2017 14th Conference on Computer and Robot Vision

Person Following Robot Using Selected Online Ada-Boosting with Stereo Camera

Bao Xin Chen*, Raghavender Sahdev*, and John K. Tsotsos
Department of Electrical Engineering and Computer Science and Centre for Vision Research
York University
Toronto Canada
{*baoxchen, sahdev, tsotsos*}@*cse.yorku.ca*

***Abstract*—Person following behavior is an important task for social robots. To enable robots to follow a person, we have to track the target in real-time without critical failures. There are many situations where the robot will potentially loose tracking in a dynamic environment, e.g., occlusion, illumination, pose-changes, etc. Often, people use a complex tracking algorithm to improve robustness. However, the trade-off is that their approaches may not able to run in real-time on mobile robots. In this paper, we present Selected Online Ada-Boosting (SOAB) technique, a modified Online Ada-Boosting (OAB) tracking algorithm with integrated scene depth information obtained from a stereo camera which runs in real-time on a mobile robot. We build and share our results on the performance of our technique on a new stereo dataset for the task of person following. The dataset covers different challenging situations like squatting, partial and complete occlusion of the target being tracked, people wearing similar clothes, appearance changes, walking facing the front and back side of the person to the robot, and normal walking.**

***Keywords*-Online Ada-Boosting, real-time tracking, person following robot**

I. Introduction

Person following robots need a robust and real-time algorithm to solve the tracking problem in a dynamic environment which may encounter unexpected circumstances; for example, the tracking target might be occluded by other instances, the lighting condition in the scene might change rapidly, and the target might change its pose dramatically (eg: squat down and pick up something from the floor or removing a bag from the person (see Figure 1)). To the best of our knowledge this is the first work which can handle situations when two people are wearing the same clothes and the tracker can still track the correct target under partial and complete occlusions in the context of person following robots; it can also deal with appearance changes, like removing a jacket, the tracker still tracks the target (human) and not the jacket. Another challenge is maintaining a given distance from the robot to the target, a natural consequence of following behaviour of the robot. The robot being used here is the Pioneer 3AT robot as shown in Figure 5. The main contributions of this paper are as follows: (i) a novel approach building on the Online Ada-Boosting

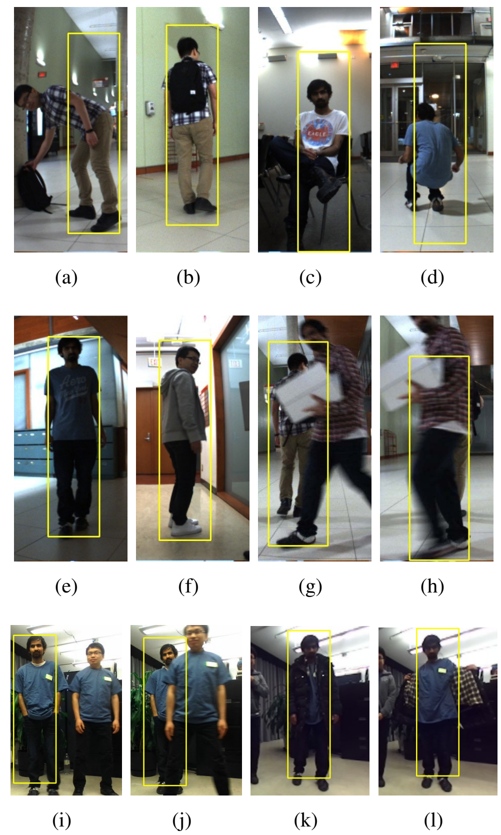


Figure 1. Different cases that our approach (Selected Online Ada-Boosting) can handle. (a) picking bag. (b) wearing bag. (c) sitting. (d) squatting. (e) illumination. (f) side facing. (g) partial occlusion. (h) complete occlusion. (i) standing side-by-side with the same clothes. (j) front crossing with the same clothes. (k), (j) appearance changed.

tracker, (ii) a novel algorithm named Selected Online Ada-Boosting which can run in real-time to follow a given target and is more robust than the current state of the art for person following robots (see Figure 1), (iii) a novel stereo dataset of different indoor environments for person following. We first describe the relevant work being used for the person following behavior and explain the Online Ada-Boosting approach in Section II. In Section III, we describe our proposed approach which modifies the Online Ada-Boosting

*Denotes equal contribution

978-1-5386-2818-8/17 $31.00 © 2017 IEEE
DOI 10.1109/CRV.2017.55

algorithm to make it more robust. Section IV describes the system design of the proposed approach. In Section V, we provide the experimental results of our approach and describe the dataset. Finally, Section VI concludes the paper and provides possible future work.

II. Related works

For person following robots, detecting and tracking humans is an important task. Dollar et al. [1] provide a survey on state of the art human detection papers. Here we describe approaches for person following robots using tracking and depth detection which is the major focus of this work.

