

NSEL Report Series
Report No. NSEL-015
June 2009

Data Processing of Laser Scans Towards Applications in Structural Engineering



Sara B. Walsh
Jerome F. Hajjar



Department of Civil and Environmental Engineering
University of Illinois at Urbana-Champaign

UILU-ENG-2009-1803



ISSN: 1940-9826

The Newmark Structural Engineering Laboratory (NSEL) of the Department of Civil and Environmental Engineering at the University of Illinois at Urbana-Champaign has a long history of excellence in research and education that has contributed greatly to the state-of-the-art in civil engineering. Completed in 1967 and extended in 1971, the structural testing area of the laboratory has a versatile strong-floor/wall and a three-story clear height that can be used to carry out a wide range of tests of building materials, models, and structural systems. The laboratory is named for Dr. Nathan M. Newmark, an internationally known educator and engineer, who was the Head of the Department of Civil Engineering at the University of Illinois [1956-73] and the Chair of the Digital Computing Laboratory [1947-57]. He developed simple, yet powerful and widely used, methods for analyzing complex structures and assemblages subjected to a variety of static, dynamic, blast, and earthquake loadings. Dr. Newmark received numerous honors and awards for his achievements, including the prestigious National Medal of Science awarded in 1968 by President Lyndon B. Johnson. He was also one of the founding members of the National Academy of Engineering.

Contact:

Prof. B.F. Spencer, Jr.
Director, Newmark Structural Engineering Laboratory
2213 NCEL, MC-250
205 North Mathews Ave.
Urbana, IL 61801
Telephone (217) 333-8630
E-mail: bfs@illinois.edu

This technical report is based on the first author's M.S. thesis under the same title which was completed in April 2009. The second author served as the thesis advisor for this work.

Funding for this research was provided by the University of Illinois at Urbana-Champaign. The authors thank Eric Becker, Kenneth Lyons, and Jeff Zucali from Faro Technologies, Inc. and Daniel Borello from the University of Illinois at Urbana-Champaign for assisting with data collection. 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 sponsors.

The cover photographs are used with permission. The Trans-Alaska Pipeline photograph was provided by Terra Galleria Photography (<http://www.terragalleria.com/>).

ABSTRACT

This research is investigating the use of high resolution 3D terrestrial laser scanners as tools to capture geometric data of complex scenes for structural engineering applications. Laser scan technology is continuously improving, with commonly available scanners now able to capture over 100,000 points per second with an accuracy of ~ 1 mm. This research focuses on determining the applicability of laser scanning to structural engineering applications, including structural health monitoring, collapse assessment, and post-hazard response assessment. One of the keys to this work is to establish a process for extracting important information from raw laser-scanned data sets. Once such a process is fully developed, laser scanning has the potential to play an important role in linking and sharing experimental and computational structural research. Previous work in this area has created a foundation of basic data processing steps. Additional steps and modifications to existing algorithms are presented to advance the performance of data processing on complete point clouds. The outcome of this research enables processing of lasers scans of complex structures to assist with advanced analysis, monitoring, and assessment.

CONTENTS

CHAPTER 1: INTRODUCTION.....	1
CHAPTER 2: LASER SCAN DATA PROCESSING	6
CHAPTER 3: DATA PROCESSING RESULTS FROM TESTBED BRIDGE EXAMPLE.....	24
CHAPTER 4: CONCLUSIONS	36
REFERENCES	38
APPENDIX A: ADDITIONAL SCAN IMAGES.....	40

INTRODUCTION

Laser scanning is a technology that uses light to capture the 3D position of points in space. The capabilities have advanced significantly in recent years, and it has been gaining more recognition as a tool for applications in numerous fields. It is now possible to collect millions of data points accurate to within millimeters in a matter of minutes. The limiting factor preventing laser scanners from being more widely utilized in more fields is data processing. A laser produces what is known as a ‘point cloud’ – a collection of coordinates marking the position of points in 3D space. A laser can only record the location of an object in space; it has no way to distinguish one surface (or object or material) from another. Thus, while the ability to collect large volumes of highly accurate geometric data is powerful, the technology still has limitations. There is currently no comprehensive software available to take laser point clouds and produce useful models. Due to the challenges of this type of data processing, laser scanning has not been adopted in many fields where it could be of great use.

1.1 Laser Scanning Technology

Laser scanners use light to measure the location of points in space. In a time-of-flight scanner, a pulse of light is sent out, and, when it hits a surface, the pulse is reflected back into the scanner. The time of travel is measured and converted to a distance. This distance, along with the angle at which the pulse is sent out, is used to record 3D coordinates of the point in space at which the pulse encountered a surface. In a phase-shift scanner a sinusoidal laser pulse is emitted and the shift in phase is measured upon its return. This shift translates to a time which is then used in the same way as with time-of-flight scanners. In this manner, the phase-shift scanner can send out multiple different waves of light at the same time to measure different points, thus dramatically increasing the number of points measured per second as compared to a time-of-flight laser. The collection of 3D coordinates identifying points in space produced by a scanner constitutes a point cloud. Scanners can only see visible surfaces within their line of sight. Due to this characteristic, in most situations it is necessary to scan a scene multiple times from a variety of locations to obtain a more comprehensive 3D image.

The laser scanners used to create the point clouds seen in this work are manufactured by Faro. As an example, the Faro Photon 80 used to collect the testbed bridge data is a phase-shift laser that has a range of 76 m. It can capture 120,000 points per second with accuracy within ± 2 mm. The field of view is 360° horizontally, and 320° vertically. A single scan taken at one quarter of full resolution, as the bridge scans were, can be completed in less than 6 minutes.

Laser scanning has many advantages over other instruments used to capture geometry or record measurements. The main advantage is that lasers can capture the 3D position of virtually all visible points on a surface for an expansive field of view. This capability

leads to very dense data sets that provide a wealth of geometric information. As new generations of lasers are produced, their accuracy, resolution, and time required to complete a scan are continually improving. In addition, the most advanced lasers available are able to complete on the order of thirty complete scans per second, thus enabling these lasers to be used during quasi-static experimental testing of structures to record movement of the structure. Currently, instruments such as Krypton or linear variable differential transformers (LVDTs) similarly capture position and displacement, often with greater accuracy than common laser scanners, but these measurements are feasible only for discrete points.

1.2 Applications in Structural Engineering

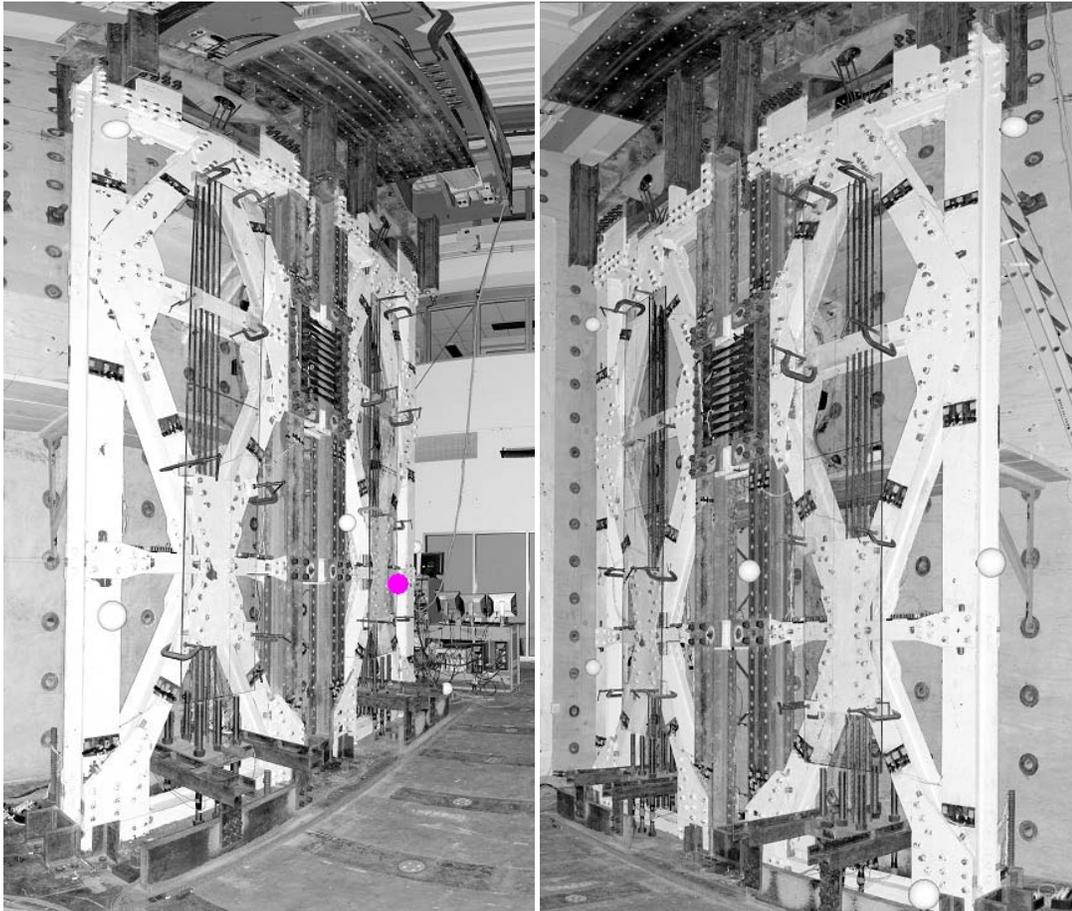
Laser scanning has been used as a tool in many fields, including construction and surveying. This section presents applications in structural engineering where laser scanning could be beneficial.

Quasi-Static Experimental Testing

In large-scale experimental testing of structures, laser scanning has the potential to provide valuable information that could create a link to help connect experimental and computational structural research. As an example, Figure 1.1a shows a planar view of a 360° single scan of an experimental setup of a large-scale test being conducted on a seismic force resisting system consisting of two large braced steel frames joined by energy dissipating fuses. Figure 1.1b shows zoomed views of the frame specimens. Three scans captured with a Faro LS880 scanner (accuracy: $\pm 3\text{mm}$) were taken from different points around the frames and registered into one 3D point cloud to create a coherent 3D model of the test specimen. Figure 1.2 shows a front-view photograph of the frames next to the same view of the registered point cloud generated by the scans.



(a) Full view of example point cloud.



(b) Zoomed image of point clouds of laboratory specimen (Faro, 2008).

Figure 1.1: Example point cloud of laboratory scene (Faro, 2008).

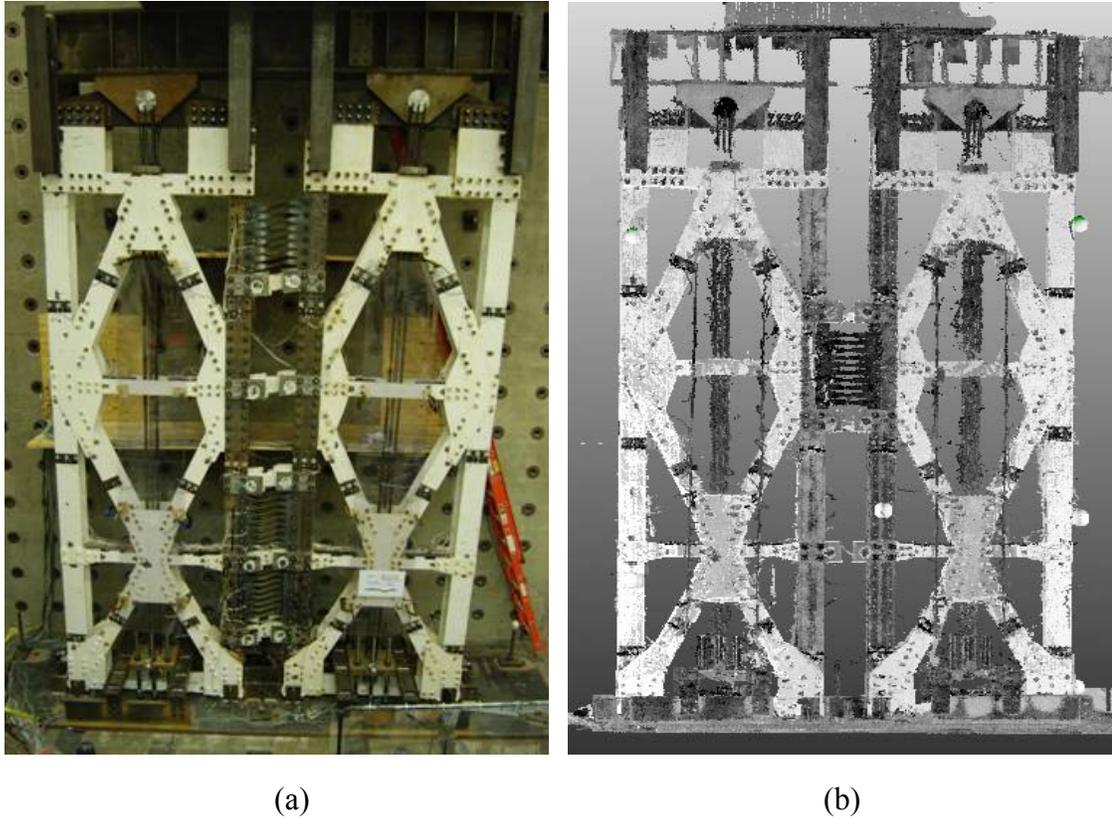


Figure 1.2: Photograph (a) and registered point cloud (b) of scanned laboratory specimen.

