Integrating Stereo Vision with a CNN Tracker for a Person-Following Robot

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. In this paper, we introduce a stereo vision based CNN tracker for a person following robot. The tracker is able to track a person in real-time using an online convolutional neural network. Our approach enables the robot to follow a target under challenging situations such as occlusions, appearance changes, pose changes, crouching, illumination changes or people wearing the same clothes in different environments. The robot follows the target around corners even when it is momentarily unseen by estimating and replicating the local path of the target. We build an extensive dataset for person following robots under challenging situations. We evaluate the proposed system quantitatively by comparing our tracking approach with existing real-time tracking algorithms.

Keywords: CNN tracker · Person following robot · Tracking · Stereo vision

1 Introduction

Person following robots have many applications such as autonomous carts in grocery stores [26], personal guides in hospitals, or airports for autonomous suitcases [1]. Person following robots in dynamic environments need to address the tracking problem under different challenging situations (appearance changes, varying illumination, occlusions, pose changes such as crouching, exchanging jackets etc.). An online convolutional neural network (CNN) is used to track the given target under different situations. The target being tracked might move around corners making it disappear from the field of view of the robot. We address this problem by computing the recent poses of the target and have the robot replicate the local path of the target when the target is not visible in the current frame. The robot being used is a Pioneer 3AT robot which is equipped with a stereo camera. We tested our approach with two stereo cameras namely the Point Grey Bumblebee2[1] and the ZED stereo camera[2].

B.X. Chen and R. Sahdev—*Denotes equal contribution.*

[1] http://www.ptgrey.com/stereo-vision-cameras-systems.

[2] https://www.stereolabs.com.

M. Liu et al. (Eds.): ICVS 2017, LNCS 10528, pp. 300–313, 2017.
DOI: 10.1007/978-3-319-68345-4_27

The main contributions of this paper are: (*i*) A Person Following Robot application using a CNN trained online in real-time (≈20 fps) making use of RGB images and a stereo depth image for tracking, (*ii*) a robot following behaviour which can follow the person even when the person is transiently not in the field of view of the camera, (*iii*) a novel stereo dataset for the task of person following. First, we describe the relevant work for human following robots and tracking using CNNs in Sect. 2. In Sect. 3, we describe our proposed CNN model and the navigation system of the robot. We describe the dataset and experimental results of our approach in Sect. 4. Finally, Sect. 5 concludes the paper and provides possible future work.

2 Related Works

Person Following Robots: Person Following robots have been researched as early as 1998 [31] where the authors used color and contour information of the target for tracking. Similar color based tracking was done in [34] and using an H-S Histogram in hue saturation value (HSV space) in [33]. These approaches could not handle appearance changes or occlusions very well. Some early optical flow based works include that of [8,36]. Optical flow requires the target motion to be different from background motion which limits its usability. Simple feature based works were presented by using edges, corners with color and texture information by Yoshimi et al. [37]. Pre-trained appearance models were used in [4]. Some other feature based methods include Lucas-Kanade features [7], SIFT features [30], HOG features [2] and height and gait with appearance based features in [25]. Recently in 2017 [6] used Selected Online Ada-Boosting to do online learning using depth as a filter to restrict the search for the target. People have been using various other sensors for person following robots like laser based approaches [24] and RGBD camera based approaches, e.g., Kinect [9,12]. Kinect has the drawback of only working indoors. Laser based approaches might not be suitable for places like hospitals, schools, or retail stores which might have a restriction on the usage of laser. Our approach uses a stereo camera which can be used both indoors and outdoors.

Object Tracking: Real-time object tracking is an important task for a person-following robot. Many state of the art algorithms exist that can achieve high accuracy (robustness), e.g., [29] (MGbSA), [17] (CNN as features), [22] (Proposal Selection), [38] (deep learning), [28] (Locally Orderless tracking), etc. However, these approaches do not target real-time performance. Some other works that focus on computation speed include [20] (Struck SVM with GPU), [41] (Structure preserving), [39] (Online Discrimination Feature Selection), [18] (Online Ada-Boosting), etc. Recent work from Camplani et al. [5] (DS-KCF) used RGBD image sequences from a Kinect sensor to track objects under severe occlusions and rank highly on the Princeton Tracking Benchmark [32] with real-time performance (40 fps). One of the earliest works using convolutional neural networks (CNNs) for tracking appeared in 2010 by Fan et al. [14]. They considered tracking as a learning task by using spatial and temporal features to estimate location and scale of the target. Hong et al. [21] used a pre-trained CNN to generate features to train an SVM classifier. Zhai et al. [38] also used a pre-trained CNN, but added a Naive Bayes classifier after the last layer of the CNN. Zhang and Suganthan [40] used one single convolutional layer with 50 4-by-4 filters in the CNN structure. The network was trained from scratch and updated every 5 frames. Gao et al. [17] used pre-trained CNN as feature generator to enhance the ELDA Tracker [16].

CNN Using RGBD Images: Training a CNN model with RGB and stereo depth images is another focus of this paper. Previous work used RGBD CNNs on object detection [19] and object recognition [13]. Couprie et al. [10] used RGBD images to train a single stream CNN classifier to handle semantic segmentation. Eitel et al. [13] trained RGB layers and D layer separately in two CNN streams. These two streams were combined in the fully connected layer.

