# Real-Time Multiple Human Perception with Color-Depth Cameras on a Mobile Robot

Hao Zhang, *Student Member, IEEE,* Christopher Reardon, *Student Member, IEEE,* and Lynne E. Parker, *Fellow, IEEE*

***Abstract*—The ability to perceive humans is an essential requirement for safe and efficient human-robot interaction. In real-world applications, the need for a robot to interact in real time with multiple humans in a dynamic, 3-D environment presents a significant challenge. The recent availability of commercial color-depth cameras allow for the creation of a system that makes use of the depth dimension, thus enabling a robot to observe its environment and perceive in the 3-D space. Here we present a system for 3-D multiple human perception in real time from a moving robot equipped with a color-depth camera and a consumer-grade computer. Our approach reduces computation time to achieve real-time performance through a unique combination of new ideas and established techniques. We remove the ground and ceiling planes from the 3-D point cloud input to separate candidate point clusters. We introduce the novel information concept, depth of interest, which we use to identify candidates for detection, and that avoids the computationally expensive scanning-window methods of other approaches. We utilize a cascade of detectors to distinguish humans from objects, in which we make intelligent reuse of intermediary features in successive detectors to improve computation. Because of the high computational cost of some methods, we represent our candidate tracking algorithm with a decision directed acyclic graph, which allows us to use the most computationally intense techniques only where necessary. We detail the successful implementation of our novel approach on a mobile robot and examine its performance in scenarios with real-world challenges, including occlusion, robot motion, nonupright humans, humans leaving and reentering the field of view (i.e., the reidentification challenge), human-object and human-human interaction. We conclude with the observation that the incorporation of the depth information, together with the use of modern techniques in new ways, we are able to create an accurate system for real-time 3-D perception of humans by a mobile robot.**

***Index Terms*—3-D vision, depth of interest, human detection and tracking, human perception, RGB-D camera application.**

## I. Introduction

EFFICIENT and robust detection and tracking of humans in complicated environments is an important challenge in a variety of applications, such as surveillance, human-machine interaction, and robotics. In human-robot teams [1] especially, people can perform key functions; therefore, endowing robots with the ability to detect and track humans is critical to safe operation and efficient robot cooperation with humans. In this paper, we address the task of human detection and tracking in complex, dynamic, indoor environments and in realistic and diverse settings, using a color-depth camera on a mobile robot.

Using a vision system to perceive humans is not an easy task. First, a human's appearance can vary significantly, since humans can be a wide range of sizes, wear different clothes, change poses, face arbitrary directions, and interact with other humans or with objects. Second, a human can be completely or partially occluded by objects, other humans, and even him or herself. Third, visual human perception must deal with common vision problems, such as illumination changes.

Visual human detection and tracking with a moving robot introduces additional challenges to the perception problem. First, a moving camera leads to a dynamic background, for which traditional segmentation-based [2] or motion-based [3] perception approaches are no longer appropriate. Second, a moving robot leads to frequent changes in viewing angles of humans (e.g., front, lateral or rear positions), and causes camera oscillations that introduce additional noise into visual data. Lastly, perceiving humans with a robot adds additional temporal constraints, such as the need to perceive humans and react to human movements as quickly and safely as possible.

This need in our application dictates the definition of real-time performance. Some works in machine vision have identified 4–5 frames per second (FPS) as real time [3], [4]. From a human's perspective, human reaction time is around 0.25 seconds [5], meaning perception greater than 4 FPS is sufficient to detect reactions to a robot's action. Based upon these considerations and the needs of our application, we consider 5 FPS as the minimum frame rate to constitute real-time perception for our system, which allows a robot to behave with similar reaction time to humans.

Although a large number of sophisticated approaches have been proposed to detect and track humans using color cameras [6], they do not make use of one important piece of information that is now available—depth. Since humans act in the 3-D space, depth can be utilized along with color information to develop a more reliable and robust human perception system. Thanks to the emergence of affordable commercial color-depth cameras such as the Microsoft Kinect and Asus Xtion Pro LIVE RGB-D cameras [7], it is now much faster, easier and cheaper to deploy a 3-D vision system on a robot. The additional depth dimension generates more useful information,

Manuscript received September 29, 2012; revised July 16, 2013; accepted July 23, 2013. Date of publication August 21, 2013; date of current version September 11, 2013. This paper was recommended by Associate Editor J. Han.

The authors are with the Distributed Intelligence Laboratory, Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996 USA (e-mail: haozhang@utk.edu; creardon@utk.edu; leparker@utk.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2013.2275291

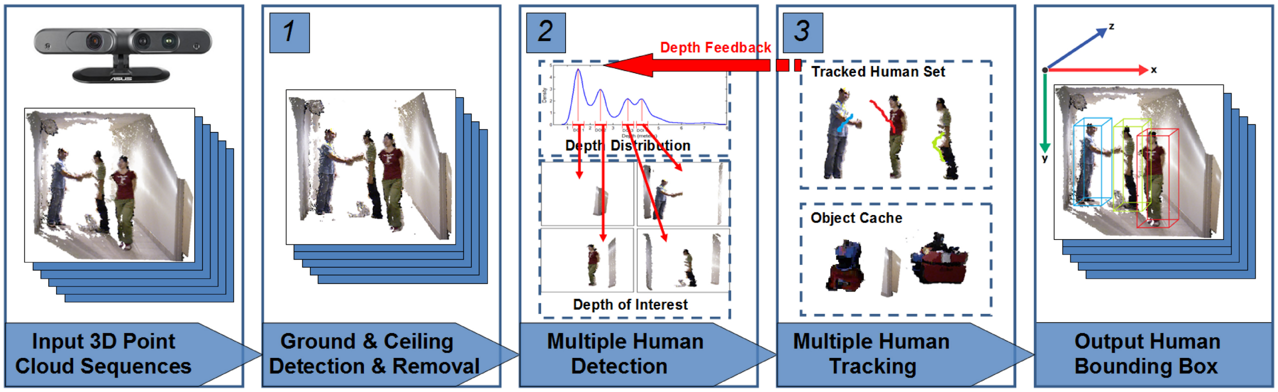


Fig. 1. Description of the major steps in our multiple human detection and tracking system. Starting with the input 3-D point cloud sequences, the system: 1) identifies the ground and ceiling planes, and removes them from the point cloud, 2) applies DOIs along with a cascade of detectors to identify a set of candidates, and 3) associates candidates with tracked humans or detected objects, tracks humans, and feeds depth information back to guide candidate detection. Our system outputs tracking information for each human, such as a 3-D bounding cube and the human's centroid.

such as height and volume, which enables a robot to better observe its environment and localize in the 3-D space.

In this paper, we introduce a new, real-time human perception system to detect and track multiple humans in dynamic indoor environments, using a mobile robot that is equipped with a color-depth camera. Our system creates a new interleaved tracking-by-detection framework. To improve detection performance, the new concept of depth of interest (DOI) is introduced that enables us to efficiently obtain a set of possible human candidates in 3-D point clouds. Then, a cascade of detectors is used to reduce the candidate set by rejecting nonhuman objects. The remaining candidates are handled by a decision process using a directed acyclic graph (DAG) to further distinguish between humans and objects and maintain object detection and human tracking information. Detection and tracking are interleaved in the sense that the tracking model utilizes a fine detector to classify new objects and humans, while the depths of tracked humans are fed back to the detection module to better allocate DOIs.

### A. System Overview

An algorithmic overview of our multiple human perception system is depicted in Fig. 1, which clarifies our methodology by breaking it down into logical blocks. Our system takes 3-D point clouds as input, which are acquired from a color-depth camera mounted on a robot, and outputs human tracking information. The major procedures for human perception are:

1) *Ground and ceiling plane detection and removal*: After the 3-D cloud points are preprocessed, the ground and ceiling planes are detected based on a prior-knowledge guided plane fitting algorithm. Then, all points belonging to the planes are removed from the point cloud.
2) *Multiple human detection*: We first estimate the distribution of depth values in the point clouds and extract DOIs that are likely to contain humans but also may contain objects. Then, a set of candidates is identified by segmenting point clusters within each DOI. Finally, a cascade of detectors is applied to reject as many nonhuman candidates as possible.
3) *Multiple human tracking*: We use a decision DAG-based algorithm to efficiently handle the detected candidates. Candidate association with humans and nonhuman objects is achieved using a two-layer matching algorithm. Then, humans are tracked with extended Kalman filters. The depth values of tracked humans are also fed into the next detection step to guide candidate detection.

