Detecting and Tracking People using an RGB-D Camera via Multiple Detector Fusion

Wongun Choi *[1] Caroline Pantofaru †[2] Silvio Savarese ‡[1]

[1]Electrical and Computer Engineering, University of Michigan, Ann Arbor, USA
[2]Willow Garage, Menlo Park, CA, USA

Abstract

The goal of personal robotics is to create machines that help us with the tasks of daily living, co-habiting with us in our homes and offices. These robots must interact with people on a daily basis, navigating with and around people, and approaching people to serve them. To enable this co-existence, personal robots must be able to detect and track people in their environment. Excellent progress has been made in the vision community in detecting people outdoors, in surveillance scenarios, in Internet images, or in specific scenarios such as video game play in living rooms. The indoor robot perception problem differs, however, in that the platform is moving, the subjects are frequently occluded or truncated by the field-of-view, there is large scale variation, the subjects take on a wider range of poses than pedestrians, and computation must take place in near real time.

In this paper, we describe a system for detecting and tracking people from image and depth sensors on board a mobile robot. To cope with the challenges of indoor mobile perception, our system combines an ensemble of detectors in a unified framework, is efficient, and has the potential to incorporate multiple sensor inputs. The performance of our algorithm surpasses other approaches on two challenging data sets, including a new robot-based data set.

1. Introduction

The primary mission of a personal robot is to help and co-habit with people. At any time, a robot might have to interact with a person to perform a task such as delivering a beverage or receiving an object to put away. Other tasks are best performed without bothering the human co-habitant, by navigating around a home or an office smoothly, safely and unobtrusively. To function effectively in a human environment, a robot must be able to detect and track people in its vicinity. Perception in indoor human environments can be extremely challenging as the world that the robot "sees" is complex: all of its sensors are mounted on-board a moving platform, the scene is full of clutter, and the dynamic range is very high. Moreover, the people it observes may take on a wide range of poses and motion patterns, be subject to occlusions and self-occlusions, and be truncated by the robot's field of view (Fig. 1). On top of all this, a robot must sense and react to people in near real time to enable interaction.

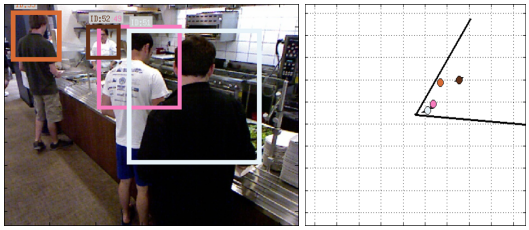

Figure 1. Example tracking results generated by our algorithm. The left panel shows the detected targets in the RGB image. The right panel shows a projection of the 3D location onto the ground plane.

In this paper, we propose a new system for detection and multi-target tracking of people as seen in RGB-depth imagery acquired from an indoor mobile platform, as exemplified in Fig. 1. To cope with the challenges of this task, we propose to integrate multiple complementary sensing modalities and perceptual cues acquired using an ensemble of detectors. This ensemble includes image-based pedestrian and upper body detectors, an image-based face detector, a skin detector, as well as a depth-based shape detector and motion detector. These detection algorithms are fused into a coherent framework using a sampling based method (Reversible Jump Markov Chain Monte Carlo particle filtering) constructed on a tracking-by-detection formulation.

Experimental results show that our approach outperforms competing algorithms in a number of detection and localization tasks as seen in two challenging datasets. These

*wgchoi@umich.edu
†pantofaru@willowgarage.com
‡silvio@eecs.umich.edu

978-1-4673-0063-6/11/$26.00

datasets include RGB-D sequences acquired from a mobile platform, containing imagery of people performing activities in a real office environment and under real-world variation and constraints.

The main contribution of this paper is an algorithm for person detection and tracking which is capable of handling i) people in various poses, ii) occlusion, iii) a mobile platform, iv) scene clutter, and v) real-time performance. In addition, we will contribute the code and data used to evaluate our algorithm.

2. Related Work

There is a long history of approaches to person detection and tracking in the computer vision and robotics literature. Many of these techniques (such as [11, 12, 9, 10]) have been shown to be successful in outdoor scenarios or when scoped to the scenarios for which they were designed. They have been less successful, however, for indoor environments. In outdoor scenes, people are mostly observed in an up-right 'pedestrian' position, whereas in indoor scenes people are often seen in more variable configurations (e.g. sitting on chairs, truncated by the image boundary, or occluded.)

Methods have been proposed to track targets by learning a person specific appearance model given an initial position [4, 8]. These methods work relatively well when the background is not cluttered, but often suffer from the problem of track drift [18] and require manual selection of initial target positions. The improvement in people detection algorithms made it possible to design robust tracking-by-detection algorithms [26, 27, 5, 7, 6, 15]. Wu et al. [27] integrated an image based detection algorithm into a tracking framework. Breitenstein et al. [5] proposed to use the detector confidence value together with the detection output to generate a robust tracking algorithm. Whereas Khan et al. [15] proposed an MCMC particle filtering method to track multiple interacting targets and employed it to analyze the behavior of ants.