3 Approach

Here we describe our proposed CNN models and the learning process. The input to the CNN is the RGB channel and the computed depth from the stereo images, we call this as RGBSD (RGB-Stereo Depth). Stereo Depth (SD) is computed using the ZED SDK[3]. The CNN Tracker outputs the depth and the centroid of the target. The depth and centroid are then used by the navigation module of the robot to follow the target and replicate the path when required.

3.1 CNN Models with RGBSD Images

We develop three different CNN models and use each of them separately to validate our approach. The first model (CNN_v1) uses RGBSD layers as a single image to feed the ConvNet. Similar to conventional CNN architectures, the network contains convolutional layers, fully-connected layers, and an output layer (see Fig. 1). The second model (CNN_v2) uses 2 convolutional streams and the input is RGB channels for one stream and just the stereo depth image for the other (see Fig. 1). In the fully connected layer, the input is a combination of the flattened output from those two convolutional streams. The third ConvNet (CNN_v3) is a regular RGB image based CNN. It has a similar structure as that of the first model. Now we describe our approach to initialize and update the CNN tracker.

Initial Training Set Selection: In order to use the CNN model to track a person, we must initialize the CNN classifier. The initialization is done from scratch using random weights. A pre-defined rectangular bounding box is placed in the center of the first frame. To activate the robot following behaviour, a person must stand inside the bounding box at a certain distance from the robot or the target to be tracked can be manually selected. Once the CNN is activated, the

[3] https://github.com/stereolabs/zed-opencv.

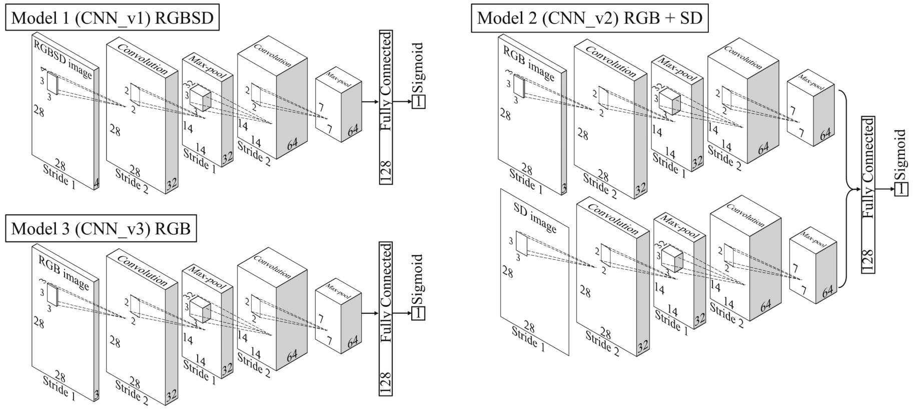


Fig. 1. Three CNN models: Model 1 takes a 4-channel RGBSD image as input; Model 2 takes an RGB image and an SD image as input; Model 3 takes an RGB image only as the input. The parameters of the CNN in each of the layers are chosen empirically for real-time performance.

patch in the bounding box is labeled as class-1. The patches around the bounding box are labeled as class-0. Since these two classes are highly unbalanced, we uniformly select n patches from class-0, and copy the class-1 patch n times to form the training set ($n = 40$ in our experiment). This initial training set is used to train a CNN classifier until it has a very high accuracy on the training set. This might make the classifier overfit the training set. To handle this strong over-fitting, we assume that the target pose and appearance should not change dramatically in the first 50 frames (about 2–3 s).

Test Set Selection: Once the CNN classifier is initialized or updated, we use it to detect the target in the next frame. When a new frame is available along with the stereo depth layer, we search the test patches in a local image region as shown in Fig. 2(a). We also restrict the search space with respect to the depth as shown in Fig. 2(b). If the patches in the image do not have the depth within $previous\_depth \pm \alpha$, we do not consider them (Fig. 2(c)), where α is the search region in depth direction (we use $\alpha = 0.25$ m). By doing this, most of the patches belonging to the background will be filtered out before passing to the CNN classier. Only the highest responses on class-1 will be considered as the target in the current frame. If no target is detected (e.g., highest responses on class-1 < 0.5) after 0.5 s, it will enter the target missing mode. Then, the whole image is scanned to create a test set.

