

POSE OF I-BEAMS FOR CONSTRUCTION SITE AUTOMATION

David E. Gilsinn, National Institute of Standards and Technology
Geraldine S. Cheok, National Institute of Standards and Technology
Alan M. Lytle, National Institute of Standards and Technology
david.gilsinn@nist.gov

Abstract: Automation of construction processes can result in reduced project costs and increased worker safety. A process that lends itself to automation is the picking and placing of objects. However, determining the pose (position and orientation) of an object is critical. LADAR (laser detection and ranging) data provides 3D information of a scene, but the data are noisy, contain outliers, and have phantom points along edges of objects. A preliminary algorithm to preprocess the data and to compute the object pose is presented. The algorithm was validated through comparison with experimental measurements.

Keywords: Binning, LADAR, object recognition, object segmentation, pose determination.

1. INTRODUCTION

Inefficiencies in component, material, and trades tracking on construction sites has been repeatedly voiced at the National Institute of Standards and Technology (NIST) workshops (Automated Steel Construction, Data Exchange Standards at the Construction Site) as a top concern. Improved asset tracking systems will enable both leaner construction and enhanced security, as well as lay the foundation for levels of automation envisioned in an Intelligent and Automated Construction Job Site. The combination of LADAR scanning technology, real-time object recognition, automatic identification, and tracking technologies provide powerful potential mechanisms for assessing real-time status of construction site operations and lay ground work for autonomous construction systems.

The Construction Metrology and Automation Group (CMAG) of NIST has an ongoing effort in Construction Object Recognition and Tracking [1]. Current efforts in this project are focused on determining the pose of an object given a point cloud (in the form of x, y, z data) of a scene as obtained from a LADAR. The objective of these efforts is to rapidly determine the pose of an object, without user intervention, so that an automated crane can locate the object, move towards it, and align itself (i.e., gripping mechanism) to pick-up the object. A more exact position and orientation of the object will be determined as the crane nears the object using on-board sensors.

The current efforts include developing an algorithm to segment the data (remove data not belonging to the object of interest) and to determine the pose of a targeted object, an I-beam in this case. The I-beam specifications are inputs to the algorithm and are used to create a bounding box that encloses the I-beam. This bounding box is used as a means to eliminate extraneous objects from the scene by

comparing the bounding boxes of identified objects with that formed from the I-beam specifications.

This paper presents a summary of the initial work involved in implementing an algorithm to post-process data obtained from a data base of I-beam specifications as well as data obtained from experiments conducted to quantify the uncertainty of the algorithm. The overall algorithm is composed of several separate sub-algorithms. These will be discussed in the next sections.

2. DATA SETS

Two data sets are involved with the processing. The first is a set containing known objects and is the I-beam data base, which is a file that consists of groups of I-beam specifications, such as the beam name or identifier, its length, its depth, flange width, and web and flange thicknesses. After reading the data base, the main program converts the information to 24 points that represent each I-beam (Fig. 1).

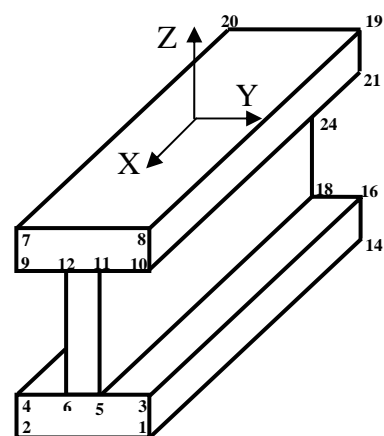


Fig.1. I-Beam vertex representation.

The second data set is one of the scanned LADAR data sets that are given as files of x, y, z points, one triplet per line, relative to the scanner axis system. The number of points in these data sets ranges from about 10 000 to about 80 000 (highest scanner resolution).