In Situ Measurements

Laser scanning may also be used to capture in situ damage or collapse of a structure. Individual laser scans of a scene can be captured from different viewpoints to permit the creation of a complete 3D record of a structure. Newer laser technologies are also able to scan under water.

An example of this type of scenario is the collapse of the I-35W bridge in Minneapolis, Minnesota in 2007. Engineers used a laser scanner manufactured by Riegl to capture the geometry of the collapsed structure. The intent was to use the data to assist with verification for failure analysis of the bridge. Utilizing laser scanner technology allowed the geometric data of the collapsed structure to be captured quickly and accurately.

Recently, work has been done exploring the use of lasers for tracking in-situ deformation of undamaged structural components, such as beams and columns (Park et al., 2007; Gordon and Lichti, 2007; Hawarey and Falk, 2004). This application could be developed further for use in health monitoring of existing structures. Tracking of components with laser scanning during construction has also been investigated (Bosche and Haas, 2008).

Potential Uses of Structural Laser Scan Data

If data is collected for a static structure, the geometry could be meshed and then used as a basis for discrete, applied, or finite element modelling. Creating a model directly from scan data would potentially save the time required to build a model from scratch. It

would also ensure greater accuracy of the model by incorporating exact actual dimensions, imperfections, and other details that a manually generated model may exclude (Arayici, 2007).

Data from a static structure could also be used to map sensors onto a model to compare analysis and laboratory or field measurements. This type of application would be useful for creating a visualization tool for viewing and sharing research results.

Augmented reality (AR) is a technology that combines real-world data with computer-generated data. For example, a real-world laser scan of a structure could be augmented with a computer-generated surface mesh and viewed through special goggles that provide the perspective of standing inside the actual structure. This type of tool could be useful in many scenarios ranging from sharing a laboratory setup to looking inside a collapsed building. Using laser scan data to guide robots through environments is an application that has been recently explored (Rusu et al., 2008; Nüchter and Hertzberg, 2008).

1.3 Objectives

This project stems from a desire to explore the potential of using laser scanning as a tool in structural engineering research. A literature review revealed the disparity in quality between laser scanning hardware and software, whereby the point cloud scans are clear and have high resolution, but few software systems exist that process the data in a manner suitable for use in structural engineering applications. This knowledge initiated the attempt to begin building a strategy for laser data processing.

The objectives of this project were to explore the potential for laser scanning in the field of structural engineering towards applications such as structural health monitoring and collapse assessment, complete a literature review about the status of data processing of laser scans, and initiate implementation of capabilities for processing of complex structural point clouds.

LASER SCAN DATA PROCESSING

To utilize laser data for specific applications, it is necessary to process the points to derive information and evolve the data into a more useful form. Processing begins with extracting as much information as possible about each individual point and then employing relationships between points to derive properties of the underlying surface. A choice of methodology is required, followed by specific processing steps, including: registration, neighborhood selection, outlier removal, tangent plane estimation, tangent plane orientation, feature point detection, segmentation, surface fitting and object detection. Two point clouds are utilized in this chapter for demonstration of the algorithms discussed. For initial development, a synthetic point cloud generated by the author was used. The synthetic cloud consists of simple surfaces and served as an example to check basic functionality of algorithms. The point cloud captured of the laboratory specimen shown in Figure 1.2 was also processed to determine and address the challenges of actual laser scan data. To determine the effectiveness of the algorithms on an example from the field, a point cloud of a collapsed bridge was captured in situ, and the results of processing that cloud are discussed in Chapter 3.

2.1 Point-based vs. Mesh-based Methods

There are two schools of thought on how to approach a raw point cloud: point based methods and mesh based methods. Point based methods begin processing a data set by directly utilizing the points themselves (Pauly, Keiser, and Gross, 2003; Demarsin et al., 2006; Daniels et al., 2007). Mesh based methods begin by creating a triangulation, or surface reconstruction, of the cloud (Huang and Menq, 2002; Bényko et al., 2001).

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 (Vanco and Brunnett, 2007). There has been less work done in the literature on point-based methods as compared to mesh-based methods. During this project, literature regarding data processing via point-based methods was explored, and several algorithmic extensions were developed and implemented.

Point-based methods utilize local neighborhoods of points to estimate local surface properties; accordingly, processing begins with setting the parameters for neighborhood selection. The tangent plane and surface variation of the local underlying surface are estimated at each point based on the neighborhood. Properties of the neighborhood are also utilized to remove outlying and extraneous data points to simplify the data set. Once the cloud is edited using individual point information, inter-point relationships are used to ensure consistency and improve property estimates where necessary. Inter-point relationships also applied towards region growing for segmentation. Segments are fitted

with surfaces, and inter-surface relationships are used to detect objects. Each of these steps is discussed in detail in the following sections.

2.2 Registration

Registration is a pre-processing step required if multiple scans are taken to capture a scene. It aligns and combines multiple data sets into one complete 3D cloud. Several approaches have been developed to address the registration task. One of the most popular algorithms is the Iterative Closest Point (ICP) algorithm (Besl and McKay, 1992) which requires an initial rough alignment between two clouds and then iteratively searches for point matches to solve for a transformation. Other methods utilize targets (Yu, Ferencz, and Malik, 2001) or feature points (Gelfand et al., 2005; Barnea and Filin, 2008) to determine the required transformations. However, registration is now a tool that is commonly available in laser manufacturer software programs, and hence, it was not a focus of this work.

2.3 Local Neighborhood Selection

To estimate local properties, it is first necessary to select a local neighborhood of points around each point. Assigning the number of points to be included, the ‘k nearest neighbors’ (KNN) approach (Rabbani et al., 2006; Hoppe et al., 1992; Vanco and Brunnett, 2004), is advantageous as it ensures that every neighborhood will be defined by the same number of points, regardless of changes in density of the point cloud. Point density is often an indicator of noise; larger areas are required to achieve quality results in noisy regions, and KNN accounts for this. Additionally, the degenerate case of a point with no neighbors will never be encountered with KNN. Defining a radius, or fixed distance, is another approach for neighborhood selection. ‘Fixed distance neighbors’ (FDN) are less adaptive to changing density throughout a cloud and could encounter the scenario of no neighbors existing within the defined radius. The advantage of FDN is the range of the neighborhood will be limited to the immediate local area. The KNN approach was chosen for implementation in this work.

The properties estimated for points are dependent upon the size of the neighborhood chosen, k , so this parameter is critical. If a point cloud has a high level of noise, an increased neighborhood size is needed to avoid having the noise negatively influence the local tangent plane estimation. In general, however, smaller neighborhoods ensure estimates reflect *local* surfaces properties and minimize time and memory required for computation.

Trials revealed that a neighborhood size of ten to fifteen points generated high quality results for the synthetic point cloud utilized for initial algorithm development. A slightly larger neighborhood of twenty five to thirty five points produced good results for the testbed bridge point cloud. Potential for adaptive neighborhood selection was addressed by Hoppe et al. (1992). The premise is to increase the number of neighbors until the eigenvector associated with the tangent plane at each point becomes distinct during covariance analysis (described in Section 2.5), as this would mean the neighborhood is large enough to negate the effects of noise. This concept was not implemented for this

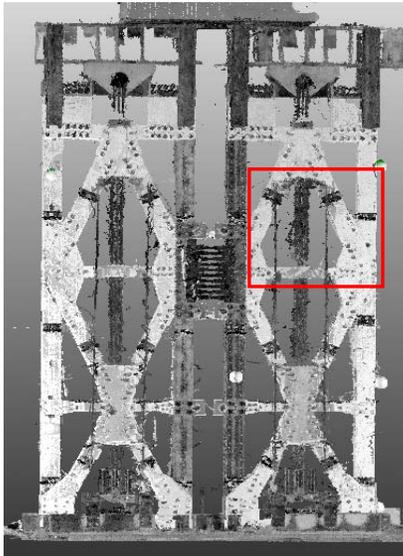
work, but pursuing adaptivity for parameter selection may be a worthwhile effort for future work.

2.4 Outlier and Extraneous Detail Detection

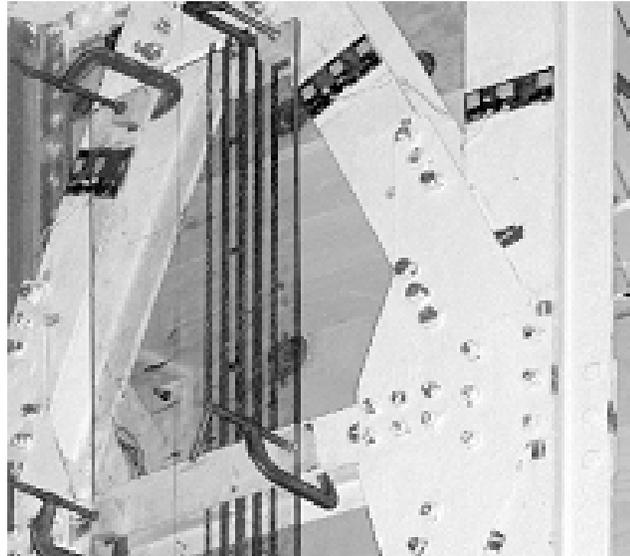
Points can be falsely identified by a laser scanner at places in a scene where no surface actually exists. This can be due to the laser beam splitting at the edge of an object or a person or object moving in the scene during scanning. Outlying points lead to errors in later processing steps, so it is critical to identify and remove them.

The following check was developed based on an algorithm by Rusu et al. (Rusu et al., 2008) to identify outliers in a cloud: For each point, a local neighborhood is selected as discussed in Section 2.3, and the average distance to a neighbor, d_n , is computed. The average distance to a neighbor, μ_d , and standard deviation, σ_d , for the entire cloud are determined. If $d_n > \mu_d + 2 \sigma_d$ for any point, the point is marked as an outlier and removed from the data set. The number of standard deviations can be adjusted based on the degree of filtering desired. In the scanned laboratory specimen, Plexiglas was used to cover the post-tensioning strands. Refraction of the laser beam through the Plexiglas resulted in a lot of noise and outlying points in that region. Figure 2.1 shows a region of the cloud with the outlier points indicated.

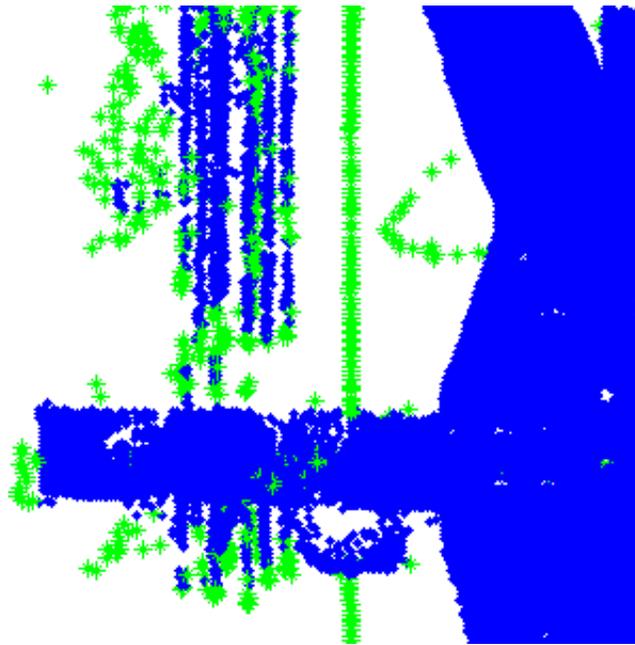
Another category of points that lead to errors in later processing steps are points recorded for small-scale details in a scan. These extraneous objects are not a part of the structure and will cause errors for processing the surrounding surfaces. An algorithm was developed for detection and removal of these points. Points detected for small, complex details were found to exhibit very high surface variation. Surface variation, γ_p , is estimated based on local neighborhood for each point in the cloud as discussed in Section 2.6. An average value, μ_γ , and standard deviation, σ_γ , for surface variation is computed for the entire point cloud. If γ_p is greater than $\mu_\gamma + 2 \sigma_\gamma$, the point is indicated as a small detail and removed from the cloud. As with outlier removal, the number of standard deviations can be adjusted to remove more or less detail based on the cloud and the desired results. Figure 2.2 shows a section of the laboratory specimen point cloud with detected small detail points due to a clamp and the Plexiglas.



