

Detecting Occluded People for Robotic Guidance

E. Martinson

Abstract— Often overlooked in human-robot interaction is the challenge of people detection. For natural interaction, a robot must detect people without waiting for them to face the camera, get far enough away to be fully present, or center themselves fully within the field of view. Furthermore, it must happen without requiring immense amounts of processing that are not practical for real systems. In this work we focus on person detection in a guidance scenario, where occlusion is particularly prevalent. Using a layered approach with depth images, we can substantially improve detection rates under high levels of occlusion, and enable a robot to detect a target that is moving into and out of the field of view.

I. INTRODUCTION

Person detection is a fundamental skill in human robot interaction. Before a robot can begin a conversation, hand over an object, or work collaboratively with a human partner, the robot must first know if a person is even present in the environment. In healthcare robotics, the challenges are particularly difficult, as people cannot reliably change their behavior to suit the robot. This is the larger domain we are developing solutions for.

Our specific goal is a navigational aid for blind individuals. Currently, visually impaired individuals are dependent on GPS-based navigation systems [1] for guidance through unknown environments. Such aides, however, often fail indoors, and have no way of detecting or incorporating new obstacles in their route selection. We believe that a robot can help, assisting visually impaired people move through difficult environments by giving them vibration feedback [2]. For person detection, the challenge is to detect, recognize, and track people behind the robot, including, foremost, their human partner, as well as other people in the surroundings. This is similar to human following robots [3] (this example tracks people from color histograms), but because the robot is more than just reactive to human motion, it adds new tracking challenges. Path planning amid dynamic obstacles increase the amount of sensor motion, and sometimes force lost tracks to follow paths with a non-holonomic platform. In this paper, we focus on developing a fast and accurate person detection method to enable track recovery as soon as possible, even though the body may be significantly occluded to obstacles or sensor field of view limitations.

* Research supported by the Toyota Motor Company.

E. Martinson is with the Toyota InfoTechnology Center, Mountain View, CA 94943 USA (phone: 650-694-4143; e-mail: emartinson@us.toyota-itc.com).

In this paper, we first identify several typical problem areas in human detection for guiding people through an environment. We then describe a new solution for detecting people in depth images at close range (<5 m) that is robust to occlusions. Finally, we evaluate detection with (1) moving people and/or robots, (2) a study of 19 people from different rotations and levels of occlusion, and (3) a robot guidance scenario in which 4 different detection systems are evaluated.

II. RELATED WORK

The most common methods for detecting people are (1) face detection, and (2) HOG-based classifiers. Face detection is available in OpenCV [4] and is common with RGB only sensors, as well as RGB-D and/or intensity images. When depth data are available, faces can be used to initialize a track [5], or included as part of a larger classifier [6]. The problem is that in some guidance scenarios faces are available in less than 20% of the images due to people facing away from the camera (a relative pose problem), but more commonly because the face is outside the field of view – either vertically or horizontally. Examples of such difficult detection scenarios can be seen in Figure 1.

HOG, or histogram of oriented gradients, based classifiers detect people without the need for a face [7]. In particular, these classifiers focus on detection regardless of the relative pose of the individual, and have been applied to both RGB [8] [9], and RGB-D sensors [10]. HOG features, however, are computationally intensive to calculate, and the best systems, including deformable parts models [8], involve substantial layers of abstraction. The best algorithms can only process between 1-5 Hz (depending on model complexity) without GPU acceleration. In general, robots cannot, in terms of physical space and power consumption, afford to dedicate an entire computer to speeding up person detection. And even with the additional computation, investigations by Dollar et al [11] suggest that these methods still struggle with partial occlusions, even when trained specifically for them.

There are also depth image solutions that do not use RGB data. Kirchner et al [12] and Xia et al [13] look for head-like contours in the depth data, which solves relative pose problems, but still fails when heads are missing or partially occluded. A solution that does not require the head is to use the local curvature of the object. Local surface normals [14] can be used for detecting moving, sitting, standing, and occluded individuals. Accuracy in detecting partially occluded individuals was reported, but testing focused on partial horizontal, or transverse, occlusions (e.g. body missing from waist down). Both transverse and vertical, or



Figure 1a. A close range individual with both shoulders visible. This view is considered normal in the evaluation dataset.