Update CNN Tracker: To update the classier, a new training set needs to be selected. The update step is performed only if the detection step finds the target (class-1) in the test set. In order to maintain robustness, the most recent 50 class-1 patches are retained from the previous frames to form the class-1 patch pool which is implemented as a First-In-First-Out queue. The patches around the target form the class-0 patch pool. In this new training set, we again uniformly

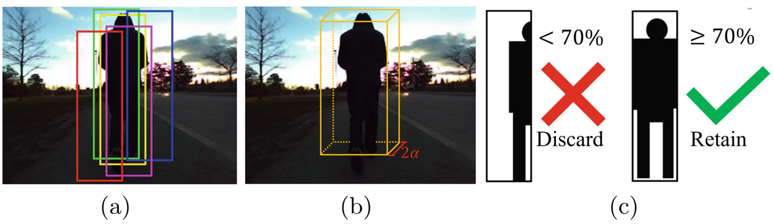
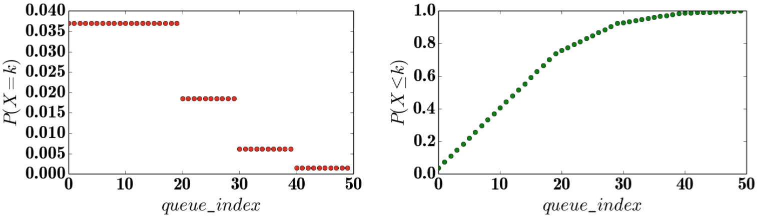


Fig. 2. 3D search region for test set (a) candidate test patches in 2D region (based on a sliding window approach), (b) search region with respect to depth, (c) pixels in black are within ±α *meters* from the previous depth. If black pixels are less than 70% of the patch, the patch will be discarded, else, it will be retained. The number 70% is chosen experimentally as this covers the human body completely in most of the cases. According to (c), the red and blue patches in (a) will be discarded, the green, pink, and yellow patches will be retained. (Color figure online)

select n patches from class-0 patch pool. For selecting n patches from class-1 patch pool, we sample the patches based on a Poisson distribution with $\lambda = 1.0$ and $k = \lfloor \frac{queue\_index}{10} \rfloor$ (see Eq. 1 and Fig. 3). This gives a higher probability of selecting patches from the recent history rather than selecting older patches. This training set is used to update the classifier. The Poisson distribution based sampling of class-1 patches avoids overfitting and provide a chance to recover from bad detection in the previous frame(s).

$$P(k) = e^{-\lambda} \frac{\lambda^k}{k!} \tag{1}$$



(a) normalized PMF

(b) CDF

Fig. 3. Poisson distribution with $\lambda = 1.0$ and $k = \lfloor \frac{queue\_index}{10} \rfloor$, where *queue_index* is the patch index in First-In-First-Out queue. To select an index, just randomly generate a real number from 0 to 1.0. Then, base on (b) the CDF graph, an index is selected.

3.2 Navigation of the Robot

In this section, we describe the navigation aspect of the robot. There are 2 cases: (i) when the robot can see the target (human) in the image; (ii) when the robot cannot see the target. A proportional integral derivative (PID) controller [27] is used in the former case while the path of the target is replicated in the latter. A local history of the target poses is maintained to compute the local path of the robot. The robot moves to the last observed pose of the target to find the target and continue the following behaviour. There are 4 basic components involved here: Localization of the robot, Target Pose Estimation, Robot following using a PID based controller, and a local path planner (trajectory replication).

Robot Following Using PID Controller: In this section we describe the robot following behavior for the case when the human can be seen in the image. A pre-specified distance, D is maintained between the robot and the target. The linear velocity, v of the robot is directly proportional to the error in current depth, $(d-D)$, where d is the current depth of the target. The angular velocity, ω is proportional to the error in the x coordinate of the target $(x-X\_mid)$. X_mid is the centre of the image in the horizontal direction. Only the Proportional and Integral components of the PID controller are used. We use $D = 1.0\,\mathrm{m}$. Following equations detail the velocities as a function of the error terms.

$$v = K_p * (d - D) + K_i * \int_T (d - D)dt; \tag{2}$$

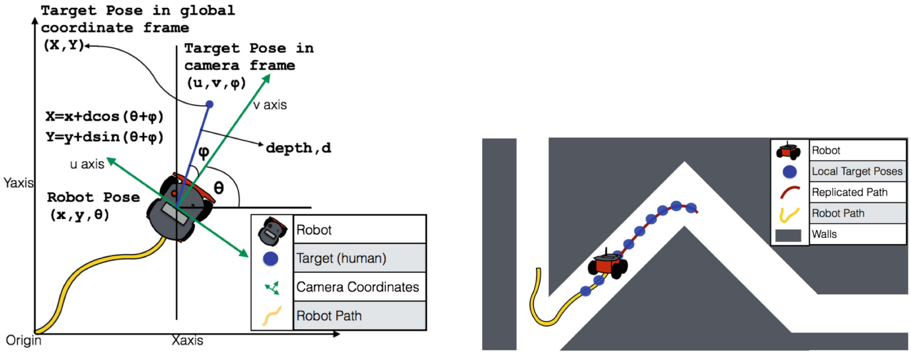
$$\omega = K_p^{'} * (x - X\_mid) + K_i^{'} * \int_T (x - X\_mid)dt; \tag{3}$$

where K_p, K_i, K_p', K_i' are the PI constants, $(d - D)$, $(x - X\_mid)$ are the error terms for the linear and angular velocities and dt is the time difference between successive frames.

