraction of candidate defect points  $P_d$  is to divide these points into clusters  $c_d$ . Each cluster consists of a certain number of defective points,  $P_{di}$ . Since the main goal is to localize and quantify the damage, clustering is an important step

for grouping the data into smaller datasets. Once the clustering is completed, it is possible to individually compute the properties of the detected defects such as length, width, area and volume.

For this research, hierarchical clustering that portions data over a variety of scales by creating a cluster tree is used. A cluster thresholding method is then developed in order to exclude the clusters  $c_{di}$  consists of  $P_{di}$  with number of points smaller than a certain threshold. The details of both hierarchical clustering and cluster thresholding are explained in the following sections. Cluster thresholding has to be completed before hierarchical clustering is performed. A cluster evaluation method, which is a silhouette-based method, is also developed in order to optimize the number of clusters  $c_d$  for a given dataset.

#### 7.3.5.1 Cluster Thresholding

The candidate defect points  $P_d$  that are computed by performing normal comparison include some false positives,  $P_{df}$ , which result from either remaining noise or remaining extraneous points on the surface. These points have to be removed prior to clustering in order to avoid possible clustering issues such as having an excessive number of clusters with a small number of points or incorrect clustering.

To address this issue, a cluster thresholding method is developed. This method removes the  $P_{df}$  from the  $P_d$  by using a distance measure. First, the minimum number of points  $P_{cmin}$  that could form a cluster is defined. This value is defined by the user depending on the scope of defect types. The distances between each point pair are then calculated by using the *KNN* method described in Section 5.2. The distances between

each point of  $P_d$  and its neighborhood set defined by the  $P_{cmin}$  are compared with a threshold,  $T_{clus}$ . This threshold needs to be larger than the resolution, and it needs to be small enough to avoid under clustering. Under clustering means combining two close clusters, which are not parts of a single defect, into one large defect cluster. A  $T_{clus}$  value that is 10 times larger than the computed resolution is used in this research. Figure 7.24 shows the distribution of the located damaged regions, before and after cluster thresholding. It can be seen from Figure 7.24(b) that the small clusters, which do not represent defects, are effectively removed from the defect list.

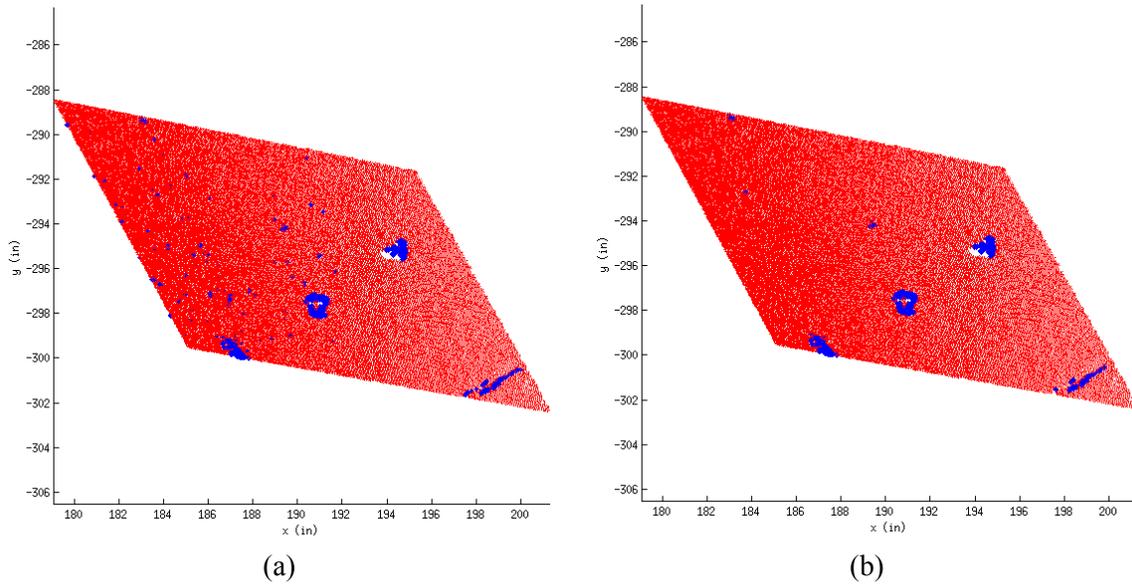


Figure 7.24: (a) Defect locations for the investigated portion of the bottom deck of DeKalb County Bridge before cluster thresholding and (b) after cluster thresholding.

### 7.3.5.2 Hierarchical Clustering

Hierarchical clustering is a method that creates a multilevel hierarchy, where clusters at one level are joined as clusters at the next level. For this research, even though only a single level of clusters are required for large amounts of data, hierarchical

clustering is preferred since it operates on dissimilarity measures instead of actual observations.

The first step of the hierarchical clustering is to find the similarity or dissimilarity between every pair of objects in the dataset. In this step, the distance between objects is calculated. The result of this computation is a distance or dissimilarity matrix. There are several available distance measures. In this research the squared Euclidian distance, which is described in Section 5.2 in detail, is used as the distance measure. It is selected in order to be consistent with the implemented methods given in Chapter 5.

The second step is to group the objects into a binary hierarchical cluster tree. The distance information generated in the first step is used to link pairs of objects that are in close proximity. The clusters formed in the first run are linked to each other or other objects to form bigger clusters. This process is carried out until all the objects in the dataset are linked to each other in the form of a hierarchical tree.

The final step is to cut the hierarchical tree into clusters based on a certain criterion. This criterion is generally the number of clusters  $c_d$ , and usually, it is defined by the user prior to the clustering process. The developed cluster evaluation method eliminates the necessity of defining the  $c_d$  in advance.

The *clusterdata* function implemented in MatLab (2013) performs all the listed steps automatically. The parameters and corresponding values for hierarchical clustering are given in Table 7.12

As mentioned before, the primary constraint of the hierarchical clustering function is that it divides the data into a specific and user defined number of clusters.

This limitation is eliminated by introducing a new cluster evaluation method described in Section 7.3.5.3.

Table 7.12: Parameters and corresponding values for hierarchical clustering.

<b>Method</b>	<b>Parameters</b>	<b>Value</b>
Hierarchical Clustering	Distance measure	Euclidian distance

### 7.3.5.3 Cluster Evaluation

It is not possible to know the number of defects on a surface prior to point cloud processing. Thus, it is essential to adjust the selected clustering method such that the number of clusters  $c_d$  is not a used defined parameter. In order to achieve this, a MatLab (2013) function, which uses the silhouette criterion values to evaluate the optimal number of clusters, is developed. This criterion is chosen since it is suited well for hierarchical clustering solutions with squared Euclidian distances.

The details of the silhouette criterion are discussed in Kaufman and Rousseeuw (2009). The similarity of a point to points in its own cluster, when compared to points in other clusters, is measured by the silhouette value. The silhouette value of  $p_i$  is defined as  $s_i$  and it is computed by using Equation 7.1.

$$s_i = \frac{(b_i - a_i)}{\max(a_i, b_i)} \quad 7.1$$

where  $a_i$  is average distance from  $i$ th point to other points in the same cluster, and  $b_i$  is minimum average distance from  $i$ th point to points in a different cluster.

The range of the silhouette value is from  $-1$  to  $1$ . If the computed silhouette value for  $p_i$  is high, then  $p_i$  is well-matched to its own cluster, and it is poorly-matched to neighboring clusters. The clustering assumed to be efficient if most of the points in each cluster have high silhouette values. Any distance metric can be used in order to

compute the silhouette clustering evaluation criterion; however, for this research, the Euclidian distance is used in order to be consistent with the preceding sections.

First, a cluster list is provided. This list consists of the possible number of clusters that are assumed to range from 1 to 100; the minimum number of clusters to be evaluated is 1 and the maximum number is 100. The largest  $c_d$  value is set to 100 since the point cloud processing is performed in an incremental manner, where small portions of the data are processed first and then, the results are combined. Thus, for a small patch the maximum number of expected clusters is chosen to be 100. The evaluation algorithm starts with calculation of the silhouette values for each point. If the mean  $\mu_s$  of the computed silhouette values within a cluster is smaller than 0.8, that cluster is marked. The total number of marked clusters is subtracted from the total number of clusters used for that run, and the algorithm iterates one more time. The iteration stops when there aren't any marked clusters after the last run. Figure 7.25 represents the optimized clustering results for the defects shown in Figure 7.24, for the bottom deck of the DeKalb County Bridge. The parameters and the corresponding values (or ranges) for cluster evaluation are given in Table 7.13. This study showed that silhouette values can be effectively used for clustering evaluation to ensure that the obtained clusters represent the actual defects.

Table 7.13: Parameters and corresponding values (or ranges) for cluster evaluation.

<b>Method</b>	<b>Parameters</b>	<b>Value (or Range)</b>
Cluster Evaluation	Criterion	Silhouette
	Cluster list	1 to 100

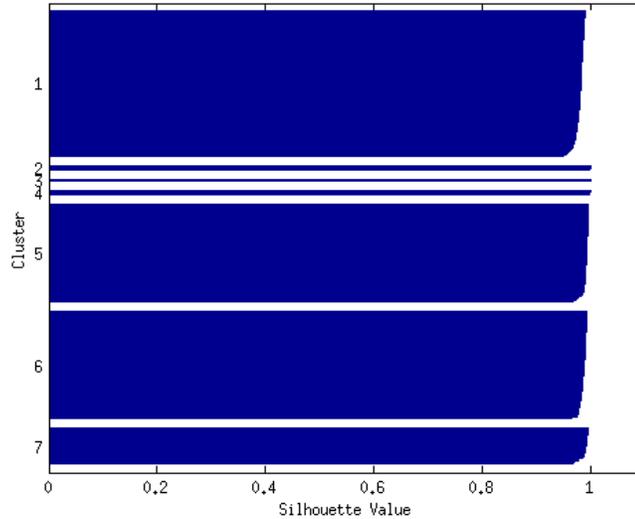


Figure 7.25: Cluster evaluation results for the dataset from the bottom deck of Dekalb County Bridge given in Figure 7.24.

### 7.3.6 Damaged Area and Volume

The next step after clustering is to compute the area and the volume of each cluster, which represents a defect, in order to obtain quantitative information on the detected defects. In this research, three methods are used to calculate the area and volume associated with each surface defect. The first method, which is the using voxel grid representations for volume estimation, was already described in Section 7.2. For this method, each damage cluster is treated as an object, and the volume calculations are completed as mentioned. This volume computation method, which is based on voxelization, is suitable for large defects such as large spalled concrete regions, where there are significant number of points on the surface that represent each defect. However, voxelization method is not suitable for small defects and/or locations with non-homogeneous point distribution, since the defect cannot be represented as a closed volume.

The last two methods, which are based on convex hull and mesh grid respectively, are especially important for the locations with non-homogeneous point distribution, since voxelization cannot be efficiently performed at these locations. The details of both methods are discussed in Sections 7.3.6.1 and 7.3.6.2 respectively.

### 7.3.6.1 Convex Hull for Damage Quantification

Convex hull wraps all the given points in a dataset to create a surface around them. The smallest convex set that contains a set of points is defined as the convex hull in the Euclidian plane or Euclidian space (De Berg et al., 2000). The convex hull can be imagined as a rubber band stretching around a set of given points as shown in Figure 7.26.

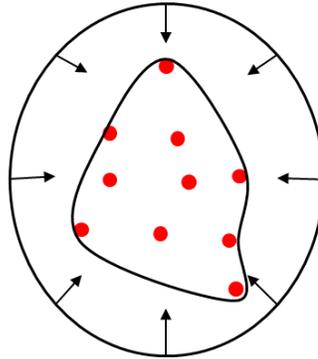


Figure 7.26: Rubber band analogy for convex hull.

The convex combinations of a finite point set  $X$  are defined as the convex hull of these points. Each point  $p_{xi}$  in  $X$  is assigned a weight  $w_i$ , and a weighted average is computed by using these weights.  $w_i$  values are non-negative, and the summation of all weights is equal to 1. The resulting convex combination, for each choice of weights, is a point in the convex hull. The convex hull is given by Equation 7.2.

$$\left\{ \sum_{i=1}^{|X|} w_i p_{xi} \mid (\forall i : w_i \geq 0) \wedge \sum_{i=1}^{|X|} w_i = 1 \right\} \quad 7.2$$

Generally, the defects, which are in the scope of the surface normal-based damage detection method, are associated with significant changes in local properties occur at the defect locations on the surface of structures. Thus, the extracted damage points,  $P_{di}$ , for each defect represent an open 3D shape, which can be directly used neither for area nor for volume calculations.

For area calculations, it is required to project the  $P_{di}$  to the reference plane represented by one of the surface points and the computed reference normal  $\vec{n}_r$ . The point projection is done by using Equation 7.3:

$$p_{dpi} = p_{di} - \text{dot}(p_{di} - p_{ref}, \vec{n}_r) \cdot \vec{n}_r \quad 7.3$$

where  $p_{di}$  is an individual point in  $P_{di}$ , and  $p_{dpi}$  is an individual point in the projected point set  $P_{dpi}$ .

The area  $A_{di}$  of the projected points  $P_{dpi}$  is calculated by using the *convhull* function given in MatLab. It should be noted that the *convhull* function is only suitable for 2D datasets. For volume calculations, another approach is followed. Moreover, the convex hull representation of a given set of points does not represent the exact geometry of the defect. Figure 7.27 represents the differences between the constructed convex hull and the actual shape of a sample of defect points. As can be observed from Figure 7.27, convex hull method is not suitable for defect types that consist of concave regions.



Figure 7.27: (a) Convex hull representation of a sample defect dataset and (b) actual geometric representation of the defect.

The set of points  $P_{dvi}$  that represents the volume of a defect is defined by the sum of the defect points  $P_{di}$  and the projected points  $P_{dpi}$ . This summation provides the set of the points that represents the complete volume of each defect. Figure 7.28(a) shows the defect detection results for a portion of the bottom deck of DeKalb County Bridge and Figure 7.28(b) shows the  $P_{dv1}$  for D1. The volume  $V_{di}$  of the 3D space surrounded by each  $P_{dvi}$  is computed by using the *convhulln* function implemented in MatLab.

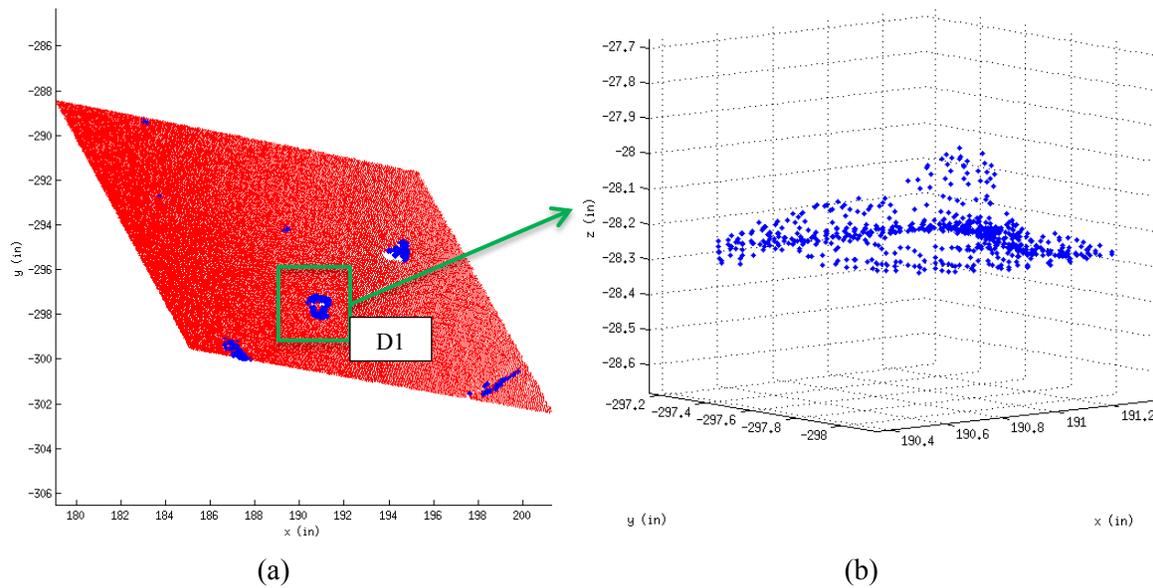


Figure 7.28: (a) Defect detection results for the investigated portion of the bottom deck of DeKalb County Bridge; and (b) 3D representation of the points of defect D1.

The *convhulln* function is a quick hull-based convex hull function. The details of the algorithm within this function are discussed in Barber et al. (1996). This algorithm expands the 2D quick hull algorithms for n-D applications. The quick hull algorithm creates a convex hull around a given set of points by recursively selecting furthest points from a reference location and adding them to the convex hull set one by one. The points that fall into the convex hull are removed from the candidate list of expansion points for the next run. The volume is computed automatically from the tetrahedrons that compose the convex hull. The volume calculation is a part of the implemented algorithm.

Figure 7.29 gives the labels of the detected defects. The 3D representation of the defect points for D1 are shown in Figure 7.30(a), and the *convhulln* result of the same points is given in Figure 7.30(b). The area and volume values computed for each defect is listed in Figure 7.12.

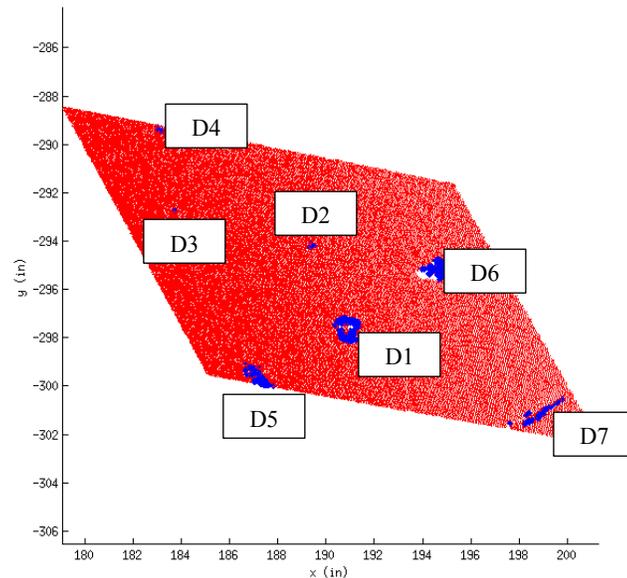


Figure 7.29: Labels of detected damaged locations for investigated portion of the bottom deck of DeKalb County Bridge.

It should be noted that since the volume is calculated by using the convex hull approach, the results might be larger than the exact volume values, depending on the shape of the defect. For example, a portion of D1 consists of points that create a concave surface; this part is marked with a red ellipse in Figure 7.30. Thus, the volume computed via *convhulln* is larger than the exact volume of the defect. In order to address this volume computation problem, another method, which uses a mesh grid approach, is implemented. The details of mesh grid method are described in Section 7.3.6.2.

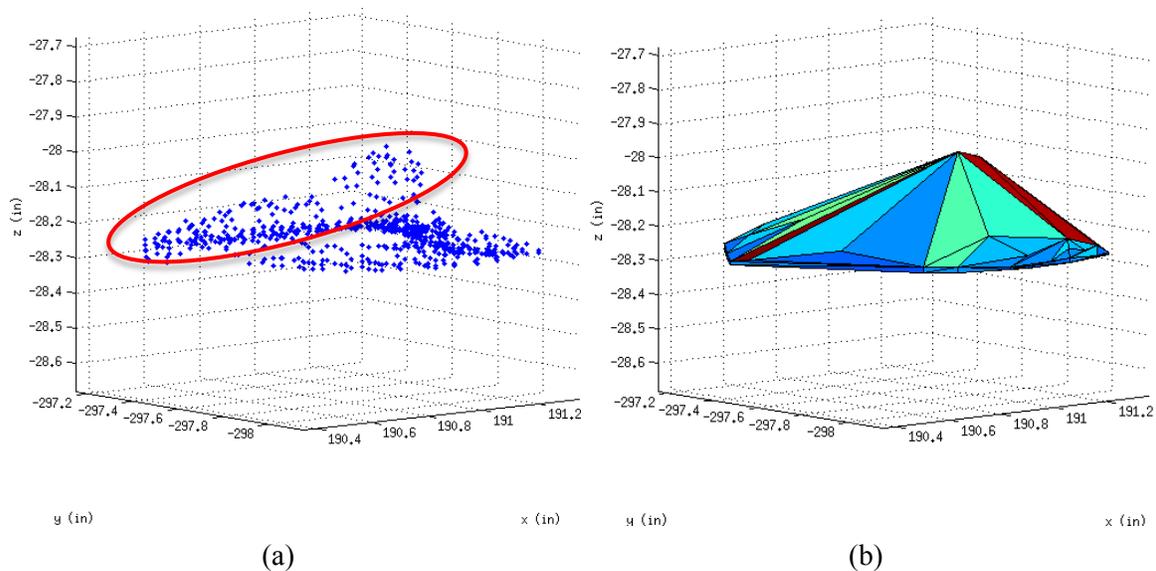


Figure 7.30: (a) 3D representation of the points of defect D1; and (b) *convhulln* result for the points of D1.

Table 7.14: The area and volume results for the convex hull application for the defects shown in Figure 7.29.

<b>Defect Label</b>	<b>Number of points</b>	<b>Area (<math>in^2</math>)</b>	<b>Volume (<math>in^3</math>)</b>
<b>D1</b>	312	1.03	8.5E-02
<b>D2</b>	14	0.01	5.8E-04
<b>D3</b>	6	0.01	1.2E-04
<b>D4</b>	13	0.02	1.9E-03
<b>D5</b>	210	0.66	5.5E-02
<b>D6</b>	233	0.64	5.4E-02
<b>D7</b>	78	0.42	2.8E-02

### 7.3.6.2 Mesh Grid for Damage Quantification

The third method used for damage quantification, following graph-based damage quantification method described in Section 7.2.3 and the convex hull method described in the previous section, is developed in order to address the shortcomings of the first two damage quantification methods discussed in previous sections. The main issue with the first two methods is that they are not effective in handling defects that consist of defect points with varying point densities.

First, the projected points,  $P_{dpi}$ , of the defect  $i$ , which are located on the surface represented by one of the surface points and the computed reference normal  $\vec{n}_r$ , are used to create a 2D mesh grid. Later, the reference surface is matched with the x-y plane. Then, a mesh grid is formed over the defect points, by using the *meshgrid* function given in MatLab. The mesh size is defined by the resolution of the investigated point cluster. The method, which is used to determine the appropriate voxel sizes from point cloud resolutions, is discussed in Section 7.2.1. The input matrix for the *meshgrid* function is in the form of two vectors that have  $x$  and  $y$  coordinate information of the investigated point cluster, and the output is  $xq$  and  $yq$  vectors, where each coordinate couple represents a point on the mesh.

For further processing,  $z$  values of the defect points  $P_{di}$  are stored. A surface is created over the generated mesh grid by using the stored  $z$  values as elevations. A MatLab function, which is called *griddata*, is used to interpolate the scatter defect points. This function fits a surface of the form  $z = f(x, y)$  to the scattered data in the vectors  $x$ ,  $y$  and  $z$ . At each query point specified by  $(xq, yq)$ , the *griddata* function interpolates the surface, and returns the interpolated values,  $zq$ . Even though the fitted

surface is generated based on the mesh grid coordinates represented by  $xq$  and  $yq$ , the surface always passes through the data points defined by  $x$  and  $y$ .

To calculate the volume of each defect, four points that represents the corners of each square located on the generated mesh grid and the corresponding four points on the fitted surface are used. Each eight-point set represents one of the bars that form a defect, and the volume of the whole defect is computed by summing up the volume of all individual bars. The bars with less than  $0.001in$  height are excluded from the computation, since they represent the locations on the mesh grid where a concave feature is formed. Figure 7.31(a) represents the defect points that belong to D1, and Figure 7.31(b) shows the *griddata* result for D1. It can be observed from Figure 7.31 that the mesh grid approach allows capturing the distribution of the points on a fitted surface effectively (the *griddata* results for the rest of the listed defects can be found in Appendix B). This approach prevents miscalculations that are resulted from concave regions on the surface of defects, as it happened with the convex hull approach.

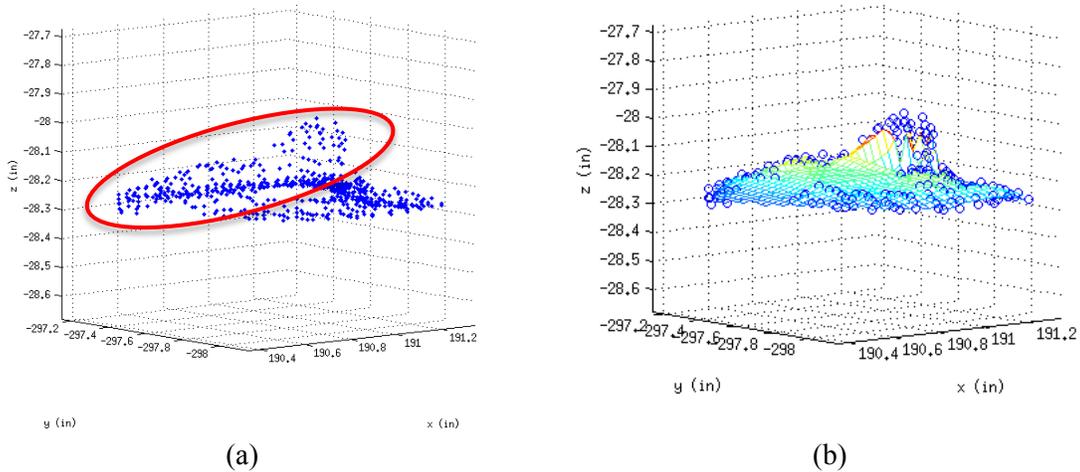


Figure 7.31: (a) 3D representation of the points of defect D1; and (b) *griddata* result for the points of D1.

For the area calculations, only four points that represent the corners of each square of the mesh grid are used. The locations of the excluded bars, which are found during volume computations, are used to eliminate the false positives from the area calculations. Figure 7.32 represents the defect points, which was projected on the reference plane, are shown on the fitted mesh grid. The regions shown with the black circles in Figure 7.32 represents the regions that are excluded from both area and volume calculations.

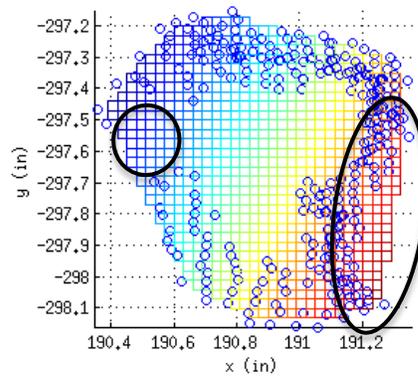


Figure 7.32: Representation of the points of D1 projected on the reference surface for area computation, by using the mesh grid approach.

The results of the area and volume computations for the mesh grid application, for the defects shown in Figure 7.27, are given in Table 7.15. The volume results obtained with mesh grid approach are significantly less than the volume results obtained with convex hull approach. It should be noted that the convex hull approach is only suitable for the defects that form a convex surface; however, for all other damaged area representations, the mesh grid approach is more effective for both area and volume computations.

Table 7.15: Area and volume results for mesh grid application for defects shown in Figure 7.29.

<b>Defect Label</b>	<b>Number of points</b>	<b>Area (<math>in^2</math>)</b>	<b>Volume (<math>in^3</math>)</b>
<b>D1</b>	312	0.74	3.44E-02
<b>D2</b>	14	0.01	2.26E-04
<b>D3</b>	6	0.00	1.09E-04
<b>D4</b>	13	0.02	1.43E-03
<b>D5</b>	210	0.48	2.90E-02
<b>D6</b>	233	0.56	1.56E-02
<b>D7</b>	78	0.31	5.89E-03

Table 7.16: Parameters and corresponding ranges for alignment check.

<b>Method</b>	<b>Parameters</b>	<b>Range</b>
Mesh grid	Voxel grid dimension, $\Delta_x$ - $\Delta_y$	0.03 in. – 0.5 in.

#### 7.4 Parameters and corresponding ranges for investigated damage types

It is required to have a combination of methods in order to detect a specific damage type. Each method is sensitive to certain parameters and therefore, it is necessary to investigate a range of values for each parameter to determine its effect on the overall damage detection accuracy. Table 7.17 lists the methods, associated parameters and ranges for damage types that were given in Table 3.1; the detailed information on each method and its associated parameters could be found in related sections of this chapter, Chapter 7.

In this research, in order to perform damage detection, some of the listed input parameters are directly computed from the investigated point cloud by using the local properties, whereas the others are defined by the user.

Since the main objective is to develop generalized damage detection strategies, the developed algorithms are mostly dependent on parameters computed directly from the point clouds. However, there are still some input parameters that are taken as constant values or defined by the user. It is important to understand the dependency of the obtained damage localization and quantification results on these listed parameters. For this research, a set of detailed validation applications, which are laid out in Chapter 8 and 10, were performed in order to show that the computed and user-defined parameters can be effectively used for locating and quantifying the investigated surface damage types for a variety of point cloud datasets. The results showed that the developed algorithms are efficient in detecting the listed damage types from datasets with varying properties.

Even though the ranges for the input parameters are listed in Table 7.17, for each point cloud dataset, the minimum detectable damage size (and accuracy) for each damage type is dependent on the properties of the investigated point cloud dataset. Thus, the limitations for the developed damage detection strategies can be estimated by using these surface properties. The calculated detectable damage dimensions and expected accuracy are listed in Figure 10.44 and Figure 10.45, after the detailed information on all available point clouds is presented in Section 10.1.

Table 7.17: Parameters and corresponding ranges for investigated damage types.

<b>Damage Type</b>	<b>Method</b>	<b>Parameters</b>	<b>Range (or default value)</b>
<b>Bent members / Points of Discontinuity</b>	Voxelization	Height function increment	0.1 in. – 2 in.
		Voxel grid dimensions	0.03 in. – 0.5 in.
	Alignment check	Height function increment	0.1 in. – 2 in.
<b>Rupture / Spalling</b>	Normal Comparison for Plain Point Clouds	Angle threshold, $T_\alpha$	0°-10°
	Area computation	Direction for connectivity map	Clockwise and counter clockwise
	Volume Integration	Distance increment	0.1 in. – 2 in.
<b>Cracks, Tears, Corrosion</b>	Normal Comparison for Plain Point Clouds	Angle threshold, $T_\alpha$	0°-10°
	Normal Comparison for Texture-mapped Point Clouds	Intensity threshold, $T_{int}$	0-1
	Hierarchical Clustering	Distance measure	Euclidian distance
	Cluster Evaluation	Criterion	Silhouette
		Cluster list	1 to 100
	Convex hull	Quick hull options	-
	Mesh grid	Voxel grid dimension	0.03 in. – 0.5 in.

## **8. Implementation of Neural Network Classifier for Crack Detection**

In the previous chapter, the details of the developed surface-based damage detection algorithm were discussed. Detection examples for both plain and texture-mapped point clouds were then provided within the same chapter. This discussion was continued by laying out the strategies developed specifically for quantifying the detected surface damage. The damage quantification methods were used to compute the area and volume associated with each detected defect. However, these quantification methods were not suitable for recording required crack dimensions, which can be listed as length and thickness. Thus, this current chapter starts with discussing a methodology developed for computing these necessary crack dimensions automatically.

The first step towards automated crack dimension extraction is to adjust the existing clustering algorithm given in Section 7.3.5 for cracks, and then quantify the dimensions of each separated point cluster with the proposed crack dimension extraction method, which is discussed in Section 8.2. The details of the cluster tuning for crack detection and crack dimension extraction are discussed in Sections 8.1 and 8.2 respectively. The effectiveness of the developed method is then tested on several patches extracted from the surface of the concrete testing frame, and the results are displayed in Section 8.3. The introductory information on this dataset is given in Section 4.1, and the detailed information on the same frame is presented in Section 10.1.3.

In Section 8.4, this time, the developed damage detection methods are used for locating and quantifying concrete spalling. Several patches, which are extracted from the surface of the concrete testing frame, are processed. A set of representative dimensions are extracted from the detected damage areas, and these dimensions are compared with

the corresponding hand-measurements, in order to validate the efficiency of the detection algorithms.

The results obtained through defect detection and dimension extraction algorithms prove that these developed methods provide an opportunity to use laser scanning technology for detecting small defects, such as cracks and spalling regions, effectively. However, it should be noted that since the defect detection is achieved through investigating the local variations at a point of interest  $p_q$  on the surface, the developed detection algorithms are sensitive to surface impurities, which cause the normal computed for the investigated  $p_q$  to deviate from the computed reference normal. The details of the surface patch investigation for computing reference normals are discussed in Section 7.3.1. In Chapter 7, the reference normal computation step is followed by a discussion on the normal comparison strategies used for selecting candidate damage locations in; these comparison strategies are discussed in Sections 7.3.2 and 7.3.3 in detail. The impurities mentioned above, and/or any reoccurring patterns, result in false positives. Thus, the results obtained using the developed defect detection algorithms include several falsely detected clusters, which do not represent an actual defect. In order to avoid detecting these false positives and to improve the accuracy of the developed algorithms, a neural network classifier is introduced. The details of this classification strategy and the classifier's overall performance on a real dataset, the concrete testing frame, are discussed in Section 8.5.

## 8.1 Clustering Tuning for Small Defects

As mentioned in Section 7.3.5, the step that follows the extraction of candidate defect points  $P_d$ , which are detected by using the surface-based damage detection method described in Section 7.3, is to divide these points into clusters  $c_d$ .

For clustering small defects, the same methodology described in Section 7.3.5 is followed. First, the clustering thresholding is applied, and then the hierarchical clustering is performed. However, it was observed from the initial results that the proposed clustering algorithm combines closely located defect clusters into a single, large cluster. Thus, a new step that prevents under-clustering is added to the overall clustering process, since the obtained defect detection results are significantly affected by the clustering accuracy.

This additional step follows the same logic behind the radius-based outlier removal that is described in Section 5.3.1. First, a distance map for each point in a cluster  $c_{di}$ , which consists of  $P_{di}$  points, is extracted. This results in a  $P_{di}$  by  $P_{di}$  matrix; each row of this matrix includes the distances  $d_{ni}$  of each point  $p_q \in P_{di}$  to every other point in the cluster. Then, a search radius  $r$  is defined by the user. This  $r$  is used to create a virtual sphere around the point of interest.  $d_{ni}$  values for each point, an entire row, is checked to ensure that there is at least one point that falls into the sphere defined by search radius  $r$ . If so, the point is kept in the cluster; otherwise, it is removed. The removed point then becomes a free agent for the succeeding clustering iteration. This iteration process is continued until the radius criterion is satisfied by all the points in the resulting clusters. In this research,  $r$  value is set to the crack thickness limitation given for medium-severe case, 0.125 in, for reinforced concrete decks or slabs in the National

Bridge Inventory (FHWA, July 2014). The assumption is that any point, belonging to a small defect, will have at least one neighbor point within the radius defined by this length.

Figure 8.1 represents the clustering results obtained for BC-1-1, a patch taken from concrete testing frame, by using both of the discussed clustering methods: the initial (Section 7.3.5) and the updated clustering (the current section) methods. Figure 8.1(a) shows the result of the clustering method given in Section 7.3.5; one of the detected defects, marked with a black rectangle, is not clustered properly. Subsequently, the improved clustering algorithm, discussed in this section, is applied to the same defect dataset; Figure 8.1(b) displays the results. The image of the patch used for this example, BC-1-1, is given in Figure 7.18. The beam and column legends for the concrete testing frame, which is shown in Figure 7.4, are given in Figure 7.9. For the concrete testing frame, further information on the defect locations and sizes are included in Appendix C.

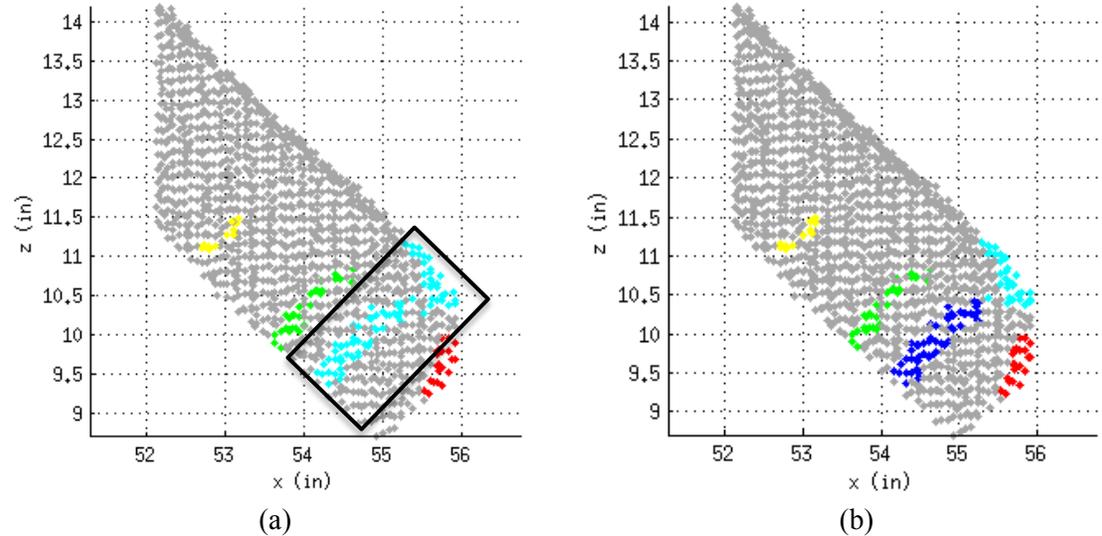


Figure 8.1: Results for both (a) initial and (b) improved clustering methods for BC-1-1.

## 8.2 Crack Dimension Extraction

The step that follows the updated damage clustering method is to extract the dimensions of each detected defect; for concrete, these defects can be listed as small spalled regions, cracks, and etc. In this section, a simple bounding box approach, which is used for computing the required dimensions of each defect, is discussed. Points of each defect cluster  $P_{di}$  are first projected to the computed reference plane. The reference plane computation for surface-based damage detection is already discussed in Section 7.3.1, and the process of projecting  $P_{di}$  to this reference plane for defect quantifications is discussed in Section 7.3.6.1.

The general aim of the bounding box approach is to surround all the projected points,  $p_{dpi}$ , with a rectangle. This process starts with fitting a rectangle to  $p_{dpi}$  by using the difference between the minimum and maximum values of  $x$  and  $y$  axes. Later, this fitted rectangle is rotated until an optimum orientation, which minimizes the area value bounded within the resulting rectangle  $bb_i$ , is obtained. Figure 8.2 shows a representation of bounding box approach on the same patch, BC-1-1, used for Figure 8.1.

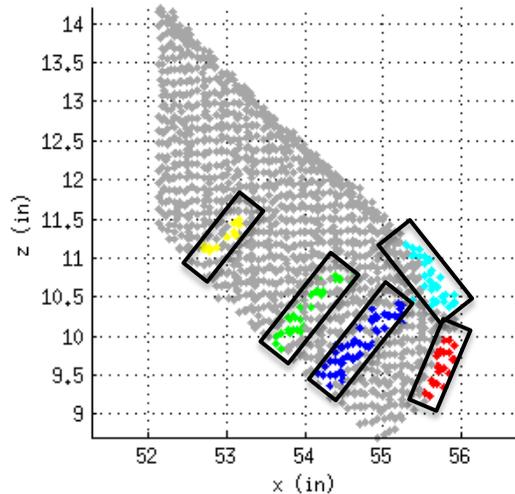


Figure 8.2: Bounding box representation for BC-1-1.

The dimensions of  $bb_i$  are then used for both understanding the characteristics of the defects and for extracting the required dimensions of each investigated defect. As mentioned at the beginning of this section, the results obtained using the developed surface damage detection method may include several defect types, such as small spalled regions and cracks. Once  $bb_i$  is fitted around  $p_{dpi}$ , the ratio of length  $l_i$  to width  $w_i$  is checked. For this research, the limiting ratio  $l_i/w_i$  *min* is taken as 2, and the limiting maximum crack thickness  $t_{max}$  is taken as twice the crack thickness limitation value given for the medium-severe case, 0.125 in, for reinforced concrete decks or slabs in the National Bridge Inventory (FHWA, July 2014), which is 0.25 in. Thus, any defect that has a  $l_i/w_i$  less than 2 and/or an average computed thickness  $t_i$  larger than 0.25 in is labelled as a spalling region. The defect area and volume are computed for these labelled defects instead of the length and thickness values. The defect area and volume are computed using the quantification methods described in Section 7.3.6.

The length of the fitted bounding box is directly recorded as the crack length, since the original measurements taken from the concrete testing frame were recorded by using a similar approach. For example, for a curved crack, only the distance between the start and end points of the investigated crack was measured. However, for the crack thickness computation, a different approach is followed.  $t_{max}$  is used as a height increment along the length of the bounding box, meaning that each  $bb_i$  is divided into several smaller rectangles  $bb_{ij}$ . Linear regression is performed to the points that fall into each  $bb_{ij}$ . In order to perform this linear fitting, the *polyfit*, which is a function implemented in MatLab (2013), is used. The residual values are then computed as a vector signed numbers. The residual pairs, one from each side of the fitted first degree

polynomial, are used to compute an average crack thickness  $t_{ij}$  for each  $bb_{ij}$ . The largest  $t_{ij}$  is then recorded as the computed crack thickness. The general process of crack thickness computation is shown in Figure 8.3. In Figure 8.3, the black box represents  $bb_i$ , and each green box corresponds to a single  $bb_{ij}$ . The points within each green box are used for a first-order polynomial fit; the fitting results are shown with blue lines. The residuals pairs within each  $bb_{ij}$  are then used to compute the average thickness  $t_{ij}$ .

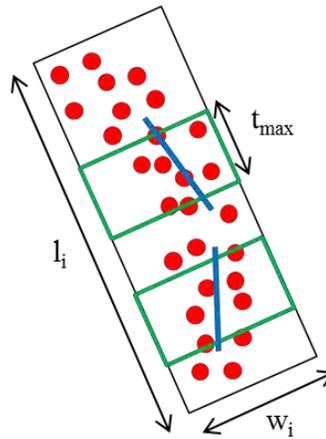


Figure 8.3: Presentation of a sample bounding box, and the approach for crack thickness computation.

### **8.3 Crack Dimension Validation**

The developed crack dimension extraction method is validated against 21 cracks, which are extracted from the concrete testing frame. The results of the validation study are displayed in Table 8.1. The developed defect detection algorithms determined the crack locations for all selected 21 cases accurately.

Furthermore, these obtained results showed that the percentage error for the computed length values change between 0 to 10%, whereas the error values associated with thickness measurements vary between 5 and 34%. It is observed that as the crack thickness reduces, especially for cases where the crack thickness is less than 0.1 inches, the computation error increases significantly. However, it should be noted that these error values are mostly related to the quality of the collected scan data rather than the defect detection method. It is already mentioned in previous chapters that the detection parameters are automatically selected by using the local properties of the investigated surfaces; thus, the effectiveness of the detection algorithms is directly proportional to the quality of the laser scan data itself. For this particular dataset, the collected point cloud was not homogeneous over the entire surface of the testing frame, and there were minor registration issues.

Table 8.1: Results of crack validation study.

Crack Number	Label	Crack Label	Measured Dimensions		Computed Dimensions		Error (%)	
			Thickness (in)	Length (in)	Thickness (in)	Length (in)	Thickness	Length
1	A1-1	A1-1-C1	0.10	1.78	0.10	1.87	6.5	5.1
2	A1-4	A1-4-C4	0.12	1.05	0.13	1.13	6.1	6.9
3	B1-1	B1-1-C4	0.14	1.56	0.16	1.59	15.8	2.5
4	B1-4	B1-4-C3	0.25	1.39	0.26	1.31	5.1	-5.6
5		B1-4-C4	0.07	1.64	0.09	1.75	24.8	6.5
6	B2-1	B2-1-C3	0.12	1.79	0.15	1.94	19.8	8.4
7	B2-4	B2-4-C1	0.05	2.03	0.07	1.98	32.8	-2.5
8		B2-4-C2	0.13	1.87	0.16	1.99	19.5	6.2
9	B2-5R	B2-5R-C2	0.14	2.24	0.14	2.34	6.5	4.3
10		B2-5R-C4	0.04	2.17	0.05	2.38	22.6	9.8
11	C1-1	C1-1-C3	0.19	1.94	0.21	1.85	11.4	-4.7
12		C1-1-C4	0.12	1.90	0.17	1.96	34.7	2.9
13	C1-4	C1-4-C1	0.11	1.36	0.12	1.43	14.9	5.3
14		C1-4-C2	0.23	1.58	0.24	1.64	4.1	3.4
15		C1-4-C3	0.22	1.78	0.23	1.87	6.6	4.7
16		C1-4-C4	0.10	1.69	0.12	1.80	26.1	6.3
17	C2-1	C2-1-C1	0.14	1.77	0.17	1.87	16.3	5.7
18		C2-1-C2	0.09	1.65	0.11	1.60	24.7	-3.2
19		C2-1-C3	0.12	1.54	0.13	1.64	14.9	6.2
20	C2-4	C2-4-C2	0.09	1.87	0.12	1.98	28.7	5.6
21		C2-4-C3	0.11	2.26	0.13	2.17	18.9	-3.7
<b>Mean</b>							<b>16.3</b>	<b>5.1</b>
<b>Standard Deviation</b>							<b>9.1</b>	<b>4.4</b>

## 8.4 Spalling Dimension Validation

In this section, similar to the crack dimension validation, the dimensions of the 20 detected spalled concrete regions are validated against the measurements collected from the investigated locations of the concrete testing frame. First, the recorded measurements are displayed in Table 8.2. The spalled region dimensions are measured by using the following reference measurement drawings for two types of spalling: triangular and rectangular spalling, which are shown in Figure 8.4. For spillings, the measurements for each number pair, where each number indicates a corner, are recorded. The images of the spalled concrete regions for the concrete testing frame are shown in Appendix C. Then, the computed defect dimensions are given in Table 8.3 and finally, in Table 8.4, the comparison between the measured and computed dimensions is given in terms of percentage error. The developed defect detection algorithms determined the spalling locations for all selected 20 cases accurately.



Figure 8.4: Reference measurement drawings for spalled concrete regions.

Moreover, these obtained results showed that the mean error for the validation set is 2.5%. Even though this mean error value is small, Table 8.4 shows that there are significant variations in individual error values; these values vary between 0.2 and 32%. However, when the results are investigated in depth, it is found that the location for which the large error values are obtained coincide with the incomplete regions of the

point cloud. Due to the laser scanner's height, the bottom planes of several beams were not captured. As a result, the computed damage dimensions, obtained from these locations, differ from the recorded measurements significantly. These findings support the conclusion made in the previous section, which states that the error values are mostly related to the quality of the collected scan data rather than the defect detection method.

Table 8.2: Measured dimensions for concrete spalling validation.

Spalling Number	Label	Spalling Label	Measured Dimensions			
			Spalling			
			1-2 (in)	2-3 (in)	3-1(in)	
			1-2 (in)	2-3 (in)	3-4 (in)	4-1 (in)
1	<b>BC1-1</b>	BC1-1-S1	2.65	1.65	2.26	2.63
2	<b>BC1-4</b>	BC1-4-S1	2.23	2.35	2.73	1.85
3	<b>BC1-5</b>	BC1-5-S1	2.64	2.18	1.80	2.08
4	<b>BC2-1</b>	BC2-1-S1	2.54	0.95	2.33	1.83
5	<b>BC2-4</b>	BC2-4-S1	2.15	2.44	1.37	
6	<b>BC2-5</b>	BC2-5-S1	1.83	2.17	1.36	2.10
7	<b>C1-1</b>	C1-1-S1	1.05	1.98	2.09	
8		C1-1-S2	1.81	1.11	1.90	
9	<b>C1-4</b>	C1-4-S1	1.79	1.27	1.68	
10		C1-4-S2	1.84	0.98	1.94	
11	<b>C2-1</b>	C2-1-S1	2.29	1.23	1.80	
12		C2-1-S2	2.08	1.43	1.89	
13	<b>C2-4</b>	C2-4-S1	1.35	1.94	2.03	
14		C2-4-S2	1.23	2.03	1.96	
15	<b>CD1-1</b>	CD1-1-S1	2.19	3.17	3.45	2.45
16	<b>CD1-4</b>	CD1-4-S1	3.13	2.43	2.14	2.06
17	<b>CD1-5</b>	CD1-5-S1	2.05	2.13	2.45	2.19
18	<b>CD2-1</b>	CD2-1-S1	2.16	1.69	2.75	1.68
19	<b>CD2-4</b>	CD2-4-S1	2.39	2.01	2.24	
20	<b>CD2-5</b>	CD2-5-S1	2.24	2.09	1.68	2.10

Table 8.3: Computed dimensions for concrete spalling validation.

Spalling Number	Label	Spalling Label	Computed Dimensions			
			Spalling			
			1-2 (in)	2-3 (in)	3-1(in)	
			1-2 (in)	2-3 (in)	3-4 (in)	4-1 (in)
1	<b>BC1-1</b>	BC1-1-S1	2.64	1.60	2.21	2.11
2	<b>BC1-4</b>	BC1-4-S1	2.21	2.11	2.76	1.84
3	<b>BC1-5</b>	BC1-5-S1	2.52	1.98	1.81	2.12
4	<b>BC2-1</b>	BC2-1-S1	1.78	0.65	1.96	1.84
5	<b>BC2-4</b>	BC2-4-S1	2.18	2.35	1.36	
6	<b>BC2-5</b>	BC2-5-S1	1.96	2.02	1.45	2.11
7	<b>C1-1</b>	C1-1-S1	1.12	1.88	2.11	
8		C1-1-S2	1.84	1.24	1.94	
9	<b>C1-4</b>	C1-4-S1	1.81	1.22	1.69	
10		C1-4-S2	1.81	0.94	1.98	
11	<b>C2-1</b>	C2-1-S1	2.32	1.12	1.91	
12		C2-1-S2	1.89	1.42	1.88	
13	<b>C2-4</b>	C2-4-S1	1.32	1.98	2.03	
14		C2-4-S2	1.26	2.21	1.92	
15	<b>CD1-1</b>	CD1-1-S1	2.12	2.95	3.85	2.42
16	<b>CD1-4</b>	CD1-4-S1	3.26	2.46	2.29	2.12
17	<b>CD1-5</b>	CD1-5-S1	2.10	2.19	2.32	2.18
18	<b>CD2-1</b>	CD2-1-S1	2.15	1.76	2.78	1.69
19	<b>CD2-4</b>	CD2-4-S1	2.39	2.24	2.28	
20	<b>CD2-5</b>	CD2-5-S1	2.23	1.95	1.58	2.20

Table 8.4: Results of concrete spalling validation.

Spalling Number	Label	Spalling Label	Error (%)			
			1-2 (in)	2-3 (in)	3-1(in)	
			1-2 (in)	2-3 (in)	3-4 (in)	4-1 (in)
1	<b>BC1-1</b>	BC1-1-S1	0.2	2.9	2.3	19.8
2	<b>BC1-4</b>	BC1-4-S1	0.9	10.3	1.2	0.6
3	<b>BC1-5</b>	BC1-5-S1	4.4	9.3	0.6	1.9
4	<b>BC2-1</b>	BC2-1-S1	30.0	31.6	15.8	0.5
5	<b>BC2-4</b>	BC2-4-S1	1.5	3.6	0.5	
6	<b>BC2-5</b>	BC2-5-S1	7.3	6.7	6.8	0.5
7	<b>C1-1</b>	C1-1-S1	6.2	5.1	0.8	
8		C1-1-S2	1.8	12.1	2.1	
9	<b>C1-4</b>	C1-4-S1	1.0	3.8	0.9	
10		C1-4-S2	1.8	3.8	2.0	
11	<b>C2-1</b>	C2-1-S1	1.5	8.6	5.8	
12		C2-1-S2	9.2	0.7	0.7	
13	<b>C2-4</b>	C2-4-S1	1.9	2.2	0.0	
14		C2-4-S2	2.6	8.8	1.8	
15	<b>CD1-1</b>	CD1-1-S1	3.3	6.9	11.5	1.3
16	<b>CD1-4</b>	CD1-4-S1	4.3	1.4	7.0	3.1
17	<b>CD1-5</b>	CD1-5-S1	2.2	2.9	5.4	0.5
18	<b>CD2-1</b>	CD2-1-S1	0.6	4.2	1.2	0.9
19	<b>CD2-4</b>	CD2-4-S1	0.2	11.3	1.6	
20	<b>CD2-5</b>	CD2-5-S1	0.6	6.5	5.7	4.8
					<b>Mean</b>	<b>Standard Deviation</b>
					<b>2.5</b>	<b>5.9</b>

## **8.5 Classification through Artificial Neural Networks**

The developed damage detection algorithms provide an adaptive system for damage detection, since the parameters used for the detection algorithms are computed automatically using the associated surface properties. In most of the previous crack detection studies, which are predominantly image-based, some of the important parameters, such as camera-object distance, are not considered or are assumed to be constant. This prevents most of the current approaches from being used for crack quantification, since these methods are specifically developed for crack detection rather than quantification. For current approaches, it is required to maintain a constant focal length, resolution, or distance to the object in order to be able to extract crack dimensions (Kaseko and Ritchie, 1993; Cheng et al., 1999; Abdel-Qader et al., 2003; Abdel-Qader et al., 2006; Choudhary and Dey, 2012; Rabah et al., 2013; Adhikari et al., 2014; Laefer et al., 2014).

Laser scanning technology provides a 3D representation of the entire structure under investigation. As has been discussed, with the current developments in this technology, it is possible to collect high-density point clouds that capture the surface properties accurately for structures. The damage detection method proposed for this research, which is presented in the previous sections, eliminates the requirement for prior knowledge on focal length, resolution, or distance to the investigated object, since all the required parameters for the defect detection are extracted from the point cloud automatically. However, it should be noted that the proposed method, similar to many other crack detection methods, results in false positives along with actual defects. These false positives could be a result of a repeated pattern, any surface impurity, and etc.; basically,

any surface variation that causes the normal computed for the investigated  $p_q$  to deviate from the computed reference normal and/or results in significant variations in local intensity values. Two example cases that are commonly encountered for this dataset are shown in Figure 8.5. In Figure 8.5(a), the locations of small surface holes are shown with black circles, and a repetitive pattern for intensity variation, resulted from inefficient texture-mapping, is represented in Figure 8.5(b). It is required to eliminate the false positives from the detected damage clusters in order to improve the overall efficiency of the proposed defect detection algorithm. To perform this, a trained artificial neural network classifier is used to differentiate the real defects (cracks and/or small spalling regions) from false positives.

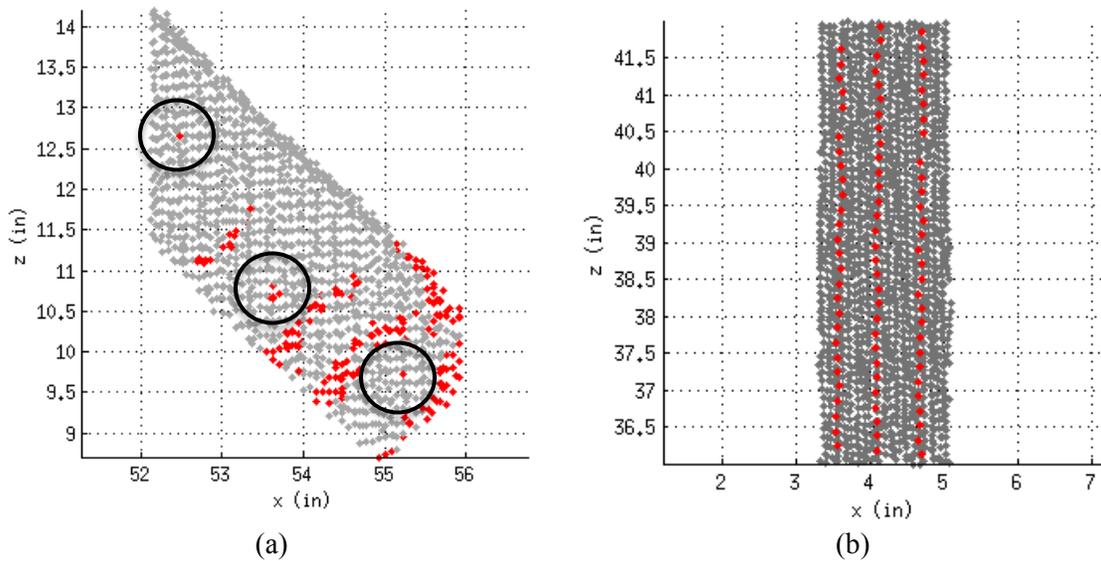


Figure 8.5: Sample false positive patterns: (a) small holes existing on the surface and (b) lines with significant intensity variation.

Jahanshahi and Masri (2012) performed an image-based study, which uses 2D images to create 3D surface representations, in order to eliminate the dependency on the previously listed parameters: constant focal length, resolution, or distance to the object.

The first step of this study is segmentation, which is used for isolating the patterns that can be potentially classified as defects. The useful information about the scene objects is extracted by using morphological image processing and then, structuring elements are used to complete the segmentation. Segmentation is followed by feature extraction and finally, the study is completed by classification.

In our research, a similar methodology for classification is followed. First, the defects are segmented as objects by using the developed clustering methods; the details of the developed clustering algorithms are described in the previous sections. Second, a feature set that stores quantitative information on each defect cluster is formed. Finally, this feature set is used to train, validate and test the neural network classifier.

#### 8.5.1 Feature Extraction

A feature is defined as a set of finite values that represents the quantitative attributes or properties of any segmented object; in our case, clusters. It is crucial to generate an appropriate feature set that includes all the important characteristics that helps identifying similar patterns. In this research, a feature set , which is similar to the one proposed in Jahanshahi and Masri (2012), is used, with two additions representing the properties of intensity distribution, since both application have a similar nature. The feature set defined in Jahanshahi and Masri (2012) is extracted by using only the 2D properties of the segmented objects; however, since the detected defect clusters represent a 3D dataset, for this research, some the described features are extracted by using 3D information.

The features that are included in the feature set can be listed as eccentricity, area of the projected cluster divided by the area of the bounding ellipse, solidity, absolute value of the correlation coefficient, compactness, and mean and standard deviation of intensity values.

The first feature, eccentricity, is defined as the eccentricity of the ellipse that has the same second-moments as the projected cluster points. For clusters, first, the defect points are projected on the reference plane and then, the eccentricity is computed. The process of projecting cluster points to a selected reference plane for defect quantifications is discussed in Section 7.3.6.1. The second feature is the area of the projected cluster divided by the area of the bounding ellipse. The third feature is called solidity; this feature represents the proportion of the points in a 3D convex hull that also belong to the defect cluster. The details for the convex hulls are described in Section 7.3.6.1. The fourth feature is the absolute value of the correlation coefficient associated with the projected defect points. The fifth feature is the compactness; this feature is defined as the ratio between the square root of the computed area and its perimeter. Finally, the sixth and seventh features are the mean and standard deviation of the intensity distribution over the investigated patch.

The features listed above are computed for each defect cluster, and the values are stored in a feature matrix. This feature matrix is then used for training, validating and testing the neural network classifier.

### 8.5.2 Classification

There are several classifier options that can be used for this application. Some of the possible classifiers can be listed as Bayes classifier (Rish, 2001), k-nearest neighborhood classifier (Keller et al., 1985), multilayer feed-forward artificial neural networks (Hornik et al., 1989), support vector machines (Hearst et al., 1998), and etc. However, in most of the recent studies performed for crack detection, where several classifiers are compared, the accuracy of the results obtained by using neural network classifiers is shown to be higher than the other listed methods (Kaseko and Ritchie, 1993; Cheng et al., 1999; Abdel-Qader et al., 2006; Choudhary and Dey, 2012; Jahanshahi and Masri, 2012; Adhikari et al., 2014). Thus, for this research, the neural network classifier is selected to be used for classification.

An artificial neural network, which is composed of processing elements that are interconnected via synaptic or weighted connections, is a parallel processing optimization system. In a neural network, weighted interconnections are used to process inputs received at a processing element in a layer, and then the resulting outputs are transmitted to the following set of processing elements. For nonparametric pattern classification, feed-forward neural networks, which are trained by using a back-propagation algorithm, are the most commonly used neural approaches (Rumelhart et al., 1995). This type of neural network can be composed of three or more layers of processing elements: the input layer, hidden layer/layers, and output layer. Through weighted interconnections, each processing element in a layer is connected with all processing elements in the preceding and following layers. For pattern classification applications, the input layer consists of a set of feature vectors; thus, the size of the input layer is always equal to the number of

features. However, it should be noted that the number of processing units in hidden layers is completely dependent on the complexity of the pattern recognition problem. The number of neurons in the hidden layer is generally empirically determined by trial and error (Kaseko et al., 1994). The number of layers and the number of neurons in the hidden layers of neural networks have to be chosen carefully in order to obtain good classification results. Since the computed feature matrix is composed of similar entities, in this research, the neural network configuration given in Jahanshahi and Masri (2012) is used. The classifier used for this application is a three-layer feed forward neural network, which has 2 output neurons and 10 neurons in the hidden layer.

The developed damage detection algorithms are used to process 106 randomly selected surface patches from the concrete testing frame; these processed patches are extracted from the concrete testing frame shown in Figure 7.4. As a result, 201 candidate defect clusters are detected and separated for further analysis. These defect clusters are plotted, and the actual defects and the false positives are manually separated for every patch.

The generated feature set is composed of 201 damage and non-damage feature vectors. Out of 201 feature vectors, 74 vectors represent the properties of actual defects, whereas 127 vectors are composed of features extracted from false positives. Appendix D includes additional information on the patches used for this application. For training, 70% of the complete feature set is used; 15% is used for validation and finally, 15% is used for testing.

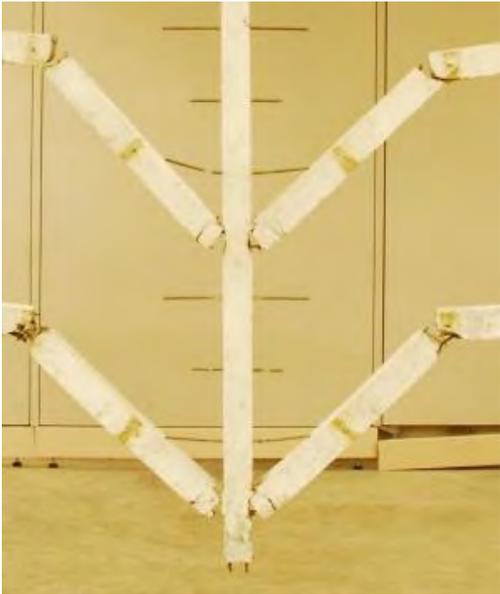
The performance of the selected classifier is shown by using four items: accuracy, precision, sensitivity, and specificity. Accuracy shows the proportion of true

classifications in the test set (15% of the entire feature set). Precision is defined as the proportion of the true positive classifications against all positive classifications. Sensitivity is the proportion of actual positives that were correctly classified, and specificity is the proportion of negatives that were correctly classified. The results are shown in Table 8.5. Figure 8.6(a) shows a portion of the concrete testing frame, and Figure 8.6(b) presents the post-classification damage detection results on the corresponding region of the point cloud.

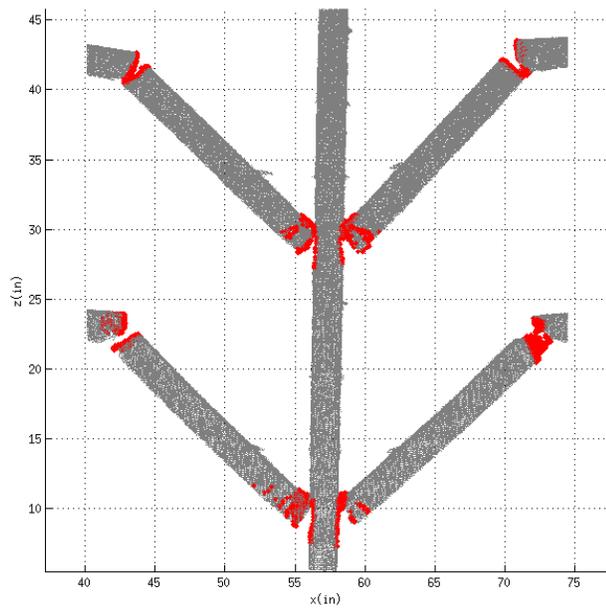
Table 8.5: Performance results for the neural network classifier.