Figure 1b. A close range body missing both the head and one shoulder. This view is considered partially visible in the evaluation dataset.

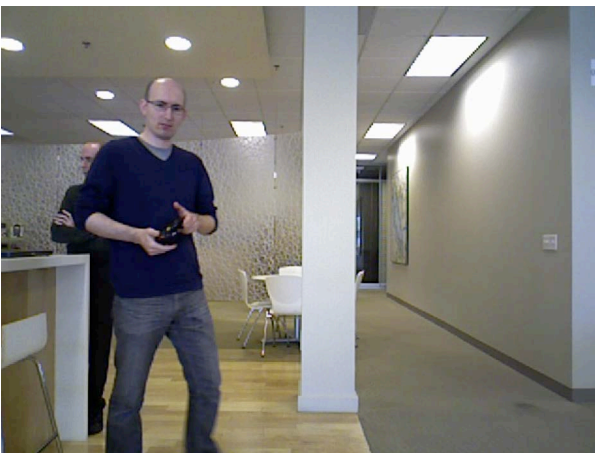


Figure 1c. Two people present in the image. The actual target of the application is partially obstructed by the individual in the foreground, but two sets of eyes are present for detection.

sagittal, (e.g. left or right side of body missing) occlusions are common in a guidance scenario.

Spinello et al [15] were similarly inspired by object curvature when examining velodyne data. After extracting connected objects, they extracted a series of 17 features for each row in the image. These features were based on spline, curve, and line fitting to the data points in each row of the object, and were meant to describe local curvature. AdaBoost was then used to learn a model for each of several height layers of a person – and an object consisting mostly of “people” layers was classified as a person. Although feature calculation can be run at 10 Hz, more recent improvements to accuracy have slowed the system down to 1Hz [16], relying on tracking to detect people in intermediate frames.

III. THE GUIDANCE SCENARIO

There are many different solutions for detecting people that are fully, or even mostly, visible and upright. Once those assumptions are violated the problem becomes much more difficult. For this reason, we focused on detection when being guided by a robot. The specific scenario that we evaluated had a human partner follow our blind-aid robot (Figure 2) through an indoor office environment. In the real scenario, the robot would be constantly communicating to the human the direction of motion to reach the goal via a tactile-belt around the individual’s waist, based on the person’s detected location [2]. For this evaluation, however, the robot was tele-operated by another individual. The human partner began behind the robot and followed it around a curve that passed between a large pillar and a waist high counter top that the human operator was standing at. On average, the human partner was approximately 1.5 m from the robot. This distance increased to as much as 3 m when the robot had to get around the curve and out of the way. To track the person, the robot had a Microsoft Kinect mounted at chest level facing to the rear. This sensor recorded RGB and depth images at 30 Hz. The first 1000 images were saved for evaluation (section V.C). These images have the following characteristics:

- 890 images containing part of at least one person
- 719 images with visible shoulders and/or eyes
- 171 images where largest person has >30% sagittal and transverse occlusion (e.g. only a partially visible person missing eyes and at least one shoulder)
- 331 images with 2 people visible or partially visible

Figure 1 shows a few examples from this data set. The first example is the most common view of the human partner. Note that the head and legs are missing because the camera field of view is not wide enough to capture at such close range. The middle image is also very common. In addition to the face being partially off screen, only one shoulder is visible as the human has strayed to the side of the path. This is also representative of being blocked by other vertical obstructions, like pillars, wall corners, and doors. The final example shows multiple people present in the image.

In general, these images show the challenges for robotic guidance. The normal scenario contains significant transverse and sagittal occlusions due to a limited field of view. Although cameras can sometimes be moved to compensate, physical limitations, such as robot height and the availability



Figure 2. (left) The blind aid robot, a mobile platform with the ability to track people and investigate the environment. (Right) A person following the blind-aid robot through an indoor setting.

of pan/tilt, do not always make this possible. Therefore, an effective person detection system must be able to operate with significant levels of occlusion.

