Proceedings of 2017 IEEE
International Conference on Mechatronics and Automation
August 6 - 9, Takamatsu, Japan

A Human-Following Approach Using Binocular Camera

Lei Pang, Leijie Zhang, Yingying Yu, Junzhi Yu, Zhiqiang Cao, Chao Zhou
State Key Laboratory of Management and Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
University of Chinese Academy of Sciences, Beijing 10049, China.
{panglei2015, zhangleijie2014, yuyingying2015, junzhi.yu, zhiqiang.cao, chao.zhou}@ia.ac.cn

Abstract – **This paper presents a tracker using binocular camera capable of tracking a specific human in both indoor and outdoor environments. Based on the kernelized correlation filter (KCF), the proposed tracker detects the target in multiple scales with the integration of HoG, color naming (CN), and local depth patterns (LDP) features, which improve the tracking performance. Specifically, a precise positioning method termed depth analysis is introduced to effectively overcome the problem of template drift. In addition, a motion controller is implemented to guide the motion of the mobile robot. The effectiveness of the proposed tracker is verified through video experiments and the robotic experiment.**

Index Terms – Human-following, visual tracking, binocular camera, kernelized correlation filter, depth analysis.

I. Introduction

Through decades of research and development, robots are not only active in factory assembly lines, they also move to every aspect of our daily life. Mobile robots exhibit strong abilities to adapt to various environments and can help humans accomplish many complex tasks. To facilitate interaction with humans, the ability of following human is an important attribute for mobile robots. Recently, many human-following robots have been proposed. Gupta *et al.* designed a visual-based human-following mobile robot with a robust tracking algorithm [1]. Babaians *et al.* presented a human-following robot [2], which follows humans through skeleton tracker with state-of-the-art OpenTLD visual tracker using Kalman filter. Hoshino and Morioka designed a human-following robot [3] based on a laser range scanner and a kinect sensor.

The main task for the human-following robots is to detect and follow the target human without missing the target. The target detection and tracking are crucial for the task execution. Many methods with different sensors are proposed to detect the target. Yoo *et al.* proposed detection and tracking approaches for human legs using a single LRF [4]. Jung *et al.* employed a support vector data description to detect a human by using an LRF [5]. Besides, bearing-sensitive PIR sensor arrays are also used in the human-following task [6]. Visual-based methods have been widely used where cameras can provide abundant environmental information [1, 7].

In the aspect of visual object tracking, some state-of-the-art RGB trackers have achieved high-speed tracking, such as L1T [8], CT [9], Struck [10], TLD [11]. The kernelized correlation filters tracker (KCF) [12] is proposed by Henriques *et al.* in 2015 with high accuracy and high efficiency. Despite the KCF tracker cannot handle the problem of scale changes, it still has outstanding performances, compared to other state-of-the-art approaches. Owing to the high efficiency, the KCF tracker can be further developed without the pressure of processing speed.

In reality, human-following mobile robots face some challenges including variations in illumination, pose changes of the target, and inconstant walking speed of the target. If the 3D information can be provided, the afore-mentioned challenges shall be better solved. Yoon *et al.* presented a RGB-D-based visual target tracking method with the combination of RGB and depth information [13]. Sun *et al.* proposed a detection and tracking approach using RGB-D camera, which can be used for the service robots in indoor environments [14]. It should be noted that RGB-D cameras usually work in indoor environments, it is lack of adaptability in outdoor. In this case, binocular camera provides an effective solution.

In this paper, a human tracker using binocular camera is presented, which can work in both indoor and outdoor environments. On the basis of kernelized correlation filter (KCF), the features are integrated for target detection in three different scales. Besides, a new solution to reduce the template drift is introduced to improve the tracking performance.

The rest of the paper is organized as follows. Section II introduces the KCF tracker and multiple features adopted in the tracker. The proposed tracker is described in Section III in detail. In Section IV, experimental results are given. Finally, Section V concludes the paper.