(a) Point cloud

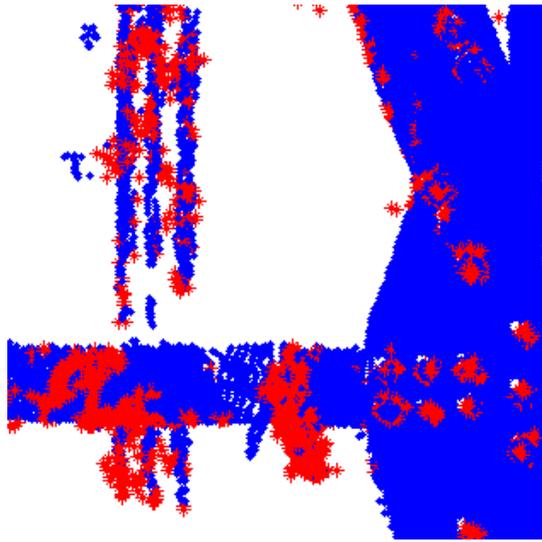


(b) Zoomed view of point cloud region with Plexiglas

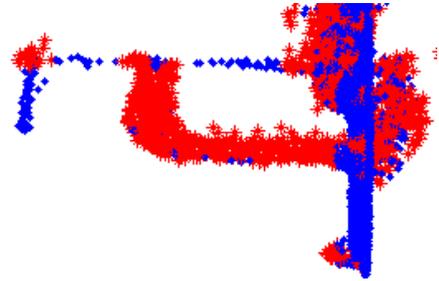


(c) Detected outlier points

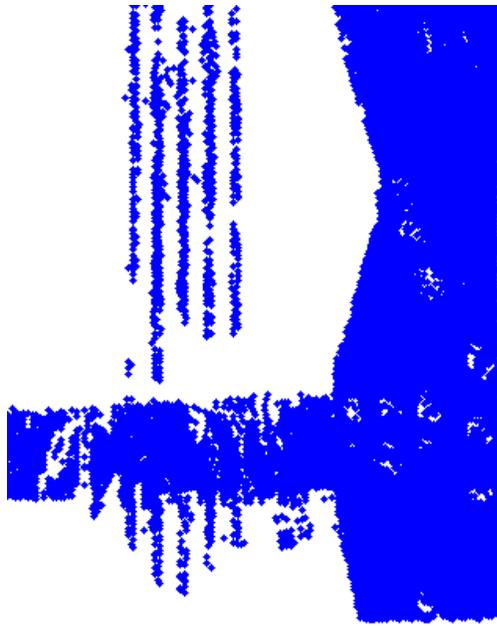
Figure 2.1: Cross-section of laboratory specimen point cloud with detected outlier points indicated.



(a) Extraneous details indicated



(b) C-clamp detail detected



(c) After removal of extraneous details

Figure 2.2: Region of laboratory specimen point cloud with extraneous points indicated and removed.

2.5 Tangent Plane Estimation

For a localized area, the tangent plane for a point, p , represents the underlying surface. The method for estimating a tangent plane from a collection of neighborhood points

implemented in this work utilizes covariance analysis (Hoppe et al., 1992; Pauly et al., 2002). The covariance matrix of a neighborhood is computed as:

$$C = \sum_{i=1}^k (p_i - \bar{p})^T (p_i - \bar{p})$$

where

$$\bar{p} = \frac{1}{k} \sum_{i=1}^k p_i$$

If $\lambda_0 \leq \lambda_1 \leq \lambda_2$ are the eigenvalues of the covariance matrix associated with the eigenvectors v_0, v_1, v_2 , the normal vector of the tangent plane is estimated as v_0 . Figure 2.3a illustrates the concept of tangent plane estimation from a local neighborhood, and Figure 2.3b shows estimated normal vectors on a portion of a point cloud.

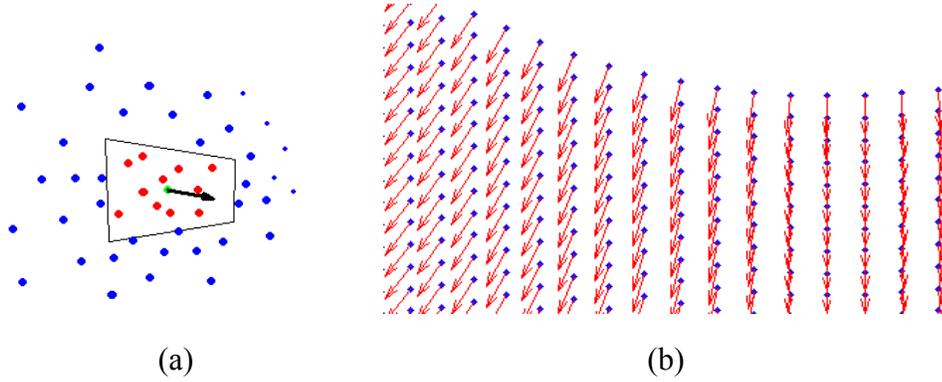


Figure 2.3: Concept of tangent plane estimation based on local neighborhood (a) and results of implemented tangent plane estimation for a region of a point cloud (b).

The process of covariance analysis and tangent plane estimation is repeated for every point in the cloud. The tangent plane normal vectors are stored for use in the next processing steps.

2.6 Curvature Estimation

Another property that can be estimated for every point based on its local neighborhood is curvature or ‘surface variation’. This indicates how much the points in a local area vary from the tangent plane. This property was utilized as mentioned in Section 2.4 for removal of extraneous points. Also, average surface variation is computed later for segments defined through region growing. The surface variation can help in determining the type of surface (planar or curved) represented by the segment. For example, a perfectly planar surface would indicate zero curvature. Curvature estimation was implemented by revisiting covariance analysis (Pauly et al., 2002). From the eigenvalues ($\lambda_0 \leq \lambda_1 \leq \lambda_2$) of the covariance matrix, curvature at a point (γ_p) is estimated as:

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

2.7 Tangent Plane Orientation

When tangent planes are initially estimated, there is a 180° ambiguity in the assignment of a normal vector. For neighboring points lying on the same surface, the normal vectors could be pointing opposite directions. It is desirable to have tangent planes for points on a single surface oriented consistently; a method for achieving this was developed by Hoppe (Hoppe et al., 1992) and adapted for this work. The orientation of a single point is assigned as desired, and the appropriate orientation is then propagated through the rest of the cloud. Graph theory utilizing spanning trees organizes the order of propagation.

In the implemented algorithm, a Euclidian Minimum Spanning Tree (EMST) is generated to form a connected path throughout the entire point cloud. It was discovered during development that the propagation of an EMST through a point cloud can be affected by the spacing of points in the cloud. Uniform point spacing can potentially prevent the tree from growing throughout the entire cloud. To avoid this, an adaptation to the original algorithm was implemented. Prior to generating the EMST, extremely small, randomly directed irregularities are introduced to the point spacing. This allows the tree to propagate properly, regardless of initial spacing, and the irregularity is then removed so it does not affect any other processing steps. After the EMST is created, its density is increased by adding edges between each point and its neighbors, creating a Riemannian graph, and the weights of the edges are defined to represent the similarity in orientation of the tangent planes of the points. Two vectors with the same orientation will have a dot product $|\mathbf{n}_i \cdot \mathbf{n}_j|$ equal to 1. Edges are assigned weights equal to $1 - |\mathbf{n}_i \cdot \mathbf{n}_j|$, so links between similarly oriented points have very low weights. Links between points with very different orientations have weights approaching 1. The MST of the Riemannian graph is traversed from one root node throughout the entire cloud to propagate tangent plane orientations in the desired direction. This path of propagation is ideal because it first checks points against nearby points with similar orientations or nearly opposite orientation (indicating a point likely on the same surface, but directed incorrectly), and it avoids crossing sharp edges (links across sharp edges would have very high weights). Along the path, the normal vector of each point is checked against the previous point. If the dot product of the two vectors is negative (indicating a difference approaching 180°), the orientation of the newest point is flipped. Figure 2.4 shows a portion of the synthetic point cloud with the minimum spanning tree for propagation generated by this algorithm. Figure 2.5 shows the estimated normal vectors at a cross section of the synthetic point cloud before and after orientation, and Figure 2.6 shows the same on a cross-section of an I-beam from the laboratory point cloud.



Figure 2.4: Synthetic point cloud with MST for propagation shown.

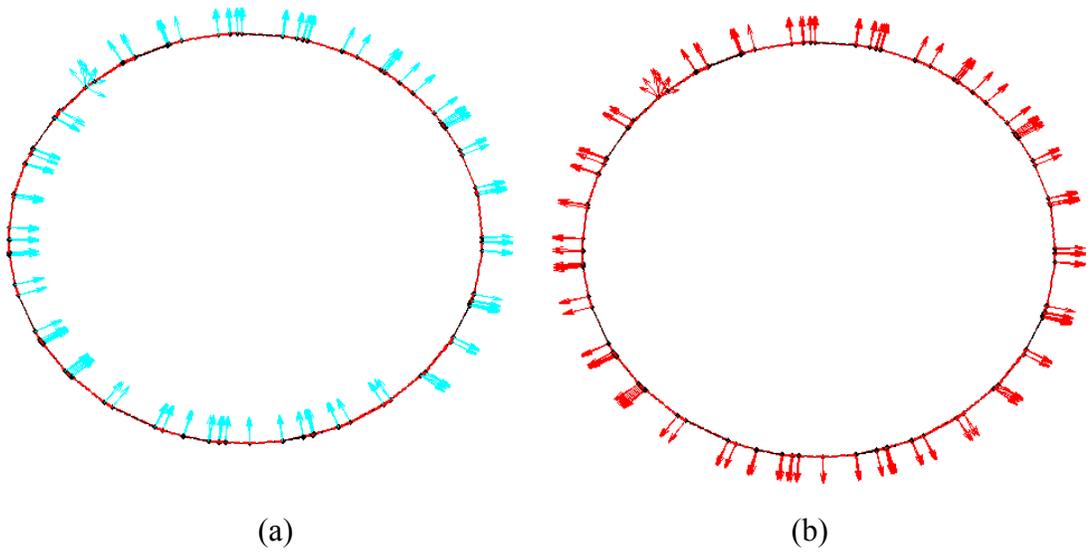
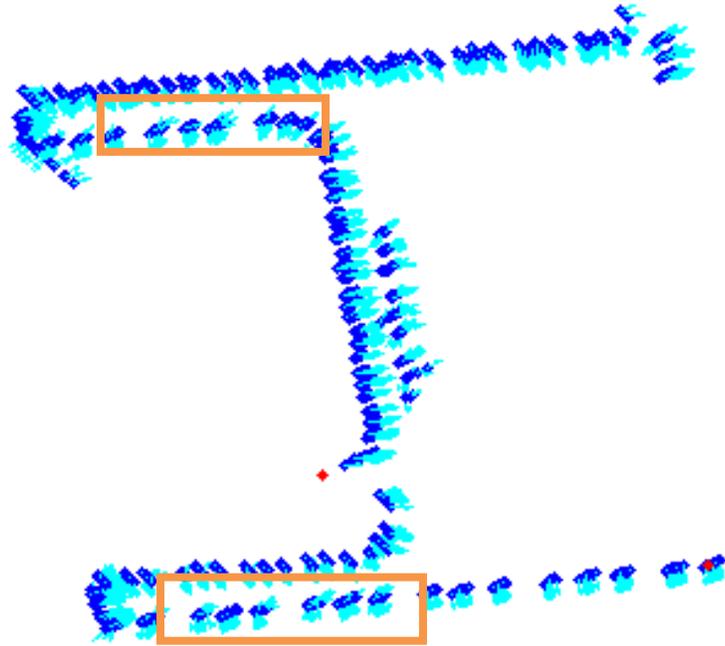
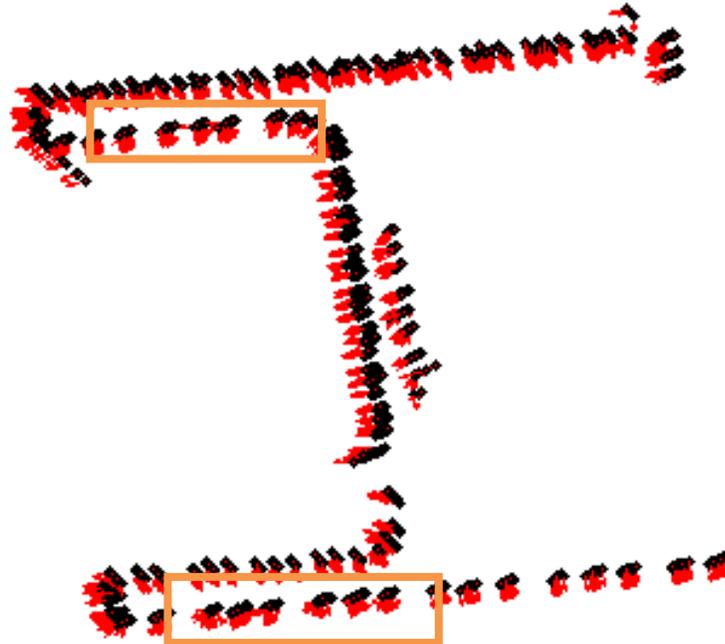


