TRACKING AND CLASSIFYING OBJECTS ON A CONVEYOR BELT USING TIME-OF-FLIGHT CAMERA

Omar Arif, Matt Marshall, Wayne Daley, Patricio A. Vela, Jochen Teizer, Soumitry J. Ray, and John Stewart
Georgia Institute of Technology
Atlanta, Georgia 30332, U.S.A.
omararif@gatech.edu, teizer@gatech.edu

Abstract

The ability to obtain 3D information is vital for many applications in construction, manufacturing, and vehicle automation and autonomy. TOF (Time-of-Flight) sensors, which provide depth information at each pixel in addition to intensity, are becoming more widely available and more affordable. This paper examines the applicability of TOF sensors to several real world problems that can be relevant in automated assembly or sorting applications. The setting is an indoor environment, and the experiments investigate the ability of TOF sensors to provide sensing for a robot whose task is to handle products moving on a conveyor belt. The range information is used to compute the dimensions of- and to recognize- objects moving on a conveyor belt. The geometric and recognition information is then passed on to the robot for further action. The results indicate that there are immediate opportunities for the use of TOF sensors in automating such applications.

KEYWORDS: Time-of-Flight Camera, Range Imaging, Recognition, Visual tracking.

INTRODUCTION

The dynamic nature of a typical construction site requires fast real-time modeling to aid in decision making. Therefore real-time modeling of a construction site necessitates fast 3D data acquisition and processing system. Laser scanners produce high quality dense 3D dataset (a typical scan contains millions of points) of static objects only. In addition to this, the data acquisition and processing is slow and renders it infeasible for real-time modeling. However, 3D range cameras offer another alternative by facilitating affordable, wide field-of-view, automated static and dynamic object detection and tracking at frame rates better than 1Hz (real-time).

Range imaging technology extends the potential of current 3D imaging systems such as laser scanners or stereo vision. Range cameras provide a combination of amplitude, intensity and dense range (i.e. distance) information at every pixel of a two-dimensional sensor array very quickly, thereby capturing dimension and position of objects in real-time. The geometric information aids in locating moving objects such as a person moving in a construction site or an object moving on a conveyor belt. Furthermore, they can potentially be used in vehicles for detecting obstacles to avoid crashes. In places where machines are automated, range cameras can be used to detect the position, motion and speed of objects and people entering the workspace of automated machines in real-time. Such real-time detection enables better coordination of the operations of humans and machines in the work space.

BACKGROUND

Range cameras hold promise for various applications in construction such as obstacle detection for enhancing work zone safety, monitoring and tracking of workers, project progress monitoring, etc. Gonsalves and Teizer (2009) performed tracking of human targets (construction workers) by segmenting and modelling the segmented object using a star skeleton model. To trace the path of the construction workers, a particle filter was used. Furthermore, motion analysis was performed by determining the angles between the different body parts to analyze the posture of the workers.

Teizer et al. (2007) demonstrated that range image data provides feedback regarding the location of objects in the sensor field-of-view, which is of use for active safety features and tools for safe operation on a construction site. To automate heavy equipment operation, real-time object recognition and modelling is necessary (Son et al. 2008). Using a data driven approach, the object of focus was extracted from the background and was modelled using a convex-hull algorithm. The generated model demonstrates the potential to assist in movement and operation of automated equipment. Kim et al. (2009) proposed a technique to model static 3D objects for use in automated operation of construction equipment. The technique consisted of four steps: data acquisition, pre-processing, segmentation and 3D model generation. Individual objects are segmented automatically by a split and merge algorithm. To generate the 3D model, the feature points of segmented objects were connected using a convex hull algorithm.

Hansen et al. (2008) propose a method to track people working in an occluded environment. Through a homographic mapping, the range data points are projected to the ground plane. Each of the individual cluster pixels are tracked using an Expectation Maximization (EM) algorithm for maximum likelihood estimation of the parameters of a mixture of Gaussian clusters in the flat map. To track moving objects around heavy equipment, a 3D spatial modeling technique approach was proposed by Chi et al. (2007). The 3D spatial information was used to build an occupancy grid representation, whereby the grids were clustered to represent objects. For identifying the objects, Hausdorff-based image matching was used to compare the captured objects with priori models from a predetermined object dataset. Teizer et al. (2007) used an agglomerative hierarchical approach to detect, model, and track the position of static and moving obstacles. This work improves the perception and awareness of operators regarding the potential obstacles surrounding them in real-time.