Localization: Localization of the robot requires estimating the robot pose with respect to a global coordinate frame. In the 2D case, this is x,y coordinates and the orientation, θ of the robot. The robot must maintain an estimate of its pose as it moves in the presence of dynamic obstacles. Here we address localization using wheel odometry. Wheel odometry is reliable for short distances with an error of less than 4% for environments with a smooth surface (e.g., indoor flooring, outdoor pavement, sidewalk, etc.) for our robot (Pioneer3AT). For this work, the robot is tested in university hallways/corridors which often have minimal features or are featureless (blank walls), hence Visual Odometry based approaches [15] do not give accurate localization. Moreover, the environment is dynamic (has humans walking) which makes Visual odometry even less reliable.

For our work, it is only important that the pose of the robot is accurate for any short time (e.g., 5 s). This is the time we require localization information of the robot to compute the local path of the target and previously accumulated errors due to dead reckoning [3] do not matter.

Target Pose Estimation: The pose (World coordinates) of the target with respect to the camera frame is estimated using the depth and the focal length of the camera [23]. Knowing the pose of the robot and target pose with respect



(a) (b)

Fig. 4. (a)Estimation of the target pose in the global frame (top view) (b) Local Trajectory of the target poses is stored, when the robot cannot see the target in the image the robot simply replicates the latest local history of target poses stored to find the target. In this work, local history of 100 poses is stored.

to the camera frame, the 2D pose of the target can be estimated accurately in a global frame. Figure 4(a) shows the top view for computing target pose.

Trajectory Replication/Path Planner: Here we describe the navigation algorithm used to follow the human when the robot cannot see the human. This part is used when the person is turning around a corner or around a tree in an outdoor context. The robot always keeps a local history of the recent p poses of the target with respect to the global coordinate frame, this is called the recent trajectory of the target (See Fig. 4(b)). We use $p = 100$ here. If the robot cannot see the target transiently for 0.5 s, it implies that the human turned around a corner or is blocked by something else, so the robot replicates the recent trajectory of the target. By doing so, the robot reaches the last observed pose of the target. After reaching this position, the robot should be able to find the target and resume the following behaviour using the PID based controller. If for some reason the robot cannot find the target after replicating the path, the robot turns on the spot to see if it can find the target, if not the robot stops there and the following behaviour terminates. On the other hand, if the robot finds the target while replicating the local path, the robot shifts to the PID based following behavior. Some of the cases when the target might not be found include when target runs away after the turn or turns somewhere else unexpectedly or vanishes due to some reason. In all these cases it is reasonable to assume that the robot would not be able to find the target. A similar behaviour is expected if a human is following another human.

The overview of our proposed approach is described in Fig. 5. The input to our system is an RGB image and a computed stereo depth image. These images are then run through an online CNN which runs at a frame rate of 20 fps. The CNN returns the depth, the centroid coordinates of the target being tracked and a flag which indicates the presence/absence of the target. If the target is

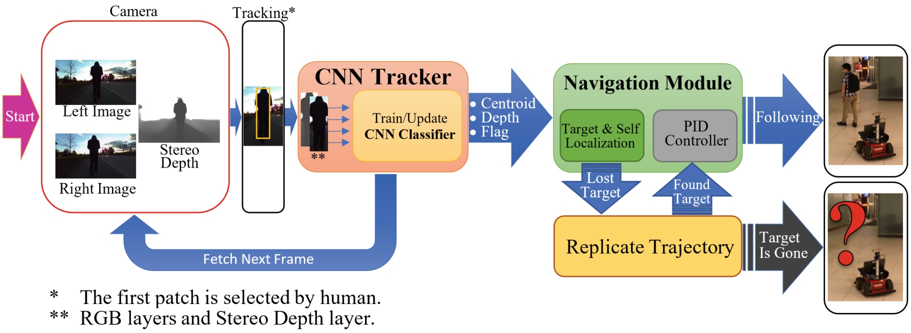


Fig. 5. Overview of the system design of our approach.

present in the scene a PID based controller is used to steer in such a way so as to keep the target in the center of the image; in case of absence of the target, the local path of the target is replicated by the robot to continue the following process. We run our robot at speeds up to 1.0 m/s. The Robot Operating System (ROS) was used for integrating the different components in this work. We tested our approach on a Dell Alienware Laptop with Intel core i7, 7th Gen, 2.8 GHz processor and a GTX 1070 mobile graphics card.

4 Dataset and Experiments

Dataset: Several Datasets exist for pedestrian detection and tracking[4]. In particular, the Princeton Tracking Benchmark [32] provides a unified RGBD dataset for object tracking which includes various occlusions and some appearance changes. But, each sequence is very short (maximum 900 frames, most of them are under 300 frames). Many other works exist that aim at solving the person following problem, but there is a lack of a standardized dataset which could be used to validate the tracking algorithm used for person following robots. In this work, we built an extensive stereo dataset (left, right, and depth images) of 9 indoor and 2 outdoor sequences. Each sequence has more than 2000 frames and up to approximately 12000 frames. The dataset has challenging sequences which have pose changes, intense illumination changes, appearance changes (target removing/wearing a jacket, exchanging jacket with another person, removing/wearing a backpack or picking-up/putting-down an object), crouching and walking, sitting on a chair and getting up, partial and complete occlusions, occlusions by another person wearing same clothes and some other different situations. The dataset also has image sequences when the target is not visible transiently in the image and reappears after some time. The dataset is built in different indoor and outdoor environments in a university context. Some of the samples from the dataset can be seen in Fig. 6. The images are captured at a frame rate of 20 Hz

[4] http://homepages.inf.ed.ac.uk/rbf/CVonline/Imagedbase.htm#people.

