Real Time Person Detection and Tracking by Mobile Robots using RGB-D Images

Duc My Vo, Lixing Jiang and Andreas Zell

***Abstract*— Detecting and tracking humans are key problems for human-robot interaction. In this paper we present an algorithm for mobile robots to detect and track people reliably, even when humans go through different illumination conditions, often change in a wide variety of poses, and are frequently occluded. We have improved the performance of face and upper body detection to quickly find people in each frame. This combination enhances the efficiency of human detection in dealing with partial occlusions and changes in human poses. To cope with the higher challenges of complex changes of human poses and occlusions, we at the same time combine a fast compressive tracker with a Kalman filter to track the detected humans. Experimental results on a challenging database show that our method achieves high performance and can run in real time on mobile robots.**

I. INTRODUCTION

Detecting and tracking multiple humans on mobile robot platforms still remains a challenging task. State-of-the-art algorithms have not yet solved challenging problems of human detection and tracking. First, the mobile robot has often to track the moving humans in a large variability of pose and appearance. Second, the mobile robot frequently has to cope with challenges of full occlusions or self-occlusions. Moreover, due to frequent movement, the mobile robot has often to change the field of view, causing fast changes of the human appearance in each frame. Thus it is not easy for the mobile robot to reliably track people over long periods of time. Eventually, the mobile robot has to interact with many people in real time, resulting in a limitation of computational costs of the tracking system.

Taking inspiration from several state-of-the-art approaches [1], [2], [3], we propose a new algorithm of detecting and tracking multiple people on mobile robots. The first important component is a set of person detectors, helping mobile robots in each frame to reliably find the location of humans and update the human trackers. This set includes the face detector and upper body detector. The face detector is helpful and strongly reliable when the human face is visible and the upper body detector has significant advantages when dealing with the occlusion of the lower body or the face. Due to complicated changes in human pose and appearance, our detectors can not find the position of a person in every frame. Hence, to keep tracking people efficiently, we use a tracking method, based on the combination of a fast compressive tracker and a Kalman filter. This combination enhances the efficiency of our system to adapt to human changes of pose, scales and appearance as well as to partial or full occlusion. In addition, we utilize the depth information from RGB-D images to reduce computational costs and false positives, resulting in a real time performance of human detection and tracking on the mobile robot.

D. M. Vo and L. Jiang are with the Chair of Cognitive Systems, headed by Prof. A. Zell, Computer Science Department, University of Tübingen, Sand 1, D-72076 Tübingen, Germany {duc-my.vo, andreas.zell}@uni-tuebingen.de

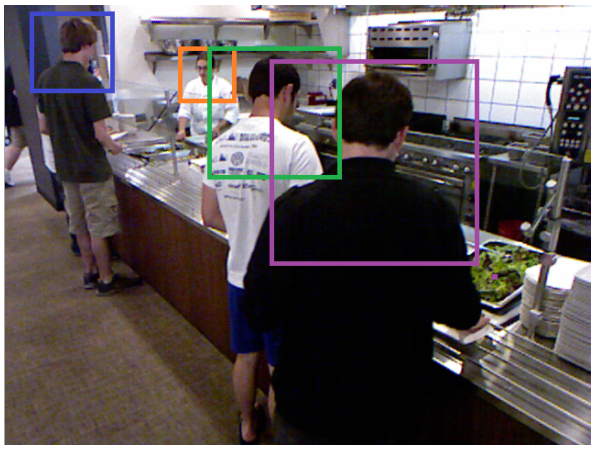

Fig. 1. Example tracking result of our algorithm.

The remaining parts of this paper are organized as follows: in Section II we mention the state-of-the-art algorithms of face recognition which motivated our research. In Section III, our method is presented in detail. In Section IV the experimental results obtained from databases are described. We conclude this paper in Section V, mentioning our intentions with regard to our future work.