3. BINNING ALGORITHM

In order to make the scanned data sets more manageable, the data points are accumulated into volume cubes or *voxels* that encompass the entire range of the data set. This process is called *binning*. The algorithm prompts the user for the number of $x, y,$ and z axes partitions which are used to determine voxel size. The size of each voxel is calculated as a triplet (dx, dy, dz) , e.g., $dx = (xmax - xmin)/(\# \text{ of partitions in } x)$. Each (x, y, z) point is associated with an (i, j, k) value for each bin. In terms of x , for example, this is computed for each point r of the scanned data as $i(r) = \text{fix}[(x(r) - xmin)/dx] + 1$, with due consideration of the boundary voxels. Similar index calculations are performed for $j(r)$ and $k(r)$ for each of say N points. Once the $(i(r), j(r), k(r))$ indices are computed for each point, the bin count (number of points) per voxel can be determined. The bin count of all voxels is initially set to zero. Due to the binning, many (x, y, z) points will have the same $(i(r), j(r), k(r))$ indices. The algorithm iterates by starting with index triplet $(i(I), j(I), k(I))$ and finding all matches of this triplet by a simple linear search. The number of matches, say M , gives the bin count for the $(i(I), j(I), k(I))$ voxel. These matched indices are then eliminated from the $(i(r), j(r), k(r))$ list and a new allocated list is formed of length $N-M$. The old list is deallocated. This begins to reduce the number of $(i(r), j(r), k(r))$ triplets that need to be linearly searched. The algorithm then begins again with the new $(i(I), j(I), k(I))$ triplet, finds the matches, counts them for that voxel and then eliminates those triplets. The process continues until there are no more $(i(r), j(r), k(r))$ triplets. All voxels would then have a bin count associated with them. Many would have a bin count of zero.

The binning process was selected instead of surface fitting for several reasons. It allows objects to be modeled by polygon structures, since it is not necessary to develop a detailed model of an object in order to align an automated crane with the object. Polygon models provide sufficient structure for alignment and they allow fast comparisons with similar polygon models of the ideal structures by only needing to match relatively few vertices.

4. VERTICAL VOXEL STRINGS

LADAR scans can generate a large number of mixed or phantom pixels in a scan [2] (see Fig. 6). These phantom pixels are usually caused by a portion of the

beam being reflected from an edge of an object and the other portion being reflected from another object behind the first. The resulting measurement places a point somewhere between the two objects. This section describes a process used to eliminate these points as well as eliminate such points as ground hits.

The algorithm is based on the idea that those columns with floor hits or phantom pixels would have a number of voxels with small bin counts and in general very few voxels connected to them in the column. It should be noted that voxels can have more than one phantom pixel in them depending on voxel resolution. By a connected voxel, we mean one with a nonzero bin count either directly above or directly below. These are also referred to as *neighboring voxels*. A vertical string of voxels is defined to be the set of non-zero voxels in one column that are direct neighbors of each other. The most significant vertical voxel string in a column is the longest string in each column. This is determined by examining each vertical column of voxels and associating a value of one in a buffer for that column if a bin is non-zero, otherwise the buffer is set to zero. The string length and maximum string length for the column are initialized to zero. Each voxel up the column is examined. If the buffer value is one then the string length is incremented and the maximum string length is set to the current string length if it is longer than a previous string length in the column. Once the buffer becomes zero the current string length is reset to zero and the next voxel is examined until the column is finished. The maximum string length per column is automatically produced. Also, the maximum overall string length is the maximum of all of the column string lengths. Columns that include those voxels that are to be eliminated are determined by comparing the maximum string length up a column with the maximum overall string length. If the maximum column string length is less than a prescribed fraction of the maximum overall string length then the bin counts in that column are all set to zero. Currently that factor is selected interactively, but future algorithm enhancements will include a more general factor determination. Columns with floor hits or phantom pixels usually have maximum column string lengths of one or two as opposed to maximum column string lengths of 40 or 50 or more for tall objects, again depending on voxel resolution.