	Percentage (%)
<b>Accuracy</b>	93.51
<b>Precision</b>	94.45
<b>Sensitivity</b>	95.89
<b>Specificity</b>	90.38

The results show that the accuracy of the neural network classification is significantly high for this specific application. However, it should be noted that the detected defect clusters sometimes fail to represent all the surface damage that is present on an investigated patch. Some of the defects may not be detected depending on the properties of the point cloud. For these cases, even though the accuracy of the classifier is high for the generated feature set, the obtained results may not represent the overall efficiency of the developed damage detection algorithm. At the same time, it should be mentioned that the performance of the trained neural network on another structure cannot be estimated from the obtained results; however, in literature, there are several examples that show the accuracy of a trained neural network reduces when the classifier is tested on different structures.



(a)



(b)

Figure 8.6: (a) Image of a portion of the concrete testing frame, and (b) defect detection results shown on corresponding portion of the point cloud.

## **9. Combining Current Visual Bridge Inspection Strategies with Camera Integrated Laser-based Surface Damage Detection**

Most of the bridge surface defects, such as cracks, rupture, spalling, etc. that can be detected by laser scanners are visible. The common practice is to document these damaged locations by images. However, as mentioned in Chapter 1, it is not very efficient to use digital images for routine inspection even though this technology is well-integrated in the current visual inspection strategies. Digital images, most of the time, do not provide quantitative information, and it is hard to retake images from the same location with the same camera orientation unless the cameras are attached to a certain location on the bridges to work as leave in sensors. The use of imaging is also very subjective; the image locations solely decided on by the inspector in charge. Thus, for the same location, the documentation varies significantly depending on the inspector. This causes reporting issues, and most of the time, the recorded condition rating for a certain location varies.

Enhancing routine inspection strategies is especially important nowadays, since the total number of structurally deficient or functionally obsolete bridges is more than 20% of all existing bridges. These structures have to be monitored carefully and the necessary precautions have to be taken. Routine inspection is a major part of the general inspection process; thus, improving the current visual inspection strategies would help reduce the money spent. The current condition of State bridges is discussed in Section 9.1.

The main problem with the routine inspections is that it is very hard to record quantitative information from visually observed damaged locations. This is even harder when the bridge components are inaccessible. Traffic is interrupted, and special

equipment is used to perform the routine inspection in these cases. This process is dangerous for the inspector, and it is also subjective and very expensive. Laser scanners can capture all the surface information, in high resolution, from all laser scanner accessible locations. Since the visual information is organized in a scan, it is possible to document relative positions of defect areas. However, this is hard to achieve when only the local information obtained through digital imaging is used.

The issue with the laser scanning technology, on the other hand, is that there is not an easy way to process the captured datasets. Depending on the resolution and the number of scans included in one dataset, the size of the finalized dataset is too large for regular systems to process. Besides, the dataset represents the whole structure, not the individual elements. Thus, it is especially important to understand how to reduce the collected dataset into smaller units that are easy to process. The steps for data processing are discussed in previous chapters in detail.

Once the point cloud is divided into point clusters, which are processed individually for damage detection, it is important to tie the results to commonly accepted condition rating criteria. This provides an opportunity to document quantitative information on damaged areas in a widely-accepted format. Thus, laser scanners can be used for aiding the visual inspections, and they will enhance the quality of the collected information. Section 9.2 displays the condition rating criteria used in this research.

Finally, even though it is not the focus of this research, it is important to understand how enhancing current inspection strategies would improve load rating computations for individual bridge elements, since the condition rating is a part of the general load rating equation. Load ratings are used to ensure bridge safety, check if the investigated bridges

comply with Federal regulations, determine rehabilitation or replacement needs, and decide on posting needs. Section 9.2 explains how the condition ratings affect the load rating computations, and how laser scans can be used for retrieving required geometric information.

### 9.1 Current Condition of State Bridges

A significant percentage of the nation’s bridges are either functionally obsolete, meaning those bridges were built using outdated standards, or structurally deficient, where one or more of their major components have deteriorated, and they are not safe. Even though this percentage has decreased over the last decade due to the increasing efforts of States, still 24.9% of the nation’s bridges are in either deficiency category. Figure 9.1 represents the state bridge condition ratings from 2013; the color of each is adjusted based on the percentage of structurally deficient bridges. The changes in the states’ structurally deficient bridge inspection ratings between 2007 and 2013 are given in Table 9.1 and continued in Table 9.2.

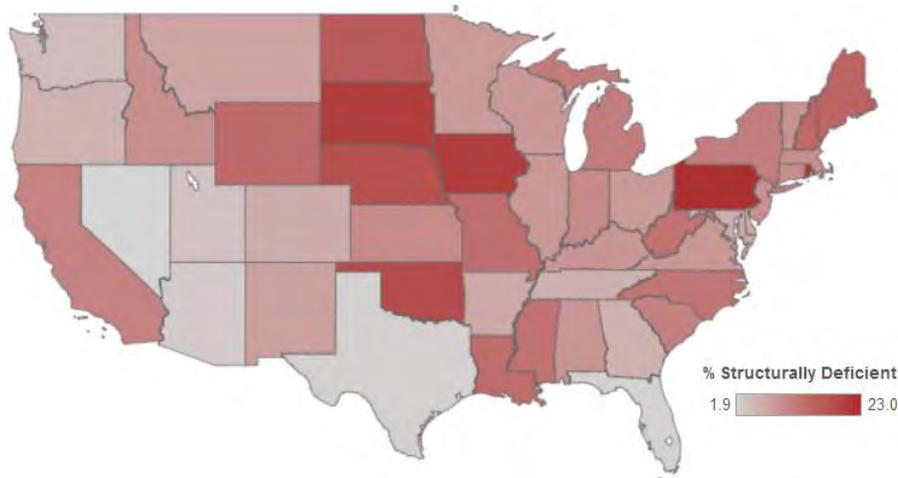


Figure 9.1: 2013 State bridge condition ratings (Governing., June 2013).

Table 9.1: Changes in states' structurally deficient bridge inspection ratings between 2007 and 2013 (FHWA, July 2014).

State	Change in S.D. Bridges	Percentage Change	2013 S.D. Bridges	2013 Bridges	2007 S.D. Bridges	2007 Bridges
Alabama	-536	-27.6	1,405	16,078	1,941	15,881
Alaska	-32	-19.4	133	1,196	165	1,229
Arizona	51	27.3	238	7,862	187	7,387
Arkansas	-151	-14.6	880	12,748	1,031	12,534
California	-480	-14.8	2,769	24,955	3,249	24,189
Colorado	-49	-8.4	536	8,612	585	8,374
Connecticut	52	14.4	413	4,218	361	4,175
Delaware	35	166.7	56	864	21	856
District of Columbia	-4	-16	21	252	25	245
Florida	-48	-15.6	259	12,070	307	11,664
Georgia	-213	-20.3	835	14,769	1,048	14,563
Hawaii	-4	-2.7	144	1,125	148	1,117
Idaho	47	13.1	406	4,232	359	4,104
Illinois	-257	-10.2	2,275	26,621	2,532	25,998
Indiana	-125	-6	1,944	18,953	2,069	18,494
Iowa	-187	-3.6	5,043	24,398	5,230	24,776
Kansas	-453	-15.1	2,554	25,171	3,007	25,464
Kentucky	-136	-9.9	1,234	14,116	1,370	13,639
Louisiana	39	2.2	1,827	13,050	1,788	13,342
Maine	10	2.8	366	2,402	356	2,387
Maryland	-63	-15.9	333	5,291	396	5,128
Massachusetts	-103	-17.5	487	5,136	590	5,019
Michigan	-459	-26.1	1,298	11,022	1,757	10,923
Minnesota	-72	-6.2	1,086	13,137	1,158	13,067
Mississippi	-834	-26.8	2,274	17,044	3,108	17,007
Missouri	-1119	-25	3,357	24,350	4,476	24,071
Montana	-113	-23.1	376	5,126	489	4,982
Nebraska	343	14.3	2,739	15,370	2,396	15,475
Nevada	-16	-30.8	36	1,853	52	1,705
New Hampshire	-45	-11.3	355	2,438	400	2,364
New Jersey	-133	-17.6	624	6,566	757	6,448
New Mexico	-109	-26.8	298	3,935	407	3,850
New York	-83	-3.8	2,078	17,442	2,161	17,361
North Carolina	27	1.2	2,308	18,168	2,281	17,783
North Dakota	-39	-5.1	726	4,439	765	4,458
Ohio	-708	-24	2,242	27,015	2,950	27,999
Oklahoma	-1701	-28.7	4,227	22,912	5,928	23,528
Oregon	-100	-18.8	431	7,656	531	7,318

Table 9.2: Changes in states' structurally deficient bridge inspection ratings between 2007 and 2013, cont'd (FHWA, July 2014).

State	Change in S.D. Bridges	Percentage Change	2013 S.D. Bridges	2013 Bridges	2007 S.D. Bridges	2007 Bridges
Pennsylvania	-754	-12.6	5,218	22,660	5,972	22,325
Rhode Island	-4	-2.3	167	766	171	748
South Carolina	-249	-19.2	1,048	9,275	1,297	9,221
South Dakota	-14	-1.1	1,210	5,875	1,224	5,924
Tennessee	-178	-13.3	1,157	20,058	1,335	19,838
Texas	-967	-43	1,283	52,561	2,250	50,272
Utah	-127	-52	117	2,974	244	2,851
Vermont	-284	-53.1	251	2,731	535	2,712
Virginia	-28	-2.3	1,186	13,765	1,214	13,418
Washington	-38	-9.3	372	7,902	410	7,686
West Virginia	-129	-12	944	7,125	1,073	7,007
Wisconsin	-110	-8.4	1,198	14,088	1,308	13,798
Wyoming	40	9.9	443	3,099	403	3,030

More specifically, the bridge statistics for Massachusetts are given in Table 9.3, and the comparison between Massachusetts bridges and U.S. bridges is displayed in Table 9.4.

Table 9.3: Massachusetts bridge statistics (FHWA, July 2014).

Total Bridges	5136
Total Deficient Bridges	2694
Structurally Deficient Bridges	487
Functionally Obsolete Bridges	2207

Table 9.4: Massachusetts bridges compared to U.S (FHWA, July 2014).

Percent Structurally Deficient	9.5%
U.S. Structurally Deficient	10.5%
Percent Functionally Obsolete	43.0%
U.S. Functionally Obsolete	13.9%

At this point, the main goals of FHWA are to make the repair of structurally deficient urban bridges, increase the annual investment levels for bridge repair,

reconstruction, and renovation, develop a national strategic plan for addressing the nation's structurally deficient and functionally obsolete bridges, set a national goal to decrease the number of structurally deficient bridges to 8% by 2020 (ASCE., July 2014).

FHWA reported that the estimated total cost for repairing or replacing only the deficient bridges is around \$76 billion. However, the total investment backlog for nation's bridges, which covers all cost-beneficial bridge needs along with the rehabilitation or replacement costs, is estimated to be \$121 billion. This means an investment amount of \$20.5 billion must be made annually in order to eliminate the bridge backlog by 2028. However, currently the amount of money annually spent on the nation's bridges is only \$12.8 billion.

The mentioned cost-beneficial bridge needs include routine inspection of bridges. Laser scanner technology provides an opportunity to enhance the current routine inspection methods, and hopefully reduce the cost associated with these strategies.

## **9.2 Bridge Condition Ratings**

As described in the background chapter, this research focuses on condition rating, which compares the current physical state of the structure to what it was the day it was built, and structural evaluation, which gives the overall condition of the structure based on all major deficiencies, and its ability to carry loads. The aim is to classify the damage severity and to assign labels to the detected damage in a standardized format based on sample condition rating guidelines from different States.

The summary of condition rating guidelines for the deck, superstructure and substructure, which is taken from Manual Bridge Inspection Manual of Ohio, is already

given in Table 2.1. In this section, another sample condition rating and deficiency reporting guides for deck (Item 58), superstructure (Item 59), substructure (Item 60), and channels (Item 61), taken from the Casey Overpass inspection report (O'Artista and Beintum, 2010), are shown as Table 9.5. Even though the condition rating guidelines given in Table 9.5 are less detailed than the guidelines in Table 2.1, they contain information on deficiency reporting as well as condition rating.

Table 9.5: Condition rating and deficiency reporting guides for Massachusetts Department of Transportation (O'Artista and Beintum, 2010).

<b>CONDITION RATING GUIDE</b>			(For Items 58, 59, 60 and 61)
CODE	CONDITION	DEFECTS	
N	NOT APPLICABLE		
G 9	EXCELLENT	Excellent condition.	
G 8	VERY GOOD	No problem noted.	
G 7	GOOD	Some minor problems.	
F 6	SATISFACTORY	Structural elements show some minor deterioration.	
F 5	FAIR	All primary structural elements are sound but may have minor section loss, cracking, spalling or scour.	
P 4	POOR	Advance section loss, deterioration, spalling or scour.	
P 3	SERIOUS	Loss of section, deterioration, spalling or scour have seriously affected primary structural components. Local failures are possible. Fatigue cracks in steel or shear cracks in concrete may be present.	
C 2	CRITICAL	Advance deterioration of primary structural elements. Fatigue cracks in steel or shear cracks in concrete may be present or scour may have removed substructure support. Unless closely monitored it may be necessary to close the bridge until corrective action is taken.	
C 1	"IMMINENT" FAILURE	Major deterioration or section loss present in critical structural components or obvious vertical or horizontal movement affecting structure stability. Bridge is closed to traffic but corrective action may put it back in light service.	
0	FAILED	Out of service - beyond corrective action.	

<b>DEFICIENCY REPORTING GUIDE</b>	
<b>DEFICIENCY:</b>	A defect in a structure that requires corrective action.
<b>CATEGORIES OF DEFICIENCIES:</b>	
<b>M= Minor Deficiency-</b>	Deficiencies which are minor in nature, generally do not impact the structural integrity of the bridge and could easily be repaired. Examples include but are not limited to: Spalled concrete, Minor pot holes, Minor corrosion of steel, Minor scouring, Clogged drainage, etc.
<b>S= Severe/Major Deficiency-</b>	Deficiencies which are more extensive in nature and need more planning and effort to repair. Examples include but are not limited to: Moderate to major deterioration in concrete, Exposed and corroded rebars, Considerable settlement, Considerable scouring or undermining, Moderate to extensive corrosion to structural steel with measurable loss of section, etc.
<b>C-S= Critical Structural Deficiency -</b>	A deficiency in a structural element of a bridge that poses an extreme unsafe condition due to the failure or imminent failure of the element which will affect the structural integrity of the bridge.
<b>C-H= Critical Hazard Deficiency -</b>	A deficiency in a component or element of a bridge that poses an extreme hazard or unsafe condition to the public, but does not impair the structural integrity of the bridge. Examples include but are not limited to: Loose concrete hanging down over traffic or pedestrians, A hole in a sidewalk that may cause injuries to pedestrians, Missing section of bridge railing, etc.
<b>URGENCY OF REPAIR:</b>	
<b>I = Immediate-</b>	[Inspector(s) immediately contact District Bridge Inspection Engineer (DBIE) to report the Deficiency and to receive further instruction from him/her].
<b>A = ASAP-</b>	[Action/Repair should be initiated by District Maintenance Engineer or the Responsible Party (if not a State owned bridge) upon receipt of the Inspection Report].
<b>P = Prioritize-</b>	[Shall be prioritized by District Maintenance Engineer or the Responsible Party (if not a State owned bridge) and repairs made when funds and/or manpower is available].

In order to determine the condition rating of individual members, several state condition rating guidelines are investigated. Although all the developed condition rating guidelines are based on the National Bridge Inventory (NBI), which is prepared by FHWA, they are significantly different from each other in terms of the descriptions given in the condition rating tables. Even though the general outline for the condition rating is

always the same, ranging from 0 to 10 (where 0 represents the imminent failure, and 10 is the excellent condition), the descriptions for each damage level varies from state to state. Ohio Department of Transportation's Manual of Bridge Inspection provides very detailed descriptions for individual bridge items; thus, in this research this manual is used as the reference point for condition rating determination.

Following tables represent the separate guidelines followed to determine the condition rating of each individual member for different damage types. Table 9.6 describes the condition rating criteria for alignment of the structural members, which is used to investigate bent members. Table 9.7 and Table 9.8 give detailed descriptions on condition rating criteria for steel beams/girders/slabs used for rupture and point of discontinuity; and, finally, Table 9.9 gives condition rating criteria for concrete beams/girders/slabs used for cracks and spalling. All listed tables are taken from Massachusetts Department of Transportation's sample bridge inspection report and Ohio Department of Transportation's Manual of Bridge Inspection (O'Artista and Beintum, 2010; Ohio Department of Transportation., 2010). Table 9.10 and Table 9.11, which are taken from AASHTO Bridge Element Inspection Guide Manual, are also included since they provide more detailed condition state descriptions for reinforced concrete items compared to Table 9.9. Finally, in order to rate the timber piers, the National Bridge Inventory Condition Ratings listed by the Federal Highway Administration is used (FHWA., July 2014). Table 9.12 lists the condition rating criteria used for timber piers. The damage detection results obtained from several datasets and their associated condition ratings are given in the following chapter, Chapter 10.

Table 9.6: Criteria for alignment check (Ohio Department of Transportation., 2010).

Rating	Condition	Primary Members
1	Excellent	
	Very Good	No noticeable misalignment or distortion
	Good	Minor misalignment or distortion due to construction
2	Satisfactory	Moderate misalignment or distortion due to impact
	Fair	Out of plane distortion of tension zones/members. Moderate misalignment or distortion due to impact ( <i>example – up to 2” localized sweep of bottom flange on multi-beam steel bridge</i> )
3	Poor	Vertical deflection (sag) due to deteriorations or excessive dead loads. Major misalignment or distortion due to impact ( <i>example - more than 2” sweep of bottom flange on 2 beams on a multi-beam steel</i> ). Up to 1/8” horizontal movement for every 1-foot vertical on web of a steel beam.
	Serious	Vertical sag up of the span due to distortion ( <i>example 2-inch sag for a 100’ span</i> ). More than 1/8” horizontal movement for every 1-foot vertical on a steel beam web.
4	Critical	Vertical sag up of the span due to distortion ( <i>example more than 2-inch sag for a 100’ span</i> ) Out of plane distortion of compression zones/members. More than 1/4” horizontal movement for every 1-foot vertical on a steel beam web.
	Im. Failure	Any condition worse than above
	Failed	

Table 9.7 : Condition rating criteria for steel beams/girders/slab (Ohio Department of Transportation., 2010).

	Condition	*Section Loss and Deterioration	**Cracks
1	Excellent	None	
	Very Good	No measurable section loss or very minor section loss	
	Good	Insignificant section loss, minor	Minor problems; repaired, arrested or retrofitted cracks
2	Satisfactory	Minor section loss ( <i>ex. corr. hole in redundant fascia web or any interior beam stiffener or behind a bearing, Minor pitting, isolated areas of loss throughout member, some areas of deterioration</i> )	Some minor deterioration; Broken welds possible but no cracks in the base metal
	Fair	Beams are sound with some deterioration, moderate section loss ( <i>ex. Some areas of heavy pitting under expansion joints, corrosion hole allowed only in fascia beams in redundant structure, less than 1/3 loss in flanges in max-bending regions</i> )	Minor cracking ( <i>ie. small, all less than 2" long, stable cracks in base metal in compression zone, cracks in weld metal</i> )
3	Poor	Advanced section loss ( <i>ex. 1 non redundant or interior member with more than 1/3 loss in flange at flexure zone or more than 50% loss in web at supports, corrosion hole in interior or non redundant beam</i> )	Fatigue or out-of-plane bending cracks ( <i>ie. cracks, any longer than 2" in compression zone, or small crack(s) in tension zone</i> ). Note: any crack in the base metal of a FCM parallel to the primary stress shall be "Poor" or less.
	Serious	Section loss is seriously affecting beams, local failures are possible ( <i>ex. Extensive perforations or loss through beams, perforations through many beams</i> )	Fatigue or out-of-plane bending cracks ( <i>ie. several cracks, any longer than 2" in compression zone, or several small crack(s) in tension zone</i> ). Multiple out of plane or fatigue cracks in one beam. Note: any crack in the base metal of a FCM perpendicular to the primary stress shall be "Serious" or less.
4	Critical	Advanced deterioration ( <i>ex. One beam is crushing</i> ) lane should be closed or closely monitored. Distortion in a load path of a redundant member	Advanced deterioration, cracks have removed support or load path in base metal ( <i>ex. many working crack(s) in more than one location or longer than 2", on the same beam and in tension zones</i> )
	Imminent Failure	Major section loss, deterioration or cracking that is worse than above ( <i>ex. Beams are crushing</i> ) and closed to traffic. Distortion in a load path of a non redundant member	
	Failed	Beyond corrective action	

Table 9.8: Percentage steel section loss values with respect to depth of section loss (Ohio Department of Transportation., 2010).

<b>% Section Loss plate thicknesses</b>				
<b>Depth of Section Loss</b>	<b>Original Plate Thickness (in)</b>			
	<b>1/4</b>	<b>1/2</b>	<b>5/8</b>	<b>3/4</b>
1/16	25%	13%	10%	8%
1/8	50%	25%	20%	17%
3/16	75%	38%	30%	25%
1/4	100%	50%	40%	33%
5/16	corrosion hole	63%	50%	42%
3/8		75%	60%	50%
7/16		88%	70%	58%
1/2		100%	80%	67%
9/16		corrosion hole	90%	75%
5/8		100%	83%	
11/16		corrosion hole	92%	
3/4			100%	

Table 9.9: Condition rating criteria for concrete beams/girders/slab (Ohio Department of Transportation., 2010).

Rating	Spalling with Exposed Reinforcement, Unsound Areas	Damp or Dark Areas (Saturation)	Cracking	
	% of area not including the SLAB fascias			
<b>1</b>	Excellent	No signs of distress, no discoloration		
	Very Good	Minor delaminations, minor spalling	Minor discoloration	Isolated hairline cracking with no rust staining, no dampness, no leakage
	Good	<b>Up to 1%</b>	<b>Up to 5%</b>	Minor problems, hairline cracking with isolated leaking, efflorescence. No rust staining. Map cracking combined with mottled areas up to 5%
<b>2</b>	Satisfactory	<b>Up to 5%</b> (unsound areas up to 10%)	<b>Up to 10%</b>	Minor structural cracking with leaking, efflorescence and rust staining. Map cracking combined with mottled areas up to 10%
	Fair	<b>Up to 10%</b> , (unsound areas up to 20%) slab with more than 1/3 primary bars exposed in one transverse plane.	<b>Up to 20%</b>	Structural cracking with leaking, efflorescence and rust staining. Map cracking combined with mottled areas up to 20%
<b>3</b>	Poor	<b>Combined total not exceeding 30%</b> with more than 4 adjacent exposed reinforcing bars having greater than 10% section loss to the original diameter.		Advanced cracking with heavy leaking, efflorescence and rust staining. Hairline map cracking combined with mottled areas up to 30%
	Serious	<b>Up to 30%</b> spalling, delamination with more than 5 adjacent reinforcing bars have greater than 25% section loss to the original diameter OR <b>Up to 50%</b> damp/dark areas		
<b>4</b>	Critical	<b>More than 30%</b> spalling, delamination with more than 10 adjacent reinforcing bars having greater than 25% section loss to the original diameter OR <b>More than 50%</b> damp/dark areas; Evidence that full depth hole(s), leading to structural failure of the slab in the traveled lane.		
	Imminent Failure	Major deterioration in critical structural components or obvious vertical or horizontal movement affecting structure stability. Bridge is closed to traffic but correction action may put back in light service		
	Failure	Bridge closed, collapsed		

Table 9.10: Condition state definitions for reinforced concrete items (AASHTO., 2010).

Defect	Condition State 1	Condition State 2	Condition State 3	Condition State 4
Cracking	None to hairline	Narrow size and/or density	Medium size and/or density	The condition is beyond the limits established in condition state three (3) and/or warrants a structural review to determine the strength or serviceability of the element or bridge.
Spalls / Delaminations/ Patched Areas	None	Moderate spall or patch areas that are sound	Severe spall or patched area showing distress	
Efflorescence	None	Moderate without rust	Severe with rust staining	
Load Capacity	No Reduction	No Reduction	No Reduction	

Table 9.11: Element definitions for condition states given in Table 9.10 (AASHTO., 2010).

Defect	Hairline - Minor	Narrow-Moderate	Medium-Severe
Cracking	< 0.0625 inches (1.6 mm)	0.0625 – 0.125 inches (1.6 – 3.2 mm)	>0.125 inches (3.2 mm)
Spalls/ Delaminations	N/A	Spall less than 1 inch (25 mm) deep or less than 6 inches in diameter	Spall greater than 1 inch (25 mm) deep or greater than 6 inches in diameter or exposed rebar
Cracking Density	Spacing Greater than 3.0 feet (0.33 m)	Spacing of 1.0 and 3.0 feet (0.33 – 1.0 m)	Spacing of less than 1 foot (0.33 m)
Efflorescence	NA	Surface white without build-up or leaching	Heavy build-up with rust staining

Table 9.12: Condition rating criteria for substructure that contains timber piers.

Rating		Substructure	
		Timber	
1	9	<b>Excellent</b>	No noticeable or noteworthy deficiencies which affect the condition of the substructure item.
	8	<b>Very Good</b>	Insignificant damage caused by drift or collision with no misalignment and no corrective action required.
	7	<b>Good</b>	Insignificant decay, cracking or splitting of timber substructure unit.
2	6	<b>Satisfactory</b>	Some initial decay, cracking or splitting of timber in a timber substructure unit. Fire damage limited to surface scorching of timber with no measurable section loss.
	5	<b>Fair</b>	Moderate decay, cracking or splitting of timber with minor, measurable section loss. Some exposure of timber piles as a result of erosion, reducing the embedment.
3	4	<b>Poor</b>	Substantial decay, cracking, splitting or crushing of primary timber members, requiring some replacement. Fire damage with significant section loss of timber which may reduce the load carrying capacity of the member. Extensive exposure of timber piles as a result of erosion, reducing the penetration and affecting the stability of the unit.
	3	<b>Serious</b>	Severe section loss in critical areas. Major fire damage to timber which will substantially reduce the load carrying capacity of the member. Bearing areas seriously deteriorated with considerable loss of bearing.
4	2	<b>Critical</b>	Primary timber members crushed or split and ineffective. Pier has settled.
	1	<b>Imminent Failure</b>	Bridge closed. Corrective action may be put back in light service.
	0	<b>Failed</b>	Bridge closed. Replacement necessary.

### **9.3 Bridge Load Ratings**

As mentioned before, the main objective of routine bridge inspections is to assess the current condition of an investigated structure. The recorded condition ratings and as-is component states are later used for computing bridge load ratings. The safe load capacity of the investigated bridge is then determined by using the calculated bridge load ratings. It should be noted that this section is aimed to provide a brief introduction on bridge load ratings, so mainly the effect of recorded condition ratings on the load rating computations is discussed. Therefore, detailed information on individual components of the load rating equation is not displayed.

The load rating capacity of a bridge is computed generally by using as-built bridge plans including all the modifications and the rehabilitation plans. The input from the latest inspection report, which is in accordance with the National Bridge Inventory (FHWA, July 2014), is included to introduce the effect of deterioration on load ratings. This input is composed of the latest recorded condition ratings and as-is member geometry (member shape descriptions).

There are two load rating levels: inventory rating and operating rating. Inventory rating checks whether a bridge is safe for the State legal loads within the federal weight laws and LRFD exclusion limits. Operation rating, on the other hand, represents the absolute maximum permissible load level to which the structure may be subjected for the vehicle type used in the rating.

For both load rating levels described in the previous paragraph, load ratings are calculated using one of three available load rating methods. These methods can be listed as: load and resistance factor rating (LRFR), load factor rating (LFR), and allowable

stress rating (ASR). Special trucks are used for each load rating method, and the load ratings are calculated by applying the associated truck load on the system.

Load rating is expressed as a rating factor,  $RF$ , in LRFD, and it is expressed in a tonnage for a particular vehicle in LFR and ASR. In this section, only the equations given in LRFD are displayed;  $RF$  is calculated using Equations 9.1-9.3 (AASHTO., 2011; FHWA., 2012; ADOT., July 2014).

$$RF = \frac{C - (\gamma_{CD})(DC) - (\gamma_{DW})(DW) \mp (\gamma_P)(P)}{(\gamma_{LL})(LL + IM)} \quad 9.1$$

$$C = \phi_c \phi_s \phi R_n \quad 9.2$$

$$\phi_c \phi_s \geq 0.85 \quad 9.3$$

where

$RF$  = Rating Factor,

$C$  = Capacity,  $C = f_R$  for the service limit state,

$f'_c$  = Specified compressive strength of concrete (ksi),

$f'_{ci}$  = Specified compressive strength of concrete at time of initial loading or prestressing (ksi),

$f_R$  = Allowable stress specified in the LRFD code or as stated,

$R_n$  = Nominal member resistance (as calculated),

$DC$  = Dead load effect due to structural components and attachments,

$DW$  = Dead load effect due to wearing surface and utilities,

$P$  = Permanent loads other than dead loads,

$LL$  = Live load effect,

$IM$  = Dynamic load allowance,

$\gamma_{DC}$  = LRFD load factor for structural components and attachments,

$\gamma_{DW}$  = LRFD load factor for wearing surfaces and utilities,

$\gamma_P$  = LRFD load factor for permanent loads other than dead loads = 1.0,

$\gamma_{LL}$  = Evaluation live load factor,

$\phi_c$  = Condition factor,

$\phi_s$  = System factor and,

$\phi$  = LRFD resistance factor.

The condition factors,  $\phi_c$ , corresponding to certain condition descriptions, which are in accordance with NBI, are displayed in Table 9.13. As it can be observed from Equations 9.1 and 9.2, the changes in  $\phi_c$  directly affect the calculated load ratings. Thus, it is important to determine condition ratings effectively. As mentioned in previous sections, laser scanners provide an opportunity to improve the current visual inspection methods through automated defect localization and quantification. Thus, it can be concluded that the mentioned improvements in routine inspection strategies would also affect load ratings calculations.

Table 9.13: Condition factors (FHWA., 2012).

Condition Description	NBI Rating	$\phi_c$
good or satisfactory	6 or higher	1.00
fair	5	0.95
poor	4 or lower	0.85

Another important load rating input that can be extracted using laser scanners is the as-is geometric properties of bridge members. This process can be automated, and changes that occur between consecutive inspections can be recorded easily. The extracted

geometric information, which generally contains cross-section information, is used to calculate the remaining capacity of a member. A representative example is given in Figure 9.2, where Figure 9.2(a) displays the model drawing of a C-section from the steel lab frame, and Figure 9.2(b) shows the corresponding damaged cross-section extracted from the captured point cloud. Since the geometric information is an important part of load rating calculations, the use of laser scanning technology for member dimension extraction can significantly improve the current load rating calculation strategies.

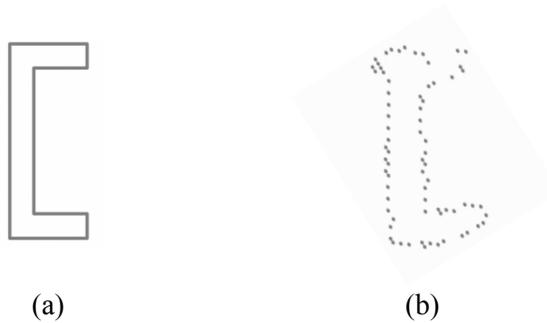


Figure 9.2: (a) Model drawing of a C-section and (b) damaged cross-section representation extracted from laser scan.

## **10. Damage Detection Applications and Results**

In Chapters 5 through 8, several methods for both structural sensing and surface damage assessment are introduced. Representative examples are given, required parameters for each are listed, and the effects of these listed parameters on the obtained results are discussed. In Chapter 9, the current status of visual inspection strategies are presented, and the selected condition assessment criteria for several items of bridges, which are used to convert detected damage information into common reporting outputs, are outlined.

In this chapter, in order to validate the accuracy of the algorithms developed in previous chapters and to test their efficiency, several synthetic, experimental, and field datasets have been processed. The introductory information on the available datasets is given in Section 4.1.

This chapter begins with Section 10.1, which gives detailed information on the datasets that are used both for structural sensing and damage assessment; this section extends the introductory information given in Section 4.1. Section 10.1 is composed of individual sections that focus on each utilized dataset separately. Section 10.2 first lists the validation datasets used for showing the results for the object detection methods described in Chapter 6 and then continues with laying out the validation and testing datasets used for presenting the results for each damage type that is in the scope of this research. The damage types, for which the same damage detection algorithms are used, are grouped together. Finally, Section 10.3 presents the object detection results for the datasets listed in Section 10.2 and similarly, Section 10.4 presents the surface damage detections results for the datasets given in Section 10.2. In general, validation refers to

the cases when the detection results, either for object detection or damage detection, are compared with the actual values recorded for the specific application. For object detection, the actual values represent the member type and/or member dimensions, whereas for damage detection, they are hand-measured dimensions. Testing, on the other hand, represent culminating examples for this research and refers to the cases for which both object and damage detection is completed; however, the obtained results are not compared with actual values since there is not any available recorded validation sets for these cases.

This chapter ends with Section 10.5, which presents the condition ratings computed for individual elements of the processed datasets described in Section 10.1. These condition ratings are obtained by using the condition rating criteria given in Section 9.2.

## **10.1 Available Datasets**

In this research, several datasets are used for both structural sensing and damage assessment. This section consists of detailed information on the processed datasets and generated model libraries, which can be listed as synthetic point clouds and model libraries, a steel testing frame, a concrete testing frame, a DeKalb County Bridge in Dekalb County, Illinois, and the Bowker Overpass in Boston, Massachusetts. Each section, from Section 10.1.1 to Section 10.1.5, is dedicated to an individual dataset from the list given above. The introductory information on each dataset is given in Section 4.1.

### 10.1.1 Synthetic Point Clouds and Model Libraries

Synthetic point clouds and their associated model libraries are generated to provide a comprehensive database that includes descriptive properties of individual objects. This database is then used to interpret the results obtained through both the global feature-based and graph-based object detection algorithms. As mentioned in Chapter 5, an essential part of object recognition process is creating an extensive model library. In this research, a model library that is composed of a variety of object descriptions including simple prismatic shapes, such as rectangular prisms, cylinders, etc., and also more sophisticated objects is created.

Representative examples for both synthetic point clouds and model library representations are given in previous chapters. First, a sample synthetic dataset is shown in Chapter 8, Figure 5.13 and Figure 5.14. This sample dataset is used to validate the efficiency of the point cloud processing algorithms described in Chapter 5 and also the global feature-based object detection method given in Section 6.3. The results are included in the related sections. Second, an example model library representation of a C-section is given in Section 6.4.3, and the result of the object detection is shown in Section 6.4.4.

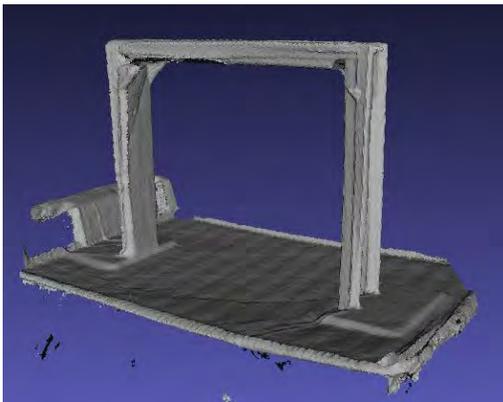
In this research, synthetic datasets are only used for extracting and storing the descriptive properties of some common objects; validation is later carried on with recorded laser scanner data. The validation results for the datasets discussed in Section 10.1 are then shown in Section 10.3.

### 10.1.2 Steel Testing Frame

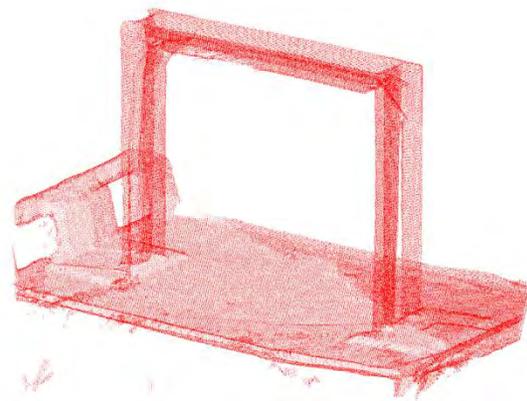
The first point cloud dataset, for which the processing results are discussed, is the steel testing frame. It is a 2D frame that consists of three 22" long identical 1.5"x4"x1/4" C-sections. Figure 10.1(a) and Figure 10.1(b) represent the surface model and point cloud of this frame respectively.

This dataset was collected with the Microsoft Kinect, and the surface model is obtained by using the ReconstructMe software. The information on the utilized equipment and software are discussed in Chapter 4 and Section 5.1 respectively.

Even though this frame represents a simple geometry, it was very useful in terms of testing the efficiency of the developed algorithms prior to performing large scale applications. The steel testing frame is used for validating graph-based object detection method and for performing alignment checks. The results are given in Sections 10.3 and 10.4.



(a)



(b)

Figure 10.1: (a) Surface model and (b) point cloud representations of lab frame.

### 10.1.3 Concrete Testing Frame

The concrete testing frame dataset was collected from a collapsed experimental test specimen that was used to predict progressive collapse resistance of a small-scale 2D physical model of a reinforced concrete structure in 2007 (Sasani et al., 2007; Sagioglu, 2012). The test setup is a 114" by 63" frame that consists of 3 stories with 4 equal length spans. Dimensions of the test setup are given in Figure 10.2, and the beam and column legends for the testing frame are shown in Figure 10.3.

This test setup was subjected to both dynamic and static loading. Figure 10.4(a) shows the testing frame with dynamic loading and Figure 10.4(b) shows the testing frame under pull down loading.

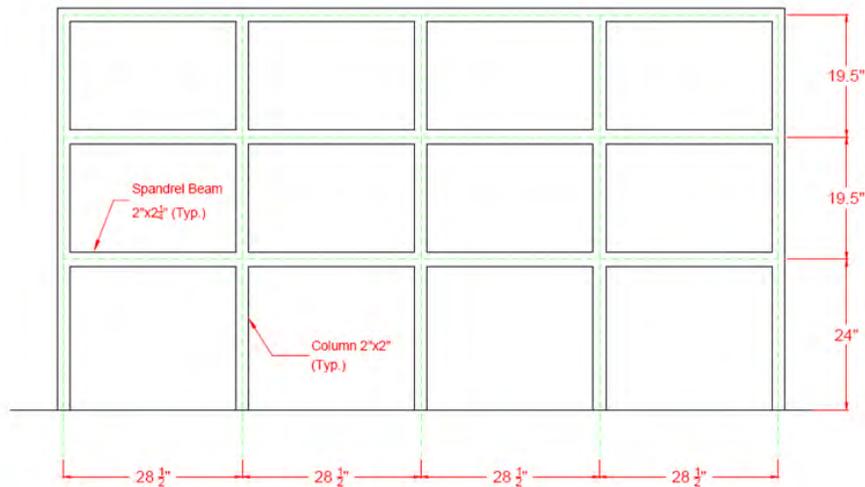


Figure 10.2: Dimensions of the test setup (Sagioglu, 2012).

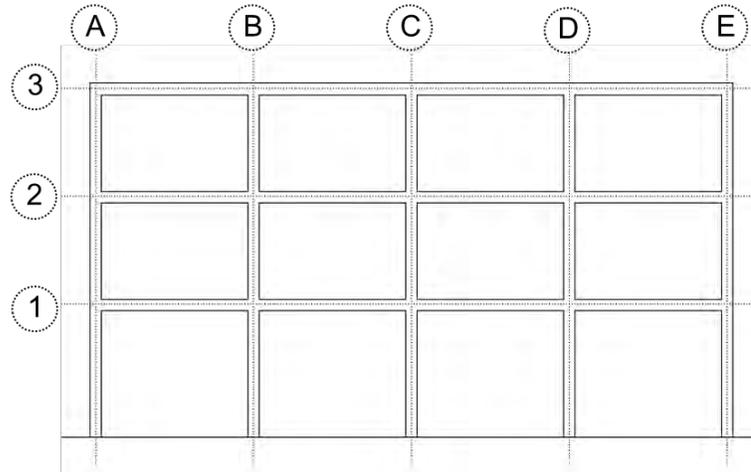
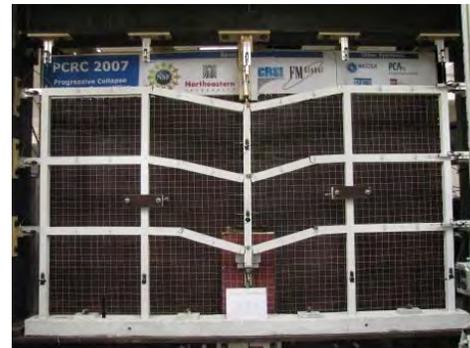


Figure 10.3: Beam and column legends of the testing frame.

The concrete testing frame was first subjected to dynamic loading by placing loads on beams at specific locations, and then breaking the glass column that was located in the middle of the bottom story, which is marked with red in Figure 10.4(a). Later, the test was continued with static loading since the frame resisted dynamic loading in the first phase of the test. Static loading representing gravity loading (pulling downwards in the figure) was applied under the middle column of the second story where the glass column had been located before dynamic loading.



(a)



(b)

Figure 10.4: (a) Testing frame with dynamic loading and (b) under pull down loading.

In order to capture the point cloud of this frame, a FARO Focus3D scanner was used. In total, 10 scans are collected around the test setup at varying elevations. These separate scans were later registered by the Faro Scene software in order to get the complete point cloud dataset for the concrete testing frame. The point cloud of the concrete testing frame is shown in Figure 10.5. The information on the used equipment and software are discussed in Chapter 4 and Section 5.1 respectively.

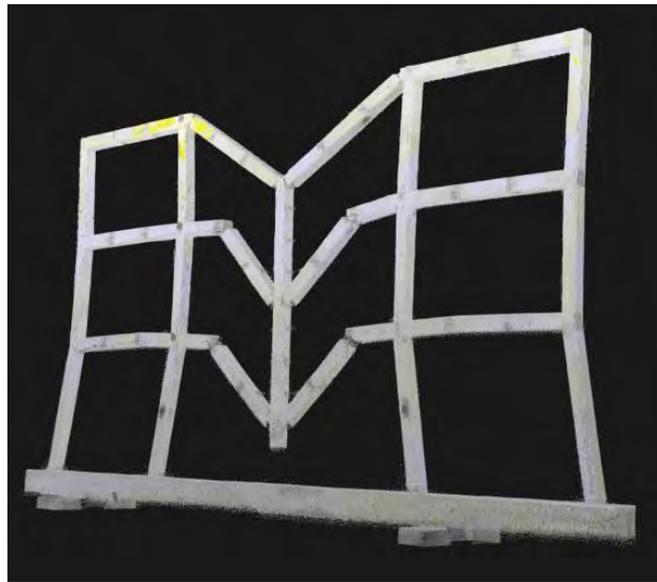


Figure 10.5: Point cloud representation of the concrete testing frame.

It should be noted that even though the combination of the captured scans provide coverage for the entire frame, the resultant point cloud is not adequate to obtain a uniform point density on the surface across the entire structure. Thus, the upper beams, which were exposed to the scanner more than the lower beams due to the location of the scanner during scanning, have higher point densities. The variation of the point densities on the surface of the concrete testing frame is discussed in Section 7.2.1.

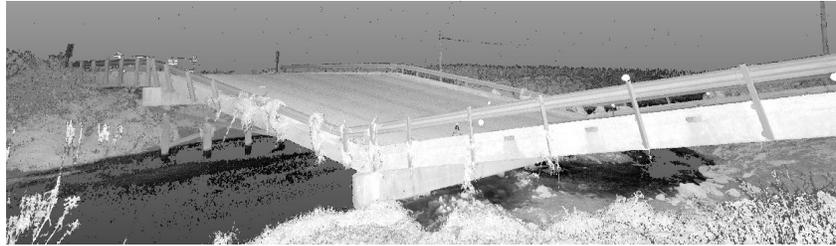
The concrete testing frame is first used for validating graph-based object detection method; the results are given in Sections 10.3. Later, it is used for both validating and testing the developed damage detection algorithms. This frame is used for detecting the following damage types: cracks, concrete spalling, bent members, and points of discontinuity. Most of the damage detection results for the concrete testing frame are given in Chapter 8; in which the obtained crack and concrete spalling results are improved by using a neural network classifier. The rest of the damage detection results obtained both for validation and testing are included in Section 10.4. Finally, the condition ratings computed for individual components of the concrete testing frame, beams and columns are given in Section 10.5.

#### 10.1.4 DeKalb County Bridge

In order to test the viability of the developed algorithms on a full-scale structure in the field, laser scanning was performed to collect the point cloud dataset of a collapsed bridge, DeKalb County Bridge in Illinois. The bridge was located in a rural area in Illinois and consisted of a precast concrete deck, reinforced concrete pier caps, and timber piers. One set of timber piers failed, causing the bridge deck to collapse at that location. A forensic investigation of the collapse is available in Borello et al. (2010). Figure 10.6(a) shows the picture of the bridge and Figure 10.6(b) shows the registered point cloud data.



(a)



(b)

Figure 10.6: (a) Picture and (b) point cloud of DeKalb County Bridge.

This captured point cloud, similar to the previously described datasets, is a combination of multiple scans registered together. A FARO Photon 80 scanner is used to capture 14 scans of the scene. A variety of perspectives was used in order to capture as much of the 3D geometry of the structure as possible; each viewpoint exposed a unique set of surfaces of the bridge. Each captured scan consists of approximately 41 million points, which resulted in a total of more than 500 million points for the registered point cloud. The captured scans were registered using the FARO Scene software, with the assistance of spherical reflectors placed on the structure. The information on the utilized equipment and software are discussed in Chapter 4 and Section 5.1 respectively.

Point cloud data captured in the field consists of noisy data containing irregular and unusual surfaces and objects, as well as sparse and missing points in occluded

sections of the scene. The purpose of analyzing the point cloud of this test-bed bridge is to test the robustness of the algorithms under such conditions.

For this research, only a portion of the DeKalb County Bridge was used; the processed portion of the bridge is shown in Figure 10.7. This dataset is first used for validating the global-feature based object detection method. It is then used for testing the developed damage detection algorithms; the volumetric changes in the timber piles due to damage, alignment of these timber piles after collapse, and the cumulative damage on the deck are investigated. The results are discussed in Sections 10.3 and 10.4. The condition ratings computed for individual components of the processed portion of the DeKalb County Bridge are given in Section 10.5.

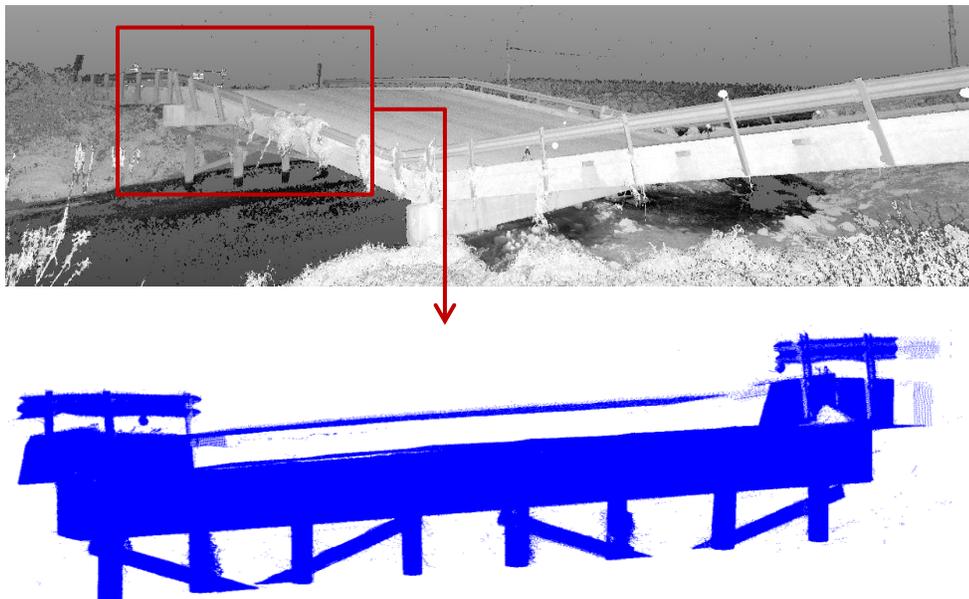


Figure 10.7: Point cloud of a portion of DeKalb County Bridge.

### 10.1.5 Bowker Overpass

The second full-scale point cloud dataset that was collected is from the Bowker Overpass, which is located in Boston, Massachusetts. This bridge is a steel girder bridge with a composite deck carrying Charlesgate Street over Commonwealth Avenue, Beacon Street, and Interstate 90 (Wikipedia., July 2014). Figure 10.8 shows the location of the Bowker Overpass on the map, and Figure 10.9 provides an aerial view of the structure with location plan.

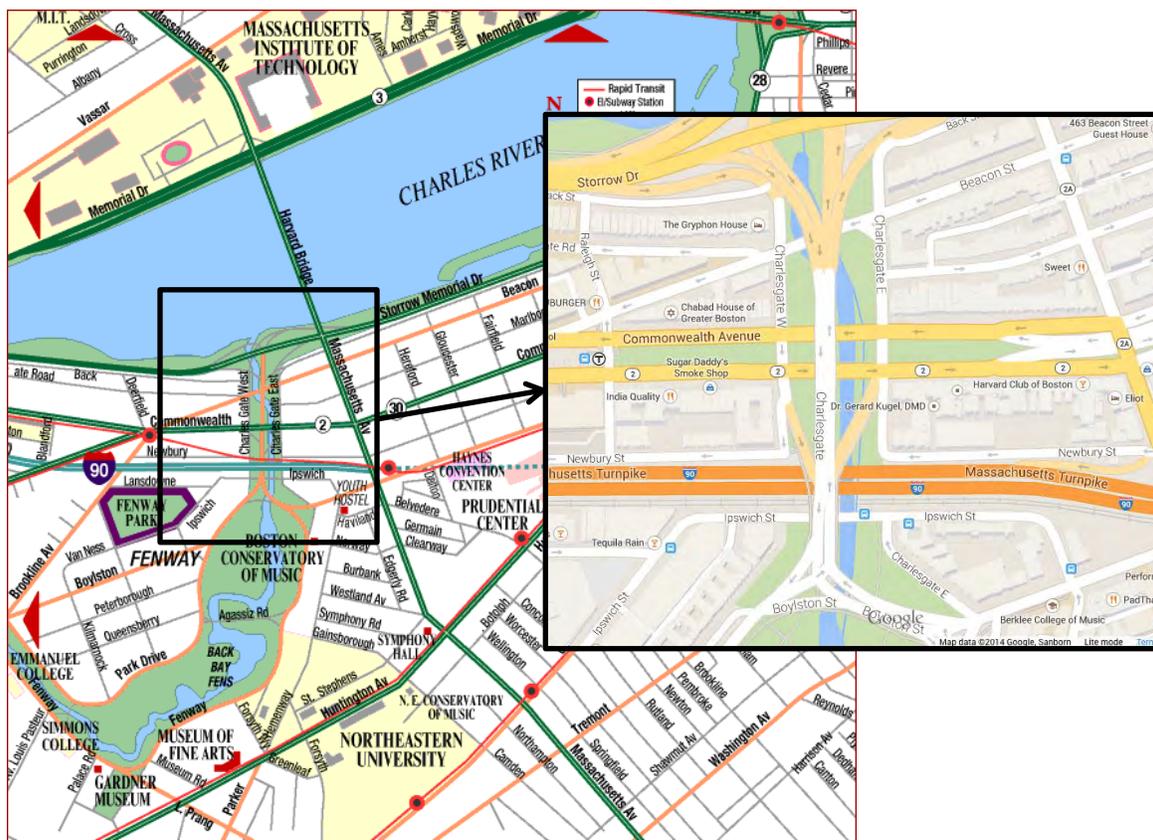


Figure 10.8: Location and aerial view of Bowker Overpass (Google Maps., July 2014; Accessmaps., May 2014).

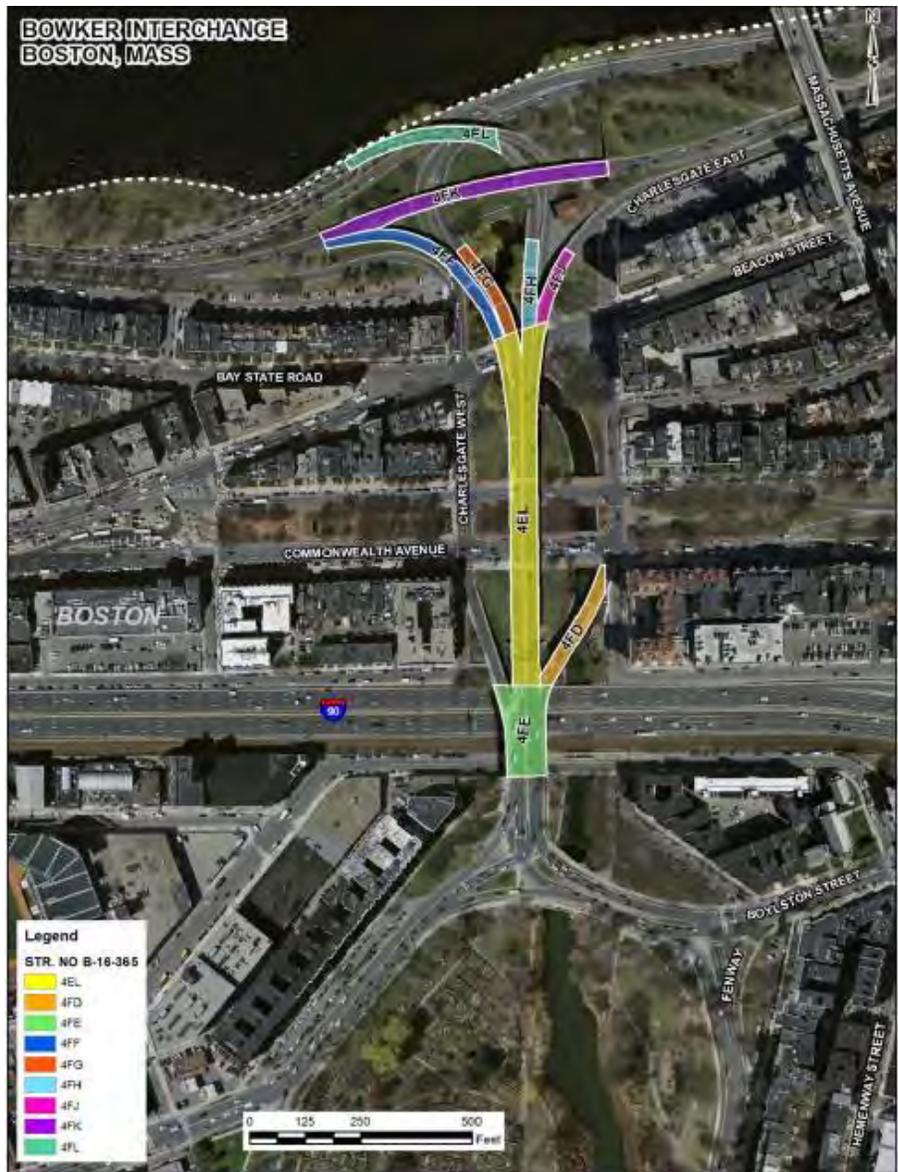


Figure 10.9: Location plan on the aerial view of Bowker Overpass (Mercuri and Mirliss, 2013).

The Bowker Overpass suffers from a significant amount of deterioration that includes concrete spalling, cracking, steel corrosion, section losses, and etc. The Massachusetts Department of Transportation (MassDOT) Highway Division has already announced on their website that public safety requires immediate repairs of the Bowker Overpass. At present, MassDOT is in the process of developing a project to repair the

bridge by replacing the deck and retrofitting the pin and hanger assemblies. These repairs aim to address the near term safety issues on the Bowker Overpass.

For this research, two parts of the overpass that had heavily damaged regions were scanned. The first location is Span 7, which is a part of Bridge 4FH as shown in Figure 10.9. The east fascia girder of Span 7 and the underside of the bridge, as well as the top of the bridge deck on the portions of the bridge adjacent to the east fascia, which are shown in Figure 10.10, were scanned. The second location is the joint that connects on Span 23 and Span 24 on the east side, which is part of the Bridge 4FJ as shown in Figure 10.9. These specific locations suffer from severe deterioration and contain examples for multiple defect types such as section loss, concrete spalling, exposed reinforcement, etc. Figure 10.10 shows the portion of the plan of the Bowker Overpass that highlights the scanning locations.

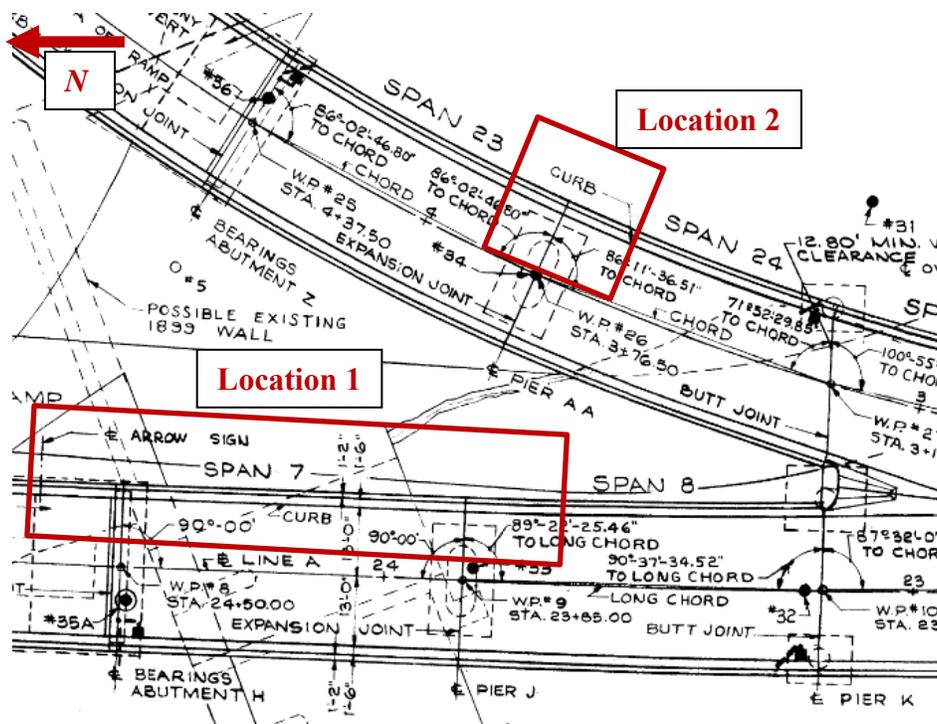


Figure 10.10: Plan of the Bowker Overpass with the focus on Bridge 4FH and Bridge 4FJ (MassDOT., 1963).

In order to provide complete coverage on both locations shown in Figure 10.10, 18 high-resolution laser scans are captured by using a FARO Focus 3D scanner. 3D data is captured along with the images. The scan registration and texture-mapping are completed using Faro Scene software, with the assistance of spherical reflectors placed on the structure. In a total, there are more than 40 billion points in the registered point cloud dataset. Figure 10.11 shows the entire point cloud, which is the combination of 18 scans. Figure 10.12 and Figure 10.13 represent the isolated point clouds for the Bridge 4FH, Span 7 and Bridge 4FJ, corner respectively. The information on the used equipment and software are discussed in Chapter 4 and Section 5.1 respectively. The report on the initial site investigation and laser scanning of the Bowker Overpass are given in Appendix E.



Figure 10.11: Entire point cloud.

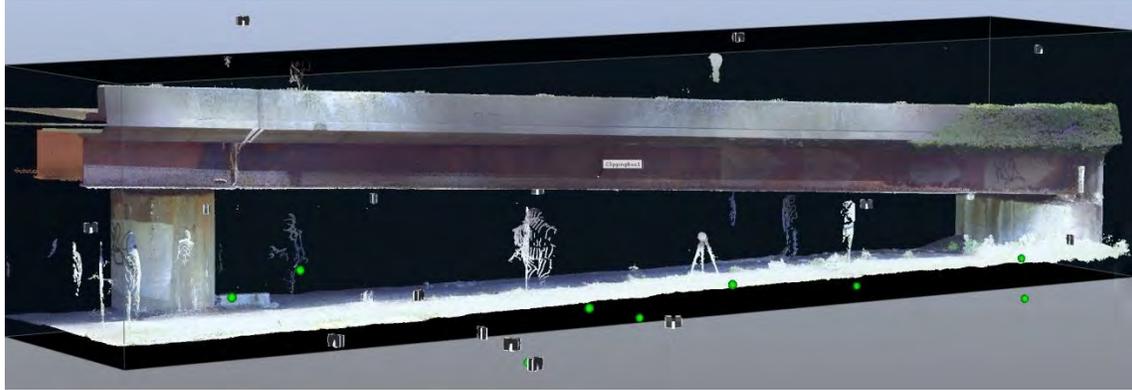


Figure 10.12: Point cloud of 4FH, Span 7.

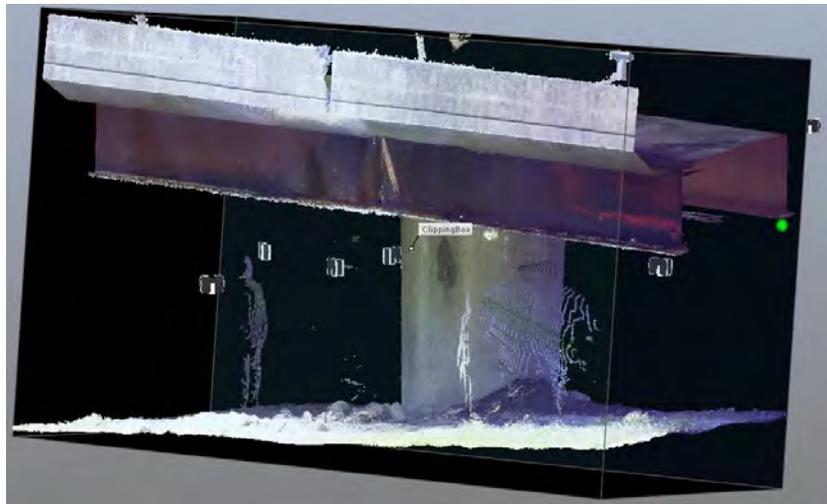


Figure 10.13: Point cloud of 4FJ, corner.

In this research, only several small portions of the Bowker Overpass are used for processing; the processed portions of the overpass are shown in the related parts of Sections 10.3 and 10.4. A portion of the overpass point cloud data is used for validating the graph-based object detection method. Several other regions of the point cloud are then used for both validating and testing the developed damage detection algorithms; detected damage types include cracks, spalled concrete regions, section losses, and etc. The results

are discussed in Sections 10.3 and 10.4. The condition ratings computed for individual components of the processed portions of the Bowker Overpass are given in Section 10.5.

## **10.2 Validation and Testing**

This section presents the lists of the generated libraries, validation and testing datasets used for both structural sensing and surface damage assessment. Table 10.1 lists the datasets used for validating the results obtained by using the object detection methods discussed in Chapter 6, including both global-feature based object detection and graph-based object detection. Table 10.2 presents the datasets used for validating and testing the developed damage detection algorithms for the damage categories given in Chapter 3. Each damage category is composed of several damage types; these damage types are also listed in Table 10.2. The results of the crack detection for the concrete testing frame are presented in Chapter 8; thus, no further discussions are included in the current chapter related to this dataset. It should also be noted that the corrosion detection and validation are performed on Bowker Overpass dataset; however, the results for the corrosion detection are included in Section 10.4.3, where damage types associated with large deformations with localized change in topology is discussed, since the corrosion locations on the investigated patches were either detected along with spalled concrete regions, or they were detected in the form of delamination.

Table 10.1: Libraries and validation datasets for object detection methods.