A. Real-Time Tracking for Person Following Robots

In 1999, Ku et al. [2] attached a rectangular shape to the back of the person as the interest region with a particular color. Their method could solve the simple detection problem, but it did not provide any robustness. In 1998, Piaggio et al. [3] started using optical flows for a person following robot. Similar work was done in [4] and [5] as well. However, optical flow has the restriction that the person and background must have different motions which are not always the case. In 2003, Beymer et al. [6] used wheel odometry to subtract background motion and estimate the person location. However this only works well on uniform surface. In 2003, Tarokh et al. [7] used colour and shape of the person's clothes as features for detection. Although their method improved the robustness over Ku et al. [2], they did not consider situations when the target changes his/her appearance heavily. In 2006, Yoshimi et al. [8] used feature points (edges or corner points) detection and combined the pre-registered color and texture of the clothes. This method provided good robustness when the person is making a turn or walking in upright poses. In 2007, Calisi et al. [9] used a pre-trained appearance model to detect and track the person. Their method could provide a good tracking result if they trained the model well enough with a lot of data. However, dynamic environments are unpredictable, and the target might change appearance from time to time. Similarly, in 2007, Chen et al. [10] used sparse Lucas-Kanade features to track the target. But the features could be lost if the person is turning, or changing appearance. Again in 2007, Takemura et al. [11] used H-S Histogram in hue-saturation-value (HSV) color space, where HSV is robust to illumination since V (lightness) can be considered separately. In 2009, Satake et al. [12] used depth templates and SVM to train a human upper body classifier to track the person. However, this method did not handle cases, such as crossing, partial occlusion, etc. In 2010, Tarokh et al. [13] used HSV and controlled the light exposure to handle light variations. An update was made in 2014 to improve the following speed [14]. Some other fundamental feature tracking algorithms were also used in later literature, e.g., SIFT feature based [15] in 2012, HOG feature based [16] in 2013 and [17] in

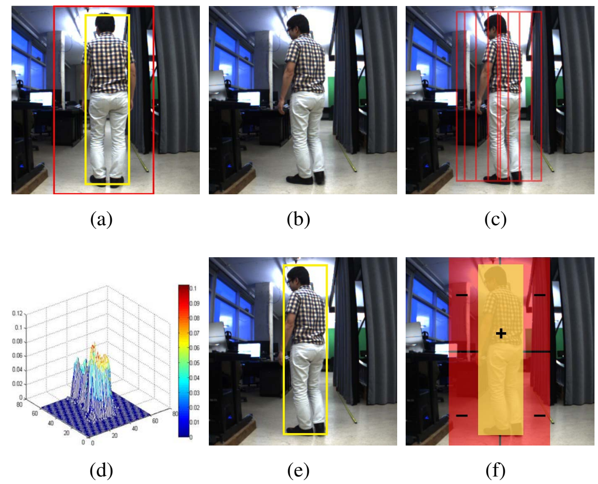


Figure 2. OAB updating process: (a) yellow box is the target region, the red box is the search region. (b) is the next frame. (c) is searching and evaluating the patches in the search region. (d) is the confidence map of the evaluation. (e) is the best matching with minimum error. (f) update the classifier with positive and negative patches. After (f) then go back to (a) to search in the next frame. Similar to [20], [21]

2014, etc. In the latest work (2016), Koide et al. [18] applied height and gait with appearance features for person tracking and identification, but height and gait are only limited to the target walking in an upright position. The method is not robust when the target changes its clothes or puts on a backpack ([19] also has this problem).

B. Depth detection

In this paper, we use depth to assist the tracking model for improving the reliability. Yoon et al. [22] gained aid from depth information to improve the computational speed and accuracy. Depth could also help with background and foreground issues by eliminating the sudden depth changing pixels, e.g., occlusion. Doisy et al. [23] used the Kinect camera and a laser sensor to propose an algorithm which solves the person depth information for person following. Bajracharya et al. [24] used depth from a stereo camera for detecting and tracking pedestrians in outdoor environments.

Nowadays, there are many different types of depth sensors in the market. In the modern publications, researchers prefer RGB-D cameras, eg: Kinect [25], ASUS xTion [19] and [22]. These cameras provide very good depth information only if the robot is running indoor without strong sunlight. Our approach uses a Point Grey Bumblebee 2 stereo camera which can be used both indoors and outdoors. Laser sensors provide another approach to detect depth [16] and [17]. But a laser sensor is expensive and often not permitted in places like hospitals, universities, malls and other similar places.

To obtain the depth information of each pixel in an image, we use a stereo image based algorithm to compute the depth. Since focal length and baseline are constants in a single stereo camera, we are only interested in disparity [26].

C. Online Ada-Boosting (OAB) Tracker

Boosting algorithms have been used in many areas in machine learning and computer vision ([27], [28], [29], [30]). Boosting usually trains with offline datasets. Online Ada-Boosting algorithm for tracking an object in real-time has been described by Grabner et al. in [20] and [21]. To achieve real-time tracking, Grabner et al. used Haar wavelet features to improve robustness when appearance changes gradually, which was described by Wang et al. in [31].