II. KCF Tracker And Features

A. Kernelized Correlation Filter Tracker

In this section, the kernelized correlation filter (KCF) tracker is briefly reviewed [12]. Ridge regression method is adopted by KCF to train target detection classifier, since it admits a simple closed-form solution and can achieve performance that is close to more sophisticated methods, such as support vector machines [15]. The goal of training is to find a function $f(z) = w^T z$ that minimizes the squared error over samples x_i and their regression targets y_i,

$$\min_w \sum_i (f(x_i) - y_i)^2 + \lambda \|w\|^2 \tag{1}$$

where λ is a regularization parameter to control overfitting. The ridge regression has the closed-form solution:

$$w = \left(X^T X + \lambda I\right)^{-1} X^T y \tag{2}$$

978-1-5090-6759-6/17/$31.00 ©2017 IEEE

The circulant matrix trick can speed up the process of collecting all the translated samples around the target. With a base sample $x = (x_1, x_2, \ldots, x_n)$, a cyclic shift of x is $Px = [x_n, x_1, \ldots, x_{n-1}]$. The circulant matrix, $X=C(X)=\{P^u | u=0,1,\ldots,n-1\}$, is the concatenation of all the cyclic shift visual samples. A property of circulant matrices is that all circulant matrices can be made diagonally by the discrete Fourier transform (DFT), regardless of the generating vector x [16]. This property can be expressed as $X = Fdiag(\hat{x})F^H$, where F is the DFT matrix that does not depend on x, and $\hat{x} = \mathcal{F}(x)$ denotes the DFT of the generating vector. In the following, we will always use a hat ^ as shorthand for the DFT of a vector. Therefore, (2) can be converted to the frequency domain, as shown below: $\hat{w}^* = \frac{\hat{w}^* \odot \hat{y}}{\hat{x}^* \odot \hat{x}^* + \lambda}$, where $\hat{x}^*$ denotes the complex-conjugate of $\hat{x}$, and $\odot$ is defined as the element-wise product. Compared to the prevalent methods, this solution saves the computational cost of extracting patches explicitly and solving a general regression problem [12].

The kernel trick is introduced to improve performance in non-linear situation. Input sample x can be mapped to a non-linear feature-space $\varphi(x)$ with kernel trick, and w can be expressed by linear combination of the samples, $w=\sum_i \alpha_i \varphi(x_i)$. In the case of no-linear regression, $f(z) = w^T z = \sum_{i=1}^{n} \alpha_i \kappa(z, x_i)$, where $\kappa(z, x_i) = < \varphi(z), \varphi(x_i) >$ is the kernel function. For the most commonly used kernel function, the circulant matrix trick can also be used, and the dual space coefficients α can be expressed as below:

$$\hat{\alpha}^* = \frac{\hat{y}}{\hat{k}^{xx} + \lambda} \tag{3}$$

where k^{xx} is the first row of the kernel matrix and defined as kernel correlation in [12]. In this paper, we adopt the common nonlinear Gauss kernel $k^{xx'} = \exp(-\frac{1}{\sigma^2}\|x - x'\|^2)$, we can obtain the kernel correlation between vector x and x':

$$k^{xx'} = \exp\left(-\frac{1}{\sigma^2}\left(\|x\|^2 + \|x'\|^2 - 2\mathcal{F}^{-1}\left(\hat{x}^* \odot \hat{x}'\right)\right)\right) \tag{4}$$

As the algorithm only requires dot-product and DFT/IDFT, the computational cost is in $O(nlog\, n)$ time.

The circulant matrix trick can also be applied in detection to speed up the whole process. In the next frame, the patch z at the same location is treated as the base sample to compute the response in Fourier domain, and a response map can be obtained by:

$$y = \mathcal{F}^{-1}\left(\hat{k}^{xz} \odot \hat{\alpha}\right) \tag{5}$$

The position with the maximum value in y can be predicted as new position of the target.

