

# **Semi-Automated As-Built Modeling of Light Rail System Guide Beams**

Pingbo Tang<sup>1</sup>, Burcu Akinci<sup>1</sup>, Daniel Huber<sup>2</sup>

<sup>1</sup>Civil and Environmental Engineering Department, Carnegie Mellon University

<sup>2</sup>Robotics Institute, Carnegie Mellon University

## **Abstract**

For a light rail system, smooth contact between the vehicle and the guide beam is critical for reducing the friction and the vibration of an operating vehicle. Therefore, the shape of guide beams needs to be controlled with mm-level accuracy during the construction. Currently, most methods for detecting shape defects of guide beams, such as experimental run of vehicles, are costly and tedious. In addition, these methods can only identify defects after the completion of the construction, and cause reworks and delays of defects fixings.

From dense point clouds collected by laser scanners, inspectors can manually extract geometric features and conduct virtual inspections of guide beams. However, the manual geometric feature extraction process impedes effective utilization of point clouds for the shape analysis of guide beams. Aiming at improving the efficiency of utilizing laser scanning technology for guide beam quality control, this research developed a semi-automatic approach for simultaneously extracting the axis parameters (e.g., radius) and cross-section features (e.g., width) of a guide beam using a Hough-Transform based approach, and discusses factors (e.g., data density) influencing the performance of this approach.

**Keywords:** Laser Scanning; Geometric Feature Extraction; Information Retrieval; Quality Control; Surveying; 3D Modeling;

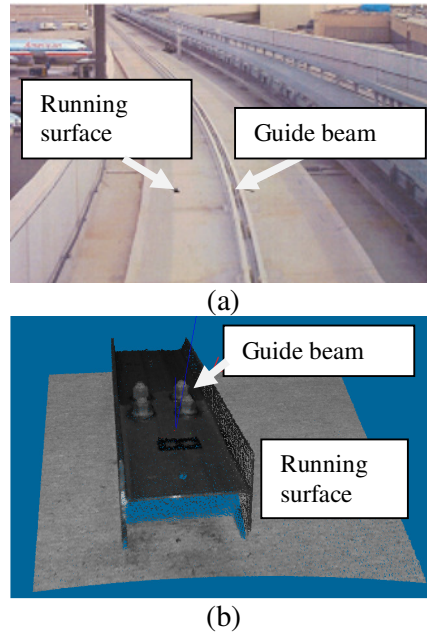
## **Introduction**

The precise installation of a guide beam for a light rail system (Figure 1a) is critical for both the riding quality and the maintenance costs. Smooth contact between the guide wheels of a vehicle and the guide beam are important for reducing the vehicle frictions and hence reducing the vibration of an operating vehicle and the maintenance costs of the vehicle and the guide beam. Therefore, the quality control of the shape of a guide beam is important for the performance of a light rail system and the cash flow of the project. Currently, two approaches for detecting shape defects of guide beams are mainly used: 1) Visual inspection; 2) Experimental run of vehicles. The former requires inspectors to walk along the track and manually identify shape defects (e.g., distortions of the flanges). This method requires highly-skilled inspectors to make decisions based on their experiences, and is subjective and time-consuming. Experimental runs of vehicles expose a constructed light rail system to real working conditions. During experimental runs, engineers can observe the vibrations of the vehicle to identify sections of problematic guide beams. Such real-scale experiments could be expensive. Moreover, both the visual

inspection and the experimental run methods are conducted after the construction, when it might be too late or too expensive to fix those detected defects.

Laser scanning is a promising method for capturing geometric details of guide beams during the construction. A laser scanner can densely sample the surfaces of a guide beam in minutes (1 cm step size within 50 m for the scanner used in this research), and generate dense point clouds (Figure 1b). With such dense data, inspectors can reconstruct 3D as-built models of guide beams using commercial 3D modeling tools, and manually conduct virtual inspections on these models. One major issue is that this manual approach requires inspectors to manually extract axis (e.g., radius) and cross section features (e.g., width) from point clouds. This becomes a tedious approach considering the fact that guide beams can run for miles resulting in a very large amount of 3D data.

For improving the efficiency of utilizing laser scanners for more effective quality control of guide beams, this paper presents a semi-automated approach for simultaneously extracting the axis and cross-section features of an I-beam.