### B. Contributions

Our system combines several novel and previously uncombined techniques to create a system that is capable of real-time tracking of multiple human targets and objects from a mobile robot. The contributions of this paper include:

1) The introduction of the new DOI concept for detecting humans in color-depth images, which allows us to avoid using the computationally expensive window scanning over the entire image and speeds up processing to help us achieve real-time performance;
2) The new single-pass, decision DAG-based framework that incorporates human-object classification, data association, and tracking, which allows us to apply the most computationally expensive techniques only to the most difficult cases. This framework saves processing time and further makes our system perform in real time;
3) The use of a detector cascade followed by the decision DAG over 3-D point clouds provides an approach that explicitly addresses the previously unaddressed combination of human-human interaction, human-object interaction, humans assuming nonupright body configurations, and reidentification of tracked humans.

Together, our DOI concept, the use of a cascade of detectors, and our decision DAG-based framework allow us to construct a multiple human perception system that is robust to occlusion and illumination changes and operates in real time, on mobile platforms equipped with standard, consumer-grade computation capability and an RGB-D camera.

The remainder of the paper is organized as follows. Section II overviews literature in the area of human perception. Section III introduces our approaches to ground/ceiling plane removal and detection and tracking of multiple humans in preprocessed 3-D point clouds. Experimental results are presented in Section IV. Finally, Section V concludes this paper.

## II. Related Work

A large number of human detection and tracking methods have been proposed in the past few years. We begin with an overview of approaches using 2-D cameras to detect humans in outdoor environments in Section II-A. Then, Section II-B reviews previous work in human tracking. Finally, 3-D-based human perception approaches are discussed in Section II-C.

### *A. 2-D-Based Pedestrian Detection*

Nearly all state-of-the-art human detectors are dependent on gradient-based features in some form. As a dense version of the SIFT [8] features, histogram of oriented gradients (HOG) was introduced by Dalal and Triggs [9] to perform whole body detection, which has been widely accepted as one of the most useful features to capture edge and local shape information [6]. Other detectors to identify humans in 2-D images include:

1) Shape-based detectors: Wu *et al.* [10] designed the edgelet features, which use a large set of short curve segments to represent local human shapes;
2) Part-based detectors: Bourdev *et al.* [11] developed poselet features, which employ a dictionary of local human parts with similar appearance and pose to represent pedestrians;
3) Motion-based detectors: Dalal *et al.* [12] proposed the histogram of optical flow (HOF) features that apply motions modeled by an optical flow field's internal differences to recognize moving pedestrians. Dollár *et al.* [6] performed a thorough and detailed evaluation and comparison of these 2-D-based detectors.

Pedestrian detectors generally assume that pedestrians are upright, which we do not require to be true in our application; we allow for humans to perform actions with a wide variety of body configurations. In addition, pedestrian detectors typically follow a sliding window paradigm, which applies dense multiscale scanning over the entire image. This paradigm generally has a high computational complexity, and is therefore not suitable for the real-time requirement in our application. This paper addresses real-time human perception tasks using a color-depth camera on a moving platform in indoor environments with a complicated dynamic background.

### *B. Multiple Target Tracking*

Many target tracking approaches [13] from stationary cameras exist that are based on background subtraction [14]. However, in applications with a moving camera, the tracking task becomes considerably harder, as it becomes extremely difficult to subtract the background reliably and efficiently. In these cases, tracking-by-detection appears to be a promising methodology to track multiple objects and is widely used by many state-of-the-art tracking systems [15]. In the tracking-by-detection framework, objects are first detected independently in each frame. After per-frame detection is performed, data are associated across multiple temporal adjacent frames, and targets are typically tracked using classic tracking algorithms, including mean-shift tracking [16] and dynamic Bayesian filters [17], such as Kalman filters [18] and particle filters [19].

Several other approaches have also reported better tracking performance. Okuma *et al.* [20] combined mixture particle filters with AdaBoost, and Cai *et al.* [21] further improved this method by applying independent particle sets to increase multiple tracking robustness. Zhang *et al.* [22] designed a graph-based formulation that allows an efficient global solution in complex situations. Ess *et al.* [23] developed a probabilistic graphical model to integrate different feature modules. To reduce drift, data association can be optimized by considering multiple possible associations over several time steps in multihypothesis tracking [24], or by finding best assignments in each time point to consider all possible associations in joint probabilistic data association filters [25]. Several recently proposed methods also explicitly deal with occlusions. Partial occlusion was addressed by a part-based model [26], and full occlusion was handled with approaches based on tracklet matching [27], visible and occluded part segmentation [28], or an explicit occlusion model [22].

We introduce a new tracking-by-detection framework using a one-pass decision DAG, which is able to run in real time and address previously unaddressed issues, for example, tracking occluded humans who are interacting with other humans or objects.

### *C. 3-D-Based Human Detection and Tracking*

Several human detection and tracking approaches based on 3-D sensing systems have also been discussed, which can be categorized in terms of depth sensing technologies.

1) 3-D lasers: Spinello *et al.* [29] suggested a pedestrian detection system using 3-D laser range data that involves dividing a human into parts with different height levels and learning a classifier for each part.
2) Stereo cameras: A dense stereo vision system [30] was designed to detect pedestrians using HOG features and Support Vector Machine (SVM) classifiers, and a different system was suggested in [31] to use Kalman filters with color features to track moving humans.
3) Time-of-flight cameras: A method using relational depth similarity features [32] was proposed to detect humans by comparing the degree of similarity of depth histograms in local regions, and Xu *et al.* [33] developed a method based on a depth split and merge strategy to detect humans.
4) RGB-D cameras: Salas [34] designed a method that combines appearance-based detection and blob tracking to detect upright pedestrians in an indoor environment with a static background, Xia *et al.* [35] created another human detector by identifying human heads from depth images acquired by a static camera, and Luber *et al.* [36] detected pedestrians indoors using an off-line a priori detector with on-line boosting and tracked humans with a multihypothesis Kalman filter.

The work most closely related to ours was conducted by Choi *et al.* [37], which proposed a particle filter-based method to fuse observations from multiple independent detectors,

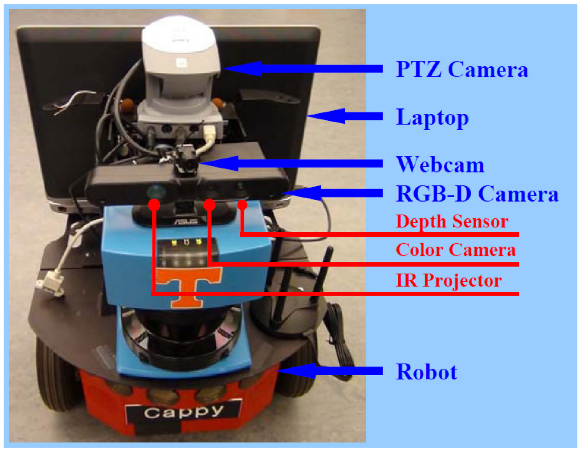


Fig. 2. Installation of Asus Xtion Pro LIVE RGB-D camera on a Pioneer 3-DX robot.

and track humans with a Kinect camera on a mobile robot. However, the detection in this research was based on a sliding window technique over 2-D images, which is highly computationally expensive. In addition, in human-robot teaming, the ability to reidentify humans and discriminate in human-human and human-object interaction scenarios is of significant importance, as robots often must work with a group of coworkers who can repeatedly leave and enter the robot's view and interact with each other and objects. We address all of these issues which were not incorporated in the previous work.

## III. Multiple Human Detection

As discussed above, our objective in this paper is to develop a robust human perception system, with an ultimate goal of allowing a mobile robot to efficiently interact and cooperate with humans in human-robot teaming. Our human perception system is based on the methodology of tracking-by-detection. We begin the discussion by describing 3-D point cloud preprocessing procedures, and a guided sample consensus approach to identify and remove the ground and ceiling planes. Then, we discuss our interleaved tracking-by-detection approach to efficiently track humans in real time. Finally, we describe our system's implementation.

### *A. Camera Calibration and Pixel-Level Preprocessing*

We use an Asus Xtion Pro LIVE color-depth camera to acquire 3-D point clouds. The color-depth camera is installed on top of a Pioneer 3-DX mobile robot, as depicted in Fig. 2. Before acquiring 3-D point cloud data, the color-depth camera must be calibrated to obtain its intrinsic parameters, such as focal distances, distortion coefficients and image centers. Because RGB-D cameras acquire color and depth information separately, the camera must be calibrated to accurately map between depth and color pixels. Then, a 3-D point cloud is formed using the color and depth information. Fig. 3 depicts a 3-D point cloud along with its color and depth images.