In OAB, the tracking target is assumed to be given in the first frame (selected by human or detected by an off-line detection algorithm). The selected patch is used as a positive example to train the classifier. Then random patches are extracted from four regions (upper right, upper left, bottom right, bottom left, see Figure 2(f)) in the search area as negative examples. These random patches contain negative features, e.g., windows, wall, furniture, etc. An initial classifier is trained from these positive and negative patches. In the second frame, the target is detected using the classifier. The patch in the search region with minimum error is the best responding example. This patch is used as a positive example and the surrounding random patches from the four regions as negative examples to update the classifier. The steps performed on the second frame are continued on the subsequent frames (see Figure 2).

In order to achieve real-time boosting, OAB does not use all weak classifiers to calculate a strong classifier [21]. Instead, it selects N weak classifiers from all M global weak classifiers. In the following equations, H^{weak} is the set of all weak classifiers, $H^{selected}$ is the set of selected weak classifiers from H^{weak}, y is the prediction of boosting, and α_n is the weight of each selected classifiers.

$$H^{weak} = \{h_1^{weak}, ..., h_M^{weak}\} \tag{1}$$

$$H^{selected} = \{h_1^{selected}, ..., h_N^{selected}\} \tag{2}$$

$$h_n^{selected} = h_m^{weak} \tag{3}$$

$$y = \sum_{n=1}^{N} \alpha_n * h_n^{selected} \tag{4}$$

α_n in Equation 4 is calculated according to the error of selected weak classifier $h_n^{selected}$.

III. Approach

To the best of our knowledge, this is the first paper that introduces the Online Ada-Boosting tracking algorithm (OAB) [21] for a person following robot. On top of the OAB algorithm, we add a depth image as an additional tool to assist the Ada-Boosting approach. We call this new modification as Selected Online Ada-Boosting (SOAB).

A. Computing depth from stereo images

The depth of each pixel can be easily calculated from the following equation [32]:

$$Z = \frac{fB}{x_l - x_r} \tag{5}$$

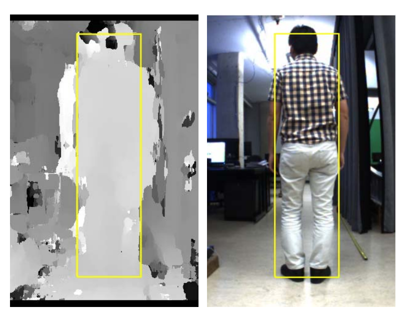
f is focal length. B is the baseline. x_l and x_r are the left and right image coordinates.

B. Classifier initialization

[20] and [21] initialized the first frame with a human to draw the bounding box. Here we present two ways to initialize the target to the tracker: *user defined* and a *pre-defined* bounding box.

For the *pre-defined* case, a bounding box was placed in the center of the image frame. The target has to walk into the bounding box at a particular distance from the robot. If all these conditions are satisfied, then the robot starts to initialize the classifier and follows the person. In our experiment, we draw a bounding box at pixel coordinates (272, 19) with the width equal to 100 pixels and the height as 390 pixels, and the default disparity is 200 (this is the initial disparity for the first frame).

For the *user defined* case, we do it as follows. Since the initial position of the person is known, and the depth image is given by equation 5, we can estimate the initial disparity of the person easily. Here we need to overcome a big problem: the depth image is very noisy (see Figure 3(a)). Let the initial patch be called I_p. We sort the pixels in I_p according to their disparity value (Note: the larger the disparity, the closer the distance. See equation (5)). Then we remove the disparities before 50th percentile and remove the disparities after 75th percentile. After that, we compute the mean of the remaining disparities as the initial depth. This method works, because the body of our target will almost fill the whole initial patch from our experiments (see Figure 3(b)), and noisy disparities would not be more than 25% in the initial patch. Removing 75% of the disparities will give us a precise result. This was found experimentally that retaining the disparities in



(a) (b)

Figure 3. (a) normalized disparity image. (b) image from the left camera.