5. SEGMENTING OBJECTS

Once the outlier voxels, such as phantom pixels and ground hits have been eliminated, entities called *objects* can be identified. The basic premise of the object segmentation portion of the overall algorithm is that objects are made of neighboring voxels. Thus this portion of the algorithm accumulates neighboring voxels into object structures. The sub-

algorithm used here is based on one proposed in Nikolaidis and Pitas [5] for calculating the volume of connected components in 3-D. The process starts by constructing a three dimensional array, called the *mask array*, of the same size as the voxel array and assigning the value one to the mask element (i, j, k) associated with the voxel indexed (i, j, k) if that voxel element is non-zero, otherwise the mask array element is set to zero. The segmentation proceeds by first finding a non-zero mask element and setting it to zero after storing its voxel information on a stack. It then proceeds with setting to zero each of the mask array elements for every neighboring non-zero mask element that is found, and then placing the associated voxel information associated with each of those mask elements on a temporary holding stack. For those neighbors, their neighbors are examined in an outward expanding group of neighboring voxels. When there are no more non-zero neighboring mask elements to be set to zero, the stack is unloaded into an object structure. Setting the mask array elements to zero for all of the voxels in an object means that when the algorithm goes back to the mask array to start another object it won't find a mask element associated with the object just created.

6. PRINCIPAL AXES

In order to determine the pose of a defined object, it is useful to identify a set of axes that represent the distribution of the voxel centers about the center of data mass. If an object is defined by a group of voxels, each with a center (x, y, z) associated with it, then a set of orthogonal axes relative to the center of data mass can be determined that extend in the directions of the longest axis of data, the second longest, and the third longest. These are called the *principal axes*. For a further discussion of this algorithm and its other applications in graphics see Lengyel [3]. Objects that are only linear or planar do not have a full set of principal axes and will not be considered as legitimate objects in this study. The principal axes are representative of the directions in which the data varies. To determine the principal axes assume that points P_1, P_2, \dots, P_N represent N points in 3D Euclidean space. We first calculate the mean position or center of data mass m by

$$m = \frac{1}{N} \sum_{i=1}^N P_i \quad (1)$$

We then construct a 3 x 3 matrix called the covariance matrix

$$C = \frac{1}{N} \sum_{i=1}^N (P_i - m)(P_i - m)^T \quad (2)$$

This covariance matrix is a symmetric matrix that represents the correlation between each pair of $P_i = (x_i, y_i, z_i)$ points. The natural axes of the set of points are determined as follows. First, the eigenvalues of C are found and ordered largest to smallest. The associated eigenvectors are then found. The eigenvector associated with the largest eigenvalue points in the direction of the points having the largest variation, generally the longest axis. The eigenvector associated with the next largest eigenvalue points in the direction of the next longest axis and similarly with the third. The process of developing the principal axes is related to the statistical technique called principal components analysis. For a discussion of this topic see Montgomery, et al. [4].

7. CREATING BOUNDING BOXES

Although there are uncertainties in locating the boundaries of LADAR scanned objects, it is possible to establish reference surfaces and vertices that can be used to compare against associated reference surfaces and vertices on the I-beams given in the data base. Due to the statistical uncertainties of locating object edges, we use polygons to bound the object. One advantage of this approach is that if a polygon is also constructed around the ideal object from the database, such as an ideal I-beam, then comparisons can be made between well defined polygons. In the case of I-beams, polygons defined as *bounding boxes* are used. The approach is relatively straightforward. Every legitimate object is defined in terms of voxels and each of the voxels has a center (x, y, z) point. Once an object, in particular the I-beam, has been identified in terms of voxels then the vector from the LADAR scanner origin to each voxel center in the I-beam is projected against the principal axes. The lengths of the different sides of a box enclosing all of the voxels defining the I-beam is given by the difference between the maximum and minimum values of the projections along each of the principal axes. The vertices of this box are then easily determined. Next the specifications for an I-beam in the database can be used to construct the 24 points defining the I-beam (Fig. 1). These points can also be used to generate the principal axes for the reference I-beam. Once these principal axes are known and the distance to the center of data mass of the measured I-beam from the scanner is known, it is then possible to construct the x-, y-, and z-rotations to transform the reference I-beam model and map it over the bounding box of the measured I-beam data (Fig. 4). Then a sum of squared errors function can be used to compute the error between the associated vertices of the reference I-beam model and the measured I-beam bounding box. The segmented object will be associated with the known object which yields the smallest error.