II. RELATED WORK

Although there are many approaches to tracking multiple humans by mobile robots, such as sample-based joint probabilistic data association filters [4], and Kalman filters [5], most of them have not been successful to adapt to human changes of pose, scales and appearance as well as to partial or full occlusions. State-of-the-art algorithms of human detection [6], [7], make a great contribution to tracking-by-detection approaches [8], [9], thus significantly improving the tracking system of mobile robots. Choi et al. [1] proposed a method of detecting and tracking people by mobile robots, based on the algorithm of Reversible Jump Markov Chain Monte Carlo Particle Filtering (RJ-MCMC). Due to detecting humans based on relatively reliabale observation cues of humans in each frame, this method was shown to be robust to complicated changes of human poses and partial occlusions. These observation cues include a

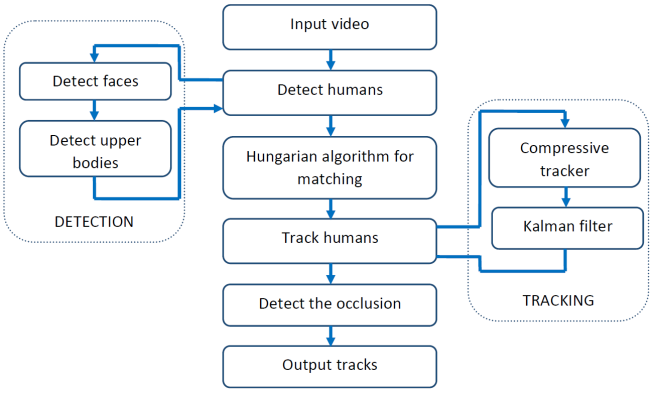

Fig. 2. Flow chart of our approach.

human detector using a Histogram of Orientations [7], a face detector using the Viola-Jones method of objection detection [6], and the detectors of skin, motion and depth-based shape. However, the computational costs of the detectors and the tracking algorithm of Reversible Jump Markov Chain Monte Carlo Particle Filtering are very expensive. For human-robot interaction, the computational complexity of this algorithm has not met the requirement of real time performance.

Other promising approaches for human tracking are online learning methods to handle the complex appearance variation of human poses. Some examples of these algorithms include incremental learning [10], online multiple instance learning [11] and visual tracking using $L1$ minimization [12]. To deal with the appearance change of the object and its partial occlusion, Zhang et al. [3] proposed the method of compressive tracking. This method uses compressed features, extracted from the tracked object, to online update a simple Bayes classifier. As a result, this classifier is able to quickly adapt to the object changes of pose, rotation, deformation, and self-occlusion. In addition, this method is suitable for real time applications due to its low computational costs. Since the mobile robot and humans often move and change their directions and orientations, an effective improvement of the compressive tracker can be a good solution to adapt to all these changes and reliably track humans.

III. APPROACH

Figure 2 illustrates an overview of the proposed person detection and tracking framework for mobile robots. Human detection is applied in each new frame. The detection module is comprised of a face detector and an upper body detector. In order to meet the requirement of real time mobile robot performance, one detector is used in the current frame and the other one is used in the next frame, and so on.

In the stage of face detection, the technique of depth-based skin color segmentation is provided to speed up the search of the face and reduce the false positive rate. Since the search areas in each frame are reduced significantly, the Viola-Jones method of face detection [6] is implemented to detect humans quickly and reliably.

In the stage of upper body detection, an upper body detector is trained on the CALVIN dataset [2]. Since searching for

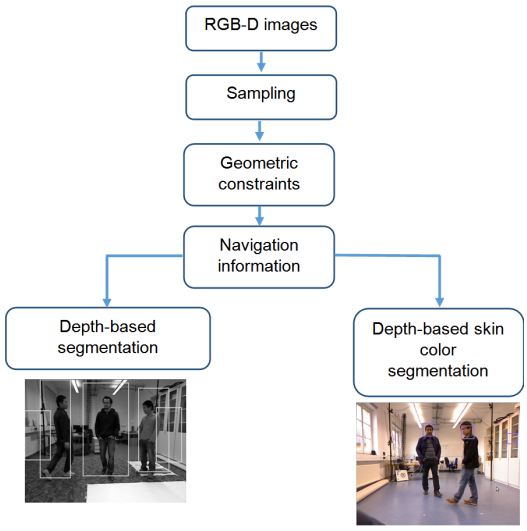

Fig. 3. Flowchart of segmentation steps for upper body and face detection.

humans in the whole image is a time consuming operation, we decrease search areas in each image and estimate human scales necessary to search in those areas. For this reason, the depth information is utilized to segment potential areas where humans probably appear, and skip non-human areas in each frame. As a result, the trained upper body detector can in real time quickly find the humans in images.