To capture detailed features of an object it is necessary to have a high resolution image, however, existing 3D range cameras have limited X-Y resolution. Theobalt et al. (2007) have proposed that super resolution methods typically used for color images produce high resolution 3D data of a scene using 3D range camera. To effectively reduce noise in the 3D spatial data acquired by range cameras, a probabilistic sensor model was proposed by Lytle (2009). The work shows, that a point cloud of a synthetic 3D image generated by the probabilistic model appears representative of the dataset that would be captured by an actual 3D range instrument. The work contributes towards noise removal algorithms for improving the operation of range imaging systems.

In this work, a TOF camera is put to use in an indoor industrial environment. Specifically, the 3D range information obtained from the sensor is used to extract useful information about objects moving on a conveyor belt. Two applications are considered: First, the dimensions of

boxes moving on a conveyor belt are measured; Second, different construction-relevant objects are recognized and classified. Novel methods are described to efficiently process the 3D information for both applications.

BELT APPLICATIONS

Figure 1 shows the setup of the experiment. The camera is mounted facing the conveyor belt (see Figure 2). Different target objects are placed on the belt. The belt is moved at different speed and the objects placed on the belt are segmented. The point cloud is then extracted. Next the point cloud is processed to obtain useful information about the object. The information is passed to the robot for further handling of the object.

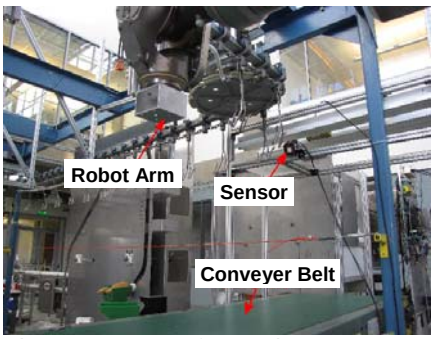


Figure 1: Setup of Experiments

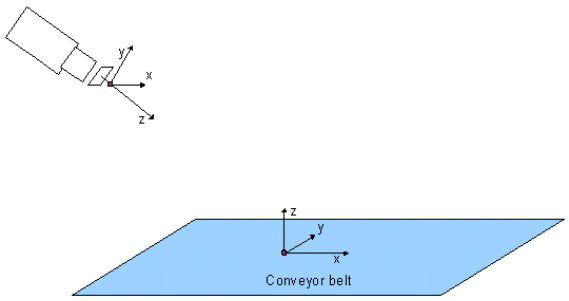


Figure 2: Transformation of the range data from camera coordinate system to belt coordinate system

Coordinate Transformation

The TOF sensor outputs range data of the world within its field-of-view in spherical coordinates. The spherical coordinates are converted into Cartesian coordinates, where the origin of the coordinate system lies at the center of camera, as explained in the next section. The 3D point cloud in the camera coordinate system is transformed to the belt coordinate system with the origin at a known reference point. The coordinate transformation is required for the following two reasons:

- The robot and the range camera need to agree on a coordinate system so that the measurements from the range camera can guide the robot.
- In the belt coordinate system, the z -axis is normal to the plane of the belt. Aligning the coordinate axis in such a manner simplifies extraction of object point clouds on the belt by referencing points above the belt plane ($z > 0$) and within the confines of the belt boundaries.

Estimating the Belt Frame

The following procedure is used to estimate the belt coordinate system, and is carried out only once at the start. Let $u \in R$ be the 3D position of a point in the camera coordinate frame. The belt is manually segmented using the intensity image as shown in Figure 3. The corresponding range data for the belt is selected. Figure 4 shows the point cloud in the camera coordinate frame, with the points corresponding to the belt shown in green. Let S be the set of 3D positions of the pixels belonging to the belt.