B. *Multiple Features Integration*

Since the Gauss kernel or polynomial kernel is based on either dot-products or norms of the arguments, this allows to sum the result for each channel in the Fourier domain [12]. We can obtain the multi-channel form of (4) as follows:

$$k^{xx'} = \exp\left(-\frac{1}{\sigma^2}\left(\|x\|^2 + \|x'\|^2 - 2\mathcal{F}^{-1}\sum_c\left(\hat{x}_c^* \odot \hat{x}_c'\right)\right)\right) \tag{6}$$

This allow us to apply strong features rather than the raw greyscale pixels to enhance the performance of the tracker. In this paper, three types of features are used in the tracker.

1) Histogram of oriented gradients

Histogram of oriented gradients is one of the most popular visual feature descriptor used in computer vision and image processing for the purpose of object detection. The feature descriptor counts occurrences of gradient orientation in localized portions of an image. In the process of feature extraction, the image is divided into small connected regions called cells, and a histogram of gradient directions is compiled in each cell. Since HoG feature operates on local cells, it is invariant to geometric and photometric transformations, except for object orientation. In this paper, the 31-dimensional HoG features [17] are used in our method.

2) Color-naming

Color-naming, or color attributes, is a perspective space with the linguistic color labels assigned by humans to describe the color [18]. Compared to RGB space, the distance in color label space is more similar to human sense, and it achieves the competitive results in some visual tasks such as objection recognition [19] and object detection [20]. Weijer *et al.* employed the mapping method to transform the RGB space into the color names space [18], which is a 11-dimensional color representation. In this paper, we adopt the color names with 10 dimensions [20].

3) Local depth patterns

LDP is a novel approach to describe the local depth feature. The features are adopted to take better use of the depth information and have favorable effect in handling the scale variation. Awwad *et al.* presented the LDP feature and a tracker to track target solely on depth data [21]. LDP feature resembles LBP in that it computes differences between cells of a local patch.

Assume that the size of the patch is $M×N$ and the size of the cell is S_{cell}, the patch is divided into $H×V$ grid of cells so that $H=M/S_{cell}$ and $V=N/S_{cell}$. The average depth of each cell is computed. Then, we assemble 3×2 or 2×3 grid of cells as a block, and the number of the blocks in the whole image is N_{LDP}. The LDP features of each block is obtained by concatenating the differences between the average depths of each cell with every other cells in the block. Therefore, the dimension of the LDP features in each block is 15. The whole LDP feature is obtained by concatenate LDP features of all blocks.

III. The Proposed Tracker

A. *The Tracker*

The schematic block diagram of the human tracker is shown in Fig. 1. Firstly, the binocular camera captures left and right images simultaneously, and the depth images are generated with 10 fps by the SGBM algorithm [22], which can satisfy the real-time applications. According to the feature map, the KCF tracker executes fast kernel correlation and fast detection to get the response maps to detect the new positions of the target. Meanwhile, the depth analysis is conducted to get the depth response maps to modify the results of the KCF tracker. Utilizing the tracking results and the depth information, the motion controller generates commands to control the mobile

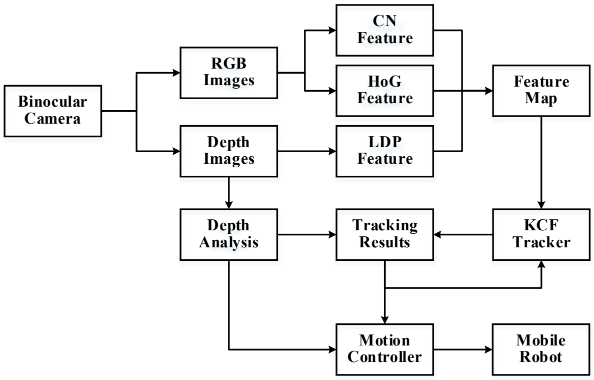


Fig. 1. The schematic block diagram of the proposed human tracker.

robot that follows the target human and keeps a special distance between them.