IV. ALGORITHM FOR DETECTING PEOPLE

The layered detection algorithm designed by Spinello et al [15] for depth data was our starting point from which to detect occluded people in guidance scenarios. The original algorithm, however, was not designed for robots. It assumed upright people, especially in the training phase, and specifically rewarded people that were upright and mostly visible during classification. It was also designed for Velodyne data, whereas we are targeting depth images (e.g. kinect and stereo-vision). Velodyne data have comparable horizontal resolution, but the distances are much greater (up to 20 m), so the effective number of data points on the body are much smaller. 60 pixels on a 0.5-m surface at 4-m becomes 25 pixels at 10-m, and 12 pixels at 20-m. Velodyne data are also much sparser in the vertical dimension, using only a total of 64-lines, at 1/3 degree resolution. So there are fewer lines on the body with which to make a decision.

These differences between scenarios and sensor have a significant effect on algorithm design. First, in contrast to algorithms designed with sparse vertical resolution in mind, we can emphasize fusion in the vertical, without the need for assumptions regarding human pose, by modeling people with a Gaussian mixture model (GMM). Like the original system, which grouped layers by relative height for the purpose of building models, a GMM finds commonalities between the input layers and models them with a separate Gaussian. However, it does this in an unsupervised fashion, and requires no assumptions about the training or testing pose, allowing for significant parts of the body to be occluded.

A second improvement is in feature selection, as some features that require fitting curves are noisy at greater distances for lack of data. Our supposition is that by taking advantage of this greater detail, and emphasizing fusion in the vertical, precision can be increased at close range. As such, we will demonstrate that only six features are necessary to appropriately characterize curvature at close range

There are three stages to the improved algorithm: (1) blob extraction, (2) layer classification, and (3) object

classification. Although some figures display color information, only the depth image is necessary for classification and segmentation.

A. Blob Extraction

The blob extraction phase uses a connected components algorithm. First, however, two filters are applied to the depth image, Z . All pixels that have depth not within the range $[0.5, 6.0]$ m are removed. Then pixels outside $[0.6, 2.0]$ -m from the ground plane were removed. Relative pixel height is determined by calculating the X, Y coordinates from the depth image and camera focal length, and then adding a constant offset for the camera's height from the floor:

$$x(row, col) = \frac{(col_{center} - col) * Z(row, col)}{focal_length}$$

$$y(row, col) = \frac{(row_{center} - row) * Z(row, col)}{focal_length} + cam_{height}$$

The constant offset assumes the camera is mounted on the robot parallel to the floor, but speeds up processing by not requiring a full matrix multiplication.

By repeatedly applying a flood fill algorithm to the depth image, the image is divided into connected components with a 10 cm depth threshold. Blobs smaller than 200 pixels are removed. The remaining blobs are analyzed. These thresholds effectively limit detection to a maximum range of 5-m, an appropriate distance for guidance. Figure 3 demonstrates example blobs extracted from an image.

B. Feature Vectors

People live, work, and often, play, in artificial environments. Whether it is inside, or outside in the city, many more surfaces are flat or at least much less curved than people. People by contrast, are curved all over. Their heads, arms, legs, chest, etc. all have some curvature to them. So, even if only part of a person is visible, that part will be curved like a person. For improving the layered detection

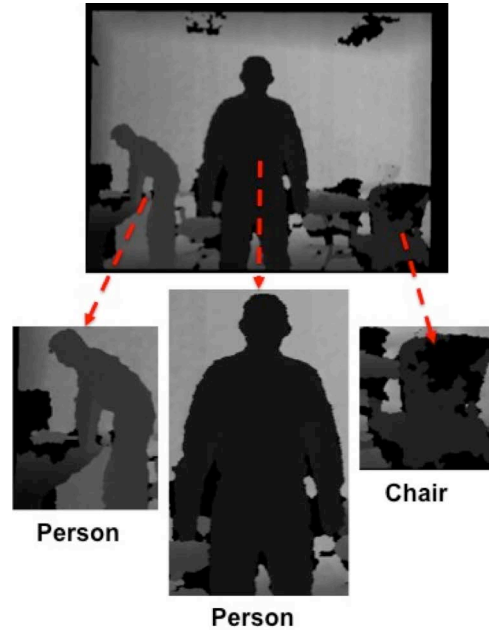


Figure 3. Three sizeable blobs are identified in the depth image: 2 people, and a chair using floodfill to identify connected components.

algorithm at close range, the new feature set, which we will call polyfit features, focuses on row-level curvature.