8. EXPERIMENTS

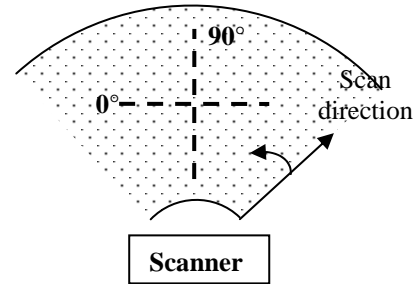
A series of scans were conducted to determine the uncertainty of the pose calculations, the influence of point density on pose uncertainty, the influence of point of view of the scanner relative to the object on pose uncertainty, and the ability of the algorithm to differentiate between two objects of similar shape. In these experiments, an I-beam in an uncluttered environment was scanned using a LADAR with a manufacturer specified range uncertainty of ± 2.5 cm (best case) and ± 5 cm (worst case). The experimental parameters include type of I-beam (2 I-beams), angle of the I-beam in its coordinate frame (0° , 30° , 45° , 60° , and 90° , see Fig. 2a), and point density (3 levels).

For each I-beam rotation, three scanner angular increments (horizontal and vertical) between points were used: 0.072° , 0.108° , and 0.18° . The lowest increment corresponds to the highest point density. Three scans were obtained at each angular increment for a total of 9 scans for each I-beam rotation angle. A total of 90 scans were obtained.

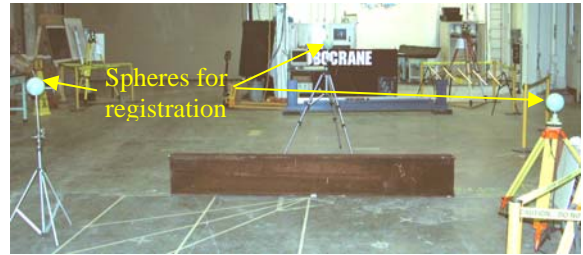
To measure the rotation of the I-beam, the four corners of the top flange were measured with a laser-based site measurement system (SMS) that has a manufacturer stated accuracy of less than 2 mm. These measurements were taken every time the beam was moved. Additionally, there were three spheres (see Fig. 2b) located within the scene for registration purposes and several points on each sphere were measured. This allows for the determination of the transformation matrix between the two coordinate frames (LADAR and SMS). Points on the scanner were also measured using the SMS to locate the scanner in the SMS coordinate frame.

9. POINT REGISTRATION

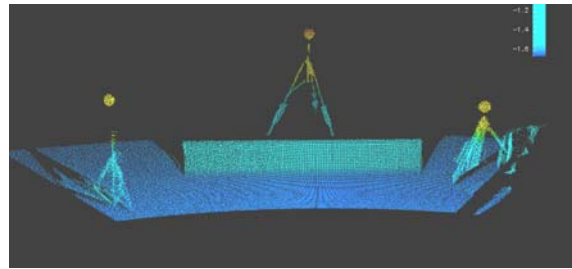
In order to estimate uncertainties of the pose of the I-beams, a transformation between the axis system of a site measurement system (SMS) and the LADAR coordinate frame had to be determined. This was done by assuming that points in both axes systems fell into a global axis system and a rigid body transformation was developed using an iterative algorithm that determined the transformations that linked the SMS and LADAR coordinates. A full discussion of the details of this algorithm is beyond the scope of this proceedings article and will be published elsewhere.



a. I-beam orientation.



b. I-Beam at 0° . Spheres used for registration.



c. Point cloud of scan.

Fig. 2. Experimental set-up.

10. PRELIMINARY RESULTS

At the time of the publication of this paper, only one data set was examined. When all the data have been examined, a discussion of the uncertainties of pose determination will be published.

Figure 3 shows a stacked contour plot of the objects identified in a scan. In the first stage of the segmentation algorithm applied to this data set, 14 objects were identified, using a voxel size of $dx = 0.123$ m, $dy = 0.107$ m, and $dz = 0.027$ m. In the second stage of the segmentation algorithm, all objects that did not have 3 principal axes were eliminated and the remaining object was the I-beam. Figure 4 shows the point cloud and the bounding boxes around the ideal I-beam (green box) and around the LADAR measured data as represented by the centers of the voxels comprising the measured I-beam (blue box). The measured box is slightly wider and shorter than the ideal box. This could be a result of the selected voxel size and will be investigated further.