B. *Integration of Features and Multiple Scales*

1) Integration of Features

CN features, HoG features, and LDP features are extracted from the patches in the RGB images and depth images to construct the feature map. The patch is divided into $H\times V$ grid of cells to extract three types of features in cells, and the size of the cell is S_{cell}. In particular, the 6 cells in a block are described with the same LDP features, which are calculated in the same block. For the CN features, we calculate obtain the CN features of the cell based on the average values in each cell. Therefore, each cell in the patch contains 31-dimensional HoG features, 10-dimensional CN features, and 15-dimensional LDP features. Three types of features are integrated to a feature map, which is a matrix that contains 56 rows and $H\times V$ columns.

2) Multiple Scales

The original KCF tracker employs the templates with the fixed template size. Considering the inconstant walking speed of the target, the scales of the templates should be changed to prevent the template drift, or even the tracking failure. Generally speaking, there are two strategies that can deal with the scale variations: multiple scales detection and distance control between the target and the mobile robot, where the former is widely used.

For the proposed tracker, three different scales containing a normal scale, a bigger scale, and a smaller scale, are adopted to detect the new position of the target. We label the size of normal scale as S_n, which is the scale size of the last frame. With a magnification factor MF, the sizes of the bigger scale and the smaller scale can be expressed by $S_b=S_n\times MF$ and $S_s=S_n/MF$, respectively.

We can get three response maps in the three scales, and the corresponding maximum values. To increase the stability of the detection in multiple scales, when we calculate the maximum values in the three response maps, the response values need multiply a reducing parameter RP. And the new scale is which we get the detection position p^*. According to the detection position and the scale change, we can get the target image and use it to train the new template and parameters. It should be noted that although the motion control of the robot lags behind the template update of the KCF tracker, fix-distance control of the mobile robot with the target can reduce the negative influence of scale variations to some extent.

C. *The Depth Analysis*

Although multi-scale templates are adopted in the tracker, there still exists problem when the target suddenly changes its pose. In such a situation, the target information will occupy a small part in the patch, which easily causes template drift, or even tracking failure. Considering the fact that the depth distributions of the target and the background usually vary widely, it is favorable and reliable to use depth information to improve the precision and stability of the tracker. In this paper, the depth analysis method is introduced to improve the performance of the original KCF tracker. The schematic diagram of the areas used in the depth analysis is illustrated in Fig. 2.

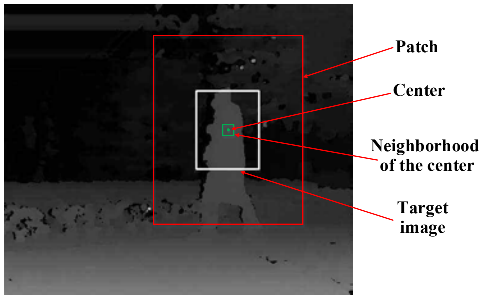


Fig. 2. The schematic diagram of the areas used in the depth analysis.

We label D as the depth estimation of the target in the current target image, and it is used to detect the target in the next frame. In every frame $\#t$, we calculate an average value D_t in a neighborhood of the target image's center to estimate the depth of the target. Usually we set the size of the neighborhood as 8×8. The updating strategy of D is as follows:

$$D=\begin{cases} D_t, & \left|D_t-D\right|<D_{gap} \\ D, & \left|D_t-D\right|\ge D_{gap} \end{cases} \tag{7}$$

where the constant D_{gap} is set to prevent the influence of drastic change of D that will affect the tracking stability.

The response map of the depth analysis can be calculated using D of the last frame and the current depth image. We assume that the depth distribution of the target approximately obeys a Gaussian distribution. Firstly, we divide the patch into $H\times V$ grid of cells. These cells have the same size with the cells of the feature map, so that we can get a matrix whose dimension is the same as the response map of KCF tracker. In every cell, the mean value of the depth is calculated and marked as $d(h, v)$. The depth response map R_{Dep} can be obtained through the following equation:

$$\phi\left(h,v\right)=\frac{1}{\sigma_d\sqrt{2\pi}}\exp\left(-\frac{\left(d\left(h,v\right)-D\right)^2}{2\sigma_d^2}\right) \tag{8}$$

The response map of the KCF tracker is R_{KCF}, we can get the weighted response map R_w:

$$R_w=w\times R_{KCF}+\left(1-w\right)\times R_{Dep} \tag{9}$$

Then, we find the maximum response value in R_w to obtain the tracking result.