Similar to [15], Choi et al. [7] proposed an algorithm for simultaneously tracking multiple targets as well as estimating camera parameters. To make tracking more robust, Wojek et al. [26] explicitly reasoned about the targets' occlusions. Recently, [6] proposed a novel procedure based on maximum weight independent sets to solve the correspondence problem among targets.

In order to improve robustness and accuracy, multiple approaches explore the idea of injecting knowledge about 3D structure of the scene into the tracking process [11, 20, 17]. [11, 20] proposed a system which combines depth information obtained from stereo cameras (or laser scans) and detection responses obtained from an RGB camera. In [3], a 2D lidar scanner is employed to detect and track people by identifying the legs' cylinder shape.

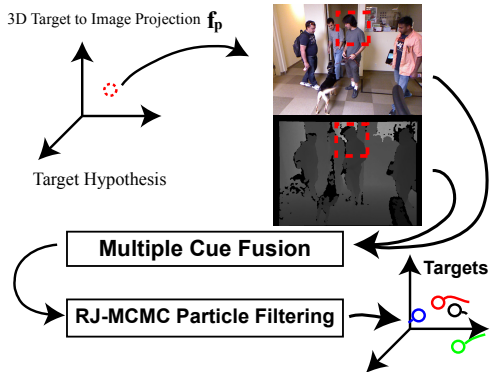

Figure 2. Our algorithm combines RGB and depth image cues to detect and track multiple targets robustly. Given a hypothesis of a valid target location, an ensemble of detectors generates a confidence map (Sec.4) which is then used by Reversible Jump Markov Chain Monte Carlo particle filtering to identify and localize people (Sec.5).

The availability of an affordable RGB-D camera from Microsoft, the Kinect [19], made it possible for everyone to obtain useful depth information. Recently, [17] proposed a pedestrian tracking algorithm using a combination of HOG and HOD features with multiple Kinect cameras. Based on assumption that the person is detected and segmented from the background, Shotton et al. [24] proposed a skeleton tracking method based on Random Forest using Kinect.

The systems presented above, however, do not focus on the indoor, mobile platform perception problem, and hence they do not integrate all of the components necessary to tackle such a task.

3. Model Representation

We tackle this problem within a sequential Bayesian framework. At each time stamp, we obtain the approximate posterior distribution of presence and location in 3D space for a set of targets using an RJ-MCMC algorithm (Sec. 5). Each hypothetical sample is then evaluated using an ensemble of detectors on the projection of the 3D location into the image plane, as in Fig. 2. Below, we describe the Bayesian model observation likelihood and motion model. Sec. 4 describes each of the detectors in the ensemble, and Sec. 5 describes the RJ-MCMC sampling framework.

3.1. Bayesian Model

Given a sequence of color and depth images $I_{1\sim t}$ in time $(1, 2, ..., t)$, our goal is to detect and track people in 3D space. For consistency, we track the top position of each person's head. This can be achieved by finding the maximum a posteri (MAP) solution of the following probabilistic formulation. Let $Z_t = Z_t^0, Z_t^1, ..., Z_t^k$ be a set of targets at time stamp t and Z_t^i be each target's head location parametrized as the 3D point (x, y, z). Following the Sequential Bayesian formulation, the posterior probability of

targets $P(Z_t|I_{1\sim t})$ can be written as:

$$P(Z_t|I_{1\sim t}) \propto \underbrace{P(I_t|Z_t)}_{(a)} \int \underbrace{P(Z_t|Z_{t-1})}_{(b)} \underbrace{P(Z_{t-1}|I_{1\sim t-1})}_{(c)} dZ_{t-1} \tag{1}$$

where (a), (b) and (c) in Eq. 1 represent the *observation likelihood*, the *motion prior* and the *posterior* at time $t-1$, respectively.

Assuming independent motion between targets, we can factorize the overall observation likelihood $P(I_t|Z_t)$ and motion prior $P(Z_t|Z_{t-1})$ as

$$P(I_t|Z_t) = \prod_{i=0}^{M_t} P(I_t|Z_t^i)\ ,\ P(Z_t|Z_{t-1}) = \prod_{i=0}^{M_t} P(Z_t^i|Z_{t-1}^i)$$

where M_t indicates the number of targets at time stamp t.

3.2. Observation Likelihood

Instead of using a single detection algorithm, we incorporate multiple weak detectors to get a strong confidence value about the presence of a target(s) in the scene (see Sec.4 for more details). Assuming independence of the observations, we can rewrite the observation likelihood of a target i ($P(I_t|Z_t^i)$) as follows:

$$P(I_t|Z_t^i) = \prod_{j=0}^{N} P_j(I_t|Z_t^i) \tag{2}$$

where j is the weak detector. The 3D location of Z_t^i does not directly correspond to a point in the color and depth images I_t, so we project Z_t^i into the image plane using the camera parameters.