	<b>Libraries</b>	<b>Validation</b>
<b>Global Feature-based Object Detection</b>	3-D object library	DeKalb County Bridge
<b>Graph-based Object Detection</b>	2-D and 3-D object library	Steel Testing Frame
		Concrete Testing Frame
		Bowker Overpass

Table 10.2: Validation and testing datasets associated with different damage types.

<b>Damage Category</b>	<b>Damage Types</b>	<b>Validation</b>	<b>Testing</b>
<b>Small deformations</b>	<b>Cracks</b>	Concrete Testing Frame	Concrete Testing Frame
		Bowker Overpass	
<b>Large deformations with no change in topology</b>	<b>Bent members / Alignment issues / Points of Discontinuity</b>	Concrete Testing Frame	Steel Testing Frame
			Concrete Testing Frame
			DeKalb County Bridge
			Bowker Overpass
<b>Large deformations with localized change in topology</b>	<b>Cross-section change/ Rupture / Spalling / Steel Delamination</b>	Concrete Testing Frame	Concrete Testing Frame
			DeKalb County Bridge
		Bowker Overpass	Bowker Overpass

### 10.3 Object Detection

This section consists of the results obtained by using the object detection methods described in Chapter 6, global-feature based object detection and graph-based object detection. The list of the datasets and the associated object detection method is already listed in Table 10.1. The following sections present the results for each dataset. First, the global feature-based object detection results are given in Section 10.3.1; it is then followed by Section 10.3.2, where the graph-based object detection results are discussed.

The listed datasets are only used for validation purposes; thus, there is not a testing section under Sections 10.3.1 and 10.3.2, only validation.

### 10.3.1 Global-feature based Object Detection

In order to perform global feature-based object detection, first the processing steps described in Chapter 5 had to be performed. Once the mentioned steps were completed, the global feature-based object detection, which is described in Section 6.3 in detail, was applied. The following section, Section 10.3.1.1.1, presents the results of the global-feature based object detection applied on a portion of the DeKalb County Bridge.

Table 10.3 shows the parameters and corresponding values used for global feature-based object detection. For this application, in order to match any detected object with a library object, a similarity of at least 90% was required (i.e., the dissimilarity limits were set at 10% as shown in the table).

Table 10.3: Parameters and corresponding values for global feature-based object detection.

<b>Method</b>	<b>Parameters</b>	<b>Range</b>
Global Feature-based Object Matching	Local dissimilarity measure	10 %
	Global dissimilarity measure	10 %

#### 10.3.1.1 *Validation*

##### 10.3.1.1.1 DeKalb County Bridge

The global feature-based object detection was executed on a portion of the point cloud, which is shown in Figure 10.14(a). Both the pier cap and piles were detected as separate objects. Pier cap is detected as an object by using the connectivity information of nine separate planes; the detected planes were matched with the corresponding object

from the library. Timber piles, on the other hand, were detected as cylinders by using the properties of the extracted clusters. Figure 10.14(b) shows the detected pier cap and eight piles superimposed with the processed portion of the point cloud. The agreement between the point cloud, design documents, and the detected pile cap and timber piles were within tolerances. The detected object properties, dimensions, location, and etc., are also exported to a computer-aided design program, AutoCAD (Autodesk Inc., July 2014); the results are presented in Figure 10.15.



Figure 10.14: (a) Scan data of DeKalb County Bridge with section utilized for data processing, and (b) point cloud of the same section with four piers detected as cylinders and pier cap surface detected as planes.

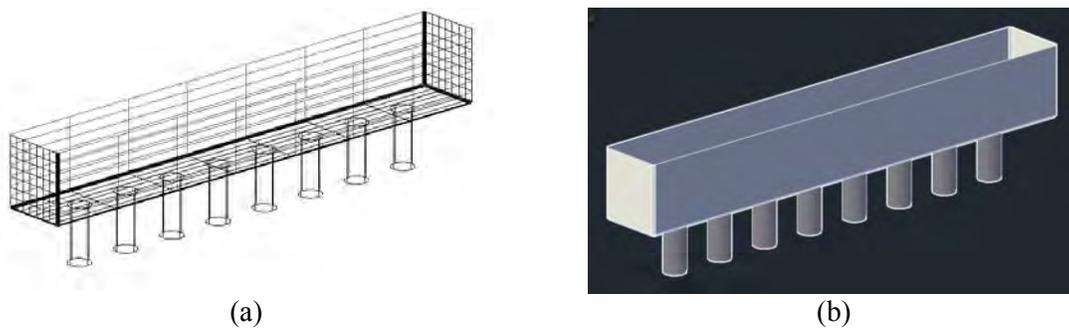


Figure 10.15: Detected objects' properties exported to a computer aided design software (CAD): (a) object representations and (b) solid surfaces.

### 10.3.2 Graph-based Object Detection

To perform graph-based object detection, first the processing steps described in Chapter 5 were performed. Once the mentioned steps were completed, the graph-based object detection, which is described in Section 6.4 in detail, was applied. The following sections, Sections 10.3.2.1.1-10.3.2.1.3, present the results of the graph-based object detection performed on the steel testing frame, concrete testing frame and a girder of Bowker Overpass, respectively.

Table 10.4 shows the parameters and corresponding values used for each step towards graph-based object detection. For skeletonization, a constant height function increment,  $h_p$ , of 2" is used, and the rest of the voxel dimensions are determined automatically based on the local point densities extracted from the point cloud. However, limiting values are defined for the minimum and maximum values for the utilized voxel size. The parameters for the iterative closest point algorithm are kept constant for all the datasets; these values are given in Table 10.4. Finally, the graph-based object detection algorithm is looking for an overall similarity of minimum 90% for both global and local features in order to match the detected object with the corresponding model representation.

Table 10.4: Parameters and corresponding ranges for the voxelization, model fitting and. graph-based object detection

<b>Method</b>	<b>Parameters</b>	<b>Range (or default value)</b>
Voxelization / Skeletonization	Voxel grid dimension, $r_p$	0.03 in. – 0.5 in.
	Height function increment, $h_p$	2 in
Iterative Closest Point	Maximum correspondence distance	2 in
	Number of iterations	100
	Transformation Epsilon	1.00E-08
	Euclidian Fitness Epsilon	1.0
Graph-based Object Matching	Local dissimilarity measure	10%
	Global dissimilarity measure	10%

### 10.3.2.1 Validation

#### 10.3.2.1.1 Steel Testing Frame

The skeleton of the entire steel testing frame was extracted both locally and globally. The local skeleton information, which represents the cross-section properties, is used for graph-based object detection. All three members are detected effectively; the detection results are given in Table 10.5. Once the object detection is completed, the geometric information of the detected objects is then exported to Revit (Autodesk Inc., July 2014) to create the information model of the investigated frame automatically. Figure 10.16 shows the generated building information model. The gusset plates were added after the model was generated.

The global skeleton information, on the other hand, is later used to check the alignment of each member. The results of the alignment check are given in Section 10.4.2.2.1.

Table 10.5: Results of graph-based object detection for steel testing frame.

	<b>Detected Objects</b>	<b>Dimensions (in)</b>
<b>Member 1</b>	C-section	$3/2 \times 4 \times 1/4$
<b>Member 2</b>	C-section	$3/2 \times 4 \times 1/4$
<b>Member 3</b>	C-section	$3/2 \times 4 \times 1/4$

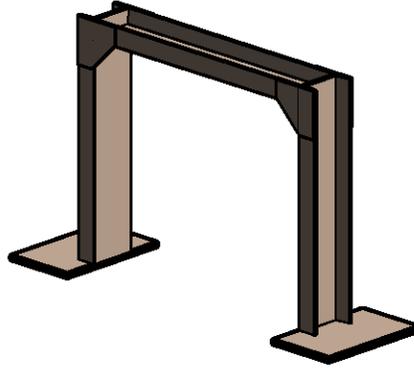


Figure 10.16: Revit model of the steel testing frame.

#### 10.3.2.1.2 Concrete Testing Frame

The same methodology from the previous section is applied to the concrete testing frame. First, the skeleton is extracted and then, the graph-based object detection is performed.

An example that represents the voxelization/skeletonization procedure is shown in Figure 10.17 and Figure 10.18. Figure 10.17(a) shows the gray-scale color mapped point cloud representation of the experimental setup and Figure 10.17(b) represents the 3D point cloud data of a section cut with a certain width, which is perpendicular to the centroid of the column. Figure 10.18(a) represents the skeleton of the system that is created by connecting the center points of the each section cut along z-axis for columns

and along x-axis for beams. Finally, Figure 10.18(b) shows the voxel representation of the section cut shown in Figure 10.17(b).

The cross-section information obtained through voxelization is then used for object detection. The results of the graph-based object detection for the columns and beams are given in Table 10.6 and Table 10.7 respectively. Beam and column legends of the concrete testing frame are shown in Figure 10.3.

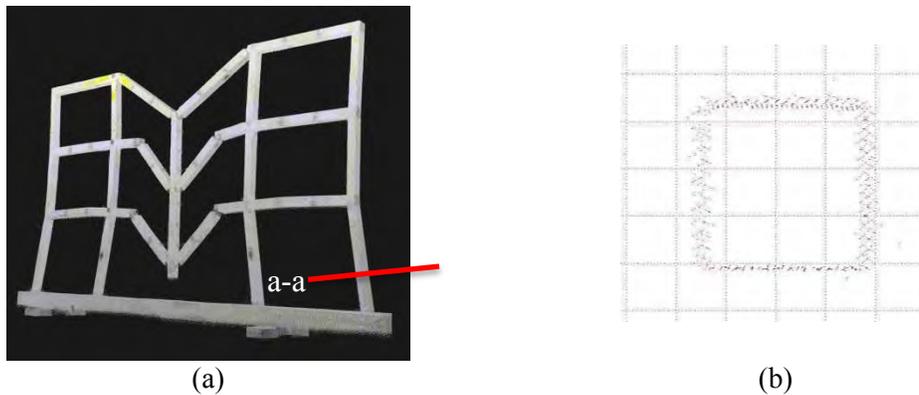


Figure 10.17: (a) The complete scan of the collapsed test setup and (b) Example section cut a-a at the specified location.

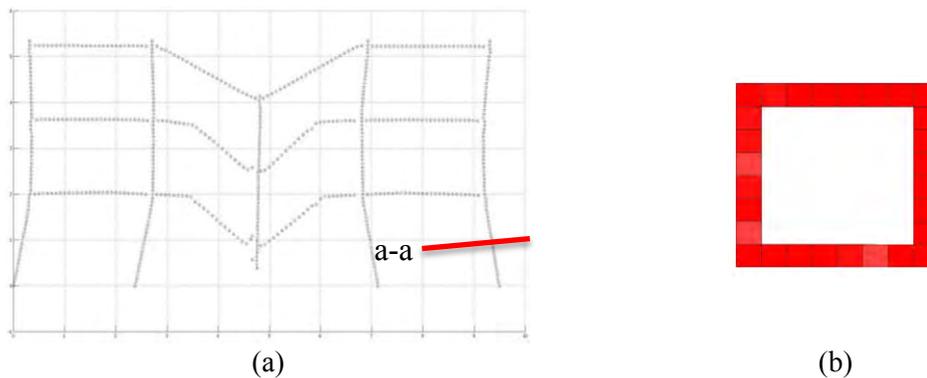


Figure 10.18: (a) Skeleton of the test setup and (b) Resulting voxel representation of the section cut a-a at the specified location associated with a certain node of the skeleton graph.

Table 10.6: Column detection results of graph-based object detection for concrete testing frame.

	<b>Detected Objects</b>	<b>Dimensions (in)</b>
<b>A1</b>	Rectangular prism	2 x 2
<b>A2</b>	Rectangular prism	2 x 2
<b>A3</b>	Rectangular prism	2 x 2
<b>B1</b>	Rectangular prism	2 x 2
<b>B2</b>	Rectangular prism	2 x 2
<b>B3</b>	Rectangular prism	2 x 2
<b>C1</b>	-	-
<b>C2</b>	Rectangular prism	2 x 2
<b>C3</b>	Rectangular prism	2 x 2
<b>D1</b>	Rectangular prism	2 x 2
<b>D2</b>	Rectangular prism	2 x 2
<b>D3</b>	Rectangular prism	2 x 2
<b>E1</b>	Rectangular prism	2 x 2
<b>E2</b>	Rectangular prism	2 x 2
<b>E3</b>	Rectangular prism	2 x 2

Table 10.7: Beam detection results of graph-based object detection for concrete testing frame.

	<b>Detected Objects</b>	<b>Dimensions (in)</b>
<b>AB1</b>	Rectangular prism	2 x 2 <sup>1/4</sup>
<b>AB2</b>	Rectangular prism	2 x 2 <sup>1/4</sup>
<b>AB3</b>	Rectangular prism	2 x 2 <sup>1/4</sup>
<b>BC1</b>	Rectangular prism	2 x 2 <sup>1/4</sup>
<b>BC2</b>	Rectangular prism	2 x 2 <sup>1/4</sup>
<b>BC3</b>	Rectangular prism	2 x 2 <sup>1/4</sup>
<b>CD4</b>	Rectangular prism	2 x 2 <sup>1/4</sup>
<b>CD5</b>	Rectangular prism	2 x 2 <sup>1/4</sup>
<b>CD6</b>	Rectangular prism	2 x 2 <sup>1/4</sup>
<b>DE1</b>	Rectangular prism	2 x 2 <sup>1/4</sup>
<b>DE2</b>	Rectangular prism	2 x 2 <sup>1/4</sup>
<b>DE3</b>	Rectangular prism	2 x 2 <sup>1/4</sup>

The detected object properties are not used to generate an information model since the frame is severely damaged. Thus, the model would not represent the actual condition of the concrete frame.

### 10.3.2.1.3 Bowker Overpass

The previous sections presented two applications for the graph-based object detection method, which were performed on laboratory testing specimens. However, this section provides an example for graph-based object detection performed on a field dataset, one of girders of the Bowker Overpass. Figure 10.19(a) represents the portion of the Bowker Overpass data that is used for object detection, and Figure 10.19(b) shows the plot of the extracted cross-section.

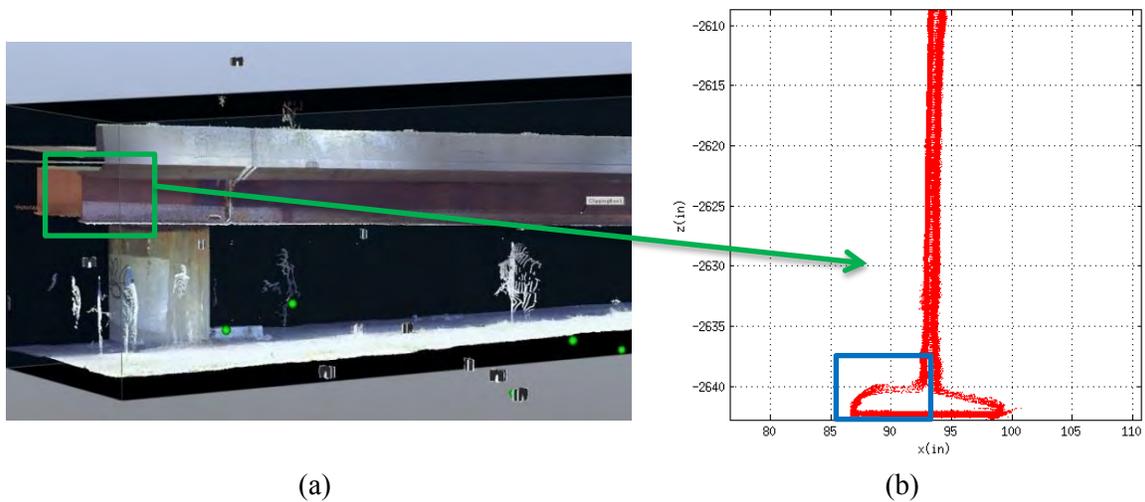


Figure 10.19: (a) Portion of Bowker Overpass used for graph-based object detection, and (b) extracted cross-section.

The object detection is specifically performed to detect the primary steel girder section of the east fascia girder; the results of the object detection are given in Table 10.8. Figure 10.20 shows the dimensions that are provided in the steel manual; the locations of the recorded dimensions can be seen in this figure. Table 10.9 shows the nominal girder dimensions for W36 sections (AISC., 2011). The only available section that has the closest dimensions to the detection results is correctly found to be a W36x135. This was cross-checked with bridge drawings. The differences between the extracted section

dimensions and the dimensions given in the manual are listed in Table 10.8, in terms of percentage error. It should be noted that, the reduction in the web thickness is due to section loss, and the increase in flange width and  $k$  dimension are due to heavy corrosion. At the same time, the laser scanner failed to capture a portion of the cross-section, the part shown with the blue rectangle in Figure 10.19, because it did not have access to the height that would allow complete coverage. However, since the cross-section was assumed to be symmetric, the object detection results were not affected from the incomplete portion of the data. This obtained result showed that even though the member was deteriorated, the detection algorithm was successful in finding the object type.

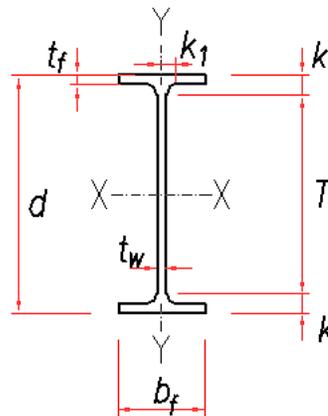


Figure 10.20: Typical W-section dimensions.

Table 10.8: Detection results and actual girder dimensions.

	<b>Detected Section Dimensions</b>	<b>Dimensions of W36x135</b>	<b>Error(%)</b>
$b_f$ (in)	12.12	11.95	1.42
$t_f$ (in)	0.75	0.79	-5.06
$T$ (in)	32.13	32.125	0.02
$k$ (in)	1.99	1.688	17.89

Table 10.9: Nominal girder dimensions for W36 sections.

Shape	Area	Depth	Web	Flange		Distance	
	A	d	Thickness	Width	Thickness	T	k
	(in <sup>2</sup> )	(in)	t <sub>w</sub>	b <sub>f</sub>	t <sub>f</sub>	(in)	(in)
<b>W36x256</b>	75.4	37.43	0.96	12.215	1.73	32.125	2.625
<b>W36x232</b>	68.1	37.12	0.87	12.12	1.57	32.125	2.500
<b>W36x210</b>	61.8	36.69	0.83	12.18	1.36	32.125	2.313
<b>W36x194</b>	57	36.49	0.765	12.115	1.26	32.125	2.188
<b>W36x182</b>	53.6	36.33	0.725	12.075	1.18	32.125	2.125
<b>W36x170</b>	50	36.17	0.68	12.03	1.1	32.125	2.000
<b>W36x160</b>	47	36.01	0.65	12	1.02	32.125	1.938
<b>W36x150</b>	44.2	35.85	0.625	11.975	0.94	32.125	1.875
<b>W36x135</b>	39.7	35.55	0.6	11.95	0.79	32.125	1.688

#### 10.4 Surface Damage Detection

This section consists of the results obtained by using the developed surface damage detection methods described in Chapter 7, including graph-based and surface normal-based damage detection. This section is divided into subsections based on the damage categories that cover all the damage types that are in the scope of the developed damage detection algorithms. The investigated damage categories are described in Section 3.1 in detail; these damage categories and the associated damage types are listed in Table 10.2.

The aim of dividing this section into damage category-based subsections, rather than damage detection strategy-based sub-sections, is to facilitate the condition rating process, since the condition rating of each individual item, for each specific damage type, is determined based on the total amount of damage computed for the damage of interest. The list of the damage types and the associated validation and testing datasets are presented in Table 10.2.

In this chapter, first, the damage detection results for small deformations that include cracks and corrosion are given in Section 10.4.1; this is followed by Section 10.4.2, where the detection results for large deformations with no change in topology, which includes bent members, alignment issues, and points of discontinuity, are presented. Finally, the damage detection results for large deformations with localized change in topology: ruptures, cross-section changes, cross-section loses, and concrete spalling, are included in Section 10.4.3. The investigated datasets are used for both validation and testing purposes; thus, there are two subsections, validation and testing, under Sections 10.4.1-10.4.3.

After laying out the processing results obtained by using datasets listed in Table 10.2, the minimum detectable damage sizes for each damage type is first computed based on the point cloud properties associated with each investigated dataset. Since only three datasets are used for damage detection, for both validation and testing, only the detectable damage sizes associated with these datasets are listed. The details are discussed in Section 10.4.4, and the results are given in Figure 10.44 and Figure 10.45.

#### 10.4.1 Cracks

To perform crack detection, the steps of surface normal-based damage detection, which is explained in Section 7.3, was followed. Further investigation on crack detection, which includes a comprehensive validation application, was carried out, and the results are presented in Chapter 8. The following section, Sections 10.4.1.1.1, discusses the results of the crack detection performed on a portion of Bowker Overpass.

Table 10.10 shows the parameters and corresponding values used for each step in crack detection. The neighborhood size and the intensity threshold are automatically computed based on the local properties of the point cloud. However, a range is defined for neighborhood size to ensure that the developed algorithms detect all the local variations. For patch investigation, first the investigated crack area was divided into 4 in. by 4 in. blocks. Each block was investigated separately to capture the local changes effectively. From each single block, three patches, with a constant patch size of  $8 \cdot k$  points, were extracted. The patch extraction process is explained in 7.3.1 in detail. In this research, for all surface normal-based damage detection applications, a constant angle threshold of 10 degrees is used.

Table 10.10: Parameters and corresponding ranges for crack detection.

<b>Methods</b>	<b>Parameters</b>	<b>Value</b>
Surface Normal Estimation	Neighborhood size, $k$	$> 3$
Surface Patch Investigation	Neighborhood size, $k$	$8 \cdot k$
Normal Comparison	Angle threshold, $T_\alpha$	$10^\circ$
	Intensity threshold, $T_{int}$	$0 < \mu_{int} \pm 2 \cdot \sigma_{int} < 1$

#### 10.4.1.1 Validation

##### 10.4.1.1.1 Bowker Overpass

In order to validate the effectiveness of the developed crack detection algorithms, two sample regions are selected and field measurements are recorded. Figure 10.21 and Figure 10.22 show the images of the investigated regions and the associated field measurements. Figure 10.23 and Figure 10.24 represent the result of the crack detection algorithm. This investigated region is extracted from the bottom deck and it is suffering

from heavy efflorescence. The results show that the developed crack detection algorithms are effective in detecting the cracking location, with or without efflorescence.

Table 10.11 and Table 10.12 give the comparison between the field measurements and computed dimensions. The field measurements were performed by using micrometers. The field measurement were taken from two successive points, at which the slope of the crack changes; these points are labelled with numbers in Figure 10.21 and Figure 10.22. The computed dimensions are extracted by using the same philosophy. For crack length computations, the methodology discussed in Chapter 8 was used; however, in order to get comparable results to the validation set for the longer and more varied cracks seen in the Bowker Overpass, instead of the full length of the crack, the distances between two successive points at the locations of slope changes were recorded.

For the first example, the overall average thickness of the crack is measured as 1 in.. The computations showed that for the detected region the average crack thickness is 0.91 in. and the maximum crack thickness is computed as 1.23 in. The thickness value was not recorded for the second sample patch.

It can be observed from the results given in Table 10.11 and Table 10.12 that the detection results obtained for the first investigated patch are more accurate compared to the second patch. This observed variation in the results is because of the distribution of the intensity values on the surface. For the second patch, most of the undamaged concrete and the crack locations had similar intensity values, which provided an extra challenge for crack detection. Since the intensity values were similar throughout this region, the results of crack detection algorithm primarily depended on the recorded deviations of the

computed surface normals from the reference normal, which was computed by using the undamaged regions of the investigated patch.

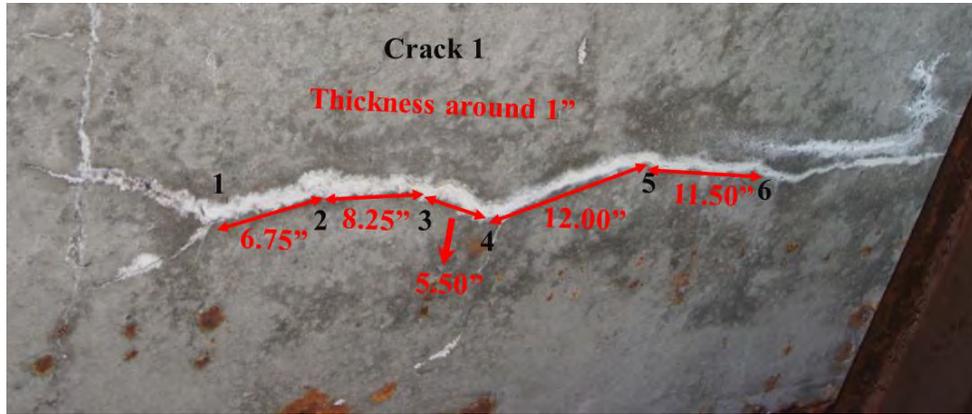


Figure 10.21: Image of the processed portion of the Bowker Overpass for crack detection.

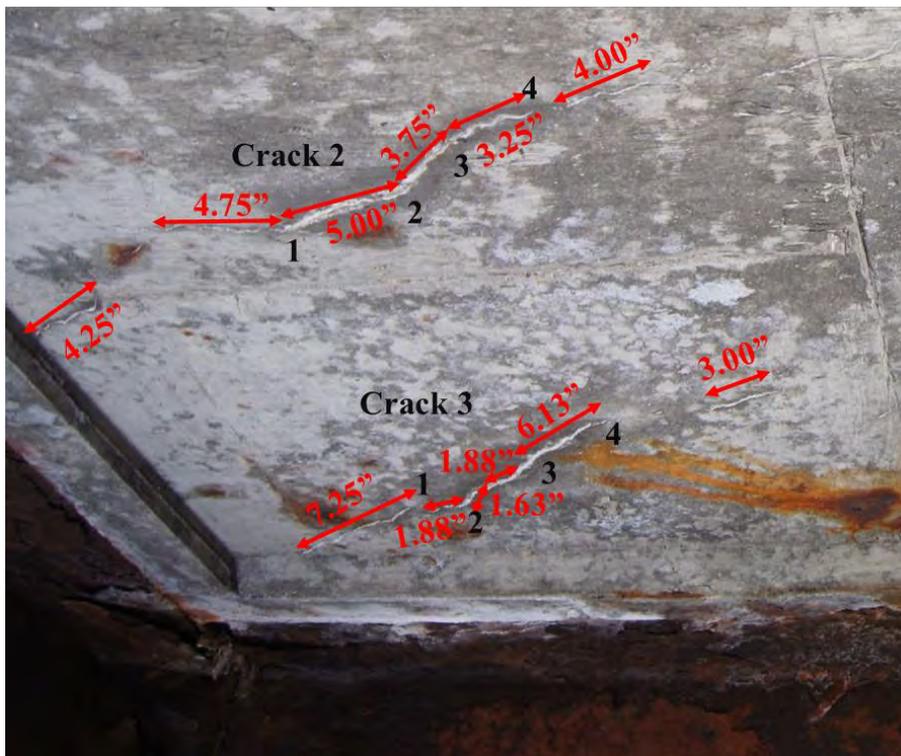


Figure 10.22: Image of the processed portion of the Bowker Overpass for crack detection.

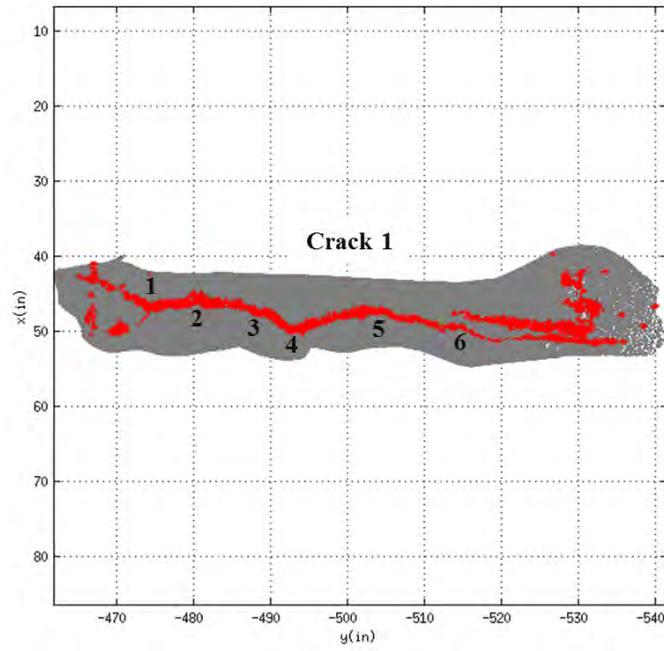


Figure 10.23: Result of the crack detection algorithms for the region shown in Figure 10.21.

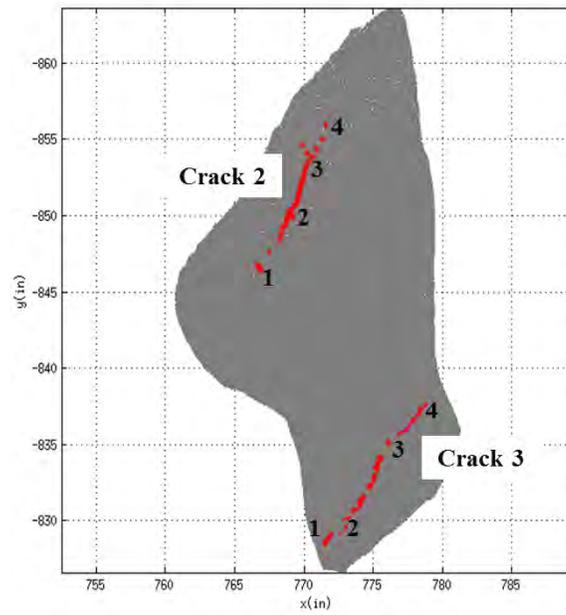


Figure 10.24: Result of the crack detection algorithms for the region shown in Figure 10.22.

Table 10.11: Comparison between the field measurements and computed crack dimensions for patch one.

<b>Defect ID</b>	<b>Measurement Points</b>	<b>Field Measurement (in)</b>	<b>Computed Dimension (in)</b>	<b>Error (%)</b>
<b>Crack 1</b>	<b>1-2</b>	6.75	6.74	0.15
	<b>2-3</b>	8.25	8.28	-0.36
	<b>3-4</b>	5.5	5.52	-0.36
	<b>4-5</b>	12	11.95	0.42
	<b>5-6</b>	11.5	11.47	0.26

Table 10.12: Comparison between the field measurements and computed crack dimensions for patch two.

<b>Defect ID</b>	<b>Measurement Points</b>	<b>Field Measurement (in)</b>	<b>Computed Dimension (in)</b>	<b>Error (%)</b>
<b>Crack 2</b>	<b>1-2</b>	5.00	5.41	-8.20
	<b>2-3</b>	3.75	3.84	-2.40
	<b>3-4</b>	3.25	3.24	0.31
<b>Crack 3</b>	<b>1-2</b>	1.88	2.02	-7.45
	<b>2-3</b>	( 1.63 + 1.88 ) 3.51	3.47	1.14
	<b>3-4</b>	6.13	5.89	3.92

In the latest inspection report (Mercuri and Mirliss, 2013), the areas from which the investigated patches are extracted were reported as “hollow sounding with map cracks with efflorescence”. However, there are not any quantitative crack measurements available for these locations. Laser scanning technology provides the opportunity to record quantitative information on the current condition of structures.

#### 10.4.2 Bent Members/ Alignment Issues / Points of Discontinuity

In order to detect bent members, points of discontinuity, and alignment problems, the first step is to generate the global skeleton of the processed dataset. The steps for obtaining the global skeleton for the processed datasets are the same with as steps described in Section 10.3.2 for performing graph-based object detection. Thus, it will not be repeated in this section. Once the global skeleton was extracted by using the method

described in Section 7.2.2 in detail, the graph-based damage detection was performed to detect bent members, points of discontinuity, and alignment issues. The following sections, Sections 10.4.2.1 and 10.4.2.2, present the results of the graph-based damage detection for bent members, points of discontinuity, and alignment problems, which is performed on the steel testing frame, concrete testing frame and a portion of the DeKalb County Bridge, respectively.

Table 10.13 shows the parameters and corresponding values used for each step towards graph-based damage detection. For skeletonization, a constant height function increment (as discussed in Section 6.3.2),  $h_p$ , of 2 in. is used, and the rest of the voxel dimensions are determined automatically based on the local point densities extracted from the point cloud. Limiting values are defined for the minimum and maximum values of voxel sizes.

Table 10.13: Parameters and corresponding ranges for voxelization/skeletonization performed for graph-based damage detection.

<b>Method</b>	<b>Parameters</b>	<b>Range (or default value)</b>
Voxelization / Skeletonization	Voxel grid dimension, $r_p$	0.03 in. – 0.5 in.
	Height function increment, $h_p$	2 in.

### 10.4.2.1 Validation

#### 10.4.2.1.1 Concrete Testing Frame

The skeleton of the concrete testing frame was already extracted for object detection. The extracted skeleton is shown in both Figure 10.18(a) and Figure 10.25. In this section, this skeleton is used to compute the alignment of each column of the frame. The alignment results for the columns are given in Table 10.14; however, the alignment

results for beams are presented in 10.4.2.2.2 since there isn't any validation measurements recorded for beams. The results are also shown in Figure 10.25; this figure shows the skeleton of the concrete testing frame on which the computed alignment values are placed on the right-hand side of each column. The obtained results for columns represent the horizontal displacement, maximum horizontal drift for each column, for the columns in terms of in/1ft in order to match the results later with condition rating criteria for alignment, which is shown in Table 9.6.

The validation set is composed of horizontal drift measurements taken from exterior columns only, A1-3 and E1-3. The error values are shown in Table 10.14. The largest error in this set is 22.22%; however, it should be noted that even though the percentage error is high, the measured and computed results differ only 0.03 in. These results show that laser scanning technology can be effectively used to compute the reporting values, which required in inspection reports.

Table 10.14: Alignment results for columns of the concrete testing frame.

<b>Member ID</b>	<b>Lab Measurement (in)</b>	<b>Computed Alignment (in/1 ft)</b>	<b>Error (%)</b>
<b>A1</b>	2.01	2.07	2.99
<b>A2</b>	0.32	0.34	6.25
<b>A3</b>	0.10	0.07	22.22
<b>B1</b>	–	2.20	–
<b>B2</b>	–	0.28	–
<b>B3</b>	–	0.16	–
<b>C1</b>	–	–	–
<b>C2</b>	–	0.20	–
<b>C3</b>	–	0.26	–
<b>D1</b>	–	1.84	–
<b>D2</b>	–	0.25	–
<b>D3</b>	–	0.78	–
<b>E1</b>	1.82	1.81	0.55
<b>E2</b>	0.28	0.25	10.71
<b>E3</b>	0.76	0.56	9.68

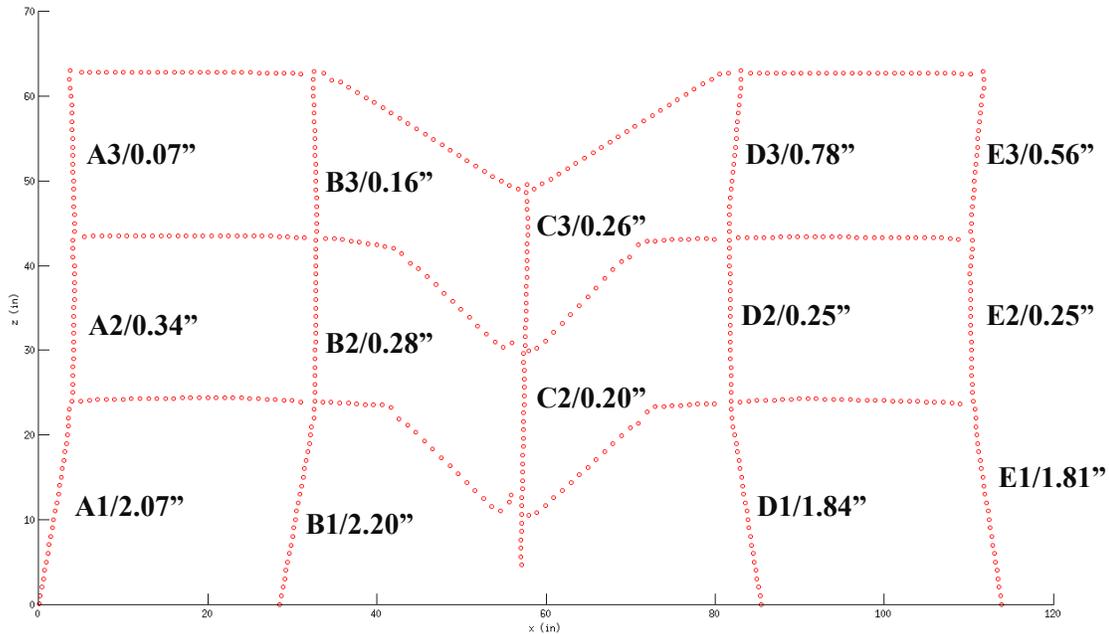


Figure 10.25: Skeleton of the concrete testing frame with computed column alignment values.

#### 10.4.2.2 Testing

##### 10.4.2.2.1 Steel Testing Frame

The skeleton of the steel testing frame was already extracted for object detection; this skeleton is used for computing the alignment of each member. Table 10.15 presents computed alignment values. The computed alignment values are local, meaning only the changes that occur within the investigated member were recorded. However, translations of the entire system and/or combination of several members together were not investigated. These results show that the steel testing frame is in an excellent condition in terms of alignment.

Table 10.15: Alignment values computed for the steel testing frame.

<b>Member ID</b>	<b>Computed Alignment (in/1 ft)</b>
<b>Member 1</b>	0.09
<b>Member 2</b>	0.11
<b>Member 3</b>	0.09

#### 10.4.2.2.2 Concrete Testing Frame

As mentioned before, the skeleton of the concrete testing frame was extracted for object detection. The extracted skeleton is shown in Figure 10.18(a) and Figure 10.26. In this section, this skeleton is used to compute the alignment of each beam of the frame. The alignment results for the beams are given in Table 10.16; the results for the columns are already discussed in Section 10.4.2.1.1. The obtained results represent the vertical sag, the largest vertical displacement on each beams, for beams in terms of in/1ft in order to match the results later with condition rating criteria for alignment, which is shown in Table 9.6.

Table 10.16: Alignment results for beams of the concrete testing frame.

<b>Member ID</b>	<b>Computed Alignment (in/1 ft)</b>
<b>AB1</b>	0.06
<b>AB2</b>	0.07
<b>AB3</b>	0.09
<b>BC1</b>	5.02
<b>BC2</b>	5.31
<b>BC3</b>	5.85
<b>CD1</b>	4.31
<b>CD2</b>	4.81
<b>CD3</b>	5.87
<b>DE1</b>	0.09
<b>DE2</b>	0.09
<b>DE3</b>	0.11

Table 10.17 shows the result of the detected points of discontinuity locations, where significant defects are followed by abrupt change in the overall alignment of the object. The skeleton points are used to record the locations of the points of discontinuity; the closest skeleton point to the discontinuity location is recorded for each detection location. The results indicate that several members of the concrete testing frame have significant points of discontinuity. The discontinuity locations were defined at points where there was either an abrupt change in the slope of a member along with detected damage or complete separation. This application requires prior knowledge on the orientations of the objects; for this case, the expected initial orientation of the columns was taken as vertical, and for the beams it was taken as horizontal.

The computed alignment results are also shown in Figure 10.26; this figure shows the skeleton of the concrete testing frame on which the computed alignment values are placed at the bottom of each column, and the points of discontinuity are marked with black circles. The obtained results are then used for computing alignment-based condition rating for each member of the concrete testing frame. These condition ratings are determined based on the condition rating criteria given in Table 9.6.

Table 10.17: Points of discontinuity locations for the concrete testing frame.

<b>Member ID</b>	<b>Points of Discontinuity</b>	<b>x(in)</b>	<b>z(in)</b>
<b>1</b>	BC3	33.70	62.71
<b>2</b>	BC3	56.70	49.09
<b>3</b>	BC2	41.95	42.05
<b>4</b>	BC2	55.95	30.85
<b>5</b>	BC1	41.61	23.24
<b>6</b>	BC1	54.61	11.00
<b>7</b>	CD3	58.55	49.09
<b>8</b>	CD3	80.55	62.60
<b>9</b>	CD2	57.92	29.94
<b>10</b>	CD2	69.92	41.06
<b>11</b>	CD1	57.94	10.50
<b>12</b>	CD1	72.94	23.34

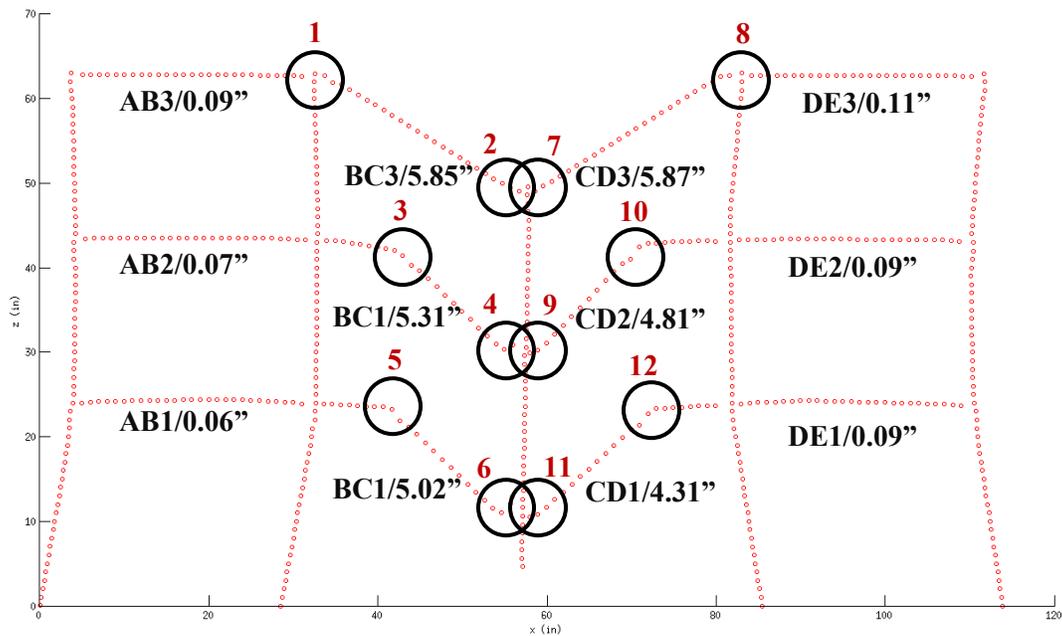


Figure 10.26: Skeleton of the concrete testing frame with computed column alignment values and points of discontinuity.

#### 10.4.2.2.3 DeKalb County Bridge

The alignment values for the timber piles detected from the processed portion of the DeKalb County Bridge, which are shown in Figure 10.14, are also computed by using their individual skeletons. The results of the alignment computation are given in Table 10.18. The obtained results represent the horizontal displacement, maximum horizontal drift for each pile, for the piles in terms of in/1ft in order to match the results later with condition rating criteria for alignment, which is shown in Table 9.6. Since the timber piles were attached to the pier cap, they were each misaligned by approximately the same amount. Thus, the alignment values computed for these piles are very close to each other. The alignment-based condition rating criteria for each pile is then determined by using the Table 9.6.

Table 10.18: Alignment results for detected timber piles of the DeKalb County Bridge.

<b>Member ID</b>	<b>Alignment (in/1 ft)</b>
<b>Pier 1</b>	1.38
<b>Pier 2</b>	1.38
<b>Pier 3</b>	1.41
<b>Pier 4</b>	1.34
<b>Pier 5</b>	1.41
<b>Pier 6</b>	1.33
<b>Pier 7</b>	1.36
<b>Pier 8</b>	1.38

#### 10.4.3 Rupture / Cross-section Change / Spalling / Steel Delamination

In order to detect the major changes in the local geometry, including rupture, cross-section change, concrete spalling, and steel delamination, two methods are used: graph-based damage detection, Section 7.2, and surface-based damage detection, Section 7.3. For the cases, when the object information is already known, the graph-based

damage detection method is a good option for finding the damage locations, and it is also used for quantifying the detected damage. On the other hand, surface-normal based damage detection methodology does not require a pre-knowledge on the object type. If available, object properties could be used to compute the reference normals as described in Section 7.3. These two damage detection methodologies are especially important since the current practice lacks an efficient system, which can record the area and/or volume of the major changes in the local geometry. Thus, quantifying the damage in an effective way, which was not possible before, plays a key role in future strategies that will integrate the laser scanning technology with visual inspections.

The following sections, Sections 10.4.3.1 and 10.4.3.2, lay out the damage detection results, obtained by using the methods listed in the previous paragraph, for first the validation sets and then the testing sets. Some portions of the Bowker Overpass data are used for validating the following damage types: cross-section loss, concrete spalling, and steel delamination. The rest of the datasets, including the concrete frame (concrete spalling) and DeKalb County Bridge (concrete spalling, cross-section change), are used for testing. A portion of the concrete testing frame was already used for validation in Chapter 8; in this Chapter it will only be used for testing. The damage types that are included in this section can be listed as section losses, concrete spalling, delamination, and cross-section changes.

Table 10.13 shows the parameters and corresponding values used for graph-based damage detection. For skeletonization, a constant height function increment (as discussed in Section 6.3.2),  $h_p$ , of 2in. is used, and the rest of the voxel dimensions are determined automatically based on the local point densities extracted from the point cloud. Limiting

values, which are defined for the minimum and maximum values of voxel sizes, are kept the same, Table 10.13.

Table 10.10 shows the parameters and corresponding values used for each step towards surface-based damage detection. The neighborhood size, Section 7.3.4, and the intensity threshold are automatically computed based on the local properties of the point cloud. For patch investigation, again the investigated crack area was divided into 4 in. by 4 in. blocks. Each block was investigated separately to capture the local changes effectively. From each single block, three patches, with a constant patch size of  $8 \cdot k$  points, were extracted. The patch extraction process is explained in detail in Section 7.3.1. In this research, for all surface normal-based damage detection applications a constant angle threshold, which is 10 degrees, is used.

Table 10.19: Parameters and corresponding ranges for surface-based damage detection.

<b>Methods</b>	<b>Parameters</b>	<b>Value</b>
Surface Normal Estimation	Neighborhood size, $k$	$>3$
Surface Patch Investigation	Neighborhood size, $k$	$8 \cdot k$
Normal Comparison	Angle threshold, $T_\alpha$	$10^\circ$
	Intensity threshold, $T_{int}$	$0 < \mu_{int} \pm 2 \cdot \sigma_{int} < 1$

### 10.4.3.1 Validation

#### 10.4.3.1.1 Bowker Overpass

The validation set used in this section was extracted from the Bowker Overpass. Since this dataset presented field conditions, it was important to process the Bowker Overpass data for validation. The obtained results have showed that the developed damage detection algorithms are effective in detecting several damage types on field

datasets. The previous validation application performed on the concrete testing frame is described in Chapter 8.

Several examples for the following damage types are investigated in this section: section losses, concrete spalling (with and without corrosion), and steel delamination. This discussion will continue with the results obtained for two sample regions, for which the cross-section loss is detected, and associated dimensions are computed.

Both of the section loss regions, which include defects designated below as S1 and S2, are extracted from Span 8, on the east face of the beam at Pier J shown in Figure 10.12. The image of the first region is shown in Figure 10.27, and the associated damage detection results for S1 are given in Figure 10.28. Figure 10.28(a) shows the overall results, the red color indicates the defect locations. Since this investigated portion of the beam is completely damaged, due to section loss and corrosion, the entire patch was detected as damaged, and it was labeled with red points.

The results of the validation study are included in Table 10.20. The measured dimensions were compared with the computed values from the scan. The results show that the laser scanners can be used effectively for dimension extraction for section losses.

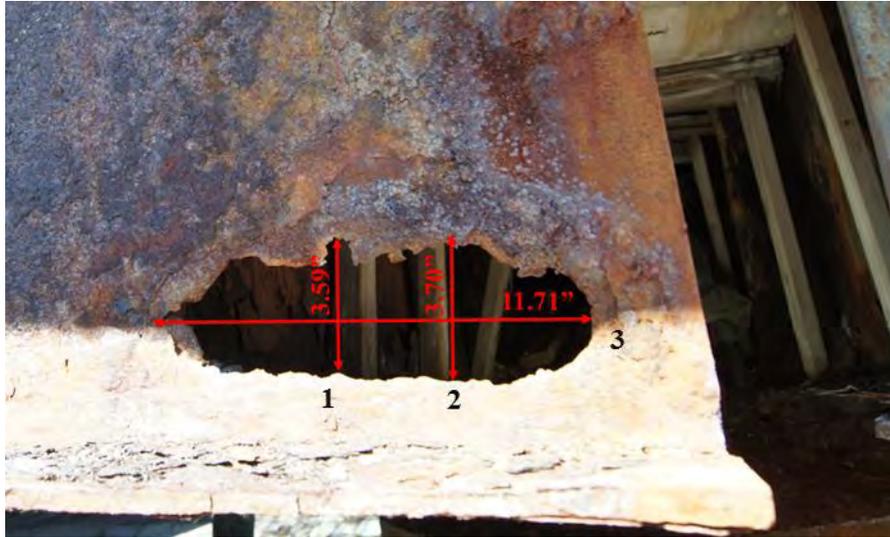


Figure 10.27: Image of first section loss region, S1.

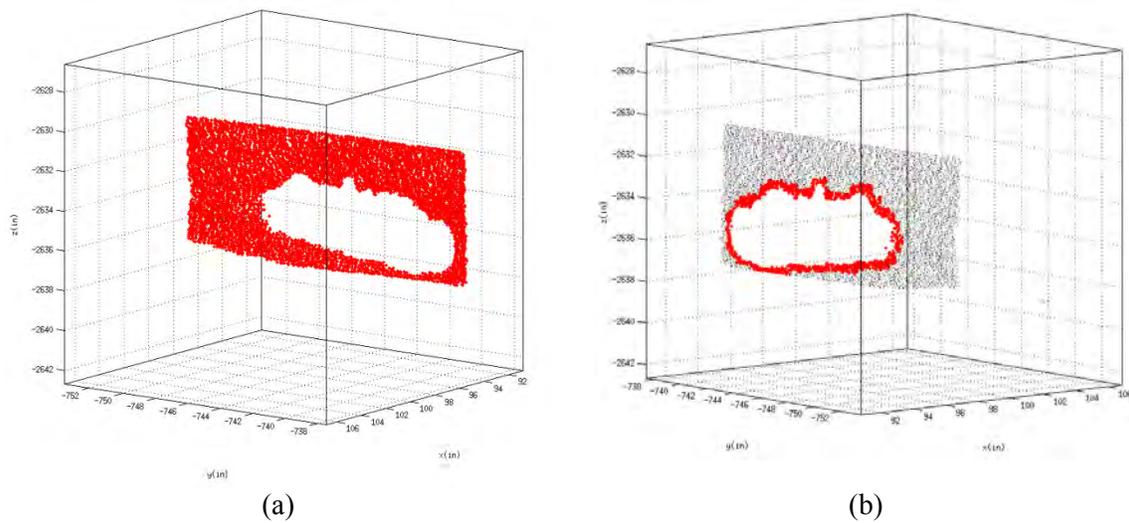


Figure 10.28: Results for (a) overall damage detection results and (b) section loss.

Table 10.20: Comparison between the field measurements and computed section loss dimensions for S1.

<b>Defect ID</b>	<b>Measurement Number</b>	<b>Field Measurement (in)</b>	<b>Computed Dimension (in)</b>	<b>Error (%)</b>
<b>S1</b>	<b>1</b>	3.59	3.55	1.11
	<b>2</b>	3.70	3.64	1.62
	<b>3</b>	11.71	11.76	-0.43

Damage region S2, which is located on the second section loss region, is shown in Figure 10.29. The damage detection results for S2 are shown in Figure 10.30. Figure 10.30(a) presents the overall results, and again the red color indicates the defect locations. The beam from which this region was taken from is entirely damaged; thus the complete patch was detected as damaged.

The validation results are listed in Table 10.21. The measured dimensions were again compared with the computed values. This time the percentage errors of the measurements were found to be relatively higher than the previous case. However, this does not necessarily indicate that the damage detection was not effective for this example. This difference might also be result of a hand-measurement error, i.e., the precise location of the hand measurement may be different than the maximum distance location computed by the damage detection algorithm. In order to check this, the inspection report for the processed portion of the Bowker Overpass was used. The dimensions given in in the inspection report are shown in Figure 10.31. It can be seen from the results given in Table 10.22 that the dimensions listed in the inspection report are closer to the computation results. In general, the computed dimensions are either smaller than the ones recorded in the inspection report, or they are nearly the same. The dimensions recorded in the inspection report are found to be conservative.



Figure 10.29: Image of second section loss region, S2.

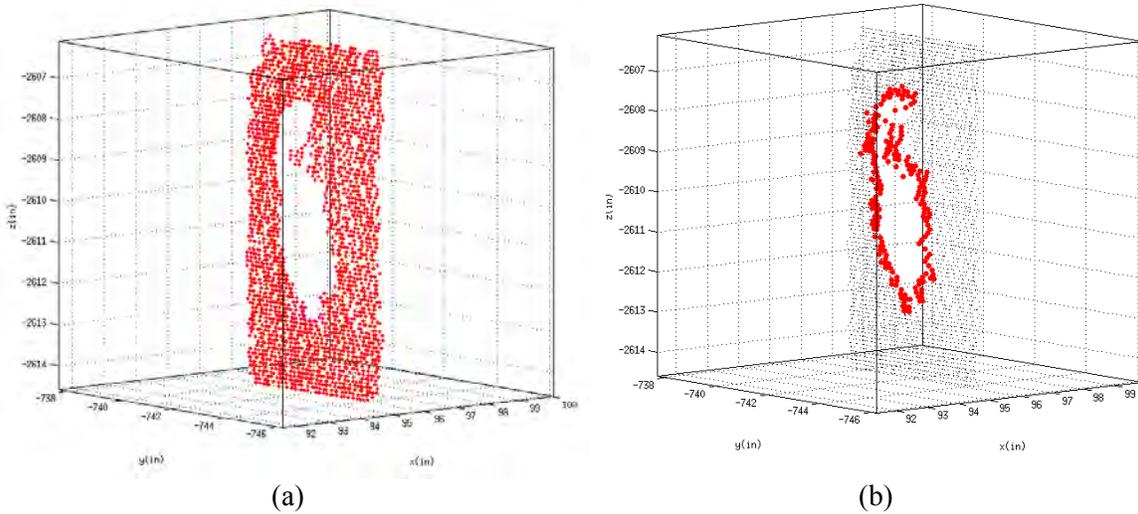


Figure 10.30: Results for (a) overall damage detection and (b) section loss for S2.

Table 10.21: Comparison between the field measurements and computed section loss dimensions for S2.

<b>Defect ID</b>	<b>Measurement Number</b>	<b>Field Measurement (in)</b>	<b>Computed Dimension (in)</b>	<b>Error (%)</b>
<b>S2</b>	<b>1</b>	5.07	5.34	-5.33
	<b>2</b>	1.79	2.06	-15.08

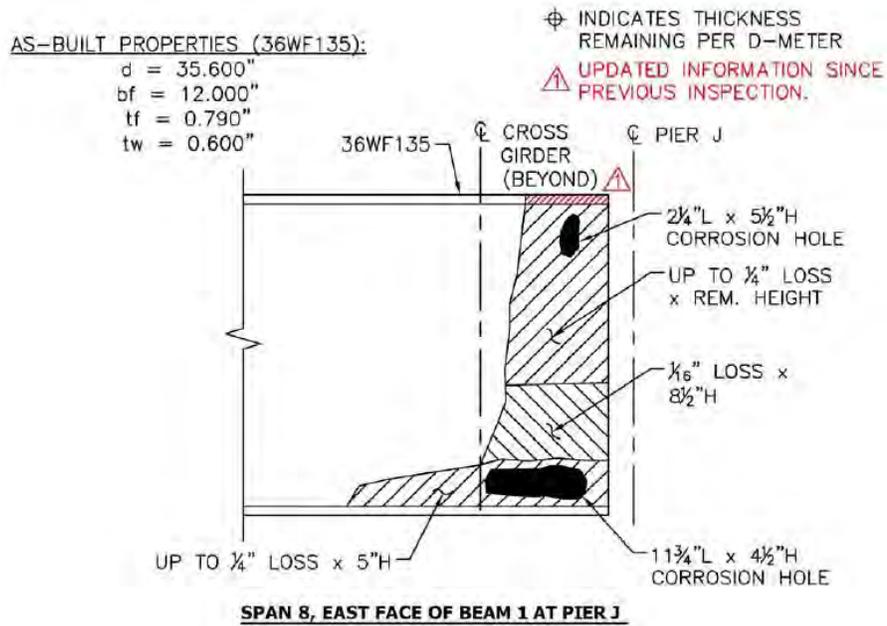


Figure 10.31: Locations for the extracted section loss patches (Mercuri and Mirliss, 2013).

Table 10.22: Comparison between the inspection report values and computed section loss dimensions.

<b>Defect ID</b>	<b>Measurement Description</b>	<b>Inspection Report (in)</b>	<b>Computed Dimension (in)</b>	<b>Error (%)</b>
<b>S1</b>	<b>Width</b>	4.50	3.84	14.67
	<b>Length</b>	11.75	11.76	-0.09
<b>S2</b>	<b>Length</b>	2.25	2.06	8.44
	<b>Height</b>	5.50	5.34	2.91

With respect to concrete spalling, the damage detections results for three regions will be described in this section. For these cases, damage detection was carried out by using the surface normal-based damage detection strategies given in 7.3. The parameters required for this damage detection strategy, which are listed in Table 10.19, were selected automatically by using the local point cloud properties. The images of the concrete spalling regions, which include the defects CS1, CS2, and CS3 respectively, are given below in Figure 10.32, Figure 10.34, and Figure 10.36. The results obtained through

surface-based damage detection are shown in Figure 10.33, Figure 10.35, and Figure 10.37.

The comparison results for the validation set are given in Table 10.23. The obtained results showed that the errors associated with the described comparisons are varying between 0% and 13%. It can be observed that as the validation dimension decreases the accuracy of the computed distance reduces. For the small dimensions, even the smallest deviation had a significant effect on the percentage error computations.

For CS3, along with the concrete spalling, the corroded area was also differentiated; this region is showed in blue in Figure 10.37. This was obtained by using the intensity thresholding, which is explained in Section 7.3.3. The changes in the normal orientations combined with the surface intensity values were used to detect the defected areas.

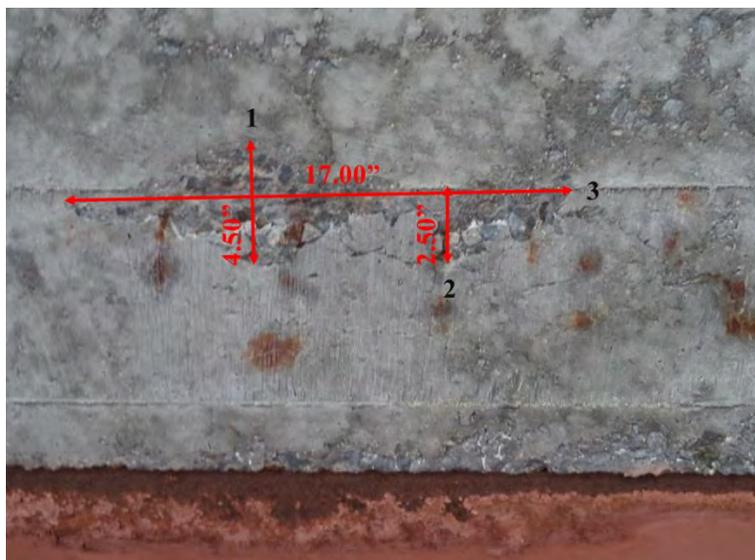


Figure 10.32: Image of CS1.

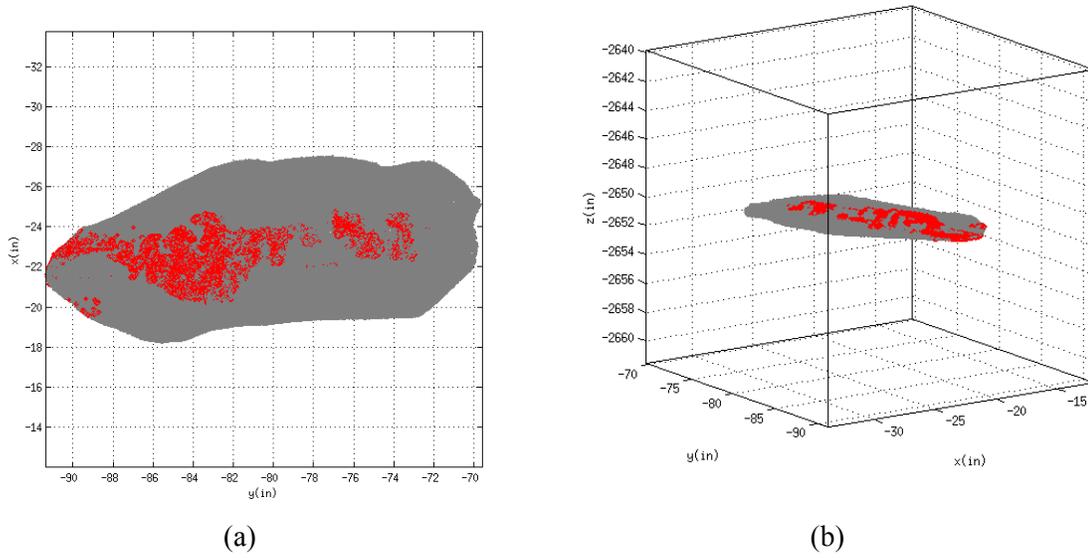


Figure 10.33: Damage Detection results for CS1: (a) surface representation and (b) 3D representation.

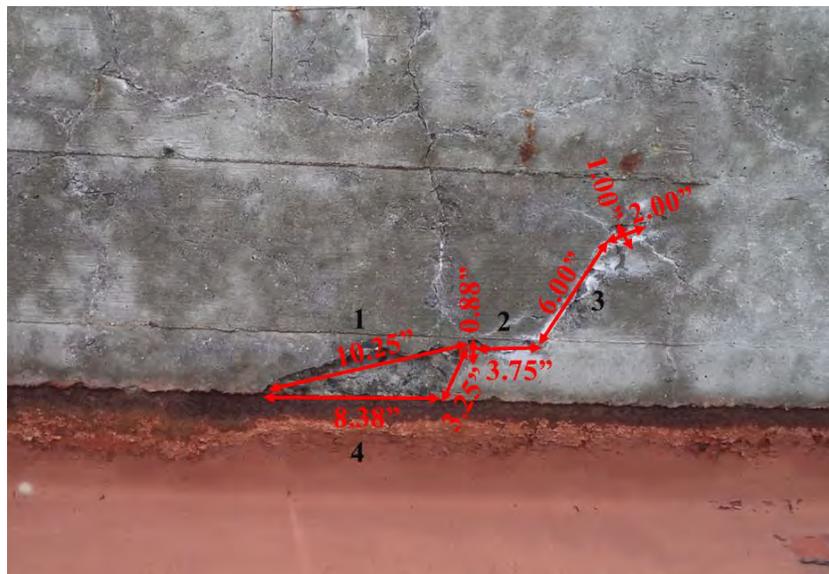
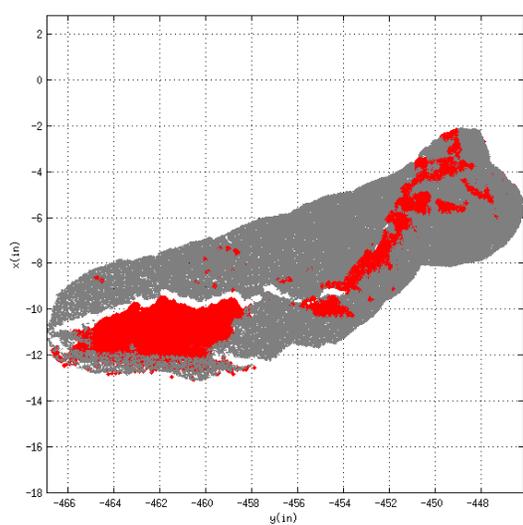
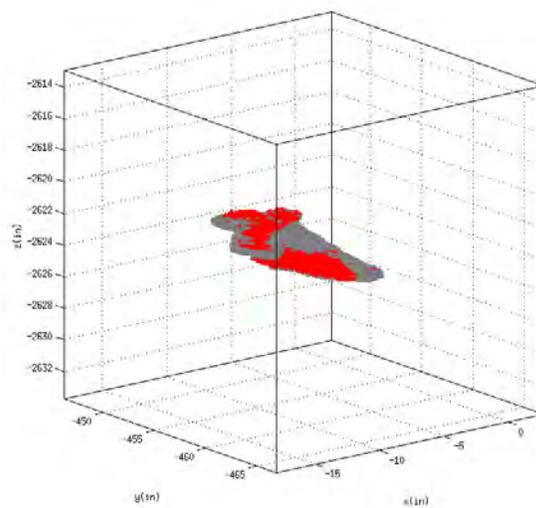


Figure 10.34: Image of CS2.