After extracting potential people blobs, the next step is to build a feature vector for each row of pixels in the blob. Rows that are less than five pixels in length are removed. For each remaining row, a line and a parabola are fit to the data points in the X and Z dimensions. We use the polyfit implementation in OpenCV, extended to double floating-point precision, to find the best line (L) and parabola (P) equations describing the data. From these resulting equations, a six dimensional vector is constructed:

1. Δc , The line depth at the center of the segment minus the center parabola depth as a measure of concavity
2. $RMSE_L$, the rms error of the fitted line equation
3. σ_L , the standard deviation of the fitted line equation
4. $RMSE_P$, the root mean squared error of the fitted parabola equation.
5. σ_P , the standard deviation of the fitted parabola equation
6. k , an estimation of curvature defined in [15]

Except for the choice of features, this is the same process used in the original Velodyne based system. As such, our analyses will include a comparison of the two feature sets.

C. Gaussian Mixture Models

The classification system uses 2 GMM's (one for the person, and one for other objects) to classify each row separately. Row level scores are then fused together to classify a single object. Each model of 30 mixtures was trained on hand-classified objects from the depth images.

Given a new vector, x , and a GMM, M , the likelihood of a vector belonging to M is determined by the equation:

$$p(\bar{x}, M) = \sum_{i=1}^{30} \frac{P_i}{\sqrt{(2\pi)^6 |\sigma_i|}} e^{-\frac{1}{2}(\bar{x} - \mu_i)^T \sigma_i^{-1} (\bar{x} - \mu_i)}, \text{ where } [\sigma_i, P_i, \mu_i] \in M$$

The log-likelihood of a new vector, x , belonging to a person model, *people*, is then:

$$O(\bar{x}) = \log(p(\bar{x}, \text{people})) - \log(p(\bar{x}, \text{not_people}))$$

Given N rows in an object sequence, S , each with its own likelihood, the simplest form of object classification is to sum up the log likelihoods, and apply a threshold. This method, however, assumes that incorrectly classified rows are evenly distributed throughout a blob. In reality, such rows tend to be clustered together, farther down the body where legs and arms dominate. Therefore, we can improve class separation by examining the cumulative sum of the log likelihood score, which, in the worst case, is comparable to simple summation.

$$\text{score} = \max \left(\sum_{row=1}^n O(\bar{x}_{row}), \forall n \leq N \right)$$

V. EVALUATION

To understand the performance of people detection in our application, the original people detection algorithm was evaluated side-by-side with the new algorithm on 3 data sets.

1. Moving people and/or robot.
2. 30 People at different relative rotations - evaluated with different levels of occlusion

3. The guidance scenario of 1000 images, evaluated by several different detection methods

The idea was to evaluate the algorithm with different sensors, lighting conditions, and relative poses.

A. Moving people and/or robot

A total of 14,375 images were collected using the rear facing Kinect sensor on the robot. They were collected over 7 different runs ranging in duration from two minutes to 15 minutes. The two shortest runs involved simply moving around in front of the camera. The remaining runs involved the scenario where a person followed the robot through an environment. All but one run contained at least two people. A total of 6 people were represented in this dataset.

To train the people detection system, a human trainer manually clicked on examples of people and other objects in the image set. Objects were then tracked between successive frames by tabulating a blob similarity score [5] to limit the required input from the trainer. 5736 positive examples, and 9197 negative examples were identified in this fashion.

A 4-fold evaluation was then completed to evaluate how well the classification systems could learn these data. Results are presented for both: (1) row level classification (Figure 4, top), i.e. the ability to classify a single slice of an object as belonging to person or other, and (2) image level classification (Figure 4, bottom), or fused score across all rows. The figures plot precision vs. recall curves for 2 different types of features (polyfit and velodyne), and two different algorithms (AdaBoost, and GMM).