Algorithm 1 shows the detailed steps of target detection with multiple scales and depth analysis. In the algorithm, *getFeature* is used to obtain the feature map where the inputs are RGB image, depth image, the patch, and the scale magnification factor. *getRgbRes* and *getDepRes* are employed to calculate KCF response map and depth response map, respectively. *detect* can get the maximum response value and corresponding position in every scale, and *getResult* is used to get the final tracking result based on the position, scale changing.

Algorithm 1 Target detection with multiple scales and depth analysis

```
Input: RGB image Img_rgb, depth image Img_dep, the detection patch Patch,
       the template of the last frame Tem.
Output: the tracking result Img_result of the current frame.
S[3]={1, MF, 1/MF};
RP[3]={1, 0.95, 0.95};
peak_value=0;
p_target=(0, 0);
for i = 0; i < 3; i ++ do
    feature_map=getFeature(Img_rgb, Img_dep, Patch, S[i]);
    rgb_response=getRgbRes(feature_map);
    depth_rsponse=getDepRes(Img_dep, Patch, S[i]);
    (max_res, p)=detect(Tem, rgb_response, depth_rsponse)
    if RP[i]×max_res>peak_value then
        peak_value=max_value;
        p_target=p;
        SN= S[i];
    end if
end for
Img_result=getResult(p_target, SN);
Tem=train(Img_result);    // training new template
```

D. Motion Controller

The human-following robot is required to track a moving human on a 2-D plane with a specified distance between them. The block diagram of the robot motion controller is shown in Fig. 3. The camera is mounted along the heading direction of the mobile robot. The coordinate frame of the camera is attached to the optical center of the camera, where z-axis is along the optical axis of the camera, and x-axis is parallel to the horizontal direction and perpendicular to the z-axis.

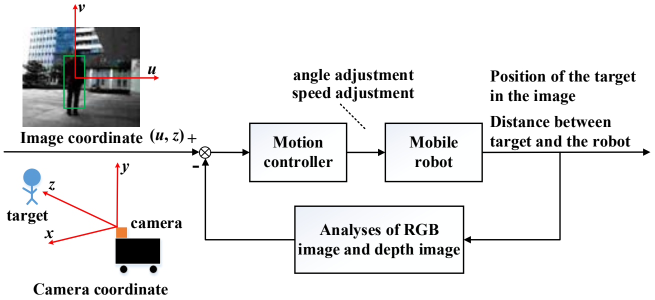


Fig. 3. The block diagram of the motion controller.

The motion controller contains two parts, the horizontal angle control and the speed control. The angle controller uses proportional control, which generates the horizontal angle adjustment according to the target position in u-axis of the image. PD control is used to adjust the speed of the robot based on the distance between the target and the camera. Because the parameters of the camera are known, the coordinates of the target in the binocular camera can be calculated easily.

IV. Experiments and Analysis

The performance of the proposed tracker is evaluated in this section. The experimental setup is shown in Fig. 4. The mobile robot is equipped with a i7 processor computer, a PointGrey BB2-08S2C-38 binocular camera, and the wheeled mobile robot platform. The binocular camera is used to obtain the RGB images and the depth images with the resolution of 640×480 pixels. The parameters used in the tracker are as follows: MF=1.05, RP=0.95, σ_d=1.5, w=0.8.