Our tracking method is based on a fast compressive tracker and a Kalman filter. The new position of our tracker is taken either from the output of the fast compressive tracker or from the predicted position of the Kalman filter and it depends on whether large occlusion regions are found in the current frame or not. If a complete occlusion is found, the Kalman filter plays an important role to predict the next position of the temporally occluded human. If no significant occlusion is recognized, the fast compressive tracker provides a more accurately predicted position than the Kalman filter. In our research, the depth information has proven to be useful for detecting occlusions.

A. Face detection

As mentioned in our previous work [13], if the image of the frontal face is visible we apply our face detector to quickly and reliably detect people. As shown in Figure 3, the information of geometric constraints, navigation and the technique of depth-based skin color segmentation are provided to make our face detector much faster and more accurate. Our face detection involves three basic steps: First, in order to reduce computational costs we use a set of sampling points spanning the whole image to collect the information of color, texture and depth. Second, the constraints of geometry and navigation information are used to remove the background. Finally, the techniques of skin detection and depth-based skin colour segmentation are applied around filtered sampling points to find the potential regions in which the face detector is able to localize the face position. In addition, we can speed up face detection by limiting the

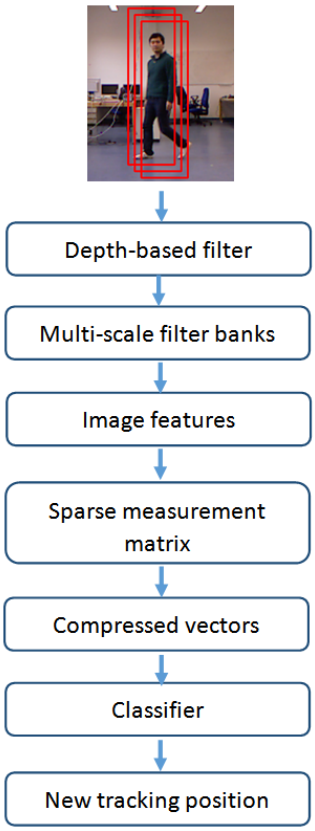

Fig. 4. Tracking steps of a fast compressive tracker.

range of facial scales, which is mentioned in [13] and thus estimate the sizes of the humans that are possible present in these regions.

B. Upper body detection

Similar to the above step of face detection, the information of geometric constraints, navigation and the technique of depth-based segmentation are helpful for removing the background and reducing search areas, as shown in Figure 3. As a result, we have a small set of search areas where the upper body detector is applied for to detect humans, based on Histograms of Oriented Gradients [7]. Basically, similar to what we do in face detection, we estimate the sizes of humans in these search areas in order to significantly reduce computational costs.

The upper body detector is trained by using a linear SVM. Particularly, search windows are divided into cells which are used to compute a Histogram of Oriented Gradients. The upper body detector classifies the search window running through every position and scale to find the human location.

C. Fast compressive tracking

If large changes in the appearance of humans by illumination, different poses or by partial occlusion exist, the data association temporally fails. When those failures happen, the fast compressive tracker plays a very important role, following the human and adapting to those complex changes as well as to partial occlusion.

To keep tracking the human, the fast tracker uses a search window, which is updated by each corresponding detection, as shown in Figure 4. First, we collect a set of image samples near the current human location in the search window. Then we estimate the distance between the human, appearing in the search window, and the camera by sampling the depth information in this search window. Similar to the segmentation step presented in our previous work [13], we use a set of sampling points spanning the whole search window to collect the information on depth. The technique of depth-based segmentation is applied around sampling points to find the human, which is the biggest segmented region in the search window. The distance between the human and the camera is estimated based on the average depth value of the sampling points belonging to the segmented region. In order to filter out samples, the above technique of depth-based segmentation is used for all samples to segment objects in each of these samples. Because a distance estimation between the human and the camera exists, a sample can be filtered out if no segmented object is in the range closer than 0.5 meters from the tracking human location.

After reducing a large amount of negative samples, the remaining samples are kept for classification. Each filtered sample $z \in \mathbb{R}^{w \times h}$ is convolved with multiple-scale filters $h_{1,1}, ..., h_{w,h}$ computed as follows

$$h_{i,j}(x,y) = \begin{cases} 1, & 1 \leqslant x \leqslant i, 1 \leq y \leq j \\ 0, & otherwise \end{cases} \quad (1)$$

where i and j are the sizes of a filtered image. These images are concatenated as a feature vectors $x = (x_1, ...x_m)^T \in \mathbb{R}^m$ where $m = (wh)^2$.