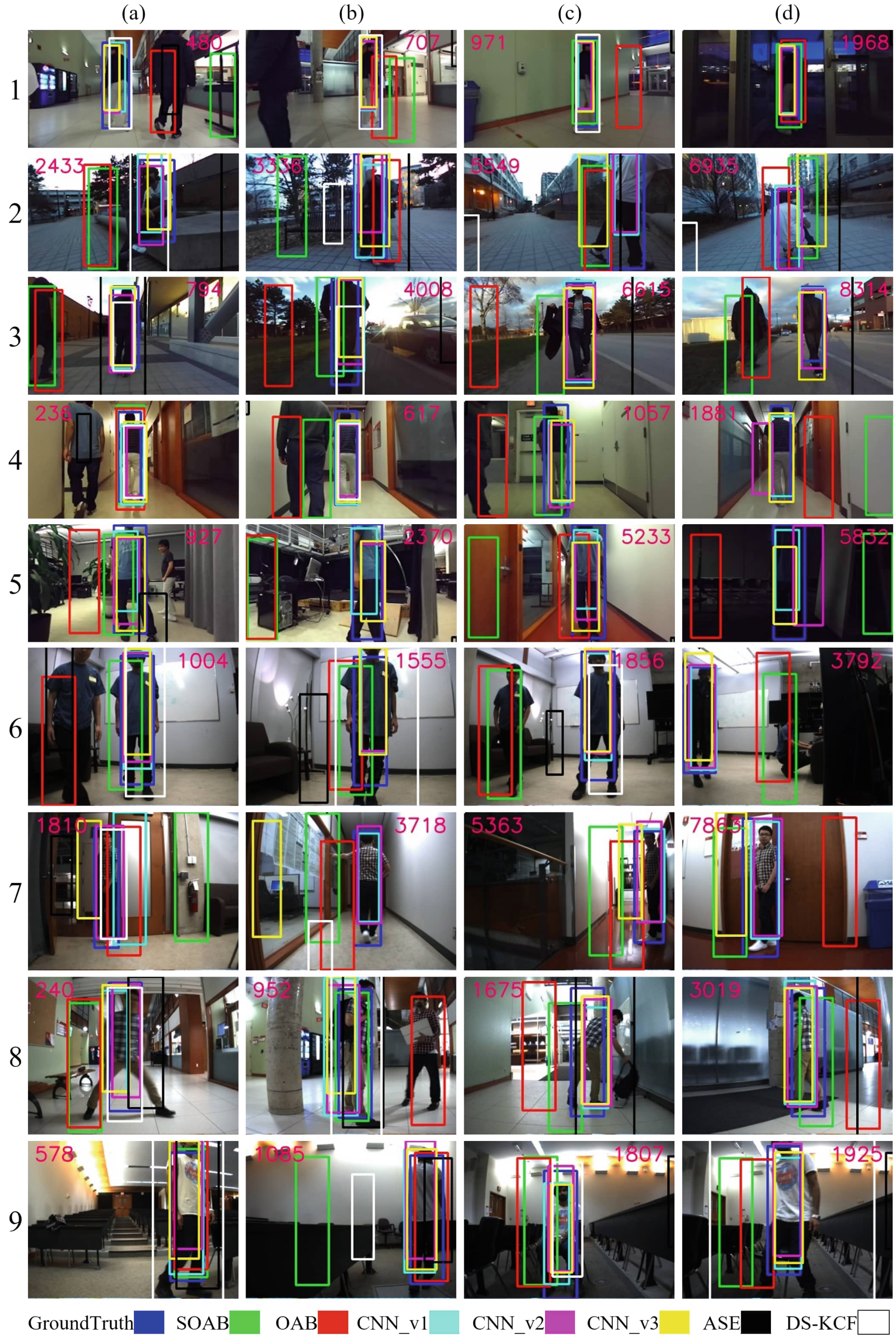
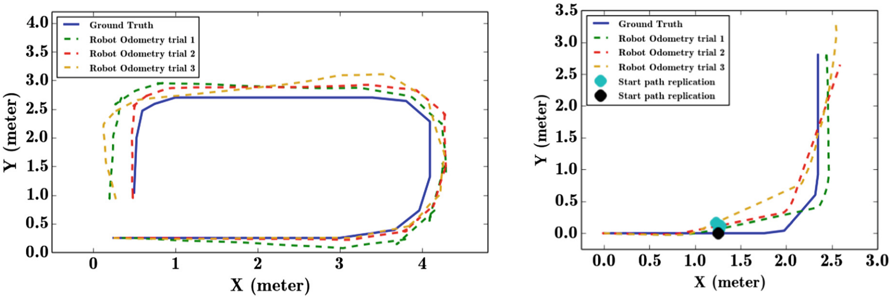


Fig. 6. Compare some tracking algorithms on our dataset. (1): Hallway 2; (2): Walking Outdoor; (3): Sidewalk; (4): Corridor Corners; (5): Lab & Seminar; (6): Same Clothes 1; (7): Long Corridor; (8): Hallway 1; (9): Lecture Hall. (SOAB [6], OAB [18], ASE [11], DS-KCF [5])

and the resolution is standard VGA (640 × 480) for bumblebee2 and (672 × 376) for ZED. We also provide with ground truth of the image sequences[5]. The ground truth contains the bounding box labeled for the target (human) which is manually labeled by human annotators for each frame.