The raw color and depth images acquired by the Xtion camera have a resolution of 640×480. To reduce computation costs, each 3-D point cloud is first downsampled to a smaller size by resizing color and depth images to 320×240. The Xtion camera captures depth by projecting infrared (IR) patterns on the scene and measuring their displacement. Due to the limitations of this depth sensing technology, the depth data is very noisy, and contains a significant amount of null or missing values, which can result from the occlusion of the IR camera's point of view or the absorption of the IR light by objects. The points without depth information and the noisy points, i.e. those with few neighbors, are removed from the 3-D point cloud. Then, histogram equalization is applied to the color pixels to remove the effect of sudden intensity changes resulting from the auto white balancing technology.

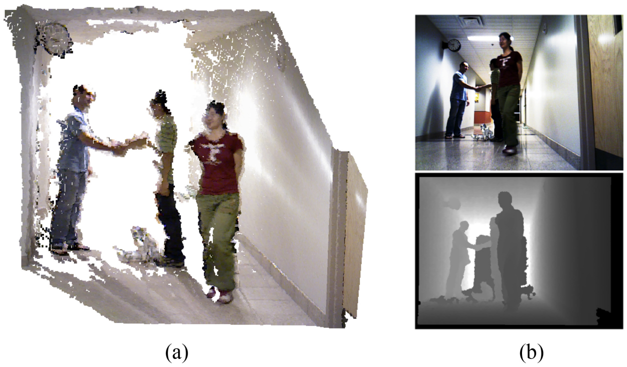


Fig. 3. Example 3-D point cloud with corresponding color and depth images, which are obtained from the RGB-D camera on a mobile robot moving in a hallway. (a) 3-D point cloud. (b) Color and depth images.

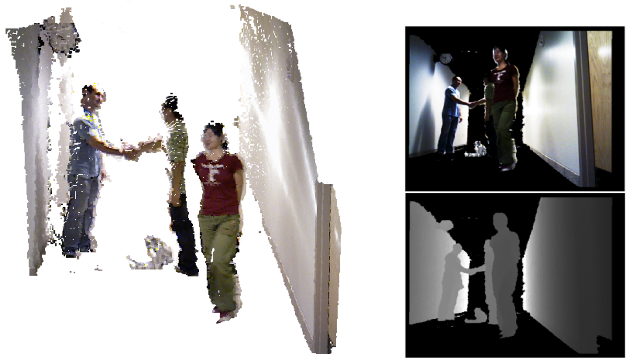
### *B. Ground and Ceiling Plane Removal*

We assume humans and robots exist and operate on the same ground plane, and that a ceiling plane is viewable above them. Since our color-depth camera is installed on a mobile robot at a small tilt angle, these planes generally consist of a significant amount of points that gradually change depth. The points on the ground usually connect objects that are located on the floor. In order to eliminate this connection, ground plane detection and removal is an important operation to separate candidate objects with similar depth values. Using the same technique, the ceiling plane is likewise detected and removed to increase processing speed.

To perform this task, we use a random sample consensus (RANSAC) approach [38], which is an iterative method to estimate the parameters of a mathematical model from a set of observations that contains outliers. We also combine the RANSAC algorithm with our prior knowledge: 1) the ground and ceiling planes should be at the bottom and top of the 3-D point cloud, and 2) each plane's surface norm is a vertical vector. Because the physical oscillations of the moving robot cause slight changes in each plane's location in the 3-D point clouds, the plane's parameters should be reestimated for each point cloud. We also observe that, because there is only a slight change between temporally adjacent point clouds, previous parameters can be used to guide parameter estimation in the current point cloud. Considering this knowledge, we introduce a new extension of the standard RANSAC algorithm, shown in Algorithm 1, that is very robust and efficient.

**Algorithm 1**: Prior-knowledge guided RANSAC

```
Input  : $I_{max}$, $\epsilon$, $\epsilon_{max}$, $X^t$, and $A^{t-1}$
Output: $A^t = [a^t, b^t, c^t, d^t]$
Extract a set of 3-D points belonging to the initial plane:
$C_0 = \{x \in X^t : \mathrm{dis}(x, A^{t-1}) \le \epsilon_{max}\}$;
for $i \leftarrow 1$ to $I_{max}$ do
    Randomly select three points that are not on a line:
    $\{x_1, x_2, x_3\} \in C_{i-1}$;
    Estimate the parameters $A_i^t$ with $\{x_1, x_2, x_3\}$;
    Extract a set of points belonging to the plane:
    $C_i = \{x \in X^t : \mathrm{dis}(x, A_i^t) \le \epsilon\}$;
    if $|C_i| < |C_{i-1}|$ then Set $C_i = C_{i-1}$;
end
Estimate $A^t$ that best fits all points in $C_{I_{max}}$;
return $A^t$
```


(a)

(b)

Fig. 4. Resulting 3-D point cloud with corresponding color and depth images after removing ground and ceiling planes. (a) 3-D point cloud. (b) Color and depth images.

Given distance tolerance $\epsilon$, maximum tolerance $\epsilon_{max}$, maximum iterations $I_{max}$, and the plane's previous parameters $A^{t-1}$ that are estimated from previous 3-D point cloud at $t-1$, Algorithm 1 estimates the parameters of the current plane, that is, $A^t = [a^t, b^t, c^t, d^t]$, from prior knowledge and current observations $X$. The parameter $\epsilon_{max}$ is a predefined maximum distance tolerance used to select search regions of the plane in order to compensate for robot oscillations. Then, all points satisfying $\mathrm{dis}(x, A^t) \le \epsilon$ are defined to belong to the plane, where the distance between a point to the plane in the 3-D space is computed by

$$\mathrm{dis}(x, A) = \frac{|ax + by + cz + d|}{\sqrt{a^2 + b^2 + c^2}}. \tag{1}$$

The initial parameters $A^0$ of the ground and ceiling planes are computed using the robot's geometric information. Then, all points in these planes are removed from the current observation for further processing. As an example, given the input point cloud as shown in Fig. 3, Algorithm 1 is applied to detect the ground and ceiling planes, and the resulting point cloud, with these planes removed, is illustrated in Fig. 4.

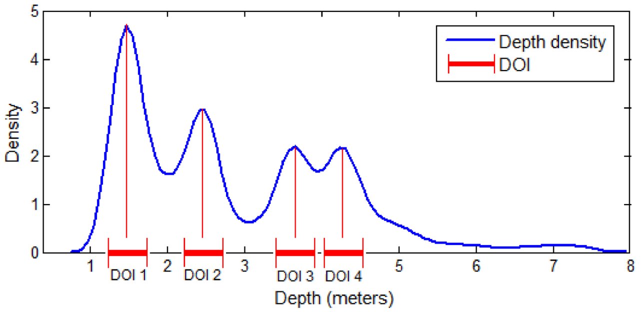


Fig. 5. Depth distribution of the point cloud in Fig. 4(a) with four extracted DOIs. The density is estimated using the Parzen window method with Gaussian kernels.

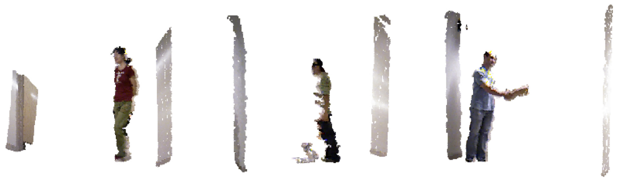

(a) (b) (c) (d)

Fig. 6. Candidates detected from the 3-D point cloud in Fig. 4(a), using the DOIs shown in Fig. 5. (a) DOI 1. (b) DOI 2. (c) DOI 3. (d) DOI 4.

### *C. Candidate Detection*

Our human detection approach is based on a new concept called DOI. Analogous to the concept of region of interest (ROI), which is defined as a highly probable rectangular region of object instances [39], a DOI is defined as a highly probable interval of human or object instances in the 3-D point cloud depth distribution. A DOI is identified by finding a local maximum in the depth distribution and selecting a depth interval centered at that maximum. The correctness of DOI is supported by the observation that any object in a point cloud includes a set of points with similar depth, or several spatially adjacent sets. Each DOI has a high probability to contain objects that we are interested in, which can correspond to humans or nonhuman objects. Since 3-D point clouds captured by color-depth sensors can contain multiple objects located at various depth ranges, the depth distributions of different clouds generally have different shapes with a different number of local maximums. Because the underlying density form is therefore unknown, a nonparametric method is required. To estimate the depth distribution, a 3-D point cloud is first downsampled to a small size (for example, 500 points). Then the nonparametric Parzen window algorithm [40] is applied on the downsampled cloud. Our estimate is based on a Gaussian kernel function [40] with a bandwidth of 0.15 meters, which we tuned through empirical testing. As an example, the estimated depth distribution of the 3-D point cloud in Fig. 4 is depicted in Fig. 5.