3.3. Motion Prior

The motion prior of each target encodes two characteristics: existence and smoothness. The former encodes the prior probability of the target's presence at adjacent time stamps. The intuition is that if a target exists at time stamp $t-1$ then it is more likely for this target to exist at time stamp t, and vice versa. The latter enforces smoothness of people's motion in 3D space, e.g. people cannot jump to distant locations or heights in a short time. We model the existence prior by two binomial probabilities P_s and P_e. P_s is the probability of *stay* and P_e is the probability of *entrance*. P_s encodes the likelihood that a target will exist in time t if it exists in time $t-1$, and P_e encodes the probability that a new target will appear in the scene. In practice, we use 0.9 for P_s and 0.1 for P_e. The motion smoothness is modeled as a Gaussian distribution over (x, y, z) centered on the location of the target at $t-1$.

4. Observation Cues

Since the color and depth images contain different information about the scene, combining them results in a more robust detection algorithm. To that end, we combine five different observation models: HOG, shape from depth, frontal face detection, skin and motion detection. In isolation, none of the detectors performs satisfactorily (e.g. the face detector misses people in profile or turned away from the camera, and the HOG detector yields many false positives and missing detections), but combining all of them can generate much more reliable results. Note that our model is flexible enough to handle additional observation models as required to increase robustness.

In this section, we will adopt log likelihood $l(I_t|Z_t^i)$ instead of the likelihood $P(I_t|Z_t^i)$ for simplicity. The entire observation likelihood $P(I_t|Z_t^i)$ can be obtained by taking the exponential of a weighted linear sum of each log likelihood. The following sections describe the detectors in detail.

$$P(I_t|Z_t^i) \propto exp(\sum_j w_j l_j(I_t|Z_t^i)) \tag{3}$$

4.1. Pedestrian and Upper Body Detector

The first cue originates from the distribution of gradients in the image as encoded by the Histogram of Oriented Gradient detector (HOG) [9]. To obtain a detection response, the HOG detector performs a dot product between the model parameter w and the HOG feature h, and thresholds the value (above zero) to find person detections. In this work, we incorporate two HOG detection models, an upper body detector and a full body detector, to cope with i) occlusion of the lower body, ii) different pose configurations and iii) different resolutions of people in images.

Previous work [7] used a Gaussian model centered on positive detections to obtain the observation model. However, such approaches often fail when there are many missed detections or false positives. Inspired by [5], we directly use the detection response to model the observation likelihood from the HOG detector.

$$l_{HOG}(I_t|Z_t^i) = w \cdot h(f_P(Z_t^i)) \tag{4}$$

where f_P is an image projection function, $h(f_P(Z_t^i))$ represents the HOG feature extracted from the image projection of Z_t^i.

4.2. Depth-based Shape Detector

To model the likelihood of people in depth images, we define the log likelihood l_{Shape} as a distance between a shape template and the observed shape of a human. The template is defined for only the head and shoulder region since this is among the most stable body element across various types of common human body configurations. Given a depth image and the location of a person, Z_t^i, a W (width) by H (height) dimensional binary vector is constructed by thresholding the depth image around Z_t^i. Then,

$$l_{Shape}(I_t|Z_t^i) = \tau_s - d(S_{temp}, S(f_P(Z_t^i); I_t)) \tag{5}$$

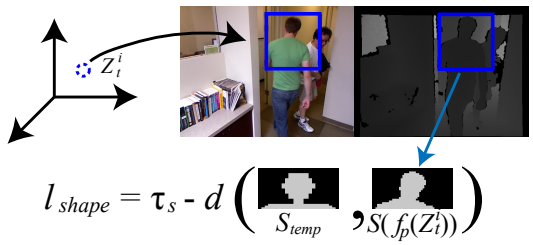

Figure 3. The shape vector is computed from the upper half of the depth image bounding box. The binary vector representing the shape of the person's head and shoulder area is compared to a template using the Hamming distance.

where τ_s is a threshold, S_{temp} is the template, $S(f_P(Z_t^i); I_t)$ is the shape vector of Z_t^i, and $d(\cdot,\cdot)$ is the distance between template and shape vector of Z_t^i. See Fig.3 and Sec.6.1 for details.