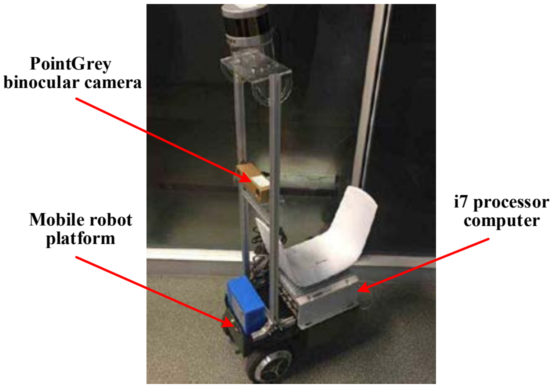


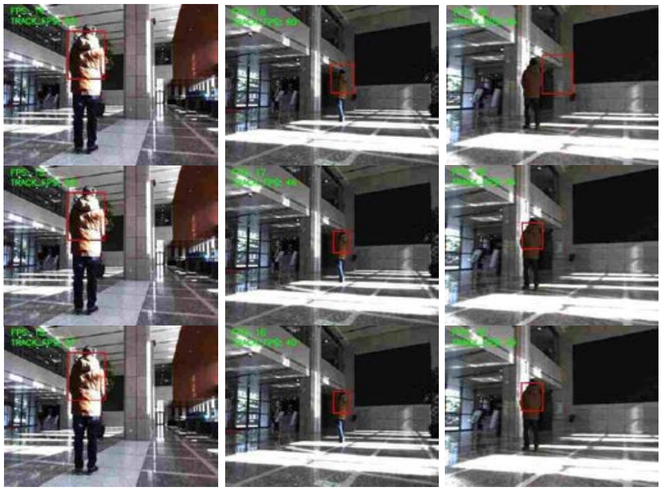
Fig. 4. The human-following mobile robot used for implementing the proposed human tracker.

A. Multiple Features Integration

In this section, we test the tracking effects with different feature combinations for a video of a building hall with complex illuminations. Three experiments are conducted, and the results are shown in Figs. 5 and 6. The first tracker uses only HoG features, the second tracker combines HoG features with CN features, while the third tracker adopts HoG features, CN features, and LDP features. Three experiments use the same target initialization in frame #1.

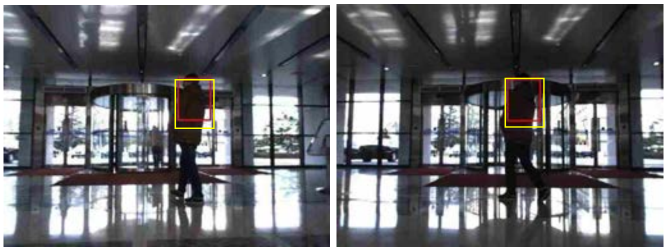
Firstly, the stabilities of the three trackers with different feature combinations are compared. In the frame #541, when the target human starts to turn left, the background information already occupy a large part in the tracking result of the first tracker with an increasing trend in the following frames. The target is completely missed in frame #609. However, the tracking performances of the other two trackers are accurate and stable. Fig. 6 gives the comparison of the second tracker and the third tracker. From frame #743 to frame #769, the distance between the target and the camera is decreased. Compared to the initial target bounding box in the first frame, the results of the third tracker are slightly better than those of the second tracker.

From the experiments conducted above, one can see that the tracker using only HoG features easily fails to track the target in these scenes containing distractions such as pillars, tree trunks, etc. The second tracker performs well in most scenes, but when the scale changes, it is slightly inferior to the third tracker. Therefore, the feature combination integrated HoG,



Frame #1 **Frame #541** **Frame #609**

Fig. 5. Tracking results of the proposed tracker using different feature combinations. The first row gives the result of the first tracker. The second row and the third row describe the results of the second and third trackers, respectively.



Frame #743 **Frame #769**

Fig. 6. Tracking results of the proposed tracker using different feature combinations. The results of the second tracker are marked with red boxes, whereas the results of the third tracker are marked with yellow boxes.