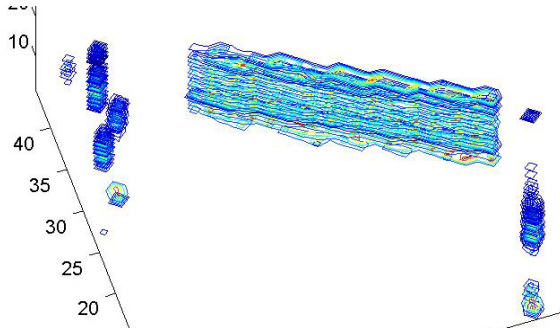


Fig. 3. Segmented Objects.
Angle = 0°, medium point density.

To estimate errors and uncertainties of these bounding boxes, measurements of the four corners of the I-beam, as it rested on the floor as shown in Fig. 2b, were compared with those obtained analytically. The four points selected are the corners of the top flange that correspond to points 8, 7, 19, and 20 in Fig. 1. Table 1 gives the measurement results of these points in the SMS coordinate system.

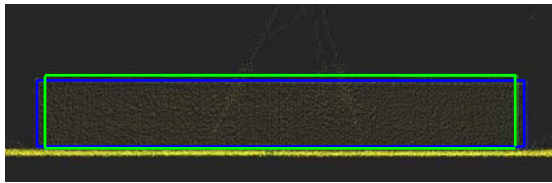


Fig. 4. Bounding boxes. The narrower, taller (green) box shows ideal I-beam. The wider, shorter (blue) box encloses the segmented I-beam.

Table 1
Top Flange Points in SMS Coordinates (m)

Point	x	y	z
8	8.5644	-3.7518	-1.3459
7	8.3901	-3.7738	-1.3492
19	8.2310	-0.7201	-1.3569
20	8.0545	-0.7403	-1.3581

The points in Table 1 were used as the “truth” or reference measurements because the maximum standard deviation of all the measurements was 3×10^{-4} m. Table 2 gives the same points as computed from the bounding box for the scanned I-beam.

Table 2
Top Bounding Box Points in Scanner Frame (m)

Point	x	y	z
8	-4.7790	4.9337	-1.1618
7	-4.8648	5.0691	-1.1628
19	-2.2990	6.5057	-1.1654
20	-2.3849	6.6411	-1.1665

In order to determine the transformation the SMS coordinates to the scanner reference frame, the points on the three spheres shown in Figs. 2b and 2c were measured using the SMS system and the LADAR. Best fit sphere models were then fitted to the measured data and the estimates of the centers in the SMS frame and the scanner frame were made.

An additional point that was used to determine the transformation matrix was the center of the scanner. The center of the scanner cannot be physically measured. Therefore, several points on the scanner were measured with the SMS system and the scanner center, in the SMS coordinate frame, was derived based on these measurements. The coordinates of the scanner center in the scanner frame is (0, 0, 0) by definition.

These four points represent the four points common in the two coordinate frames. Using the approach described in Section 9 to transform the coordinates in the SMS to the scanner frame, a rotation matrix is determined. This rotation matrix indicates that the main rotation is about the Z-axis with rotations about the X- and Y-axes approximately equal to 0.

Transforming the four points in Table 1 we obtain:

Table 3
Top Flange Points in SMS Frame Transformed to Scanner Frame (m)

Point	x	y	z
8	-4.8635	4.9218	-1.1966
7	-4.9594	5.0691	-1.1995
19	-2.2815	6.5452	-1.2165
20	-2.3767	6.6952	-1.2173

The deviations between the coordinates in Table 3 and those in Table 2 are given in Table 4.

Table 4
Deviations = [Table 3 – Table 2] (m)

	X Error	Y Error	Z Error
8	-0.0845	-0.0119	-0.0348
7	-0.0946	0.0000	-0.0367
19	0.0175	0.0395	-0.0511
20	0.0082	0.0541	-0.0508
Mean	-0.03835	0.02043	-0.04335
Std. Dev.	0.05938	0.03141	0.00881

The location of the I-beam in the scanner frame can be found from Table 2. The center point of the top of the bounding box is computed from the four corner points as (-3.5820, 5.7874, -1.1641) m. A similar calculation from the transformed SMS points in Table 3 shows the center point to be (-3.6203, 5.8078, -1.2075) m. The difference between the transformed coordinates and the scanner coordinates is (-0.0383, 0.0204, -0.0434) m. These deviations and those in Table 4 are within the accuracy of the LADAR used. It should also be noted that the center point in the scanner frame has an expected offset

from “true” center. This offset is due to the fact that there are many more measurements of the I-beam on the side of the I-beam that is closer to the scanner; thereby biasing the center towards this side (Fig. 5).