**Figure 1 (a) A light rail system and (b) laser-scanned point clouds of a guide beam**

## Related Studies

Previous research studies have investigated various methods for extracting geometric features and object poses from 3D data. Kwon *et al.* developed algorithms for extracting parameters of cuboids and cylinders from flash LADAR scanned point clouds (Kwon *et al.* 2004). Vosselman *et al.* showed the effectiveness of a set of Hough transform-Based algorithms for extracting planes, cylinders and spheres from laser-scanned data (Vosselman *et al.* 2004). Based on Gauss-Newton least square fitting method, Barker developed a MATLAB library for extracting 2D and 3D lines, circles, planes, cylinders, cones and spheres (Barker 2004). Using the Generalized Gauss-Markoff model, Gruen and Akca proposed a framework for extracting arbitrarily posed parametric surfaces by comparing data patches against a library of predefined surface templates and determining the most probable surface model for a given data patch (Gruen and Akca 2005). Gilsinn *et al.* used Principal Component Analysis (PCA) to estimate the pose of an I-beam (Gilsinn *et al.* 2005). Results of these research studies show the feasibility of automatic extraction the geometric features of light rail guide beams from dense point clouds.

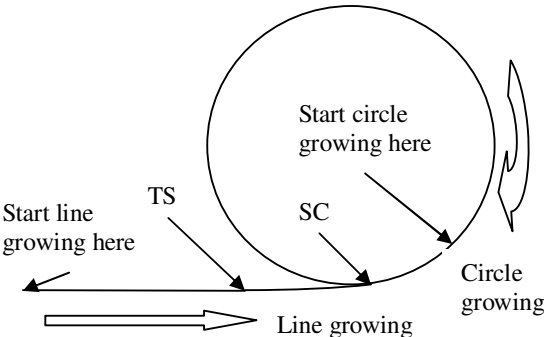
One major difficulty of applying these previously developed geometric feature extraction algorithms for automatic extraction of guide beam features is that most algorithms can only handle simple 3D scenes (e.g., horizontal beam on flat ground in an open space). Many construction sites have more complicated geometric

conditions (e.g., moving objects, and occlusions) than those in these simplified scenarios. To overcome this issue, we decided to develop a semi-automated approach, which requires inspectors to manually localize a number of axis points of the guide beam, and then the algorithm can refine such rough axis information to obtain accurate geometric features of the guide beam.

# Guide Beam Features

Generally, a guide beam has two categories of geometric features to be inspected for their shape quality control. The first category of features includes the parameters of the axis curve (axis curve features), while the second category of features are the features of the guide beam’s cross sections (cross section features).

For most light rail systems, to ensure the smooth turning of the vehicle, the curvature variation should be continuous: from straight line (infinite radius), to a spiral curve (a curve the radius of which continuously changing from infinity to a specific radius value), and to a circle with the specific radius value at the end of the spiral curve (Figure 2). Such curve shape has continuous curvature changes. In this research, we call the boundary point between the straight line and the spiral curve a



**Figure 1** Line and circle fitting and growing for finding TS and SC points

“TS” point, and the boundary point between the spiral and the circular sections a “SC” point. This research focused on recognizing the parameter of the straight line, the spiral curve length between “TS” and “SC”, and the radius of the circular section, because these parameters are controlled parameters having explicit as-designed values to be compared with the ones in the design drawings.

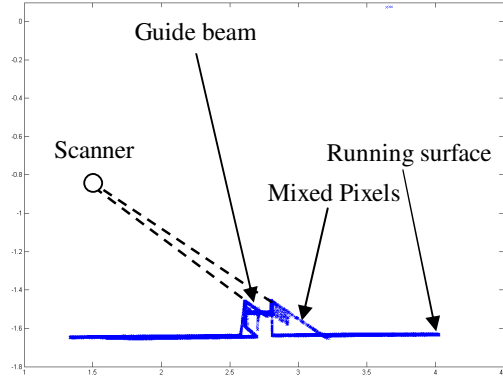
For such a cross-section of a guide beam, the critical features to be recognized are the web and two flanges, which could be simplified as three lines on the cross-section. With these three lines, it is possible to derive two geometric properties of the cross section: the width of the guide beam, and the angle between the flange and the web. This research focused on obtaining the width of the guide beam, but did not look into angles between web and flanges, since the former is usually a controlled property of the guide beam according to several interviews with domain experts. Table 1 lists the parameters focused on in this paper.