4.3. Face Detector

The Viola-Jones face detector [25] as implemented in OpenCV [16, 1] detects people reliably if the image of the face is large enough (greater than 24 pixels) and the frontal side of the face is visible. We incorporate the face detector response as another likelihood measure by calculating the maximum overlap ratio between detection output X_t^k of the face detector and image projection of hypothesis Z_t^i across all K face detections at time t:

$$l_{Face} = \max_k OR(X_t^k, T_f(f_P(Z_t^i))) \tag{6}$$

where T_f is a face cropping transformation and $OR(\cdot,\cdot)$ is the standard overlap ratio between two rectangles (the intersection over the union of the two rectangles).

4.4. Skin Color Detector

The next cue used is skin color. If a person exists in a location Z_t^i, then skin pixels corresponding to the face region are likely to be observed. To detect skin pixels, we threshold each pixel in HSV color space and apply a median filter on the binary skin image I_{Skin}. Given the output, the likelihood is obtained by computing the ratio of skin pixels lying in the face region of a hypothesis:

$$l_{Skin} = \frac{1}{|T_f(f_P(Z_t^i))|} \sum_{(x,y)\in T_f(f_P(Z_t^i))} I_{Skin}(x,y) \tag{7}$$

where $|\cdot|$ represents the area of a bounding box $\cdot$ and I_{Skin} is the filtered binary skin image.

4.5. Motion Detector

Finally, we use motion as an additional cue that implies (with high confidence) the presence of a person in the scene. In order to efficiently identify moving pixels in 3D, we apply an octree-based change detection algorithm [14] to the point clouds at two consecutive time stamps. A binary motion image is obtained by projecting each of the moving points into the image. Similar to the skin detector, the likelihood is obtained by calculating the ratio of moving pixels lying in the body region of a hypothesis.

$$l_{Motion} = \frac{1}{|f_P(Z_t^i)|} \sum_{(x,y)\in f_P(Z_t^i)} I_{Motion}(x,y) \tag{8}$$

where I_{Motion} is the binary motion image.

Note that for many of these cues, such as face detection, skin color detection and motion detection, a positive observation increases the likelihood that a person is present, but the lack of observation does not decrease the likelihood that a person is present.

5. Reversible Jump Markov Chain Monte Carlo Particle Filtering (RJ-MCMC)

Since people can enter and leave a scene at any time stamp, the vector Z_t has variable dimension. To deal with this variable dimensionality and efficiently find the MAP solution over the posterior distribution $P(Z_t|I_{1\sim t})$, we introduce a reversible-jump Markov Chain Monte Carlo (RJ-MCMC) algorithm [15]. The RJ-MCMC automatically detects new targets by using jump moves in different dimensions, e.g. adding new target or deleting existing target by a random proposals.

Computing the posterior $P(Z_{t-1}|I_{1\sim t-1})$ at $t-1$ is intractable, so instead it is approximated using N unweighted samples $P(Z_{t-1}|I_{1\sim t-1}) \approx \{Z_{t-1}^{(r)}\}_{r=1}^N$. Then the posterior $P(Z_t|I_{1\sim t})$ at t can be approximated as follows:

$$P(Z_t|I_{1\sim t}) \propto P(I_t|Z_t) \sum_r P(Z_t|Z_{t-1}^{(r)}) \tag{9}$$

Performing RJ-MCMC sampling on the posterior $P(Z_t|I_{1\sim t})$ results in the posterior distribution for the next time frame. In the following sections, we explain the details of our proposal distribution and acceptance ratio computation for the Metropolis-Hasting algorithm. See [15] for further details of the RJ-MCMC algorithm.

For the remainder of this section, we assume that a weak detection hypothesis X_t and the correspondences between targets and detections are available to guide the sampling procedure. The detections are necessary to help initiate targets and guide the sampling, but our algorithm is able to handle missing detections and false positives.

5.1. Proposal Moves

For MCMC sampling to be successful, it is critical to have a good proposal distribution which can explore the hypothesis space efficiently. Inspired by [15], we define five different reversible jump moves: *Stay*, *Leave*, *Add*, *Delete*,

and *Update*. Each move is designed as a reversible counterpart of another so as to guarantee the detailed balance of the Markov Chain.

Stay: Even if a target does not exist in a current sample $Z_t^{(r)}$, the *Stay* move proposes to keep the target in the new sample $Z_t^{(r+1)}$. Among the targets in set $S_t^{(r)}$ that are not in a sample $Z_t^{(r)}$ but existed in Z_{t-1}, one target i is randomly selected with a uniform probability $\frac{1}{|S_t^{(r)}|}$. Unlike [15] (which samples from only the previous posterior $P(Z_t^i|Z_{t-1}^i)$), we sample the new location of a target from a mixture distribution of $P(Z_t^i|X_t^i)$ and $P(Z_t^i|Z_{t-1}^i)$, where X_t^i is a corresponding detection. This makes the sampling process more robust when a target is moving. If no detection is available for target i, the new proposal is sampled from the previous posterior distribution. The proposal can be written as

$$Q_S(Z_t^{(r+1)}; Z_t^{(r)}) = \begin{cases} \frac{1}{|S_t^{(r)}|} Q_i(Z_t^{i(r+1)}), \ if \ i \in S_t^{(r)} \\ 0, \ otherwise \end{cases} \quad (10)$$

where $Q_i(Z_t^{i(r+1)})$ is equal to $P(Z_t^i|Z_{t-1}^i)$ when there is no corresponding detection, and $\frac{1}{2}[P(Z_t^i|Z_{t-1}^i) + P(Z_t^i|X_t^i)]$, otherwise.