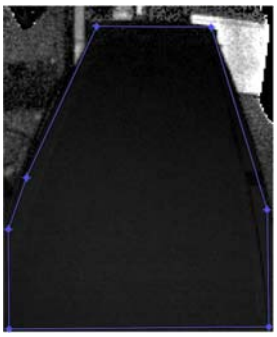


Figure 3: Belt selected using intensity image

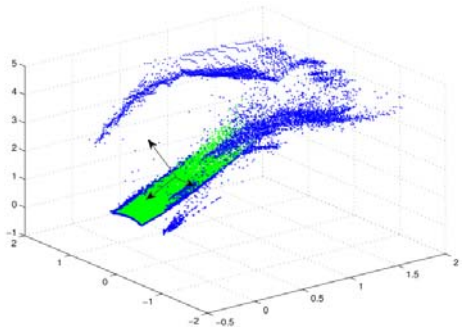


Figure 4: Point cloud in camera coordinate system

The normal e of the plane fitting the set of points is obtained by the following equation:

$$e = \arg_e \min \sum_{u \in S} e^T u + d$$

where d is the distance from the origin to the place, which is assumed to be 0. The above equation is a non-linear optimization problem whose solution is the eigenvector associated to the smallest eigenvalue of the covariance matrix C, which is defined to be C.

$$C = \frac{1}{n} \sum_{u \in S} (u - \hat{u})(u - \hat{u})^T$$

where $\hat{u}$ is the mean of the points in S. Let E be the eigenvectors of the covariance matrix. The eigenvector corresponding to the smallest eigenvalue provides the normal e to the plane of the belt. A point u is transformed to the conveyor belt coordinate system ($v \in R^2$) by the following equation

$$v = E^T.(u - \hat{u})$$

All points lying on the same plane as the belt plane have $z \approx 0$ in the belt coordinate system. All the subsequent computations are performed in belt coordinate system. The coordinate d system is then translated to a known reference point. This can be achieved by changing the mean point $\hat{u}$ with the reference point. Figure 5 shows the point cloud of the belt in belt coordinate system. The blue contour represents the bounding polygon that was selected from the intensity image.

Occupancy Grid

After transformation of the belt point cloud to the belt coordinate system, it is represented by a rectangular occupancy grid. Occupancy grids are based on the principle of allocating range points to a prepared world which is divided into a grid system of variable or fixed voxels. In this case, a 2-dimensional occupancy grid is used for the belt. The convex hull of the belt point cloud is computed and the minimum bounded rectangle (MBR) of the convex hull is computed; shown in red in Figure 5. The MBR is meshed with a grid size of .01m. The belt points are mapped to the occupancy grid. Each grid location is assigned a value, which is the average z value of the points falling within the block. If no point falls within the block, the block is assigned zero. This representation of the belt point cloud is shown in Figure 6. This representation has the following benefits:

- Range data for objects close to the camera is denser than for objects that are further from the camera. The variable density as a function of distance is apparent in Figure 5. The occupancy grid converts the range points to a surface, given by a height function defined over the occupancy grid as shown in Figure 6.
- The number of data points representing a scene are reduced and hence the memory and computational time.
- The effect of noise is also reduced because z values of all the points falling with a block are averaged.
- The resulting rectangular occupancy grid can be visualized as an image, leading to the use of image processing techniques to process the objects on the belt.

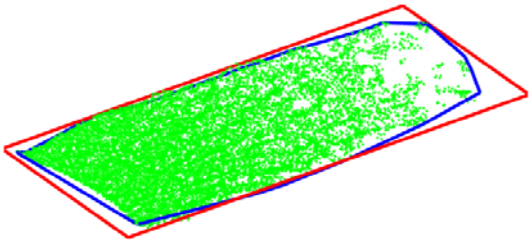


Figure 5: Point cloud of the belt in belt coordinate system. Blue: Convex hull. Red: Mininum bounded rectangle

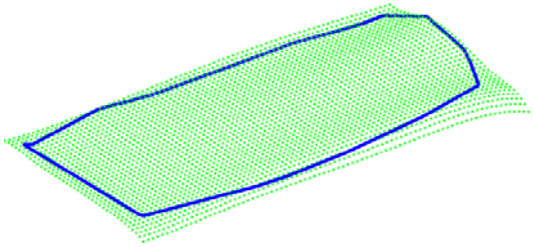


Figure 6: Rectangular occupancy grid and belt surface map

The range measurements are based on measuring the phase of the reflected signal. The signal may get reflected by multiple surfaces before returning to the sensor. In this situation the light travels by the direct as well as indirect path and the distance is then the weighted average of the path distances. If this happen, the belt point cloud may not lie on a plane. To account for that, we learn the surface of the belt using polynomial approximation and evaluate the surface at each grid location.