Evaluation Metric: The interest of person following task is to follow a person, so the size of the bounding box is not important for the robot. However, the centroid of the target plays an important role. The evaluation of tracking algorithms has been done in numerous ways. Wu et al. [35] provide details about various existing evaluation metrics that have been used for tracking. For our dataset we use the *precision-plot* as defined in [35] as the metric to evaluate the performance of our approach. We report the percentage of frames in which center of the detected bounding box is within a specific range of pixels from the ground truth (See Fig. 8). Since the initial bounding box size is about (100 × 350) for all the video sequences, we compute the average precision of all sequences using location error threshold 50 pixels to evaluate tracker performance(see Fig. 9(a)). Figure 9(b) shows the average precision plot over all sequences from Fig. 8.



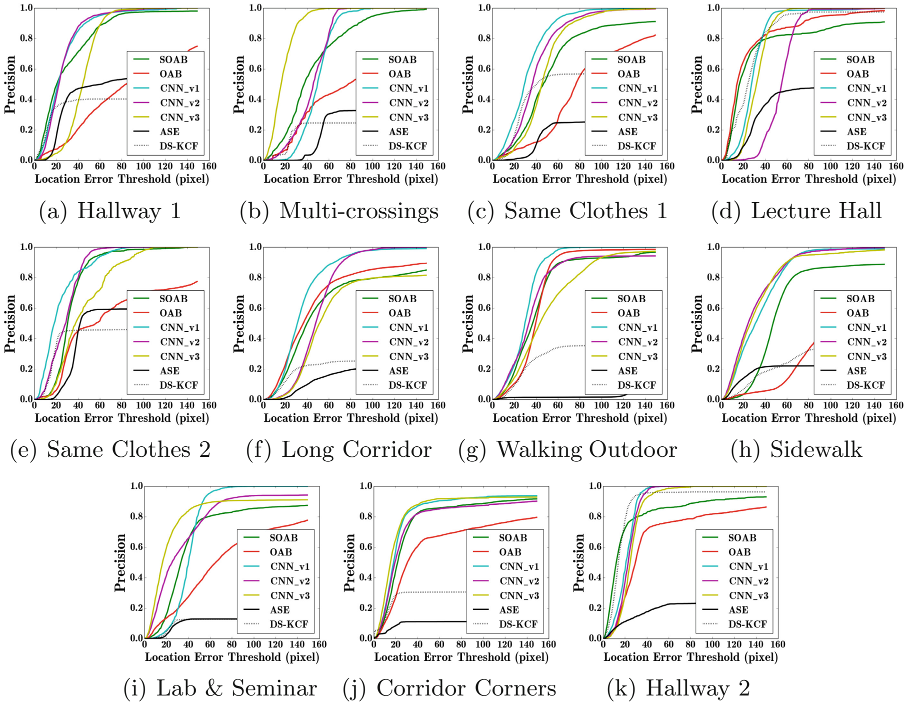
(a) normal person following case

(b) path replication case

Fig. 7. Overall performance of our robot system. (a) *Ground truth* is the path the robot should have taken ideally maintaining a 1-m distance from the target. *Robot Odometry trials* are the robot paths based on wheel odometry. (b) *Ground truth* is the same as the human path we are testing the path replication behaviour here. We have a maximum error (includes tracking, control, and wheel odometry errors) of roughly 30 cm which is not high for our task.

Experiments: We validated our proposed approach in different indoor and outdoor environments. We achieved a frame rate of approx. 20 fps depending on the search window size that we use for the depth range and the local image search region. For evaluation, we compare 3 versions of our tracking algorithm with 4 other existing stereo vision based trackers (for which the code is publicly available). We used the *precision-plot* evaluation metric as defined in [35] to report

[5] demo videos and dataset available at http://jtl.lassonde.yorku.ca/2017/05/person-following-cnn/.

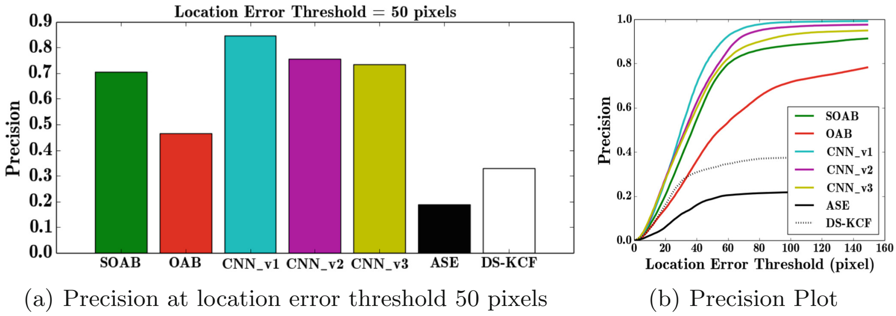


(a) Hallway 1 (b) Multi-crossings (c) Same Clothes 1 (d) Lecture Hall

(e) Same Clothes 2 (f) Long Corridor (g) Walking Outdoor (h) Sidewalk

(i) Lab & Seminar (j) Corridor Corners (k) Hallway 2

Fig. 8. *Precision-plots*: comparison between our trackers and different tracking algorithms, SOAB [6], OAB [18], ASE [11], DS-KCF [5]