Since this feature vector x is very high dimensional, we use a random projection to transform $x \in \mathbb{R}^m$ into a lower dimensional space $v \in \mathbb{R}^n$

$$v = Rx \quad (2)$$

where $R \in R^{n \times m}$ is a random projection matrix. The elements of this random projection matrix are defined as

$$r_{ij} = \sqrt{s} \times \begin{cases} 1, & with\, probability\, \frac{1}{2s} \\ 0, & with\, probability\, 1 - \frac{1}{s} \\ -1, & with\, probability\, \frac{1}{2s} \end{cases} \quad (3)$$

where $s = m/4$.

By using a naive Bayes classifier for each feature vector $v \in \mathbb{R}^n$, we can find the new position of the tracking human in the current frame, corresponding to the maximal response of this classifier. All elements in v are modeled with a Bayes classifier as follows:

$$H(v) = log\left(\frac{\prod_n^{i=1} p(v_i|y=1)p(y=1)}{\prod_n^{i=1} p(v_i|y=0)p(y=0)}\right) = \sum_{i=1}^{n} log\left(\frac{p(v_i|y=1)}{p(v_i|y=0)}\right) \quad (4)$$

where $p(y=1) = p(y=0)$, and $y \in \{0,1\}$ is a binary variable representing the sample label.

After using the classifier H in (4) to find the tracking location, the classifier H is updated to adapt to human changes of rotation, occlusion and scale as well as to adapt to complex

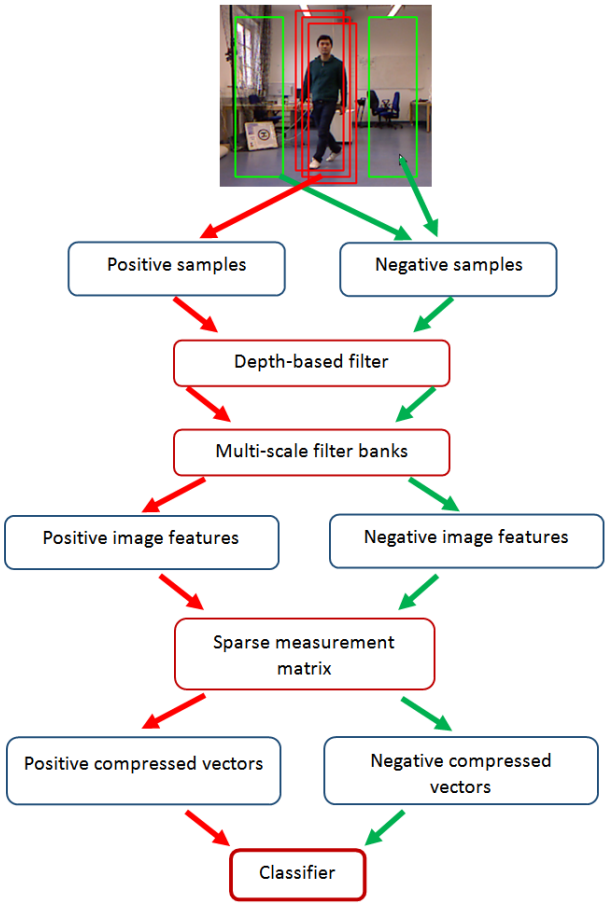

Fig. 5. Update steps of a fast compressive tracker.

changes of background and illumination. For updating, we collect a set of possitive samples near the current center of the search window and a set of negative samples far away from this position, as shown in Figure 5. Similar to the step of classification, low dimensional features $v \in \mathbb{R}^n$ are extracted from these two sets of samples following the steps of depth-based filtering, multi-scale filter banks, and random projection. We use these features to update the classifier parameters. Since the conditional distributions $p(v_i \mid y = 1)$ and $p(v_i \mid y = 0)$ are Gaussian distributions with

$$p(v_i \mid y = 1) \sim N(\mu_i^1, \delta_i^1), \quad p(v_i \mid y = 0) \sim N(\mu_i^0, \delta_i^0) \tag{5}$$