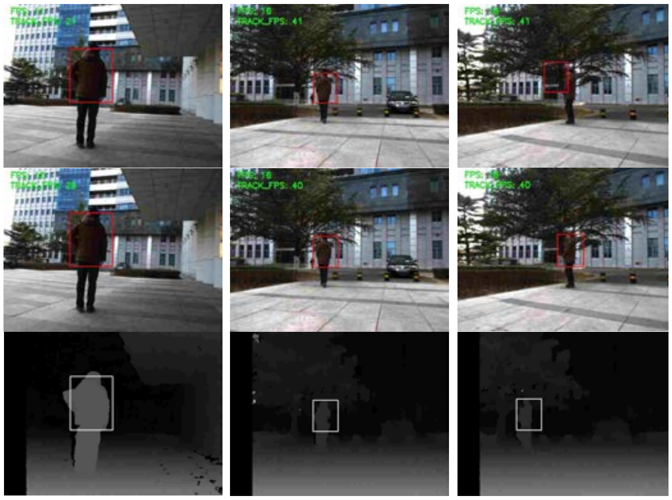
CN with LDP features is used in our tracker to provide a stable tracking.

B. *Depth Analysis and Precise Positioning*

The influence of the depth analysis in solving the problem of template drift is analyzed in this experiment. The results of the experiment are shown in Fig. 7. The video in this experiment is from outdoor environment.

Notice that the first row of Fig. 7 describes the results of the tracker without depth analysis, and the second row demonstrates the results of the tracker with depth analysis. In frame #1, two trackers use the same initialization. When the target human goes straight, two trackers both have stable performance. In frame #577, when the target human starts turning left, the background information at the edge of the target image begins to increase. Starting from frame #617, the tracker without depth analysis has a serious deviation with the target, and it fails to follow the target human. However, the tracker with depth analysis always stably follows the target.

The tracker with depth analysis performs more stably with a less template drift, especially in situations where the target human abruptly turns. It is favorable to use the depth information to improve the precision and stability of the tracker.



Frame #1 **Frame #577** **Frame #617**

Fig. 7. Tracking results of the experiment with/without depth analysis. The first row is the results of the first tracker without depth analysis. The second row is the result of the second tracker with depth analysis. The third row is the depth images of the second tracker.

C. *Mobile Robot Following a Human*

In this experiment, the mobile robot follows a target human in an outdoor environment. The experimental results are shown in Fig. 8. After the initialization is completed, the mobile robot follows the target with various movements. It is seen that the robot keeps following the target stably. The motion controller can keep the distance between the target and the mobile robot within a given range.

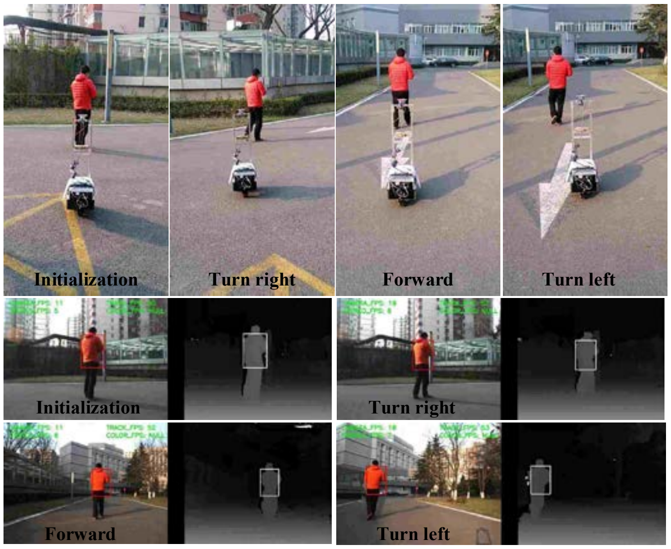


Fig. 8. Tracking results of the human-following experiment in an outdoor environment. The first row presents the results from the external viewpoint. The second and the third rows depict the tracking results in RGB images and depth images of the robot with its binocular camera.

V. Conclusion and Future Work

In this paper, we have presented a human tracker using binocular camera, which can control the robot to follow a specific human in both indoor and outdoor environments. Multiple features are used in the tracker to describe the target more accurately. Multiple scales and the depth analysis are added to the KCF tracker to improve the accuracy and stability of the tracker. Experiments verify the effectiveness of the proposed tracker. In the future, we will focus on improving the positioning precision of the depth analysis method. Besides, we will investigate the solution to tackle the occlusion problem by using the binocular camera.