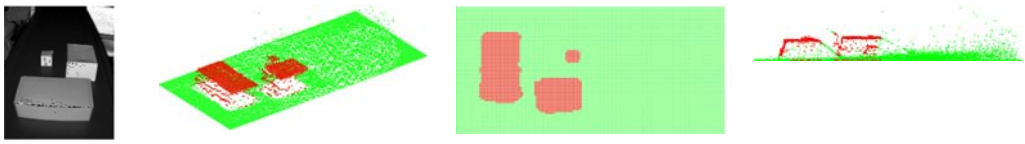
Extracting Point Cloud of Objects on the Belt

The steps explained in the previous section are carried out once using the first frame. In the subsequent frames, following steps are carried out to extract the point cloud of the objects on the belt. Only 3D range data is used and the intensity information is ignored.

- Range data obtained from the sensor is transformed to the belt coordinate system. All the points falling outside the convex hull of the belt are discarded.
- Points within the convex hull are mapped to height values defined over the occupancy grid, which is the average z value of all range points falling within a block. If no point falls within the block, the block is assigned the value zero. The value assigned to each grid location is the distance of the point cloud to the belt.
- The distance value at each grid location is subtracted from the polynomial approximation of the belt evaluated at that grid location. If the result is bigger than a predefined threshold, the point belongs to the object on the belt; otherwise it belongs to the belt.

The result of applying the above steps to a sample frame is shown in Figure 7. Figure 7(a) shows only the range points lying within the bounds of the belt. Red points belong to the objects on the belt, whereas green points belong to the belt. Segmentation results from other

views are shown in Figure 7(b-d). Two of the boxes in the intensity image appear to be connected and it may not be easy to identify individual boxes using only the intensity image. However, they are easily identified, as can be seen from the top view.



(a) Intensity image (b) Segmented point cloud (c) Top view of segmentation (d) Side view of segmentation

Figure 7. Extracted point cloud of the objects in different views.

Processing Objects on the Belt

By following the procedure explained in the previous section, point clouds of the objects on the belt are obtained. The point clouds are further processed to extract useful information about the objects such as their dimensions or their classification.

Extracting and Computing Dimensions of Boxes

The first application is to find the dimensions of the boxes moving on the conveyor belt. The following steps are carried out to measure the dimensions of the boxes.

Determine the number of boxes: After processing according to the previous section, a height mapping of the belt is obtained, *z* see Figure 8(a). Thresholding produces a binary image as shown in Figure 8(b). Morphological operations applied to the binary image fill holes and remove isolated points, producing the image in Figure 8(c). Specifically, Matlab's **bwmorph** is used with the operation **majority**. Then, **bwlabel** is used to find the number of distinct objects as shown in Figure 8(d), where each object has a unique color. Similarly, **bwareaopen** is used to discard connected objects have fewer than p pixels.

Measure dimensions of each box: To determine the dimension of each box, the point cloud corresponding to each box is projected back to the belt, and a minimum bounded rectangle (MBR) is fitted to the two dimensional point cloud as shown in Figure 8(e).