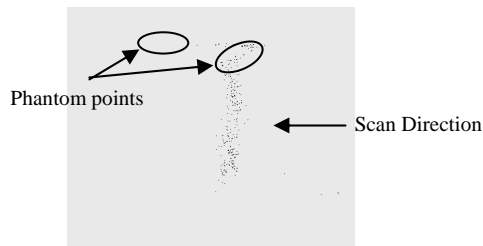


Fig. 6. Point cloud of I-beam viewed along the longitudinal axis of I-beam. Note the phantom points at both edges of the top flange.

As mentioned earlier, the rotation matrix (SMS to LADAR frame) shows that the main rotation is about the Z-axis. Therefore, only the Z-rotation will be discussed in the following paragraphs.

One of the experimental parameters was the angle of the I-beam relative to the scan direction (see Section 8 and Fig. 2a). For the data set that was examined, this angle was 0° (Fig. 6) in the I-beam frame. From Table 2, it can be shown that the longitudinal vector of the I-beam forms an angle 32.2° counterclockwise from the -X axis of the scanner frame.

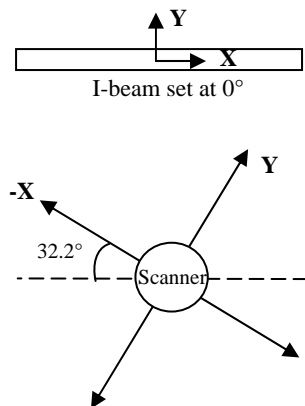


Fig. 6. Rotation of I-beam relative to the scanner coordinate frame.

The rotation about the Z-axis of the I-beam as determined using the algorithm is 32.2° in the scanner coordinate frame. This compares exactly with the measured pose of 32.2° .

11. SUMMARY AND OBSERVATIONS

Preliminary findings of an experiment to determine an object’s pose are presented. The algorithms to

segment an object and to analytically determine the object’s pose are described.

Although the LADAR scan of the I-beam and its surroundings from one perspective generated noisy data, the techniques of vertical voxel string identification, object segmentation, bounding box construction, and rigid body rotations seem to be a set of viable tools for identifying objects, as long as there is prior knowledge of what the likely object candidate is. Identification of objects in this context means obtaining sufficient information about the object and its pose to enable differentiating it from other objects and aligning an automated crane to pick up the object.

Based on the measurements from one experiment, the algorithm was able to determine the position of the I-beam to within the accuracy of the LADAR, i.e., the uncertainty of the position was within the uncertainty of the LADAR. The algorithm was also able to correctly determine the orientation of the I-beam.

As mentioned earlier, at the time of publication of this paper, only one of the 90 data sets was examined. Uncertainties of the pose determination, and effects of point density, angle of the object relative to the scan direction, and voxel size on the pose determination will be reported in a future publication when all the data have been analyzed.

REFERENCES

- [1] Gilsinn, D. E., Cheok, G. S., and O’Leary, D., “Reconstructing Images of Bar Codes for Construction Site Object Recognition,” *Automation in Construction*, 13 (1) Jan, 2004.
- [2] Hebert, M., Krotkov, E., “3-D measurements From Imaging Laser Radars: How Good Are They?,” *IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS ’91*, Nov. 3-5, 1991, Osaka, Japan.
- [3] Lengyel, E., “Mathematics for 3D Game Programming and Computer Graphics”, Charles River Media, Inc., Hingham, MA, 2002.
- [4] Montgomery, D. C., Peck, E. A., Vining, G. G., “Introduction to Linear Regression Analysis”, Third Edition, John Wiley & Sons, Inc., New York, 2001.
- [5] Nikolaidis, N., Pitas, I., “3-D Image Processing Algorithms”, John Wiley & Sons, Inc., New York, 2001.