Figure 2.5: Cross-section of synthetic point cloud with normal vectors shown before (a) and after (b) orientation.



(a) Normal vectors before orientation.



(b) Normal vectors after orientation.

Figure 2.6: Cross section of I-beam from laboratory specimen point cloud with normal vectors shown before and after orientation.

2.8 Correction of Tangent Planes near Sharp Edges

Tangent planes can be incorrectly estimated near sharp edges. Points from both sides of the sharp edge are included in a local neighborhood, and the result is a tangent plane somewhere between the planes of the surfaces lying on either side of the edge. Each point is recorded by the laser for a specific surface, and each point's tangent plane should align with a real surface. Sharp edges are not as easily detected if points with 'in-between' orientations lie between the surfaces forming the edge as the 'in-between' points will reduce the angle difference measured from point to point across the edge. The algorithmic concept developed for this work to correct this issue was adapted from the algorithms by Calderon et al. (2007) and Vanco and Brunnett (2004).

All points identified as feature points after the first iteration of tangent plane estimation are revisited for a second estimation. Each point is assigned a new neighborhood of points that does not include any other points indicated as being on a sharp edge. The normal vectors of all the points in the new neighborhood are checked against one another; if a point is found to have an orientation that significantly varies from the other neighbors (indicating it belongs to a different surface beyond the sharp edge), it is removed from the estimation. The point in question is then assigned a new tangent plane based upon the new neighborhood that includes only points on one side of the sharp edge. Typically, this re-estimation results in much cleaner normal vectors near sharp edges. Figure 2.7 shows an example of normal vectors near a sharp edge after the first and second iterations. Initially, at the corner there is an estimated normal vector approximately 45° off from the orientations of the surfaces on either side of it, and vectors on either side of the corner are also skewed. After re-estimation, every normal vector aligns with a surface of the point cloud.

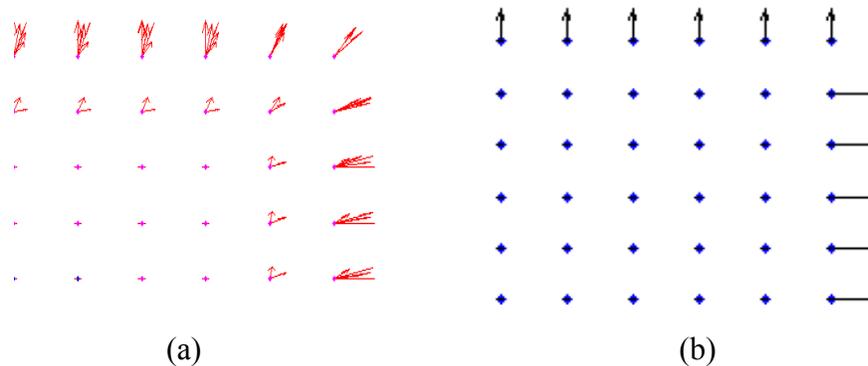


Figure 2.7: Point normal estimations after (a) first and (b) second iterations.

Figure 2.8 shows normal estimation near sharp edges in context with the synthetic point cloud. Prior to re-estimation, normal vectors near the boundary between the cylindrical and planar surfaces are skewed in multiple directions. After re-estimation, all the vectors are either horizontal to align with the cylindrical surface or vertical to align with the plane. Figure 2.9a shows a cross-section of a portion of the synthetic cloud before orientation of the normal vectors using the MST and re-estimation of the vectors; Figure 2.9b shows the same cross-section after orientation and re-estimation. The zoomed

section shows how much cleaner the normal vector estimates are after these additional processing steps.

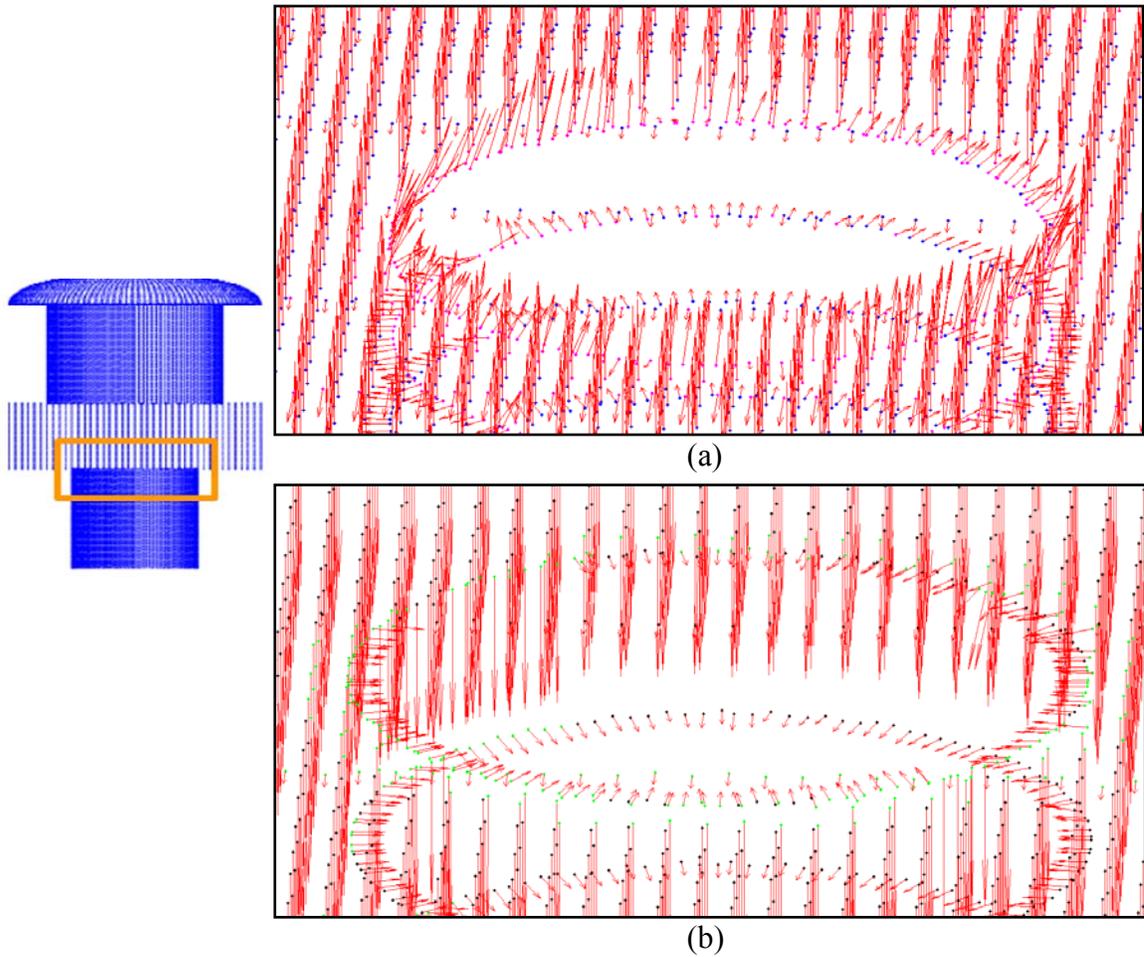
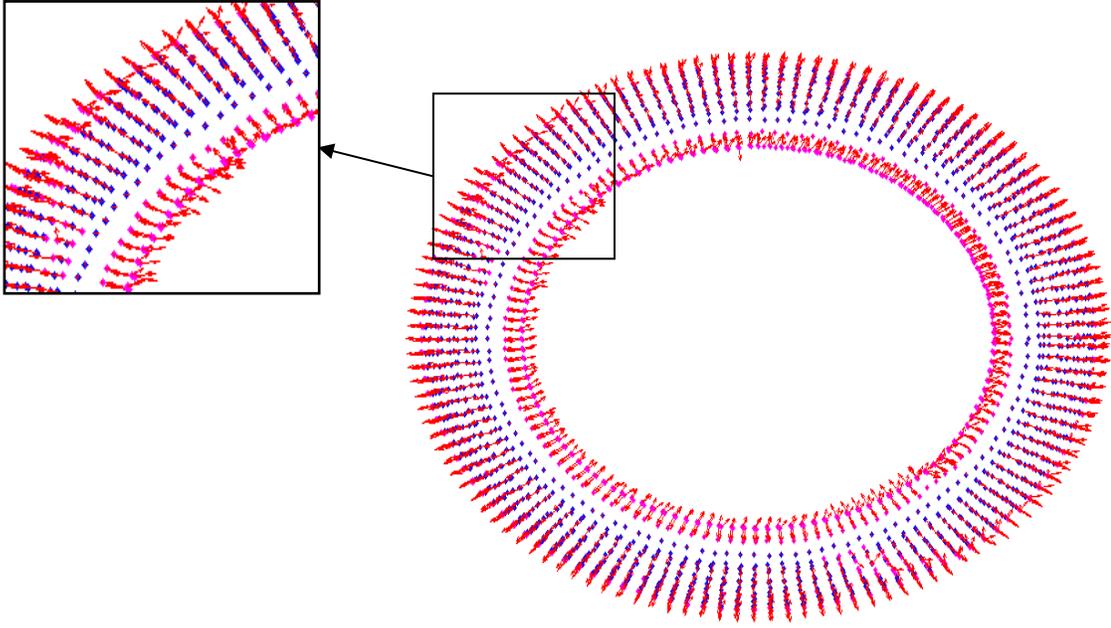
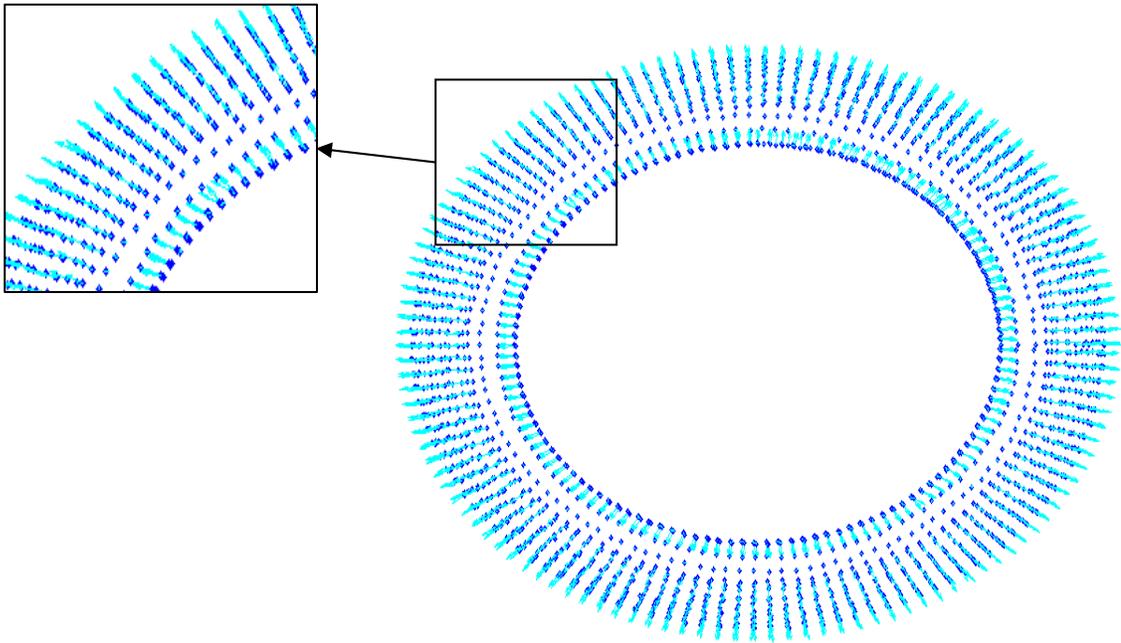


Figure 2.8: Normal vectors near sharp edge at boundary between cylindrical and planar surface in the synthetic cloud after (a) first and (b) second iterations.