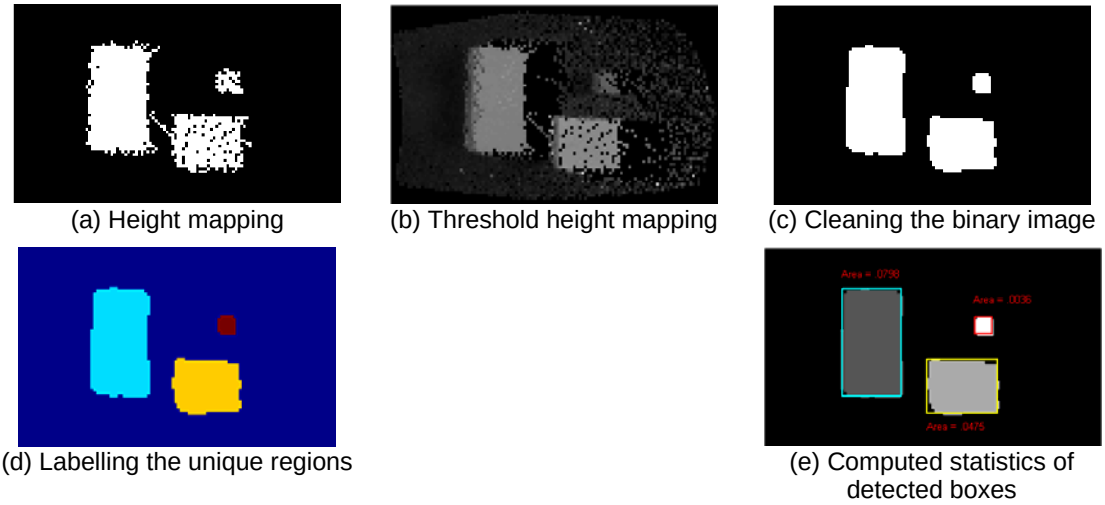
Table 1: Mean and standard deviation of the area of the box moving on the conveyor belt.

Int time	Belt speed	Orientation 1	Orientation 2	Orientation 3	AV
30	40	$\mu = .0748$	$\mu = .0766$	$\mu = .0768$	$\mu = .0761$
		$\sigma = .0023$	$\sigma = .0041$	$\sigma = .0025$	$\sigma = .0030$
30	60	$\mu = .0762$	$\mu = .0772$	$\mu = .0777$	$\mu = .0770$
		$\sigma = .0032$	$\sigma = .0027$	$\sigma = .0024$	$\sigma = .0027$
60	40	$\mu = .0771$	$\mu = .0759$	$\mu = .0761$	$\mu = .0764$
		$\sigma = .0018$	$\sigma = .0029$	$\sigma = .0023$	$\sigma = .0023$
60	60	$\mu = .0755$	$\mu = .0778$	$\mu = .0778$	$\mu = .0760$
		$\sigma = .0036$	$\sigma = .0023$	$\sigma = .0024$	$\sigma = .0028$

To determine the accuracy of the procedure, the area measurements were carried out at different belt speeds, box orientations, and sensor integration times. The results are tabulated in Table 1.

Classifying Construction Tools

The second application is the classification of construction tools moving on the conveyor belt using the 3D point cloud obtained from the sensor. Recognizing/classifying 3D objects is an important task in many applications and it has gained a lot of attention, see for example Ajmal S. Mian (2006), Bronstein et al. (2005), Frome (2004) and Johnson (1999), and the references therein. Most methods work by extracting local or global geometric features from the 3D surface. The features are translational, rotational and size invariant. Local surface descriptors are more useful when occlusion occurs (Li and Guskov 2007). Two most widely used local surface features are the estimated surface curvature and normals (Rusu, 2008). However, they are highly sensitive to noise, which makes them unattractive for 3D object recognition using noisy TOF sensor's measurements.



(a) Height mapping (b) Threshold height mapping (c) Cleaning the binary image

(d) Labelling the unique regions (e) Computed statistics of detected boxes

Figure 8. Processing the point cloud to detect boxes and compute associated statistics.

The eigenvalues of the Laplace operator have also been used for object recognition for both the 2D images (Khabou 2007) and 3D (Reuter 2006). The eigenvalues are tolerant to noise (Khabou 2007), and depend only on the intrinsic geometry of the object. In this work, eigenvalues of the Laplacian are used for object recognition. However, this work differs from the previous approaches to object recognition using eigenvalues in several aspects. First, the above mentioned approaches to object recognition using eigenvalues require the complete model of the object. In our case, the TOF camera outputs only partial 3D point cloud of the objects in view. Secondly, the target object has to be first detected in the scene and then its point cloud extracted from the background. The resulting point cloud is then processed for object recognition.

The algorithm is divided in two phases, a training phase and a tracking phase. In the training phase, the eigenvalues of the n objects that are to be recognized are computed from the extracted point cloud. The objects are placed in different orientations in front of the sensor. A total of n support vector (SV) machines are trained on the computed eigenvalues to classify each object from the rest. In the testing phase, the eigenvalues are computed and fed to the n binary classifiers (SV machines) for classification.