To efficiently generate candidates from a 3-D point cloud, the following procedures are conducted in parallel at each DOI.

1) Depth filtering: The 3-D point cloud is filtered along the depth dimension by selecting all points within each

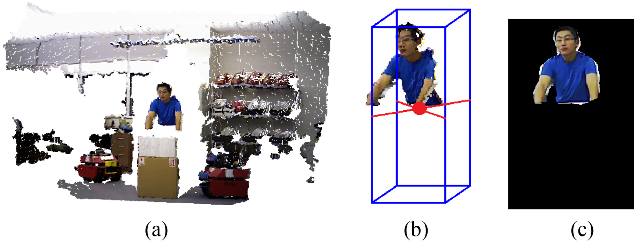


Fig. 7. Computation of the height and centroid of an occluded object. (a) Shows a raw 3-D point cloud. The height of an occluded object is defined as the distance between the highest point to the ground, as shown by the height of the bounding cube in (b). The object centroid is drawn with a red dot in the center of the bounding cube in (b). When the object's point cluster is projected to a 2-D color image of size 96 × 64, the object is placed in the center of the image according to its real size, instead of the blob size, as shown in (c).

DOI, and a depth image is computed from the filtered cloud.

2) Connected component detection: A binary mask is computed from the depth image to indicate whether a depth pixel is within each DOI. Then, connected components are detected using a connectivity of eight. Each connected component is then given a unique index.
3) Candidate generation: Each cluster of 3-D points, whose depth pixels belong to the same connected component, is extracted to form a candidate.

To reduce false negatives, depth values of all currently tracked humans are fed back from the tracking to the detection module. If a depth value being examined does not exist in the current DOIs, a new DOI is created, centered on the depth value, and the candidate generation process above is applied to the new DOI to generate additional candidates. Using this DOI-based candidate generation process drastically reduces the number of candidates and avoids the need to scan the entire cloud, greatly saving processing time in our real-time system.

To preserve the 3-D point clusters that contain only human candidates, a cascade of detectors is used to reject candidates that contain only nonhuman objects. In the detector cascade framework [3], simple detectors are first applied to reject the majority of candidates before more complex detection is performed. A positive result from the first detector triggers the evaluation with a second detector. Cascades of detectors have been shown to greatly increase detection performance by lowering the false positive ratio, while radically reducing computation time [3]. Moreover, the detector cascade can be applied in parallel on each candidate to further reduce computation time. Thus, using a cascade of detectors not only improves the accuracy of our system, but also makes it more able to function in real time. In our system, we use a sequence of heuristic detectors and a HOG-based detector to form a detector cascade in order to reject most of the nonhuman candidates. Our detector cascade includes the following.

1) Height-based detector: The height of a candidate point cluster is defined as the distance between the point with the largest height value and the ground plane, which can be computed using (1). Fig. 7b illustrates the definition of the height feature. A candidate is rejected if its height is smaller than a min-height threshold, or larger than a max-height threshold.
2) Size-based detector: The size of a candidate point cluster can be estimated with: $s(\boldsymbol{d}) = n(\boldsymbol{d})/k(z_{DOI})$, where $n(\boldsymbol{d})$ is the number of points in candidate $\boldsymbol{d}$, $z_{DOI}$ is the average depth value of the DOI that contains $\boldsymbol{d}$, and $k(\cdot)$ is the conversion factor in units of points/m$^2$, which is a function of depth and is used to take into account visual linear perspective, that is, an object contains more points when it gets closer to the camera. A candidate is rejected if its size is greater than a max-size threshold. However, it should be noted that in order to allow for occlusion, our system does not reject small-sized candidates.
3) Surface-normal-based detector: This detector is used to reject planes, such as walls and desk surfaces. Given three randomly selected points in a candidate point cluster: $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\} \in \boldsymbol{d}$, a 3-D surface normal $\boldsymbol{v} = [x, y, z]$ of the candidate can be computed by
$$\boldsymbol{v}(\boldsymbol{d}) = (\boldsymbol{x}_2 - \boldsymbol{x}_1) \times (\boldsymbol{x}_3 - \boldsymbol{x}_1). \tag{2}$$
If $\boldsymbol{v}(\boldsymbol{d})$ is in the $x$-$z$ plane, that is, $y \approx 0$, then the candidate is detected as a vertical plane, for example, a wall. If $\boldsymbol{v}(\boldsymbol{d})$ is along the $y$-coordinate, for example, $x \approx 0$ and $y \approx 0$, then it is detected as a supporting plane, for example, a table or desk top. The surface normal of a candidate is computed multiple times with different points, and majority voting is used for a robust decision.
4) HOG-based detector: The detector applies a linear SVM and the HOG features, as proposed by Dalal and Triggs [9]. Their recommended settings are also used for all parameters except that our detection window has a size of 96×64. The candidate point cluster is projected onto a color image of size 96 × 64 to enable single-scale scanning to save computation. It is desirable that the color image contains the whole candidate, including the parts that are occluded. When a candidate is partially occluded, we set the distance between the candidate's highest pixel and the bottom of the projected color image to be proportional to the candidate's height, as illustrated in Fig. 7. By using a height closer to the actual height rather than the blob height we obtain a more reliable detection result.

The parameters of the heuristic detectors are manually tuned according to empirical observation and prior knowledge, while the HOG-based detector requires a training process to learn model parameters. The candidates that survive the detector cascade are passed to the tracking module. The HOG features of each candidate are also passed to the tracking module, which are used to further distinguish humans and nonhuman objects to allow for more robust and efficient tracking.

### D. Multiple Human Tracking

In human-robot teaming, most tasks, such as human action recognition and navigation among humans, require a robot to perceive human trajectories. In these scenarios, single-frame detection is insufficient and human tracking across multiple consecutive frames becomes essential. In this paper, we implement a decision DAG-based candidate handling algorithm

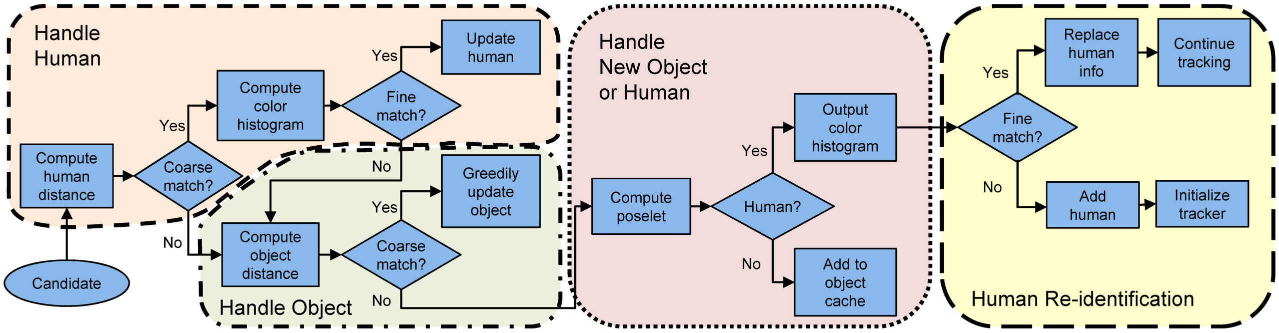


Fig. 8. Illustration of our candidate handling decision DAG that efficiently integrates human-object classification, data association, and multiple target tracking. This framework also simultaneously handles tracked humans, detected objects, and new humans and nonhuman objects, and performs human reidentification.

to simultaneously handle tracked humans, detected nonhuman objects, and new humans and objects, and reidentify humans who enter the camera's view after leaving the camera's view for a period of time, as illustrated in Fig. 8. One key advantage of our decision DAG-based algorithm is that it allows us to divide the types of candidates into separate cases and only apply the most computationally expensive techniques where necessary, thus increasing the speed of our overall system to achieve real-time performance.