we have to update the parameters μ_i^1, δ_i^1, μ_i^0 and δ_i^0 in the classifier H. These parameters are updated online as follows

$$\begin{aligned} \mu_i^1 &\leftarrow \lambda\mu_i^1 + (1-\lambda)\mu^1 \\ \sigma_i^1 &\leftarrow \sqrt{\lambda(\sigma_i^1)^2 + (1-\lambda)(\sigma^1)^2 + \lambda(1-\lambda)(\mu_i^1 - \mu^1)^2} \end{aligned} \tag{6}$$

where $\sigma^1 = \sqrt{\frac{1}{n}\sum_{k=0|y=1}^{n-1}(v_i(k) - \mu^1)^2}$ and $\mu^1 = \frac{1}{n}\sum_{k=0|y=1}^{n-1} v_i(k)$, and λ is a learning parameter.

D. Kalman filter for occlusion handling

When a new fast compressive tracker is initiated, a Kalman filter is also set up as an alternative tracker in case that the human is completely occluded by another person or large objects. That means that the output of the Kalman filter is used for tracking when the human is significantly occluded and the fast compressive tracker can not provide a reliable prediction.

A Kalman filter consists of measurement update equations and time update equations. When the compressive tracker is still tracking the human efficiently without recognized occlusion, the measurement update equations correct the Kalman filter by using the significantly reliable output from the fast compressive tracker. The time update equations are used to predict the current position of the human, and this prediction only replaces the one from the compressive tracker when an occlusion is found. The Kalman filter state vector includes five parameters which are x-y coordinates of the bounding box of the human region, the velocity in the x and y directions, and the scale of the human region. The state-space representation of the Kalman filter is given as

$$\begin{bmatrix} \hat{x}_t \\ \hat{y}_t \\ \hat{x'}_t \\ \hat{y'}_t \\ \hat{s}_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta_t & 0 & 0 \\ 0 & 1 & 0 & \Delta_t & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ x'_{t-1} \\ y'_{t-1} \\ s_{t-1} \end{bmatrix} + W_t \tag{7}$$

where $\hat{x}_t$, $\hat{y}_t$ are the coordinates, and $\hat{x'}_t$, $\hat{y'}_t$ are the velocities, $\hat{s}_t$ is the scale of the human region. Δ_t is defined as the time interval and W_t is the measurement noise. In order to update the Kalman filter, the output of the Naive Bayes classifier in the fast compressive tracker is given as the measurement input. The Kalman filter uses this data to effectively correct the system. The measurement correction equation is represented as follow

$$\begin{bmatrix} x_t \\ y_t \\ x'_t \\ y'_t \\ s_t \end{bmatrix} = \begin{bmatrix} \hat{x}_t \\ \hat{y}_t \\ \hat{x'}_t \\ \hat{y'}_t \\ \hat{s}_t \end{bmatrix} + K_t \left(\begin{bmatrix} mx_t \\ my_t \\ mS_t \end{bmatrix} - \begin{bmatrix} 1 & 0 & \Delta_t & 0 & 0 \\ 0 & 1 & 0 & \Delta_t & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_t \\ \hat{y}_t \\ \hat{x'}_t \\ \hat{y'}_t \\ \hat{s}_t \end{bmatrix} \right) \tag{8}$$

where K_t is the Kalman factor, mx_t, my_t and ms_t are the measurement variables computed from the position and size of the human assessed by the compressive tracker.

The prediction of the Kalman filter is required whenever the human location can not be found by the compressive tracker due to occlusion by other persons or objects. When the human is significantly occluded, the fast compressive tracker can filter out all samples since no segmented object closer than 0.5 meters to the tracking human location can be found. For this situation, the Kalman filter is considered in the short time interval as the alternative solution to continue to track the occluded human. After t_o frames, if a new detection is matched to the tracker, the fast compressive tracker is recovered at the new detected position. Otherwise, if we can not find any detection matching the Kalman filter during t_k frames, the target is automatically terminated.

E. Hungarian algorithm for matching

When we find a new detection of a human we use the Hungarian algorithm to search the tracker corresponding to

this detection. If it is not matching any available tracker, a new tracker is initiated on the new detected position. If the Hungarian algorithm finds the corresponding tracker, this tracker is updated by the new detected position. On the other hand, if no detections are found for the same tracker during a period of termination, this tracker is automatically terminated.