(a) Normal vectors shown before orientation and re-estimation.



(b) Normal vectors shown after orientation and re-estimation.

Figure 2.9: Cross-section of synthetic cloud with normal vectors shown.

2.9 Feature Point Detection

Sharp edges 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 is due to abrupt change of surface orientation. Feature point detection was implemented in this work by checking the normal vector of each point against those of its neighbors. If, on average, the orientation of the point differed by more than a prescribed angle threshold, α , from its neighbors, the point was flagged as a feature point. Figure 2.10 shows feature points detected at boundaries on the synthetic point cloud, and Figure 2.11 shows feature points detected on a section of the laboratory specimen point cloud. The feature points on the specimen include the bolt heads and the edges of the I-beams.

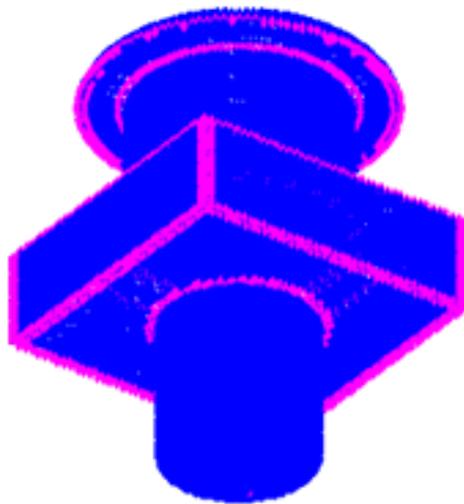
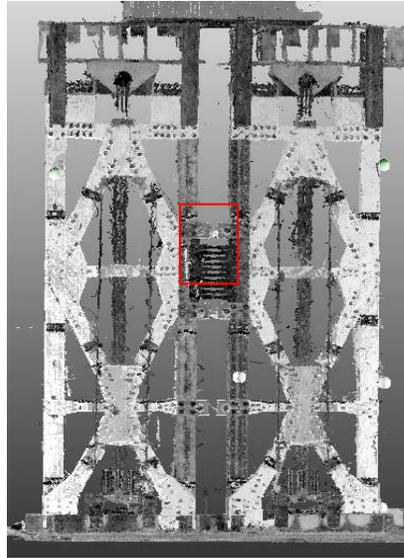


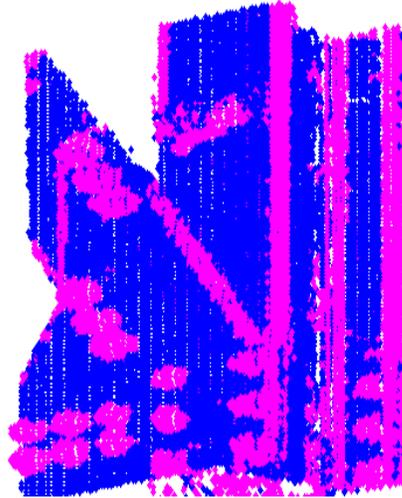
Figure 2.10: Feature points detected on the synthetic cloud.



(a) Point cloud



(b) Zoomed region of point



(c) Feature points detected

Figure 2.11: Feature points detected on a section of the laboratory specimen cloud.

2.10 Segmentation

Segmentation labels each point in a cloud so that points belonging to the same surface have the same label. Region growing is a method of segmentation that begins with individual points and collects neighboring points based on defined criteria to build up segments. There are several ways to define the criteria for including points. An algorithm based on smoothness (limited normal vector variation) was developed by Rabbani et al. (2006). Vanco and Brunnett incorporated a smoothness constraint as well as a parameter to control flatness, or overall curvature of a segment (Vanco and Brunnett, 2004). Another approach is to segment based only on estimated curvature values (Jiang, Zhang, and

Ming, 2005). After segmentation, it is possible to visualize and operate on the different surfaces in a scan, e.g., by color coding each point based on its label.

Region growing with a smoothness constraint was implemented in this work to achieve segmentation. It begins by selecting a seed point and then checking surrounding points to determine if they belong in the same segment. Whether a point should be added to the segment being grown is determined based upon a user defined smoothness criterion. The tangent plane normal of the point cannot vary beyond a specified angle threshold, α , from a neighbor point in the segment. Other criteria, such as requiring that the point is not a feature point or that the curvature is consistent within a defined threshold with the average curvature of the segment, could be included to modify the degree of segmentation.

Seed point selection is accomplished based on a strategy of segmenting planar regions first. The first seed point is chosen as the point in the cloud with the lowest curvature. If many points have the same low curvature value (approximately zero for a perfectly planar surface, for example) a random seed from that group is selected. As segmentation continues, a new seed point is always selected as the lowest curvature point remaining in the list of unsorted points.

Region growing for each segment is complete when no additional points among the neighbors meet the criteria required to belong to the segment. A new seed point is then selected and another segment is grown. Region growing for the cloud is complete when no points remain to be sorted. The results of region growing implemented with this algorithm on the synthetic cloud are shown in Figure 2.12; each color indicates an individual segment recognized by the algorithm.

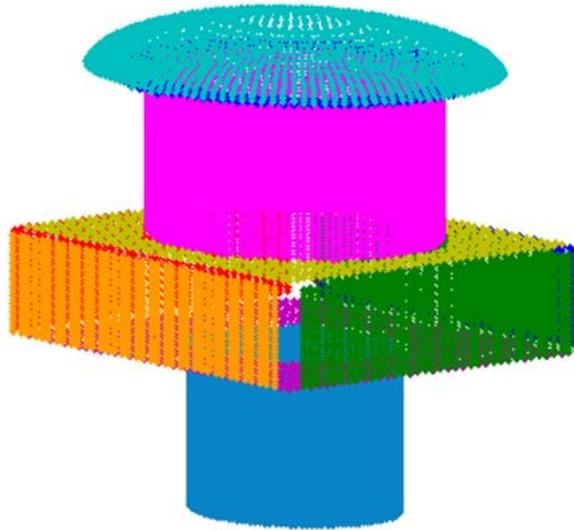


Figure 2.12: Segmentation results for the synthetic point cloud.

2.11 Surface Fitting

For each segment, surface fitting can be attempted to determine which type of surface is represented using a least squares approach (Dai and Newman, 1998). Simple surfaces (planes, cylinders, spheres, cones, etc.) can be represented by a quadric equation. The general equation for an arbitrary quadric surface in Cartesian space is as follows:

$$k_1 x^2 + k_2 y^2 + k_3 z^2 + k_4 xy + k_5 yz + k_6 zx + k_7 x + k_8 y + k_9 z = 1$$

The coefficients determine the surface type. For a set of segmented points representing a surface, the coefficients are found by solving for the least squares solutions of a group of m linear equations:

$$\mathbf{AX} = \mathbf{b}$$

where

$$A = \begin{bmatrix} x_0^2 & y_0^2 & z_0^2 & x_0 y_0 & y_0 z_0 & z_0 x_0 & x_0 & y_0 & z_0 \\ x_1^2 & y_1^2 & z_1^2 & x_1 y_1 & y_1 z_1 & z_1 x_1 & x_1 & y_1 & z_1 \\ \vdots & \vdots \\ x_m^2 & y_m^2 & z_m^2 & x_m y_m & y_m z_m & z_m x_m & x_m & y_m & z_m \end{bmatrix}$$

and

$$X = [k_1 \quad k_2 \quad k_3 \quad k_4 \quad k_5 \quad k_6 \quad k_7 \quad k_8 \quad k_9]'$$

\mathbf{b} is a vector of ones.

A sample of points from a segment is inserted into \mathbf{A} ; then \mathbf{X} is solved for as follows:

$$X = (A^T A)^{-1} A^T b$$

This approach was implemented for each segment identified during region growing of the synthetic cloud, as it consists of the primitive surface types for which this algorithm is ideal.

2.12 Object Detection

Once the point cloud has been transformed into a collection of surfaces, the next step is to determine how those surfaces combine to form objects. Identifying objects in a scene is a primary goal of laser data processing.

One way of defining an object is as a set of connected surfaces. An object definition using this approach is organized like a graph with the vertices representing surfaces and the edges representing proximity and relative orientation (Yi, 1998; Lee and Park, 2002). For example, a cube is a set of six planar surfaces, each adjoined at right angles along four sides to four other planar surfaces. Figure 2.13 shows visually the graph for its

object definition. Each node represents a planar surface, and the edges represent links between neighboring surfaces. Each edge would also contain the information of the relative orientation of the two surfaces it links.

Each region in the cloud being processed is checked against every other region to determine if they are neighbors. If two segments are identified as neighbors, the angle between their average normal vectors is computed and stored to be used to check potential model matches.

To seek objects in a given point cloud each fitted surface is checked against known model surfaces. If a surface found in the cloud matches a surface type belonging to a certain object model, that object is included in a list of potential object matches. With this type of object definition, the scale of the surfaces and the objects is not a factor. The approach simply seeks to match a type of surface. If a surface and its neighbors share a same potential object match, they are checked in combination against the relative orientations of the model. If a collection of surfaces in the cloud match the definition of a known model, the object is indicated as detected and the used surfaces are removed from future searches. This approach was implemented to identify an independent cube point cloud and to identify and extract a cuboid from the synthetic cloud. Figure 2.14 shows the surfaces of a cuboid detected in the synthetic point cloud, and Figure 2.15 shows the collection of object models created for this work. The application of this method to more complex objects is an area for future investigation.

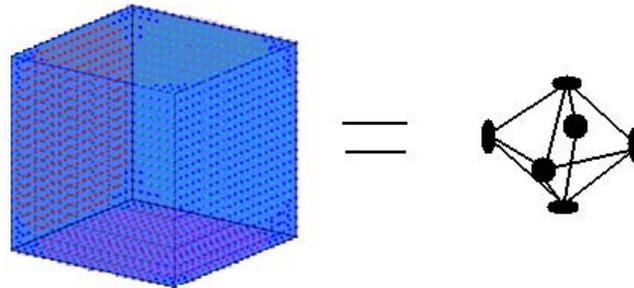


Figure 2.13: Example of object definition associated with a cuboid as a graph.

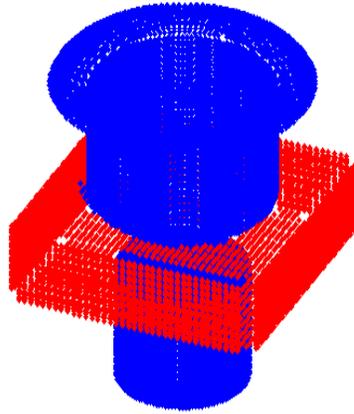


Figure 2.14: Synthetic point cloud with surfaces of a cuboid detected.

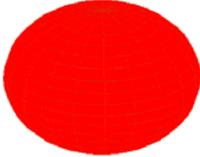
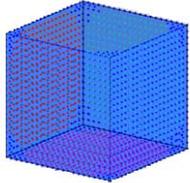
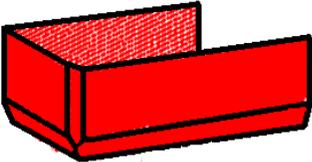
Sphere	
Cylinder	
Cuboid	
Beveled Cuboid	

Figure 2.15: Object model library.