(a)



(b)

Figure 10.35: Damage Detection results for CS2: (a) surface representation and (b) 3D representation.

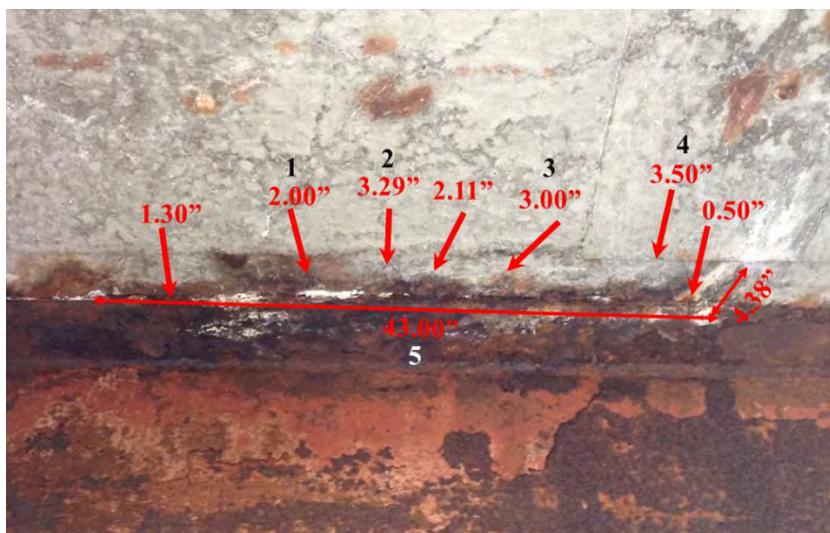


Figure 10.36: Image of CS3.

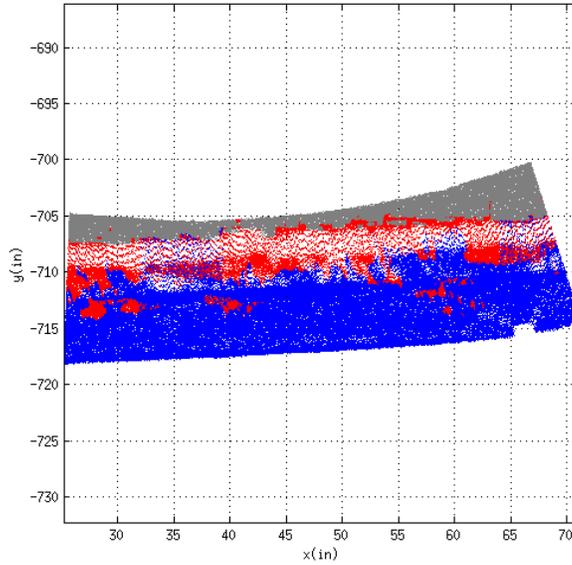


Figure 10.37: Damage Detection results for CS3.

Table 10.23: Comparison between the field measurements and computed concrete spalling for investigated patches.

<b>Defect ID</b>	<b>Measurement Number</b>	<b>Field Measurement (in)</b>	<b>Computed Dimension (in)</b>	<b>Error (%)</b>
<b>CS1</b>	<b>1</b>	4.50	4.36	3.11
	<b>2</b>	17.00	17.85	-5.00
	<b>3</b>	2.50	2.18	12.80
<b>CS2</b>	<b>1</b>	10.25	10.02	2.24
	<b>2</b>	3.75	3.95	-5.33
	<b>3</b>	6.00	6.66	-11.00
	<b>4</b>	8.38	8.36	0.24
<b>CS3</b>	<b>1</b>	2.00	2.11	-5.50
	<b>2</b>	3.29	3.26	0.91
	<b>3</b>	3.00	3.24	-8.00
	<b>4</b>	3.50	3.21	8.29
	<b>5</b>	43.00	42.65	0.81

Finally, the last damage type included in the validation section is delamination due to heavy corrosion. The Bowker Overpass is suffering from heavily corroded steel members. Corrosion has mild effects on some regions; however, for other locations, such as the regions chosen for this study, the results are significant. In this section, two

delamination locations are processed. The aim was to locate the delamination automatically and to extract the dimensions automatically. The images given in Figure 10.38 and Figure 10.39 show the regions used for processing, which includes the delamination locations DL1 and DL2.

In order to perform this detection, first the object is determined, which was done in Section 10.1.5, and then the surface orientation information was used to perform surface normal-based damage detection. Two cases were investigated, DL1 and DL2, in order to extract useful defect information. The results are shown in Figure 10.39 and Figure 10.40. The validation results are presented in Table 10.24.

It was possible to extract validation measurements; however, due to the location of the laser scanner, which was beneath the delamination, the full scope of the detected region obtained through the performed defect detection did not completely represent the actual results. This is due to the fact that the scanner was not able to see the top side of the bottom flange of the girder. Thus, it is evident that in order to obtain complete damage detection results, it is required to cover all the surfaces without any registration problems.



Figure 10.38: Image of DL1

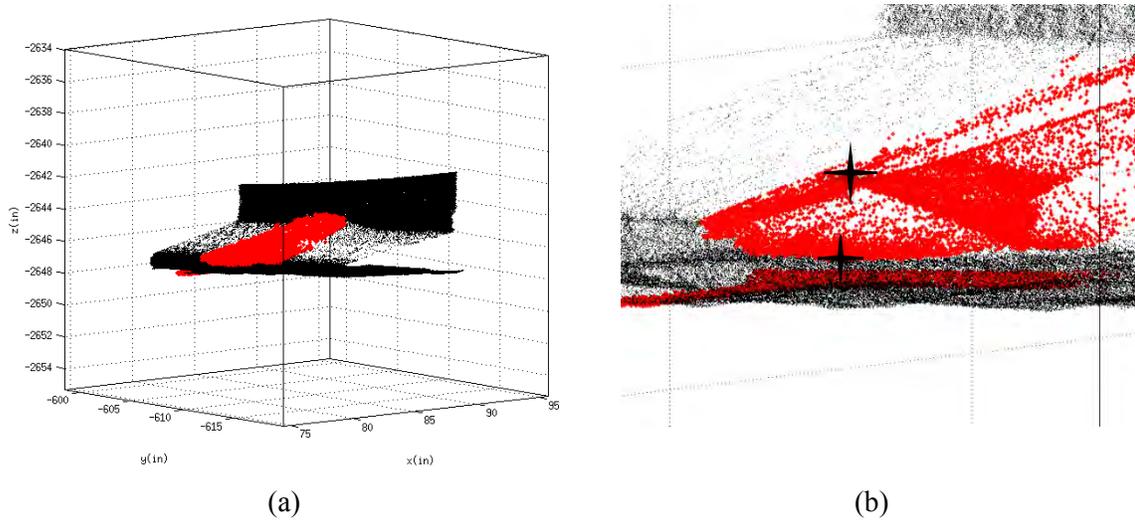


Figure 10.39: Damage Detection results for DL1: (a) 3D representation and (b) close-up view with marked interest locations.



Figure 10.40: Image of DL2.

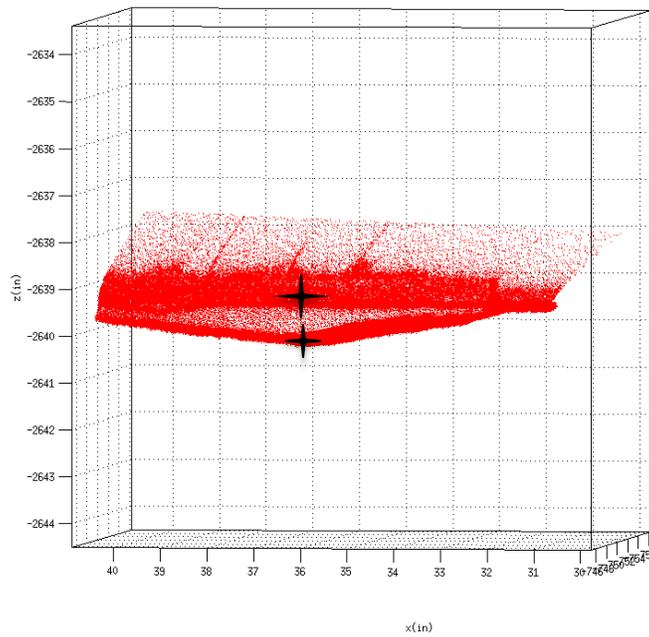


Figure 10.41: Damage Detection results for DL2: (b) 3D representation and (b) close-up view with marked interest locations.

Table 10.24: Comparison between the field measurements and computed delamination heights for investigated patches.

<b>Defect ID</b>	<b>Measurement Number</b>	<b>Field Measurement (in)</b>	<b>Computed Dimension (in)</b>	<b>Error (%)</b>
<b>DL1</b>	<b>1</b>	1.73	1.72	0.58
<b>DL2</b>	<b>1</b>	1.68	1.71	-1.79

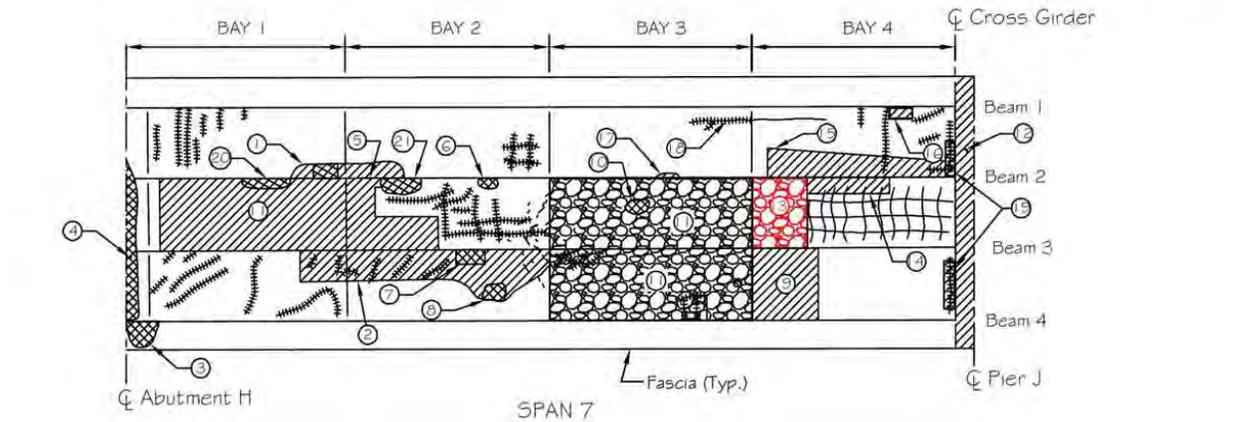
Even though, linear dimensions were extracted in order to compare the detection results with the validation sets. The developed damage detection algorithms are capable of computing the area and volume values associated with the detected damages. The details of the damage quantification methodologies are discussed in Chapters 7 and 8. For the presented defects, if applicable, the area and volume value of the detected damage is computed. The results are given in Table 10.25.

Table 10.25: Computed area and volume results for the detected damage regions.

<b>Defect ID</b>	<b>Damage Area (in<sup>2</sup>)</b>	<b>Damage Volume (in<sup>3</sup>)</b>
<b>SL1</b>	32.28	—
<b>SL2</b>	6.82	—
<b>CS1</b>	42.11	22.81
<b>CS2</b>	19.69	13.75
<b>CS3</b>	84.93	15.21
<b>DL1</b>	—	—
<b>DL2</b>	—	—

Finally, Figure 10.42 is used to illustrate the current practice of visual inspections, which consists sketching the recorded damage locations on the plan of the investigated structure. This sketch is from a sample inspection report on Bowker Overpass (Mercuri and Mirliss, 2013). As mentioned, the current practice involves presenting the inspection recordings in sketches; however these sketches do not include quantitative information on most of the entities shown, such as the cracks, spalled concrete regions, and etc. Laser scanning technology can significantly improve the current techniques used for recording

the visual inspections since this technology provides an opportunity to record and present the exact defect locations and quantitative information associated with them in a 3D environment. By using this technology, a semi-automated system can be developed in order to assist the inspectors to facilitate this data documentation process.



**LEGEND**

- Hairline crack (unless otherwise noted)
- Hairline crack with efflorescence
- Spall
- Hollow area
- Updated information from previous inspection
- Incipient spall

**CONDITIONS**

1. 10'-0" long x up to 2'-4" wide hollow area with 2'-0" long x 11" wide x 1 3/4" deep spall with exposed reinforcement.
2. 26'-0" long x up to 4'-6" wide hollow area.
3. Up to 2'-5" long x up to 2'-2" wide x up to 11" high corner spall with exposed and debonded transverse (main) reinforcement and efflorescence.
4. Up to 1'-0" long x 16'-0" wide x 4 1/2" deep spall with exposed reinforcement and efflorescence along joint.
5. 6'-0" long x up to full width hollow area.
6. 10" long x 1'-0 1/2" wide x 3" deep spall with exposed reinforcement and efflorescence.
7. 1'-3" long x 10" wide x 3 1/2" deep spall with exposed reinforcement.
8. 10" long x 1'-0" wide x 2" deep spall with exposed reinforcement.
9. 7'-0" long x full width hollow area.

10. 1'-0" diameter x up to 3 1/2" deep spall with exposed reinforcement.
11. Entire area is hollow sounding with map cracks with efflorescence
12. Timber formwork between cross girders (underside hidden).
13. 5'-0" long x full width incipient spall,
14. 6'-6" long x 1'-9" wide hollow area with spalling up to 3 1/4" long x 1'-0" wide 1 1/4" deep.
15. 12'-6" long x up to 3'-0" wide hollow area.
16. 2'-6" long x 10" wide hollow area.
17. 1'-9" long x 10" wide hollow area.
18. Hairline crack with heavy efflorescence and stalactites.
19. Efflorescence, rust staining and hollow area at edge of cross girder.
20. 6'-0" long x 1'-6" wide x 3" deep spall with exposed reinforcement.
21. 1'-3" long x 1'-6" wide x 3" deep spall with exposed reinforcement.

Figure 10.42: Span 7, underside of deck condition plan (Mercuri and Mirliss, 2013)

### *10.4.3.2 Testing*

In the previous section, the validation process for the developed damage detection algorithms is described. Several portions of the Bowker Overpass along with the measured dimensions are used in order to verify the accuracy of the developed damage detection algorithms. In this section, the rest of the datasets, which were listed in Table 10.2, are used to test the algorithms to show that these methods are applicable to a variety of structures that consists different structural components.

#### *10.4.3.2.1 Concrete Testing Frame*

The concrete testing frame was used for both crack and spalling dimension validation in Chapter 8. A total of 21 cracks and 20 spalled concrete regions were used for validation. The results showed that the proposed methods are effective in both detecting and quantifying the existing damage on structures.

Chapter 8 was focusing on locating and quantifying individual damaged regions; however, it is required to compute the total damage on a structural item in order to assign a condition rating based on the criteria tables listed in Chapter 9. Thus, this section is dedicated to computing the overall damage on the individual components of the concrete testing frame; the results are shown in Figure 10.28 and Table 10.27, for columns and beams separately. Column and beam locations can be found in Figure 10.3.

Both graph-based damage detection and surface based damaged detection methods were used to compute the total damage on individual members. For this frame, the priority was given to the graph-based damage detection; the surface normal-based damage detection was only used at the points of discontinuity, where the graph-based damage detection method sometimes fails to compute the volumetric changes accurately.

Since exactly the same parameters and their associated values/ranges from the previous sections, for both graph-based and surface-based damage detection, are used for this application, no further discussion is included. The spalled concrete regions and the cracking locations for the concrete testing frame are shown in Appendix C.

The obtained results are later used for determining the condition rating for each individual member of the concrete testing frame. These condition ratings, which are associated with concrete spalling and cracking, are assigned based on the criteria given in Table 9.9.

Table 10.26: Computed area and volume results for the detected damage regions on the columns of the concrete testing frame.

	<b>Damage Area (in<sup>2</sup>)</b>	<b>Damage Volume (in<sup>3</sup>)</b>
<b>A1</b>	4.00	0.95
<b>A2</b>	<i>No spalling</i>	<i>No spalling</i>
<b>A3</b>	<i>No spalling</i>	<i>No spalling</i>
<b>B1</b>	2.77	0.50
<b>B2</b>	<i>No spalling</i>	<i>No spalling</i>
<b>B3</b>	<i>No spalling</i>	<i>No spalling</i>
<b>C1</b>	<i>No column</i>	<i>No column</i>
<b>C2</b>	1.32	0.28
<b>C3</b>	2.13	0.49
<b>D1</b>	1.36	0.30
<b>D2</b>	<i>No spalling</i>	<i>No spalling</i>
<b>D3</b>	<i>No spalling</i>	<i>No spalling</i>
<b>E1</b>	3.18	0.57
<b>E2</b>	2.38	0.50
<b>E3</b>	<i>No spalling</i>	<i>No spalling</i>

Table 10.27: Computed area and volume results for the detected damage regions on the beams of the concrete testing frame.

	<b>Damage Area (in<sup>2</sup>)</b>	<b>Damage Volume (in<sup>3</sup>)</b>
<b>AB1</b>	<i>No spalling</i>	<i>No spalling</i>
<b>AB2</b>	<i>No spalling</i>	<i>No spalling</i>
<b>AB3</b>	<i>No spalling</i>	<i>No spalling</i>
<b>BC1</b>	20.14	4.43
<b>BC2</b>	21.24	4.80
<b>BC3</b>	5.91	1.36
<b>CD1</b>	23.99	5.32
<b>CD2</b>	6.34	1.39
<b>CD3</b>	5.32	1.22
<b>DE1</b>	<i>No spalling</i>	<i>No spalling</i>
<b>DE2</b>	<i>No spalling</i>	<i>No spalling</i>
<b>DE3</b>	<i>No spalling</i>	<i>No spalling</i>

#### 10.4.3.2.2 DeKalb County Bridge

Finally, the last dataset on which the developed damage detection strategies were performed is the DeKalb County Bridge. The region, which is used for damage detection, is shown in Figure 10.43 with a red rectangle. The aim was again to compute the overall damage on individual members. Similar to the previous section, both graph-based damage detection and surface based damaged detection methods were used to compute the total damage on individual members. For this application, the graph-based damage detection was used for timber piles and the pier cap, whereas the surface normal-based damage detection was used for the concrete deck. The results of the performed damage detection are shown in Table 10.28. The parameters (and their corresponding values) used for both graph-based and surface-based damage detection are the same with the previous sections. Thus, an explanation for these parameters is not included in this section.

The obtained results are later used for determining the condition rating for each investigate member of the DeKalb County Bridge. The condition ratings, which are associated with concrete spalling and cracking, are assigned based on the criteria given in Table 9.9; and the condition ratings associated with timber piers are determined based on the condition rating criteria given in Table 9.12.



Figure 10.43: Point cloud of DeKalb County Bridge.

Table 10.28: The results of the surface damage detection algorithms for individual items.

	<b>Damage Area (in<sup>2</sup>)</b>	<b>Damage Volume (in<sup>3</sup>)</b>
<b>Deck</b>	947.52	1326.53
<b>Pier cap</b>	1.41	0.28
<b>Pier 1</b>	4.96	1.95
<b>Pier 2</b>	4.65	1.83
<b>Pier 3</b>	7.75	3.05
<b>Pier 4</b>	31.00	61.02
<b>Pier 5</b>	3.72	0.73
<b>Pier 6</b>	1.86	0.37
<b>Pier 7</b>	3.10	1.22
<b>Pier 8</b>	0.31	0.02

#### 10.4.4 Limitations for Surface Damage Detection

Point clouds provide complete surface coverage for an investigated structure; however, due to noise (and several other external and internal error sources) individual points do not carry all the underlying surface properties. Thus, a set of neighborhood points  $k$  are selected, for most of the developed detection algorithms, such that this small set of points would be representative enough to capture these underlying surface properties.

The process of determination of  $k$  for different point clouds, for surface-normal based damage detection, is already discussed in Section 7.3.4. Once the neighborhood size is computed, the area of the patch represented with these neighborhood points is used for extracting the dimensions of the minimum detectable damage size.

Since the voxel models of point clouds are aimed to cover the entire surface of an investigated structure, the utilized voxel sizes are selected based on the surface point density. Unlike the neighborhood size approach explained above, the main goal of the voxel models is to generate a solid surface representation. Thus, the surface patch investigation methodology described in Section 7.2.1 is applied to compute the voxel dimension. This generated voxel model is later used to extract the skeleton of the investigated point cluster; thus, the voxel size poses a limiting value for the defects that are located by using skeletons.

Finally, for cracks, the goal is to capture small local variations rather than capturing the local properties of the investigated patch. Thus, the limitations, which are determined for the surface-normal based damage detection that do not use color information, are not valid for this case. Instead, the same methodology used for

computing the limiting voxel dimensions is used to compute the limiting crack dimensions since the computed voxel sizes also represents the maximum distance between two surface points. This maximum distance poses the limiting criterion for the minimum detectable crack size.

The limiting values computed for each dataset, for ten representative patches selected from each dataset, are shown in the following figures, Figure 10.44 and Figure 10.45. These patches were selected to represent a range of parameters within the point clouds. Figure 10.44 presents the limiting values for the first two damage categories, which was described in Chapter 1: small deformations and large deformations with no change in topology. Figure 10.45 shows the limiting values computed for large deformations with localized change in topology for the investigated datasets. From Figure 10.44, it can be observed that the detectable damage dimensions for the first two damage categories are heavily dependent on the point density. For the third damage category, for which the damage locations are detected by using the variations in the surface normals, a general correlation may be observed with the effect of the changes in point density values on the computed detectable damage dimensions. These results emphasize the dependency of the obtained defect detection results on the varying point densities observed on the surface of the investigated structures.

Figure 10.44: Minimum detectable damage dimensions versus point density for small deformations and large deformations with no change in topology.

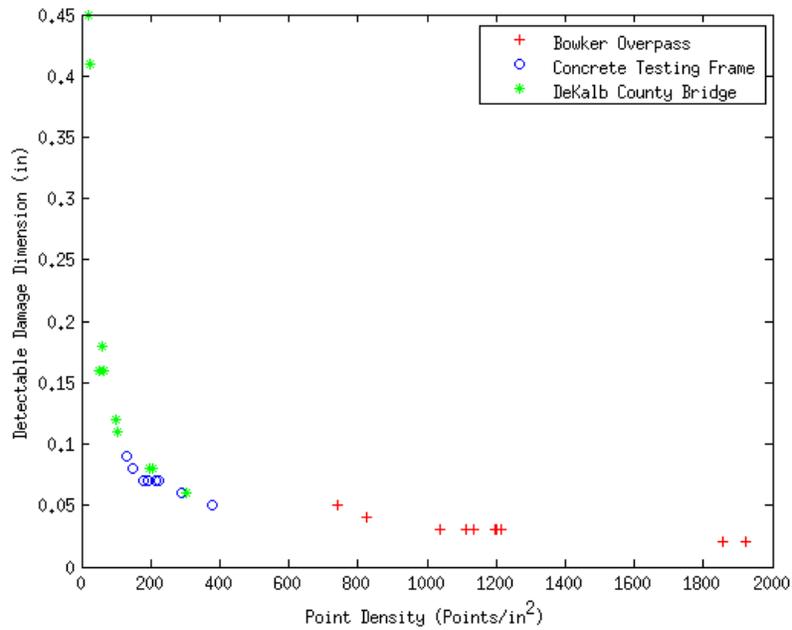
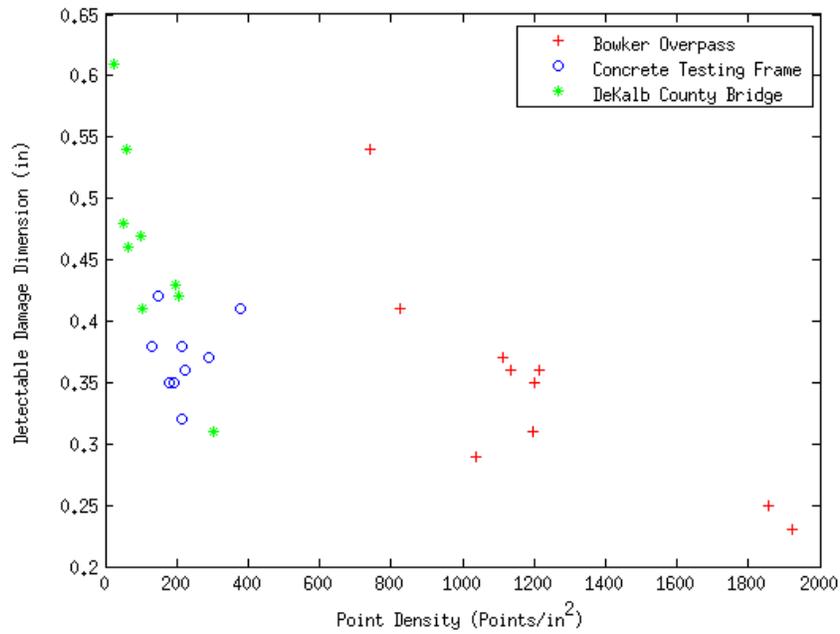


Figure 10.45: Minimum detectable damage dimensions versus point density for large deformations with localized change in topology.



## 10.5 Condition Rating

The final step of the damage detection is to determine the condition rating of the structural items. This is obtained by comparing the current physical state of the structure to what it was the day it was built. This process is important since the visual inspection based condition rating is the first part of the structural evaluation, which gives the overall condition of the structure based on all major deficiencies and its ability to carry loads. The aim is to classify the damage severity and to assign labels to the detected damage in a well-known format based on sample condition rating guidelines.

In this section, condition ratings of for the individual member of each dataset, for which the damage detection results are shown in Chapter 10, are given. The condition rating is automatically assigned for the member of interest based on the worst condition rating obtained through the condition criteria tables given in Chapter 9.

It should be noted that both local (changes in cross-sectional areas) and global changes (volumetric change and alignment information) are used to determine the representative condition rating of individual elements. All the applicable condition rating criteria tables are checked, and the governing condition rating is recorded for each element.

The automated condition rating assignment is designed for bridge members; thus, the condition rating criteria discussed in Chapter 9 covers only bridges. However, in this chapter, condition ratings for the steel member frame and concrete testing frame are also computed. This provided an initial validation on the accuracy of the condition rating assignment methodology. Finally, the condition rating computed for the processed portions of the DeKalb County Bridge and Bowker Overpass are listed.

### 10.5.1 Concrete Testing Frame

In order to test the accuracy of the condition rating assignment, the condition rating for the member of the steel testing frame was computed as if they were primary members of bridge superstructure. The ratings were given based on the amount of cracking, spalling, and alignment issues. In the worst case of cracking and spalling, the unsound areas on the members was up to 10% with the presence of structural cracking, and for the worst case alignment check, the horizontal movement was 1.84"/1' for one of the columns and the vertical sag was 5.85"/1' for one of the failed beams. The lists of the computed condition ratings both for columns and beams are given in Table 10.29 and Table 10.30.

Table 10.29: Condition rating results for columns of the concrete testing frame.

	<b>Condition Rating</b>	<b>Condition Description</b>	<b>Governing Damage Type</b>	<b>Criteria</b>
<b>A1</b>	1	Imminent failure	Alignment	Extremely more than 0.25"/ft horizontal movement
<b>A2</b>	2	Critical	Alignment	Slightly more than 0.25"/ft horizontal movement
<b>A3</b>	4	Poor	Alignment	Up to 0.125"/ft horizontal movement
<b>B1</b>	1	Imminent failure	Alignment	Extremely more than 0.25"/ft horizontal movement
<b>B2</b>	2	Critical	Alignment	Slightly more than 0.25"/ft horizontal movement
<b>B3</b>	3	Serious	Alignment	More than 0.125"/ft horizontal movement
<b>C1</b>	-	-	-	-
<b>C2</b>	3	Serious	Alignment	More than 0.125"/ft horizontal movement
<b>C3</b>	2	Critical	Alignment	Slightly more than 0.25"/ft horizontal movement
<b>D1</b>	1	Imminent failure	Alignment	Extremely more than 0.25"/ft horizontal movement
<b>D2</b>	2	Critical	Alignment	Slightly more than 0.25"/ft horizontal movement
<b>D3</b>	2	Critical	Alignment	Slightly more than 0.25"/ft horizontal movement
<b>E1</b>	1	Imminent failure	Alignment	Extremely more than 0.25"/ft horizontal movement
<b>E2</b>	2	Critical	Alignment	Slightly more than 0.25"/ft horizontal movement
<b>E3</b>	2	Critical	Alignment	Slightly more than 0.25"/ft horizontal movement

Table 10.30: Condition rating results for beams of the concrete testing frame.

	<b>Condition Rating</b>	<b>Condition Description</b>	<b>Governing Damage Type</b>	<b>Criteria</b>
<b>AB1</b>	6	Satisfactory	Alignment	Moderate misalignment (<0.1")
<b>AB2</b>	6	Satisfactory	Alignment	Moderate misalignment (<0.1")
<b>AB3</b>	6	Satisfactory	Alignment	Moderate misalignment (<0.1")
<b>BC1</b>	0	Failure	Alignment	Discontinuity
<b>BC2</b>	0	Failure	Alignment	Discontinuity
<b>BC3</b>	0	Failure	Alignment	Discontinuity
<b>CD1</b>	0	Failure	Alignment	Discontinuity
<b>CD1</b>	0	Failure	Alignment	Discontinuity
<b>CD1</b>	0	Failure	Alignment	Discontinuity
<b>DE1</b>	6	Satisfactory	Alignment	Moderate misalignment (<0.1")
<b>DE2</b>	6	Satisfactory	Alignment	Moderate misalignment (<0.1")
<b>DE3</b>	5	Fair	Alignment	Moderate misalignment (>0.1")

### 10.5.2 DeKalb County Bridge

The items of the portion of the Dekalb County Bridge that were used for both structural sensing and damage assessment are included in this section (Guldur and Hajjar, 2014).

The deck was already collapsed, thus the assigned condition rating for this item is 0-failure. The pier cap, on the other hand, was in a better shape with only minor concrete spalling detected on the surface. The conditions of the timber piles were varying; it ranged from minor cracking, splitting and measurable section loss to completely crushing and splitting. The condition rating results are given in Table 10.31.

Table 10.31: Condition rating results for a portion of the DeKalb County Bridge.

	<b>Condition Rating</b>	<b>Condition Description</b>	<b>Governing Damage Type</b>	<b>Criteria</b>
<b>Deck</b>	0	Failure	Concrete Spalling	More than 30% spalling and discontinuity
<b>Pier cap</b>	6	Satisfactory	Concrete Spalling	Up to 5% spalling
<b>Pier 1</b>	3	Serious	Timber pier damage	Severe section loss>30%
<b>Pier 2</b>	3	Serious	Timber pier damage	Severe section loss>30%
<b>Pier 3</b>	1	Imminent Failure	Timber pier damage	Discontinuity, complete splitting
<b>Pier 4</b>	1	Imminent Failure	Timber pier damage	Discontinuity, complete splitting
<b>Pier 5</b>	4	Poor	Timber pier damage	Cracking and splitting <30%
<b>Pier 6</b>	4	Poor	Timber pier damage	Cracking and splitting <30%
<b>Pier 7</b>	3	Serious	Timber pier damage	Severe section loss>30%
<b>Pier 8</b>	5	Fair	Timber pier damage	Cracking and splitting <10%

### 10.5.3 Bowker Overpass

For the Bowker Overpass, the condition ratings for only the regions, on which the validation was performed, are included.

The condition rating for the underside of the deck, for the investigated span, was found to be 5-fair since structural cracking with leaking, efflorescence was observed along with combined areas of spalling with exposed reinforcement less than 10%. For the steel girder, the condition rating was assigned as 4-poor since advanced section loss was observed. The list of the computed condition ratings is given in Table 10.32. These results are in accordance with the latest inspection report of the investigated regions, which was prepared by MassDOT (Mercuri and Mirliss, 2013).

Table 10.32: Condition rating results for two locations on the Bowker Overpass.

	<b>Condition Rating</b>	<b>Condition Description</b>	<b>Governing Damage Type</b>	<b>Criteria</b>
<b>Concrete Deck</b>	5	Fair	Concrete Spalling	Up to 10% spalling
<b>Steel Girder</b>	4	Poor	Section loss	More than 50%

## **11. Summary and Conclusion**

Not only are the bridges of the United States in a state of serious deterioration, but the methods used to perform inspections are slow and sometimes unreliable. Visual inspections may be dangerous and are not repeatable because they are subjective. This research has investigated the use of high-accuracy, camera-integrated laser scanners as sensors for capturing continuous, dense, texture-mapped surface data to advance the current practice by partially or fully automating inspections for damage localization, quantification, and documentation, thus augmenting and assisting current sensor-based and visual inspection strategies. This chapter first provides a summary of this research and its major contributions. The conclusions are laid out in Section 11.1 and future work is proposed in Section 11.2

At the beginning of this dissertation, an extensive literature review was presented to lay out relevant work that was previously completed and that is in progress. This literature review demonstrated that using laser scanning technology for structural sensing and damage detection is a trending research area; however, the number of studies performed on this topic is limited. With recent improvements in both laser scanner technology and computer processing power, it is now possible to better investigate the full potential of laser scanning technology for many engineering applications.

Our research has focused on the use of laser scanning technology for structural assessment and surface damage detection. The use of this technology required developing new methodologies that use computer vision for interpreting 3D data and extracting useful information in a format that could be interpreted by practitioners. We extended and used existing laser point cloud processing algorithms for processing range data in order to

identify objects, their location, orientation, and size, in a scene by using a pre-defined model library, which stores both 2D and 3D shape descriptors of the objects.

The implemented algorithms for point cloud processing were mostly taken from previously completed work; however, they were altered to be used in our applications. Since one objective of this research was to locate and quantify surface damage, object detection was used to extract surface patches with similar properties to separate the point cloud into meaningful clusters for comparative analysis. To perform object detection, two major methods were implemented: global feature-based object detection and graph-based object detection.

Once the objects were detected, these object representations were utilized for obtaining the geometric model of the structure. Defect localization was achieved by developing new strategies for comparing the captured data, which represent the as-is condition of the structure, with the correctly detected and fitted objects from the model library in order to locate and quantify discrepancies. Since the comparison strategies described above differ significantly depending on the defect type, the first step of this research was to determine the possible defect types that can be detected using laser point clouds. Generalized comparison strategies, associated with each damage type, were created by using both the surface information and volume information obtained through laser point cloud processing. This methodology allowed us to categorize and develop damage detection strategies for a wide range of damage types including cracks, corrosion, ruptures, bent members, points of discontinuity, and concrete spalling.

Two sets of damage detection strategies were developed for this research: graph-based damage detection and surface-based damage detection. The first developed damage

detection method, graph-based damage detection, uses the skeleton information of an object in order to locate and quantify surface damage such as ruptures, bent members, points of discontinuity, and concrete spalling. Either model induced or predicted as-design properties were compared with the information obtained from the skeleton that represents the as-is condition of an investigated object. The discrepancies were located, and the surface damage was quantified.

The second method developed for damage detection is the surface normal-based damage detection. This method was developed to expand the variety of the detected damage types to include cracks and corrosion along with ruptures and spalling. The modal properties of the detected surfaces and/or objects were used to locate the defective areas on the structural surfaces. Detecting only the locations of the defects was not sufficient to automatically quantify the damage. Thus, a clustering methodology was developed to group the detected defect point into individual damage clusters. To improve the clustering process, a silhouette-based cluster evaluation method was used to optimize the final number of defect clusters. The developed damage area and volume computation strategies were then used to record quantitative information on the detected damage clusters. First, a convex hull based damage quantification method was implemented; however, it was observed that this method is only applicable to a certain set of damage configurations. Hence, a new mesh-grid based damage quantification system was developed to quantify both area and volume of the detected damage.

For cracks, a clustering step, which is a tuned version of the previous clustering method, was introduced to separate the cracks detected on a single patch into individual clusters. The previous clustering process was enhanced by introducing a strategy that

merges closely located clusters to avoid over clustering. Finally, for calculating the length and thickness information from cracks, a method that performs regression analysis was developed. The thicknesses of the cracks were determined through residual analysis, and the average crack thickness was recorded.

Once the detected small defects were segmented into individual clusters, a set of finite values representing quantitative attributes and/or properties are assigned to each defect cluster. In total, seven features were used: eccentricity, area divided by eccentricity ellipse, solidity, absolute value of correlation coefficient, compactness, and mean and standard deviation of the intensity distribution. A three layered feed forward neural network was used for classification, and the accuracy, precision, sensitivity and specificity were recorded to analyze the performance of the used classifier.

Condition assessment criteria were also integrated into the damage detection algorithms to determine the condition rating automatically from the texture mapped point clouds. The condition assessment criteria requirements from a representative state department of transportation were observed, and the most comprehensive defect criteria for each investigated damage types were used in the algorithm development. Finally, all the listed developed methodologies for structural sensing and damage assessment were applied to several datasets, starting with synthetic point clouds, continuing with several lab datasets and finally ending with two bridge datasets. The following sections, Sections 11.1 and 11.2, present the conclusions of this study and suggestion for future work, respectively.

## 11.1 Conclusions

The primary conclusions of this research are described as follows:

- Both global feature-based and graph-based object detection methods can effectively detect predefined objects from point clouds by using a dissimilarity measure.
- A novel graph-based object detection method was shown to successfully detect local changes in the geometry of an investigated member to locate and quantify the following damage types: ruptures, bent members, alignment issues, points of discontinuity, and spalled concrete regions.
- A novel surface-based damage detection strategy was shown to efficiently locate and quantify the targeted defect types, which are cracks, corrosion, ruptures, and spallings.
- A unique neural network classifier implementation, which was performed on a specific point cloud dataset, showed that this classifier could be effectively used for crack detection on point clouds with an accuracy of 93%.
- The condition assessment strategy provided the opportunity to document quantitative information on damaged areas in a widely-accepted format.
- Thus, it was shown that laser scanners can be used for aiding visual inspection and enhancing the quality of the collected information.

Overall, this research showed that the strategies developed for texture-mapped laser-scans with varying quality can be used to generate automated damage detection, quantification, documentation methods and to create a self-acting condition assessment system.

## 11.2 Future Work

This research lays the groundwork for several possible future research projects, the details of which are discussed below.

Even though the developed algorithms are capable of detecting surface damage locations efficiently, there remains many possibilities for improvement. First, most of the utilized input parameters for the developed algorithms are automatically extracted using local surface properties. This is an effective method for computing the necessary inputs; however, parameter selection can be specifically optimized for each structure type. Future research could focus on selecting the best optimization strategy for the discussed applications and testing the optimization strategy's capabilities on an extensive calibration and validation set. Second, the method used for crack clustering could be improved. The feature set extracted from the investigated patches could be altered to contain descriptive object information, which includes individual defect information rather than just having a generalized defect indicator. Third, in this research, the performance of only one classifier, the neural network classifier, is observed. Different classifiers could be used to perform a comparative study.

A second area for future work is the automation of the documentation process for visual inspections. For surface defects, the current reporting methods involve data collection on site and then, sketching the recorded damage data on a plan off site. This can be enhanced by developing appropriate software that will use initial plan drawings and/or bridge information models to sketch the recorded defect information. It is also possible to improve these strategies further by adding the condition assessment

information extracted from the laser scanners, in order to improve current load rating calculation strategies.

In this research, as well as in many other studies, the information model generation from point clouds is investigated. However, the conversion from information models to finite element models is a new research area. Laser scanning technology can be used to identify and evaluate damage, both with and without the use of structural analysis models, formed either from raw point clouds or semantic models. Further investigation could be conducted to evaluate the performance of the augmented analysis procedures in performing accurate behavior predictions and providing inspection feedback.

It is difficult to predict the loading pattern that causes a change in the topology of a fractured or collapsed member, or structure, with available methods. Laser scanners can be used as a validation tool after investigating structures under expected loading scenarios, which can later be used for predicting what may have caused the observed changes in topology (e.g., which loading scenario), identifying damaged structural elements, identifying new topology for comparative analysis, and validating some collapse/fracture models and post-event evaluation.

Camera-integrated laser scanners are the only sensor type that is used in this research. Combining information collected via laser scanners with other available sensor technologies outputs would enable developing a more complete inspection, monitoring and documentation strategy, where different types of sensors complement each other.

Finally, another application that could benefit from using the developed model and damage detection algorithms is surveying of large-scale area surveying using LiDAR systems. Geographic information systems along with LiDARs can be used to develop

robust, automated post-hazard damage assessment strategies, while the collected information could be used to enhance coordinating post-hazard response teams effectively.

## References

1. Aaccessmaps. (Accessed in May 2014). "Boston - Fenway - Backbay." [http://www.aaccessmaps.com/show/map/us/ma/boston\\_fenway\\_backbay](http://www.aaccessmaps.com/show/map/us/ma/boston_fenway_backbay).
2. AASHTO. (2010). *Bridge Element Inspection Guide Manual, 1st Edition*. American Association of State Highway and Transportation Officials, Washington, DC.
3. AASHTO. (2011). *The Manual for Bridge Evaluation, 2nd Edition*. American Association of State Highway and Transportation Officials, Washington, DC.
4. Abdel-Qader, I., Abudayyeh, O. and Kelly, M. E. (2003). "Analysis of Edge-Detection Techniques for Crack Identification in Bridges." *Journal of Computing in Civil Engineering*, 17(4), 255-263.
5. Abdel-Qader, I., Pashaie-Rad, S., Abudayyeh, O. and Yehia, S. (2006). "PCA-based Algorithm for Unsupervised Bridge Crack Detection." *Advances in Engineering Software*, 37(12), 771-778.
6. Adams, R. A. (1999). *A Complete Course Calculus*. University of British Columbia, Addison Wesley Longman Ltd., Boston, MA.
7. Adan, A., Xiong, X., Akinci, B. and Huber, D. (2011). "Automatic Creation of Semantically Rich 3D Building Models from Laser Scanner Data." *Proceedings of the International Symposium on Automation and Robotics in Construction (ISARC)*, Seoul, Korea, 342-347.
8. Adhikari, R., Moselhi, O. and Bagchi, A. (2014). "Image-based Retrieval of Concrete Crack Properties for Bridge Inspection." *Automation in construction*, 39, 180-194.
9. ADOT. (Accessed in July 2014). "Arizona Department of Transportation Bridge Load Rating Guidelines." Arizona Department of Transportation. <https://www.azdot.gov/docs/default-source/bridge-group/loadratingguidelines.pdf?sfvrsn=2>.
10. AISC. (2011). *Steel Construction Manual, 14th Edition*. American Institute of Steel Construction, Chicago, Illinois.
11. Anil, E., Akinci, B., Garrett, J. and Kurc, O. (2013). "Representation of Damage Information for Post-Earthquake Damage Assessment of Reinforced Concrete Frames." *Proceedings of Computing in Civil Engineering*, Los Angeles, CA, 825-832.

12. Anil, E. B., Akinci, B., Garrett, J. H. and Kurc, O. (2013). "Characterization of Laser Scanners for Detecting Cracks for Post-earthquake Damage Inspection." *Proceedings of International Symposium on Automation and Robotics in Construction and Mining*, Montreal, QC, Canada, 313-320.
13. Anil, E. B., Tang, P., Akinci, B. and Huber, D. (2013). "Deviation Analysis Method for the Assessment of the Quality of the As-is Building Information Models Generated from Point Cloud Data." *Automation in Construction*, 35, 507-516.
14. Ankerst, M., Kastenmüller, G., Kriegel, H.-P. and Seidl, T. (1999). "3D Shape Histograms for Similarity Search and Classification in Spatial Databases." *Proceedings of Advances in Spatial Databases*, Hong Kong, China, 207-226.
15. Arayici, Y. (2008). "Towards Building Information Modelling for Existing Structures." *Structural Survey*, 26(3), 210-222.
16. ASCE. (Accessed in July 2014). "2013 Report Card for America's Infrastructure." American Society of Civil Engineers. <http://www.infrastructurereportcard.org/>.
17. Autodesk Inc. (Accessed in July 2014). "AutoCAD User Manual." [http://docs.autodesk.com/ACDMAC/2013/ENU/PDFs/acdmac\\_2013\\_users\\_guide.pdf](http://docs.autodesk.com/ACDMAC/2013/ENU/PDFs/acdmac_2013_users_guide.pdf).
18. Autodesk Inc. (Accessed in July 2014). "Revit User Manual." [http://revit.downloads.autodesk.com/download/9\\_1/HelpBuildingENU.pdf](http://revit.downloads.autodesk.com/download/9_1/HelpBuildingENU.pdf).
19. Barber, C. B., Dobkin, D. P. and Huhdanpaa, H. (1996). "The Quickhull Algorithm for Convex Hulls." *ACM Transactions on Mathematical Software* 22(4), 469-483.
20. Barla, A., Odone, F. and Verri, A. (2003). "Histogram Intersection Kernel for Image Classification." *Proceedings of International Conference on Image Processing*, Barcelona, Spain, 513-516.
21. Barnea, S. and Filin, S. (2008). "Keypoint Based Autonomous Registration of Terrestrial Laser Point-Clouds." *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(1), 19-35.
22. Belton, D. and Lichti, D. D. (2006). "Classification and Segmentation of Terrestrial Laserscanner Point Clouds Using Local Variance Information." *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5), 44-49.

23. Benkő, P., Martin, R. R. and Várady, T. (2001). "Algorithms for Reverse Engineering Boundary Representation Models." *Computer-Aided Design*, 33(11), 839-851.
24. Berenyi, A., Lovas, T. and Barsi, A. (2010). "Terrestrial Laser Scanning: Civil Engineering Applications." *International Archives Of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(5), 80-85.
25. Berkmann, J. and Caelli, T. (1994). "Computation of Surface Geometry and Segmentation Using Covariance Techniques." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(11), 1114-1116.
26. Bernardini, F. and Rushmeier, H. (2002). "The 3D Model Acquisition Pipeline." *Computer Graphics Forum*, 21(2), 149-172.
27. Besl, P. J. (1988). *Surfaces in Range Image Understanding*. Springer Series in Perception Engineering, New York, NY.
28. Besl, P. J. and Jain, R. C. (1988). "Segmentation Through Variable-order Surface Fitting." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2), 167-192.
29. Besl, P. J. and McKay, N. D. (1992). "Method for Registration of 3-D Shapes." *Sensor Fusion IV: Control Paradigms and Data Structures*, Boston, MA, 586-606.
30. Bingöl, F., Mann, J. and Larsen, G. C. (2010). "Light Detection and Ranging Measurements of Wake Dynamics Part I: One-dimensional Scanning." *Wind Energy*, 13(1), 51-61.
31. Borello, D. J., Andrawes, B., Hajjar, J., Olson, S. M., Hansen, J. and Buenker, J. (2009). *Forensic Collapse Investigation of a Concrete Bridge with Timber Piers*. Research Report ICT-09-042, Illinois Center for Transportation (ICT), Springfield, IL.
32. Borello, D. J., Andrawes, B., Hajjar, J. F., Olson, S. M. and Hansen, J. (2010). "Experimental and Analytical Investigation of Bridge Timber Piles Under Eccentric Loads." *Engineering Structures*, 32(8), 2237-2246.
33. Bosché, F. (2010). "Automated Recognition of 3D CAD Model Objects in Laser Scans and Calculation of As-built Dimensions for Dimensional Compliance Control in Construction." *Advanced Engineering Informatics*, 24(1), 107-118.
34. Bosche, F. and Haas, C. (2008). "Automated Retrieval of 3D CAD Model Objects in Construction Range Images." *Automation in Construction*, 17(4), 499-512.

35. Breyer, W. H. (1987). *CRC Standard Mathematical Tables and Formulas*. CRC Press, Boca Raton, FL.
36. Bruno, I. J., Kemp, N. M., Artymiuk, P. J. and Willett, P. (1997). "Representation and Searching of Carbohydrate Structures Using Graph-theoretic Techniques." *Carbohydrate Research*, 304(1), 61.
37. Buckner, B. D., Markov, V., Lai, L.-C. and Earthman, J. C. (2008). "Laser-scanning Structural Health Monitoring with Wireless Sensor Motes." *Optical Engineering*, 47(5), 1-9.
38. Canny, J. (1986). "A Computational Approach to Edge Detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 679-698.
39. Chang, P. C., Flatau, A. and Liu, S. (2003). "Review Paper: Health Monitoring of Civil Infrastructure." *Structural Health Monitoring*, 2(3), 257-267.
40. Chen, S.-E. (Accessed in July 2014). "Laser Scanning Technology for Bridge Monitoring." <http://www.intechopen.com/books/laser-scanner-technology/laser-scanning-technology-for-bridge-monitoring>.
41. Chen, S.-E., Rice, C., Boyle, C. and Hauser, E. (2011). "Small-format Aerial Photography for Highway-bridge Monitoring." *Journal of Performance of Constructed Facilities*, 25(2), 105-112.
42. Chen, Z. and Hutchinson, T. C. (2010). "Probabilistic Urban Structural Damage Classification Using Bitemporal Satellite Images." *Earthquake Spectra*, 26(1), 87-109.
43. Cheng, H., Chen, J.-R., Glazier, C. and Hu, Y. (1999). "Novel Approach to Pavement Cracking Detection based on Fuzzy Set Theory." *Journal of Computing in Civil Engineering*, 13(4), 270-280.
44. Chi, S., Caldas, C. H. and Kim, D. Y. (2009). "A Methodology for Object Identification and Tracking in Construction based on Spatial Modeling and Image Matching Techniques." *Computer-Aided Civil and Infrastructure Engineering*, 24(3), 199-211.
45. Choudhary, G. K. and Dey, S. (2012). "Crack Detection in Concrete Surfaces using Image Processing, Fuzzy Logic, and Neural Networks." *IEEE Fifth International Conference on Advanced Computational Intelligence*, Nanjing, Jiangsu, China.
46. Committee on the Safety of Existing Dams. (1983). *Safety of Existing Dams: Evaluation and Improvement*. National Research Council, Washington, DC.

47. Cooley, J. W. and Tukey, J. W. (1965). "An Algorithm for the Machine Calculation of Complex Fourier Series." *Mathematics of Computation*, 19(90), 297-301.
48. Curless, B. (1999). "From Range Scans to 3D Models." *ACM SIGGRAPH Computer Graphics*, 33(4), 38-41.
49. Cyr, C. M. and Kimia, B. B. (2001). "3D Object Recognition using Shape Similarity-based Aspect Graph." *Proceedings of Eighth IEEE International Conference on Computer Vision* Vancouver, British Columbia, Canada, 254-261.
50. Dai, M. and Newman, T. S. (1998). *Hyperbolic and Parabolic Quadric Surface Fitting Algorithms—Comparison Between the Least Squares Approach and the Parameter Optimization Approach*. University of Alabama at Huntsville, Huntsville, Alabama.
51. Daniels, J., Ha, L. K., Ochotta, T. and Silva, C. T. (2007). "Robust Smooth Feature Extraction from Point Clouds." *IEEE International Conference on Shape Modeling and Applications, SMI'07*, Lyon, France, 123-136.
52. De Berg, M., Van Kreveld, M., Overmars, M. and Schwarzkopf, O. C. (2000). *Computational Geometry: Algorithms and Applications*. Springer, Heidelberg, Germany.
53. Demarsin, K., Vanderstraeten, D., Volodine, T. and Roose, D. (2007). "Detection of Closed Sharp Edges in Point Clouds Using Normal Estimation and Graph Theory." *Computer-Aided Design*, 39(4), 276-283.
54. Dorninger, P. and Nothegger, C. (2007). "3D Segmentation of Unstructured Point Clouds for Building Modeling." *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35(3/W49A), 191-196.
55. Dyn, N., Hormann, K., Kim, S.-J. and Levin, D. (2001). "Optimizing 3D Triangulations Using Discrete Curvature Analysis." *Mathematical Methods for Curves and Surfaces*, 1, 135-146.
56. Eatherton, M. R., Ma, X., Krawinkler, H., Mar, D., Billington, S., Hajjar, J. F. and Deierlein, G. G. (2014). "Design Concepts for Controlled Rocking of Self-Centering Steel-Braced Frames." *Journal of Structural Engineering*, Technical Paper 04014082, 1-11.
57. Fan, T.-J., Medioni, G. and Nevatia, R. (1987). "Segmented Descriptions of 3-D Surfaces." *IEEE Journal of Robotics and Automation*, 3(6), 527-538.
58. FARO Technologies Inc. (Accessed in July 2014). "FARO Focus3D: Features, Benefits & Technical Specifications."

- file:///C:/Users/B.Guldur/Downloads/TechSheet\_Laser%20Scanner%20Focus3D%20(4).pdf.
59. FARO Technologies Inc. (Accessed in July 2014). "FARO Photon 80/20." <http://archive.cyark.org/temp/FAROphton8020datasheet.pdf>.
  60. FARO Technologies Inc. (Accessed in July 2014). "FARO Scene 5.3 User Manual" [https://faro.blob.core.windows.net/sitefinity/3d-app-center-downloads/E1208\\_SCENE\\_LT\\_5-3\\_Manual\\_EN.pdf](https://faro.blob.core.windows.net/sitefinity/3d-app-center-downloads/E1208_SCENE_LT_5-3_Manual_EN.pdf).
  61. FHWA (Accessed in July 2014). "National Bridge Inventory." Federal Highway Administration. <http://www.fhwa.dot.gov/bridge/nbi.cfm>.
  62. FHWA. (2012). *Steel Bridge Design Handbook: Load Rating of Steel Bridges*. FHWA-IF-12-052 - Vol. 18, Federal Highway Administration, Washington, DC.
  63. FHWA. (Accessed in July 2014). "National Bridge Inventory Condition Ratings." Federal Highway Administration. <http://www.fhwa.dot.gov/bridge/nbi.cfm>.
  64. Gelfand, N., Mitra, N. J., Guibas, L. J. and Pottmann, H. (2005). "Robust Global Registration." *Proceedings of the Third Eurographics Symposium on Geometry Processing*, Vienna, Austria, 197-206.
  65. Geomatic 3D Systems Inc. (Accessed in May 2014). "3D Scanners - A Guide to 3D Scanner Technology." <http://www.rapidform.com/3d-scanners/>.
  66. Gibson, S. F. F. (1995). "Beyond Volume Rendering: Visualization, Haptic Exploration, and Physical Modeling of Voxel-based Objects." *Visualization in Scientific Computing*, Chia, Italy, 10-24.
  67. Gielsdorf, F., Rietdorf, A. and Gründig, L. (2004). "Geometrical Modelling of Buildings - the Frame for the Integration of Data Acquisition Techniques." *Proceedings of FIG Working Week*, Athens, Greece, 1-10.
  68. Google Maps. (Accessed in July 2014). "Charlesgate Overpass." <https://www.google.com/maps/place/Charlesgate,+Boston,+MA+02215/@42.3488954,-71.0923586,17z/data=!3m1!4b1!4m2!3m1!1s0x89e37a1cb742d745:0xfc98a860690dc00>.
  69. Gordon, S. and Lichti, D. (2004). "Terrestrial Laser Scanners with a Narrow Field of View: The Effect on 3D Resection Solutions." *Survey Review*, 37(292), 448-468.

70. Gordon, S. J. and Lichti, D. D. (2007). "Modeling Terrestrial Laser Scanner Data for Precise Structural Deformation Measurement." *Journal of Surveying Engineering*, 133(2), 72-80.
71. Governing. (Accessed in June 2013). "Condition Ratings for State and Local Bridges." <http://www.governing.com/gov-data/transportation-infrastructure/state-local-government-owned-bridge-condition-ratings.html>.
72. Guldur, B. and Hajjar, J. F. (2013). "Laser-based Automatic Cross-Sectional Change Detection for Steel Frames." *Proceedings of the 9th International Workshop on Structural Health Monitoring 2013*, Stanford University, Stanford, CA, 1-8.
73. Guldur, B. and Hajjar, J. F. (2014). "Automated Damage Assessment from 3-D Laser Scans." *Proceedings of the Istanbul Bridge Conference*, Istanbul, Turkey, 1-11.
74. Guldur, B. and Hajjar, J. F. (2014). "Damage Detection on Structures Using Texture Mapped Laser Point Clouds." *Proceedings of the ASCE/SEI Structures Congress 2014*, Boston, MA, 324-334.
75. Haar, A. (1910). "Zur theorie der orthogonalen funktionensysteme." *Mathematische Annalen*, 69(3), 331-371.
76. Hawarey, M. and Falk, M. O. (2004). "Using Laser Scanning Technology to Measure Deflections in Steel Columns." *Iron & Steel Technology*, 1(3), 40-45.
77. Hearst, M. A., Dumais, S., Osman, E., Platt, J. and Scholkopf, B. (1998). "Support Vector Machines." *Intelligent Systems and their Applications, IEEE*, 13(4), 18-28.
78. Hetzel, G., Leibe, B., Levi, P. and Schiele, B. (2001). "3D Object Recognition from Range Images Using Local Feature Histograms." *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Kauai, HI, 394-399.
79. Higgins, C. and Turan, O. T. (2011). "Imaging Tools for Evaluation of Gusset Plate Connections in Steel Truss Bridges." *Journal of Bridge Engineering*, 18(5), 380-387.
80. Hinks, T. (2011). "Geometric Processing Techniques for Urban Aerial Laser Scan Data." Ph.D. Dissertation, College of Engineering, Mathematical and Physical Sciences, University College Dublin, Dublin, Ireland.

81. Hirth, B. D., Schroeder, J. L., Gunter, W. S. and Guynes, J. G. (2014). "Coupling Doppler Radar-derived Wind Maps with Operational Turbine Data to Document Wind Farm Complex Flows." *Wind Energy*, Wiley Online, 1-12.
82. Hisada, M., Belyaev, A. G. and Kunii, T. L. (2001). "A 3D Voronoi-based Skeleton and Associated Surface Features." *Proceedings. Ninth Pacific Conference on Computer Graphics and Applications*, Tokyo, Japan, 89-96.
83. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W. (1992). "Surface Reconstruction from Unorganized Points." *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, New York City, NY, 71-78.
84. Horn, B. K. P. (1984). "Extended Gaussian Images." *Proceedings of the IEEE*, 72(12), 1671-1686.
85. Hornik, K., Stinchcombe, M. and White, H. (1989). "Multilayer Feed Forward Networks are Universal Approximators." *Neural Networks*, 2(5), 359-366.
86. Hu, M.-K. (1962). "Visual Pattern Recognition by Moment Invariants." *IRE Transactions on Information Theory*, 8(2), 179-187.
87. Huang, J. and Menq, C.-H. (2002). "Automatic CAD Model Reconstruction from Multiple Point Clouds for Reverse Engineering." *Journal of Computing and Information Science in Engineering*, 2(3), 160-170.
88. Huber, D., Akinci, B., Oliver, A., Anil, E., Okorn, B. and Xiong, X. (2011). "Methods for Automatically Modeling and Representing as-built Building Information Models." *NSF Engineering Research and Innovation Conference*, Atlanta, GA, 1-8.
89. Ingensand, H., Ryf, A. and Schulz, T. (2003). "Performances and Experiences in Terrestrial Laserscanning." *Proceedings of the 6th Conference on Optical 3D Measurement Techniques*, Zurich, 1-8.
90. Ip, C. Y. and Gupta, S. K. (2007). "Retrieving Matching CAD Models by using Partial 3D Point Clouds." *Computer-Aided Design & Applications*, 4(5), 629-638.
91. Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y. and Ramani, K. (2005). "Three-dimensional Shape Searching: State-of-the-art Review and Future Trends." *Computer-Aided Design*, 37(5), 509-530.
92. Jahanshahi, M. R. and Masri, S. F. (2012). "Adaptive Vision-based Crack Detection using 3D Scene Reconstruction for Condition Assessment of Structures." *Automation in Construction*, 22, 567-576.

93. Jahanshahi, M. R., Masri, S. F. and Sukhatme, G. S. (2011). "Multi-image Stitching and Scene Reconstruction for Evaluating Defect Evolution in Structures." *Structural Health Monitoring*, 10(6), 643-657.
94. Janos, T. (Accessed in July 2014). "Precision Agriculture." [http://www.tankonyvtar.hu/en/tartalom/tamop425/0032\\_precizios\\_mezogazdasag/ch04s04.html#id486775](http://www.tankonyvtar.hu/en/tartalom/tamop425/0032_precizios_mezogazdasag/ch04s04.html#id486775).
95. Jelalian, A. V. (1992). *Laser Radar Systems*. Artech Print on Demand, Boston, MA.
96. Johnson, A. E. and Hebert, M. (1999). "Using Spin images for Efficient Object Recognition in Cluttered 3D Scenes." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5), 433-449.
97. Jolliffe, I. (2002). *Principal Components in Regression Analysis, 2nd Edition*. Springer, Medford, MA.
98. Kaseko, M. S., Lo, Z.-P. and Ritchie, S. G. (1994). "Comparison of Traditional and Neural Classifiers for Pavement-crack Detection." *Journal of Transportation Engineering*, 120(4), 552-569.
99. Kaseko, M. S. and Ritchie, S. G. (1993). "A Neural Network-based Methodology for Pavement Crack Detection and Classification." *Transportation Research Part C: Emerging Technologies*, 1(4), 275-291.
100. Kashani, A. G., Crawford, P. S., Biswas, S. K., Graettinger, A. J. and Grau, D. (2014). "Automated Tornado Damage Assessment and Wind Speed Estimation Based on Terrestrial Laser Scanning." *Journal of Computing in Civil Engineering*, Technical Paper 04014051, 1-10.
101. Käsler, Y., Rahm, S., Simmet, R. and Kühn, M. (2010). "Wake Measurements of a Multi-MW wind Turbine with Coherent Long-range Pulsed Doppler Wind Lidar." *Journal of Atmospheric and Oceanic Technology*, 27(9), 1529-1532.
102. Kastenmüller, G., Kriegel, H.-P. and Seidl, T. (1998). "Similarity Search in 3D Protein Databases." *Proceedings of the German Conference on Bioinformatics*, Koln, Germany, 1-3.
103. Kaufman, L. and Rousseeuw, P. J. (2009). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, Hoboken, NJ.
104. Kayen, R., Pack, R. T., Bay, J., Sugimoto, S. and Tanaka, H. (2006). "Terrestrial-LIDAR visualization of surface and structural deformations of the 2004 Niigata Ken Chuetsu, Japan, earthquake." *Earthquake Spectra*, 22(S1), 147-162.