**Table 1** Targeted features of a guide beam

Feature Category	Feature Details (Symbol for Representing It)
Axis Feature	<ul style="list-style-type: none"> <li>Line parameters of the straight section</li> <li>Spiral curve length (L): curve length between TS and SC</li> <li>Circular section radius (R)</li> </ul>
Cross-section Feature	<ul style="list-style-type: none"> <li>Flanges (F1 and F2) and Web (W)</li> <li>Intersection points of W and F1/F2 (P1, P2)</li> <li>Width (w): distance between P1 and P2</li> </ul>

## Overview of the Guide Beam Feature Extraction Approach

In most cases, guide beam axes are straight, or curves with small curvature (with radius larger than 100 m), so that the axis of a small guide beam segment (e.g., 1 m in length) can be approximated by a straight line linking the two ends of the segment. That line is called the “approximate axis” of a segment in this paper.



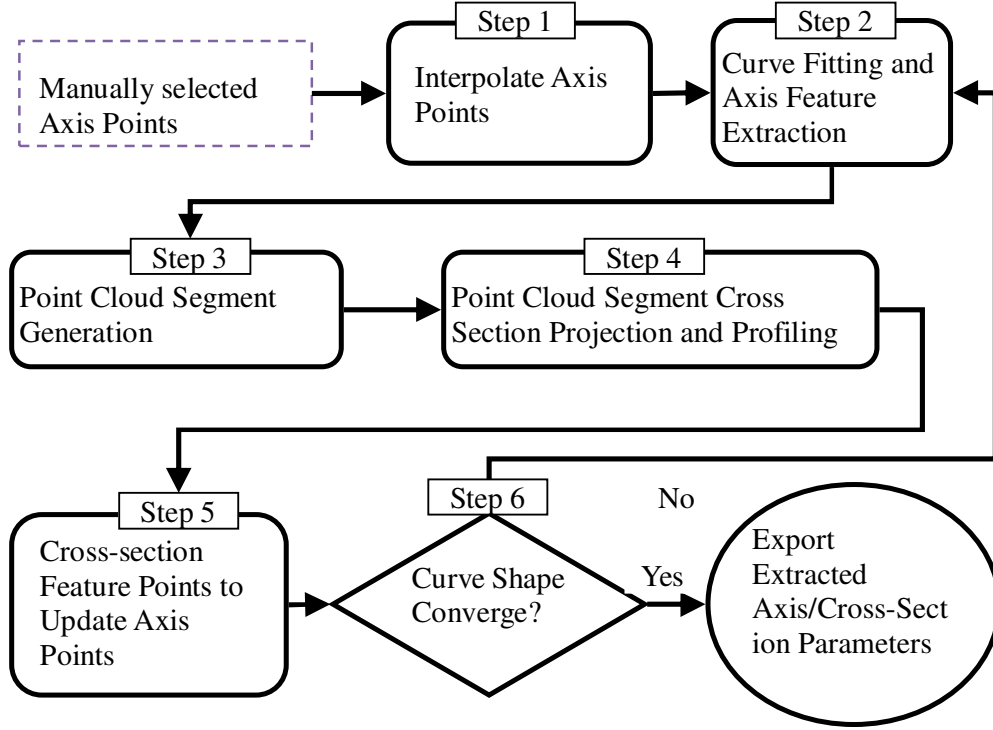
**Figure 2** A segment of guide beam point cloud projected along the beam axis to form a cross section profile

For point clouds of a straight or approximately straight beam segment, projecting all points along its approximate axis results in 2D points profiling the cross section of the beam (Figure 3). The diagonal lines shown in this figure are formed by “mixed pixels” (Tang et al. 2007),

which are data artifacts caused by laser spots spanning over spatial discontinuities (object edges). Figure 3 shows the location of the scanner to indicate that these diagonal lines are along the viewing directions of the scanner for guide beam edges.