Leave: If a target *Stays* in sample $Z_t^{(r)}$, the *Leave* move proposes to remove the target from the new sample $Z_{t+1}^{(r)}$. This is the reverse jump move of *Stay*. Among the target set $L_t^{(r)}$ that exist in $Z_{t+1}^{(r)}$ and existed in Z_t, one i is randomly selected with a uniform probability $\frac{1}{|L_t^{(r)}|}$ and removed. The proposal can be represented as

$$Q_L(Z_t^{(r+1)}; Z_t^{(r)}) = \begin{cases} \frac{1}{|L_t^{(r)}|}, \ if \ i \in L_t^{(r)} \\ 0, \ otherwise \end{cases} \quad (11)$$

Add: This proposal initiates a new target given the new detections X_t^{new} which do not correspond to any existing targets. Among the detections $A_t^{(r)} = X_t^{new} \backslash Z_t^{(r)}$ that are not in the current target set $Z_t^{(r)}$, one i is randomly selected with a uniform distribution $\frac{1}{|A_t^{(r)}|}$ and the new location of target $Z_t^{i(r)}$ is proposed from a distribution $P(Z_t^{i(r)}|X_t^i)$. The corresponding proposal distribution can be written as

$$Q_A(Z_t^{(r+1)}; Z_t^{(r)}) = \begin{cases} \frac{1}{|A_t^{(r)}|} P(Z_t^{i(r)}|X_t^i), \ if \ i \in A_t^{(r)} \\ 0, \ otherwise \end{cases} \quad (12)$$

Delete: Similar to the *Leave* proposal, *Delete* is the reverse jump move of *Add*. Among new detections $D_t^{(r)} = X_t^{new} \cap Z_t^{(r)}$ that are already added in $Z_t^{(r)}$, one i is randomly drawn with uniform probability $\frac{1}{|D_t^{(r)}|}$ and removed from $Z_t^{(r+1)}$.

$$Q_L(Z_t^{(r+1)}; Z_t^{(r)}) = \begin{cases} \frac{1}{|D_t^{(r)}|}, \ if \ i \in D_t^{(r)} \\ 0, \ otherwise \end{cases} \quad (13)$$

Update: *Update* proposes a new location of a specified target. Among all existing targets in a sample $Z_t^{(r)}$, a single target i is randomly selected and a new location for the sample is proposed from the proposal distribution $Q(Z_t^{i(r+1)}; Z_t^{i(r)}) \sim \mathcal{N}(Z_t^{i(r)}, \Sigma_U)$. Note that one *Update* move can be "reversed" by another *Update* move. The proposal can be expressed as follows:

$$Q_U(Z_t^{(r+1)}; Z_t^{(r)}) = \begin{cases} \frac{1}{|Z_t^{(r)}|} Q(Z_t^{i(r+1)}; Z_t^{i(r)}), \ if \ i \in L_t^{(r)} \\ 0, \ otherwise \end{cases} \quad (14)$$

5.2. Acceptance Ratio

Following the Metropolis Hasting algorithm, we compute the acceptance ratio of the new sample $Z_t^{(r+1)}$ by the product of the three ratios:

$$a = \frac{P(I_t|Z_t^{(r+1)})}{P(I_t|Z_t^{(r)})} \frac{P(Z_t^{(r+1)}|I_{1\sim t-1})}{P(Z_t^{(r)}|I_{1\sim t-1})} \frac{Q(Z_t^r; Z_t^{r+1})}{Q(Z_t^{r+1}; Z_t^r)} \quad (15)$$

The first term expresses the ratio between the image likelihoods, the second term represents the ratio between approximated predictions and the last encodes the ratio between proposal distributions. Since we change the state of only one target's presence or location, most of the factors can be canceled out in the above computation. This characteristic makes the algorithm efficient and capable of processing videos in real-time.

Algorithm 1 RJ-MCMC sampling

Require: Given $X_t, I_t, \{Z_{t-1}^{(r)}\}_{r=1}^N$
  Initialize $Z_t^{(1)}$
  while $r < N$ **do**
    Sample $Z_t' \sim Q(Z_t; Z_t^{(r-1)})$
    Compute a following Eq.15
    $Z_t^{(r+1)} \leftarrow Z_t'$ with probability a
  end while

6. Experimental Evaluation

In this section, we discuss the experimental evaluation of our algorithm, including the details of the actual implementation. Testing was performed on both data collected from on-board a mobile robot (the PR2) moving throughout an office environment, as well as a stationary Kinect sensor placed in a different office environment.