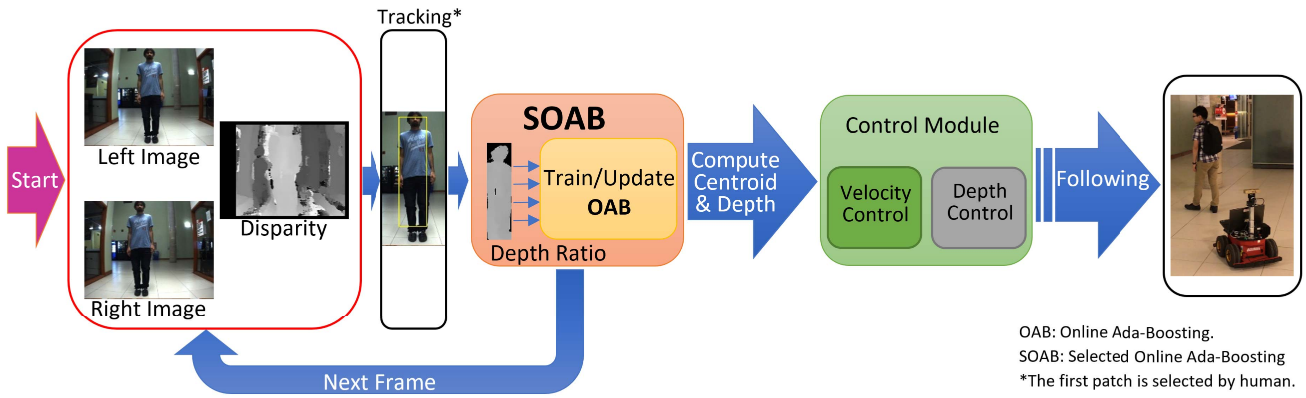


Figure 4. Proposed Approach: Tracking Module and the Control Module

this range gives best performance. In next subsection III-C, we will discuss when to update the classifier.

C. Selected Online Ada-Boosting (SOAB)

In this section, we will describe how to optimize OAB with given depth information on each pixel from Section III-A.

One of the weaknesses of the OAB algorithm is that the target might not always maintain the same size in the scene. The size of the target could be changed when it is occluded, changing poses, or tracking improperly in the current frame (see Figure 1). These weak detections pollute our classifier badly. Once the classifier adapts to those unwanted features, the tracker looses the target easily. Here unwanted features include background and foreground features. So, the depth of each pixel plays a significant role to calculate the proportion of unwanted features in the current positive patch. We call this proportion the depth ratio, R. Before computing this depth ratio for the positive patch, we need to determine where our target is in the previous frame (here we focus on the distance between the robot and the person).

Once the initial disparity (called $preDisp$) is computed from Section III-B , we estimate the disparity in the second frame. To do this, we run the original OAB algorithm to detect the positive patch in the second frame. Assuming that the displacement of the target can not be more than a threshold β (on the Bumblebee Camera, we use $\beta = 15$), the possible disparities that belong to the person are $preDisp \pm \beta$. Then we compute the mean of the pixels in $preDisp \pm \beta$ range as the current disparity (called $curDisp$). We assign $curDisp$ to $preDisp$ and repeat this for later frames to perform tracking.

$$curDisp = Mean(I_p[I_p \in preDisp \pm \beta]) \quad (6)$$

The next step is to update the classifier. We do this differently than OAB. We introduce the depth ratio R to evaluate the current positive patch containing a minimum

Data: CameraStream
fetch left and right image from CameraStream;
select target to track;
calculate $curDisp$;
$preDisp \leftarrow curDisp$;
pre-train OAB;
while *true* **do**
  fetch left and right image from CameraStream;
  run OAB to extract a positive patch I_p;
  $curDisp \leftarrow Mean(I_p[I_p \in preDisp \pm \beta])$;
  $R \leftarrow \frac{\sum [I_p \in preDisp \pm \beta]}{w*h}$;
  if $R \geq \gamma$ **then**
    update the classifier;
  end
  $preDisp \leftarrow curDisp$;
end

Algorithm 1: SOAB

amount of unwanted features. R equals the ratio of the number of pixels that are used to calculate $curDisp$ to that of the total number of pixels in the current patch. The width of patch I_p is w, and the height is h.

$$R = \frac{\sum [I_p \in preDisp \pm \beta]}{w * h} \quad (7)$$

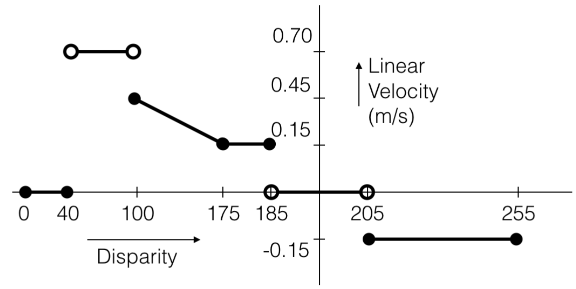
Now our algorithm (SOAB) makes the decision. If the depth ratio R is greater than a threshold γ, then we update the classifier using the current positive patch. Otherwise, we do not update the classifier.

IV. SYSTEM DESIGN

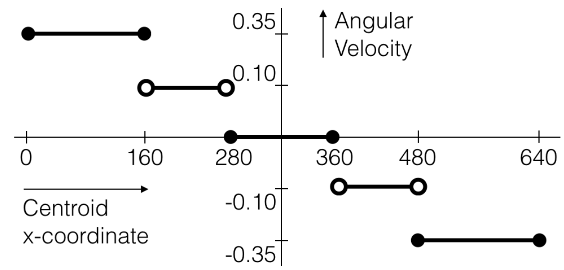
In this part, we describe the design of our system. Here we use a Pioneer 3AT robot (see Figure 5.) which is a four wheeled differential drive robot with an on-board computer. It is configured with a Point Grey Bumblebee Stereo Camera which acts as the only sensor on the robot to sense its environment. The system is built using the robot operating system (ROS) to integrate different components involved in the system. Figure 4 gives an overview of our system design.