DATA PROCESSING RESULTS FROM TESTBED BRIDGE EXAMPLE

To test the viability of the algorithms on a full-scale example in the field, scans were collected of a collapsed bridge. Comprehensive scans of actual scenes present many additional challenges beyond those encountered in a synthetic point cloud consisting of simple surfaces. As discussed in Chapter 2, actual point clouds contain noise, outlier points, and extraneous details. Also, actual scans usually consist of multiple scans registered together. Registration aligns the clouds, but there can be small errors. Even minor misalignments of two scans can lead to erroneous results during processing. Real scans contain irregular and unusual surfaces and objects. They can also include gaps and missing points in areas of the scene that were occluded during scanning. The purpose of analyzing the testbed bridge scan was to encounter some of these challenges and test the robustness of the algorithms.

3.1 Data Collection From Testbed Bridge

The bridge was located in a rural area 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.

Fourteen scans of the scene were captured with the Faro Photon 80 scanner from different points around the structure. Some of the scans were taken from street level while others were taken on the banks or from below the level of the deck. The variety of perspectives was needed to capture as much of the geometry of the structure as possible; each viewpoint exposed a unique set of surfaces of the bridge. Each scan captured approximately 41 million points, which resulted in a total of more than 500 million points for the registered point cloud. The scans were registered using Faro software. Figures 3.1 and 3.2 show photographs and scan data of the whole bridge and the piers. Appendix A contains further scanned images of the bridge.



(a) Photograph



(b) Scan data

Figure 3.1: Photograph and scan data of the testbed bridge.



(a) Photograph



(b) Scan data

Figure 3.2: Photograph and scan data of the standing piers and piercap of the testbed bridge.

3.2 Data Processing Results

While the complete registered point cloud data set for the bridge was very large, due to limitations on computational power it was most effective to execute the data processing algorithms on a portion of the point cloud at any one time. An area near one end of the standing pier cap was chosen as an example for this chapter, as indicated in Figure 3.3.

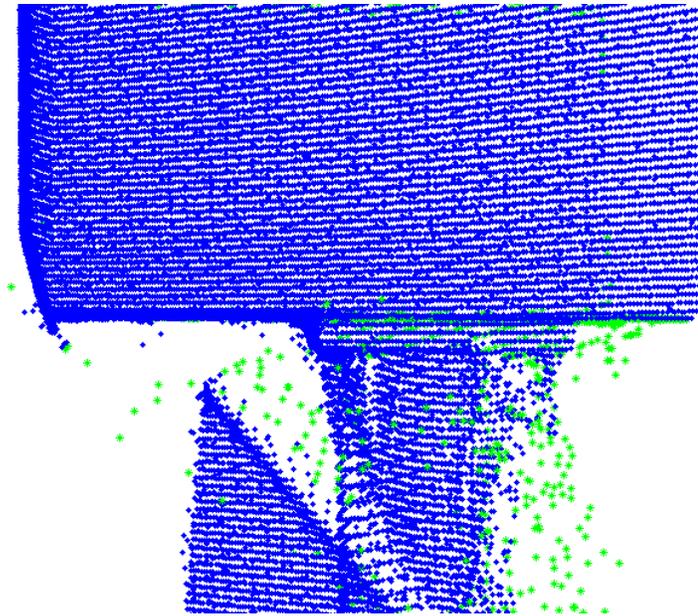
Parameter selection is critical to achieving good results, and a range of values were tested. The parameters found to produce the best results for this scan data were a neighborhood size, k , of 30 points and an angle threshold, α , of .15 radians.



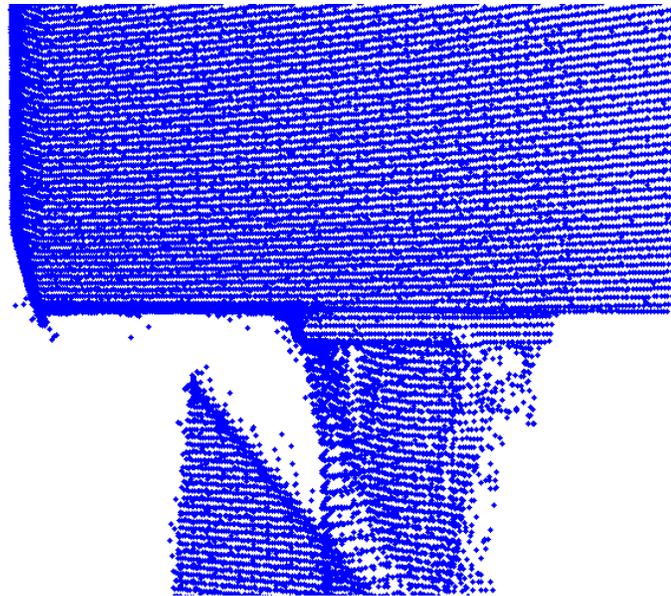
Figure 3.3: Scan data of testbed bridge with section utilized for data processing indicated.

During scanning some outlier points were captured, as is expected with any actual scan data. In the section of interest, a high number of outliers are located between the cross brace and the pier. As the laser reached the edge of the brace during scanning, the beam may have split, falsely recording points in space beyond the brace.

Figure 3.4a indicates the outlier points automatically detected by the algorithm from Section 2.4, and Figure 3.4b shows the cloud with those points removed.



(a) Data with outlier points indicated.



(b) Data with outlier points removed.

Figure 3.4: Section of testbed bridge point cloud with outlier points detected and removed.

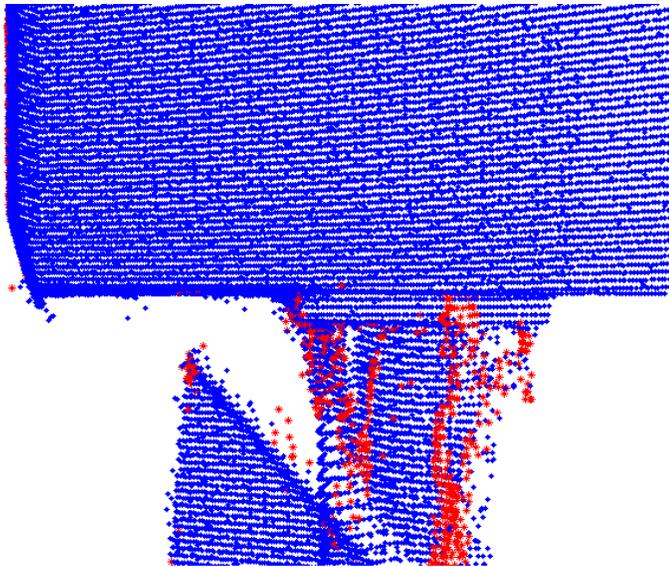
As discussed in Section 2.4, point clouds often contain extraneous details that make processing more difficult. The testbed bridge was scanned in-situ, and Figure 3.5 shows a scan that includes grass located between a pier and a brace possibly left by flooding.

Figure 3.6a indicates the points associated with the grass identified by the algorithm of Section 2.4. Figure 3.6b shows the cloud with those points removed.

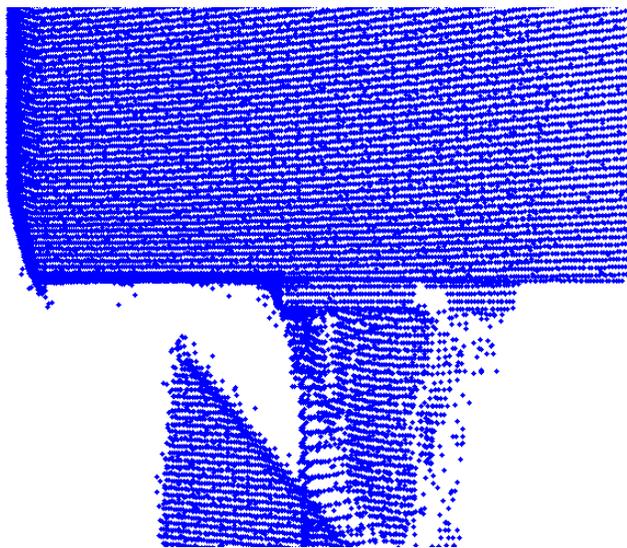


Figure 3.5: Section of testbed bridge point cloud showing grass located between a pier and a brace.

After removal of outlying and extraneous points, tangent planes were estimated and oriented automatically by the algorithms implemented as described in Sections 2.5 and 2.7 at each point in the cloud. Figures 3.7 shows tangent plane normal estimations for two example piers before and after orientation. Figure 3.8 shows oriented normals at all points.

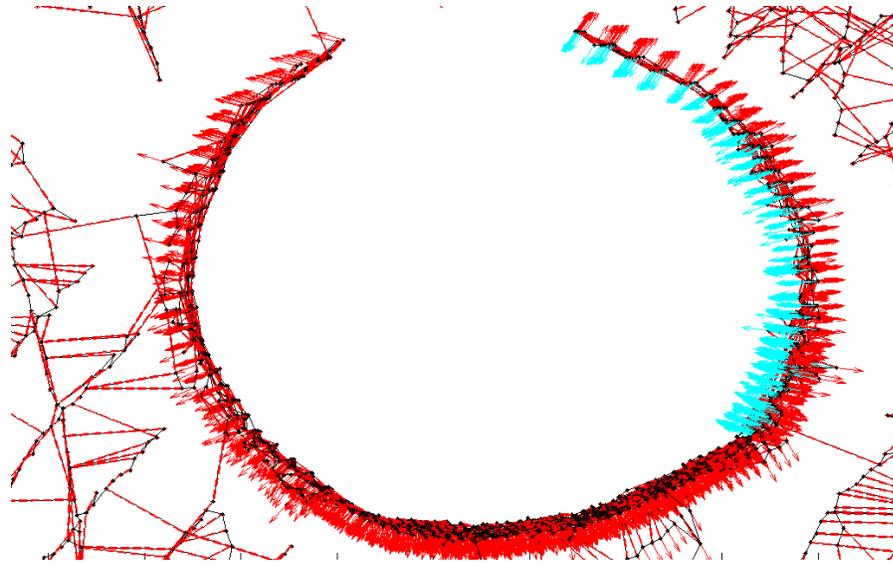


(a) Data with extraneous points indicated

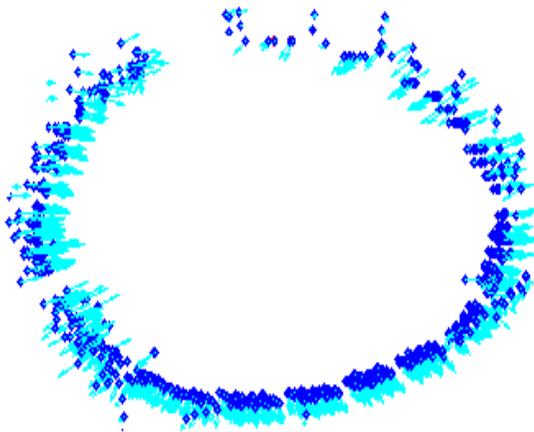


(b) Data with extraneous points removed

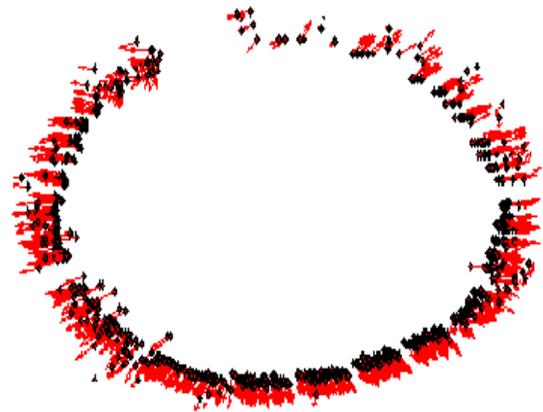
Figure 3.6: Section of testbed bridge point cloud with extraneous points indicated and removed.



(a) Example 1, normal vectors shown before (aqua) and after (red) orientation.