6.1. Implementation details

To begin with, an RGB-D sensor (in our case the Kinect [19]) provides a pair of images, one RGB and one depth. From these images, detection proposals are generated using a combination of HOG detections, face detections and 3D point cloud clusters [23].

The 3D clusters are generated by constructing a 3D point cloud from the depth image and the camera parameters [23]. Given the robot's known 3D base location, 3D points on the floor plane can be reliably removed. The remaining points are clustered using Euclidean clustering [22]. A single proposal for the location of a human head is given by the highest point in a cluster. The proposals from all of the clusters are filtered by height (1.3∼2.3 m) as appropriate to our application and environment. The final remaining proposals are associated with existing tracks using the Hungarian algorithm and 3D distance. This provides initialization for our algorithm and guides the MCMC sampling procedure.

To obtain HOG-based proposals and a confidence map efficiently (sec. 4.1), we use the GPU implementation in OpenCV [1]. The detector can process 640x480 images in 100∼200 milliseconds. The full body detections are obtained from the model trained on the INRIA dataset [9] and upper body detections from the CALVIN model [13]. OpenCV also provides reliable face detections (sec. 4.3) using the Haar feature-based face detection algorithm with a frontal face model [16]. Despite using OpenCV and the GPU, the HOG detector (full and upper body) and face detector are the slowest components in our system, so all three are processed in parallel to the sampling algorithm.

To apply the depth-based upper body template, a hamming distance is computed between the template (fig.3) and the shape in a depth-image window (sec. 4.2). The introduction of a more sophisticated distance such as a weighted distance, or learning a depth template could provide a better estimate, but we leave those as future work.

Skin pixels are found by thresholding the HSV between (2, 60, 40) and (15, 200, 200) (sec. 4.4). Finally, the octree-based motion detector is discretized to 3cm (sec. 4.5).

In the current implementation, the model weights for each detector component are experimentally chosen using validation data since no large dataset of RGB-D data from a moving, indoor platform is available. In future work, we intend to learn the weights.

Sampling is performed as follows. In each frame, we sample 2500 samples with 500 burn-in samples and 50 samples for thinning to reduce correlation. The posterior distribution at each time stamp is approximated by 40 unweighted samples. This sampling takes about 100 milliseconds to process one frame.

Since our algorithm is intended to run on a robot, our implementation is based on the ROS system [2]. ROS is a message passing infrastructure for performing computation in a distributed fashion. ROS also provides standard functionality common to many robot platforms and applications, such as the 'tf' transform library which can be combined with the Kinect's camera calibration to estimate the robot and camera's location relative to a fixed point in the world.

Our implementation can process 5 frames per second on average (using a GPU). The code for the entire system will be posted on-line.

6.2. Dataset

We quantitatively evaluate our algorithm using two challenging, new datasets. The first (static dataset) is acquired in an office scenario with a fixed Kinect camera [19, 21]. The dataset contains 17 videos each spanning 2 to 3 minutes. The Kinect is mounted approximately 2 meters high and tilted downwards to improve the vertical field of view. Videos portray humans under different pose configurations (e.g. sitting on a chair, standing up), observed from different view points (front, side, 3/4) and subject to various degrees of occlusion or self-occlusion.

The second dataset (the on-board dataset) was collected with a Kinect mounted on-board a robot (PR2). The robot drives (tele-operated) around an office building, seeing people in offices, walking in corridors, in the cafeteria, and generally going about their day. Both the robot and the people may be moving. Subject distance from the camera, the number of subjects in a frame, occlusion, lighting (including difficult skylights) and pose varies. This dataset includes 18 segments.

For both datasets, we hand-annotated the people with bounding boxes around upper bodies, 3D locations inferred from the bounding boxes and depth images, and target ID. The annotation is provided for four images per second.

6.3. Results

We compared our algorithm against two baseline methods: the Deformable Parts Model (DPM) [12] full body detector and upper body detector as trained by [13]. The DPM detector is known to be more accurate than the HOG detector, but the implementation of the HOG detector on a GPU is faster. We will integrate the DPM detector into our system in the future.

We report the *log-average miss rate* (LAMR) proposed by [26] as well as miss-rate vs. false-positive-per-image (FPPI) curve. As in [26], LAMR is computed by drawing equally spaced samples in log space of the FPPI. We use two evaluation protocols for identifying true positives. The first is based on the degree of overlap between detected bounding boxes and ground truth bounding boxes(*bounding box overlap*), and the second is a 3D distance threshold for localization evaluation.