Initial components of the system are responsible for tracking the target (human) and computing the centroid and depth of the target being tracked. Based on these values, the control module computes the corresponding linear and angular velocities for the robot (see Figure 6). The controller maintains a predefined distance from the human being followed. It is ensured that the centroid of the human target bounding box is always near the centre of the image within a pre-specified area. This is done by simply steering in the direction to which the person is moving. If the person appears to be moving left in the image, the robot moves leftwards to keep the centroid of the detected human near the center of the image. The robot maintains a set depth from the target. If the person is moving towards the robot, the robot moves backward and vice versa. The linear velocity of the robot is a function of the disparity alone and the angular velocity is a function of the x-coordinate of the centroid of the human being tracked (see Figure 6). These functions were obtained experimentally and would change with the change in the robot platform.

We run this system on a laptop with Intel core i7, 2nd Generation, 2.5GHz processor and 16GB RAM (the requirement is lower for our algorithm). The design of various components involved here is presented in Figure 4.



(a) Linear velocity vs. Disparity Plot



(b) Angular Velocity vs. centroid of the target

Figure 6. Controller Module of our system. (a) The function represents the linear velocity as a function of the target's disparity in the current frame (b) represents the angular velocity as a function of the x-coordinate of the centroid of the target

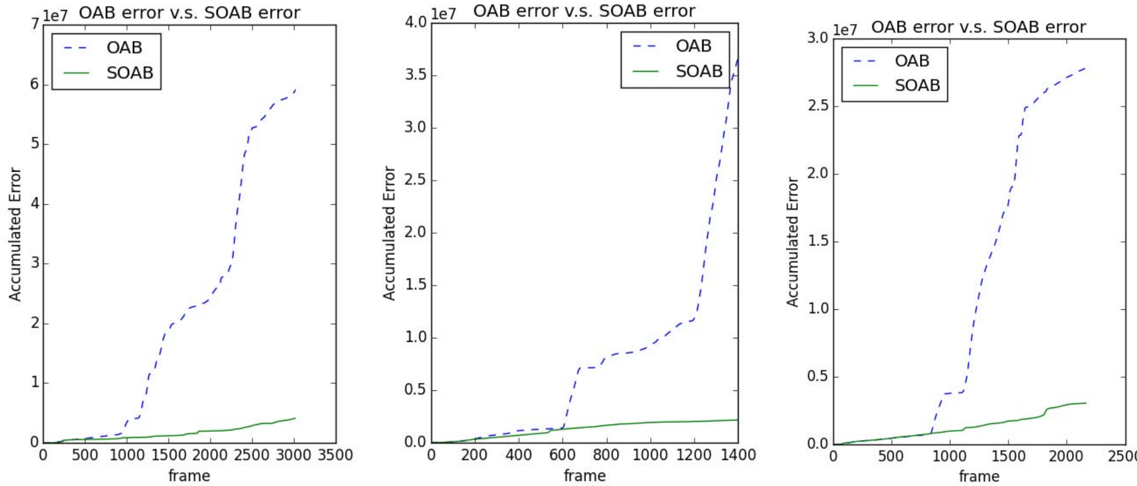
V. Experiments and Evaluation

Since the proposed method is different from what people did in the past, we could not find an existing dataset which satisfies our need (a stereo dataset for a human following robot under challenging situations). As a result, we build a dataset of 4 image sequences to test the robustness of the person following robot system. The person being followed in our dataset exhibits varying motions and challenging poses in different indoor environments (see Figure 1). The dataset is built from image sequences captured by the robot in these places. The robot is following a person in a university hallway, a living room, and a lecture hall. We make the dataset of these three places publicly available at our project page[1]. We provide ground truth for our dataset which has the bounding box drawn on the target being tracked. Demo videos of the robot following behavior of our proposed approach can also be found at the project page[1]. The dataset consists of the person being followed under varying illumination conditions, different poses of the person being followed, partial and complete occlusion of the person being followed and multiple people present in the scene. The resolution of the images is $640 * 480$ pixels in our dataset. We are able to track people while the robot is moving at up to speeds of 0.70 m/s. It should be noted that our proposed system could be deployed on any mobile robot platform. We tested our proposed approach on a Pioneer 3AT robot (see Figure 5). Our algorithm can run in real-time at a frame rate of 15fps on a single CPU core.

Figure 5. Pioneer 3AT robot mounted with a Point Grey Bumblebee stereo camera.