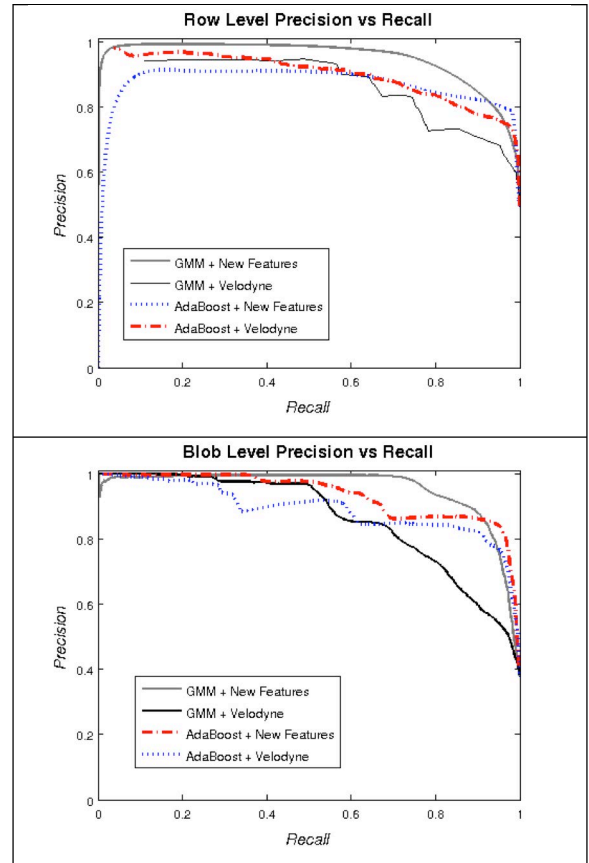


Figure 4. (Top) Row-level precision vs recall for moving person dataset. (Bottom) Blob-level precision vs recall for moving person dataset

Looking first at row level data (Figure 4, top), there are four distinct curves, depending on which algorithm and features that were selected. Using the original AdaBoost algorithm with the smaller polyfit features shows a decrease in performance, compared to the older velodyne set. However, the new polyfit features are better represented by the GMM - maintaining a precision of greater than 95% with recall of 75%. By contrast, the AdaBoost algorithm with the velodyne features falls below 95% precision with recall of only 34%. Using a GMM with the velodyne features improve recall up to 46%, but still has lower performance than the polyfit set out to 82% recall.

Figure 4, bottom, demonstrates the improvements due to fusing row-level scores together. Fusing scores from the GMM with polyfit features raises recall to 72% for a 99% precision (vs 43% at the row-level). A similar fusion using a GMM with the velodyne features also improves classification recall 30% with 99% precision, but the final 37% is much lower than the smaller feature set. A similar cumulative sum approach with AdaBoost is not as successful. It improves performance to only 11% recall for 99% precision.

Where the algorithm failed is also very interesting. As expected, the algorithm struggles with objects that are similarly curved. Chairs are the most common source of false positive detections. Classification depends on the relative angle, and curvature of the chairs. Those which had a curved back presented squarely to the camera were the most problematic, but side views were usually correctly discarded. As chairs often contain a person, however, this is not necessarily a disastrous side effect.

Corners, however, are a more difficult problem. When a robot can see two large sides of a corner, the best shape is not a line but a parabola. Therefore, classification is dependent upon the estimation of curvature, which is a very noisy variable. Because corners remain stationary, our current solution is to assist classification with a prior knowledge of corner locations, but this may or may not always be possible.

B. Detection Study

The previous data set evaluated detection in the presence of motion. People were moving, or the robot was moving, or both were moving through the environment. Because our scenario focuses on guiding people, the majority of people detected were facing the camera - occluded, but still facing the camera. In this second data set we evaluate the effectiveness of detection at different relative orientations to the camera, and different amounts of occlusion. With the ability to move and change its viewpoint, a robot must be able to detect people regardless of the direction they face.

In this experiment, 19 different people (14 men / 5 women) who were not in the previous data set, stood and rotated in place in front of the robot in two different trials (Figure 5). In the first trial, they completed two full rotations with empty arms. In the second trial, they completed two full rotations, while carrying a large box or bag at chest height. The classifier was trained using the data from section V.A, and the detection threshold was selected with desired false positive rate <1%.

With empty arms, the person was correctly detected in 100% of the frames (average of 110 frames / person). To test

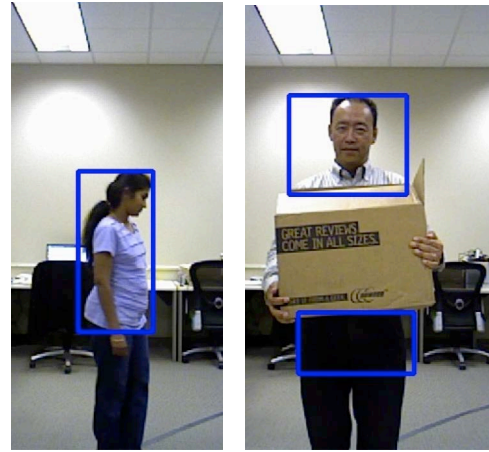


Figure 5. Samples of detected people in the detection study. (Left) Arms empty, rotating in place. (Bottom) Carrying a box, detected two distinct parts to the person.

field of view problems, blobs identified as people were resized, removing a percentage of either rows or columns from the blob, and retested. Figure 6 plots the results.