(a) Precision at location error threshold 50 pixels (b) Precision Plot

Fig. 9. Comparison over 11 sequences (SOAB [6], OAB [18], ASE [11], DS-KCF [5])

the performance of our system. The performance can be seen in Figs. 6, 8, and 9. We evaluated the performance of our approach on 11 challenging sequences which exhibit varying situations as described in the previous section. It was found that the RGBSD based CNN (CNN_v1) outperformed all other existing approaches. The RGB based CNN (CNN_v3) could not perform better than SOAB [6] in some sequences. We also compare our approach with Danelljan et al. [11] (ASE with monocular images) and Camplani et al. [5] (DS-KCF with RGBD images). We show the performance of our overall robot system in Fig. 7. A demo video of our approach on the robot under different situations can be found at the link (See footnote 5).

5 Conclusion and Future Work

In this paper, we described a robust person following robot system using an online real-time Convolutional Neural Network in the context of robotics. The proposed system could perform very well in dynamic environments under challenging situations. The presented approach could find the person even when the robot could not see it by replicating the local trajectory of the target being followed. Possible future work includes incorporating dynamic obstacle avoidance techniques with the person following robot to give it more intelligence. Person following could also be addressed for places with known maps like using a social robot to follow people in a specific house, malls, retail stores and other places.

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. Ferreira, B.Q., Karipidou, K., Rosa, F., Petisca, S., Alves-Oliveira, P., Paiva, A.: A study on trust in a robotic suitcase. In: Agah, A., Cabibihan, J.-J., Howard, A.M., Salichs, M.A., He, H. (eds.) ICSR 2016. LNCS, vol. 9979, pp. 179–189. Springer, Cham (2016). doi:10.1007/978-3-319-47437-3_18
2. Awai, M., Shimizu, T., Kaneko, T., Yamashita, A., Asama, H.: Hog-based person following and autonomous returning using generated map by mobile robot equipped with camera and laser range finder. In: Lee, S., Cho, H., Yoon, KJ., Lee J. (eds.) Intelligent Autonomous Systems 12, Advances in Intelligent Systems and Computing, vol. 194, pp. 51–60. Springer, Heidelberg (2013). doi:10.1007/978-3-642-33932-5_6
3. Borenstein, J., Feng, L.: Umbmark: a benchmark test for measuring odometry errors in mobile robots. In: Photonics East 1995, pp. 113–124. International Society for Optics and Photonics (1995)
4. Calisi, D., Iocchi, L., Leone, R.: Person following through appearance models and stereo vision using a mobile robot. In: VISApp Workshop on Robot Vision, pp. 46–56 (2007)
5. Camplani, M., Hannuna, S.L., Mirmehdi, M., Damen, D., Paiement, A., Tao, L., Burghardt, T.: Real-time RGB-D tracking with depth scaling kernelised correlation filters and occlusion handling. In: British Machine Vision Conference, Swansea, UK, 7–10 September 2015. BMVA Press (2015)
6. Chen, B.X., Sahdev, R., Tsotsos, J.K.: Person following robot using selected online Ada-boosting with stereo camera. In: 2017 14th Conference on Computer and Robot Vision (CRV), pp. 48–55. IEEE (2017)