1) *Human-Object Classification:* In order to further separate humans and nonhuman objects and explicitly address partial occlusion, the poselet-based human detector [11], a state-of-the-art body part-based detector, is applied in our human-object classification module. Poselets are defined as human parts that are highly clustered in both appearance and configuration space. This detector separately classifies human parts using trained linear SVMs with poselet features, and combines their outputs in a max-margin framework. Although this detector can alleviate the occlusion problem by relying on the unoccluded parts to recognize a human, it is very time consuming to compute poselet features. Because of the time constraints of our real-time system and our desire to use consumer-grade computation hardware, this key disadvantage prevents us from simply applying this technique to every candidate in all point cloud frames. As a result, a poselet-based detector cannot be used as a part of the detector cascade in our candidate detection module, and instead must be used only where most necessary.

In order to use the poselet-based detector most effectively for our real-time system, our decision DAG-based candidate handling algorithm applies this technique only on a subset of candidates. To achieve this goal, we first introduce the object cache, which is defined as the set of nonhuman candidates. Given the object cache, the poselet technique is used only on the candidates that do not match with any tracked humans or nonhuman objects in the cache. Thus, for each object, including both humans and nonhuman objects, application of the poselet-based classification is a one-time procedure, even if the object stays in the robot's view over a long time period, across multiple frames.

The object cache is maintained in the following way. A new candidate in the robot's view, classified by the poselet-based detector as a nonhuman object, is added to the object cache. Alternatively, if a candidate is not new, that is, it matches coarsely with an object in the cache, that object is replaced by the candidate. If an object in the cache does not match any candidate for a period of time, it is removed. It should be noted that no tracking is performed over the nonhuman objects in the object cache, and the object cache evolves by replacing old objects with new ones. The object cache plays an important role in improving the efficiency and accuracy of our tracking module. It not only reduces the number of poselet-based detection procedures to significantly reduce computation time, but also provides negative instances to discriminatively update human models during run-time for robust fine matching of candidates with tracked humans, as discussed next.

2) *Data Association:* This module is applied to match candidates with tracked humans or detected nonhuman objects in the object cache, based on the assumption that at most one candidate is matched with at most one human or detected nonhuman object. Our data association process is divided into coarse and fine matching phases.

a) *Coarse Matching:* Coarse matching between detected candidates and tracked humans is based on position and velocity information. Formally, the Euclidian distance in the 3-D space between a candidate $\boldsymbol{d}$ and a human $\boldsymbol{t}$ is first computed by

$$dis(\boldsymbol{d}, \boldsymbol{t}) = \|(\boldsymbol{c_t} + \dot{\boldsymbol{c}}_t \Delta t) - \boldsymbol{c_d}\| \tag{3}$$

where $\boldsymbol{c_d}$ and $\boldsymbol{c_t}$ are the positions of $\boldsymbol{d}$ and $\boldsymbol{t}$, respectively, $\dot{\boldsymbol{c}}_t$ is the velocity of the human, and $\Delta t$ is the time interval between frames. If this distance is smaller than a predefined threshold $\epsilon_{ct}$, then there is a coarse match between the candidate and human. Because objects in our system are detected but not tracked, their velocity information is not available, so only position information is used to coarsely match a candidate and object. Similarly, if the Euclidian distance $\|c_{\boldsymbol{d}} - c_{\boldsymbol{o}}\| < \epsilon_{co}$, the candidate $\boldsymbol{d}$ and the nonhuman object $\boldsymbol{o}$ are coarsely matched, where $c_{\boldsymbol{o}}$ is the position of the nonhuman object $\boldsymbol{o}$, and $\epsilon_{co}$ is a predetermined distance threshold.

b) *Fine Matching:* Fine matching is applied to further match candidates with tracked humans, and also to reidentify humans when they reenter the camera's view.

We use color information to create an appearance model for each tracked human, which is learned and updated in an online fashion using an online AdaBoost algorithm for feature selection, as proposed by Grabner *et al.* [41]. We train a strong classifier for each human $\boldsymbol{t}$ to determine whether a candidate $\boldsymbol{d}$ matches a human, which is a linear combination of selectors

$$h_{\boldsymbol{t}}^{strong}(\boldsymbol{d}) = \mathrm{sgn}\left(\sum_{i=1}^{N} \alpha_i h_i^{sel}(\boldsymbol{d})\right) \tag{4}$$

where sgn is the signum function, $N$ is the number of selectors to form a strong classifier, and $\alpha$ is the weight for the selector $h^{sel}$, which chooses the weak classifier with the lowest error from a pool of $M$ weak learners. A weak learner $h^{weak}$ represents a feature $f(\boldsymbol{d})$ that is computed on the candidate $\boldsymbol{d}$. A color histogram in the RGB color space is used for our features, and is computed from the candidate's color image that is projected from its 3-D point cluster, as shown in Fig. 7(c). We use nearest neighbor classifiers, with a distance function $D$, as our weak learners

$$h^{weak}(\boldsymbol{d}) = \mathrm{sgn}(D(f(\boldsymbol{d}), \boldsymbol{p}) - D(f(\boldsymbol{d}), \boldsymbol{n})) \tag{5}$$

where $\boldsymbol{p}$ and $\boldsymbol{n}$ are cluster centers for positive and negative instances. The weak learner is updated from a positive and a negative instance in each learning process. Each positive instance to the human-specified classifier $h_{\boldsymbol{t}}^{strong}$ is provided by the tracked human $\boldsymbol{t}$. Each negative instance is randomly sampled from other tracked humans or nonhuman objects in the object cache.

Our fine matching approach has several advantages. First, it creates an adaptive human appearance model that provides a natural way to adapt to human appearance changes caused by occlusions and different body configurations. Moreover, our matching approach is based on a discriminative classification framework, which selects the most discriminative features to distinguish a specific human from other tracked humans and nonhuman objects in a more reliable and robust way. Finally, our color histogram features are an accurate representation of a candidate, since the background is masked out in our color images by applying DOIs, as shown in Fig. 7(c). Together, these advantages improve our system's performance through the reduction of errors.

3) *Extended Kalman Filtering:* Humans in the robot's field of view are tracked locally in our human tracking module, that is, human positions and velocities are tracked relative to the robot. Based on the assumption that humans and robots move smoothly in the global coordinates, humans also move smoothly in the local coordinates. The centroid of each human is tracked in the 3-D space, using the extended Kalman filter (EKF) [42]. EKF is able to track nonlinear movements with a low computational complexity, making it suitable to address nonlinear tracking tasks in real-time applications.

The following procedures for initialization, update and deletion for the EKF process were integrated into our candidate handling framework (Fig. 8).

a) *Initialization:* A new human tracker is created if a candidate is detected as a human that is not currently tracked. However, to address human reidentification tasks, when a nontracked human is detected, instead of immediately initializing a new tracker, the deactivated trackers are first checked to detect whether the human has already been observed. If the human matches a previously tracked subject, the deactivated tracker is reactivated instead, reidentifying the human.

TABLE I

CHARACTERISTICS OF OUR DATASET WITH VARYING DIFFICULTIES CHECK MARKS INDICATE THE CHALLENGE EXISTS

| | Dataset 1 | Dataset 2 | Dataset 3 |
|---|---|---|---|
| Number of samples | 8 | 8 | 4 |
| Frames per sample | 300 | 540 | 1800 |
| Has occlusion | ✓ | ✓ | ✓ |
| Has robot motion | | | ✓ |
| With non-upright human | | ✓ | ✓ |
| With human re-entrance | ✓ | | ✓ |
| With human-object interaction | | ✓ | ✓ |
| With human-human interaction | | | ✓ |

b) *Update:* At each frame, EKF predicts each tracked human's current state, and corrects this estimated state using the observation that is provided by the data association module. Then, the updated estimate is used to predict the state for the next frame. If a tracked human is not associated with any candidate, the tracker is updated with the previous observation.

c) *Termination:* A human tracker instance only persists for a predefined period of time without being associated by any candidates. After this threshold is passed, it is automatically terminated. However, in order to allow for recovering the identity of a human who reenters the camera's field of view after leaving for a short period of time, the trackers are terminated by deactivation instead of deletion.

### E. Implementation

In the candidate detection module, the parameters of the height-based detector are manually set; the min-height threshold is set to 0.4 meters, and the max-height threshold is set to 2.3 meters. The max-size threshold in the size-based detector is set to 3 meters$^2$. Our HOG-based detector is modified from the HOG implementation in [9]. Our detector is trained using bootstrapping. We first train an initial detector with the H3-D dataset [11], using all of the positive and a subset of the negative samples. Then, we apply the initially trained detector on samples of our newly created datasets, as described in Section IV-B, and collect samples leading to false positives and false negatives. Finally, we do a second round of training by including these samples in the training set. In the multiple human tracking module, we use the pretrained poselet-based classifier, which is implemented as described by Bourdev *et al.* [11]. The coarse matching threshold is set to be 1 meter for humans and candidate pairs, and 0.5 meters for object and candidate pairs. When performing fine matching with online AdaBoost, we use $N = 30$ selectors that select color histogram features from a feature pool of size $M = 250$. The EKF termination threshold for human trackers is set to 5 minutes.