Removing columns from the image, i.e. a sagittal cut where pixels have been removed from the left edge of the blob, was more challenging than removing rows, i.e. a transverse cut removing lines from top down. For either type of cut, a 40% occlusion still means a recall of greater than 90%. But where the sagittal cut occluding 60% has a recall of only 19%, a comparable transverse cut results in a recall of 79%. This makes sense, as the algorithm is specifically trained on detecting people from rows in the image. By contrast, removing columns only decreases the accuracy of the polynomial fit to the data.

With occlusions to the middle of the body, when people carried a bag or box at chest height, the detection recall was still high (missed 7 of 2475 images). The most common problem, however, was that the head was detected separately from the rest of the body when the person was facing the camera (Figure 6, right). This is a segmentation issue, as the large object and head are not easily connected in the depth image due to significant depth disparity. A secondary detection problem then occurred in correctly classifying legs, without torso or head, as people. This is comparable to a transverse cut of 70% or more in Figure 6.

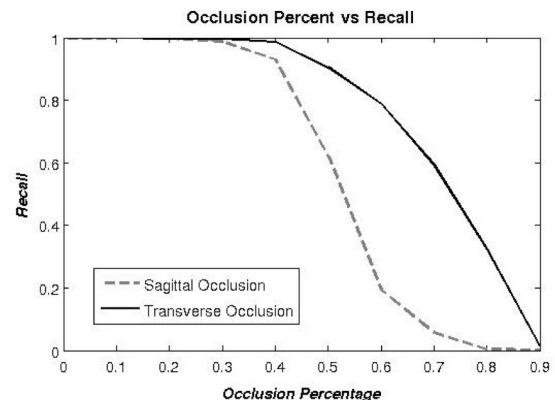


Figure 6. Occlusion percentage vs recall with people rotating in place.

C. Guidance Scenario

In this final evaluation section, the new algorithm is evaluated on the guidance scenario described in section III. Four different person detection methods were evaluated including: face detection using a Haar cascade [4], deformable parts models (DPM) [8], the velodyne features with a weighted linear classifier, and the polyfit features with a GMM. The monocular camera methods are included as a benchmark, given their popularity. However, we found our data set to be insufficient in scope for training these methods, and instead relied on the INRIA data set. The layered detection methods were trained on the data from section 4.1, and the detection threshold was selected to achieve 1% false positive rate. Table 1 summarizes the results.

	False Positive	Low Occl. False Neg. (FN)	Highly Occluded Body, FN	Missing 2nd Person	Speed
Face Detection Haar Cascade	36/1000	365/719 (51%)	171/171 (100%)	133/331 (40%)	2.5 Hz
Deformable Parts Models	56/1000	32/719 (4%)	170/171 (99%)	82/331 (25%)	1 Hz
Velodyne Feat [Boosting]	9/1000	71/719 (9.8%)	45/171 (26.3%)	321/331 (97%)	10 Hz
Polyfit Feat [GMM]	2/1000	0/719	46/171 (27%)	138/331 (42%)	14 Hz

In the guidance scenario, using polyfit features with a GMM, the person was detected in every full frontal frame, an improvement of 4% over the next best method, DPM. It also missed only 27% of the blobs with significant occlusion, compared to virtually all of the highly occluded people missed by DPM. With the second person, the new algorithm struggled with detecting someone leaning on an obstacle. Neither DPM or face detection, by contrast, used the misleading depth information in that case, and were better overall at detecting the 2nd person.

In general, the polyfit features with the GMM model are very strong detection system for this guidance scenario. It is possible that with better training data, DPM could be improved, although [11] suggests improvement will be limited for highly included individuals. But where DPM runs at 1 Hz in our testing, our new system runs, single threaded, at 14 Hz and comparable precision. Even the previous feature set designed for the velodyne runs at only 10 Hz, but has worse performance. Furthermore, the stages are easily separable for multi-threaded processing. Using a flood fill algorithm to extract connected components runs at 83 Hz, while detection runs at 21 Hz per thread. Therefore, the combined system can be boosted to >30Hz or faster than the camera refresh rate with multiple threads.