Eigenvalues of the Laplacian

To compute the eigenvalues of the Laplacian, the graph Laplacian is constructed. The graph Laplacian is a matrix $L = D - W$, where W is a matrix given by

$$W_{ij} = \begin{Bmatrix} \exp\left(-\left\|u_i - u_j\right\|/2\sigma^2\right) \\ 0, otherwise \end{Bmatrix}, \text{ if (i,j) is an edge}$$

and D is diagonal weight matrix with $D_{ii} = \sum_j W_{ij}$. Laplacian L is a symmetric, positive semi definite matrix. Compute the eigenvalues of the generalized eigenvector problem:

$$L\phi = \beta D\phi$$

where ϕ and β are the generalized eigenvectors and eigenvalues of the laplacian L. The eigenvalues are $0 \le \beta_1 \le \beta_2 ... \le \beta_k$. Leave out the eigenvalue 0, and use the next k eigenvalues as a feature vector corresponding to the point cloud.

Experiment

Six construction objects (elements of formwork; a helmet; an extrusion) of various sizes and shapes were considered as shown in Figure 9. In the training phase, the objects were placed in 10 different orientations. Some of the orientations of the first two objects are shown in the bottom row of Figure 9. In each orientation, 10 different measurements were taken. The Laplacian eigenvalues from the resulting 600 points clouds were computed. 14 eigenvalues were retained. Figure 10 shows the means and standard deviations of the eigenvalues of the six objects. The eigenvalues are invariant to orientation. The small variations present in the eigenvalues arise from incomplete sensing of the objects due to occlusion.

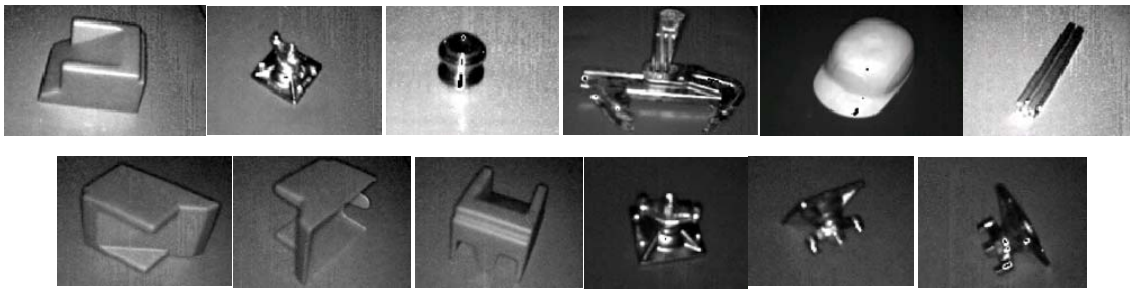


Figure 9. Objects recognition: (a) First row: six objects used for recognition. (b) Bottom row: Sample orientations of the first two objects

Six binary Support Vector Machine (SVMs) classifiers were trained on a dataset of 600 eigenvalues to separate each object from the rest. For the SVMs a Gaussian kernel was used. For testing, the objects were placed in different orientations and recognized using the SVMs. If two or more SVMs produce a positive output, it is taken as an error. The results are tabulated in Table 2. The last three columns show the computed and averaged eigenvalues from 3, 5, and 8 consecutive frames that were fed into the SV machine. The last row lists the total error.

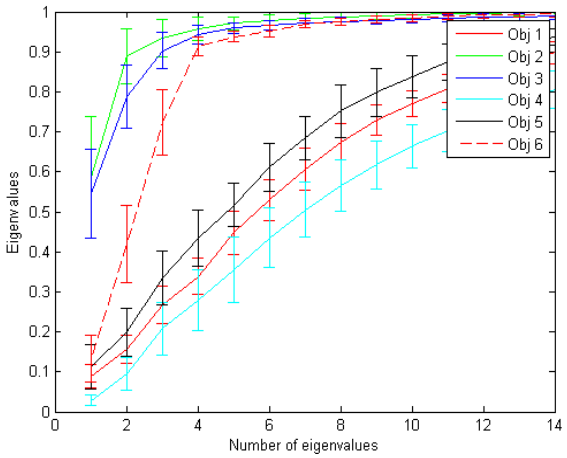


Figure 10. Mean and standard deviation of each object's eigenvalues.