First, we tested our algorithm on an experimental sequence of images. The target in the sequence is turning around, and squatting down (see Figure 8). From the result, we could distinguish that SOAB with depth ratio threshold $\gamma = 0.60$ outperforms the original OAB. By selecting the patches to update the classifier does make a huge improvement. In Figure 8(f), OAB did a mistake and updated the classifier. The classifier then learned the background as the

[1] http://jtl.lassonde.yorku.ca/2017/02/person-following



(a) Sequence Hall Way (b) Sequence Multiple Crossing (c) Sequence Same Clothes Crossing

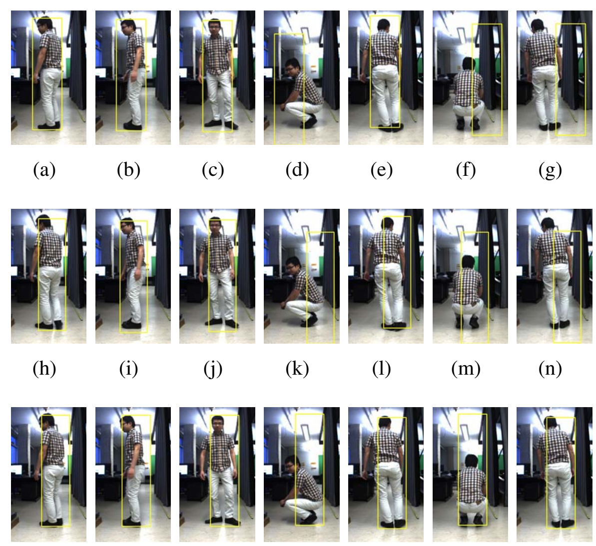
Figure 7. The graphs are comparing the accumulated square error on three different image sequences captured in different places and the target acted very differently.

important feature and as a result continuously made mistakes in later frames. On the other hand, SOAB avoids this problem by using the depth information to make decision on whether or not to update the classifier. We also select a depth ratio threshold $\gamma = 0.80$ for testing. Since the threshold is too high, SOAB skipped most of the frames. This is not how we want SOAB to behave. In the later experiment, we fixed the depth ratio threshold as 0.60.

We made another image sequence to test more challenging scenarios. The target is picking up a backpack from the ground, and someone is passing between the robot and the target in the sequence (see Figure 9). Again in this test case, SOAB achieved the best result overall. Comparing Figure 9(b), OAB learnt the background features leading to a mistake in Figure 9(c). From Figure 9(e-g), OAB learned the features of the crossing person. The second person became the target of the OAB tracker. Since the depth information is used as a gate, SOAB did not update the classifier with unwanted features when depth ratio is less than the threshold. Figure 7(a) shows the accumulated square error of OAB and SOAB. The green line in the graph increases very smoothly meaning that SOAB performed very well without losing track. But, OAB looses track at about frame 900 and becomes very unstable later, roughly at the occlusion in Figure 9(f).

Another image sequence was made to test multiple crossing with different speed. The comparison between OAB and SOAB can be seen in Figure 7(b). There are 12 crossing actions in this sequence. SOAB completed this test case without failure. But, OAB failed after the fourth crossing.

The third sequence is for testing when two people are wearing the same clothes. This sequence is the most significant one in our dataset. The result can be seen in Figure 7(c). In this sequence, two people are crossing each other, walking in a circle. As expected, the robot is following the same person all the time using SOAB.



(a) (b) (c) (d) (e) (f) (g)

(h) (i) (j) (k) (l) (m) (n)

(o) (p) (q) (r) (s) (t) (u)

Figure 8. (a-g) is tracking using original OAB algorithm. (h-n) is tracking using SOAB with depth ratio threshold $\gamma = 0.30$. (o-u) is tracking SOAB with with depth ratio threshold $\gamma = 0.60$.