In this research, we found that if a user can manually select a number of points roughly on the axis of a guide beam, these manually selected points were accurate enough to generate approximate axes of guide beam segments for generating their cross-section profiles. This observation motivates the development of an approach, which requires users to pick the rough locations of a number of guide beam axis points from point clouds, and then the algorithm uses these inputs to extract and refine cross-section and axis features of the guide beam (Figure 4).

Given the manually selected axis points, the semi-automated approach for guide beam feature extraction consists of six major steps: 1) Interpolating the manually selected axis points with a user-defined interpolation step size; 2) Identifying the locations of TS and SC points by fitting and growing straight lines and circular curves based on the interpolated axis points; 3) Generating a segment of point clouds for each line segment that link each pair of neighboring interpolated axis points; 4) Projecting points along its approximate axis and extracting cross-section features (lines representing the flanges and the web, and the intersection points between them) for each point cloud segment; 5) Using the extracted intersection points as refined axis points; 6) Using the updated axis points as input and starting the whole process to generate another set of cross section feature points along the axis. Once the updated points have no substantial deviations from the results of the last iteration, then the algorithm concludes that the curve updating process converged, and reports the latest results of curve fitting and cross section feature extraction.



**Figure 4 Overview of the developed approach for guide beam feature extraction**

## Detailed Feature Extraction Process

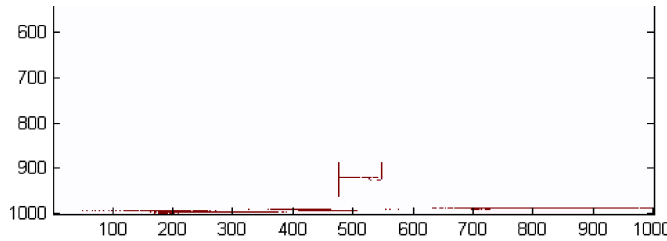
In Step 1, the algorithm utilizes the spline interpolation method to generate interpolated points along the x-axis (which we define as the direction of the straight section of the guide beam, toward the spiral section) with a fixed step size of 1cm. In other words, using the manually picked axis points, the algorithm generates a sequence of x values with 1 cm intervals, and interpolates corresponding y values.

In Step 2, the algorithm fits and grows straight lines and circles on the interpolated points for identifying TS and SC points through the following semi-automated steps. This step requires an inspector to visually inspect all interpolated points, and manually identify a point on the straight section and a point on the circular section. Then, starting from the manually picked straight section point and circular section point, the algorithm iteratively fits and grows a straight line (starting from the straight line point) or a circle (starting from the circular section point) toward the spiral section. These fitting processes use least-squares fitting. While growing straight/circular sections, the algorithm uses an “average deviation” metric to measure the fitting quality. Average deviation is defined as the sum of the deviation values of all axis points from the fitting line/circle divided by the total number of points. The algorithm will check through all possible TS and SC point locations, identify the locations achieving the minimum average deviation, and labels them as TS and SC. The concept is that since the two sections bounding a spiral section are known to be a straight line and a circular section, it would be possible to find the TS and SC points by growing two parametric models and finding the best fit (Figure 2). This step generates the parameters of the linear (line parameters) and the circular sections (radius), and the TS and SC points for getting

the length of the spiral section.

In Step 3, the algorithm splits the point clouds of a guide beam into segments of point clouds. With a sequence of interpolated axis points, the algorithm generates lines linking pairs of neighboring axis points. These line segments are linear pieces approximating the shape of the axis curve. Then, for each line segment, the algorithm identifies the x coordinates of the points in the point cloud that are falling into the x value range formed by the x coordinates of the end points of the line segment. Points for one line segment forms a slice of a point cloud capturing the shape of a beam segment.