## IV. EXPERIMENTAL RESULTS

We performed experiments using our human perception system that is implemented with a mixture of MATLAB and C++ with the PCL library [43], without taking advantage of GPU processing, on a laptop with an Intel i7 2.0GHz CPU (quad core) and 4GB of memory (DDR3). We created a new dataset suitable for the task of multiple human detection and tracking, consisting of 3-D point clouds obtained using an RGB-D camera. Half of the samples in our dataset were used to train the HOG-based detector in a bootstrapping fashion, and half were used to evaluate our system's performance.

### A. Datasets

At the time of this paper's publication, there is no publicly available 3-D human detection and tracking dataset that is collected with an RGB-D camera. Thus, we collected a large-scale dataset to evaluate the performance of our human perception system. Our dataset was recorded with an Asus Xtion Pro LIVE RGB-D camera in an indoor laboratory environment. The camera was installed on a Pioneer 3-DX mobile robot, as illustrated in Fig. 2, and a laptop was mounted on the robot to record 3-D point cloud data. Because the problem of following a target human at an appropriate and safe distance is outside the scope of this paper, the robot was remotely teleoperated by a human, who could only observe the robot's surrounding environment through the robot sensors, that is, the operator could only perceive what the robot perceives. The webcam on top of the RGB-D camera has a similar field of view as the RGB-D camera, which allows the operator to identify and track human subjects without interfering with data recording. The PTZ camera was used to observe behind the robot for safety purposes. The robot's on-board PC was used to control the robot and handle the webcam and PTZ cameras. Although they were needed for conducting experiments, it is noteworthy that the webcam and PTZ cameras do not provide any information to our human perception system, and thus are not pertinent to the essence of this paper.

Our dataset considers three scenarios with increasing difficulties. In Dataset 1, humans act like pedestrians with simple (linear) trajectories. In Dataset 2, humans conduct the task of lifting several humanoid robots and putting them away. In Dataset 3, humans pick up an object, exchange it, and one delivers the object from a laboratory down a hallway to an office room, passing and interacting with other humans on the way. The robot follows the human delivering the object during the entire task. The statistics of our datasets are summarized in Table I, with a breakdown of the increasing difficulty aspects. Each sample in our dataset is a sequence of 3-D point clouds that are saved as PCD [43] files with a frame rate of 30 FPS. Each 3-D point cloud contains 307, 200 points, corresponding to $640 \times 480$ color and depth images, and each point has six values: its coordinates in the 3-D space and RGB values.

To establish ground truth, our dataset is manually annotated using 2-D depth images as follows: First, a representative pixel on a human in a depth image is manually selected to determine the DOI that applies to the human. Using the proper DOI, we mask out the background, leaving the pixels belonging to the same human clustered together as a blob. Then, a bounding box is manually added around each human blob to indicate its $x$ and $y$ coordinates in the depth image. Finally, the bounding box and the DOI are converted to a bounding cube in the 3-D space, which is used as ground truth, and the center of a bounding cube is considered the centroid of a human.

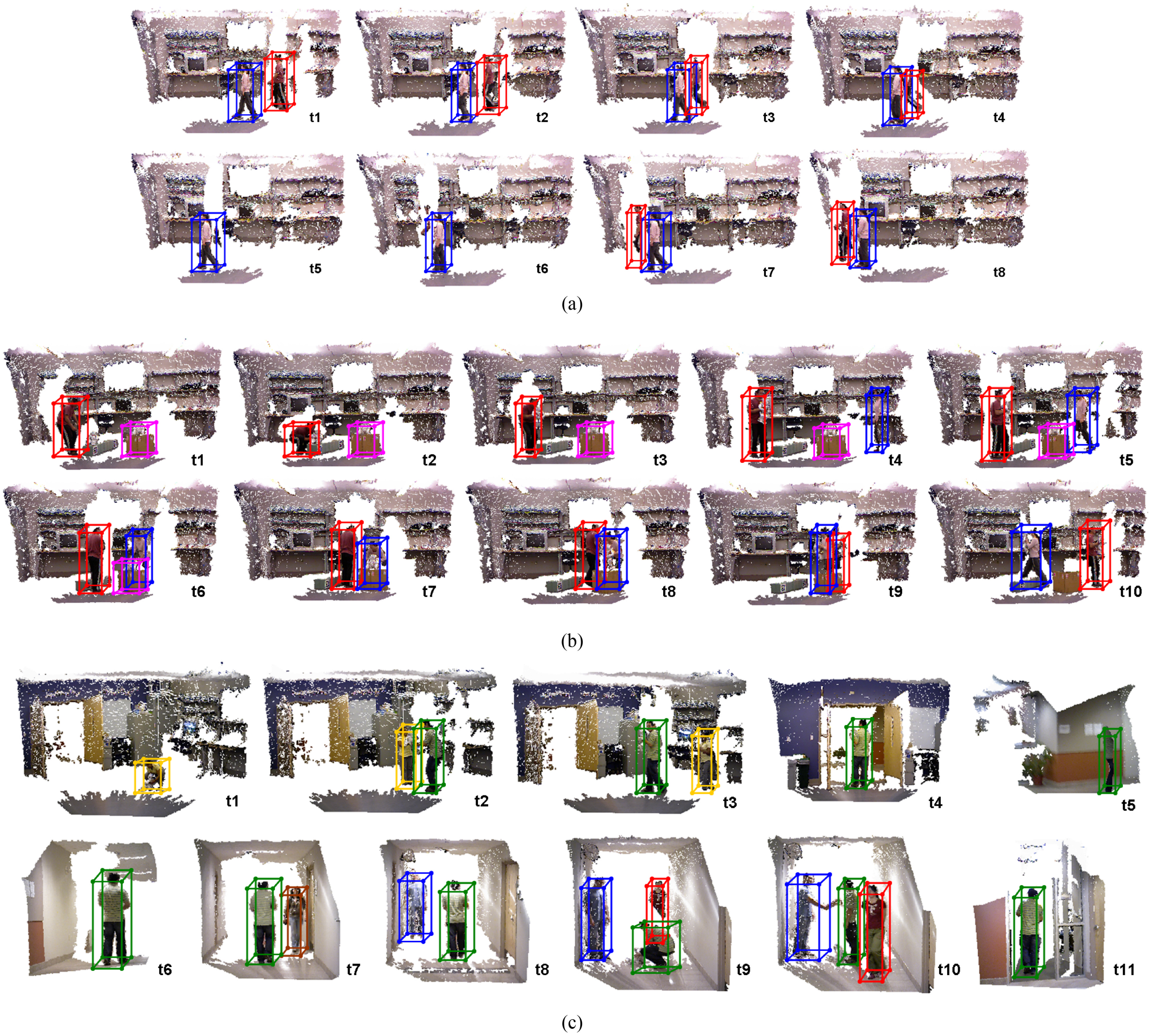
### B. Qualitative Analysis

We first analyze the tracking results from our human perception system to demonstrate its effectiveness and robustness in handling different challenges in human detection and tracking tasks. For each tracked human, a bounding cube with a consistent shape is manually drawn in the 3-D point cloud, according to the cube's vertices that are output by our system. Human identities are represented with different colors, that is, the same human is represented with the same color in a dataset. The tracking results are illustrated in Fig. 9.

**Dataset 1**: Humans act like pedestrians in Dataset 1; they always have an upright pose and generally move with a linear trajectory. It can be observed from Fig. 9(a) that nonoccluded humans in Dataset 1 are detected and tracked perfectly by our system. When a slight partial occlusion occurs, for example, Fig. 9(a) (t4) and Fig. 9(a) (t7), humans are still detected, but the accuracy of the bounding cube might decrease. However, when severe or full occlusion occurs, for example, in Fig. 9(a) (t5), the occluded human cannot be detected, which results in a false negative. Despite the fact that the mostly or fully occluded human cannot be identified, the location of the occluded human's centroid is still updated by the EKF algorithm (for a predefined period of time), using the observation from the previous time point. The advantage of this is that after a human reappears in the camera's field of view, our system is able to coarsely match the human and continue to use the same tracker to track the human, as shown in Fig. 9(a) (t7), which both saves processing time and improves accuracy.