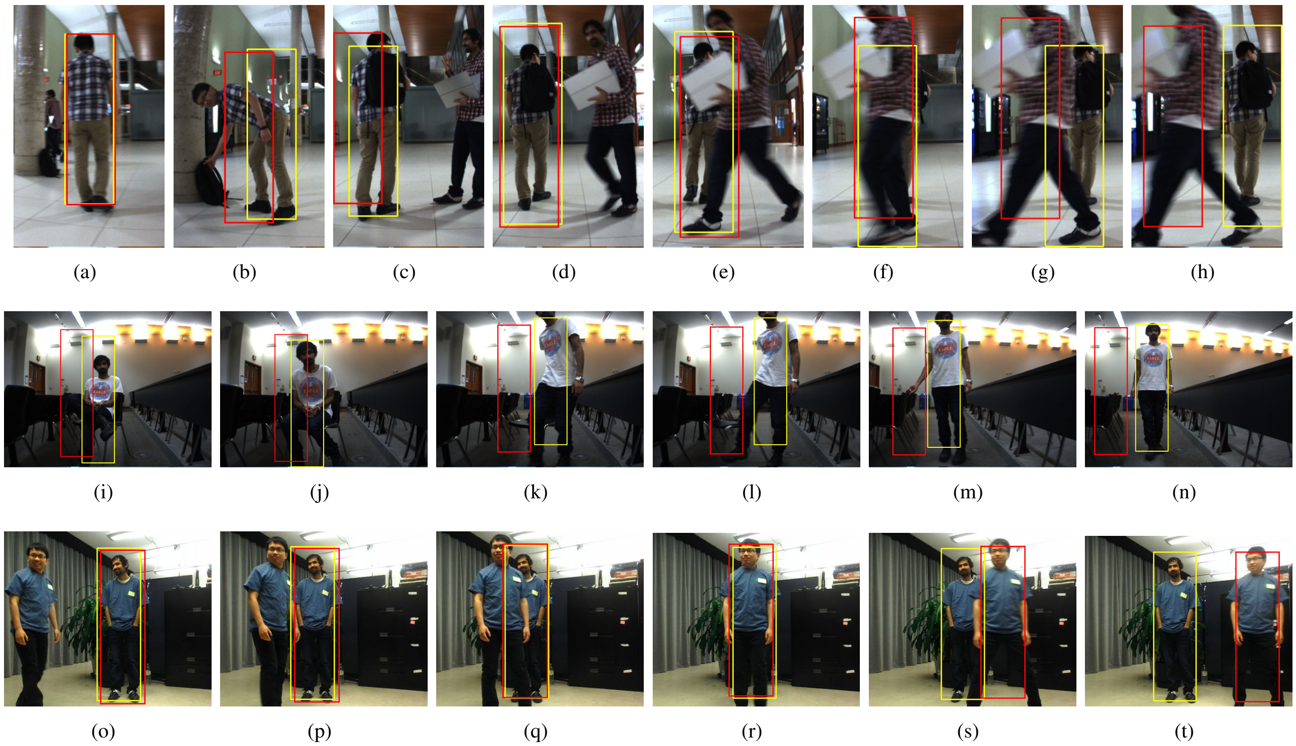


Figure 9. Red box is tracking using original OAB algorithm. Yellow box is tracking using SOAB with depth ratio threshold $\gamma = 0.60$. (a-h) are sequences from a hallway. (i-n) are sequences from a lecture hall. (o-t) are sequences showing crossings with same clothes.

VI. Conclusion and Future Work

In this paper, we described a robust person following robot system using a modified version of Online AdaBoosting algorithm with only a stereo camera. The system was optimized to perform well in a dynamic environment. Our modified version of OAB performs much better than the original algorithm (see Figure 7). We handled difficult situations dealing with similar clothes of people crossing, appearance changes in terms of removing the target's jacket, partial and complete occlusions and were able to run our approach in real-time on a mobile robot. It should also be noted that even though we present our approach for the human following robot, this can be applied to any object following robot as well, but the object needs to be known *a priori*. For instance the robot can follow objects like a handbag, shopping cart, an animal (cat/dog), etc. In this sense our approach targets not only the human following task but also generalizes to other objects as well.

We proposed changes to the OAB algorithm. We believe that there could be further improvements, e.g., using a more robust online boosting tracking algorithm called Online Multiple Instance Learning [33], or increasing the classification error if the bounding box jumps unstably from frame to frame. Another possible future work would be to include the recognition aspect by making use of a human detector to aid in the process of having a better model for the classifier. Another approach of making the following system more reliable could be adding a path planning and obstacle avoidance strategy to the robot control module in our system.

Acknowledgement

We acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the NSERC Strategic Network for Field Robotics (NCFRN), and the Canada Research Chairs Program through grants to John K. Tsotsos. The authors would like to thank Sidharth Sahdev for helping in the process of dataset generation and making the video for this work.

References

[1] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[2] C.-H. Ku, W.-H. Tsai *et al.*, "Smooth vision-based autonomous land vehicle navigation in indoor environments by person following using sequential pattern recognition," *Journal of Robotic Systems*, vol. 16, no. 5, pp. 249–262, 1999.

[3] M. Piaggio, R. Fornaro, A. Piombo, L. Sanna, and R. Zaccaria, "An optical-flow person following behaviour," in *Intelligent Control (ISIC), 1998. Held jointly with IEEE International Symposium on Computational Intelligence in Robotics*

and Automation (CIRA), Intelligent Systems and Semiotics (ISAS), Proceedings. IEEE, 1998, pp. 301–306.

[4] T. Yamane, Y. Shirai, and J. Miura, "Person tracking by integrating optical flow and uniform brightness regions," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 4. IEEE, 1998, pp. 3267–3272.

[5] G. Chivilò, F. Mezzaro, A. Sgorbissa, and R. Zaccaria, "Follow-the-leader behaviour through optical flow minimization," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 4. IEEE, 2004, pp. 3182–3187.

[6] D. Beymer and K. Konolige, "Tracking people from a mobile platform," in *Experimental robotics VIII*. Springer, 2003, pp. 234–244.