In Step 4, for each point cloud segment, the algorithm processes the data in the following three sub-steps: 1) Projecting all the points along the direction of the corresponding line segment; 2) Generating a binary cross-section occupancy image depicting the cross-section profile of the guide beam (Figure 5); 3) Applying a series of Hough transforms to the binary cross-section profile to detect lines in it. In



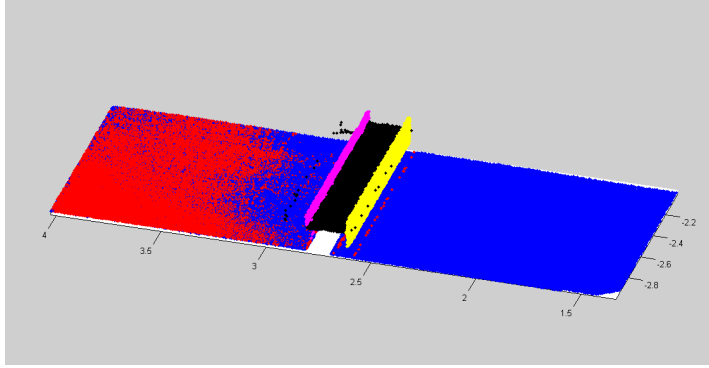
**Figure 3 A binary cross-section occupancy image**

sub-step 2, noticing that the cross section profile parts generated from data artifacts (e.g., mixed pixels that form a sloping line in Figure 3) have lower data densities, the algorithm generates 2D rectangular cells with fixed 2D

dimensions on the cross section plane, computes the number of points falling into each cell, and only label cells with more than a fixed number of points as “occupied” in order to ignore the lines formed by data artifacts. Based on large number of experiments on the data sets used in this research, the implementation in this research uses 3 mm by 3 mm as the cell size, and 20 points as the occupancy threshold, because these values were found to be able to distinguish lines formed by data points falling on actual object surfaces and diagonal lines formed by mixed pixels. The binary occupancy image generated in this sub-step serves as the input of sub-step 3.

In sub-step 3 of Step 4, the algorithm utilizes the Hough transform for detecting lines on cross sections (Vosselman et al. 2004). Hough transform is a feature extraction technique to find imperfect instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates (e.g., straight lines) are obtained as the local maxima in a so-called accumulator space explicitly constructed by the algorithm for computing the Hough transform (Vosselman et al. 2004). In this research, in order to improve the sensitivity of the line detection, the algorithm runs the Hough transform multiple times. Each time, the algorithm extracts the longest line detected (the largest peak in the parameter space), and removes all occupied pixels on that line in the occupancy image. The next run of the Hough transform will be based on the updated occupancy image with the most obvious line removed. We found that this multiple-run Hough transform approach can detect more lines (especially short lines) in an occupancy image compared with the conventional approach of running the Hough transform once and report all the lines. Figure 6 shows five groups of

points labeled by different colors, which are corresponding to five lines identified



**Figure 4 Five groups of data points corresponding to five lines detected on the cross section of a guide beam segment**

by five runs of the Hough transform on the occupancy image of a segment of point cloud. This figure shows that points on the flanges, the web, and the running surface are separated. The reason that the algorithm detected two planes in data points on the running surface is that the projection direction

is not exactly parallel to the running surface, such that the projected points are not forming a line but a wide stripe. The algorithm detected that wide stripe as two lines.

In Step 5, with the detected lines on a cross-section, the algorithm computes the intersection points between the detected horizontal lines (the web and the running surface) and the vertical lines (the flanges). These feature points can be used to generate the width of the guide beam: the distance between two flange-web intersection points is roughly the width of the web. The only issue is that the flange is about 1 cm thick, and the detected flange line is actually the centerline of the flange cross-section. Using that flange center-line for calculating the guide beam width will generate values slightly smaller than the actual width of the guide beam, as will be shown in the following section.

In Step 6, the developed approach calculates the distances between the detected flange-web intersection points and the interpolated axis points used in Step 2 on each guide beam cross-section. If these distances are larger than a specific threshold, then the whole iterative guide beam feature extraction process will be repeated until the extracted parameters converge.

## Evaluation Results and Discussion

The evaluation of the developed approach uses point clouds collected on a site for experimental run of metro vehicles. The laser scanner used in this study was a phase-based scanner, which is capable of collecting 3D point clouds with angular resolution of 0.018 degree (20,000 pixels for a row in a collected panoramic range image), and with an accuracy of about 5 mm for every point within 50 m from the scanning location. After importing such dense and accurate point clouds into a commercially available reverse engineering environment, we manually selected a number of axis points, exported them as the input of our approach, and tested three critical components of the approach: 1) Axis feature extraction; 2) Cross-section feature extraction; 3) Axis updating using the extracted cross-section feature points.