7. Chen, Z., Birchfield, S.T.: Person following with a mobile robot using binocular feature-based tracking. In: IROS 2007. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 815–820. IEEE (2007)
8. Chivilò, G., Mezzaro, F., Sgorbissa, A., Zaccaria, R.: Follow-the-leader behaviour through optical flow minimization. In: Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2004, vol. 4, pp. 3182–3187. IEEE (2004)
9. Cosgun, A., Florencio, D.A., Christensen, H.I.: Autonomous person following for telepresence robots. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 4335–4342. IEEE (2013)
10. Couprie, C., Farabet, C., Najman, L., Lecun, Y.: Indoor semantic segmentation using depth information. In: International Conference on Learning Representations (ICLR 2013), April 2013 (2013)
11. Danelljan, M., Häger, G., Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference, Nottingham, 1–5 September 2014. BMVA Press (2014)
12. Doisy, G., Jevtic, A., Lucet, E., Edan, Y.: Adaptive person-following algorithm based on depth images and mapping. In: Proceedings of the IROS Workshop on Robot Motion Planning (2012)
13. Eitel, A., Springenberg, J.T., Spinello, L., Riedmiller, M., Burgard, W.: Multimodal deep learning for robust RGB-D object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 681–687. IEEE (2015)
14. Fan, J., Xu, W., Wu, Y., Gong, Y.: Human tracking using convolutional neural networks. IEEE Trans. Neural Netw. **21**(10), 1610–1623 (2010)
15. Fuentes-Pacheco, J., Ruiz-Ascencio, J., Rendón-Mancha, J.M.: Visual simultaneous localization and mapping: a survey. Artif. Intell. Rev. **43**(1), 55–81 (2015)
16. Gao, C., Chen, F., Yu, J.G., Huang, R., Sang, N.: Robust visual tracking using exemplar-based detectors. IEEE Trans. Circuits Syst. Video Technol. **27**(2), 300–312 (2015)
17. Gao, C., Shi, H., Yu, J.G., Sang, N.: Enhancement of elda tracker based on cnn features and adaptive model update. Sensors **16**(4), 545 (2016)
18. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via on-line boosting. In: Proceedings of the British Machine Vision Conference 2006, Edinburgh, pp. 47–56 (2006)
19. Gupta, S., Girshick, R., Arbeláez, P., Malik, J.: Learning rich features from RGB-D images for object detection and segmentation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8695, pp. 345–360. Springer, Cham (2014). doi:10.1007/978-3-319-10584-0_23
20. Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.M., Hicks, S.L., Torr, P.H.: Struck: structured output tracking with kernels. IEEE Trans. Pattern Anal. Mach. Intell. **38**(10), 2096–2109 (2016)
21. Hong, S., You, T., Kwak, S., Han, B.: Online tracking by learning discriminative saliency map with convolutional neural network. In: ICML, pp. 597–606 (2015)
22. Hua, Y., Alahari, K., Schmid, C.: Online object tracking with proposal selection. In: The IEEE International Conference on Computer Vision (ICCV), December 2015
23. Kanbara, M., Okuma, T., Takemura, H., Yokoya, N.: A stereoscopic video see-through augmented reality system based on real-time vision-based registration. In: Proceedings of IEEE Virtual Reality, pp. 255–262. IEEE (2000)

24. Kobilarov, M., Sukhatme, G., Hyams, J., Batavia, P.: People tracking and following with mobile robot using an omnidirectional camera and a laser. In: Proceedings 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, pp. 557–562. IEEE (2006)
25. Koide, K., Miura, J.: Identification of a specific person using color, height, and gait features for a person following robot. Robot. Auton. Syst. **84**, 76–87 (2016)
26. Nishimura, S., Itou, K., Kikuchi, T., Takemura, H., Mizoguchi, H.: A study of robotizing daily items for an autonomous carrying system-development of person following shopping cart robot. In: 9th International Conference on Control, Automation, Robotics and Vision, ICARCV 2006, pp. 1–6. IEEE (2006)
27. O'Dwyer, A.: Handbook of PI and PID Controller Tuning Rules. World Scientific, Singapore (2009)
28. Oron, S., Bar-Hillel, A., Levi, D., Avidan, S.: Locally orderless tracking. Int. J. Comput. Vis. **111**(2), 213–228 (2015)
29. Sardari, F., Moghaddam, M.E.: A hybrid occlusion free object tracking method using particle filter and modified galaxy based search meta-heuristic algorithm. Appl. Soft Comput. **50**, 280–299 (2017)
30. Satake, J., Chiba, M., Miura, J.: A sift-based person identification using a distance-dependent appearance model for a person following robot. In: 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 962–967. IEEE (2012)
31. Schlegel, C., Jaberg, H., Schuster, M.: Vision based person tracking with a mobile robot. In: Proceedings of British Machine Vision Conference. Citeseer (1998)
32. Song, S., Xiao, J.: Tracking revisited using RGBD camera: unified benchmark and baselines. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 233–240 (2013)
33. Takemura, H., Ito, K., Mizoguchi, H.: Person following mobile robot under varying illumination based on distance and color information. In: IEEE International Conference on Robotics and Biomimetics, ROBIO 2007, pp. 1500–1505. IEEE (2007)
34. Tarokh, M., Ferrari, P.: Case study: robotic person following using fuzzy control and image segmentation. J. Field Robot. **20**(9), 557–568 (2003)
35. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015)
36. Yamane, T., Shirai, Y., Miura, J.: Person tracking by integrating optical flow and uniform brightness regions. In: Proceedings of 1998 IEEE International Conference on Robotics and Automation, vol. 4, pp. 3267–3272. IEEE (1998)
37. Yoshimi, T., Nishiyama, M., Sonoura, T., Nakamoto, H., Tokura, S., Sato, H., Ozaki, F., Matsuhira, N., Mizoguchi, H.: Development of a person following robot with vision based target detection. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5286–5291. IEEE (2006)
38. Zhai, M., Roshtkhari, M.J., Mori, G.: Deep learning of appearance models for online object tracking. arXiv preprint arXiv:1607.02568 (2016)
39. Zhang, K., Zhang, L., Yang, M.H.: Real-time object tracking via online discriminative feature selection. IEEE Trans. Image Process. **22**(12), 4664–4677 (2013)
40. Zhang, L., Suganthan, P.N.: Visual tracking with convolutional neural network. In: 2015 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2072–2077. IEEE (2015)
41. Zhang, L., van der Maaten, L.: Structure preserving object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1838–1845 (2013)