As shown in Fig. 4(left), our algorithm significantly outperforms both baseline methods. On the static dataset, the improvement is approximately 13% over both baselines. On the robot dataset, the improvement is 7% over the upper body DPM detector and 20% over the full body detector. This improvement is seen despite using the weaker HOG detector within our system. As hypothesized, the full body detector does not work well in our dataset due to occlusions,

tight field of view, and people who are in non-pedestrian poses such as sitting.

Next, we analyze the contribution of each detector module in our method, with results in Fig. 4(middle). As seen in the plot, the depth shape detector plays the most significant role, with the HOG detector providing the second largest contribution.

When people do not show their face (as happens frequently in the static dataset), the face detector is not very useful. However, when people do show their faces, the face detector is a very strong contributor to overall detection. This fact is somewhat diluted in the analysis in Fig. 4. Motion and skin indicators have similar issues, being very strong in some instances and useless in others. The reasonable performance of the full algorithm as opposed to the variability of the individual detectors is promising.

Finally, we evaluated our algorithm's localization accuracy. In Fig. 4(right), we show the LAMR measure over different 3D distance thresholds. Our algorithm achieves more accurate results when people are within approximately 5 meters of the camera. This is to be expected as the Kinect provides virtually no depth information past 5 meters distance, and in fact the depth information past 3 meters is extremely noisy.

Overall, experiments show that our algorithm outperforms state-of-the-art detectors. In addition, the fusion of multiple detection cues provides a more reliable final result. Our fusion method is capable of handling the variable performance of each individual detector.

7. Conclusion

In this paper, we have introduced an algorithm that can detect and track people in indoor spaces from a mobile platform without instrumenting the environment. As shown in the experimental evaluation, using an ensemble of detection algorithms and fusing their results using RJ-MCMC particle filtering results in robust and accurate person detection. Each detector module has different strengths and weaknesses, focusing on different body components or data characteristics, allowing the overall combination to handle occlusion, motion, truncation, and pose variation.

In future work, we intend to use data-driven training to improve the algorithm's parameters without hand tuning. For example, logistic regression can be used to map detector confidence to better likelihood values. We also intend to learn person-specific detection models to improve data association, especially when people leave the field of view of the camera for short periods of time. Better matching of tracklets will improve overall algorithm robustness and also provide more semantically meaningful track results from which motion patterns can be learned. An extension of this idea is to learn the interactions between people during different activities, like walking together or standing in line. Finally, this work is the first step in performing human pose tracking from a mobile platform in a complex environment.

Acknowledgement We would like to thank Yi-Hsuan Tsai for his help on data collection and annotation. This work is supported by a grant from ONR (award #N000141110389) and NSF AEGER (award #1052762).

References

[1] OpenCV, The Open Source Computer Vision library. http://opencv.willowgarage.com/wiki/. 4, 6

[2] ROS, The Robot Operating System. http://www.ros.org/. 6

[3] K. Arras, S. Grzonka, M. Luber, and W. Burgard. Efficient people tracking in laser range data using a multi-hypothesis leg-tracker with adaptive occlusion probabilities. In *ICRA*, 2008. 2

[4] S. Avidan. Ensemble tracking. In *PAMI*, 2007. 2

[5] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV*, 2009. 2, 3

[6] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *CVPR*, 2011. 2

[7] W. Choi and S. Savarese. Multiple target tracking in world coordinate with single, minimally calibrated camera. In *ECCV*, September 2010. 2, 3

[8] D. Comaniciu and P. Meer. Mean shift : A robust approach toward feature space analysis. In *PAMI*, 2002. 2

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 2, 3, 6

[10] P. Dollar, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. In *BMVC*, 2010. 2

[11] A. Ess, B. Leibe, K. Schindler, and L. van Gool. A mobile vision system for robust multi-person tracking. In *CVPR*, 2008. 2

[12] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9), Sept. 2010. 2, 6

[13] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008. 6

[14] J. Kammerl. Octree point cloud compression in PCL. http://pointclouds.org/news/compressing-point-clouds.html, 2011. 4

[15] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *PAMI*, 2005. 2, 4, 5

[16] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *ICIP*, 2002. 4, 6

[17] M. Luber, L. Spinello, and K. Arras. Learning to detect and track people in rgbd data. In *RGB-D Workshop, RSS*, 2011. 2

[18] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *PAMI*, 26:810–815, 2004. 2