VI. DISCUSSION

People detection is a fundamental problem in human robot interaction. As usual, however, the inclusion of a robot working around people makes the problem distinct from other, more traditional areas of research. This work focused specifically on detection in guidance scenarios, at ranges where people and robots actually interact and occlusion is very common. We identified common detection issues with

guiding people, and proposed improvements to the layered depth detection algorithm that increase precision and computational speed at this range. Finally, we evaluated the new system performance in a data collection study involving 19 people, and in a real guidance scenario.

In the future, we intend to continue exploring people detection in real environments, extending detection to people carrying objects, and non-upright poses, including people lying down or bent over. The idea, in general, is for detection work in every environment that a person might need to interact with a robot.

REFERENCES

- [1] The Seeing Eye GPS. (2014, Jan.) Sendero Group LLC. [Online]. <http://www.senderogroup.com/support/SEGPSUserGuide.pdf>
- [2] A. Cosgun, E. A. Sisbot, and H. Christensen, "Evaluation of Rotational and Directional Vibration Patterns on a Tactile Belt for Guiding Visually Impaired People," in *IEEE Haptics Symposium*, Houston, 2014.
- [3] W. Yun et al., "Robotic Person-Tracking with Modified Multiple Instance Learning," in *Int. Sym on Robot and Human Interactive Communication*, Gyeongju, Korea, 2013.
- [4] Paul Viola and Michael Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *Proc of the IEEE Conf. on Computer Vision and Pattern Recognition*, Kauai, HI, 2001, pp. 511-518.
- [5] Eric Martinson, Wallace Lawson, and J. Gregory Trafton, "Identifying People with Soft-Biometrics at Fleet Week," in *Proc. of the IEEE/ACM Conf. on Human-Robot Interaction*, Tokyo, Japan, 2013.
- [6] Wongun Choi, Caroline Pantofaru, and Silvio Savarese, "Detecting and Tracking People using an RGB-D Camera via Multiple Detector Fusion," in *ICCV Workshop on Challenges and Opportunities in Robot Perception*, Barcelona, Spain, 2011.
- [7] Navneet Dalal and Bill Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proceedings of Computer Vision and Pattern Recognition*, San Diego, 2005, pp. 886-893.
- [8] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A Discriminatively Trained, Multiscale, Deformable Part Model," in *Proc of IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, 2008.
- [9] Lubomir Bourdev and Jitendra Malik, "Poselets: Body Part Detectors Trained Using 3D Human Pose Annotations," in *Proc. of the International Conference on Computer Vision*, Kyoto, Japan, 2009.
- [10] L. Spinello and K.O. Arras, "People Detection in RGB-D Data," in *Int. Conf. Intelligent Robots and Systems*, San Francisco, CA, 2011.
- [11] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art," *Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743-761, 2012.
- [12] N. Kirchner, A. Alempijevic, and A. Virgona, "Head-to-shoulder signature for person recognition," in *Proc. of the IEEE Conf. on Robotics and Automation*, Saint Paul, MN, 2012, pp. 1226 - 1231.
- [13] Lu Xia, Chia-Chih Chen, and J. K. Aggarwal, "Human Detection Using Depth Information by Kinect," in *CVPR Workshop on Human Activity Detection*, Colorado Springs, Colorado, 2011.
- [14] Frederik Hegger, Nico Hochgeschwender, Gerhard K. Kraetzschmar, and Paul G. Ploeger, "People Detection in 3d Point Clouds Using Local Surface Normals," *Lecture Notes in Computer Science*, vol. 7500, pp. 154-165, 2013.
- [15] L. Spinello, K. Arras, R. Triebel, and R. Siewart, "A Layered Approach to People Detection in 3D Range Data," in *Proc. of the AAAI Conf. on Artificial Intelligence: Physically Grounded AI Track*, Atlanta, GA, 2010.
- [16] L. Spinello, M. Luber, and K. Arras, "Tracking People in 3D Using a Bottom-Up Top-Down detector," in *Int Conf. on Robotics and Automation*, Shanghai, China, 2011.