105. Ke, Y., Fan, S., Zhu, W., Li, A., Liu, F. and Shi, X. (2006). "Feature-based Reverse Modeling Strategies." *Computer-Aided Design*, 38(5), 485-506.
106. Keller, J. M., Gray, M. R. and Givens, J. A. (1985). "A Fuzzy k-nearest Neighbor Algorithm." *IEEE Transactions on Systems, Man and Cybernetics*, 4, 580-585.
107. Kim, C., Son, H. and Kim, C. (2013). "Automated Construction Progress Measurement using a 4D Building Information Model and 3D Data." *Automation in Construction*, 31, 75-82.
108. Klasing, K., Althoff, D., Wollherr, D. and Buss, M. (2009). "Comparison of Surface Normal Estimation Methods for Range Sensing Applications." *IEEE International Conference on Robotics and Automation*, IEEE, Kobe, Japan, 3206-3211.
109. Koenderink, J. J. and Van Doorn, A. J. (1976). "The Singularities of the Visual Mapping." *Biological Cybernetics*, 24(1), 51-59.
110. Koenderink, J. J. and van Doorn, A. J. (1992). "Surface Shape and Curvature Scales." *Image and Vision Computing*, 10(8), 557-564.
111. Kriegel, H.-P., Brecheisen, S., Kröger, P., Pfeifle, M. and Schubert, M. (2003). "Using Sets of Feature Vectors for Similarity Search on Voxelized CAD Objects." *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, San Diego, CA, 587-598.
112. Laefer, D. F., Truong-Hong, L., Carr, H. and Singh, M. (2014). "Crack Detection Limits in Unit Based Masonry with Terrestrial Laser Scanning." *NDT&E International*, 62, 66-76.
113. Lam, L., Lee, S.-W. and Suen, C. Y. (1992). "Thinning Methodologies-A Comprehensive Survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9), 869-885.
114. Lamdan, Y. and Wolfson, H. (1988). "Geometric Hashing: A General and Efficient Model-based Recognition Scheme." *Proceedings of International Conference on Computer Vision*, Tampa, FL, 238-249.
115. Lee, H. M. and Park, H. S. (2011). "Gauge-free Stress Estimation of a Beam-like Structure Based on Terrestrial Laser Scanning." *Computer-Aided Civil and Infrastructure Engineering*, 26(8), 647-658.
116. Lee, T.-C., Kashyap, R. L. and Chu, C.-N. (1994). "Building Skeleton Models via 3-D Medial Surface/axis Thinning Algorithms." *CVGIP: Graphical Model and Image Processing*, 56(6), 462-478.

117. Li-Chee-Ming, J., Gumerov, D., Ciobanu, T. and Armenakis, C. (2009). "Generation of Three Dimensional Photo-realistic Models from LiDAR and Image Data." *IEEE Toronto International Conference on Science and Technology for Humanity (TIC-STH)*, Toronto, ON, Canada, 445-450.
118. Li, M., Cheng, L., Gong, J., Liu, Y., Chen, Z., Li, F., Chen, G., Chen, D. and Song, X. (2008). "Post-earthquake Assessment of Building Damage Degree using LiDAR Data and Imagery." *Science in China Series E: Technological Sciences*, 51(2), 133-143.
119. Lindenbergh, R., Uchanski, L., Bucksch, A. and Van Gosliga, R. (2009). "Structural Monitoring of Tunnels Using Terrestrial Laser Scanning." *Reports on Geodesy*, 87(2), 231-238.
120. Liu, W., Chen, S.-e. and Hasuer, E. (2011). "Bridge Clearance Evaluation based on Terrestrial LiDAR Scan." *Journal of Performance of Constructed Facilities*, 26(4), 469-477.
121. Liu, W., Chen, S. and Hauser, E. (2011). "LiDAR-Based Bridge Structure Defect Detection." *Experimental Techniques*, 35(6), 27-34.
122. Liu, Y. and Xiong, Y. (2008). "Automatic Segmentation of Unorganized Noisy Point Clouds Based on the Gaussian Map." *Computer-Aided Design*, 40(5), 576-594.
123. Lucieer, A. and Stein, A. (2005). "Texture-based Landform Segmentation of LiDAR Imagery." *International Journal of Applied Earth Observation and Geoinformation*, 6(3), 261-270.
124. MassDOT. (1963). *Charlesgate Interchange, Storrow Drive to Gaston Circle*. Massachusetts Department of Transportation, Boston, MA.
125. MatLab (2013). *MATLAB and Statistics Toolbox Release 2013b*. The MathWorks, Inc., Natick, MA.
126. Maurer Jr, C. R., Qi, R. and Raghavan, V. (2003). "A Linear Time Algorithm for Computing Exact Euclidean Distance Transforms of Binary Images in Arbitrary Dimensions." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2), 265-270.
127. Menq, C. H., Yau, H. T. and Lai, G. Y. (1992). "Automated Precision Measurement of Surface Profile in CAD-directed Inspection." *IEEE Transactions on Robotics and Automation*, 8(2), 268-278.

128. Mercuri, V. and Mirliss, G. (2013). *MassDOT Structures Inspection and Field Report on Highway Bowker NB OFRP: Routine and Special Member Inspection*. Massachusetts Department of Transportation, Boston, MA.
129. Meyer, A. and Marin, P. (2004). "Segmentation of 3D Triangulated Data Points using Edges Constructed with a C1 discontinuous Surface Fitting." *Computer-Aided Design*, 36(13), 1327-1336.
130. Mirliss, G., Mercuri, V. and Matson, R. (2013). *MassDOT Structures Inspection and Field Report on Highway Storrow Drive WB ON RMP: Routine and Special Member Inspection*. Massachusetts Department of Transportation, Boston, MA.
131. Monserrat, O. and Crosetto, M. (2008). "Deformation Measurement using Terrestrial Laser Scanning Data and Least Squares 3D Surface Matching." *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(1), 142-154.
132. Mosalam, K. M., Takhirov, S. M. and Park, S. (2014). "Applications of Laser Scanning to Structures in Laboratory Tests and Field Surveys." *Structural Control and Health Monitoring*, 21(1), 115-134.
133. Nayar, S. K., Ikeuchi, K. and Kanade, T. (1991). "Surface Reflection: Physical and Geometrical Perspectives." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7), 611-634.
134. Nehab, D. and Shilane, P. (2004). "Stratified Point Sampling of 3D Models." *Proceedings of the First Eurographics Conference on Point-Based Graphics*, Aire-la-Ville, Switzerland, Switzerland, 49-56.
135. Niblack, C. W., Gibbons, P. B. and Capson, D. W. (1992). "Generating Skeletons and Centerlines from the Distance Transform." *CVGIP: Graphical Models and Image Processing*, 54(5), 420-437.
136. O'Artista, B. and Beintum, M. (2010). *MassDOT Structures Inspection and Field Report on 047 Casey Overpass: Routine and Special Member Inspection*. Massachusetts Department of Transportation, Boston, MA.
137. O'Artista, B. and Beintum, M. (2010). *Structures Insection Field Report - Routine and Special Member Inspection*. B-16-367, Massachusetts Department of Transportation, Boston, MA.
138. Ohio Department of Transportation. (2010). *Manual of Bridge Inspection*. ORC 5501.47, Ohio Department of Transportation, Columbus, OH.
139. Okorn, B., Xiong, X., Akinci, B. and Huber, D. (2010). "Toward Automated Modeling of Floor Plans." *Proceedings of the Symposium on 3D Data Processing, Visualization and Transmission*, Paris, France, 1-8.

140. Olsen, M. J., Chen, Z., Hutchinson, T. and Kuester, F. (2013). "Optical Techniques for Multiscale Damage Assessment." *Geomatics, Natural Hazards and Risk*, 4(1), 49-70.
141. Olsen, M. J., Cheung, K. F., Yamazaki, Y., Butcher, S., Garlock, M., Yim, S., McGarity, S., Robertson, I., Burgos, L. and Young, Y. L. (2012). "Damage Assessment of the 2010 Chile Earthquake and Tsunami using Terrestrial Laser Scanning." *Earthquake Spectra*, 28(S1), S179-S197.
142. Olsen, M. J. and Kayen, R. (2014). "Post-earthquake and Tsunami 3D Laser Scanning Forensic Investigations." *Proceedings of Forensic Engineering*, San Francisco, CA, 477-486.
143. Olsen, M. J., Kuester, F., Chang, B. J. and Hutchinson, T. C. (2009). "Terrestrial Laser Scanning-based Structural Damage Assessment." *Journal of Computing in Civil Engineering*, 24(3), 264-272.
144. Osada, R., Funkhouser, T., Chazelle, B. and Dobkin, D. (2002). "Shape Distributions." *ACM Transactions on Graphics (TOG)*, 21(4), 807-832.
145. Park, H. S., Lee, H. M., Adeli, H. and Lee, I. (2007). "A New Approach for Health Monitoring of Structures: Terrestrial Laser Scanning." *Computer-Aided Civil and Infrastructure Engineering*, 22(1), 19-30.
146. Pauly, M., Gross, M. and Kobbelt, L. P. (2002). "Efficient Simplification of Point-Sampled Surfaces." *IEEE Visualization, VIS*, Boston, MA, 163-170.
147. Pauly, M., Keiser, R. and Gross, M. (2003). "Multi-scale Feature Extraction on Point-Sampled Surfaces." *Computer Graphics Forum*, 22(3), 281-289.
148. Pfeifer, N. and Briese, C. (2007). "Laser Scanning—principles and Applications." *3rd International Exhibition & Scientific Congress on Geodesy, Mapping, Geology, Geophysics, Cadaster - GEO-SIBERIA*, Novosibirsk, Russia, 93-112.
149. Phares, B. M., Washer, G. A., Rolander, D. D., Graybeal, B. A. and Moore, M. (2004). "Routine Highway Bridge Inspection Condition Documentation Accuracy and Reliability." *Journal of Bridge Engineering*, 9(4), 403-413.
150. Pope, A. R. (1994). *Model-Based Object Recognition*. Technical Report : A Survey of Recent Techniques, Department of Computer Science, University of British Columbia, Vancouver, BC.
151. PROFACTOR. (Accessed in July 2014). "ReconstructMe." <http://reconstructme.net/>.

152. Pu, S. and Vosselman, G. (2006). "Automatic Extraction of Building Features from Terrestrial Laser Scanning." *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5), 25-27.
153. Rabah, M., Elhatab, A. and Fayad, A. (2013). "Automatic Concrete Cracks Detection and Mapping of Terrestrial Laser Scan Data." *NRIAG Journal of Astronomy and Geophysics*, 2(2), 250-255.
154. Rabbani, T., van Den Heuvel, F. and Vosselmann, G. (2006). "Segmentation of Point Clouds Using Smoothness Constraint." *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5), 248-253.
155. Reeb, G. (1946). "On the Singular Points of a Completely Integrable Pfaff Form or of a Numerical Function." *Comptes Rendus de l'Academic des Sciences*, 222, 847-849.
156. Reshetyuk, Y. (2006). "Investigation and Calibration of Pulsed Time-of-flight Terrestrial Laser Scanners." Licentiate in Geodesy, Department of Transport and Economics - Division of Geodesy, Royal Institute of Technology (KTH), Stockholm, Sweden.
157. Rish, I. (2001). "An Empirical Study of the Naive Bayes Classifier." *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, Seattle, WA, 41-46.
158. Rosenfeld, A. and Pfaltz, J. L. (1966). "Sequential Operations in Digital Picture Processing." *Journal of the ACM (JACM)*, 13(4), 471-494.
159. Rowe, J., Razdan, A., Collins, D. and Panchanathan, S. (2001). "A 3D Digital Library System: Capture, Analysis, Query, and Display." *4th International Conference on Digital Libraries (ICADL)*, Bangalore, India, 1-9.
160. Rumelhart, D. E., McClelland, J. L. and Group, P. R. (1995). *Parallel Distributed Processing*. MIT Press, Cambridge, MA.
161. Rusu, R. B. (2009). "Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments." Ph.D. Dissertation, Institute for Informatics, Technical University of Munich, Munich, Germany.
162. Rusu, R. B. and Cousins, S. (2011). "3d is Here: Point Cloud Library (PCL)." *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Sydney, Australia, 1-4.
163. Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M. and Beetz, M. (2008). "Towards 3D Point Cloud-based Object Maps for Household Environments." *Robotics and Autonomous Systems*, 56(11), 927-941.

164. Sagirolu, S. (2012). "Analytical and Experimental Evaluation of Progressive Collapse Resistance of Reinforced Concrete Structures." Ph.D. Dissertation, Civil and Environmental Engineering, Northeastern University, Boston, MA.
165. Sapidis, N. S. and Besl, P. J. (1995). "Direct Construction of Polynomial Surfaces from Dense Range Images Through Region Growing." *ACM Transactions on Graphics (TOG)*, 14(2), 171-200.
166. Sappa, A. D. and Devy, M. (2001). "Fast Range Image Segmentation by an Edge Detection Strategy." *Proceedings of Third International Conference on 3-D Digital Imaging and Modeling*, Quebec City, Canada, 292-299.
167. Sasani, M., Bazan, M. and Sagirolu, S. (2007). "Experimental and Analytical Progressive Collapse Evaluation of an Actual Reinforced Concrete Structure." *ACI Structural Journal*, 104(6), 731-739.
168. Shakarji, C. M. (1998). "Least-squares Fitting Algorithms of the NIST Algorithm Testing System." *Journal of Research-National Institute of Standards and Technology*, 103, 633-641.
169. Shih, N.-J. and Wang, P.-H. (2004). "Point-cloud-based Comparison Between Construction Schedule and As-built Progress: Long-range Three-dimensional Laser Scanner's Approach." *Journal of Architectural Engineering*, 10(3), 98-102.
170. Siringoringo, D. M. and Fujino, Y. (2009). "Non-contact Operational Modal Analysis of Structural Members by Laser Doppler Vibrometer." *Computer-Aided Civil and Infrastructure Engineering*, 24(4), 249-265.
171. Smith, E. R., King, B. J., Stewart, C. V. and Radke, R. J. (2008). "Registration of Combined Range-Intensity Scans: Initialization Through Verification." *Computer Vision and Image Understanding*, 110(2), 226-244.
172. Sobel, I. and Feldman, G. (1968). *A 3x3 Isotropic Gradient Operator for Image Processing*. Stanford Artificial Intelligence Project (SAIL), Stanford University, Stanford, CA.
173. Sohn, H., Farrar, C. R., Hemez, F. M., Shunk, D. D., Stinemates, D. W., Nadler, B. R. and Czarnecki, J. J. (2004). *A Review of Structural Health Monitoring Literature: 1996-2001.*, Los Alamos National Laboratory, Los Alamos, New Mexico.
174. Solina, F. and Bajcsy, R. (1990). "Recovery of Parametric Models from Range Images: The Case for Superquadrics with Global Deformations." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2), 131-147.

175. Son, H. and Kim, C. (2010). "3D Structural Component Recognition and Modeling Method using Color and 3D Data for Construction Progress Monitoring." *Automation in Construction*, 19(7), 844-854.
176. Staiger, R. (2005). "The Geometrical Quality of Terrestrial Laser Scanner (TLS)." *Proceedings of FIG Working Week*, Cairo, Egypt, 1-11.
177. Sternberg, H. (2006). "Deformation Measurements at Historical Buildings with Terrestrial Laser Scanners." *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5), 303-308.
178. Sundar, H., Silver, D., Gagvani, N. and Dickinson, S. (2003). "Skeleton-based Shape Matching and Retrieval." *Proceedings of Shape Modeling International*, Bournemouth, United Kingdom, 130-139.
179. Tang, P., Akinci, B. and Huber, D. (2009). "Quantification of Edge Loss of Laser Scanned Data at Spatial Discontinuities." *Automation in Construction*, 18(8), 1070-1083.
180. Tang, P., Huber, D. and Akinci, B. (2010). "Characterization of Laser Scanners and Algorithms for Detecting Flatness Defects on Concrete Surfaces." *Journal of Computing in Civil Engineering*, 25(1), 31-42.
181. Tang, P., Huber, D., Akinci, B., Lipman, R. and Lytle, A. (2010). "Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques." *Automation in construction*, 19(7), 829-843.
182. Tangelder, J. W. and Veltkamp, R. C. (2004). "A Survey of Content-based 3D Shape Retrieval Methods." *Proceedings of Shape Modeling Applications*, Genova, Italy, 145-156.
183. Teza, G., Galgaro, A. and Moro, F. (2009). "Contactless Recognition of Concrete Surface Damage from Laser Scanning and Curvature Computation." *NDT&E International*, 42(4), 240-249.
184. Torok, M., Fard, M. and Kochersberger, K. (2012). "Post-disaster Robotic Building Assessment: Automated 3D Crack Detection from Image-based Reconstructions." *Proceedings of Computing in Civil Engineering*, ASCE, Clearwater Beach, FL, 397-404.
185. Torok, M. M., Golparvar-Fard, M. and Kochersberger, K. B. (2013). "Image-Based Automated 3D Crack Detection for Post-Disaster Building Assessment." *Journal of Computing in Civil Engineering*, Technical Paper A4014004, 1-13.

186. Trujillo, J. J., Bingöl, F., Larsen, G. C., Mann, J. and Kühn, M. (2011). "Light Detection and Ranging Measurements of Wake Dynamics. Part II: Two-dimensional Scanning." *Wind Energy*, 14(1), 61-75.
187. Turkan, Y., Bosche, F., Haas, C. T. and Haas, R. (2012). "Automated Progress Tracking using 4D Schedule and 3D Sensing Technologies." *Automation in Construction*, 22, 414-421.
188. Vančo, M. and Brunnett, G. (2007). "Geometric Preprocessing of Noisy Point Sets: An Experimental Study." *Computing*, 79(2), 365-380.
189. Varady, T., Martin, R. R. and Cox, J. (1997). "Reverse Engineering of Geometric Models—An Introduction." *Computer-Aided Design*, 29(4), 255-268.
190. Vežočník, R., Ambrožič, T., Sterle, O., Bilban, G., Pfeifer, N. and Stopar, B. (2009). "Use of Terrestrial Laser Scanning Technology for Long Term High Precision Deformation Monitoring." *Sensors*, 9(12), 9873-9895.
191. Vieira, M. and Shimada, K. (2004). "Segmentation of Noisy Laser-Scanner Generated Meshes with Piecewise Polynomial Approximations." *Proceedings of the ASME Design Automation Conference*, Salt Lake City, UT, 1-8.
192. Vieira, M. and Shimada, K. (2005). "Surface Mesh Segmentation and Smooth Surface Extraction Through Region Growing." *Computer-Aided Geometric Design*, 22(8), 771-792.
193. Vieira, M. and Shimada, K. (2009). "Surface Extraction from Point-sampled Data Through Region Growing." *International Journal of CAD/CAM*, 5(1), 1-15.
194. Vosselman, G., Gorte, B. G., Sithole, G. and Rabbani, T. (2004). "Recognising Structure in Laser Scanner Point Clouds." *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 46(8), 33-38.
195. Wahbeh, A. M., Caffrey, J. P. and Masri, S. F. (2003). "A Vision-based Approach for the Direct Measurement of Displacements in Vibrating Systems." *Smart Materials and Structures*, 12(5), 785.
196. Walsh, S. B., Borello, D. J., Guldur, B. and Hajjar, J. F. (2013). "Data Processing of Point Clouds for Object Detection for Structural Engineering Applications." *Computer-Aided Civil and Infrastructure Engineering*, 28(7), 495-508.
197. Weber, C., Hahmann, S. and Hagen, H. (2010). "Sharp Feature Detection in Point Clouds." *IEEE Shape Modeling International Conference (SMI)*, Aix en Provence, France, 175-186.

198. Wikipedia. (Accessed in July 2014). "Bowker Overpass." [http://en.wikipedia.org/wiki/Bowker\\_Overpass](http://en.wikipedia.org/wiki/Bowker_Overpass).
199. Yamazaki, F. (2001). *Applications of Remote Sensing and GIS for Damage Assessment*. Structural Safety and Reliability, Swets & Zeitlinger, Leiden, Netherlands.
200. Yang, C. and Medioni, G. (1992). "Object Modelling by Registration of Multiple Range Images." *Image and Vision Computing*, 10(3), 145-155.
201. Yi, J. H. and Chelberg, D. M. (1998). "Model-based 3d Object Recognition using Bayesian Indexing." *Computer Vision and Image Understanding*, 69(1), 87-105.
202. Yu, Y., Ferencz, A. and Malik, J. (2001). "Extracting Objects from Range and Radiance Images." *IEEE Transactions on Visualization and Computer Graphics*, 7(4), 351-364.
203. Zalama, E., Gómez-García-Bermejo, J., Llamas, J. and Medina, R. (2011). "An Effective Texture Mapping Approach for 3D Models Obtained from Laser Scanner Data to Building Documentation." *Computer-Aided Civil and Infrastructure Engineering*, 26(5), 381-392.
204. Zhan, Q., Yu, L. and Liang, Y. (2010). "A point Cloud Segmentation Method based on Vector Estimation and Color Clustering." *2nd International Conference on Information Science and Engineering (ICISE)*, Hangzhou, China, 3463-3466.
205. Zhang, C. and Chen, T. (2001). "Efficient Feature Extraction for 2D/3D Objects in Mesh Representation." *Proceedings of 2001 International Conference on Image Processing*, Thessaloniki, Greece, 935-938.

## Appendices

## A. Appendix A - Results of Surface Normal Variation Study

This appendix shows the detailed results of the surface normal variation study used for creating the Figure 7.21. The results are given in Table A.1.

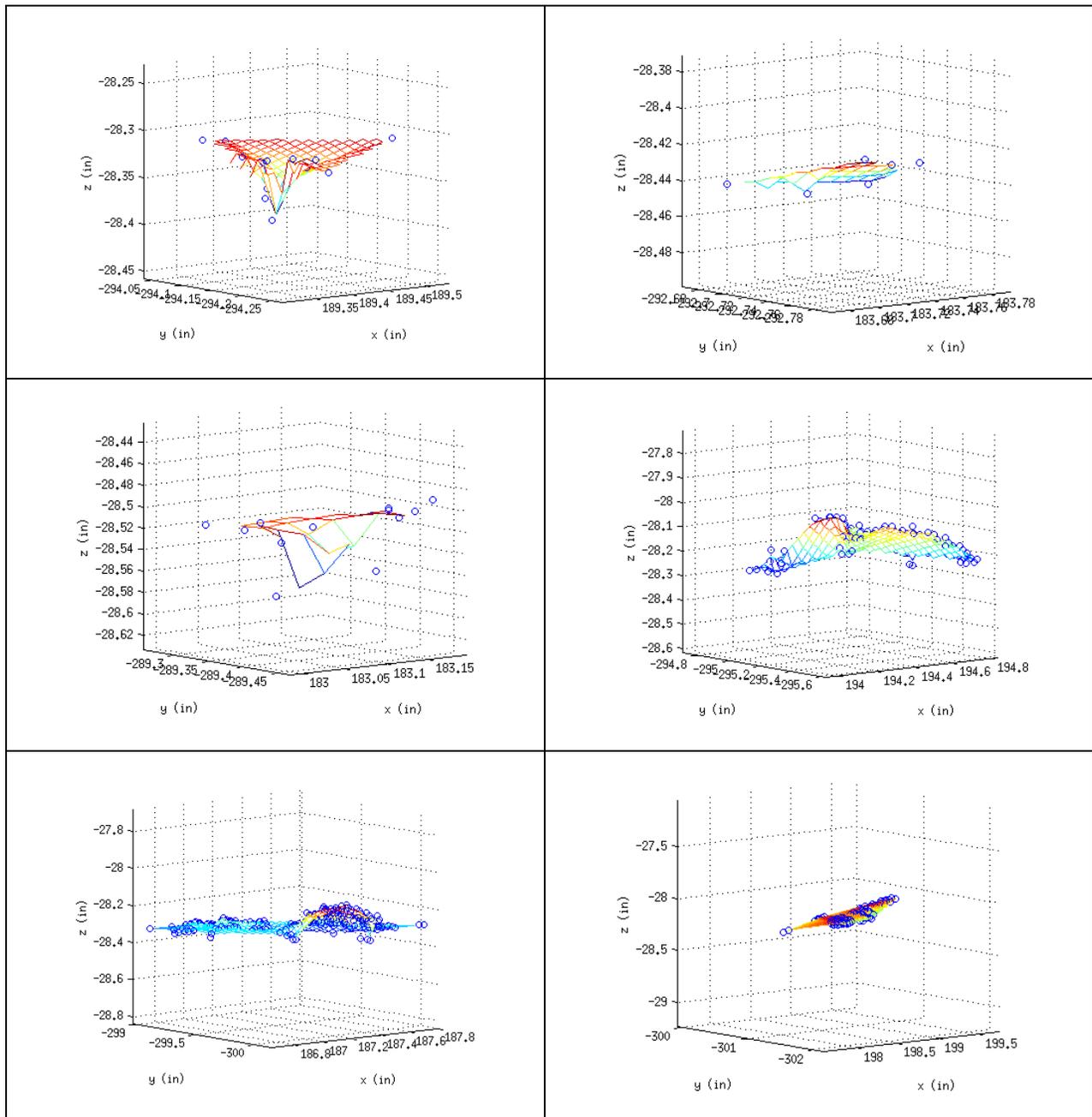
Table A.1: Surface normal variation results for a selected neighborhood.

Point #	Neighborhood Size (k)					
	k = 3	k = 6	k = 8	k = 16	k = 24	k = 32
1	27.0	3.7	5.0	2.9	1.0	0.2
2	25.8	8.2	5.6	2.1	2.0	0.2
3	29.8	9.8	4.6	2.0	1.5	0.2
4	18.2	10.5	3.5	1.3	1.0	0.2
5	72.9	3.0	0.9	0.9	1.0	0.2
6	9.6	1.1	6.6	3.5	1.8	0.2
7	46.7	4.2	1.9	2.7	1.8	0.2
8	7.9	4.4	7.2	1.5	1.7	0.2
9	33.1	7.0	8.1	1.3	0.9	0.2
10	14.2	7.5	4.8	2.1	1.0	0.2
11	33.1	1.1	1.9	2.7	2.5	0.2
12	6.0	1.5	5.4	4.2	1.5	0.2
13	33.0	3.2	3.9	1.8	0.4	0.2
14	12.2	5.3	5.4	2.0	1.8	0.2
15	17.2	11.4	8.1	4.2	1.0	0.2
16	9.3	10.2	6.0	3.8	0.3	0.2
17	61.9	9.7	4.6	3.2	1.0	0.2
18	12.6	8.7	5.5	1.2	0.4	0.2
19	9.8	4.5	1.6	1.4	2.3	0.2
20	10.4	7.0	5.4	1.2	1.6	0.2
21	33.0	10.5	2.9	2.1	1.6	0.2
22	27.4	7.1	4.6	2.5	1.5	0.2
23	26.8	7.2	5.4	3.7	1.5	0.2
24	14.2	4.2	2.7	2.8	0.7	0.2
25	3.7	7.1	5.1	3.3	1.8	0.2
26	34.2	0.3	2.7	3.3	1.8	0.2
27	61.9	11.2	8.5	2.2	1.0	0.2
28	36.3	4.5	1.1	2.7	1.2	0.2
29	36.3	4.5	1.1	2.7	1.2	0.2
30	61.9	9.7	8.5	3.2	1.0	0.2
31	37.7	7.4	4.4	1.3	1.6	0.2
32	3.7	10.2	5.1	3.3	1.8	0.2

## B. Appendix B - Representation of Mesh Grid Results

The Table B.1 presents the result of the application of mesh grid for damage quantification performed on a portion of the DeKalb County Bridge. The individual defect locations, for which the results are given in Table B.1, are shown in Figure 7.27.

Table B.1: Mesh grid results for defects D2-D7.



## C. Appendix C - Concrete Testing Frame Damage Measurements

In this appendix, the measurements of the labelled surface cracks and spalled concrete regions for the concrete testing frame are given in Table C.1. The description of the concrete testing frame is provided in Section 10.5 in detail. The first column of the table represents the location of each defect. The necessary explanations for the labels are given in Section 10.5. The labelled images that show each defect represented in Table C.1 are later displayed in Table C.2.

For cracks, the length measurement is a straight line from the start to the end of each crack. Thus, the measured length values might be smaller than the actual length of the crack, where the crack has a curved shape. The recorded crack thickness values always represent the widest opening along a crack. The measurements for the labelled cracks are given in Table C.1.

Figure C.1 represents the reference measurement drawings for two types of spalling: triangular and rectangular spalling. For spillings, the measurements for each number pair, where each number indicates a corner, are recorded. The measurements for the spalled concrete regions are given in Table C.1.



Figure C.1: Reference measurement drawings for spalled concrete regions.

The measurements for hairline cracks (and two exceptions for spalling: E0-1-S1 and E0-2-S1) are not recorded.

Table C.1: Measurements for labelled crack spalling locations for concrete testing frame.

Label	Crack Label	Spalling Label	Crack		Spalling			
			Thickness (in)	Length (in)	1-2(in)	2-3 (in)	3-1 (in)	
					1-2(in)	2-3 (in)	3-4(in)	4-1(in)
A0-1	A0-1-C1		0.08	1.16				
	A0-1-C2		0.12	1.14				
	A0-1-C3		0.07	0.36				
		A0-1-S1			0.54	0.72	0.87	
A0-2	A0-2-C1		0.04	0.70				
	A0-2-C2		0.03	0.54				
	A0-2-C3		0.06	0.47				
	A0-2-C4		0.06	0.88				
	A0-2-C5		0.05	0.04				
		A0-2-S1			0.56	0.68	0.21	
		A0-2-S2			0.27	2.07	0.58	2.14
A0-3		A0-3-S1			0.64	2.16	0.83	2.05
A0-4	A0-4-C1		0.03	1.51				
		A0-4-S1			0.22	1.78	0.21	1.82
A1-1	A1-1-C1		0.10	1.78				
	A1-1-C2		0.03	0.92				
	A1-1-C3		0.02	1.31				
		A1-1-S1			1.06	0.76	0.71	
		A1-1-S2			1.24	0.49	0.69	0.90
A1-4	A1-4-C1		0.04	1.02				
	A1-4-C2		0.02	1.00				
	A1-4-C3		0.03	1.25				
	A1-4-C4		0.12	1.05				
	A1-4-C5		0.06	1.02				
		A1-4-S1			0.55	0.78	0.51	
A1-6R	A1-6R-C1		0.02	1.89				
		A1-6R-S1			0.49	2.13	0.22	2.09
A2-1	A2-1-C1		0.03	1.48				
	A2-1-C2		0.02	1.08				
A2-4	A2-4-C1		0.03	1.49				
	A2-4-C2		0.02	1.05				
A2-5R	A2-5R-C1		0.03	2.05				
AB1-1		AB1-1-S1			1.11	1.15	0.49	
AB1-4	AB1-4-C1		0.94	1.86				
AB1-5	AB1-5-C1		0.43	2.17				
B1-1	B1-1-C1		0.02	1.28				
	B1-1-C2		0.03	1.53				

	B1-1-C3		0.07	1.79				
	B1-1-C4		0.14	1.56				
	B1-1-C5		0.06	1.34				
	B1-1-C6		0.04	0.73				
	B1-1-S1				1.39	1.13	0.54	
<b>B1-4</b>	B1-4-C1		0.05	1.47				
	B1-4-C2		Hairline					
	B1-4-C3		0.27	1.39				
	B1-4-C4		0.07	1.64				
	B1-4-S1				0.82	1.09	0.60	
<b>B1-5R</b>	B1-5R-C1		0.03	2.07				
	B1-5R-C2		0.07	2.04				
<b>B1-6R</b>	B1-6R-C1		0.29	2.07				
	B1-6R-C2		0.07	2.07				
	B1-6R-C3		0.03	1.75				
<b>B1-6L</b>	B1-6L-S1				1.11	2.04	1.38	2.05
<b>B2-1</b>	B2-1-C1		0.05	1.64				
	B2-1-C2		0.09	1.89				
	B2-1-C3		0.12	1.79				
	B2-1-C4		0.03	1.52				
	B2-1-C5		0.04	1.58				
	B2-1-S1				1.95	1.62	0.52	
<b>B2-4</b>	B2-4-C1		0.05	2.03				
	B2-4-C2		0.13	1.87				
	B2-4-C3		0.09	1.77				
		B2-4-S1				0.84	0.52	0.74
<b>B2-5R</b>	B2-5R-C1		0.09	2.15				
	B2-5R-C2		0.14	2.24				
	B2-5R-C3		0.09	1.73				
	B2-5R-C4		0.04	2.17				
<b>B3-1</b>	B3-1-C1		0.03	1.34				
	B3-1-C2		0.03	1.76				
		B3-1-S1				1.89	1.81	1.11
<b>B3-4</b>	B3-4-C1		0.05	1.51				
	B3-4-C2		Hairline					
		B3-4-S1				1.88	1.77	1.47
<b>B3-5</b>	B3-5-C1		0.05	2.07				
		B3-5-S1				1.18	2.08	1.50
<b>B3-6R</b>		B3-6R-S1			1.06	2.08	1.19	2.04
<b>BC1-1</b>		BC1-1-S1			2.65	1.65	2.26	2.63
<b>BC1-4</b>		BC1-4-S1			2.23	2.35	2.73	1.85

<b>BC1-5</b>		BC1-5-S1			2.64	2.18	1.80	2.08	
<b>BC1-6</b>		BC1-6-S1			2.55	2.13	1.69	2.19	
<b>BC2-1</b>		BC2-1-S1			2.54	0.95	2.33	1.83	
<b>BC2-4</b>		BC2-4-S1			2.15	2.44	1.37		
<b>BC2-5</b>	BC2-5-C1		0.02	1.01					
	BC2-5-C2		0.23	1.27					
		BC2-5-S1			1.83	2.17	1.36	2.10	
<b>BC2-6</b>		BC2-6-S1			1.64	2.08	1.45	2.13	
<b>C1-1</b>	C1-1-C1		0.03	1.75					
	C1-1-C2		0.09	1.96					
	C1-1-C3		0.19	1.94					
	C1-1-C4		0.12	1.90					
	C1-1-C5		0.04	1.65					
	C1-1-C6		0.02	1.75					
			C1-1-S1			1.05	1.98	2.09	
		C1-1-S2			1.81	1.11	1.90		
<b>C1-4</b>	C1-4-C1		0.11	1.36					
	C1-4-C2		0.41	1.58					
	C1-4-C3		0.22	1.78					
	C1-4-C4		0.10	1.69					
			C1-4-S1			1.79	1.27	1.68	
		C1-4-S2			1.84	0.98	1.94		
<b>C1-6R</b>	C1-6R-C1		0.16	2.10					
			C1-6R-S1			1.08	2.03	1.19	2.03
<b>C1-6L</b>	C1-6L-C1		0.07	2.06					
	C1-6L-C2		0.22	2.10					
			C1-6L-S1			1.05	2.02	1.09	2.09
<b>C2-1</b>	C2-1-C1		0.14	1.77					
	C2-1-C2		0.09	1.65					
	C2-1-C3		0.12	1.54					
	C2-1-C4		0.10	1.90					
	C2-1-C5		0.09	1.46					
			C2-1-S1			2.29	1.23	1.80	
		C2-1-S2			2.08	1.43	1.89		
<b>C2-4</b>	C2-4-C1		0.17	1.81					
	C2-4-C2		0.09	1.87					
	C2-4-C3		0.11	2.26					
			C2-4-S1			1.35	1.94	2.03	
			C2-4-S2			1.23	2.03	1.96	
<b>C2-6R</b>		C2-6R-S1			1.45	2.11	1.29	2.13	
<b>C2-6L</b>	C2-6L-C1		0.08	2.10					

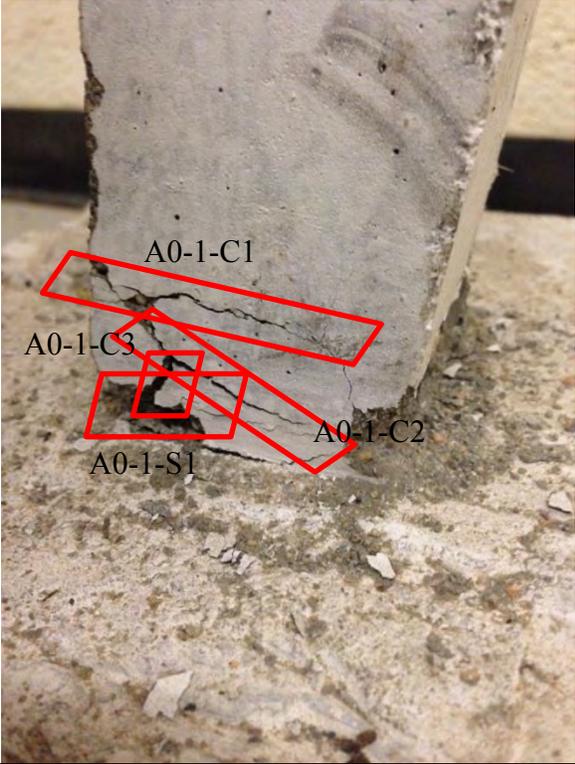
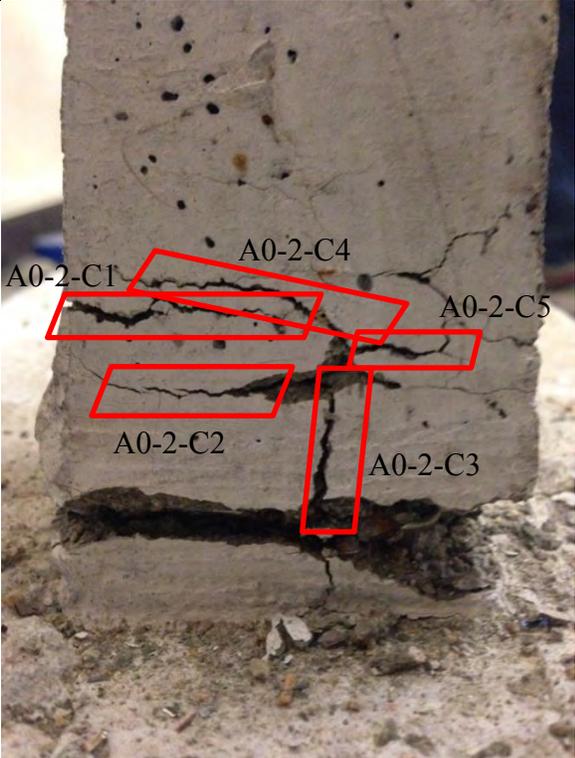
	C2-6L-C2		0.10	2.04				
		C2-6L-S1			1.243	2.160	1.248	2.134
<b>C3-1</b>	C3-1-C1		0.06	1.66				
	C3-1-C2		0.11	1.99				
		C3-1-S1			2.19	0.92	1.82	1.25
		C3-1-S2			1.86	0.99	1.79	
<b>C3-4</b>	C3-4-C1		0.04	1.88				
	C3-4-C2		0.06	1.90				
		C3-4-S1			2.02	1.00	1.91	
		C3-4-S2			1.83	0.96	1.82	
<b>C3-5</b>		C3-5-S1			2.79	2.08	2.83	2.08
<b>C3-6R</b>		C3-6R-S1			1.00	2.07	1.01	2.04
<b>C3-6L</b>	C3-6L-C1		0.08	2.08				
		C3-6L-S1			1.01	2.10	0.90	2.12
<b>CD1-1</b>		CD1-1-S1			2.19	3.17	3.45	2.45
<b>CD1-4</b>		CD1-4-S1			3.13	2.43	2.14	2.06
<b>CD1-5</b>		CD1-5-S1			2.05	2.13	2.45	2.19
<b>CD1-6</b>		CD1-6-S1			3.17	2.10	2.47	2.23
<b>CD2-1</b>		CD2-1-S1			2.16	1.69	2.75	1.68
<b>CD2-4</b>		CD2-4-S1			2.39	2.01	2.24	
<b>CD2-5</b>		CD2-5-S1			2.24	2.09	1.68	2.10
<b>CD2-6</b>		CD2-6-S1			1.17	2.05	1.46	2.02
<b>D0-1</b>	D0-1-C1		0.10	1.45				
	D0-1-C2		0.07	1.24				
	D0-1-C3		0.04	1.12				
	D0-1-C4		0.04	1.52				
<b>D0-2</b>	D0-2-C1		0.05	0.97				
		D0-2-S1			1.29			
<b>D0-3</b>	D0-3-C1		0.05	2.11				
	D0-3-C2		0.08	1.68				
	D0-3-C3		0.09	1.09				
	D0-3-C4		0.07	1.96				
<b>D0-4</b>	D0-4-C1		0.10	1.04				
		D0-4-S1			0.79	1.87		1.93
<b>D1-1</b>	D1-1-C1		0.04	1.56				
	D1-1-C2		0.02	1.78				
	D1-1-C3		Hairline					
	D1-1-C4		0.05	1.57				
	D1-1-C5		0.05	1.18				
	D1-1-C6		0.06	1.71				
<b>D1-4</b>	D1-4-C1		0.09	0.44				

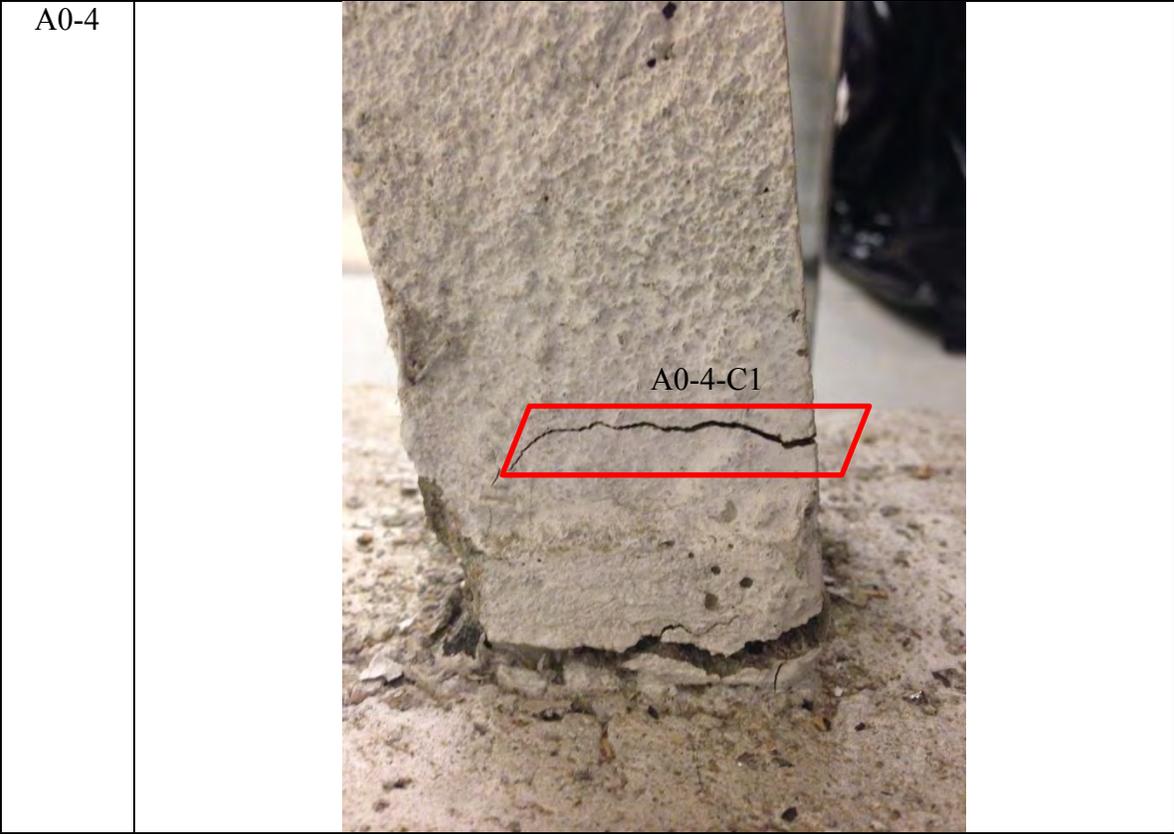
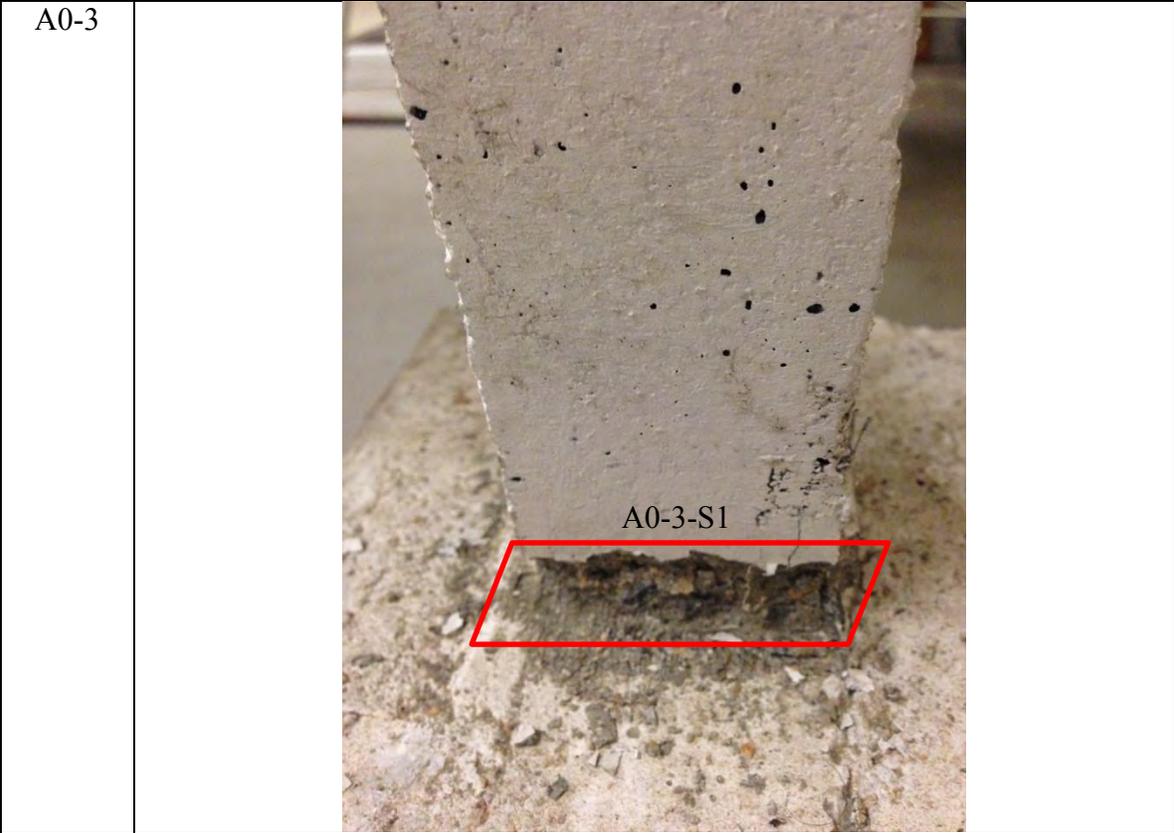
	D1-4-C2		0.03	1.64					
	D1-4-C3		0.02	1.30					
	D1-4-C4		0.04	1.51					
	D1-4-C5		0.16	1.50					
<b>D1-5L</b>	D1-5L-C1		0.05	2.09					
	D1-5L-C2		0.03	2.09					
<b>D1-6R</b>		D1-6R-S1			0.10	1.94	0.90	2.04	
<b>D1-6L</b>	D1-6L-C1		0.03	2.06					
	D1-6L-C2		0.06	2.09					
	D1-6L-C3		0.04	2.03					
<b>D2-1</b>	D2-1-C1		Hairline						
	D2-1-C2		0.09	1.47					
	D2-1-C3		0.02	1.00					
	D2-1-C4		0.06	1.39					
	D2-1-C5		0.04	1.54					
	D2-1-C6		0.18	1.59					
	D2-1-C7		0.06	1.33					
<b>D2-4</b>	D2-4-C1		0.07	1.81					
	D2-4-C2		0.02	1.57					
	D2-4-C3		0.06	0.89					
	D2-4-C4		0.03	1.44					
	D2-4-C5		0.03	1.46					
<b>D2-5L</b>	D2-5L-C1		0.07	2.18					
	D2-5L-C2		0.20	2.25					
	D2-5L-C3		0.25	2.07					
	D2-5L-C4		0.03	2.02					
	D2-5L-C5		0.04	0.83					
<b>D3-1</b>	D3-1-C1		0.27	1.61					
	D3-1-C2		0.03	1.33					
	D3-1-C3		0.05	1.27					
	D3-1-C4		0.08	1.45					
	D3-1-C5		0.02	1.47					
		D3-1-S1				1.22	1.36	0.67	
		D3-1-S2				1.20	1.58	1.67	
<b>D3-4</b>	D3-4-C1		0.04	0.68					
	D3-4-C2		0.07	1.37					
	D3-4-C3		0.02	1.33					
	D3-4-C4		0.05	1.33					
	D3-4-C5		0.06	0.62					
	D3-4-C6		0.25	1.92					
		D3-4-S1				1.53	1.38	1.32	

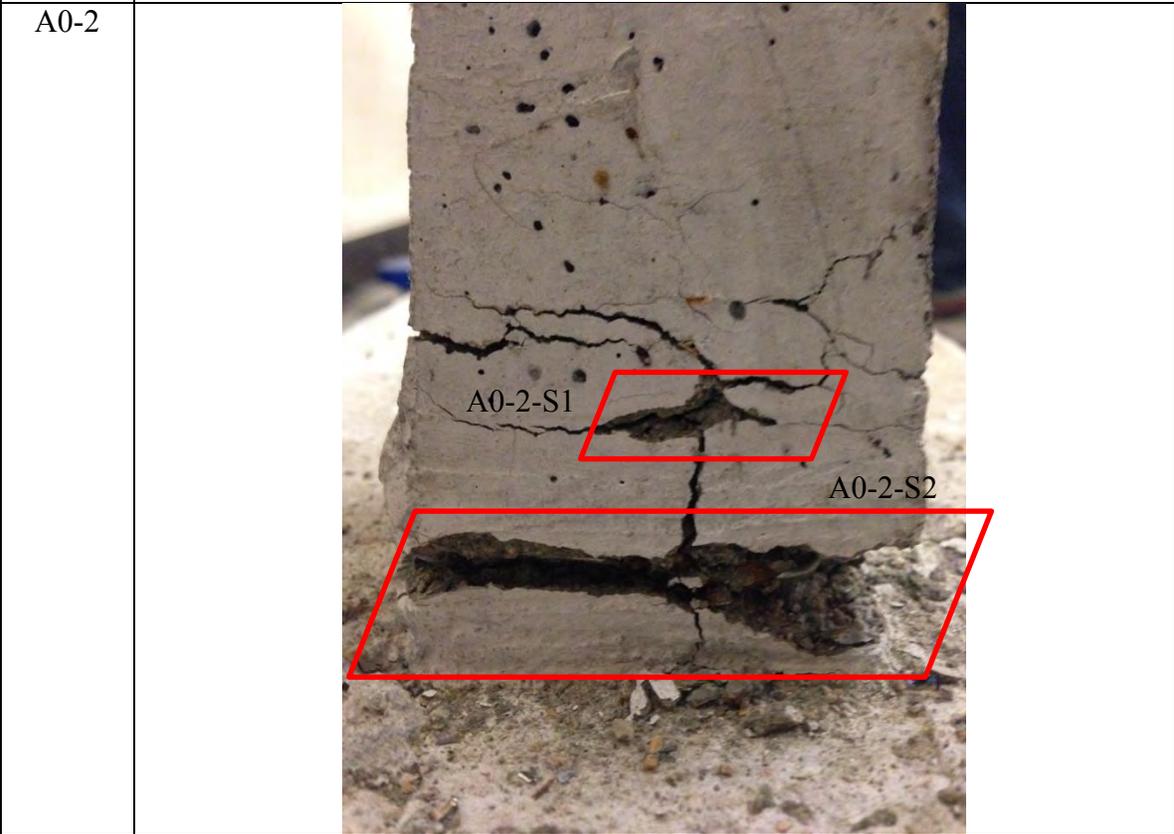
		D3-4-S2			1.68	1.05	1.62		
<b>D3-5</b>	D3-5-C1		0.05	0.83					
	D3-5-C2		0.23	0.89					
	D3-5-C3		0.33	2.16					
		D3-5-S1			1.32	2.14	0.67	2.05	
<b>D3-6R</b>	D3-6R-C1		0.09	2.06					
	D3-6R-C2		0.11	2.08					
<b>D3-6L</b>		D3-6L-S1			1.56	2.08	1.58	2.08	
<b>DE1-1</b>	DE1-1-C1		0.28	1.21					
<b>DE1-4</b>	DE1-4-C1		0.25	1.71					
	DE1-4-C2		0.16	1.13					
<b>DE1-5</b>	DE1-5-C1		0.09	2.16					
	DE1-5-C2		0.03	1.84					
<b>E0-1</b>	E0-1-C1		Hairline						
	E0-1-C2		Hairline						
	E0-1-C3		0.12	1.43					
	E0-1-C4		0.34	1.28					
		E0-1-S1					0.54		
<b>E0-2</b>		E0-2-S1						2.04	
<b>E0-3</b>	E0-3-C1		Hairline						
	E0-3-C2		0.04	1.73					
	E0-3-C3		0.33	2.05					
	E0-3-C4		0.17	1.92					
<b>E0-4</b>	E0-4-C1		Hairline						
	E0-4-C2		0.18	1.06					
	E0-4-C3		0.13	1.53					
<b>E1-1</b>	E1-1-C1		0.04	0.98					
	E1-1-C2		0.08	0.95					
	E1-1-C3		0.06	1.28					
	E1-1-C4		Hairline						
		E1-1-S1				0.80	1.47	0.84	
		E1-1-S2				0.50	0.85	0.79	
<b>E1-3</b>	E1-3-C1		0.03	0.92					
	E1-3-C2		0.05	0.48					
	E1-3-C3		0.39	1.49					
	E1-3-C4		0.07	0.65					
<b>E1-4</b>		E1-4-S1			1.48	2.14	0.72	2.13	
<b>E2-1</b>	E2-1-C1		Hairline						
	E2-1-C2		0.10	1.63					
	E2-1-C3		0.19	1.28					
	E2-1-C4		0.03	0.70					

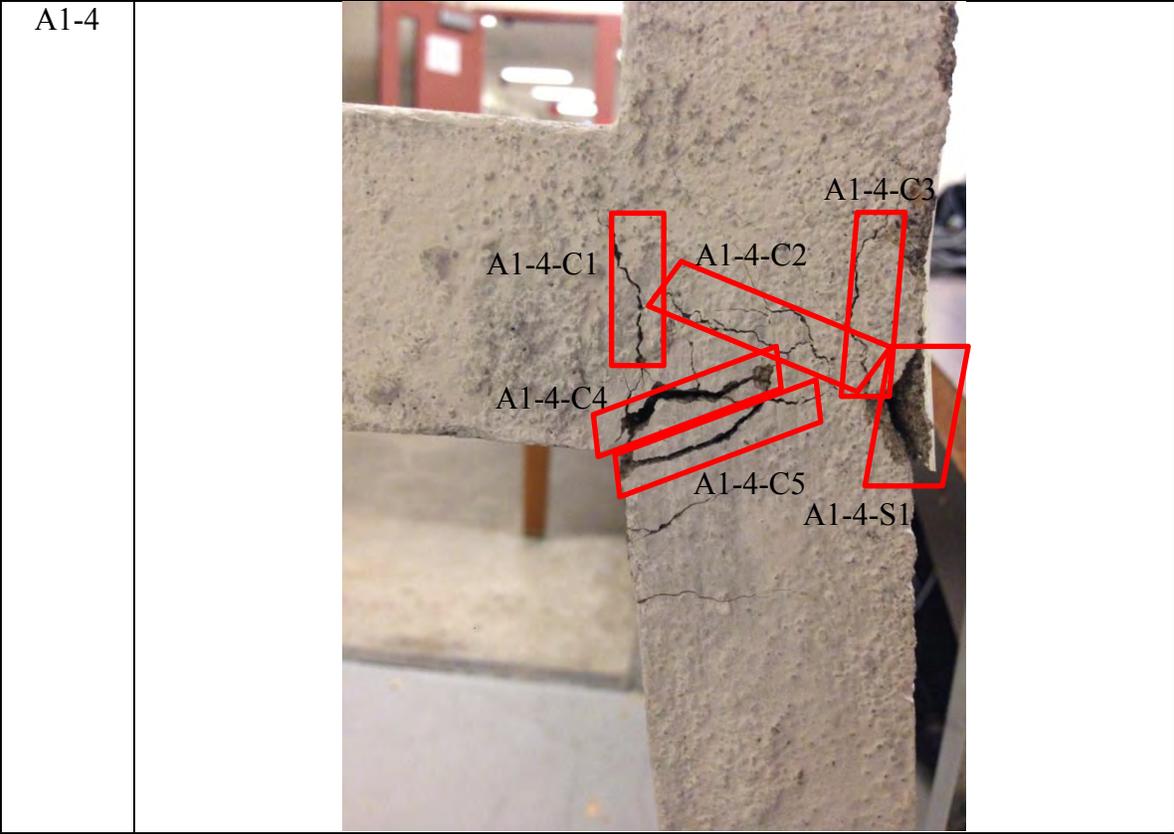
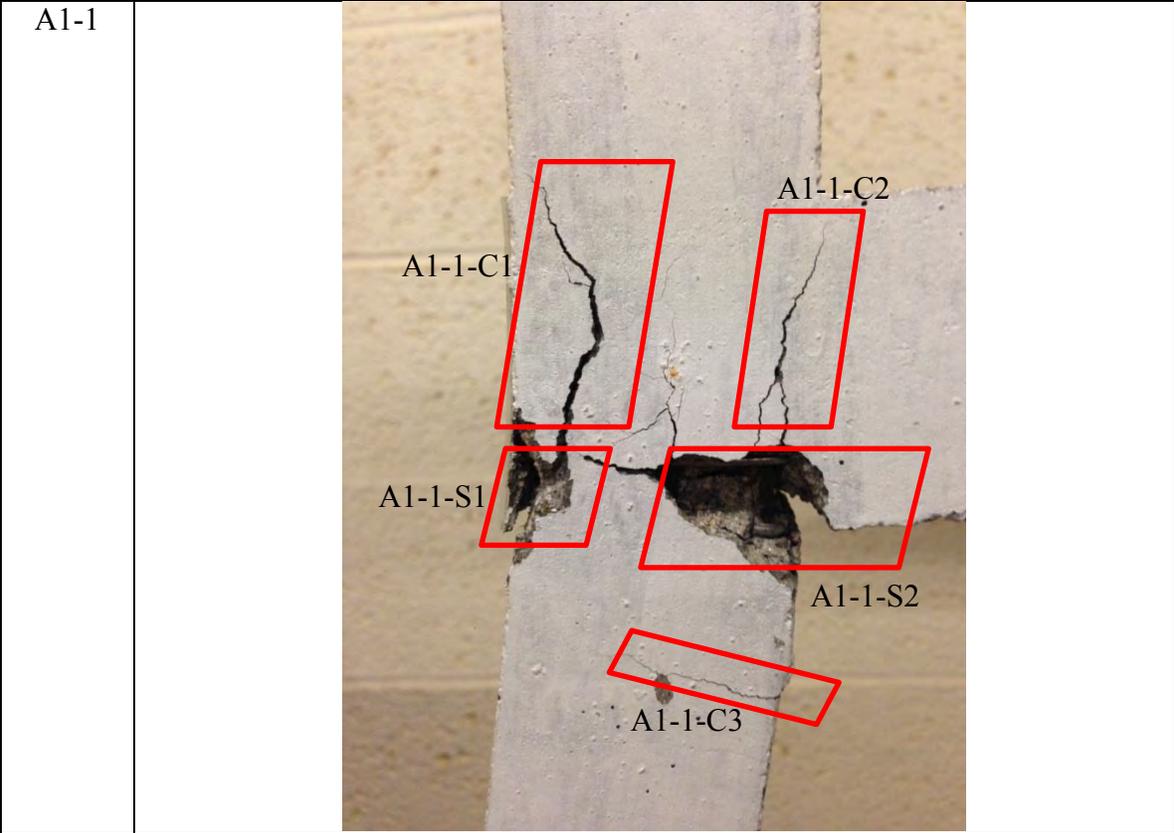
	E2-1-C5		Hairline					
		E2-1-S1			0.66	1.15	0.57	
<b>E2-3</b>		E2-3-S1			1.19	2.17	0.89	2.13
<b>E2-4</b>	E2-4-C1		0.11	1.59				
<b>E2-5L</b>	E2-5L-C1		Hairline					
	E2-5L-C2		0.11	2.14				
<b>E3-1</b>	E3-1-C1		0.06	1.57				
	E3-1-C2		0.06	1.61				
<b>E3-3</b>	E3-3-C1		Blocked					
	E3-3-C2		0.05	2.11				
<b>E3-4</b>	E3-4-C1		0.07	1.62				
	E3-4-C2		0.09	1.56				

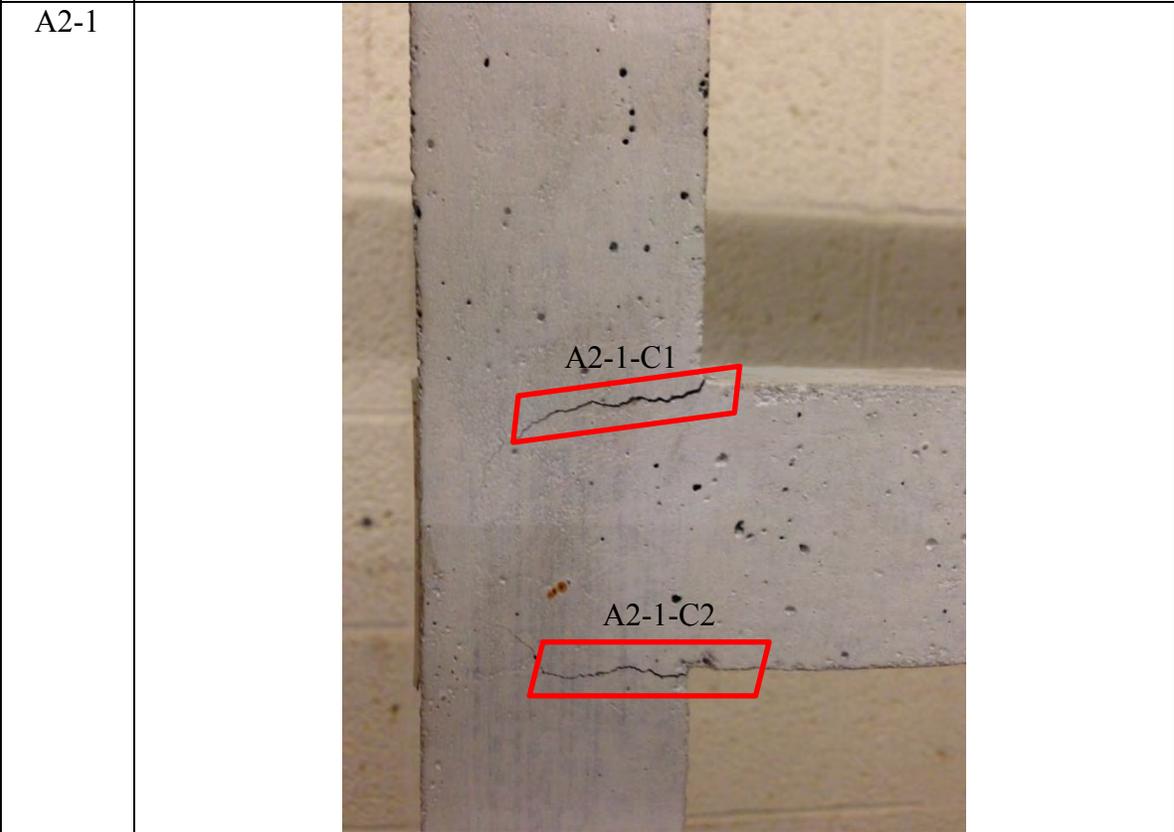
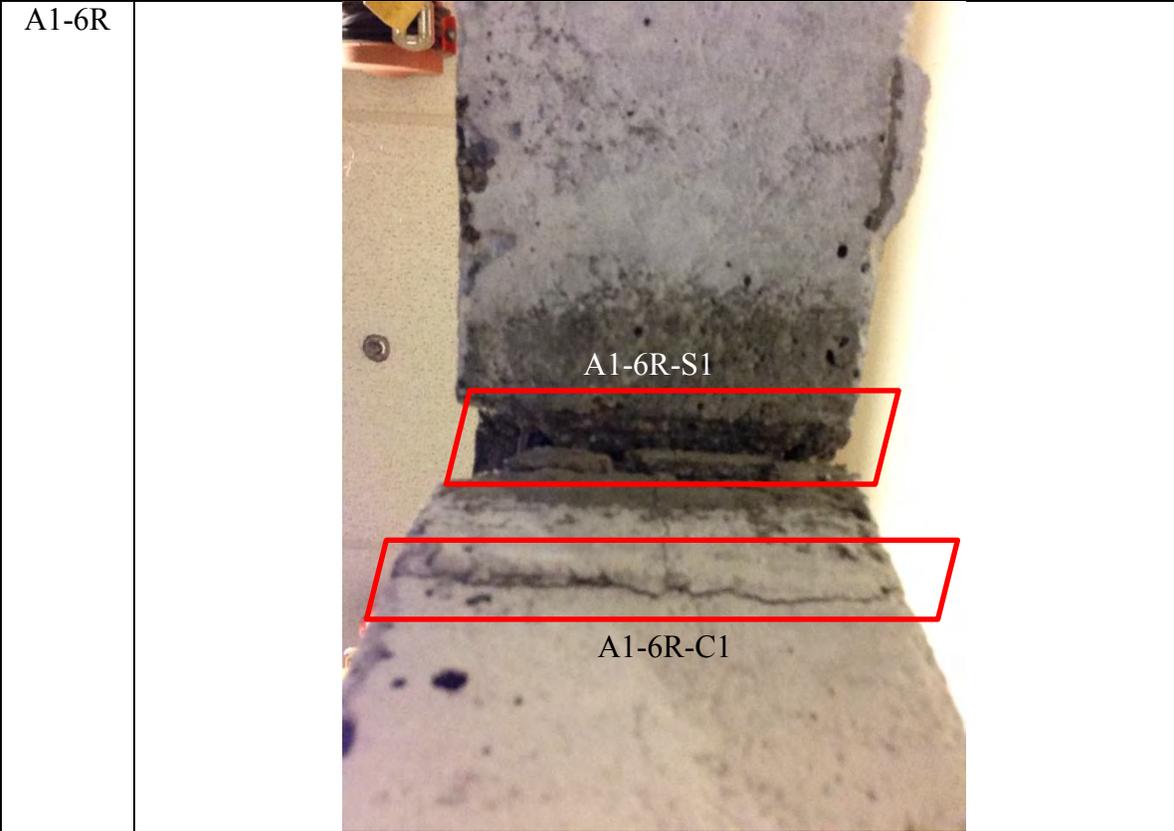
Table C.2: Images of labelled crack and spalling locations for concrete testing frame.