**Figure 5** Manually picked axis points of a section of guide beam in the data collected from a site for experimental run of Metro vehicles

Table 2 lists the results of axis feature extraction. This table indicates that for the straight section, the error of the generated axis is about 0.1 mm, which is very small. However, the extraction results of the length of the spiral curve and the radius of the circular curve are less accurate. These observations indicate that the developed approach needs to be improved for extracting axis features of curved guide beams, especially for spiral sections. More specifically, we found that the boundary point between the straight

line and the spiral curve is difficult to be located accurately, and even large variations of the possible TS point (end point of the growing straight line) would not have substantial impact on the average deviation value of the straight line section. As a result, the generated TS locations have large uncertainties. This fact implies the necessity of developing a better fitting quality measure other than the average deviation used in this research for accurately locating the TS point.

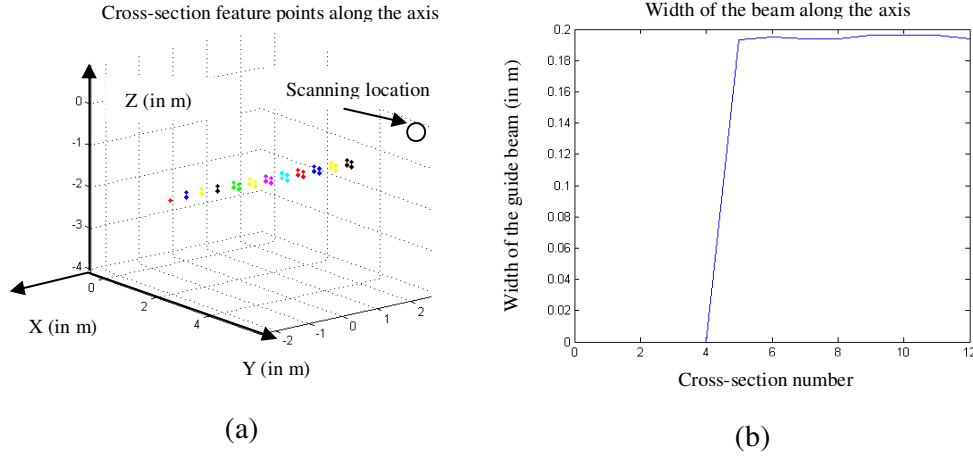
**Table 2** Guide beam feature extraction results analysis

Axis section	Recognized feature values	Errors
<b>Straight line</b>	A point on line: (-0.2099 m, 1.001 m) Line direction vector: (0.5042 m, 0.8036 m)	0.001 m (the average deviation of points on the generated line from the as-designed line)
<b>Spiral curve</b>	Length of spiral: 3.45 m (as-designed: 5.76 m)	2.31 m (difference from the as-designed spiral curve length)
<b>Circular curve</b>	Circle radius: 35.15 m (as-designed: 38.10 m)	2.95 m (difference from the as-designed circular section length)

For cross-section feature extraction, the algorithm extracted flange/web and flange/running-surface intersection points on all cross-sections for a straight guide beam (for curved guide beam, the exploration is on-going). Figure 8 shows (a) the extracted feature points of twelve cross-sections of that beam segment; and (b) the calculated guide beam width based on these feature points on those twelve cross-sections. An interesting observation is that, for cross sections closer to the scanning location, more feature points can be extracted (Figure 8a). The non-uniformity of the data density causes this phenomenon: point clouds closer to the scanning location are denser, so that in the sub-step 2 of step 4 of our approach (occupancy image generation), more cells will be recognized as “occupied” with the constant occupancy threshold 20 (any 3 mm by 3 mm cells with more than 20 points in them are labeled as “occupied”). As a result, more lines will be detected for cross-sections with higher data density (sections 5 to 12 in Figure 8); while for cross-sections further away from the scanning location (sections 1 to 4 in Figure 8), the algorithm could not extract enough lines for detecting all four feature points (two flange-web intersection points, two flange-running surface intersection points). That’s why in Figure 8 (b), for cross-sections 1 to 4, the generated width is 0 m



(there were not enough feature points to compute the width).