[7] M. Tarokh and P. Ferrari, "Case study: Robotic person following using fuzzy control and image segmentation," *Journal of Field Robotics*, vol. 20, no. 9, pp. 557–568, 2003.

[8] T. Yoshimi, M. Nishiyama, T. Sonoura, H. Nakamoto, S. Tokura, H. Sato, F. Ozaki, N. Matsuhira, and H. Mizoguchi, "Development of a person following robot with vision based target detection," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 5286–5291.

[9] D. Calisi, L. Iocchi, and R. Leone, "Person following through appearance models and stereo vision using a mobile robot." in *VISAPP (Workshop on on Robot Vision)*, 2007, pp. 46–56.

[10] Z. Chen and S. T. Birchfield, "Person following with a mobile robot using binocular feature-based tracking," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 815–820.

[11] H. Takemura, K. Ito, and H. Mizoguchi, "Person following mobile robot under varying illumination based on distance and color information," in *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*. IEEE, 2007, pp. 1500–1505.

[12] J. Satake and J. Miura, "Robust stereo-based person detection and tracking for a person following robot," in *ICRA Workshop on People Detection and Tracking*, 2009, pp. 1–10.

[13] M. Tarokh and P. Merloti, "Vision-based robotic person following under light variations and difficult walking maneuvers," *Journal of Field Robotics*, vol. 27, no. 4, pp. 387–398, 2010.

[14] M. Tarokh and R. Shenoy, "Vision-based robotic person following in fast walking," in *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3172–3177.

[15] J. Satake, M. Chiba, and J. Miura, "A sift-based person identification using a distance-dependent appearance model for a person following robot," in *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*. IEEE, 2012, pp. 962–967.

[16] M. Awai, T. Shimizu, T. Kaneko, A. Yamashita, and H. Asama, "Hog-based person following and autonomous returning using generated map by mobile robot equipped with camera and laser range finder," in *Intelligent Autonomous Systems 12*. Springer, 2013, pp. 51–60.

[17] M. Awai, A. Yamashita, T. Shimizu, T. Kaneko, Y. Kobayashi, and H. Asama, "Development of mobile robot system equipped with camera and laser range finder realizing hog-based person following and autonomous returning," *Journal ref: Journal of Robotics and Mechatronics*, vol. 26, no. 1, pp. 68–77, 2014.

[18] K. Koide and J. Miura, "Identification of a specific person using color, height, and gait features for a person following robot," *Robotics and Autonomous Systems*, vol. 84, pp. 76–87, 2016.

[19] Y. Yoon, W.-h. Yun, H. Yoon, and J. Kim, "Real-time visual target tracking in rgb-d data for person-following robots," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 2227–2232.

[20] H. Grabner and H. Bischof, "On-line boosting and vision," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 260–267.

[21] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting." in *Bmvc*, vol. 1, no. 5, 2006, p. 6.

[22] Y. Yoon, H. Yoon, and J. Kim, "Depth assisted person following robots," in *RO-MAN, 2013 IEEE*. IEEE, 2013, pp. 330–331.

[23] G. Doisy, A. Jevtic, E. Lucet, and Y. Edan, "Adaptive person-following algorithm based on depth images and mapping," in *Proc. of the IROS Workshop on Robot Motion Planning*, 2012.

[24] M. Bajracharya, B. Moghaddam, A. Howard, S. Brennan, and L. H. Matthies, "A fast stereo-based system for detecting and tracking pedestrians from a moving vehicle," *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1466–1485, 2009.

[25] A. Cosgun, D. A. Florencio, and H. I. Christensen, "Autonomous person following for telepresence robots," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4335–4342.

[26] M. Kanbara, T. Okuma, H. Takemura, and N. Yokoya, "Real-time composition of stereo images for video see-through augmented reality," in *Multimedia Computing and Systems, 1999. IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 213–219.

[27] C. Zhang and Z. Zhang, "Boosting-based face detection and adaptation," *Synthesis Lectures on Computer Vision*, vol. 2, no. 1, pp. 1–140, 2010.

[28] A. Beygelzimer, S. Kale, and H. Luo, "Optimal and adaptive algorithms for online boosting," in *25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 2016, pp. 4120–4124.

[29] Y. Sha and G.-y. Zhang, "An adaptive weighted boosting algorithm for road detection," in *Networking, Sensing and Control (ICNSC), 2010 International Conference on*. IEEE, 2010, pp. 582–586.

[30] K. Nguyen, T. Ng, and L. Nguyen, "Adaptive boosting features for automatic speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4733–4736.

[31] J. Wang, X. Chen, and W. Gao, "Online selecting discriminative tracking features using particle filter," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 1037–1042.

[32] M. Kanbara, T. Okuma, H. Takemura, and N. Yokoya, "A stereoscopic video see-through augmented reality system based on real-time vision-based registration," in *Virtual Reality, 2000. Proceedings. IEEE*. IEEE, 2000, pp. 255–262.

[33] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.