**Dataset 2**: In this dataset, humans move with a complicated but approximately linear trajectory, in which they switch positions as shown in Fig. 9(b) (t7–t9). Our EKF-based tracking algorithm performs well in this situation. Humans also exhibit a variety of body configurations in this dataset, for example, crouching as shown in Fig. 9(b) (t2), and interacting with objects, as illustrated in Fig. 9(b) (t7). In these situations, humans can be detected using our detector cascade along with the poselet-based detector, even with partial occlusions as shown in Fig. 9(b) (t9). In some cases from Dataset 2, a false positive is detected and incorrectly tracked, as indicated by the magenta-colored bounding cube in Fig. 9(b) (t1–t6), which is induced by the human-shape robot sitting on a big box in the center. The other humanoid robot sitting on a small box is not detected, as it is rejected by our height-based detector.

**Dataset 3**: Dataset 3 involves a variety of challenges, as listed in Table I. First, because the robot is moving and humans are tracked in the robot's local coordinate system, human trajectories are no longer linear. We observe that the EKF algorithm still tracks humans with high accuracy in this case, as shown in Fig. 9(c). Second, humans can leave the robot's field of view for a certain period of time. For example, the robot loses the target when the tracked human goes through the door and turns right, as shown in Fig. 9(c) (t5). Our system addresses this problem; when the human reenters the robot's field of view, the human reidentification module, using online human specific appearance models, is activated and continues to track the human with the correct index, as shown in Fig. 9(c) (t6). Third, humans perform very complicated



(a)

(b)

(c)

Fig. 9. Experimental results of the proposed human perception system over our datasets. (a) Dataset 1: Humans move like pedestrians with linear trajectory. (b) Dataset 2: Humans act with complicated body-configurations. (c) Dataset 3: A human performs a delivery task followed and observed by a moving robot.

actions, including human-object and human-human interactions. For instance, a person is passing a humanoid robot to another person in Fig. 9c (t2), and two persons are shaking hands in Fig. 9(c) (t10). In most cases, the interacting humans are separated into different candidates, as illustrated in Figs. 6(c) and (d) for the hand-shaking interaction. However, when interacting humans have very similar depth values (e.g., less than 0.1 meters), they can be incorrectly extracted as a single candidate, which can then be rejected by the size-based detector. This incorrect rejection would result in a false negative.

TABLE II

EVALUATION RESULTS OF OUR 3-D-BASED HUMAN PERCEPTION SYSTEM USING THE CLEAR MOT METRICS

| | MOTP | MOTA | FN | FP | ID-SW |
|---|---|---|---|---|---|
| Dataset 1 | 56 mm | 95.39% | 2.77% | 1.84% | 0 |
| Dataset 2 | 122 mm | 85.48% | 4.27% | 10.25% | 0 |
| Dataset 3 | 83 mm | 94.26% | 3.45% | 1.19% | 0 |

### *C. Quantitative Evaluation*

We follow the CLEAR MOT metrics [44] to quantitatively evaluate the performance of our multiple human perception system, which consists of two scores: multiple object tracking precision (MOTP) and multiple object tracking accuracy (MOTA). The MOTP distance indicates the error between the tracking results and the actual target, and thus reflects the ability of the tracking module to estimate target positions and keep consistent trajectories. The MOTA score combines the errors that are made by the perception system, in terms of false negatives (FN), false positives (FP), and the number of identity switches (ID-SW), into a single accuracy metric. A false negative occurs when a human is annotated in ground truth, but not detected by the perception system. This usually happens for persons that are severely occluded, or on the boundary of the camera's field of view. A false positive occurs when the candidate that is detected as a human does not have a match with any annotated humans in ground truth. In our system, this happens with the nonhuman objects that have a similar height, size, shape and surface property to a human. An identity switch occurs when a tracked human changes its identity. This can happen when a new human enters the scene who is similar in appearance to a human who has just left the robot's field of view, or when two humans with similar appearances switch positions. Our human perception system is evaluated using the metric threshold of 50 cm, as suggested in [44]. The evaluation results are listed in Table II.

Examining our test results, several important observations should be highlighted. First, our human perception system has a very low (perfect) number of ID switches, which is one of the most important properties of our tracking system, since differentiating humans in human-robot teaming applications is essential, especially when, for example, different human coworkers can have distinct preferences and habits. Minimizing ID switch ratio is achieved by combining the following concepts.

1) The background is masked out by the DOI information, which results in a highly accurate human appearance model.
2) An online algorithm is used to continuously update appearance models in real time.
3) Human appearance models are trained discriminatively, which helps maximize the difference between positive and negative instances.
4) The difficult objects that survive the detector cascade are saved in our object cache.

As negative examples to update human appearance models, difficult objects are more representative than other easy objects that are rejected by the cascade of detectors. Second, our system also performs fairly well when localizing targets, in terms of the MOTP scores. We discovered that occlusion usually decreases the object localization ability of our system, and we have greatly relieved this problem by centering a human in a projected 2-D color image according to its real height. Third, we achieve very good results with our most complex and difficult dataset, Dataset 3. One reason for this is that in a large number of frames, there is only one human in the scene without any occlusions. In these cases, the human is detected and tracked perfectly. Finally, our human perception system does not perform as well with Dataset 2 as the other sets, because the humanoid robot on the big box causes a large number of false positives. Moreover, our system has the highest false negative ratio on Dataset 2, due to the fact that humans have the longest occlusion duration.

### D. Comparison to 2-D Baseline

To provide a baseline comparison for the detection aspect of our approach, the most widely used 2-D HOG-based detector [9] was implemented on the same hardware. For this baseline, the detector uses a sliding window paradigm and a sparse scan with 800 windows [9]. For input to the baseline detector, color images were converted from the point cloud data in Datasets 1–3. The baseline detection results are compared in Fig. 10.

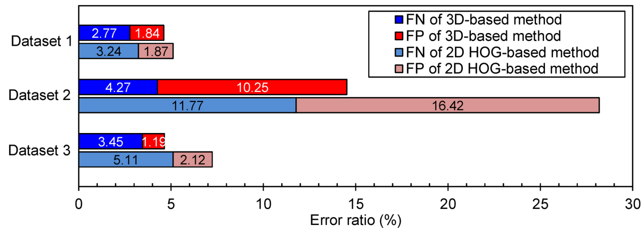


Fig. 10. Comparison of error ratios (i.e., FN and FP) between our 3-D-based approach and 2-D baseline method of [9].

Comparison of the 2-D baseline detector with our detection results shows that the addition of depth information provides a clear increase in accuracy. As discussed in Section III-C, this is because depth information allows for accurate estimation of a candidate's height, size, and surface norm; this heuristic information can be used by our cascade of detectors to greatly reduce false positives by rejecting nonhuman candidates. Depth information also greatly helps to reduce false negatives by feeding DOI information from the tracking module to the detection module to provide assistance locating humans in a new observation. In addition, using the candidate's height helps to detect partially occluded humans, as shown in Fig. 7.

In obtaining the results for accuracy shown above, the 2-D baseline detector yields a frame rate of 0.893 FPS. However, detection is only a part of the entire perception system. The additional tracking step would add additional nontrivial time and further decrease the frame rate of any system into which the baseline detector was incorporated. Because of this, using a 2-D detector such as this baseline in a real-time perception system would be impractical. Because the baseline detector was less accurate and performed so slowly, we did not undertake a comparison between our system and a full system using a 2-D detection component.

In comparison, our complete system, including detection and tracking, achieves a processing rate of 7–15 FPS, which is suitable for real-time applications. Our processing rate is improved using the following techniques.

1) Prior knowledge is used to guide the RANSAC algorithm to efficiently detect ground and ceiling planes.
2) The detector cascade efficiently rejects the majority of the candidates, which can be applied on multiple objects in parallel to further save computation time.
3) Window scanning over entire images is avoided by applying DOIs.
4) HOG features are computed with a single-scale scanning over the projected 2-D color image that contains a candidate blob.
5) Computed features in previous steps are reused in the current step (e.g., the process to compute HOG features reuses a candidate's height and size features, and the process to compute poselet features reuses HOG features).
6) A decision DAG-based candidate handling framework provides a one-pass process that efficiently combines object-human classification, data association, and multiple human tracking.

We observe that a larger number of clusters generally results in more DOIs with more candidates, which typically need more time to process. Therefore, while our experiments were conducted in an academic building and the environments were not manipulated in any way to improve our system's performance, it is certainly possible to conceive of an extremely cluttered environment that would negatively impact computation time.