The Hungarian algorithm is based on the cost values corresponding to the overlap ratio between valid targets and new detections in each frame. In order to compute the cost for each pair of a target and a detection, we based the formula on the overlap ratio between them, as follows

$$R_i^k = \frac{2 * s_{ik}^O}{s_i^D + s_k^T} \tag{9}$$

where s_{ik}^O is the overlap area, s_i^D is the area of the i^{th} detection and s_k^T is the area of the k^{th} target. The cost is computed as following

$$C_i^k = \begin{cases} 0 & if\ R_i^k \leq R_{min} \\ -log(R_i^k) & otherwise \end{cases} \tag{10}$$

where R_{min} is a threshold to evaluate whether the distance between the detection and the target is too far or not. By minimizing the cost function as mentioned in [13], an optimal solution is found to correctly match targets and detections.

IV. Experimental Setup

A. Dataset

We used the first Michigan dataset (static dataset) [1], collected in indoor environments with a fixed Microsoft Kinect camera mounted approximately 2 meters high, to test the accuracy and the processing time of our method and its competitors. This database consists of 17 log files each spanning 2 to 3 minutes. For evaluating the accuracy of our method under the conditions of a running mobile robot, the second dataset (the on-board dataset) was used with a Microsoft Kinect camera mounted on-board a robot (PR2). This dataset consists of 18 log files recorded in offices, corridors and the cafeteria. Our goal was to evaluate the performance of our method in indoor environments in which both the humans and the robot move under different illumination conditions and in which the human either changes in a variety of poses or is occluded. Figure 8 shows some sample images extracted from our datasets.

We evaluated the accuracy and the processing time of our method and its closest competitor, the Reversible Jump Markov Chain Monte Carlo Particle Filtering (RJ-MCMC) [1]. In all our experiments, humans are hand-annotated by bounding boxes around upper bodies. The experiments implemented on both Michigan datasets were carried out using C++ on a PC with 2.5 GHz Intel Core i5 CPU.

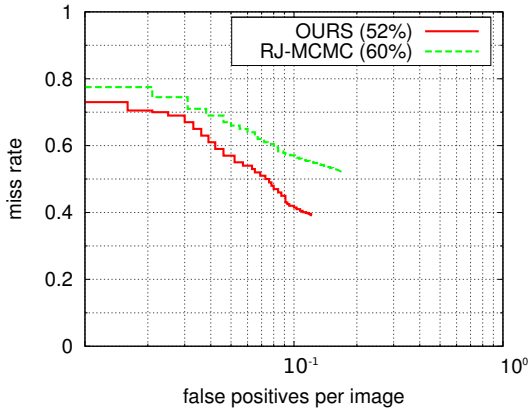

Fig. 6. Results of human tracking on the first Kinect dataset. Our algorithm, with an improvement of 8.0 %, significantly outperforms the RJ-MCMC.

B. Results

We use the log-average miss rate (LAMR), mentioned in [14], to compare the performances, shown by the curve of miss-rate versus false-positive-per-image (FPPI). The log-average miss rate is computed by averaging miss rate at nine FPPI rates evenly spaced in the range of 10^{-2} to 10^{0}. If a curve ends before reaching a given FPPI rate, the minimum miss rate is applied.

On the first dataset, we show the comparison of two algorithms in Figure 6. Our algorithm, an improvement of 8.0 %, significantly outperforms the RJ-MCMC. On the second dataset, the improvement of our algorithm is 8.55 %, as indicated in Figure 7. These results prove that the combination of the fast compressive tracker and Kalman filter is more efficient than the RJ-MCMC, even when we do not use the expensive human detectors, such as the full body human detector, the depth based shape detector, the motion detector and skin color detector. Although both the face detector and the upper body detector can detect humans reliably, they can not detect the human in certain complicated poses in many frames. In these cases, the fast compressive tracker gives a high contribution to the performance of our algorithm due to its robustness to different poses of humans as well as in partial occlusion. In addition, the Kalman filter plays a significant role as the alternative to the fast compressive tracker to deal with a full occlusion.