Table 2: Success rate of recognizing the six objects. In last three rows the eigenvalues computed from 3, 5, and 8 consecutive frames are averaged.

Class	Success Rate (Train)	Success Rate (Test)	Success Rate (Total)	Success Rate (av 3)	Success Rate (av 5)	Success Rate (av 8)
Obj 1	100%	97.8%	98.9%	100%	100%	100%
Obj 2	97.9%	95.2%	96.6%	95.8%	97.4%	99.2%
Obj 3	97.9%	95.7%	96.9%	96.6%	96.6%	99.2%
Obj 4	100%	99%	99.5%	99.2%	100%	100%
Obj 5	100%	98.5%	99.3%	99.2%	100%	100%
Obj 6	100%	100%	100%	100%	100%	100%
Total	95.9%	86.3%	91.2%	90.7%	94.1%	98.3%

Conclusion

This paper examined the utility of Time-of-Flight sensors for automatically detecting, processing, and recognizing construction objects on conveyor belts. A methodology for extracting the individual objects was described, together with a classifier strategy for identifying distinct classes of objects on the conveyor belt. The classification could be used to isolate and bin the distinct objects. Future work will close the loop using an actual manipulator.

BIBLIOGRAPHY

Ajmal S. Mian, M. B. (2006). Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1584-1601.

Bronstein, A., Bronstein, M., ands Kimmel, R. (2005). Three-dimensional face recognition. *International Journal on Computer Vision* , 5-30.

Campbell, R. et al. (2001). A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding* , 166-210.

Chi S., Caldas C., and Kim D. Y. (2007). Object Identification Based on 3D Spatial Models of Construction Sites: *ASCE Conference on Computing in Civil Engineering*, 729-736.

Frome, A. et al. (2004). Recognizing objects in range data using regional point descriptors. *Lecture Notes in Computer Science*, 224-237.

Gonsalves, R., and Teizer, J. (2009). Human motion analysis using 3D range imageing technology. *International Symposium on Automation and Robotics in Construction,* Austin TX.

Hansen D. W., Hansen M. S., Kirschmeyer M., and Larsen R. (2008) Cluster tracking with time-of-flight cameras: IEEE Computer Vision and Pattern Recognition Workshops, 1–6.

Johnson, A. et al. (1999). Using spin images for efficient object recognition in cluttered 3 d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* , 433-499.

Khabou, M. et al. (2007). Shape recognition using eigenvalues of the Dirichlet Laplacian. *Pattern Recognition* , 141-153.

Kim C., Son H., and Park Y. (2009) Rapid 3D Object Modeling Using 3D Data from Flash LADAR for Automated Construction Equipment Operations : 26th International Symposium on Automation and Robotics in Construction .

Li, X., and Guskov, I. (2007). 3D Object Recognition from Range Images Using Pyramid Matching., (p. ICCV workshop on 3D Representation for Recognition).

Lytle A. M. (2009) Development of a Probabilistic Sensor Model for 3D Imaging System: 24th International Symposium on Automation and Robotics in Construction (ISARC), 75 – 80.

Reuter, M. et al. (2006). Laplace--Beltrami spectra as 'Shape-DNA'of surfaces and solids. *Computer-Aided Design* , 342-366.

Rusu, R. et al. (2008). Persistent point feature histograms for 3D point clouds. *Intelligent Autonomous Systems*, (p. 119).

Son, H., Kim, C., Kim, H., Choi, K., and Jee, J. (2008). Real-time object recognition and modeling for heavy equipment operation. *International Symposium on Automation and Robotics in Construction,* Vilnius, Lithuania.

Teizer J. (2008). 3D range imaging camera sensing for active safety in construction, Special Issue Sensors in Construction and Infrastructure Management, ITcon,13, 103-117.

Teizer, J., Caldas C.H., and Haas, C.T. (2007). "Real-Time Three-Dimensional Occupancy Grid Modeling for the Detection and Tracking of Construction Resources". *ASCE Journal of Construction Engineering and Management*, 133(11), 880-888, Reston, Virginia.

Theobalt C., Ahmed N., Ziegler G., Seidel H.P. (2007). High-Quality Reconstruction from Multiview Video Streams. *IEEE Signal Processing*, 24(6), 45-57.