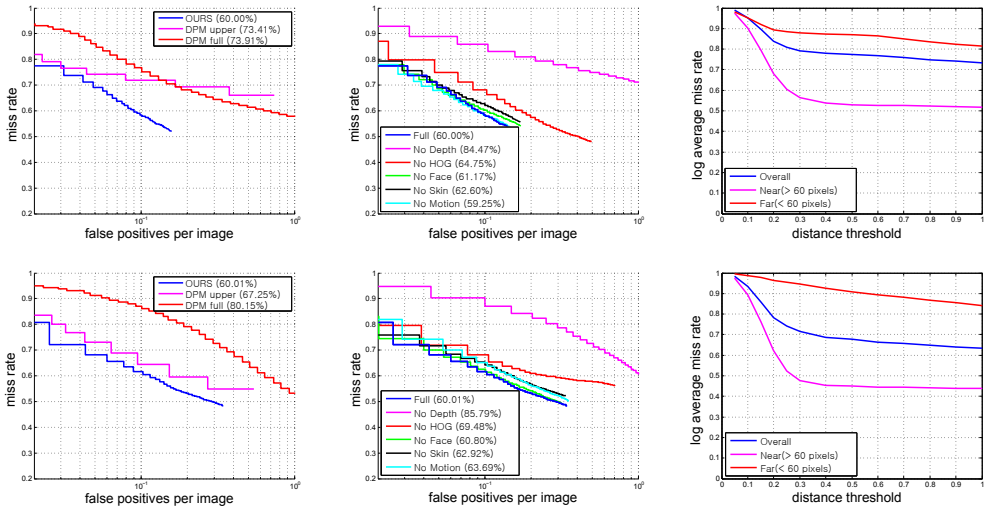

Figure 4. The top row shows results for the stationary Kinect dataset and the bottom row shows results for the on-board Kinect dataset. In both experiments our algorithm outperforms the baseline methods significantly (left panels). In the middle panel, we compare the contribution of each detector. As expected, depth and the HOG detector contribute most to the tracking algorithm. The right panel shows LAMR measure over different distance thresholds. Detections within given threshold distance in the 2D ground plane and within 30 cm in height from ground truth were considered as true positives. Targets whose upper body is larger than 60 pixels in image were considered to be "near" (approximately within 5 meters).

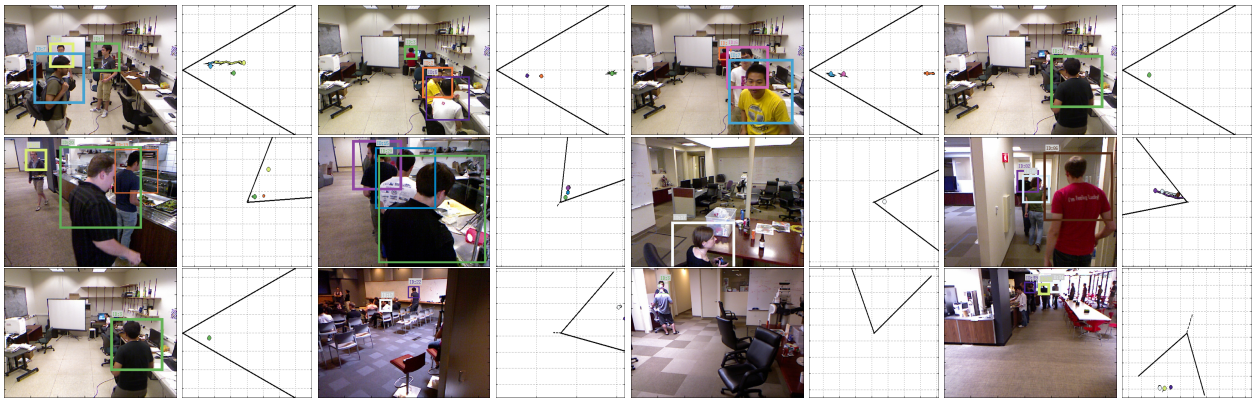

Figure 5. Examples of tracking results. First row: results on the stationary camera dataset. Second row: results on the robot dataset. Detections are shown as bounding boxes in the images, and dots projected onto the ground plane in the top-down view. Each color represents a different target. Notice that our algorithm can detect people in various poses, truncated by the image border, and despite the severe occlusions between people which are common in indoor environments. Pedestrian detectors alone will generally fail in these cases. The last row shows difficult situations in which the people are beyond the Kinect's depth range or under extreme lighting condition.

[19] Microsoft Corp. Kinect for XBOX. http://www.xbox.com/en-US/Kinect. 2, 5, 6

[20] L. Navarro-Serment, C. Mertz, and M. Hebert. Pedestrian detection and tracking using three-dimensional ladar data. *IJRR*, May 2010. 2

[21] PrimeSense. NITE natural interaction middleware. http://www.primesense.com/?p=515. 6

[22] R. B. Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universitaet Muechen, Germany, October 2009. 6

[23] R. B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In *ICRA*, Shanghai, China, May 9-13 2011. 5, 6

[24] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from a single depth image. In *CVPR*, June 2011. 2

[25] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2003. 4

[26] C. Wojek, S. Walk, S. Roth, and B. Schiele. Monocular 3d scene understanding with explicit occlusion reasoning. In *CVPR*, 2011. 2, 6

[27] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. In *IJCV*, 2007. 2