## V. Summary and Conclusion

In this paper, we presented a system for perceiving multiple humans in three dimensions in real time using a color-depth camera on a mobile robot. Our system consists of multiple, integrated modules, where each module is designed to best reduce computation requirements in order to achieve real-time performance. We remove the ground and ceiling planes from the 3-D point cloud input to disconnect object clusters and reduce the data size. In our approach, we introduce the novel concept of Depth of Interest and use it to identify candidates for detection thereby avoiding the computationally expensive sliding window paradigm of other approaches. To separate humans from objects, we utilize a cascade of detectors in which we intelligently reuse intermediary features in successive detectors to reduce computation costs. We represent our candidate tracking algorithm with a decision DAG, which allows us to apply the most computationally expensive techniques only where necessary to achieve best computational performance. Our novel approach was demonstrated in three scenarios of increasing complexity, with challenges including occlusion, robot motion, nonupright humans, humans leaving and reentering the field of view (that is, the reidentification challenge), human-object and human-human interaction. Evaluation of the system's performance using CLEAR MOT metrics showed both high accuracy and precision. The implementation achieved a processing rate of 7–15 FPS, which is viable for real-time applications. Our results showed that through use of depth information and modern techniques in some new ways, it is possible to use a color-depth camera to create an accurate, robust system of real-time, 3-D perception of multiple humans by a mobile robot.

## References

[1] M. M. Loper, N. P. Koenig, S. H. Chernova, C. V. Jones, and O. C. Jenkins, "Mobile human-robot teaming with environmental tolerance," in *Proc. ACM/IEEE Int. Conf. Human-Robot Interact.*, 2009, pp. 157–164.

[2] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2005, pp. 878–885.

[3] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *Int. J. Comput. Vision*, vol. 63, no. 2, pp. 153–161, Jul. 2005.

[4] P. Dollár, S. Belongie, and P. Perona, "The fastest pedestrian detector in the West," in *Proc. Brit. Mach. Vision Conf.*, 2010, pp. 1–11.

[5] J. T. Eckner, J. S. Kutcher, and J. K. Richardson, "Pilot evaluation of a novel clinical test of reaction time in National Collegiate Athletic Association Division I football players," *J. Athlet. Train.*, vol. 45, no. 4, pp. 327–332, 2010.

[6] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Apr. 2012.

[7] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with Microsoft Kinect sensor: A review," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–1334, Oct. 2013.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2005, pp. 886–893.

[10] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors," *Int. J. Comput. Vision*, vol. 75, no. 2, pp. 247–266, Nov. 2007.

[11] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3-D human pose annotations," in *Proc. Int. Conf. Comput. Vision*, 2009, pp. 1365–1372.

[12] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proc. Eur. Conf. Comput. Vision*, 2006, pp. 428–441.

[13] O. Lanz, "Approximate Bayesian multibody tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1436–1449, Sep. 2006.

[14] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 1999, pp. 246–252.

[15] M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1820–1833, Sep. 2011.

[16] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, Aug. 1995.

[17] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, 1st ed. Boston, MA, USA: The MIT Press, 2009.

[18] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, pp. 35–45, 1960.

[19] M. S. Arulampalam, S. Maskell, and N. Gordon, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[20] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *Proc. Eur. Conf. Comput. Vision*, 2004, pp. 28–39.

[21] Y. Cai and C. Y. Cai, "Robust visual tracking for multiple targets," in *Proc. Eur. Conf. Comput. Vision*, 2006, pp. 107–118.

[22] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multiobject tracking using network flows," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2008, pp. 1–8.

[23] A. Ess, B. Leibe, K. Schindler, and L. van Gool, "Robust multiperson tracking from a mobile platform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1831–1846, Oct. 2009.

[24] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Automat. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.

[25] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE J. Ocean. Eng.*, vol. 8, no. 3, pp. 173–184, Jul. 1983.

[26] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, "Part-based multiple-person tracking with partial occlusion handling," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2012, pp. 1815–1821.

[27] R. Kaucic, A. Amitha Perera, G. Brooksby, J. Kaufhold, and A. Hoogs, "A unified framework for tracking through occlusions and across sensor gaps," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2005, pp. 990–997.

[28] N. Papadakis and A. Bugeau, "Tracking with occlusions via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 144–157, Jan. 2011.

[29] L. Spinello, K. Arras, R. Triebel, and R. Siegwart, "A layered approach to people detection in 3-D range data," in *Proc. AAAI Conf. Artif. Intell.*, 2010.

[30] C. Keller, M. Enzweiler, M. Rohrbach, D. Llorca, C. Schnorr, and D. Gavrila, "The benefits of dense stereo for pedestrian detection," *IEEE Trans. Intell. Transport. Syst.*, vol. 12, no. 4, pp. 1096–1106, Dec. 2011.

[31] R. Muñoz Salinas, E. Aguirre, and M. García-Silvente, "People detection and tracking using stereo vision and color," *J. Image Vision Comput.*, vol. 25, no. 6, pp. 995–1007, Jun. 2007.

[32] S. Ikemura and H. Fujiyoshi, "Real-time human detection using relational depth similarity features," in *Proc. Asian Conf. Comput. vision*, 2011, pp. 25–38.

[33] F. Xu and K. Fujimura, "Human detection using depth and gray images," in *Proc. IEEE Conf. Adv. Video Signal Based Surveillance*, 2003, pp. 115–121.

[34] J. Salas and C. Tomasi, "People detection using color and depth images," in *Proc. Mex. Conf. Pattern Recognit.*, 2011, pp. 127–135.

[35] L. Xia, C.-C. Chen, and J. Aggarwal, "Human detection using depth information by Kinect," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, 2011, pp. 15–22.

[36] M. Luber, L. Spinello, and K. O. Arras, "People tracking in RGB-D data with on-line boosted target models," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2011, pp. 3844–3849.

[37] W. Choi, C. Pantofaru, and S. Savarese, "Detecting and tracking people using an RGB-D camera via multiple detector fusion," in *Proc. IEEE Int. Conf. Comput. Vision Workshops*, 2011, pp. 1076–1083.

[38] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[39] G. Kim and A. Torralba, "Unsupervised detection of regions of interest using iterative link analysis," in *Proc. Neural Inform. Process. Syst.*, 2009, pp. 961–969.

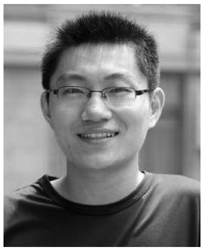
[40] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065–1076, 1962.

[41] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2006, pp. 260–267.

[42] G. Einicke and L. White, "Robust extended Kalman filtering," *IEEE Trans. Signal Process.*, vol. 47, no. 9, pp. 2596–2599, Sep. 1999.

[43] R. Rusu and S. Cousins, "3-D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 1–4.

[44] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *J. Image Video Process.*, vol. 2008, pp. 1:1–1:10, Feb. 2008.



**Hao Zhang** (S'09) received the B.S. degree in electrical engineering from the University of Science and Technology of China, Anhui, China, in 2006, and the M.S. degree in electrical engineering from the Chinese Academy of Sciences, Shanghai, China, in 2009. He is currently pursuing the Ph.D. degree in computer science at the University of Tennessee, Tennessee, USA.

His current research interests include human perception, human activity recognition, human-robot teaming, 3-D machine vision, and machine learning.

**Christopher Reardon** (S'11) received the B.S. degree in computer science from Berry College, Georgia, USA, in 2002, and the M.S. degree in computer science from the University of Tennessee, Tennessee, USA, in 2008. He is currently pursuing the Ph.D. degree in computer science at the University of Tennessee.

He had been a Programmer Analyst at the University of Tennessee for eight years. His current research interests include human-robot interaction, human-robot teams, and machine learning.

**Lynne E. Parker** (S'92–M'95–SM'05–F'10) received the Ph.D. degree in computer science from the Massachusetts Institute of Technology, Boston, USA, in 1994.

She is currently a Professor and Associate Head in the Department of Electrical Engineering and Computer Science, and the Director of the Distributed Intelligence Laboratory, University of Tennessee, Knoxville, Tennessee, USA.

Dr. Parker is the Editor-in-Chief of the Conference Editorial Board of the International Conference on Robotics and Automation, and is an elected Administrative Committee member of the IEEE Robotics and Automation Society. She previously served as the Editor of the IEEE TRANSACTIONS ON ROBOTICS for several years, and is on the Editorial Board of IEEE INTELLIGENT SYSTEMS and the *Swarm Intelligence Journal*. She was a recipient of the PECASE (U. S. Presidential Early Career Award for Scientists and Engineers) in 2000 for her research in multirobot systems.