Besides the accuracy of an algorithm, the processing time is also a very important factor in mobile robot performance. Hence we also compare our algorithm with the RJ-MCMC to point out which one meets the requirement of real time processing. The speeds of our algorithm and the RJ-MCMC on the Michigan datasets are shown on Table I. Although RJ-MCMC uses a GPU implementation, it is still much slower than our algorithm. This is explained by some improvements in reducing the search space of human detections as well as decreasing the number of search samples in compressive trackers. In particular, in each frame the fast compressive tracker just has to classify 40 samples on average instead of more than 7000 samples as the original one. This significantly improves the speed of the fast compressive tracker.

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

(e) (f) (g) (h)

Fig. 8. Examples of tracking results. The mobile robot can detect humans in different poses and in severe occlusions.

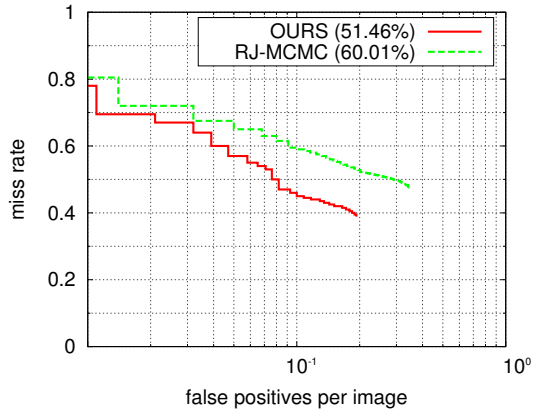

Fig. 7. Results of human tracking on the second Kinect dataset. Our algorithm, with the improvement of 8.5 %, is better than the RJ-MCMC.

TABLE I

COMPARISON OF SPEED ON THE MICHIGAN DATABASE

	First dataset	Second dataset	Platform
Ours	23.8 fps	22.2 fps	CPU
RJ-MCMC	4 fps	4 fps	GPU

V. CONCLUSION AND FUTURE WORK

In this paper, we have introduced a system for multiple person detection and tracking by a mobile robot. The results indicate that the fusion of detections from the face detector and upper body detector provides reliable observation cues for tracking multiple humans. Furthermore, the combination of the fast compressive tracker and Kalman filter is robust to motion, pose variation and occlusion. In the future, we are trying to develop an algorithm of human reidentification based on the information of color and depth in order to combine it with the current tracking system. This combination will enable the mobile robot to track people more reliably and be able to recover lost tracks caused by long term full occlusions or temporary disappearance of humans in the robot's field of view.

REFERENCES

[1] W. Choi, C. Pantofaru, and S. Savarese, "Detecting and tracking people using an rgb-d camera via multiple detector fusion," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, Nov 2011, pp. 1076–1083.

[2] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008.*, June 2008, pp. 1–8.

[3] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, ser. ECCV'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 864–877.

[4] D. Schulz, W. Burgard, D. Fox, and A. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation, 2001.*, vol. 2, 2001, pp. 1665–1670 vol.2.

[5] N. Bellotto and H. Hu, "Multisensor-based human detection and tracking for mobile service robots," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 1, pp. 167–181, Feb 2009.

[6] P. Viola and M. Jones, "Robust real-time object detection," in *International Journal of Computer Vision*, 2001.

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, vol. 1, June 2005, pp. 886–893 vol. 1.

[8] C. Wojek, S. Walk, and B. Schiele, "Multi-cue onboard pedestrian detection," in *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009.*, June 2009, pp. 794–801.

[9] W. Choi and S. Savarese, "Multiple target tracking in world coordinate with single, minimally calibrated camera," in *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 553–567. [Online]. Available: http://dl.acm.org/citation.cfm?id=1888089.1888132

[10] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vision*, vol. 77, no. 1-3, pp. 125–141, May 2008.

[11] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1619–1632, 2011.

[12] X. Mei and H. Ling, "Robust visual tracking using l1 minimization," in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, pp. 1436–1443.

[13] M. Vo-Duc, A. Masselli, and A. Zell, "Real time face detection using geometric constraints, navigation and depth-based skin segmentation on mobile robots," in *2012 IEEE International Symposium on Robotic and Sensors Environments*, 2012.

[14] C. Wojek, S. Walk, S. Roth, and B. Schiele, "Monocular 3d scene understanding with explicit occlusion reasoning," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 1993–2000.