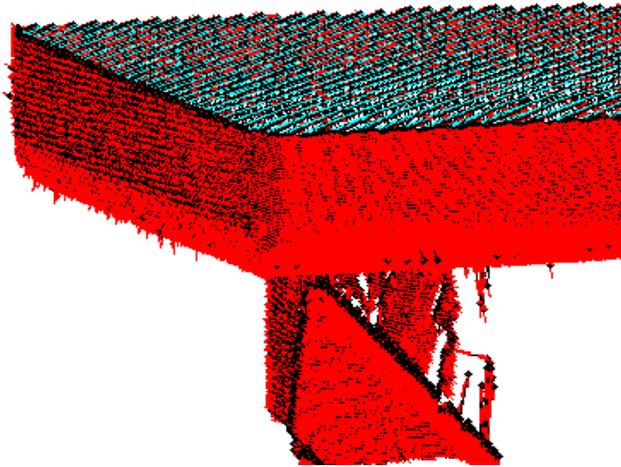


(b) Example 2 before orientation

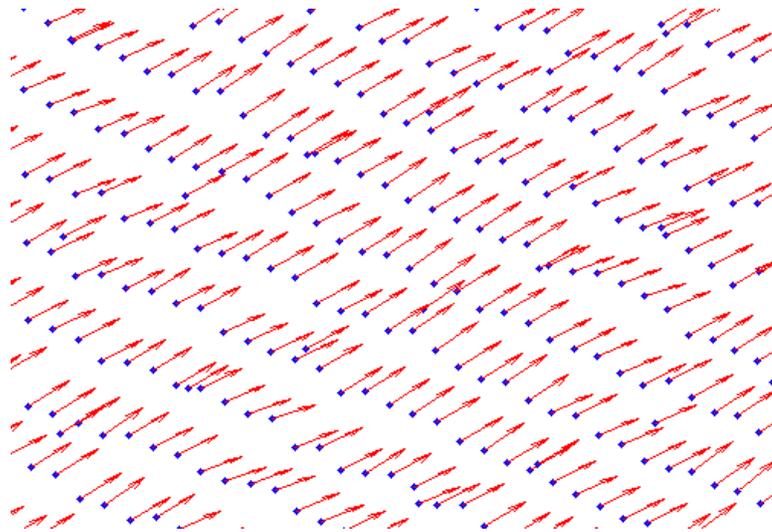


(c) Example 2 after orientation

Figure 3.7: Two examples of testbed bridge pier cross-sections with normal vectors shown.



(a) Normal vectors for the testbed bridge data.



(b) A zoomed region of the testbed bridge data with normal vectors shown.

Figure 3.8: Testbed bridge data with normal vectors shown.

Once normal vectors for each point were estimated and properly oriented, feature points were identified by checking normal vector variation from point to point as discussed in Section 2.9. Detected feature points for this portion of the testbed bridge cloud are indicated in Figure 3.9. Feature points can reveal contours in scan data that may not be apparent when viewing it prior to processing. For example, the pier cap does not have a simple rectangular cross-section; the feature points reveal its beveled edges.

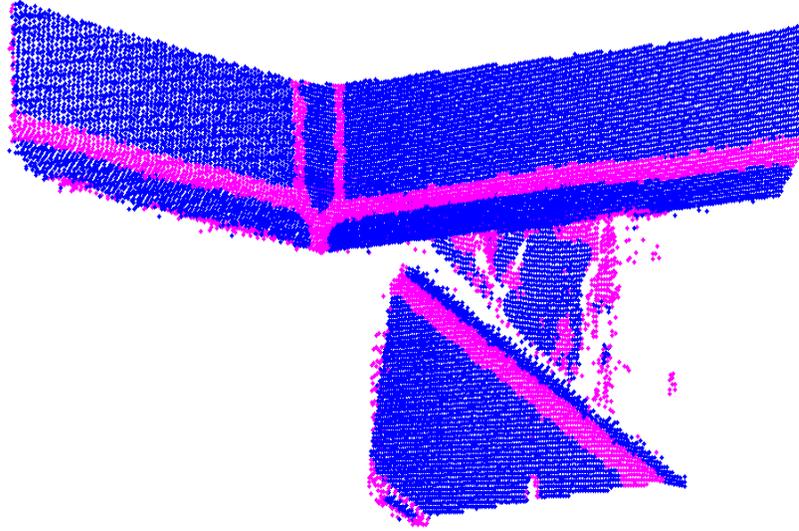


Figure 3.9: Portion of testbed bridge scan data with feature points shown.

Next, segmentation was accomplished utilizing the region growing from Section 2.10. The results are shown in Figure 3.10. Each color indicates points identified as belonging to an individual surface. Each region has properties associated with it including number of points, average curvature, and average normal orientation.

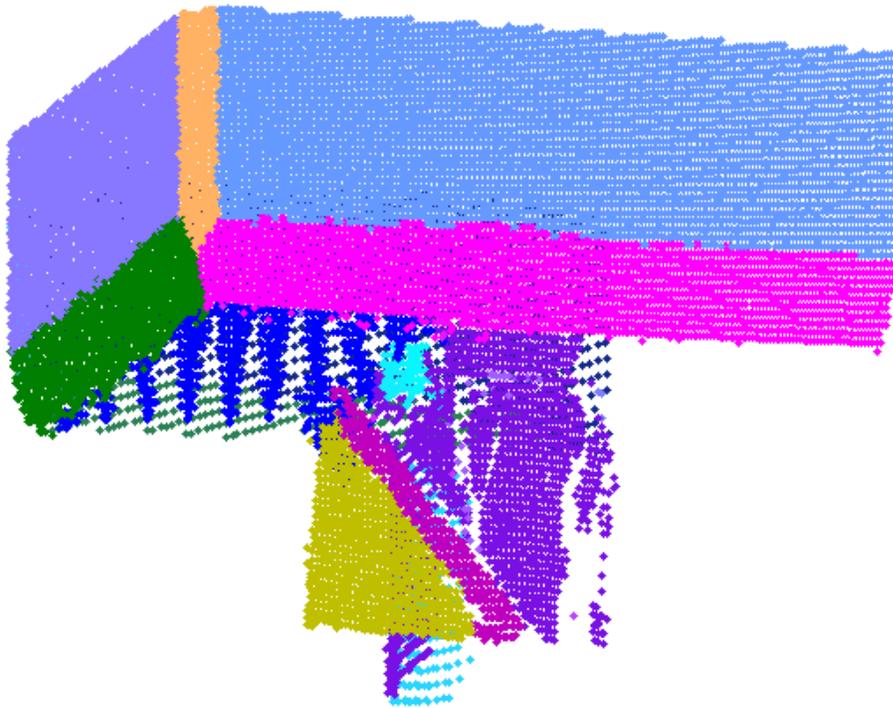
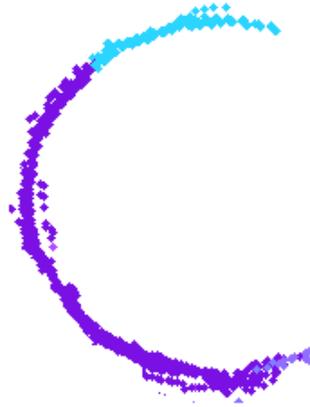
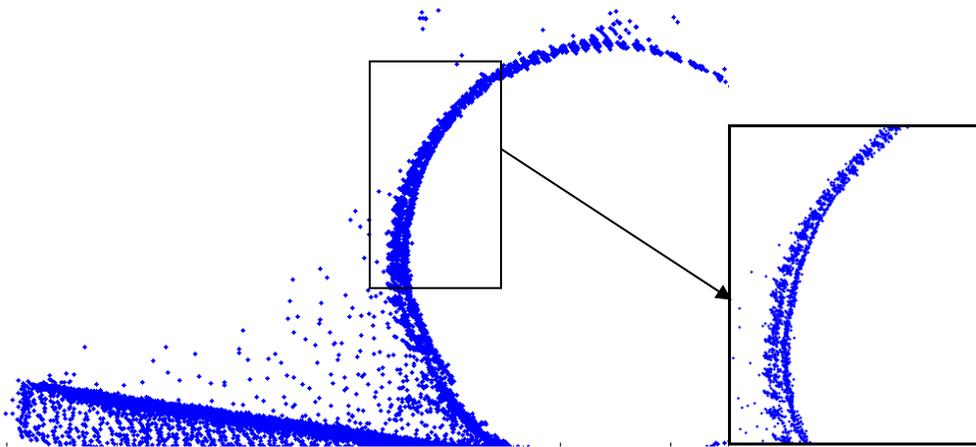


Figure 3.10: Segmentation results of portion of testbed bridge scan data.

Some over-segmentation of the pier occurred. This was due to a minor offset resulting from registration. Figure 3.11 shows a cross-section of the pier with the points from different scans distinguished by different colors. The region with the offset is indicated.



(a) Cross-section view of over-segmentation of pier



(b) Cross-section view of offset in scan data

Figure 3.11: Over-segmentation of pier due to registration offset.

To correct for over-segmentation, regions that belong to the same surface should be merged by the user. This procedure could possibly be automated, but interactive intervention is sometimes necessary to interpret where processing has led to incorrect results, particularly in the case when the error is due to an inconsistency with the point cloud.

Surface fitting was implemented on the pier segment. A sampling of points was taken from the segment, and the least squares solution for the quadric equation was found as described in Section 2.11. In this point cloud, the object of a pier is defined by a single surface. Therefore, the object definition can be the surface equation. Point clouds of

structures may often contain components that are repeated several times. In this case, there are eight identical bridge piers. A different pier, shown after segmentation in Figure 3.12, has a nearly identical equation after surface fitting. In this case, where the piers are a distinct surface type in the cloud, an equation matching check can be utilized as object detection.

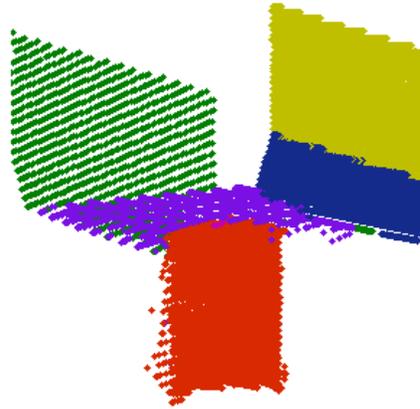
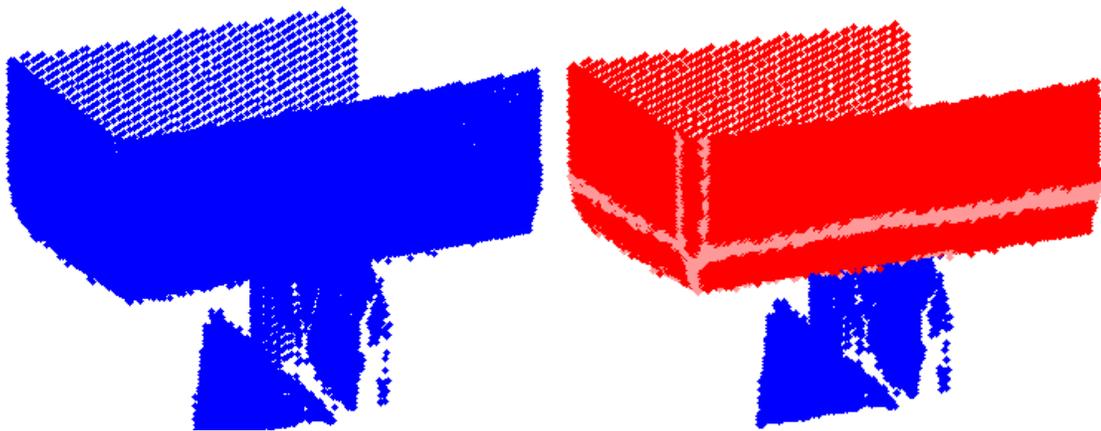


Figure 3.12: Second bridge pier after segmentation.

In addition, an object detection check for the pier cap was developed. First, a single surface match is searched for in the collection of surfaces detected in the cloud. If one surface match is found, neighboring surfaces in the cloud are checked to determine if they meet the criteria for also belonging to the object. Surface type and relative orientation are taken into account to determine if a match exists. In this way, as discussed in Section 2.12, objects can be thought of as graphs with surfaces represented by nodes and relative orientations and proximity represented by edges. Figure 3.13 shows the region of the testbed bridge cloud with the surfaces detected as belonging to the pier cap highlighted.



(a) Testbed bridge point cloud

(b) Detected surfaces of pier cap

Figure 3.13: Testbed bridge point cloud with surfaces detected as belonging to the pier cap indicated.

CONCLUSIONS

4.1 Future Work

The work discussed in the previous sections represents progress towards an end goal of utilizing laser scan data for investigation and analysis of structures. Potential next steps to be taken to continue the evolution of this task are discussed below.