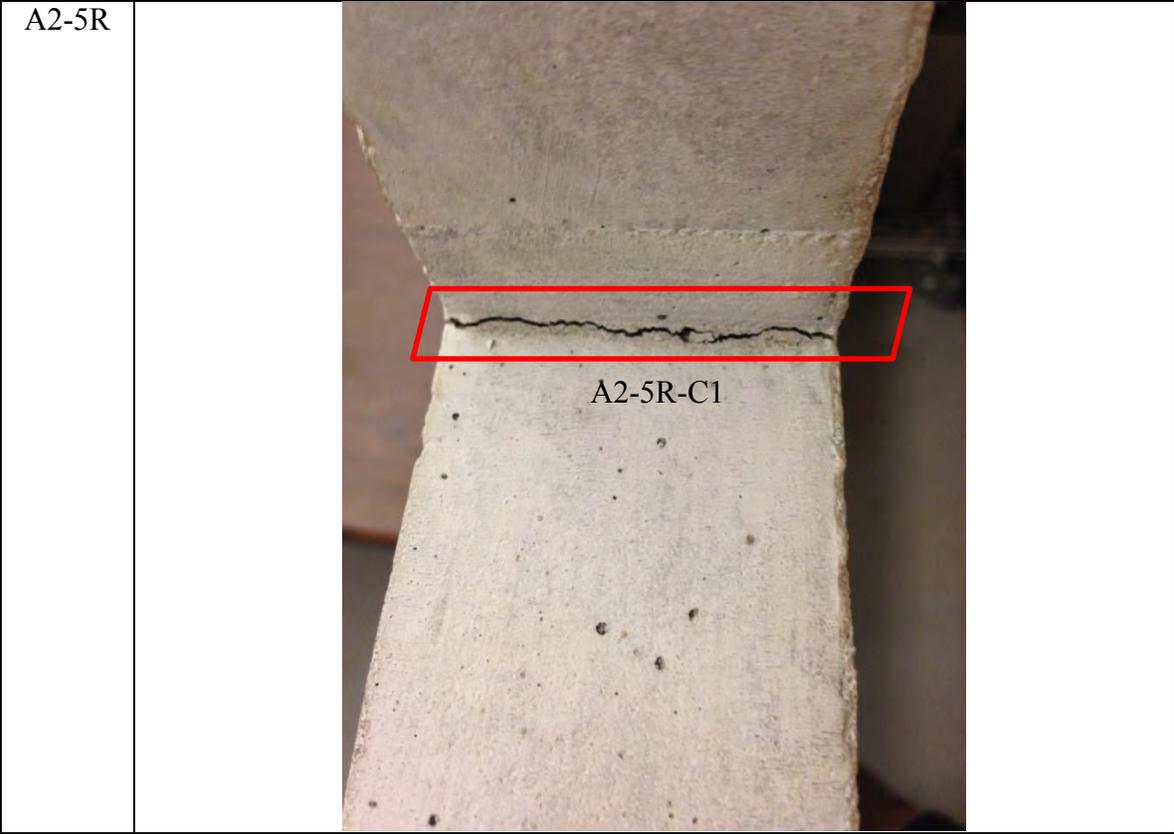
A0-1	 <p>A photograph of a concrete specimen labeled A0-1. The specimen is a vertical cylinder with a rough, weathered surface. Several cracks and areas of spalling are highlighted with red rectangular boxes. The labels are: A0-1-C1 (top crack), A0-1-C3 (middle crack), A0-1-S1 (spalling area), and A0-1-C2 (bottom crack).</p>
A0-2	 <p>A photograph of a concrete specimen labeled A0-2. The specimen is a vertical cylinder with a rough, weathered surface. Several cracks are highlighted with red rectangular boxes. The labels are: A0-2-C1 (top crack), A0-2-C4 (top crack), A0-2-C5 (top crack), A0-2-C2 (middle crack), and A0-2-C3 (bottom crack).</p>









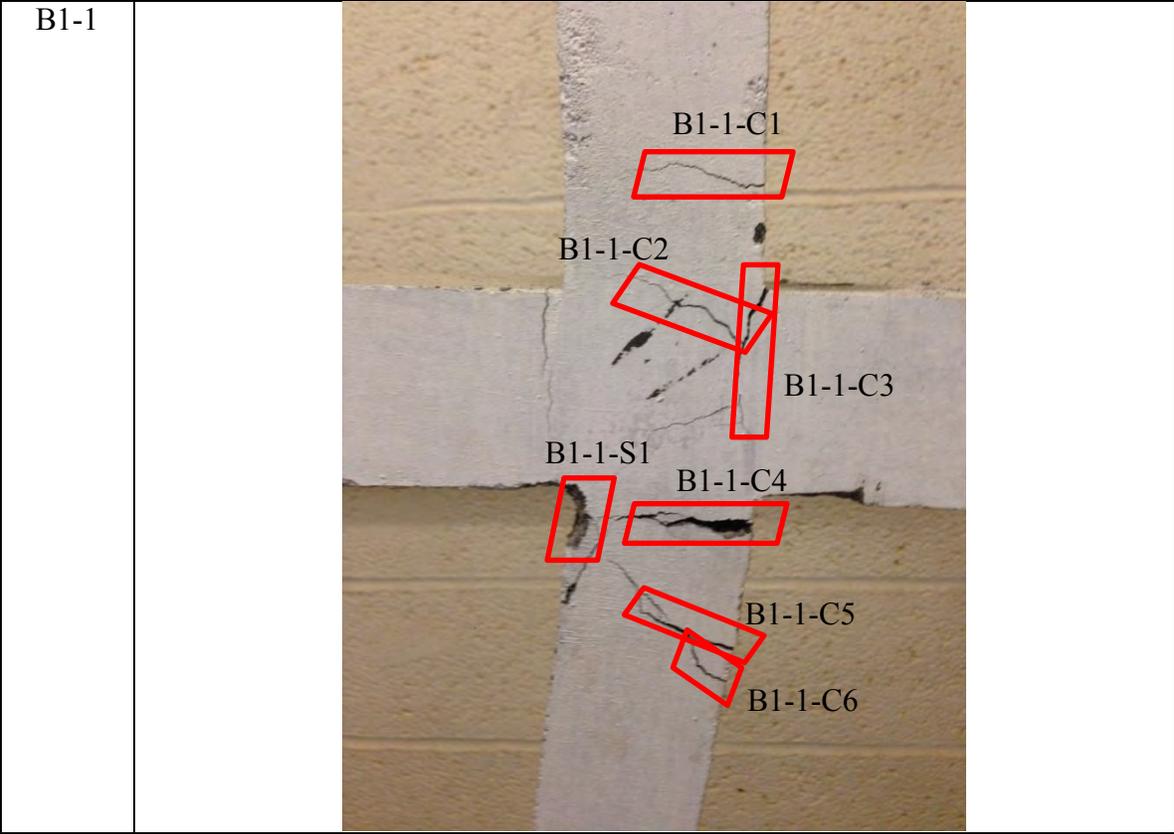
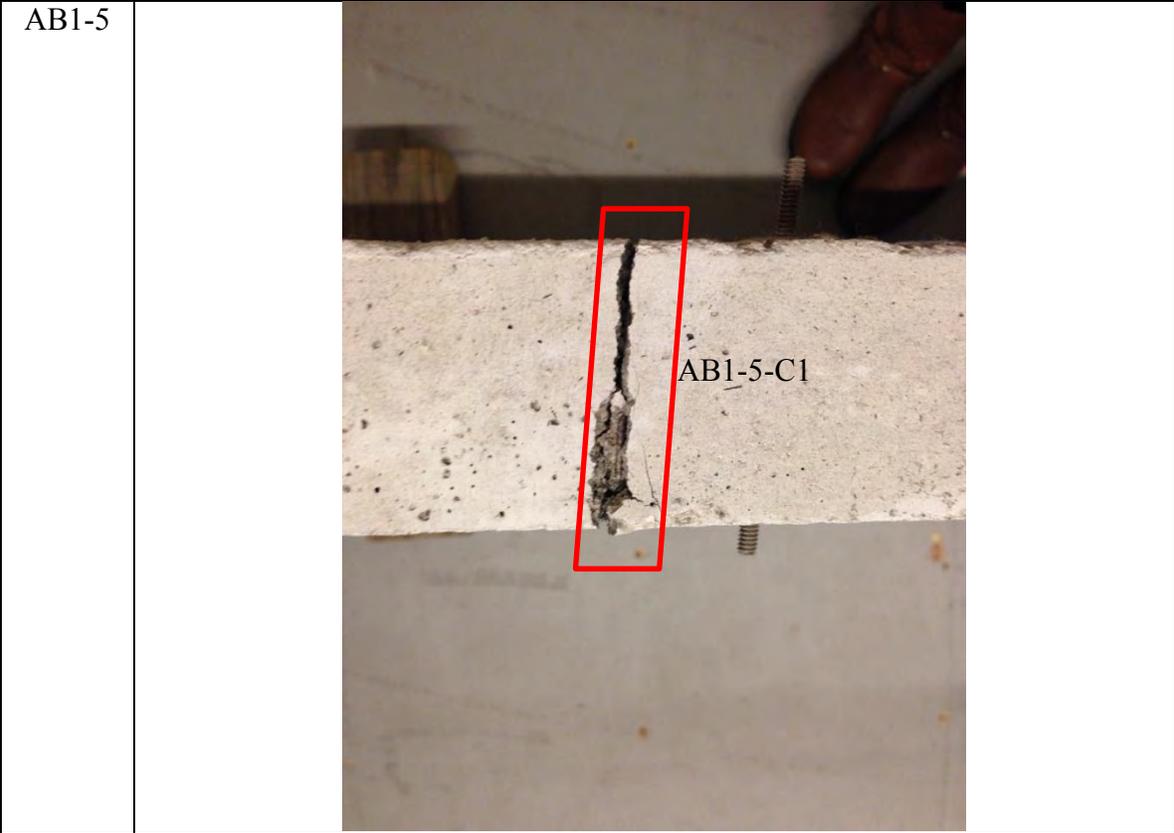


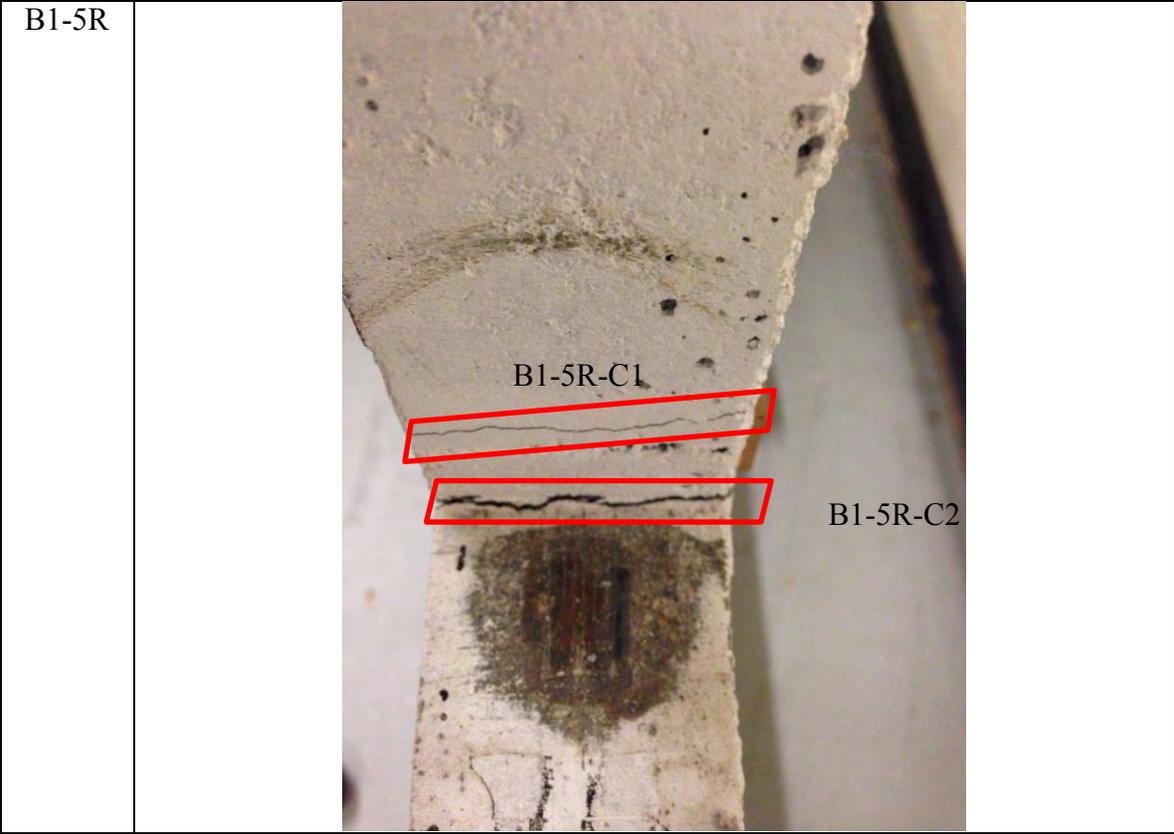
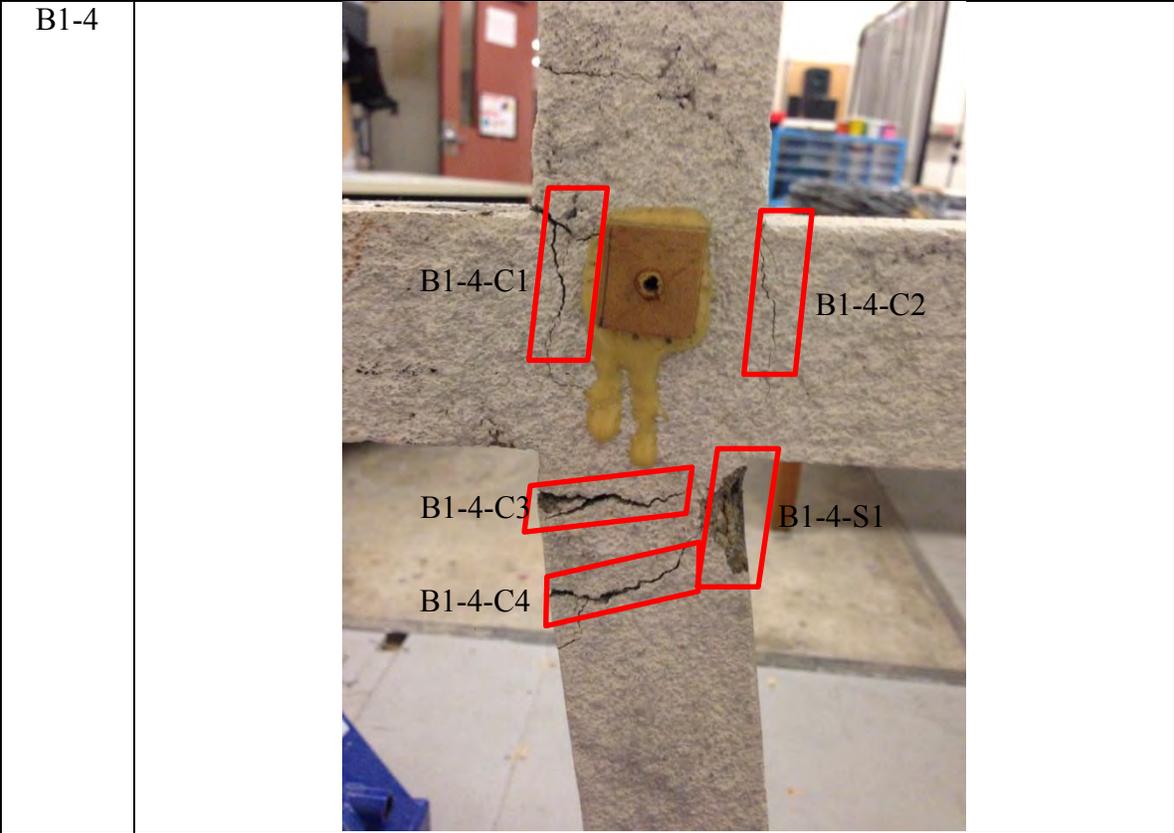
AB1-1

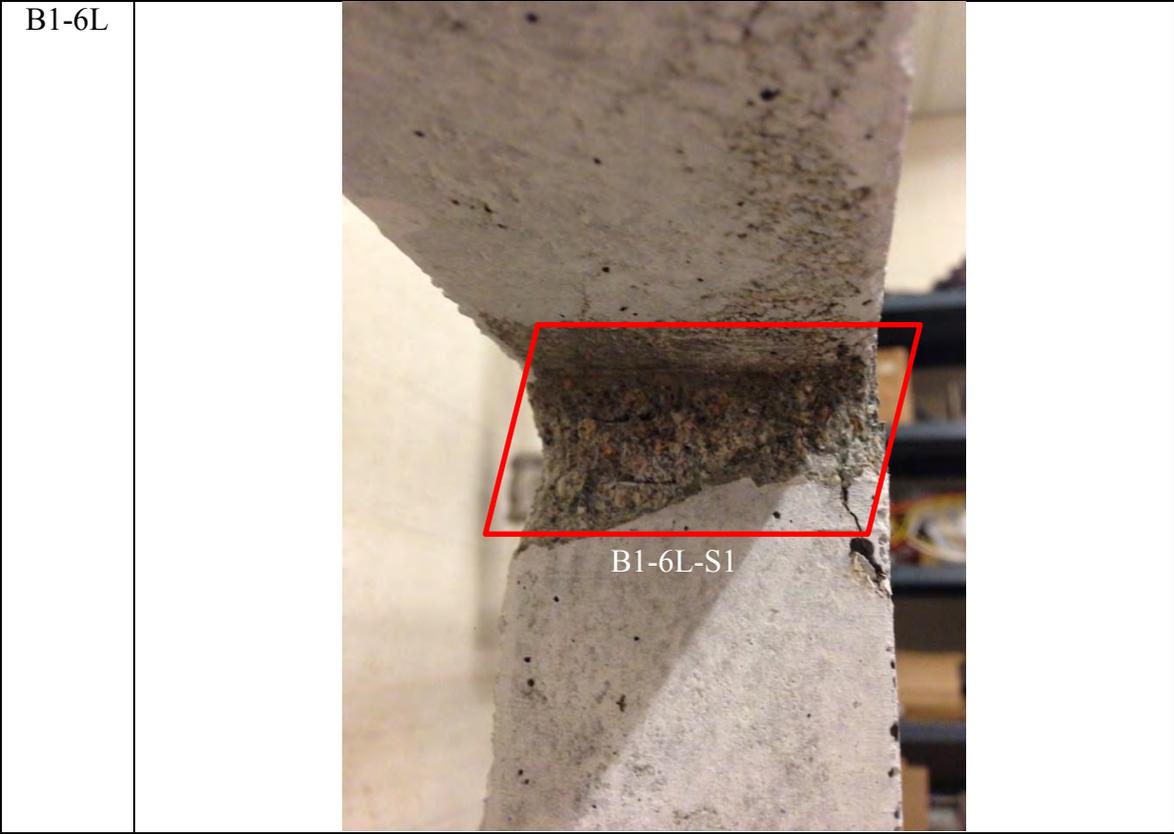
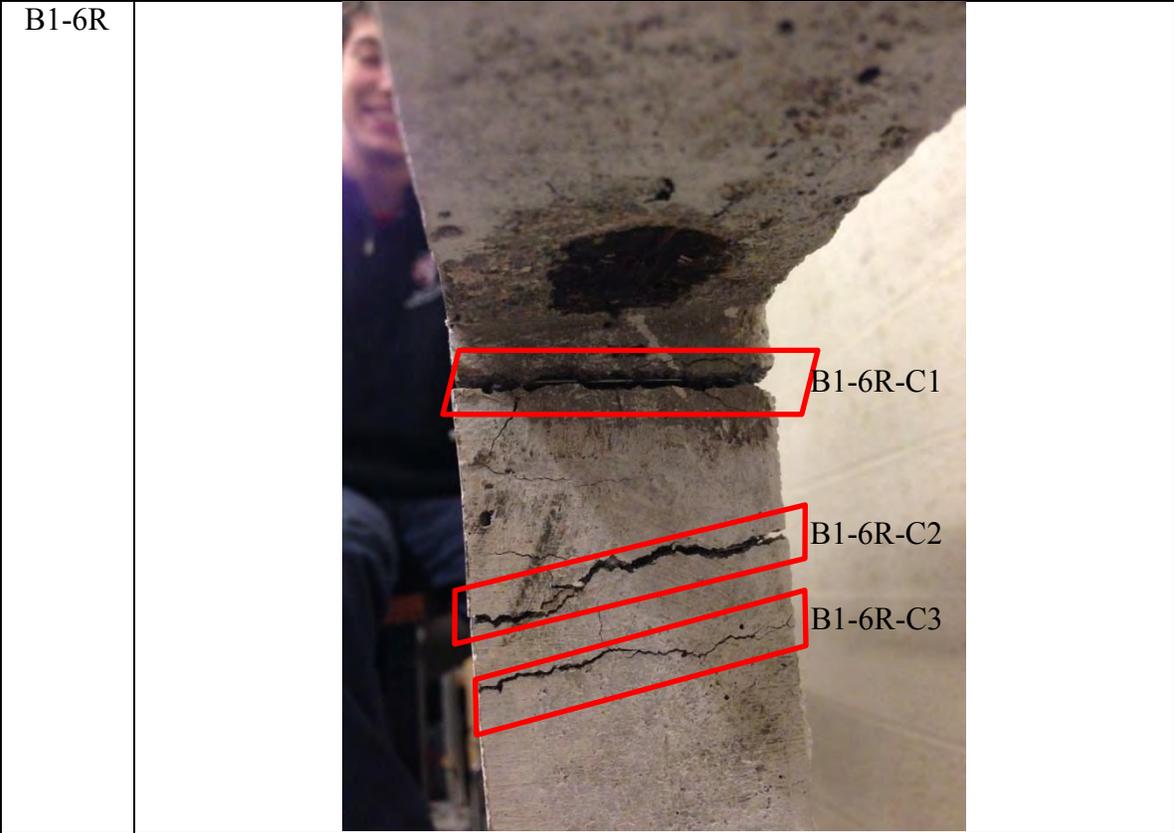


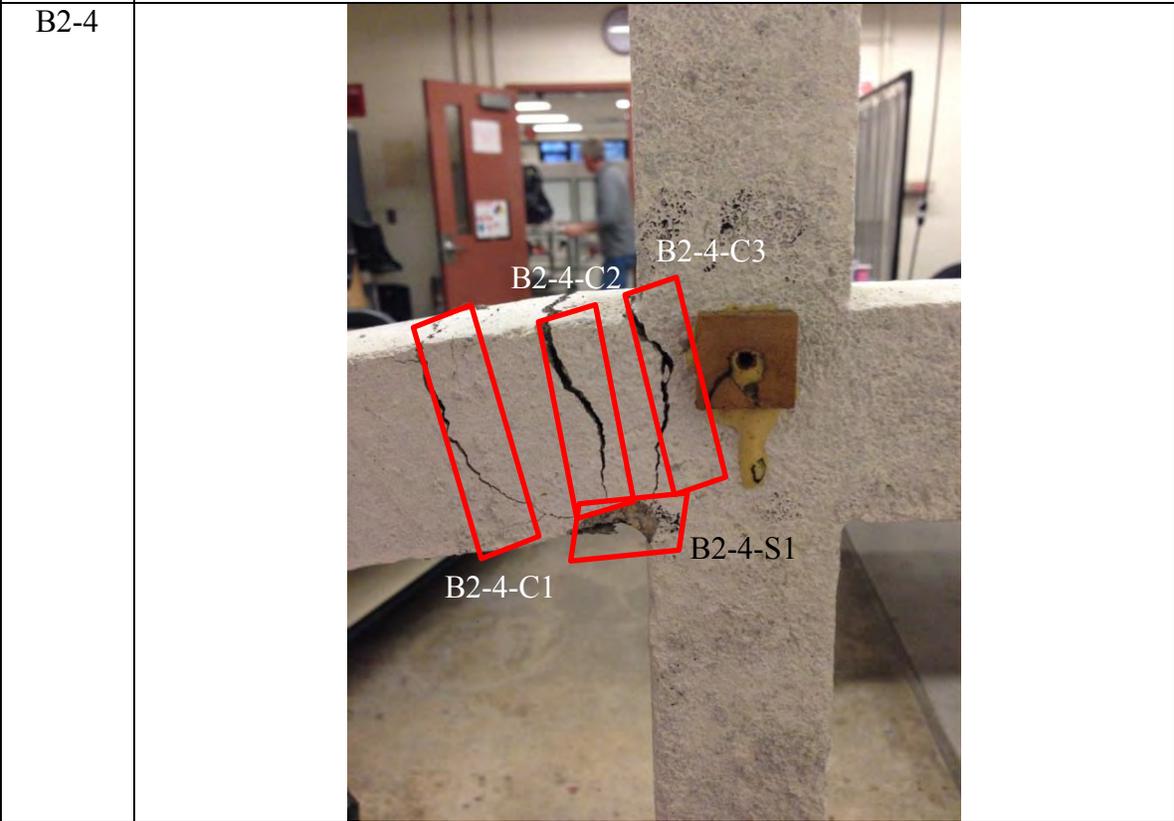
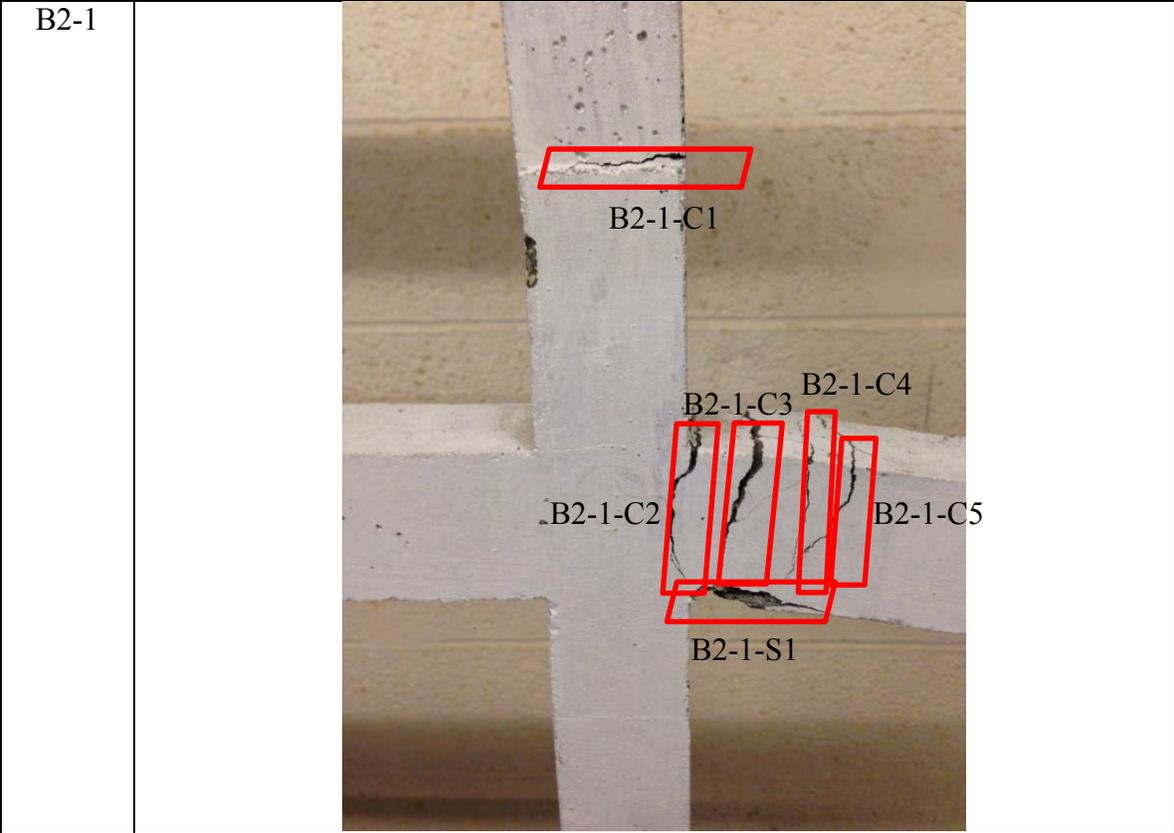
AB1-4

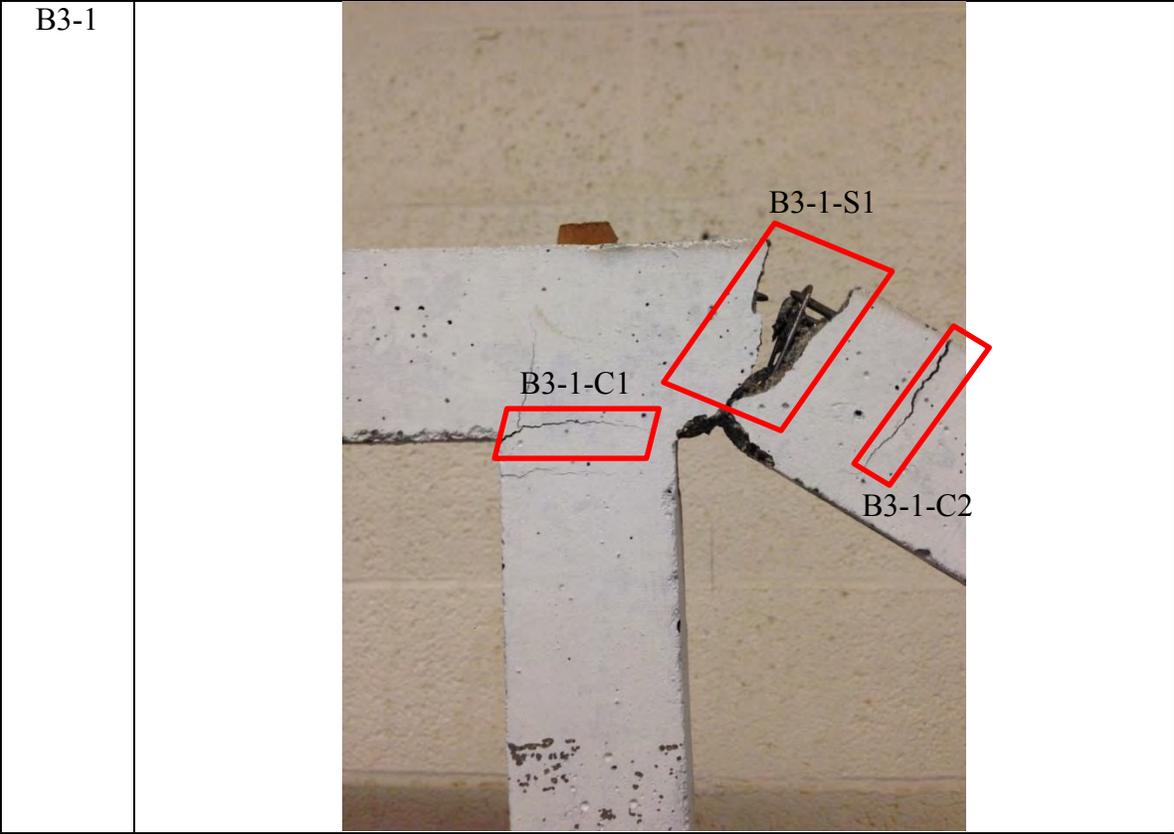
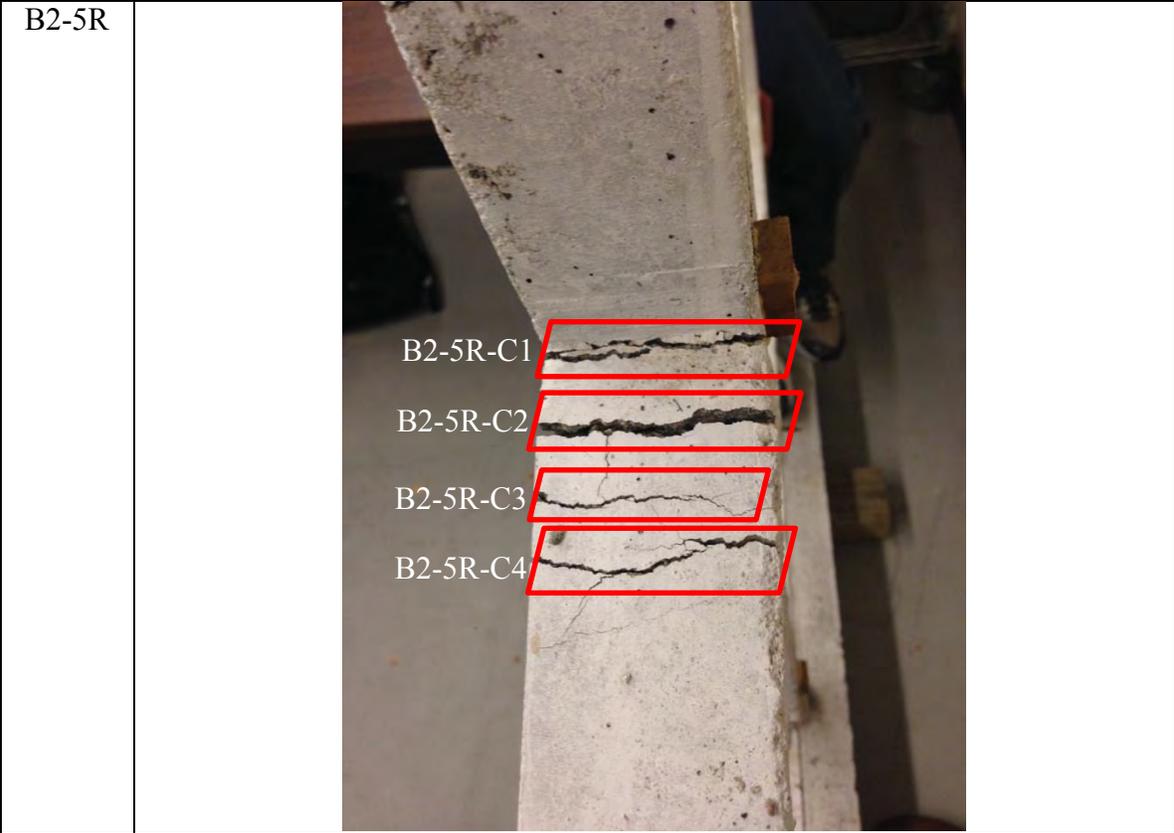




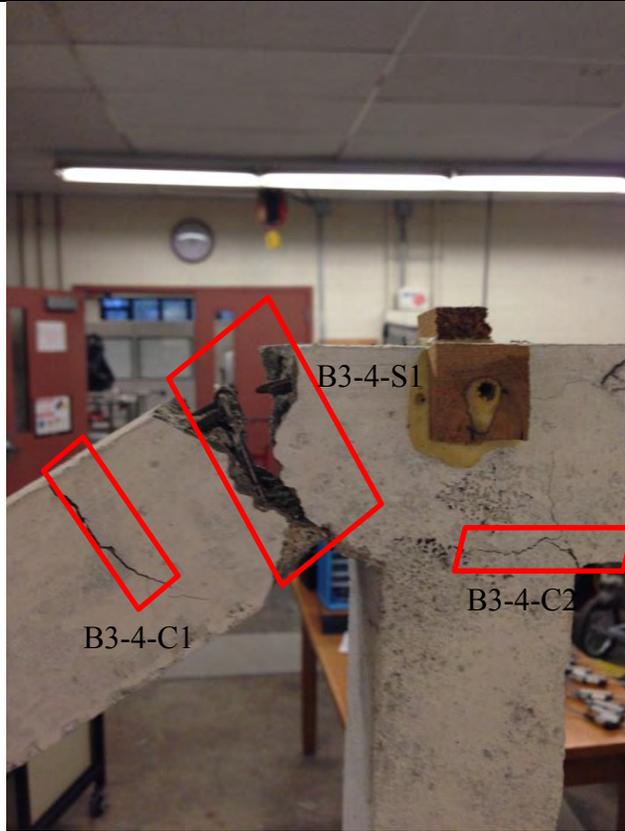




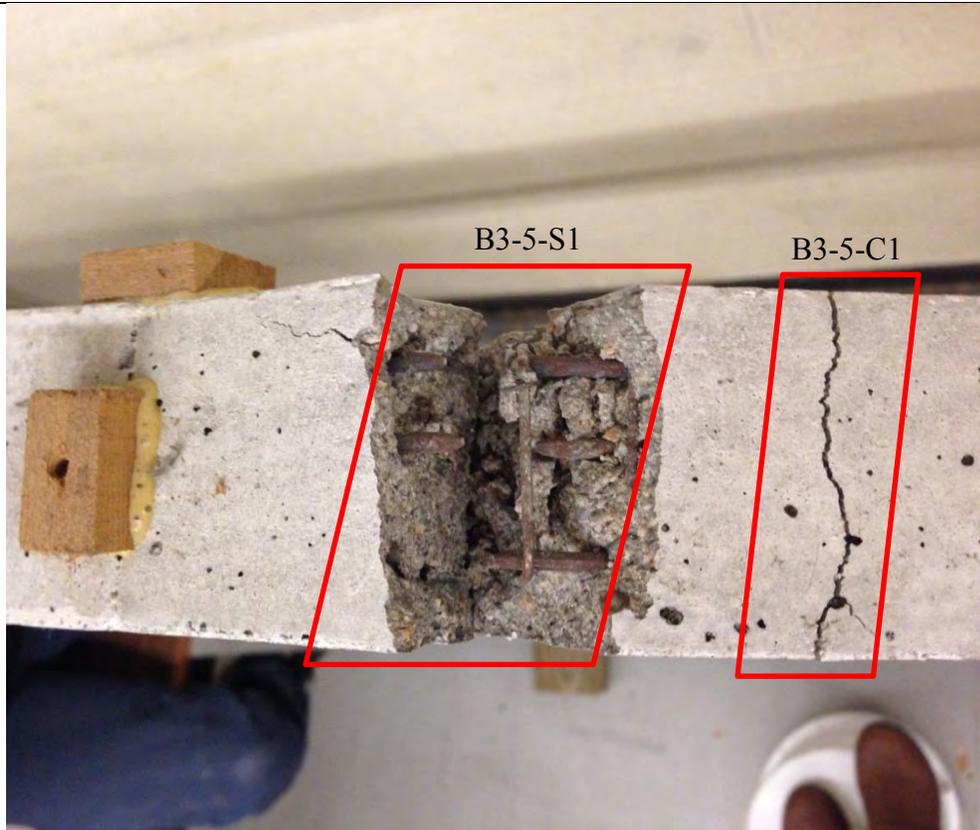




B3-4

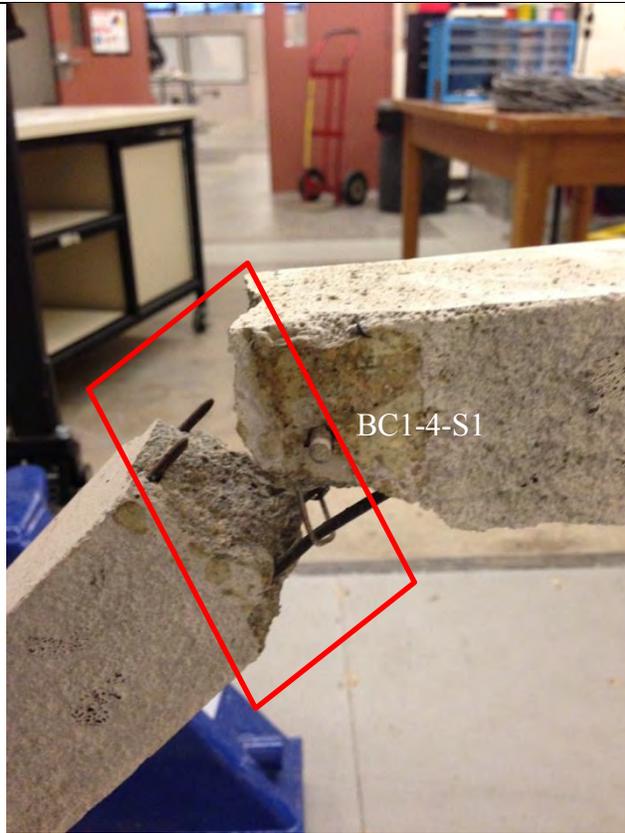


B3-5





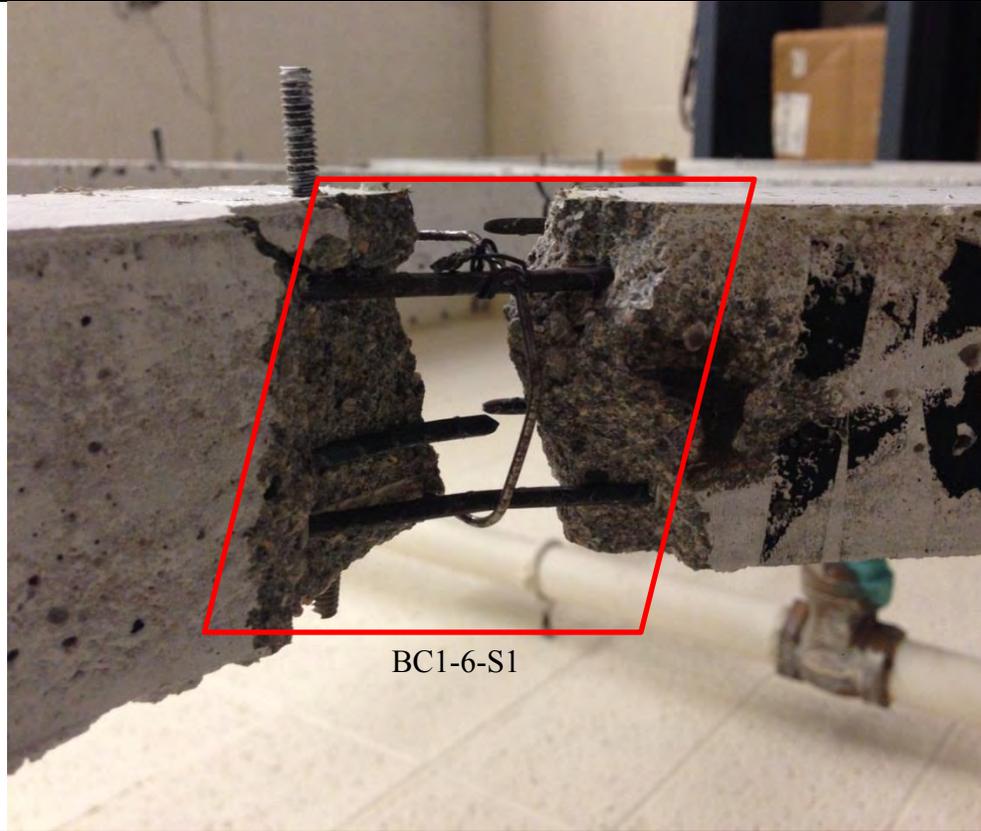
BC1-4



BC1-5



BC1-6



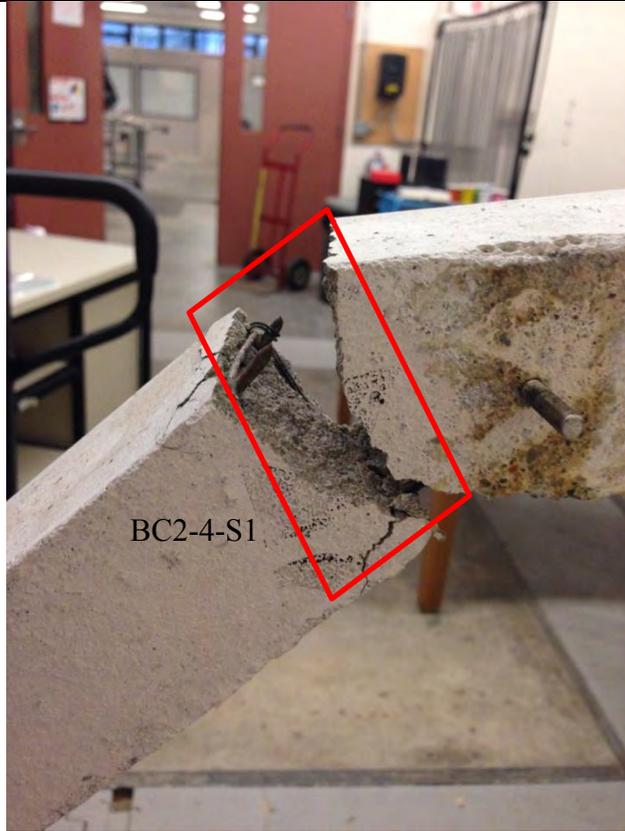
BC1-6-S1

BC2-1

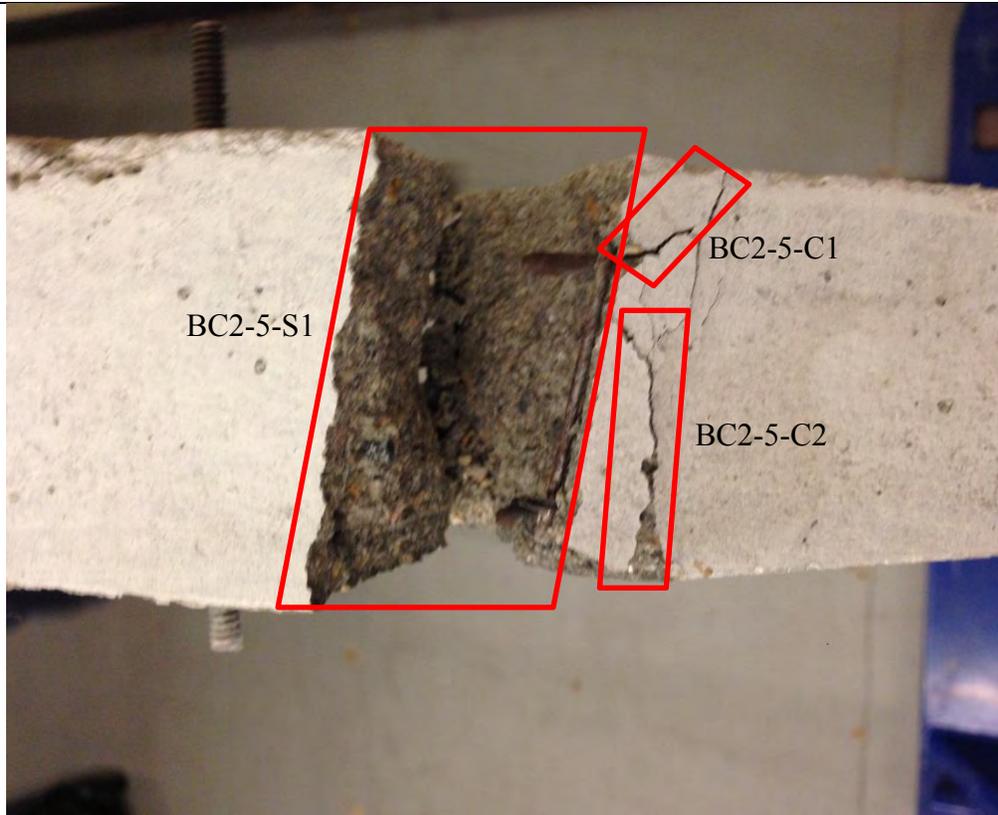


BC2-1-S1

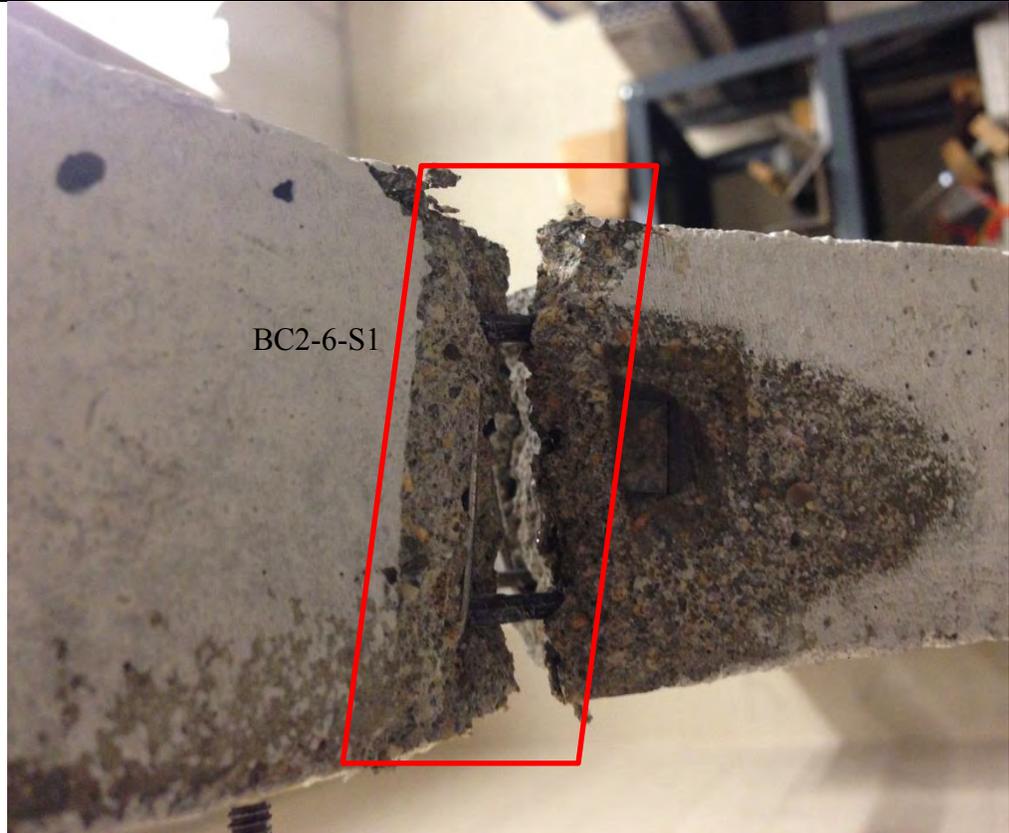
BC2-4



BC2-5

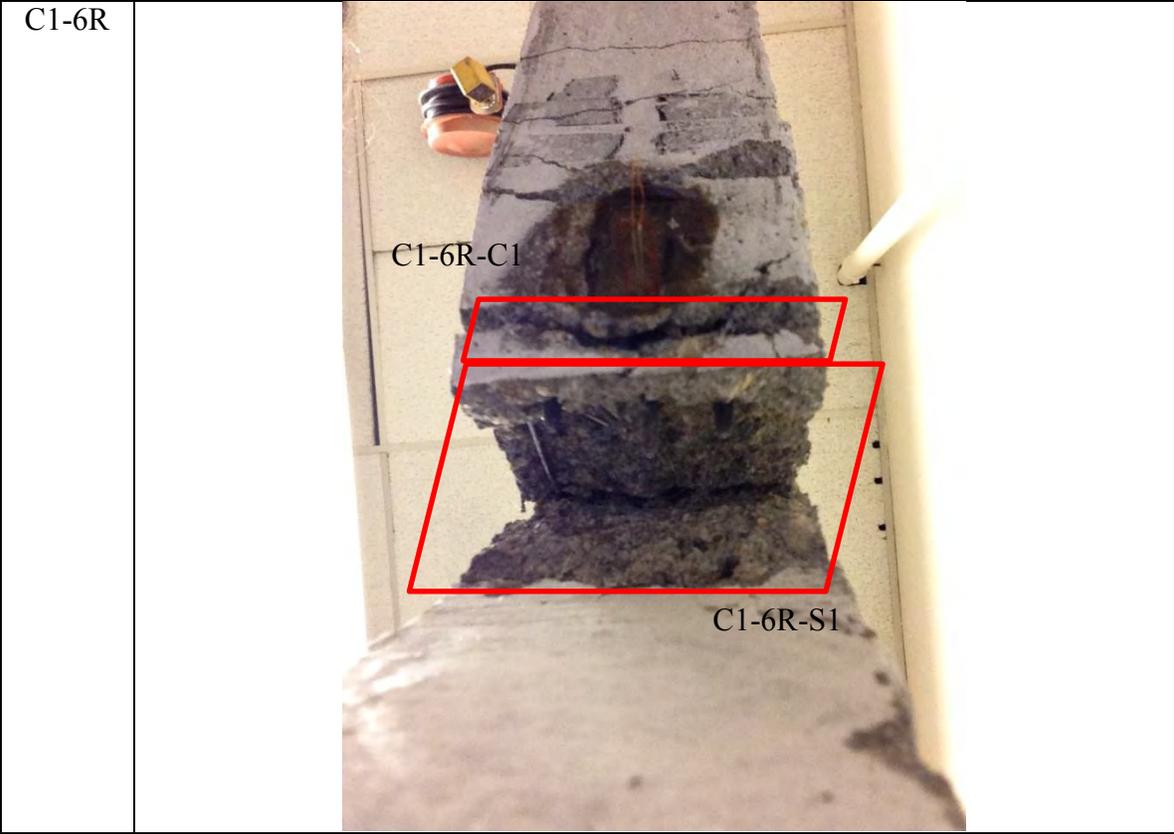
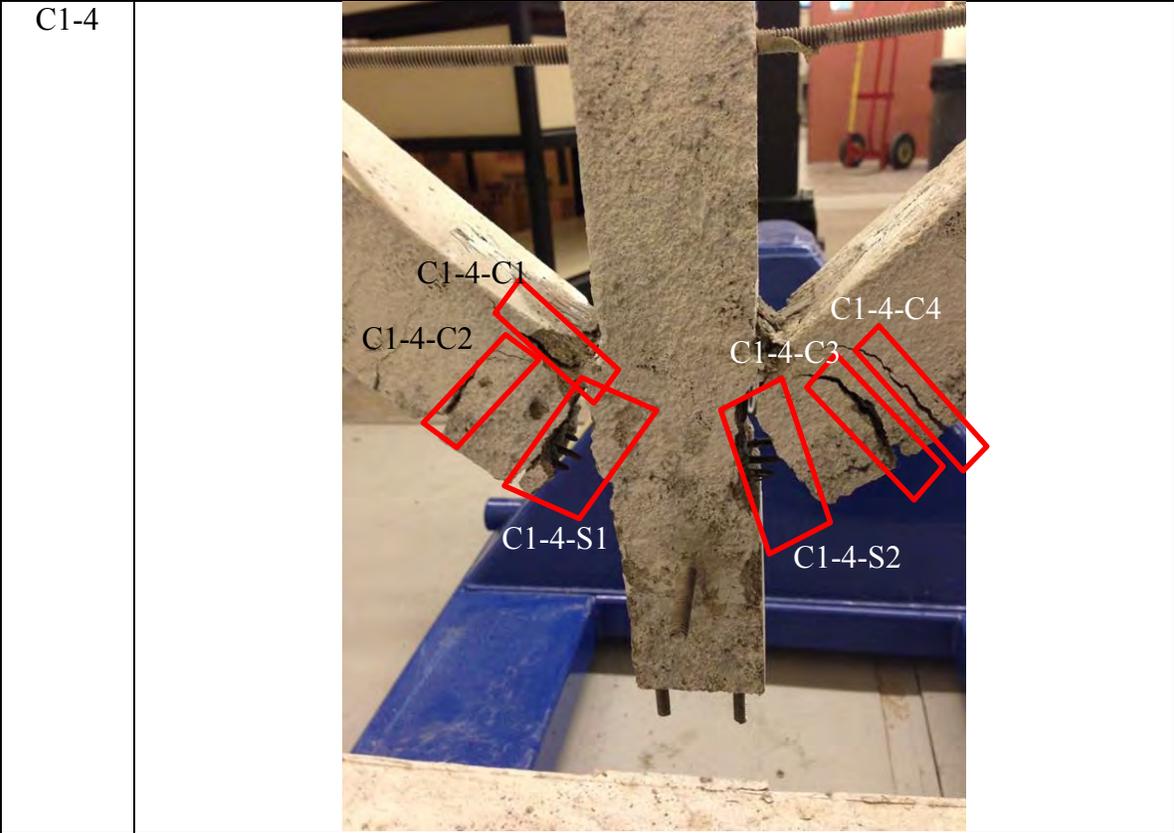


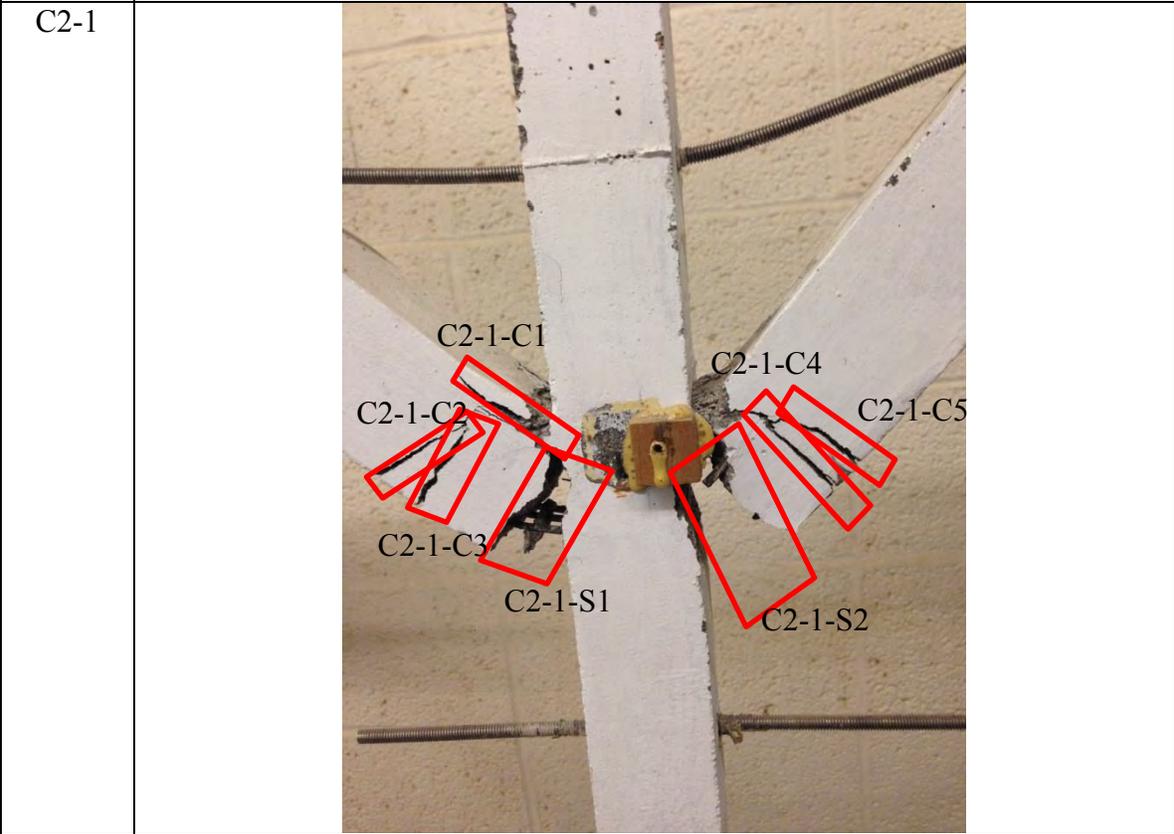
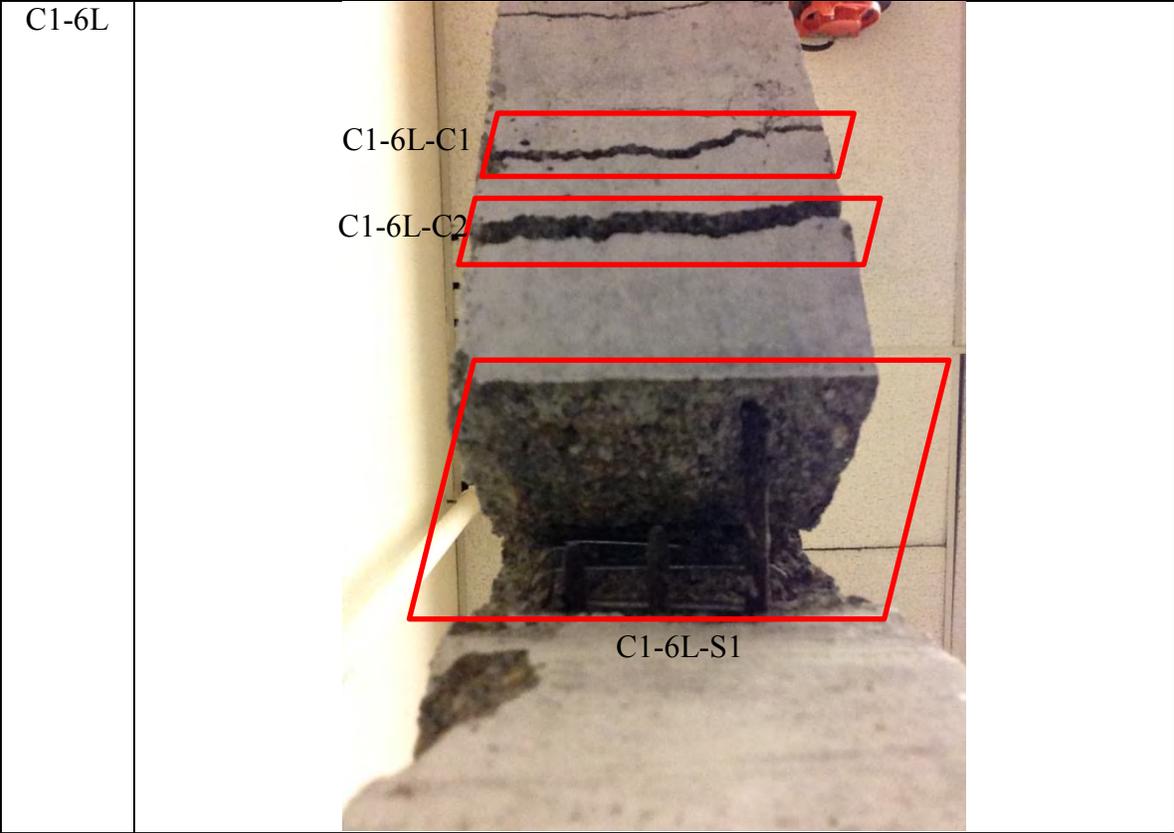
BC2-6