**Figure 8 (a) Extracted feature points on 12 cross-sections along a straight guide beam; (b) Automatically extracted guide beam widths of the 12 guide beam cross-sections**

For cross-sections 5 to 12, the obtained guide beam widths vary from 0.193 to 0.196 m; while the as-designed guide-beam width is 0.2 m. In-depth analyses indicate that the algorithm only extracts one line locating at the center of flanges, which are slabs of about 1 cm thick. The distance between two center lines of flanges should be slightly smaller than the actual width of the guide beam, which is defined as the outer-outer distance between two flanges. Therefore, the automatically generated guide beam widths are expected to be smaller than the as-designed value, and the testing results are consistent with this expectation: adding 5 mm (half of the flange thickness) back to the generated widths results in accurate results, the errors of which are equal or less than 2 mm.

For updating of the guide beam axis, several preliminary analyses and visual inspections indicate that utilizing the feature points can roughly adjust the selected axis points to particular cross-section feature points (e.g., cross-section center) for more accurate axis feature extraction. Due to the space limit of this paper, these results are not presented.

## Conclusions and Future Research

This paper presents an integrated approach for combining the two aspects of guide beam feature extraction from 3D laser-scanned point clouds: axis feature extraction and cross-section feature extraction. The evaluation results of this approach on point clouds collected by a phase-based scanner indicate the potential value of this approach. The major findings of this research are two-fold. First, it has been showed that a fitting-and-growing approach can extract parameters of the straight section, spiral section, and circular section of a curved guide beam, and the primary difficulties of accurately extracting these parameters occur in the fitting of the curved sections and localizing the connection point between the straight section and the spiral section. Second, a multiple-run Hough-transform-based line detection approach can accurately extract cross-section lines of a straight guide beam, and can automatically generate widths of the beam with 2 mm accuracy. The major difficulties occur in the non-uniform data densities of the point clouds, which require different parameter settings to achieve the best line detection results for

different parts of the point clouds. Instead of setting constant parameter values of the algorithms, in the future, we plan to explore an approach which can automatically adjust the algorithm parameter settings according to the data density variations of point clouds.

Future studies should target at improving the axis feature extraction and cross-section feature extraction algorithms to overcome the difficulties mentioned above. In addition, in order to achieve an integration of axis extraction and cross-section feature extraction algorithms to get an iterative guide beam feature extraction approach for obtaining accurate axis and cross-section feature through several circles of the “axis fitting – cross section feature recognition” process, we plan to further test the mechanism of using recognized cross-section features to update axis points for converging to an accurate axis.

## Acknowledgements

This research was partially supported by the National Science Foundation (NSF) under Grant No. 0420933 and 0121549 and by the Pennsylvania Infrastructure Technology Alliance (PITA) in conjunction with Bombardier Transportation. Any opinions, findings, conclusions, or recommendations presented in this publication are those of authors and do not necessarily reflect the views of the NSF and Bombardier Transportation.

## References

- Barker, R. (2004). "LSGE: The Least Squares Geometric Elements Library." NPL Management Ltd, ed., NPL Management Ltd, Open Source MATLAB Toolbox.
- Gilsinn, D. E., Cheok, G. S., Witzgall, C., and Lytle, A. (2005). "Construction Object Identification from LADAR Scans: An Experimental Study Using I-Beams." *NISTIR 7286*, National Institute of Standards and Technology, Gaithersburg, MD.
- Gruen, A., and Akca, D. (2005). "Least squares 3D surface and curve matching." *Isprs Journal of Photogrammetry and Remote Sensing*, 59(3), 151-174.
- Kwon, S. W., Bosche, F., Kim, C., Haas, C. T., and Liapi, K. A. (2004). "Fitting range data to primitives for rapid local 3D modeling using sparse range point clouds." *Automation in Construction*, 13(1), 67-81.
- Tang, P., Huber, D., and Akinci, B. "A Comparative Analysis of Depth Discontinuity and Mixed Pixel Detection Algorithms." *Proceedings of the 6th International Conference on 3-D Digital Imaging and Modeling*, Montréal, Québec, Canada, 29-38.
- Vosselman, G., Gorte, B. G. H., Sithole, G., and Rabbani, T. (2004). "Recognising Structure in Laser Scanner Point Clouds." *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 46(part 8/W2), 33-38.