Complex and Free-Form Surface Definitions

Section 2.11 discussed the fitting of simple surfaces to point cloud data. While useful for defining many manmade objects and structural components, these types of surfaces are still limited in their application to complex scenes. To enable processing of a variety of scenes, it will be necessary to expand surface definition capabilities. One way to accomplish this would be to develop splines to describe more complex and free form surface types (Wang, Pottman, and Liu, 2006).

Meshing for Analysis

Surface meshing is a valuable tool for visualization, and to utilize an object identified in a point cloud as part of a structural analysis, a volume mesh is also needed. Surface meshing of data sets has been explored in prior relevant research (Dey and Levine, 2008; Wang et al., 2008). However, there has been less work done in the area of volume meshing of point clouds.

Building Information Models (BIM)

Building Information Models (BIMs) are 3D models of structural components with additional associated object information. The 3D, object-oriented nature of BIM suggests a potential link with 3D laser scanning. As BIM becomes more widely utilized in structural engineering design, a processing tool for incorporating scan data into BIM, or vice versa, could be relevant and useful.

4.2 Conclusions

This work represents the implementation of initial steps towards processing of structural laser scan data. The motivation for development of such processing tools stemmed from extensive literature review that revealed the disparity between the advance of laser scanning technology and the non-existence of advanced data processing software. Algorithms from existing literature including tangent plane estimation, feature point detection, and segmentation were implemented. Steps (e.g. extraneous point removal) and modifications were added to increase the robustness of the processing for actual scan data captured in the laboratory or field. The results of processing on actual scan data captured of a collapsed bridge in situ were shown to illustrate the effectiveness of the

algorithms. In structural engineering, with appropriate processing tools, laser scanning has the potential to play a role in tasks like structural health monitoring and collapse assessment. As indicated earlier in this chapter, there is still much to be done towards the development of a comprehensive set of data processing tools. However, a foundation has been laid upon which future work can build.

REFERENCES

- Arayici, Y. (2007), An approach for real world data modelling with the 3D terrestrial laser scanner for built environment. *Automation in Construction*, 816-829.
- Barnea, S. and Filin, S. (2008), Keypoint based autonomous registration of terrestrial laser point-clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63, 19-35.
- Bénko, P., Martin, R., and Varady, T. (2001), Algorithms for reverse engineering boundary representation models. *Computer-Aided Design*, 33, 839-851.
- Besl, P. and McKay, N. (1992), A Method for Registration of 3-D Shapes, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14 , 239-257.
- Bosche, F. and Haas, C. (2008), Automated retrieval of 3D CAD model objects in construction range images, *Automation in Construction*, 499-512.
- Calderon, F., Ruiz, U., and Rivera, M. (2007), Surface-Normal Estimation with Neighborhood Reorganization for 3D Reconstruction. Image Analysis and Applications - 12th Iberoamerican Congress on Pattern Recognition, Vina del Mar-Valparaiso: Springer Verlag, pp. 321-330.
- Dai, M., and Newman, T. (1998), Hyperbolic and Parabolic Quadric Surface Fitting Algorithms, *UAH CS Technical Report*.
- Daniels, J., Ha, L., Ochotta, T., and Silva, C. (2007), Robust Smooth Feature Extraction from Point Clouds, *IEEE International Conference on Shape Modeling and Applications*.
- 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, 276-283.
- Dey, T. and Levine, J. (2008), Delaunay meshing of isosurfaces, *Visual Computer*, 24, 411-422.
- Gelfand, N., Mitra, N., Guibas, L., and Pottman, H. (2005), Robust Global Registration, *Eurographics Symposium on Geometry Processing*.
- Gordon, S. and Lichti, D. (2007), Modeling Terrestrial Laser Scanner Data for Precise Structural Deformation Measurement, *Journal of Surveying Engineering*, 72-80.
- Hawarey, M. and Falk, M. (2004), Using Laser Scanning Technology to Measure Deflections in Steel Columns, *Iron and Steel Technology*, 40-43.
- Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W. (1992), Surface Reconstruction from Unorganized Points, *ACM SIGGRAPH 1992 Conference Proceedings*, 71-78.
- Huang, J. and Menq, C. (2002), Automatic CAD Model Reconstruction from Multiple Point Clouds for Reverse Engineering, *Transactions of the ASME*, 160-170.

- Lee, Y., and Park, R. (2002), A surface-based approach to 3-D object recognition using a mean field annealing neural network, *Pattern Recognition*, 35, 299-316.
- Nüchter, A., and Hertzberg, J. (2008), Towards semantic maps for mobile robots, *Robotics and Autonomous Systems*, 56 , 915-926.
- Park, H., Lee, H., Adeli, H., and Lee, I. (2007), A New Approach for Health Monitoring of Structures: Terrestrial Laser Scanning, *Computer-Aided Civil and Infrastructure Engineering*, 22, 19-30.
- Pauly, M., Gross, M., and Kobbelt, L. (2002), Efficient Simplification of Point-Sampled Surfaces, *IEEE Visualization '02*.
- Pauly, M., Keiser, R., and Gross, M. (2003), Multi-scale Feature Extraction on Point-Sampled Surfaces, *Computer Graphics Forum*, 22(3), 281-289.
- Rabbani, T., van den Heuvel, G., and Vosselman, G. (2006), Segmentation of Point Clouds Using Smoothness Constraint, *ISPRS Commission V Symposium 'Image Engineering and Vision Metrology'*, 248-253.
- Rusu, R., Marton, Z., Blodow, N., Dolha, M., and Beetz, M. (2008), Towards 3D Point Cloud Based Object Maps for Household Environments, *Robotics and Autonomous Systems*, 56, 927-941.
- Vanco, M. and Brunnett, G. (2007), Geometric preprocessing of noisy point sets: an experimental study, *Computing*, 79, 365-380.
- Wang, J., Oliveira, M., Zhang, H., and Kaufman, A. (2008), Reconstructing regular meshes from points, *Visual Computer*, 24, 361-371.
- Wang, W., Pottmann, H., and Liu, Y. (2006), Fitting B-Spline Curves to Point Clouds by Curvature-Based Squared Distance Minimization, *ACM Transactions on Graphics*, 25, 214-238.
- Yi, J. H. (1998), Model-Based 3D Object Recognition Using Bayesian Indexing, *Computer Vision and Image Understanding*, 69, 87-105.
- Yu, Y., Ferencz, A., and Malik, J. (2001), Extracting Objects from Range and Radiance Images, *IEEE Transactions on Visualization and Computer Graphics*, 7, 351-363.

APPENDIX A: ADDITIONAL SCANNED IMAGES OF TESTBED BRIDGE

This appendix contains additional scans of the testbed bridge investigated in this work. Each scan with the Faro Photon 80 laser captured approximately 41 million points.



(a)



(b)



(c)



(d)



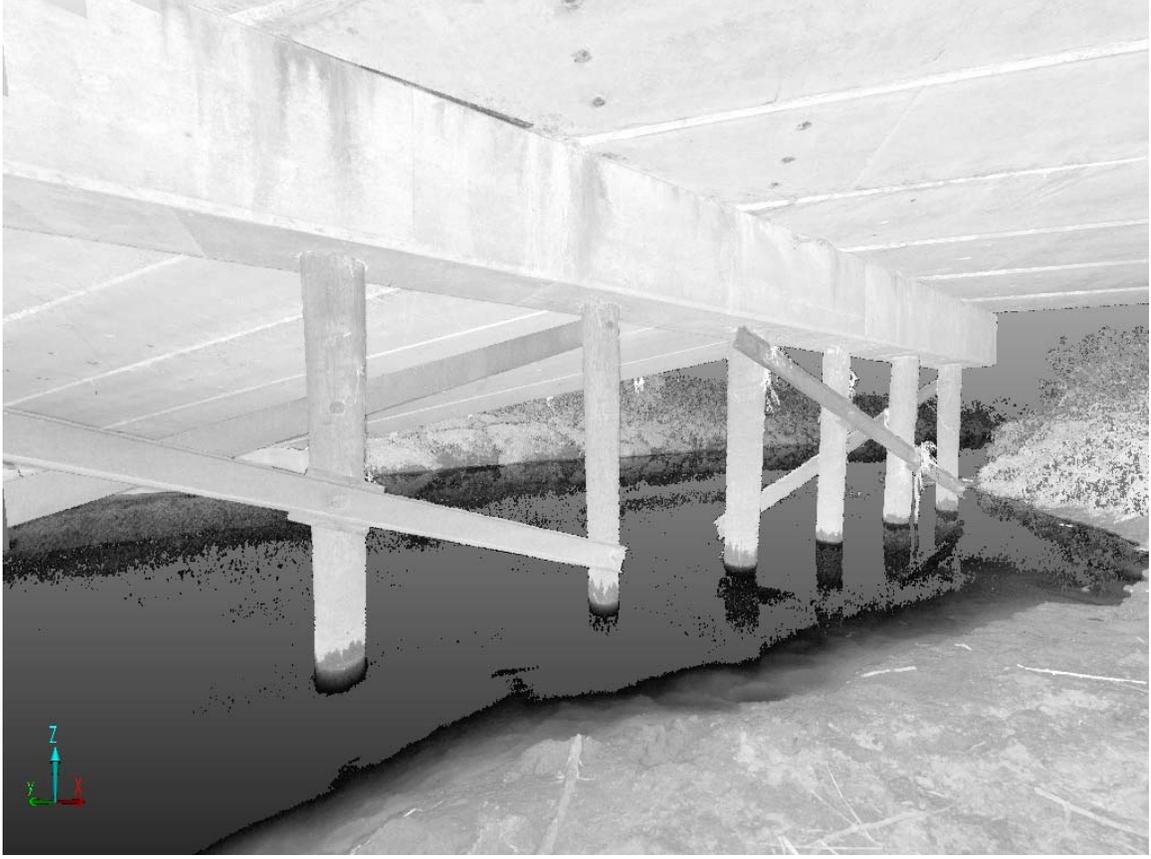
(e)



(f)



(g)



(h)

Figure A1: Laser scans of testbed bridge

List of Recent NSEL Reports

<i>No.</i>	<i>Authors</i>	<i>Title</i>	<i>Date</i>
001	Nagayama, T. and Spencer, B.F.	Structural Health Monitoring Using Smart Sensors	Nov. 2007
002	Sun, S. and Kuchma, D.A.	Shear Behavior and Capacity of Large-Scale Prestressed High-Strength Concrete Bulb-Tee Girders	Nov. 2007
003	Nagle, T.J. and Kuchma, D.A.	Nontraditional Limitations on the Shear Capacity of Prestressed, Concrete Girders	Dec. 2007
004	Kwon, O-S. and Elnashai, A.S.	Probabilistic Seismic Assessment of Structure, Foundation, and Soil Interacting Systems	Dec. 2007
005	Nakata, N., Spencer, B.F., and Elnashai, A.S.	Multi-dimensional Mixed-mode Hybrid Simulation: Control and Applications	Dec. 2007
006	Carrion, J. and Spencer, B.F.	Model-based Strategies for Real-time Hybrid Testing	Dec. 2007
007	Kim, Y.S., Spencer, B.F., and Elnashai, A.S.	Seismic Loss Assessment and Mitigation for Critical Urban Infrastructure Systems	Jan. 2008
008	Gourley, B.C., Tort, C., Denavit, M.D., Schiller, P.H., and Hajjar, J.F.	A Synopsis of Studies of the Monotonic and Cyclic Behavior of Concrete-Filled Steel Tube Members, Connections, and Frames	April 2008
009	Xu, D. and Hjelmstad, K.D.	A New Node-to-node Approach to Contact/Impact Problems for Two Dimensional Elastic Solids Subject to Finite Deformation	May 2008
010	Zhu, J. and Popovics, J.S.	Non-contact NDT of Concrete Structures Using Air Coupled Sensors	May 2008
011	Gao, Y. and Spencer, B.F.	Structural Health Monitoring Strategies for Smart Sensor Networks	May 2008
012	Andrews, B., Fahnestock, L.A. and Song, J.	Performance-based Engineering Framework and Ductility Capacity Models for Buckling-Restrained Braces	July 2008
013	Pallarés, L. and Hajjar, J.F.	Headed Steel Stud Anchors in Composite Structures: Part I – Shear	April 2009
014	Pallarés, L. and Hajjar, J.F.	Headed Steel Stud Anchors in Composite Structures: Part II – Tension and Interaction	April 2009
015	Walsh, S. and Hajjar, J.F.	Data Processing of Laser Scans Towards Applications in Structural Engineering	June 2009