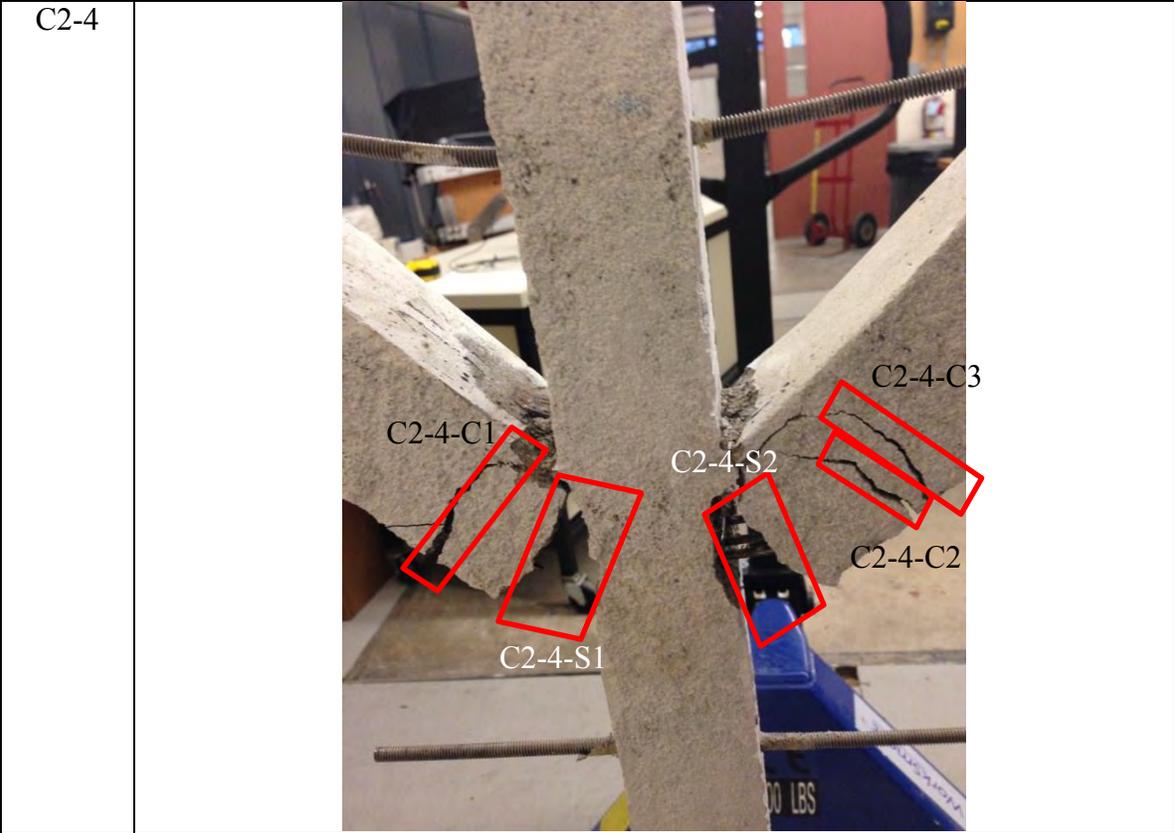


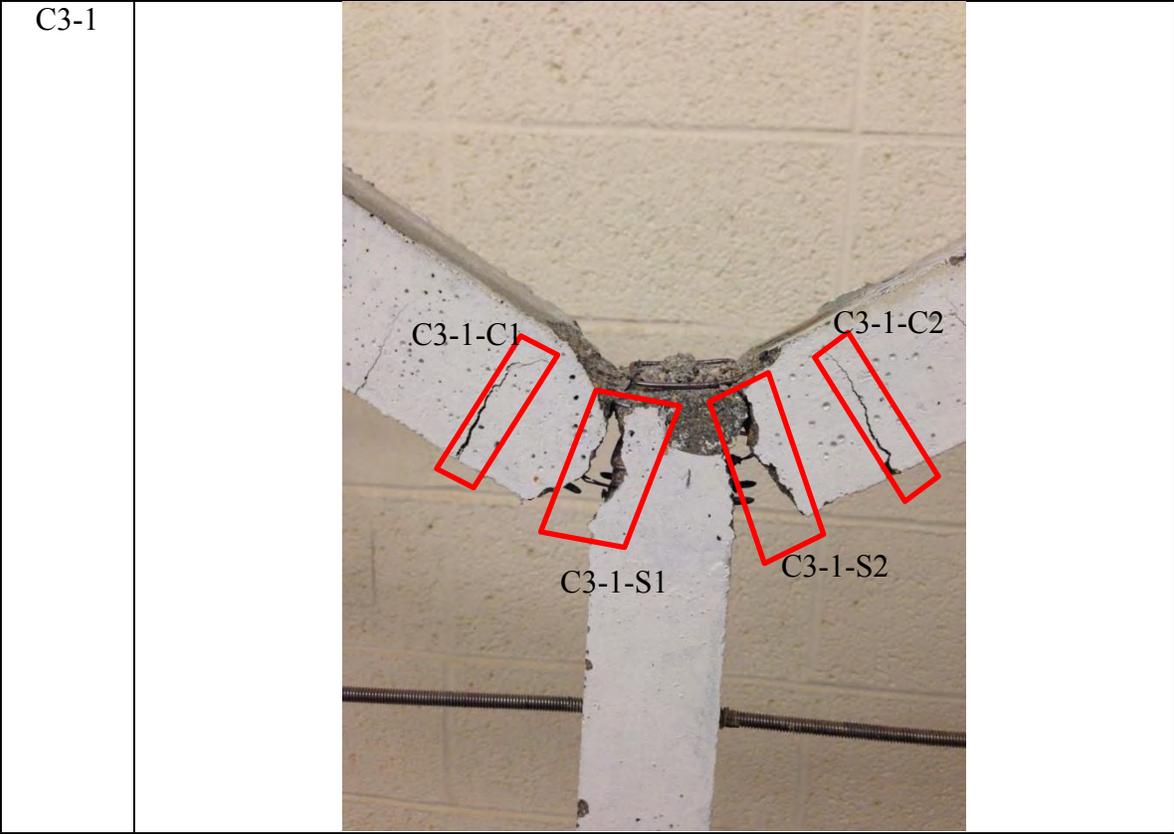
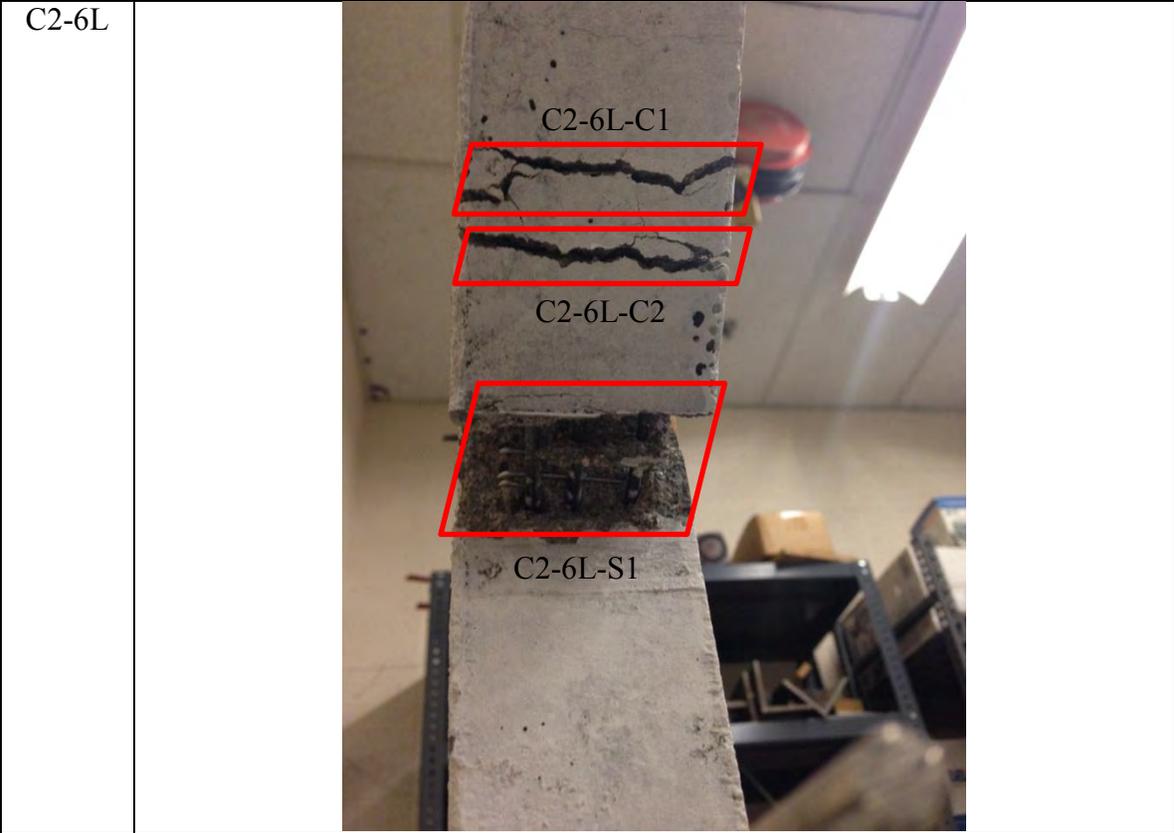
C1-1



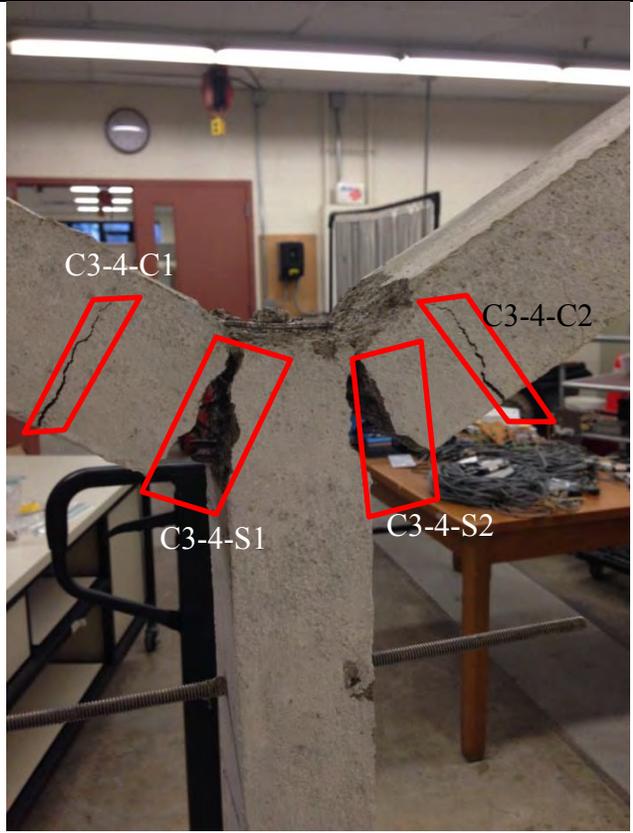






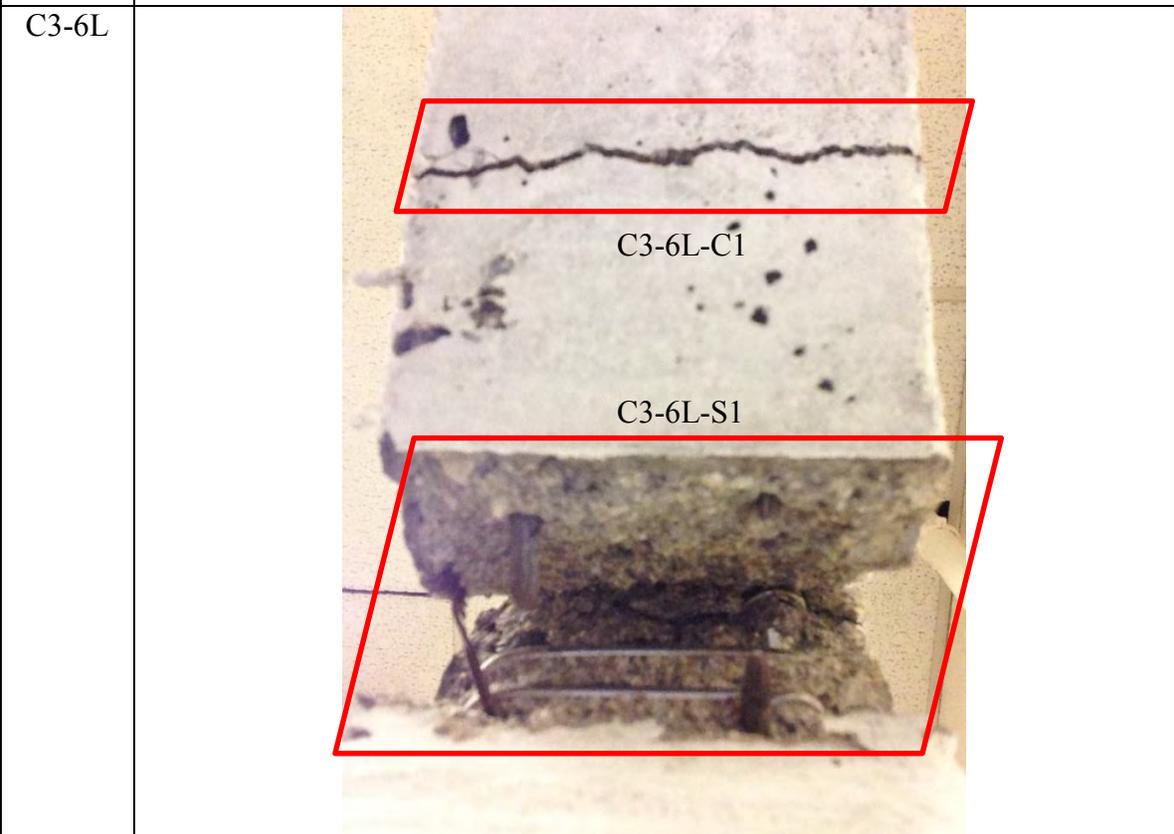
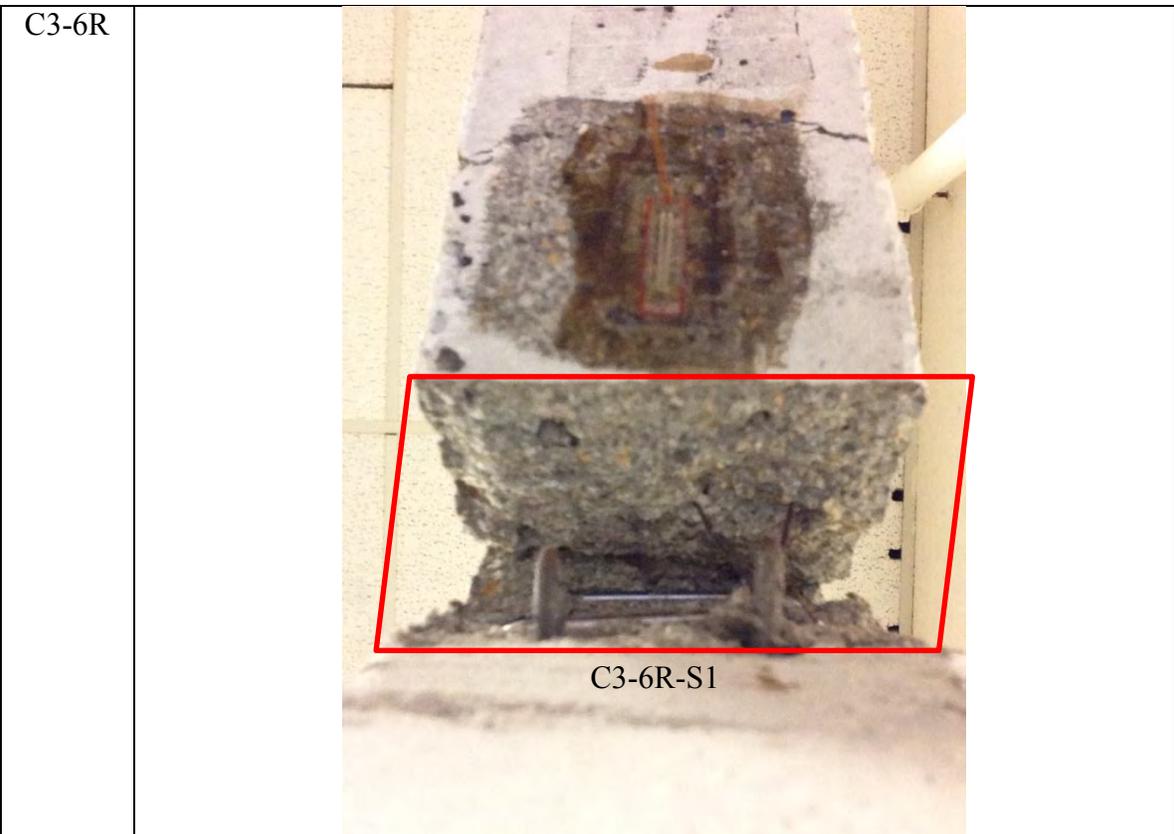


C3-4



C3-5

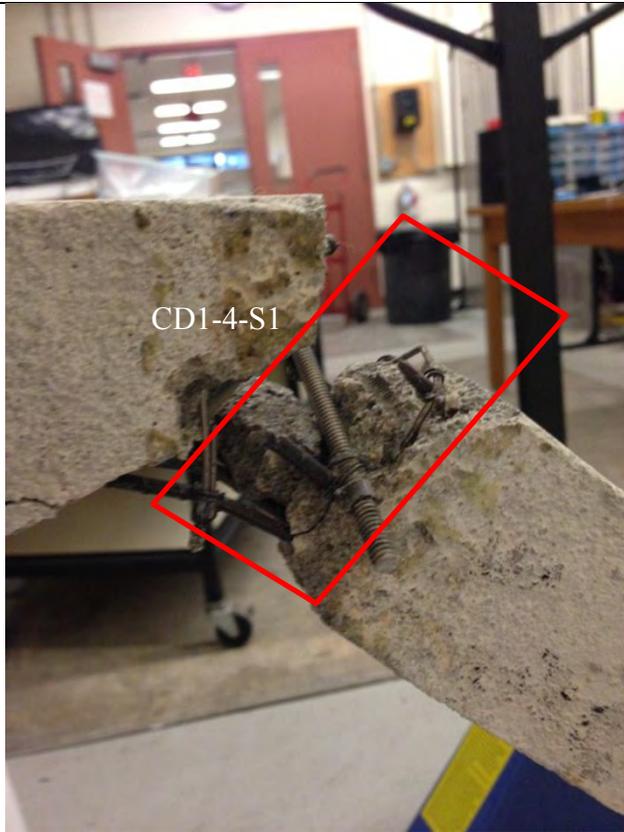




CD1-1

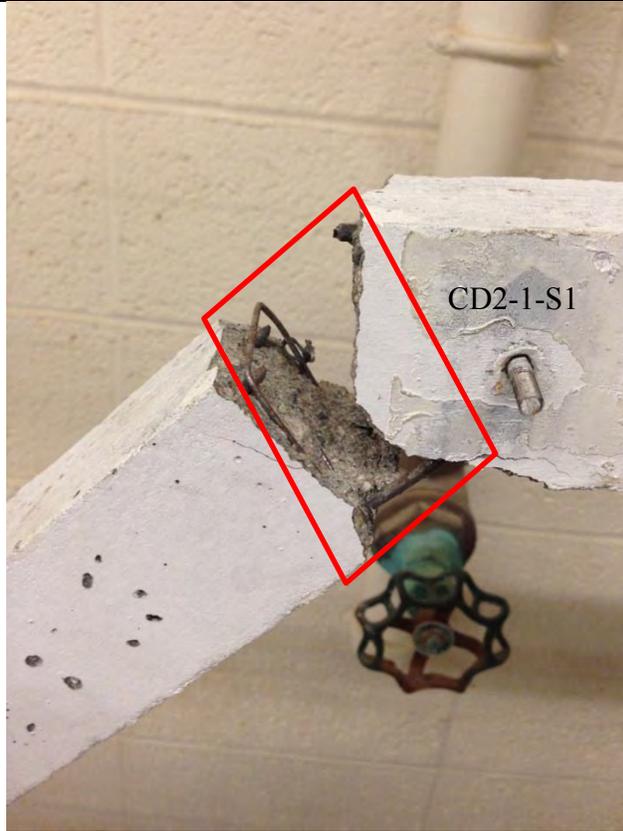


CD1-4

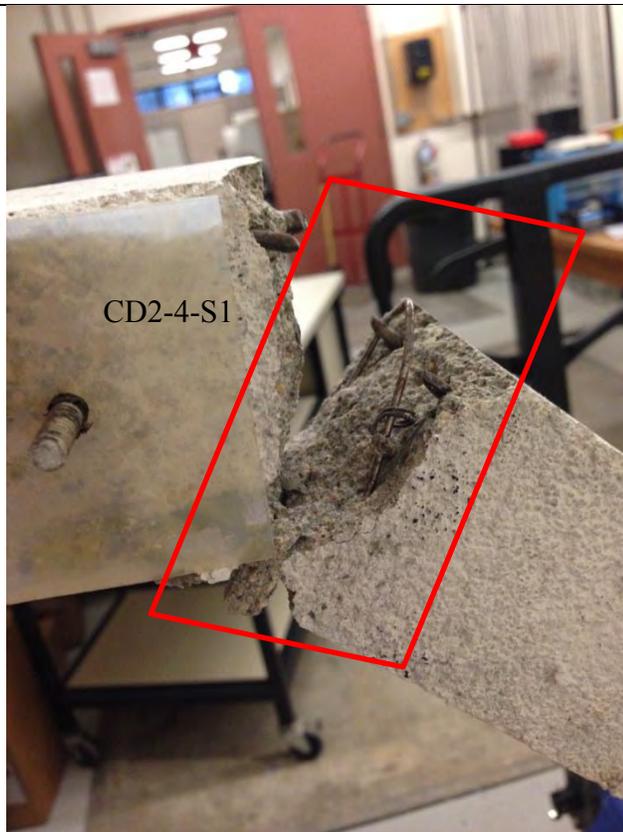




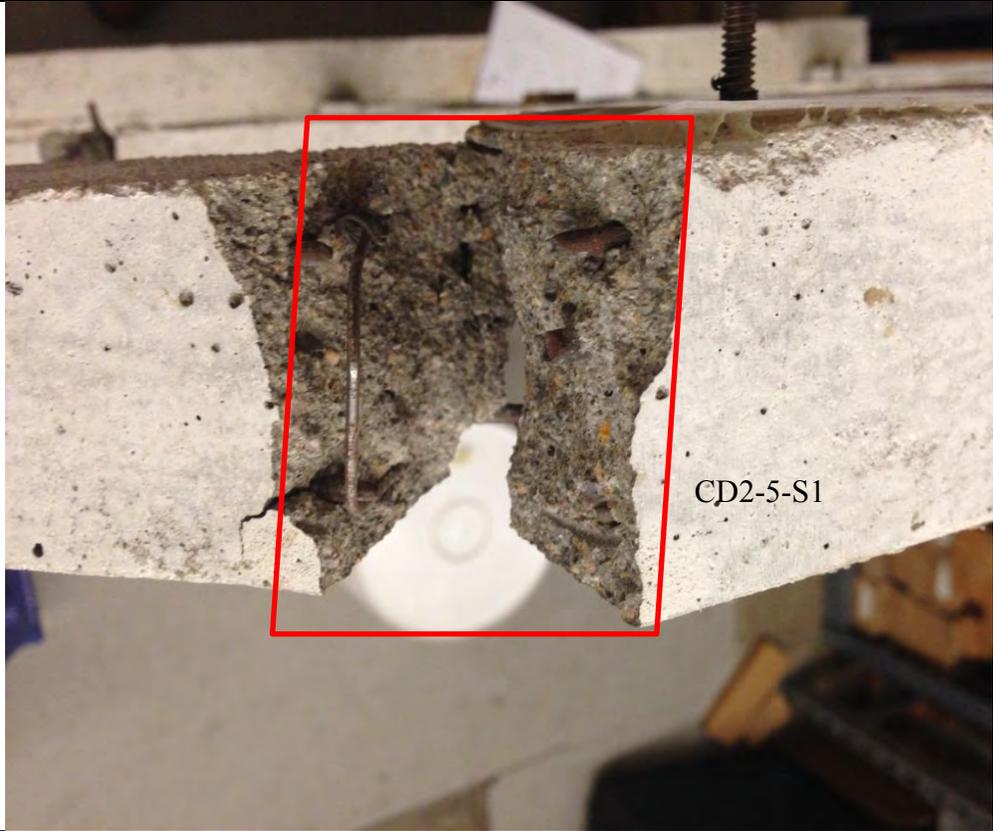
CD2-1



CD2-4

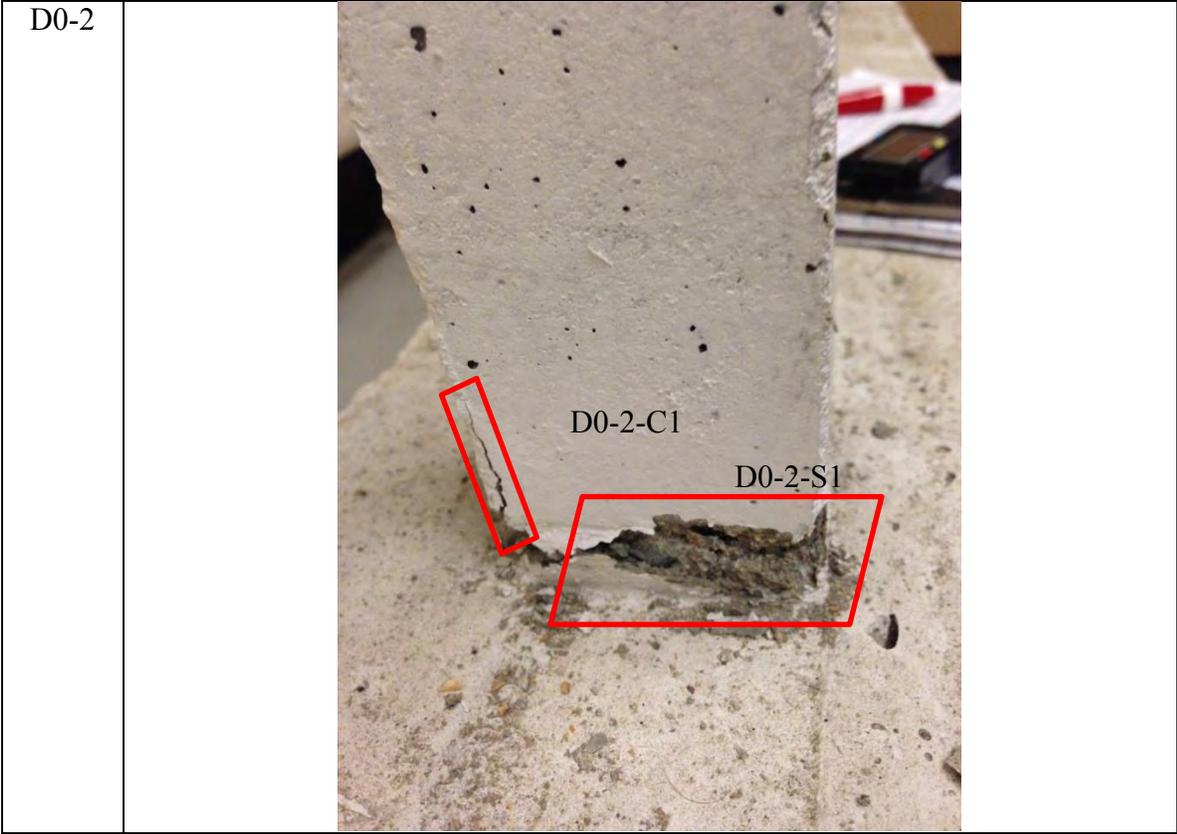
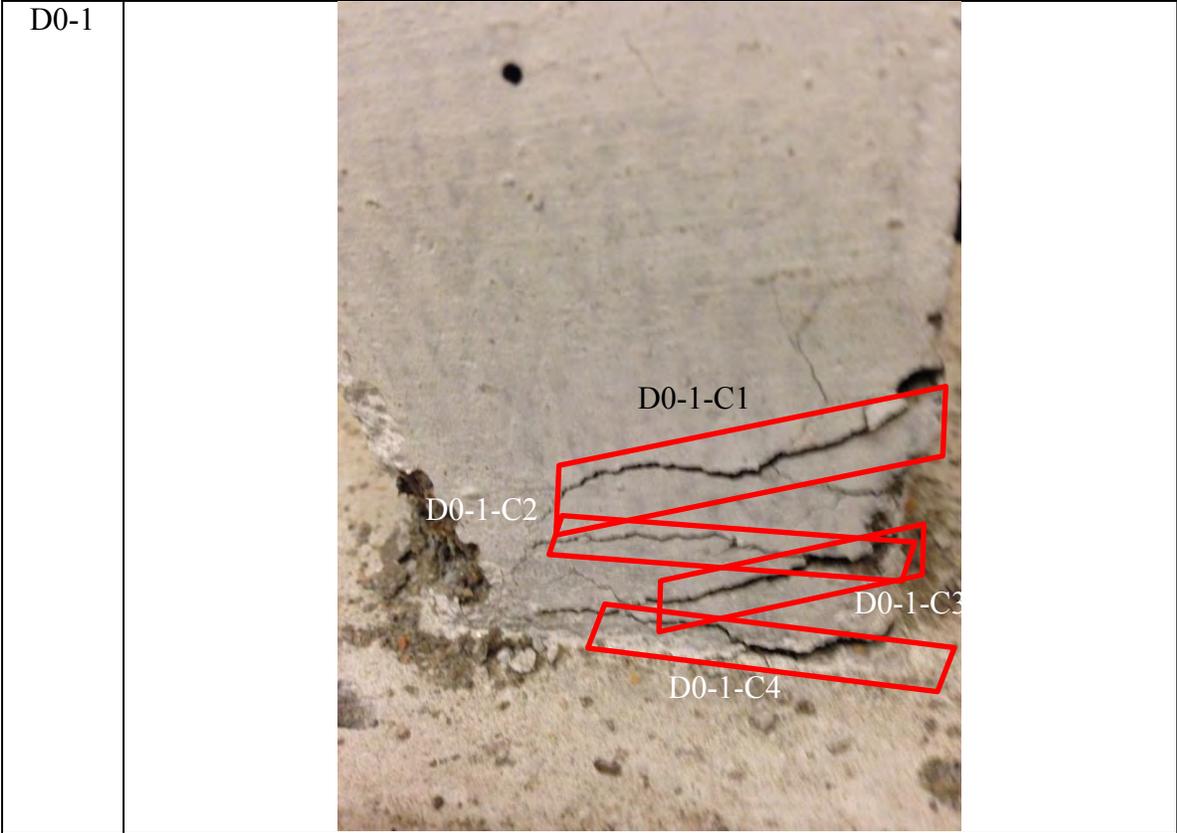


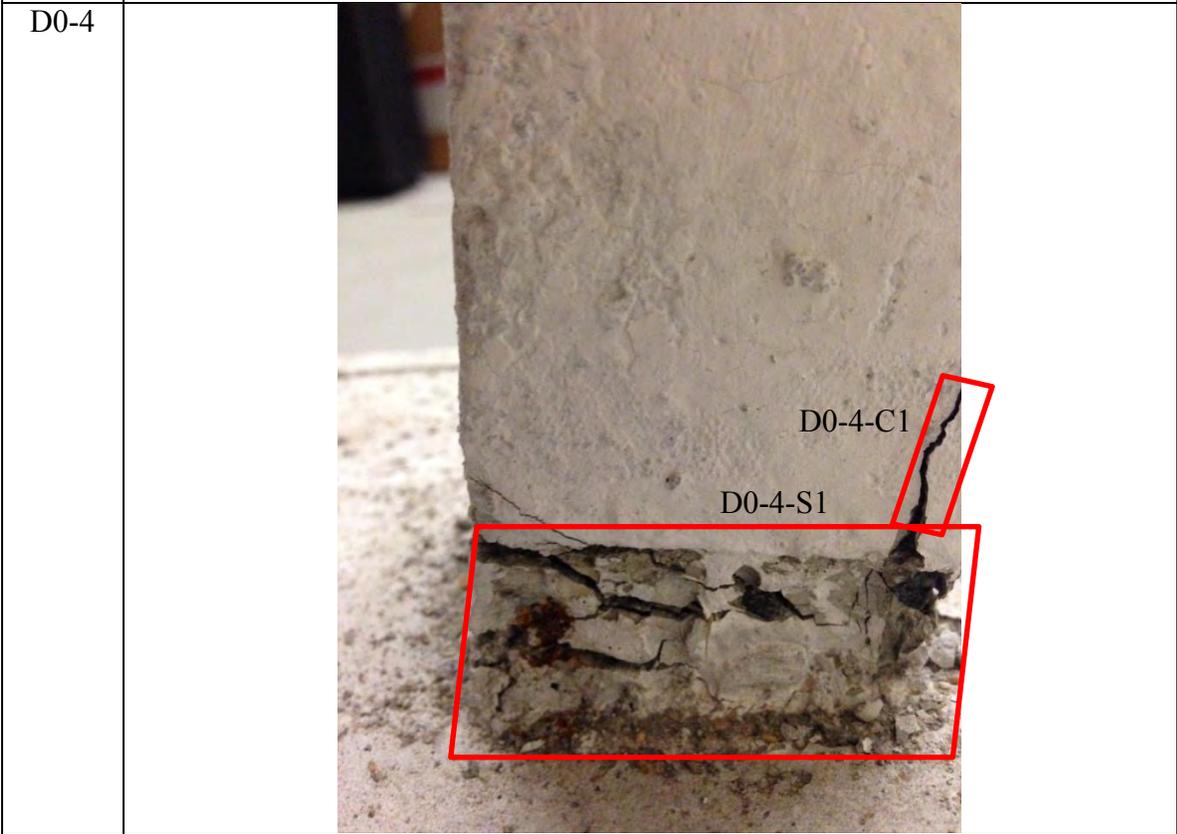
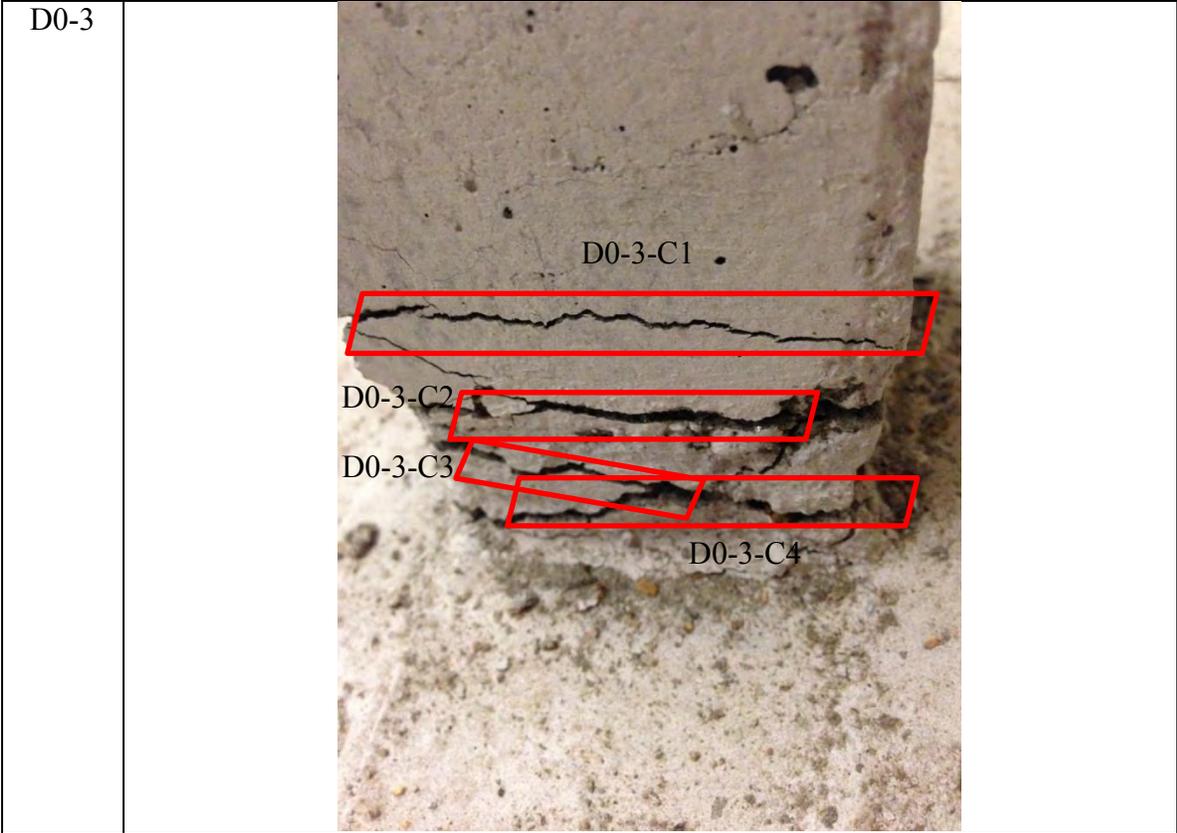
CD2-5

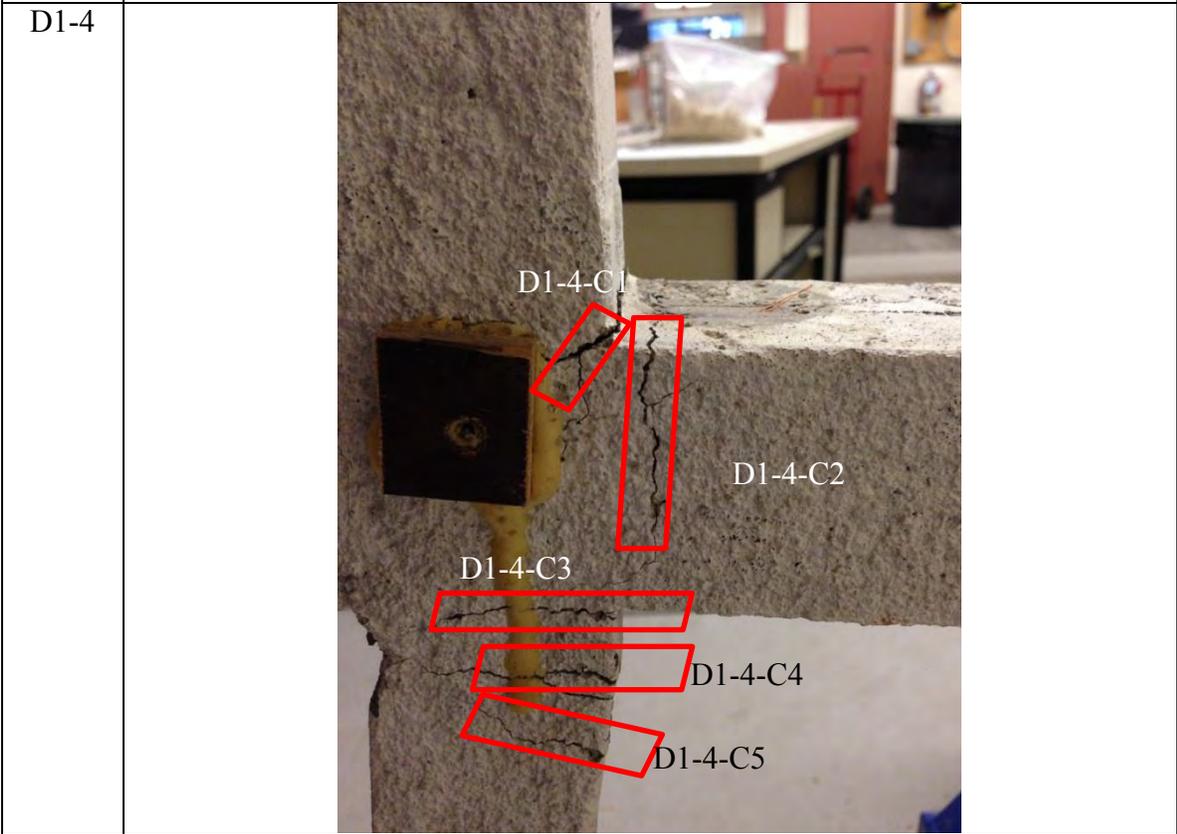
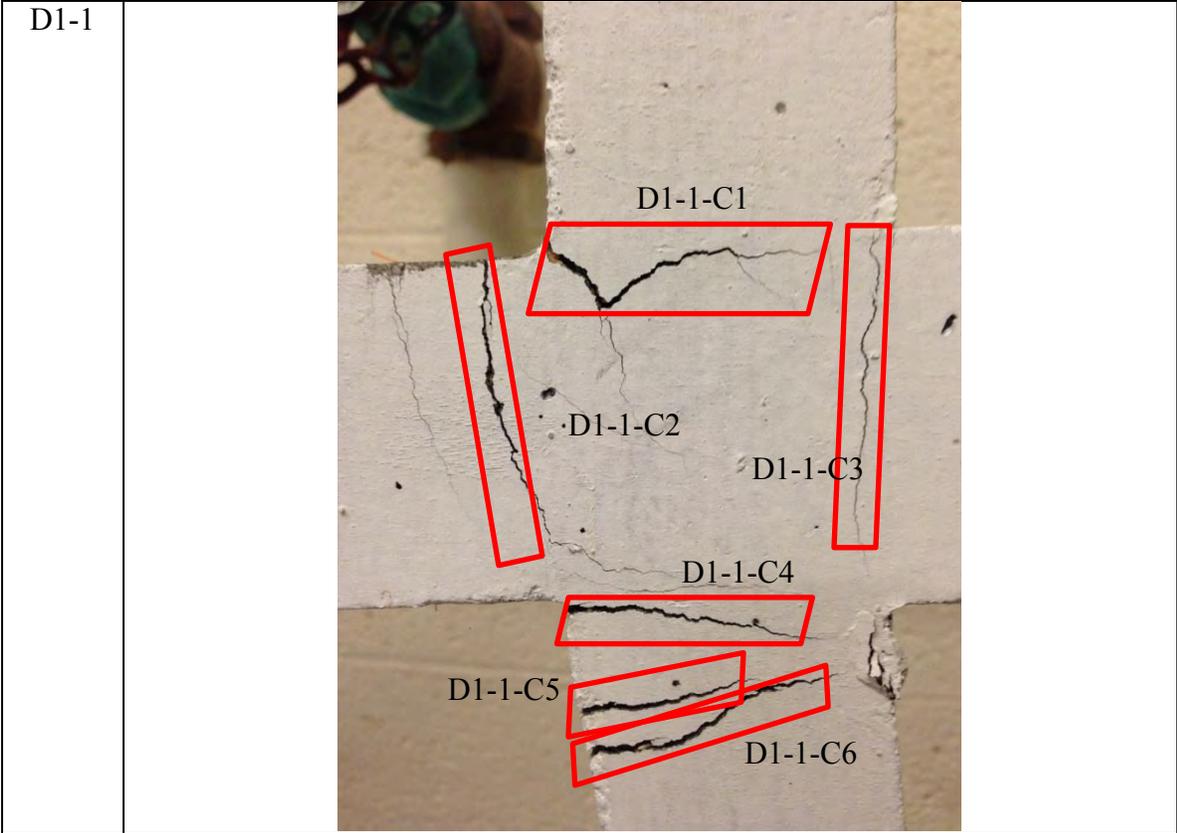


CD2-6

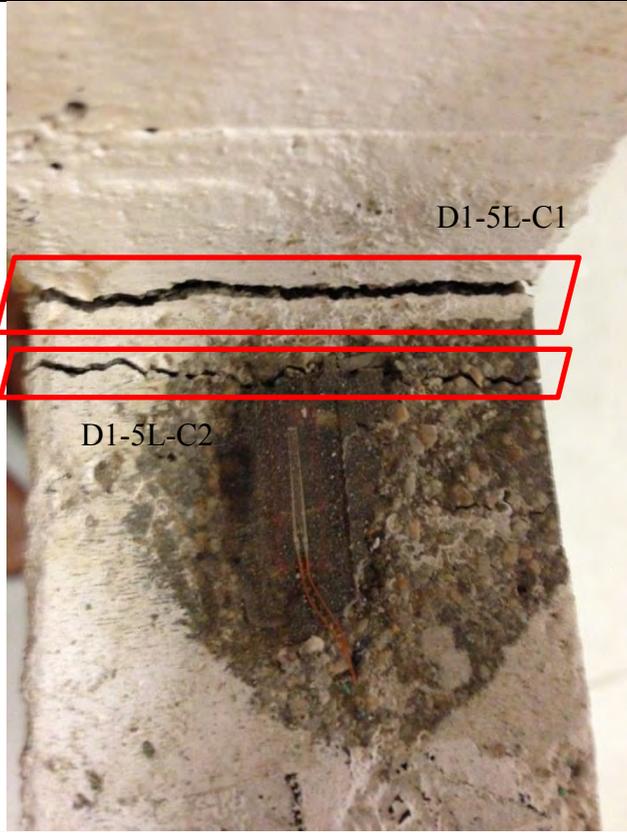




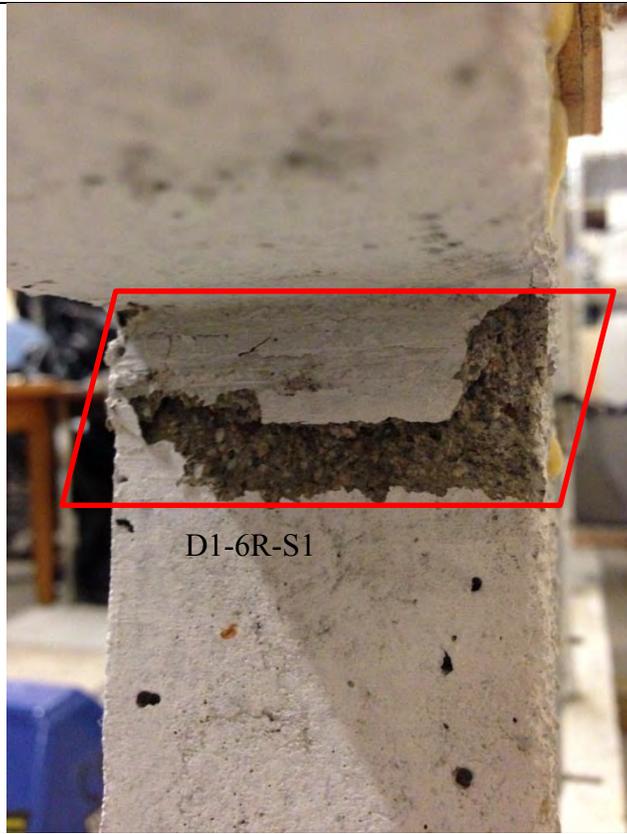




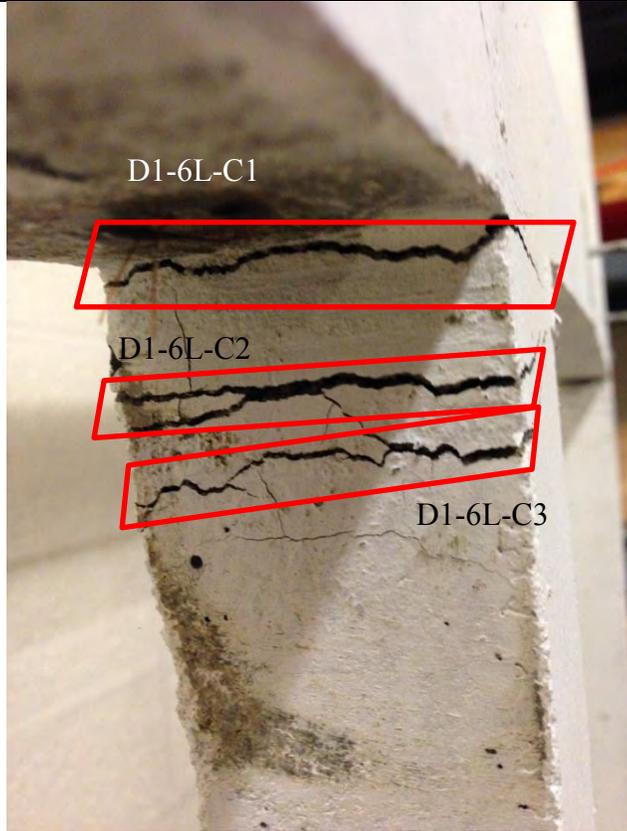
D1-5L



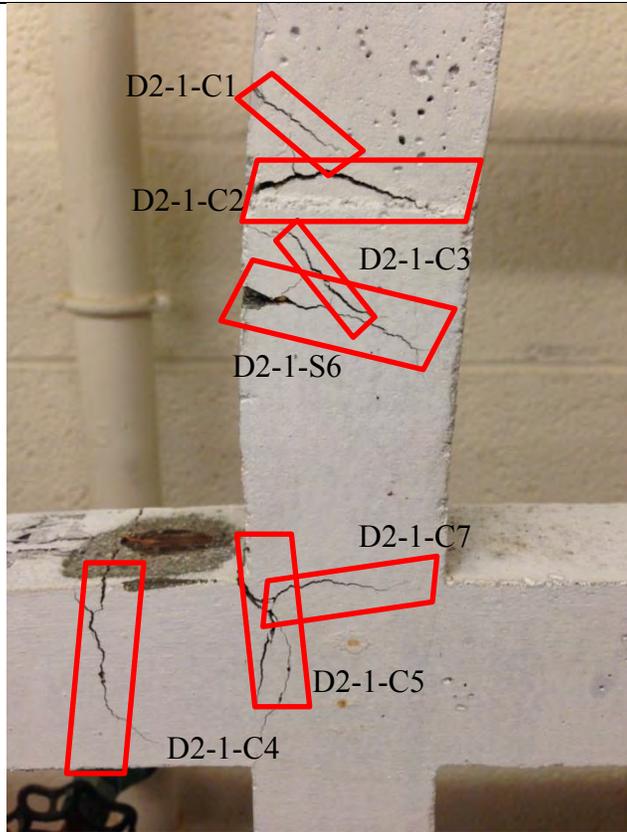
D1-6R



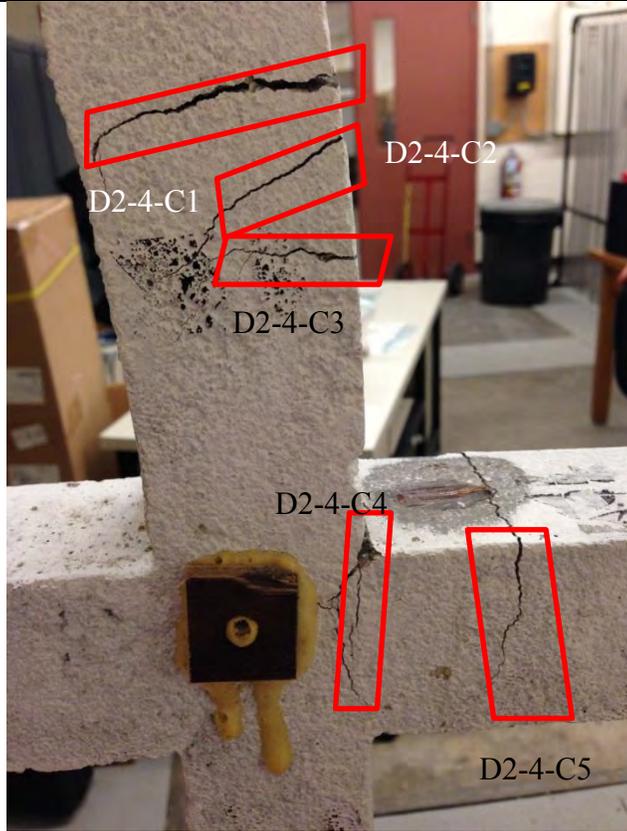
D1-6L



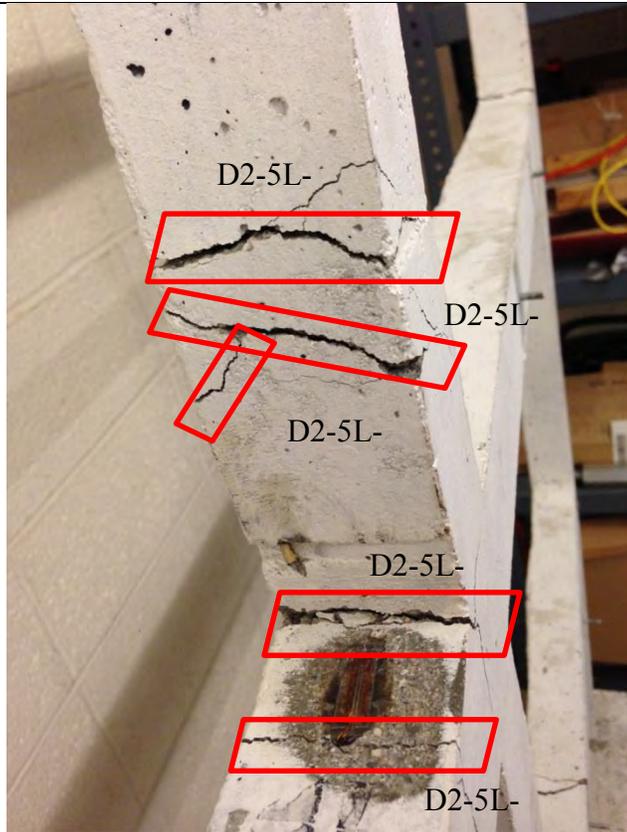
D2-1

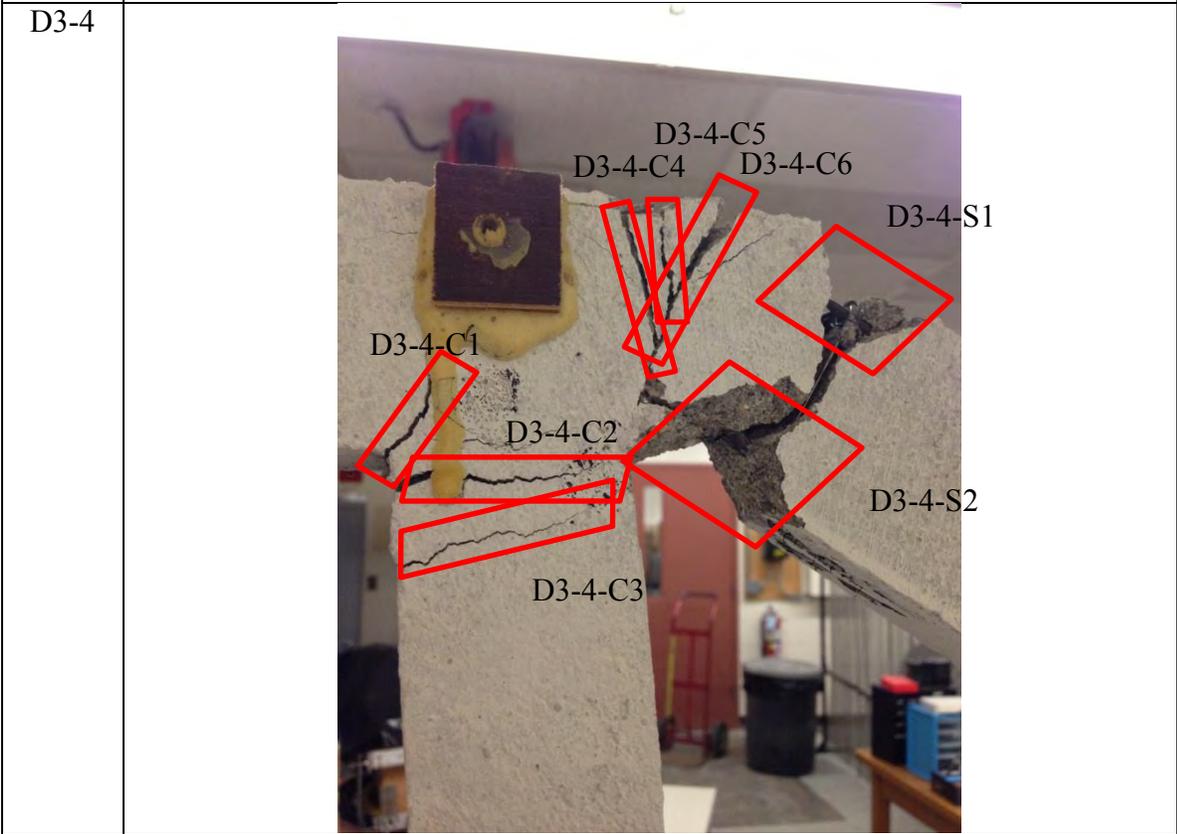
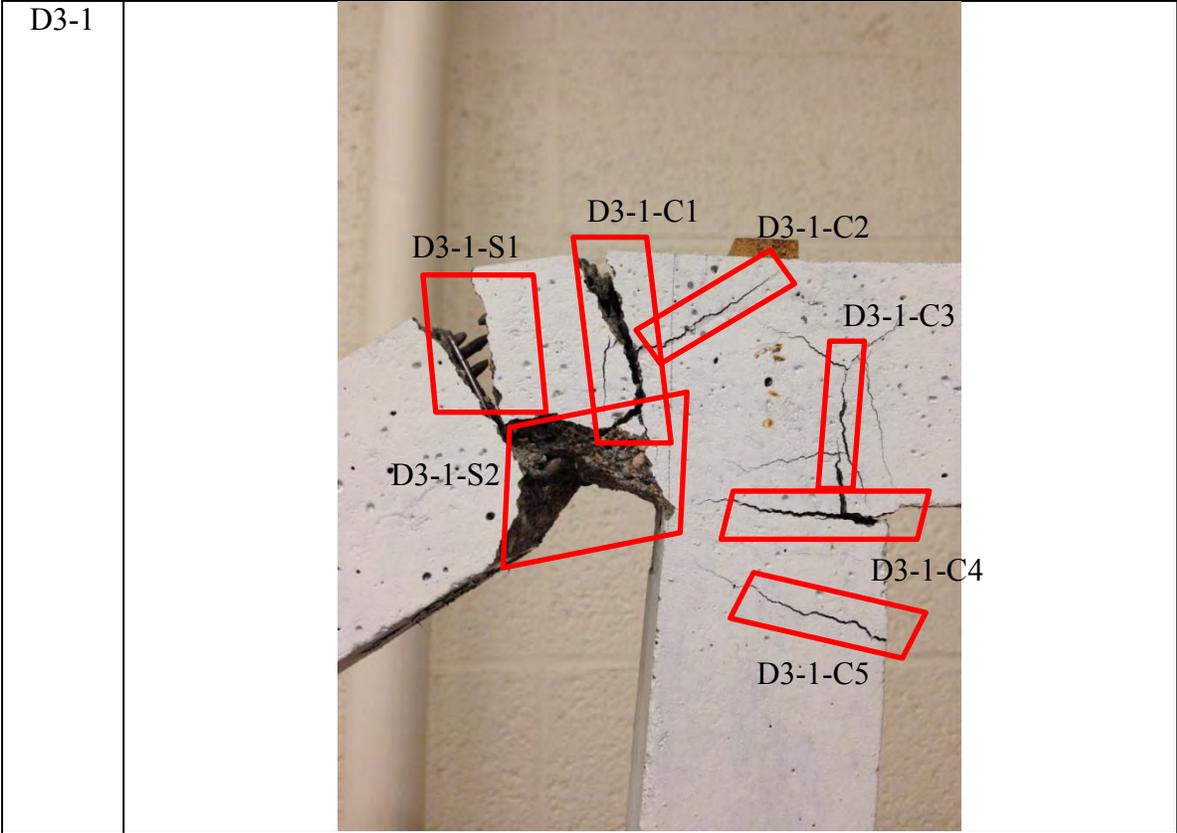