Acknowledgement

This work was supported in part by the National High Technology Research and Development Program of China (863 Program) under Grant 2015AA042201, and in part by the National Natural Science Foundation of China under Grants 61633020, and in part by the Beijing Natural Science Foundation under Grant 4161002. Zhiqiang Cao is the corresponding author.

References

[1] M. Gupta, S. Kumar, L. Behera, and V. K. Subramanian, "A novel vision-based tracking algorithm for a human-following mobile robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, doi: 10.1109/TSMC.2016.2616343, 2016.

[2] E. Babaians, N. K. Korghond, A. Ahmadi, M. Karimi, and S. S. Ghidary, "Skeleton and visual tracking fusion for human following task of service robots," *International Conference on Robotics and Mechatronics*, 2015, pp. 761-766.

[3] F. Hoshino, and K. Morioka, "Human following robot based on control of particle distribution with integrated range sensors," *IEEE/SICE International Symposium on System Integration*, 2011, pp. 212-217.

[4] Y. Yoo, and C. Woojin, "Detection and following of human legs using the SVDD (support vector data description) scheme for a mobile robot with a single laser range finder," *International Conference on Electrical, Control and Computer Engineering*, 2011, pp. 97-102.

[5] E. J. Jung, J. H. Lee, B. J. Yi, J. Park, S. Yuta, and S. T. Noh, "Development of a laser-range-finder-based human tracking and control algorithm for a marathoner service robot," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 6, pp. 1963-1976, 2014.

[6] Y. Yang, G. Feng, S. Wang, X. Guo, and G. Wang, "Using bearing-sensitive infrared sensor arrays in Motion localization for human-following robots," *Asian Control Conference*, 2013, pp. 1-5.

[7] Y. Isobe, G. Masuyama, and K. Umeda, "Human following with a mobile robot based on combination of disparity and color images," *10th France-Japan/8th Europe-Asia Congress on Mecatronics*, 2014, pp. 84-88.

[8] X. Mei and H. Ling, "Robust visual tracking using ℓ_1 minimization," *IEEE International Conference on Computer Vision*, 2009, pp. 1436-1443.

[9] K. Zhang, L. Zhang, and M.-H. Yang, "Fast compressive tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 2002-2015, 2014.

[10] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M. M. Cheng, S. L. Hicks, and P. H. S. Torr, "Struck: structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096-2109, 2016.

[11] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409-1422, 2012.

[12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583-596, 2015.

[13] Y. Yoon, W. h. Yun, H. Yoon, and J. Kim, "Real-time visual target tracking in RGB-D data for person-following robots," *International Conference on Pattern Recognition*, 2014, pp. 2227-2232.

[14] Y. Sun, L. Sun, and J. Liu, "Real-time and fast RGB-D based people detection and tracking for service robots," *World Congress on Intelligent Control and Automation*, 2016, pp. 1514-1519.

[15] R. Rifkin, G. Yeo, and T. Poggio, "Regularized least-squares classification," *Nato Science Series Sub Series III Computer and Systems Sciences*, vol. 190, pp. 131-154, 2003.

[16] R. M. Gray, "Toeplitz and circulant matrices: a review," *Communications & Information Theory*, vol. 2, no. 3, pp. 155-239, 2005.

[17] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627-1645, 2010.

[18] J. v. d. Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1512-1523, 2009.

[19] F. S. Khan, J. Weijer, and M. Vanrell, "Modulating shape features by color attention for object recognition," *International Journal of Computer Vision*, vol. 98, no. 1, pp. 49-64, 2012.

[20] M. Danelljan, F. S. Khan, M. Felsberg, and J. v. d. Weijer, "Adaptive color attributes for real-time visual tracking," *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1090-1097.

[21] S. Awwad, F. Hussein, and M. Piccardi, "Local depth patterns for tracking in depth videos," *ACM International Conference on Multimedia*, 2015, pp. 1115-1118.

[22] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328-341, 2008.