D2-4

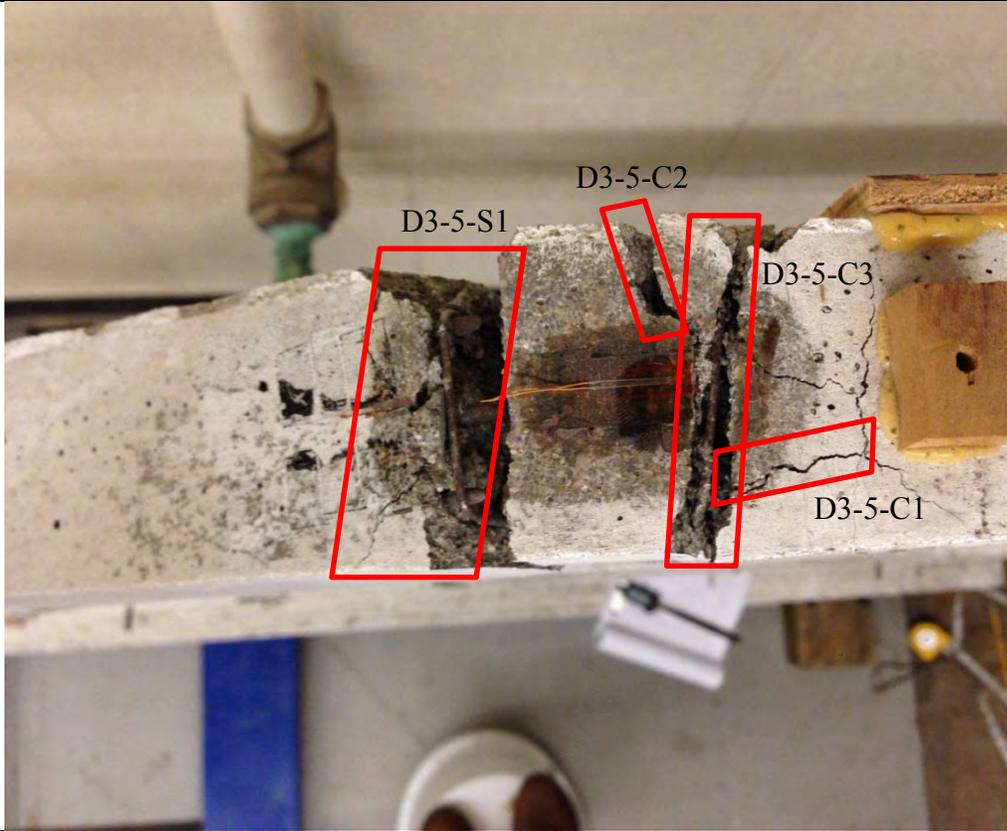


D2-5L

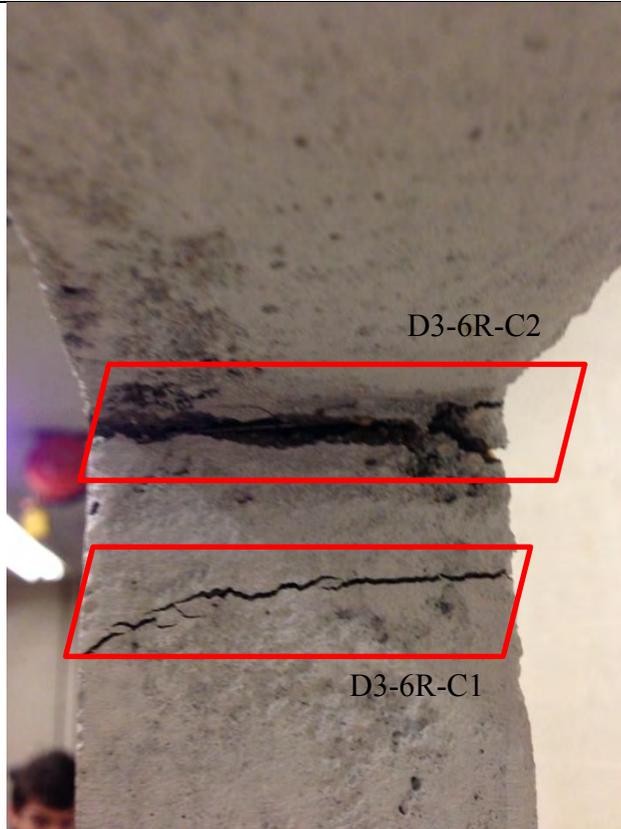




D3-5



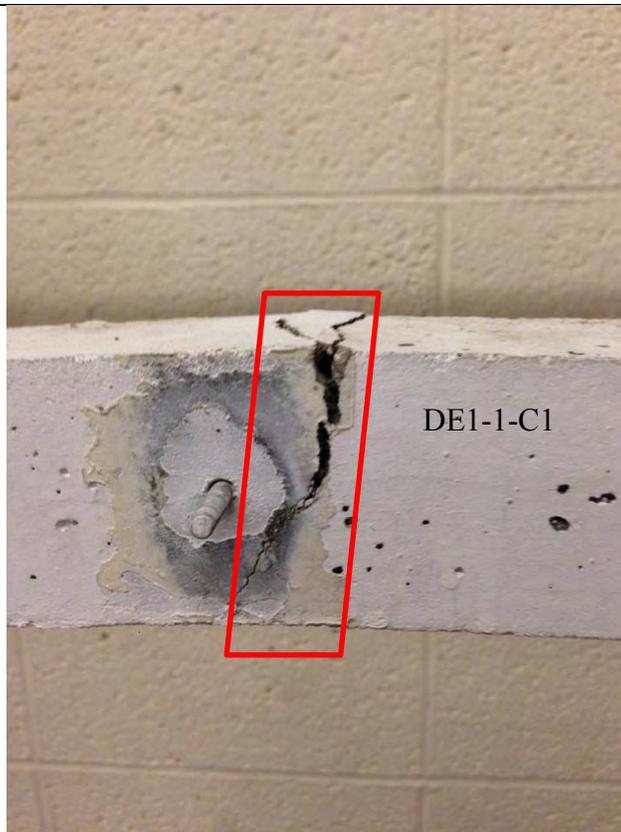
D3-6R



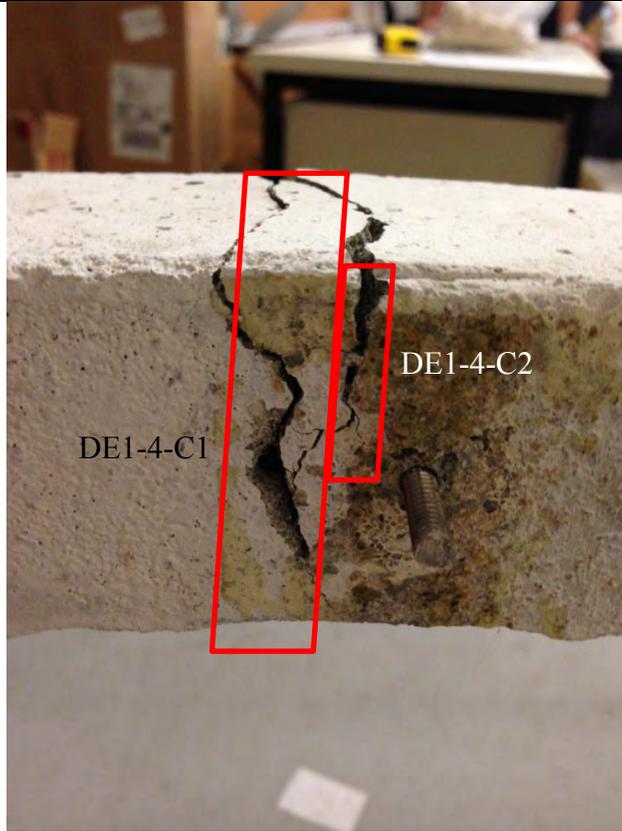
D3-6L



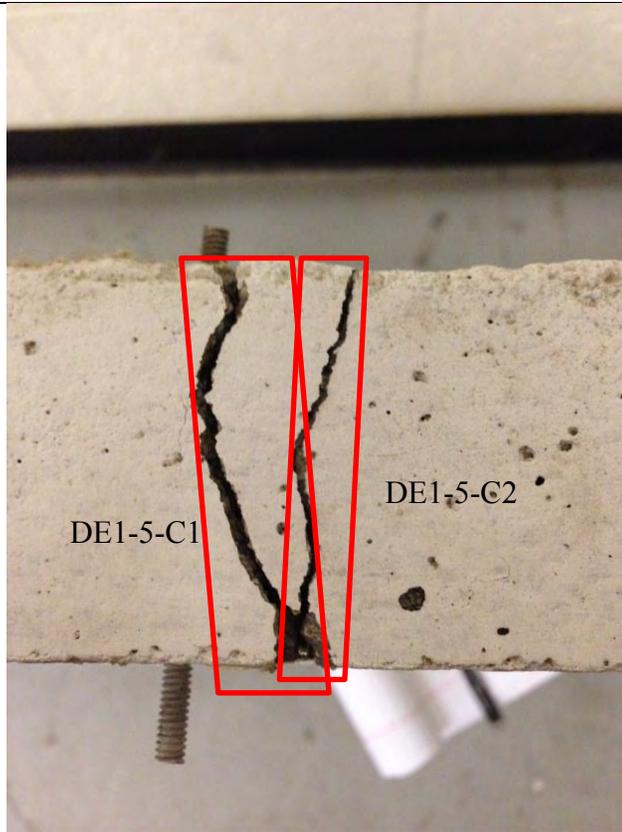
DE1-1

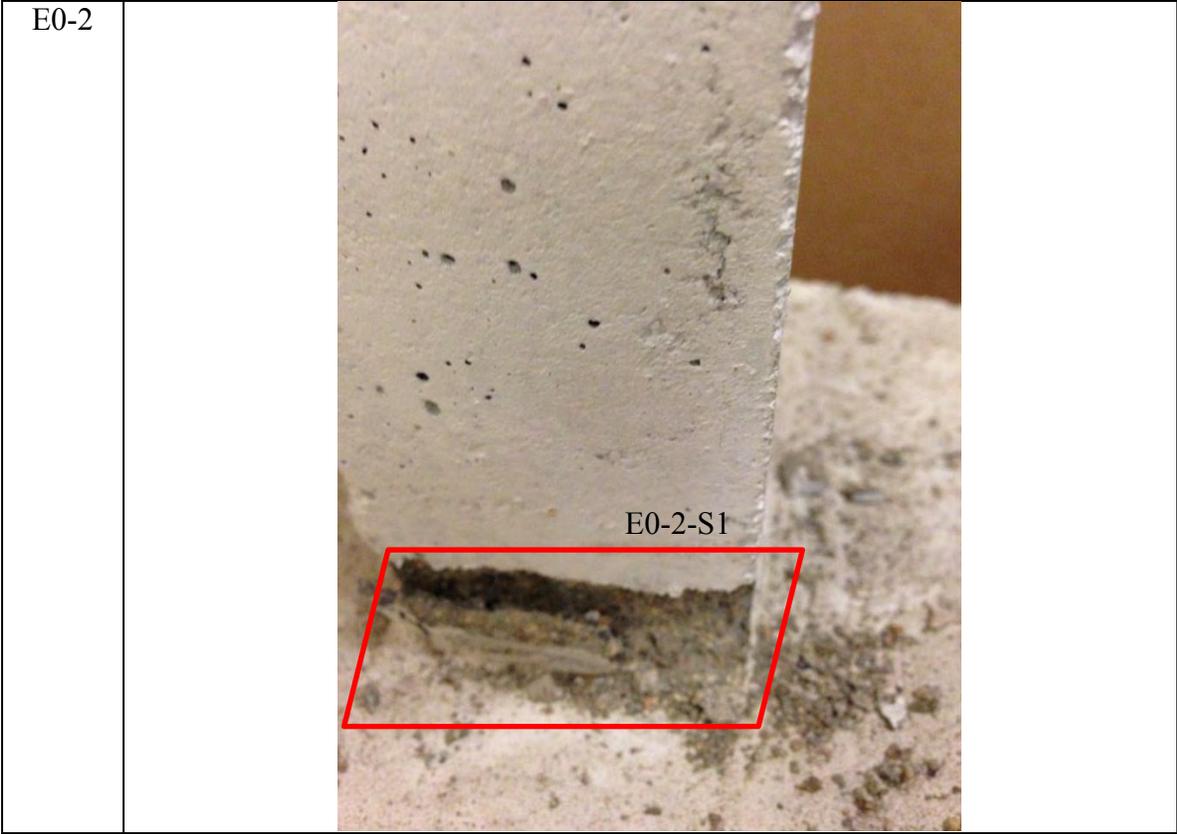
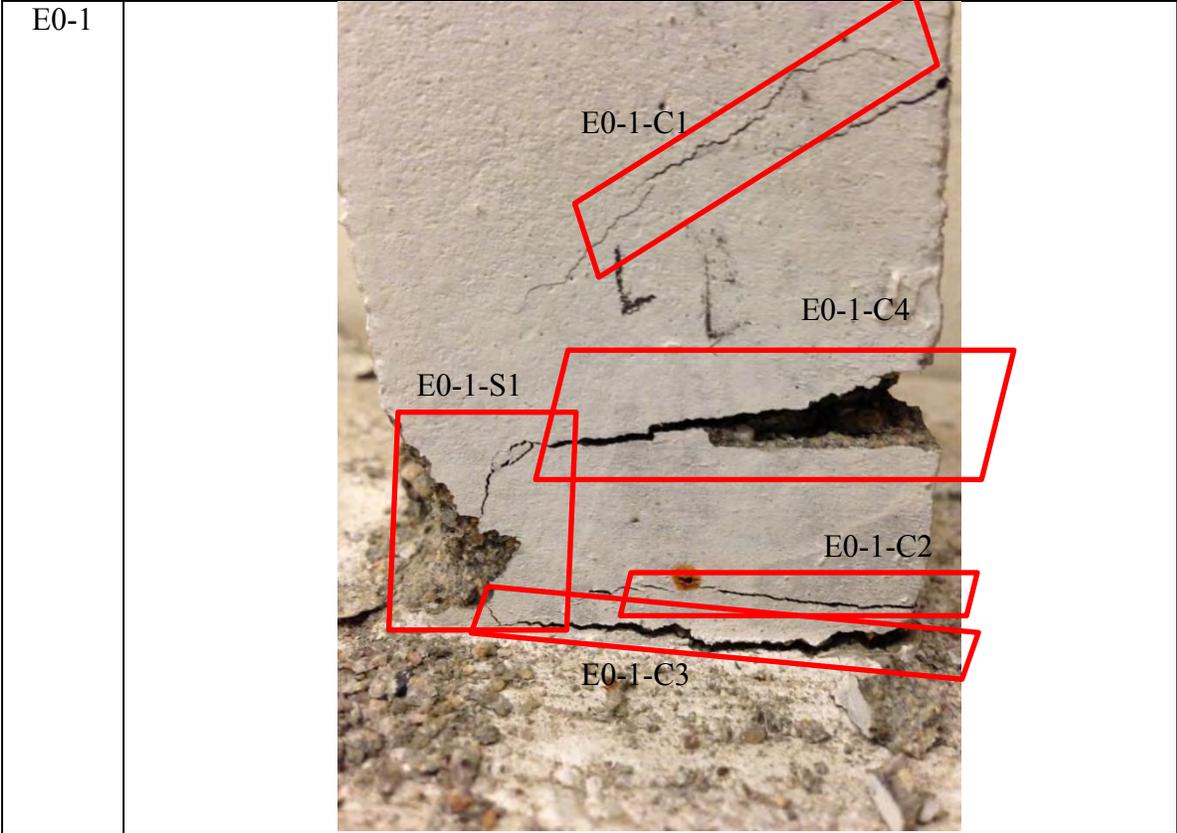


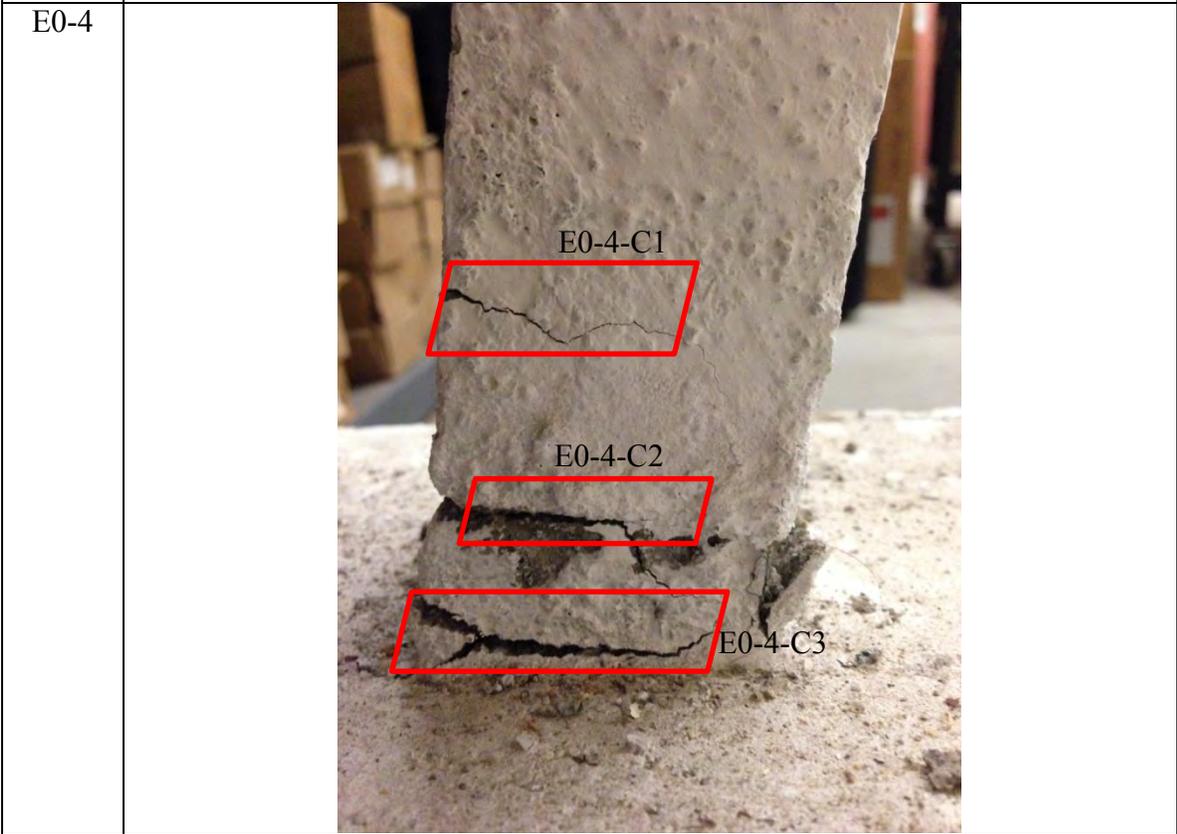
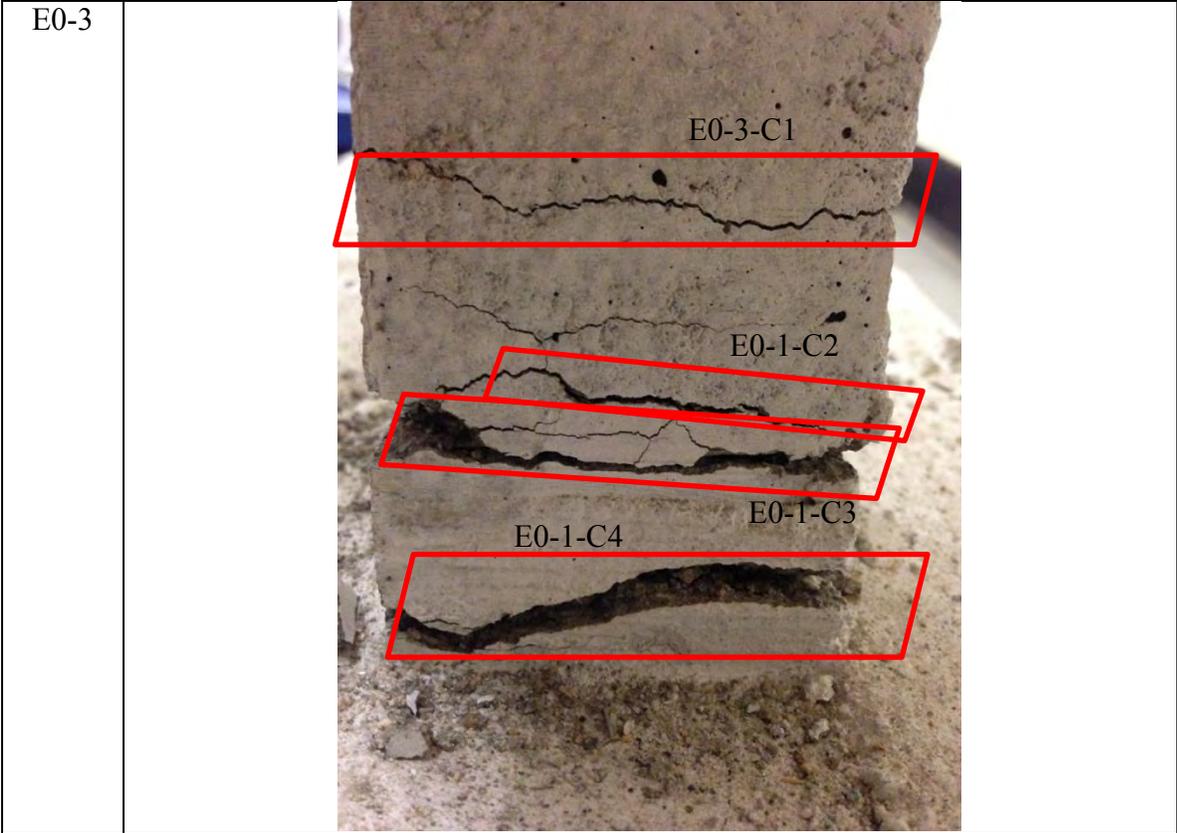
DE1-4

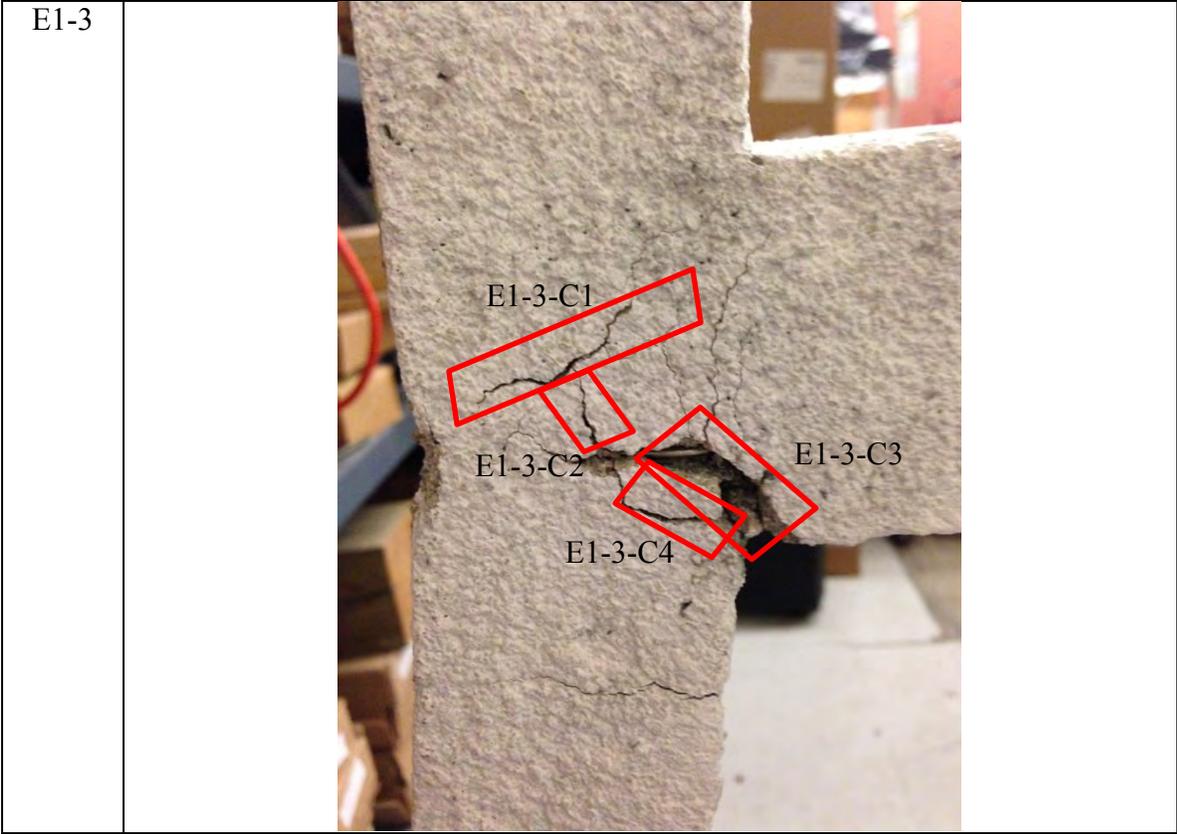
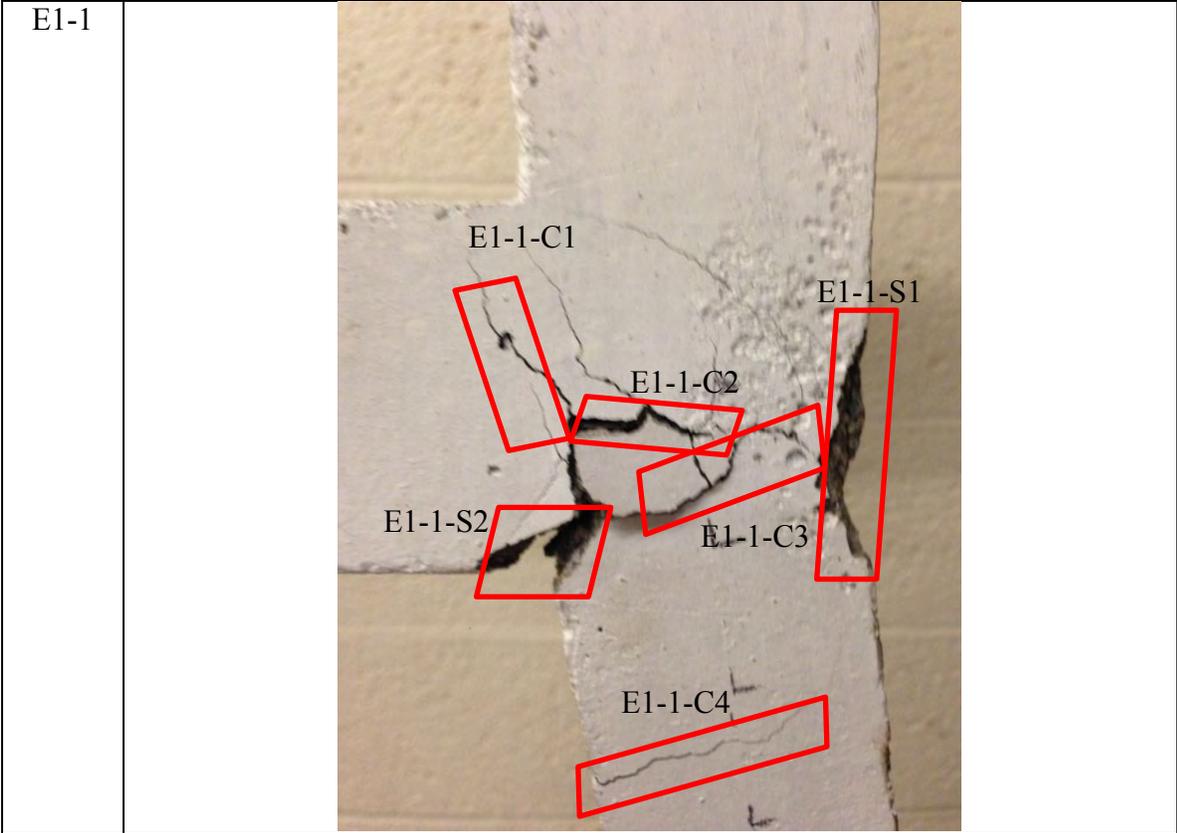


DE1-5

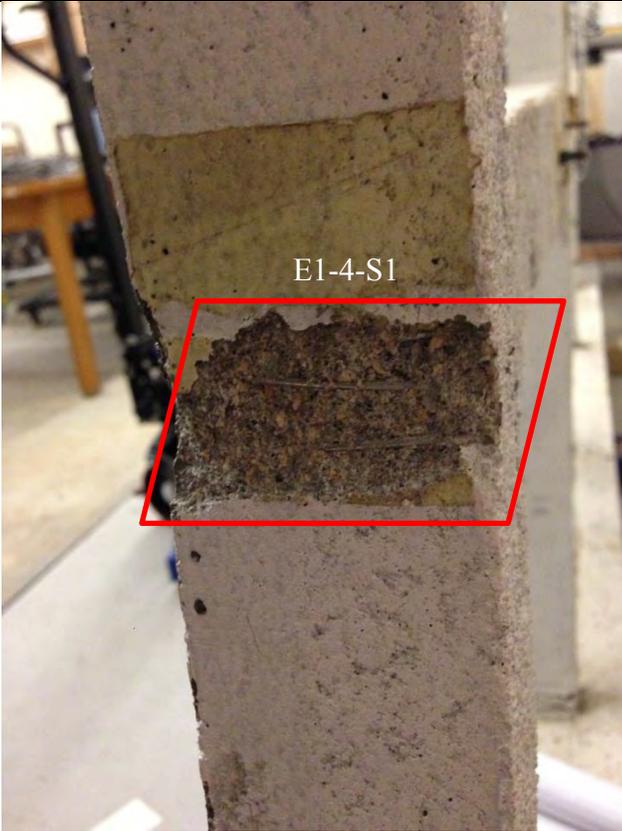




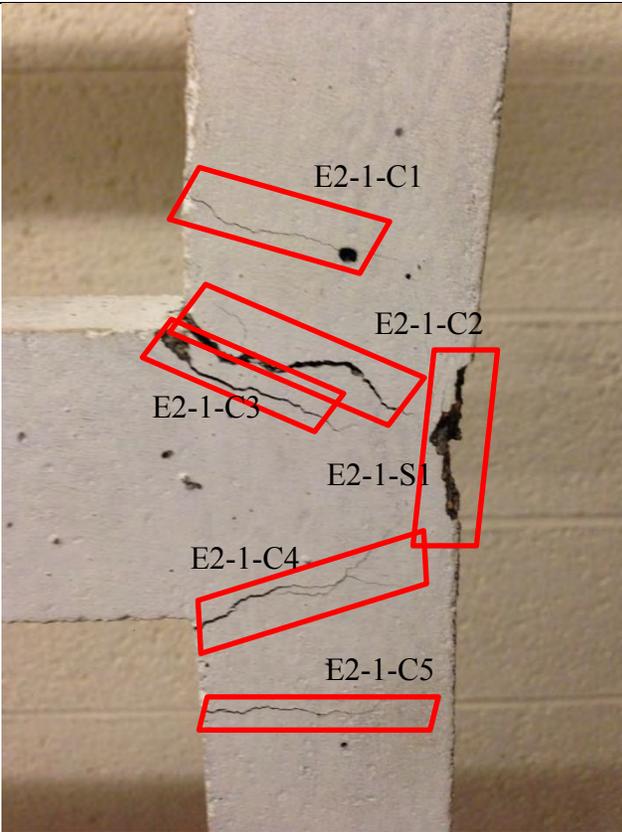




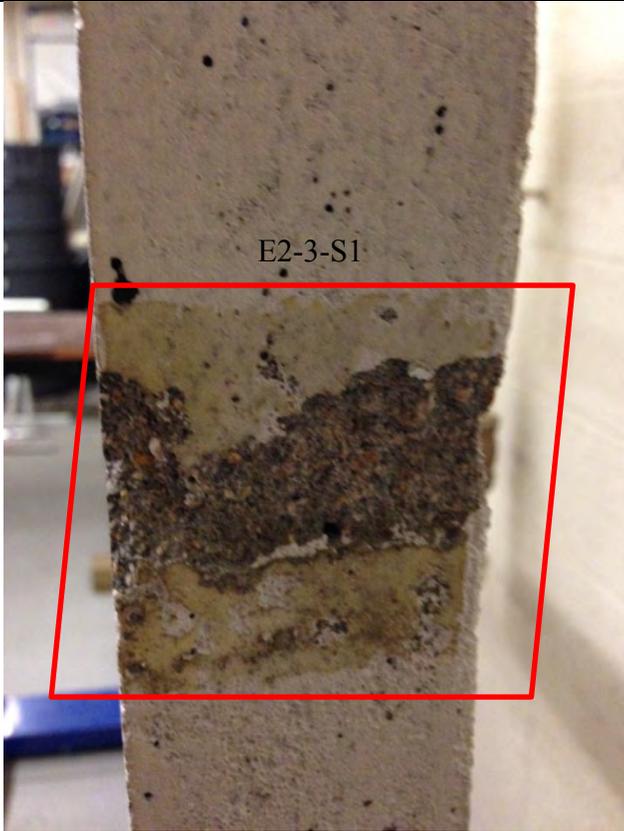
E1-4



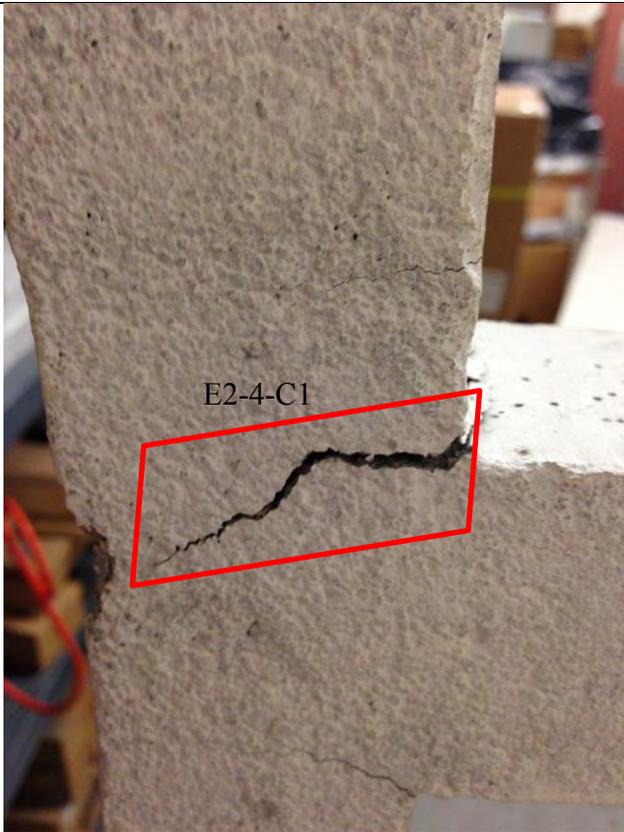
E2-1



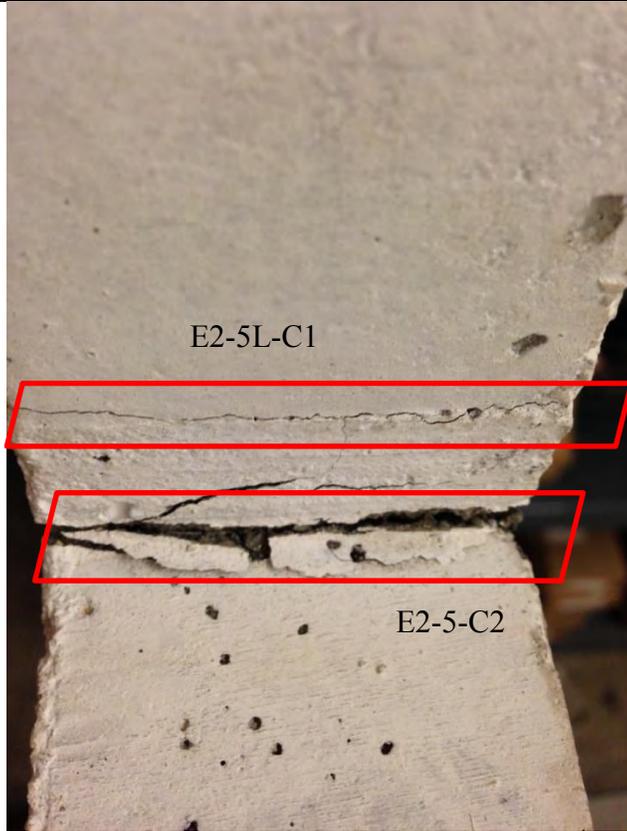
E2-3



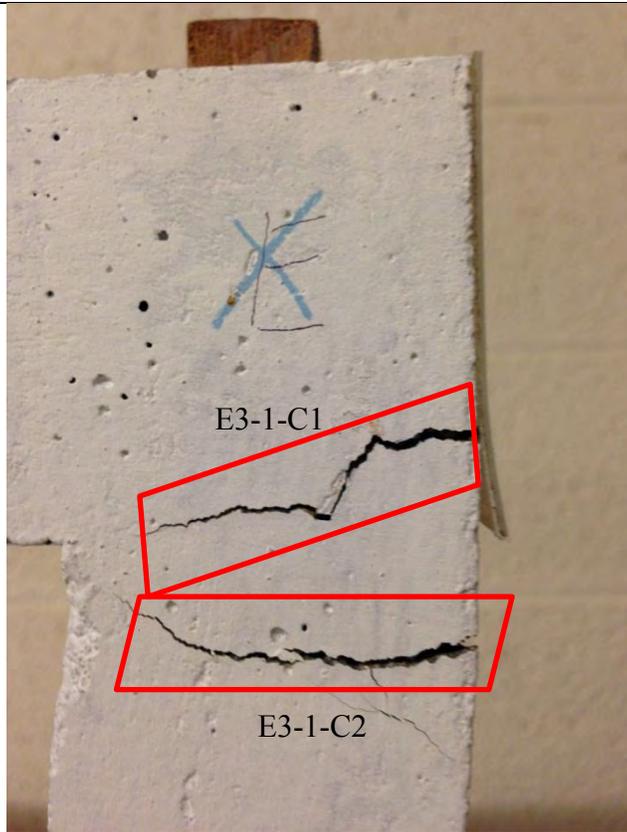
E2-4

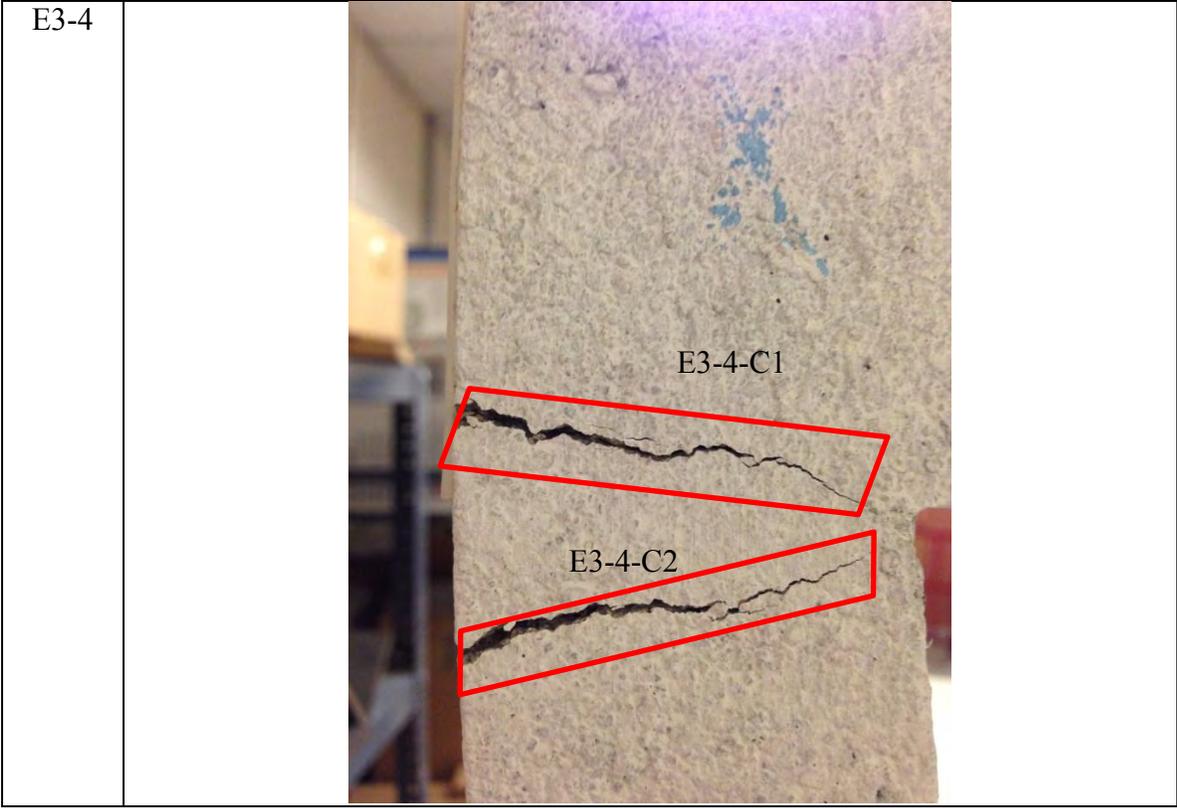
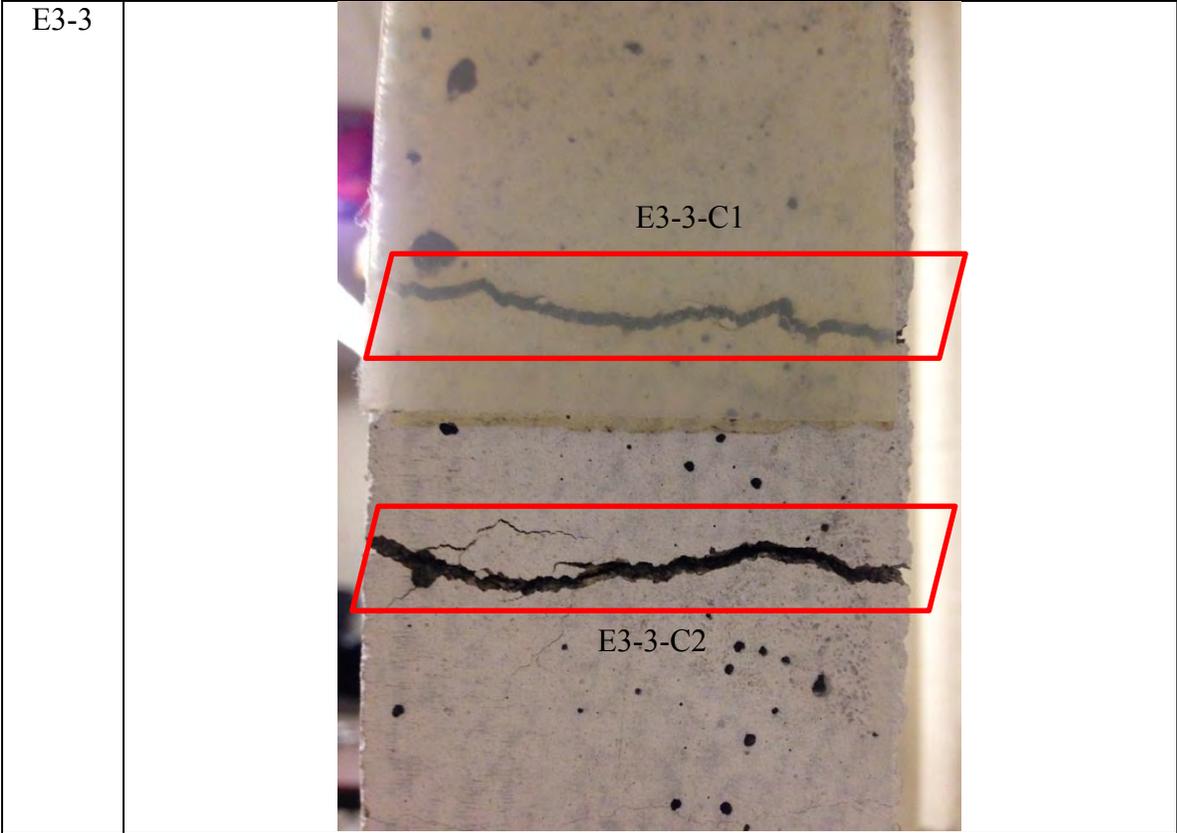


E2-5L



E3-1





## D. Appendix D – Neural Network Training Set Information

This appendix provides additional information on the patches used for training the artificial neural network described in Chapter 8. In Table D.1, for each processed surface patch, the patch number, total number of points on the patch, computed intensity threshold, total number of detected defects, total number of actual defects, and total number of false positives are listed. The total number of all detected defect clusters is computed as 201. Out of these 201 feature vectors, 74 of the feature vectors represent actual defects, and the rest, 127 clusters, are composed of false positives.

Table D.1: Information on neural network training patches.

<b>Patch Number</b>	<b># points</b>	<b>Threshold</b>	<b># Defects</b>	<b>TRUE</b>	<b>FALSE</b>
1	1239	0.63	5	3	2
2	1153	0.58	4	2	2
3	1147	0.58	2	1	1
4	1182	0.58	4	2	2
5	1209	0.54	2	1	1
6	1200	0.57	1	0	1
7	1243	0.57	3	1	2
8	1287	0.6	1	0	1
9	1199	0.58	1	1	0
10	1349	0.57	2	0	2
11	1339	0.61	4	1	3
12	1367	0.62	0	0	0
13	1500	0.59	3	2	1
14	1466	0.56	1	1	0
15	789	0.62	1	0	1
16	1561	0.64	1	0	1

<b>17</b>	1602	0.62	3	0	3
<b>18</b>	1266	0.59	2	1	1
<b>19</b>	1588	0.65	0	0	0
<b>20</b>	1700	0.69	1	0	1
<b>21</b>	1777	0.64	3	0	3
<b>22</b>	1546	0.64	1	0	1
<b>23</b>	1847	0.66	1	0	1
<b>24</b>	1833	0.67	2	0	2
<b>25</b>	1863	0.67	3	0	3
<b>26</b>	1608	0.67	2	0	2
<b>27</b>	1665	0.69	0	0	0
<b>28</b>	2043	0.68	1	0	1
<b>29</b>	2161	0.67	3	0	3
<b>30</b>	1673	0.61	0	0	0
<b>31</b>	1345	0.66	1	0	1
<b>32</b>	2464	0.71	2	1	1
<b>33</b>	1993	0.71	3	0	3
<b>34</b>	1754	0.69	0	0	0
<b>35</b>	2105	0.71	2	0	2
<b>36</b>	2225	0.58	4	0	4
<b>37</b>	1648	0.61	2	2	0
<b>38</b>	820	0.53	1	0	1
<b>39</b>	769	0.52	0	0	0
<b>40</b>	1489	0.63	4	3	1
<b>41</b>	1430	0.57	5	0	5
<b>42</b>	1380	0.61	0	0	0
<b>43</b>	1312	0.53	1	0	1
<b>44</b>	1359	0.57	1	1	0
<b>45</b>	1774	0.64	4	0	4
<b>46</b>	1404	0.64	0	0	0
<b>47</b>	1943	0.63	1	0	1
<b>48</b>	1802	0.67	0	0	0
<b>49</b>	2010	0.71	4	0	4

<b>50</b>	1679	0.73	0	0	0
<b>51</b>	2365	0.67	2	0	2
<b>52</b>	1989	0.72	1	0	1
<b>53</b>	2113	0.74	5	0	5
<b>54</b>	1799	0.73	0	0	0
<b>55</b>	2587	0.69	2	0	2
<b>56</b>	2097	0.75	5	0	5
<b>57</b>	2104	0.67	1	1	0
<b>58</b>	2055	0.69	0	0	0
<b>59</b>	2588	0.57	0	0	0
<b>60</b>	2670	0.67	4	3	1
<b>61</b>	2772	0.61	3	2	1
<b>62</b>	2124	0.62	1	1	0
<b>63</b>	1369	0.64	3	3	0
<b>64</b>	1713	0.63	3	2	1
<b>65</b>	2055	0.69	3	2	1
<b>66</b>	2641	0.66	2	2	0
<b>67</b>	2399	0.73	1	1	0
<b>68</b>	2757	0.74	5	4	1
<b>69</b>	2861	0.62	1	1	0
<b>70</b>	2732	0.72	5	4	1
<b>71</b>	3477	0.74	2	1	1
<b>72</b>	2305	0.62	2	1	1
<b>73</b>	1180	0.61	2	0	2
<b>74</b>	819	0.52	2	0	2
<b>75</b>	1293	0.62	2	1	1
<b>76</b>	1555	0.59	3	1	2
<b>77</b>	1396	0.61	0	0	0
<b>78</b>	901	0.53	2	1	1
<b>79</b>	1450	0.62	0	0	0
<b>80</b>	2058	0.64	2	0	2
<b>81</b>	1458	0.65	3	0	3
<b>82</b>	1306	0.59	1	1	0

<b>83</b>	1267	0.63	0	0	0
<b>84</b>	2230	0.69	3	0	3
<b>85</b>	1782	0.65	1	0	1
<b>86</b>	1528	0.61	2	0	2
<b>87</b>	1497	0.64	0	0	0
<b>88</b>	2712	0.69	2	0	2
<b>89</b>	2119	0.72	3	0	3
<b>90</b>	2040	0.67	0	0	0
<b>91</b>	1768	0.69	0	0	0
<b>92</b>	3466	0.74	3	0	3
<b>93</b>	2234	0.75	3	0	3
<b>94</b>	2069	0.68	0	0	0
<b>95</b>	1889	0.72	0	0	0
<b>96</b>	3838	0.78	3	0	3
<b>97</b>	1482	0.64	1	1	0
<b>98</b>	805	0.62	0	0	0
<b>99</b>	1467	0.56	3	3	0
<b>100</b>	2244	0.69	1	1	0
<b>101</b>	2399	0.58	3	3	0
<b>102</b>	2757	0.64	3	2	1
<b>103</b>	2861	0.58	2	2	0
<b>104</b>	2732	0.59	4	3	1
<b>105</b>	3477	0.61	2	2	0
<b>106</b>	2305	0.69	2	2	0
		<b>Sum</b>	<b>201</b>	<b>74</b>	<b>127</b>

## **E. Appendix E - Initial Site Investigation and Laser Scanning of Bowker Overpass**

This appendix provides detailed information on the current condition of the Bowker Overpass, Section E.1, and the followed scanning strategy Section E.2. This section finishes by presenting the captured point clouds in Section E.3.

### **E.1 Inspection and Field Report**

The two locations that are selected for scanning are already given in Section 10.1.5. Figure 10.10 shows the portion of the plan of the Bowker Overpass that highlights the scanning locations. This section provides information on the current condition of the selected scan locations.

#### **E.1.1 4FH**

Mercuri and Mirliss (2013) state that “Bridge B-16-365 (4FH) is a three span structure which is part of the Bowker Interchange. This structure carries the northbound traffic from the Bowker Interchange mainline structure (B-16-365 (4EL)) onto the Storrow Drive Westbound on-ramp. This bridge is oriented from north to south.” For the general orientation of the bridge, refer to Figure 10.9. Figures from Figure E.2 through Figure E.8 are composed of images that show several defects observed on the Bridge 4FH, Span 7; the image of this span is given in Figure E.1.



Figure E.1: Side view of Bridge 4FH, Span 7 from east.



Figure E.2: Underside of Bridge 4FH, Span 7 and delamination on the south end.



Figure E.3: Section loss on east-side exterior fascia girder on Bridge 4FH, Span 7.



Figure E.4: Corrosion on the south side of the joint of Bridge 4FH, Span 7.



Figure E.5: Corrosion and delamination on the north side of the joint of Bridge 4FH, Span 7.



Figure E.6: Concrete spalling and cracking on the underside of Bridge 4FH, Span 7 deck.



Figure E.7: Cracks and corrosion on the underside of Bridge 4FH, Span 7 deck.



Figure E.8: Concrete spalling and exposed reinforcement on the underside of Bridge 4FH, Span 7 deck.

### E.1.2 4FJ

Mirliss et al. (2013) state that “Bridge B-16-365 (4FJ) is a three span structure which is part of the Charlesgate Bowker Interchange. This structure carries the northbound off-ramp traffic from the Bowker Interchange mainline structure (B-16-365(4EL)) onto Storrow Drive eastbound.” For the general orientation of the bridge, refer to Figure 10.9. Figures from Figure E.10 through Figure E.13 provide images that show several defects observed on the investigated corner of the Bridge 4FJ; the image of this corner is given in Figure E.9.



Figure E.9: Side view of the joint from the east on Bridge 4FJ with concrete spalling, delamination, etc.



Figure E.10: Section loss, concrete spalling and exposed reinforcement from the exterior view of the joint of Bridge 4FJ.



Figure E.11: Concrete spalling under the expansion joint of Bridge 4FJ.



Figure E.12: Corrosion on the north side of the joint of Bridge 4FJ.



Figure E.13: Corrosion on the south side of the joint of Bridge 4FJ.

## E.2 Laser Scan Locations for Field Scanning

### E.2.1 4FH – Span 7 – Top and Bottom Deck

In order to provide complete coverage on the underside and east elevation of Span 7, the following plan for laser scanner locations is executed. Green rectangles represent the locations for the laser scanner. Additionally, four scans were captured from top of the deck; these four scans were equally spaced along the entire length of Span 7. In total, nine scans were captured. Figure E.14 and Figure E.15 show the locations for the laser scanner on the plan of Span 7. Table E.1 and Table E.2 consist of the images taken during the field scanning of the Bridge 4FH, Span 7; each image is followed by the location information.

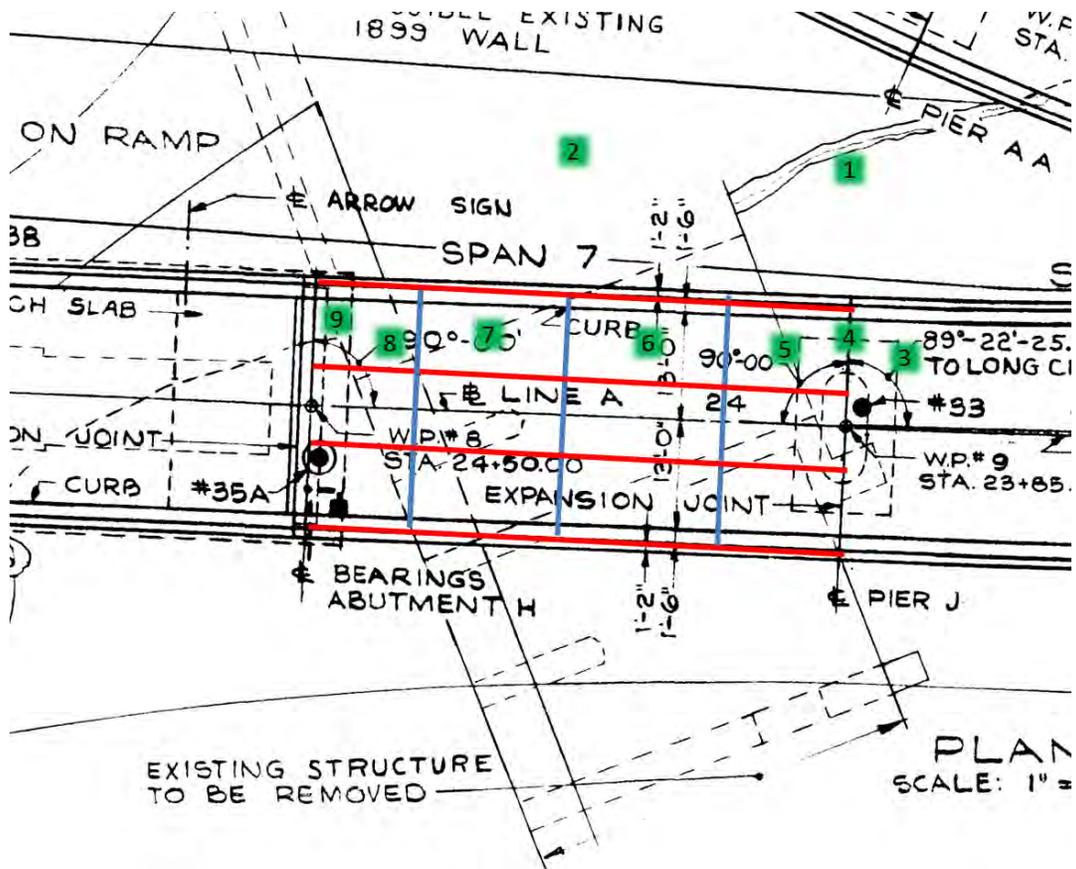


Figure E.14: Laser scanner locations for bottom deck of Bridge 4FH, Span 7.

Table E.1: Images of the scanner locations captured during field scanning of Bridge 4FH, Span 7, bottom deck.



4FH, Span 7, bottom deck scan location 1.



4FH, Span 7, bottom deck scan location 2.



4FH, Span 7, bottom deck scan location 3.



4FH, Span 7, bottom deck scan location 4.



4FH, Span 7, bottom deck scan location 5.



4FH, Span 7, bottom deck scan location 6.



4FH, Span 7, bottom deck scan location 7.



4FH, Span 7, bottom deck scan location 8.



4FH, Span 7, bottom deck scan location 9.

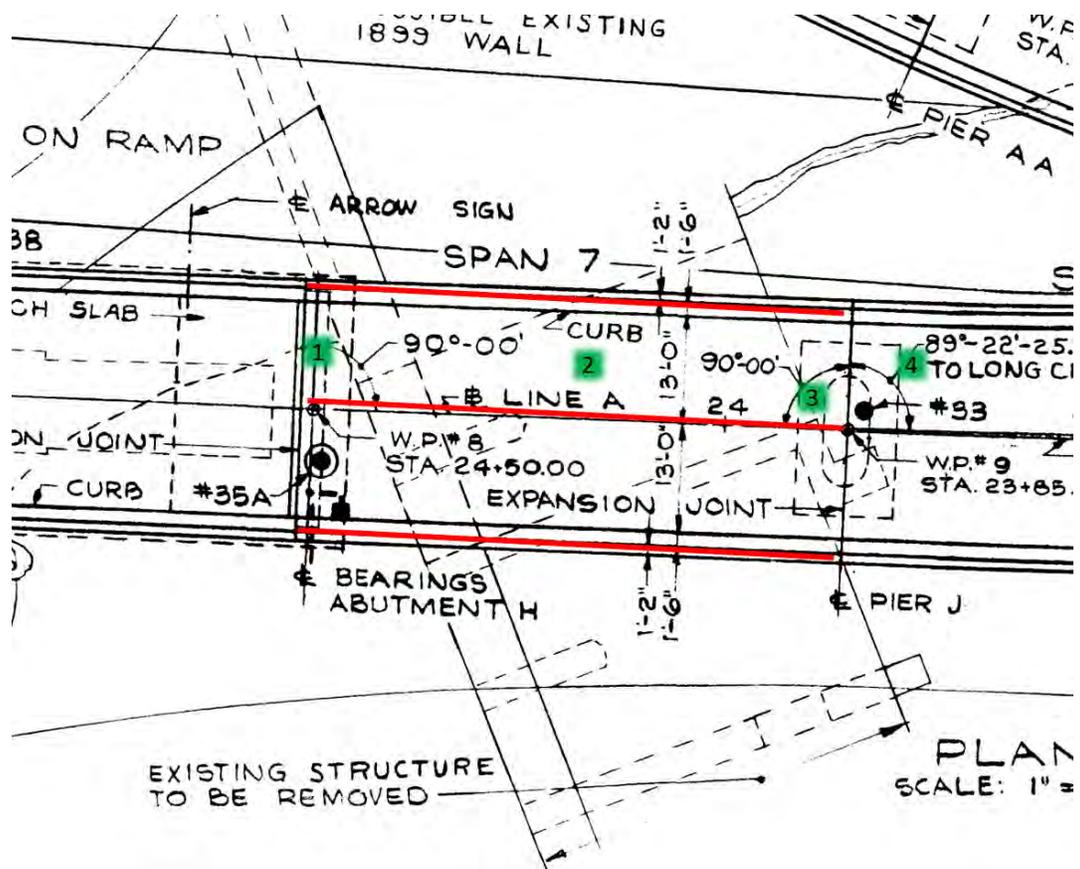


Figure E.15: Laser scanner locations for top deck of Bridge 4FH, Span 7.

Table E.2: Images of the scanner locations captured during field scanning of Bridge 4FH, Span 7, top deck.



4FH, Span 7, top deck scan location 1.



4FH, Span 7, top deck scan location 2.



4FH, Span 7, top deck scan location 3.



4FH, Span 7, top deck scan location 4.

### E.2.2 4FJ - Joint

In order to provide complete coverage at the joint that connects Span 23 and Span 24 on Bridge 4FJ, the following plan for laser scan locations is executed. Green rectangles represent the locations for the laser scanner. Figure E.16 shows the five locations for the laser scans for the joint. Table E.3 is composed of the images taken during the field scanning of the selected joint of the Bridge 4FJ; each image is followed by the location information.

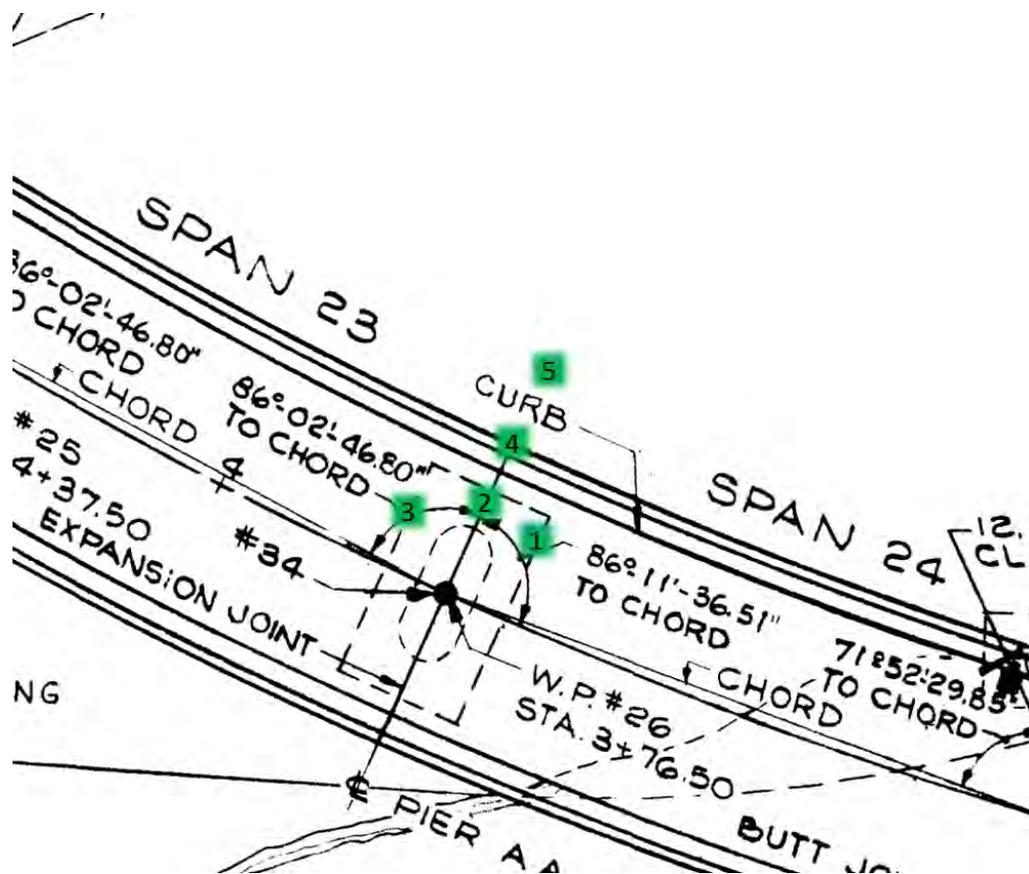


Figure E.16: Laser scan locations for joint between Span 23 and Span 24.

Table E.3: Images of the scanner locations captured during field scanning of Bridge 4FJ, corner.



4FJ, corner, scan location 1.



4FJ, corner, scan location 2.



4FJ, corner, scan location 3.



4FJ, corner, scan location 4.



4FJ, corner, scan location 5.

### E.3 Captured Point Clouds

Laser scanning is performed using Faro Focus 3D – 120 in high-resolution mode. 3D data is captured along with the images. The scan registration and texture-mapping are completed using Faro Scene software. Figure E.17 shows the entire point cloud, which is the combination of 18 scans. Figure 10.12 and Figure 10.13 represent the isolated point clouds for the Bridge 4FH, Span 7 and Bridge 4FJ, corner respectively.



Figure E.17: Entire point cloud.

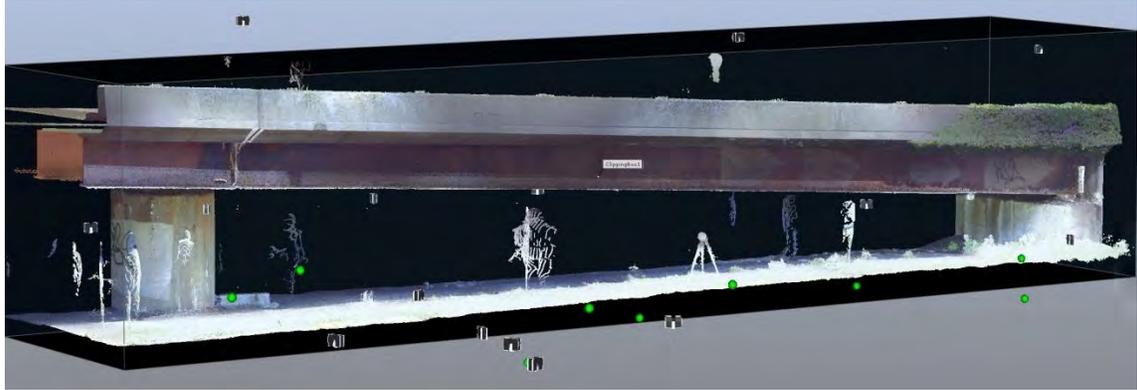


Figure E.18: Point cloud of 4FH, Span 7.

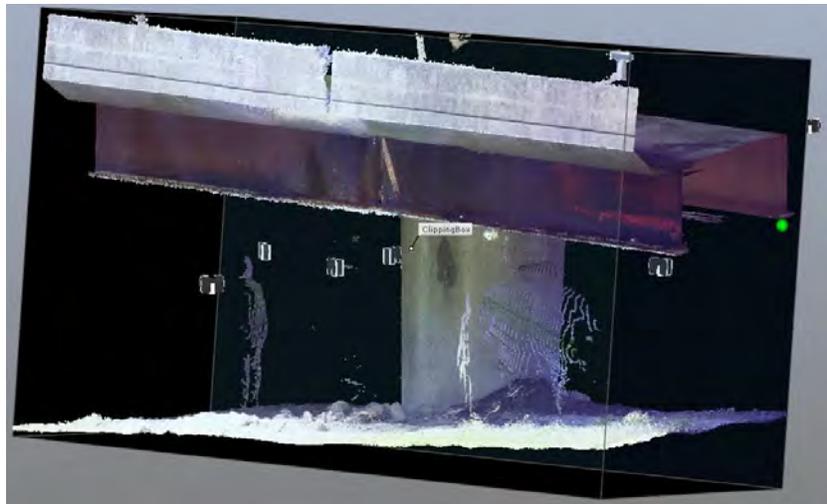


Figure E.19: Point cloud of 4FJ, corner.



# Northeastern

## Department of Civil and Environmental Engineering Reports

### Northeastern University

<b>REPORT NO.</b>	<b>AUTHORS</b>	<b>TITLE</b>	<b>DATE</b>
NEU-CEE-2014-03	Guldur, B.; Hajjar, J F.	Laser-Based Structural Sensing and Surface Damage Detection	September 2014
NEU-CEE-2014-02	Saykin, V. V.; Song, J.; Hajjar, J. F.	A Validated Approach to Modeling Collapse of Steel Structures	September 2014
NEU-CEE-2013-02	Denavit, M. D.; Hajjar, J. F.	Description of Geometric Nonlinearity for Beam-Column Analysis in OpenSees	September 2013
NEU-CEE-2013-01	Hajjar, J. F.; Sesen, A. H.; Jampole, E.; Wetherbee, A.	A Synopsis of Sustainable Structural Systems with Rocking, Self-Centering, and Articulated Energy-Dissipating Fuses	June 2013
NEU-CEE-2011-01	Hajjar, J. F.; Guldur, B.; and Sesen, A. H.	Laboratory for Structural Testing of Resilient and Sustainable Systems (STReSS Laboratory)	September 2011