Vision-Based Target-Following Guider for Mobile Robot

Mingyi Zhang, Xilong Liu, De Xu, *Senior Member*, *IEEE*, Zhiqiang Cao, *Senior Member*, *IEEE* and Junzhi Yu, *Senior Member*, *IEEE*

***Abstract*—A vision-based target-following guider for mobile robot is presented in this paper. It consists of three parts: the visual tracking part, the target re-detection part, and the visual servo part. In the visual tracking part, the target contour band (TCB) is proposed for irregular sampling, which can improve the performance of the correlation filter-based methods. In the target re-detection part, potential targets are searched by an off-line trained detector. Then an online training module is used to determine the real target and achieve accurate positioning. The interaction matrix of point features is used in the visual servo part for motion control to maintain the relative pose between the target and the robot. The visual tracking part and the target re-detection part are tested respectively on many videos, which are proved to work well. To show the effectiveness of our method, some state of the art methods are also test. The target-following guider is evaluated on mobile robots. In our experiments, the robot can robustly follow the human target over a long distance, which strongly proved the validity of our target-following guider.**

***Index Terms*—mobile robot, target following, target re-detection, visual servo**

I. Introduction

LEADER following is an important function of mobile robot, which is widely used in areas such as AGV, intelligent luggage, home service, etc. [1-3]. The target here usually refers to a human user but is not limited to humans. In recent decades, many scholars have been involved in this research. Some hardware and software modules are needed in a target-following guider. Algorithms used in the guider should be developed according to the sensors and computing platform of the mobile robot. For target-following guiders, vision sensors [4, 5] are used more commonly.

Binocular vision is used commonly because both vision information and depth information can be obtained [6, 7]. But, it has some application restrictions due to the low real-time, high computational complexity and so on. RGB-D is a newly developed visual sensor, which has been used in recent years [8]. But it is sensitive to sunlight disturbances, which makes it only suitable for fully enclosed indoor scenes.

Target-following methods based on binocular vision and RGB-D more or less rely on depth information. In fact, visual information is adequate to be used individually [9]. Chou *et al*. proposed a monocular vision based target tracking method for mobile robot platforms with the particle filter, called PSIPT [10]. The method uses the particle filter to track the target and assesses the distance from the target. In addition, an AdaBoost classifier for target detection is used. Husain *et al*. [25] proposed an automated system that is able to track and grasp a moving object within the workspace of a manipulator using range images. Real-time tracking is achieved by a geometric particle filter on the affine group. Hu *et al*. used monocular vision to achieve the tracking of a specific person indoors [11], which requires the color of the upper body of the person as prior knowledge. The histogram detection and the head-shoulder contour are used to improve the robustness of indoor tracking. Jean *et al*. proposed a robust visual servo system [26] for the object tracking application of a nonholonomic mobile robot, which consists of an adaptive shape tracking algorithm and a robust visual servo controller. The adaptive shape tracking algorithm is designed to automatically detect the shape contours of moving objects, extract the shape parameters, and continuously track the object in the shape parameter space. Stein *et al*. proposed a leader following method combining the Risk Rapid-exploring Random Tree and RiskRRT algorithm, to pass difficult paths through the guidance of human leader [27]. Zohir *et al*. combined the V-disparity obstacles detection approach with the U-disparity based localization and a fuzzy controller to execute all the steps of the leader following task [28]. Ess *et al*. proposed a mobile vision system for multi-person tracking in busy environments [12]. The system integrates the vision modules for visual odometry, pedestrian detection, depth estimation, and tracking. It integrates continuous visual odometry with tracking-by-detection to deal with frequent occlusions and egomotion of the camera rig.

A robust target tracking method is the core part of the target-following guider. There is significant overlap between the research of robot following and video tracking. Many methods in video tracking area are worth to be referenced. But not all methods can be applied to the mobile robot. Because the mobile robot has strict requirements on robustness, real-time, etc. During the designing of the guider, the state of the art methods have been concerned. In recent years, tracking methods based on correlation filter and deep convolution neural network (DCNN) have been highlighted. Among them, methods based on DCNN attract much attention for their excellent feature extraction ability. However, for most

Manuscript received May, 05, 2018; revised August 15, 2018 and November 11, 2018; accepted January 02, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61633020, 61421004, 61503376, 61733004, and the Beijing Natural Science Foundation under Grant 4161002.

M. Y. Zhang, X. L. Liu and D. Xu with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 101408, China (e-mail: zhangmingyi2014@ia.ac.cn, xilong.liu@ia.ac.cn, phone: +86134-2601-2076, de.xu@ia.ac.cn).

Z. Q. Cao and J. Z. Yu with the State Key Laboratory for Management and Control of Complex Systerms, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: zhiqiang.cao@ia.ac.cn, junzhi.yu@ia.ac.cn).

DCNN-based methods, with the limited computing resource of the mobile robot, they are difficult to meet the real-time requirement. Some improvements have been proposed to improve their real-time. In 2016, David presented the GOTURN tracker [23], which leveraging the benefits of offline training on extensive data to avoid online fine-turning and realize real-time (100FPS on GPU and 20FPS on CPU). However, compared with the classical methods, DCNN based trackers represented by GOTURN, which can meet the requirement of real-time, still have a gap in accuracy. In addition, methods based on correlation filter stand out by strong real-time performance and robustness. In 2010, Bolme *et al*. [13] proposed the MOSSE tracker based on a correlation filter. Following the idea of MOSSE, Martins *et al*. proposed the circulant structure of tracking-by-detection with kernels tracker (CSK), which solves the problem of dense sampling in mathematics [14]. Further, Martins *et al*. [15] proposed the KCF tracker incorporating the HOG feature and the kernel trick in the CSK tracker. Its frame rate can reach 100~400 frames per second (FPS), which lays the cornerstone of the mobile robot based application for correlation filter methods. Over these years, scholars have presented some excellent tracking methods by combining the advantages of DCNN and the correlation filter, such as CFCF [24]. Features learned by DCNN are used in the correlation filter based method. The tracking performance is improved due to the good quality of features. However, the real-time of these methods are not satisfactory, which is far from the requirements of the mobile robot.

In addition, a target re-detection part is indispensable for the target-following guider. In the field of re-detection, the existing methods can be divided into two categories. One is to use the offline trained detection method. Another is to add online training module in the tracker. In recent years, some following frameworks including the re-detection part are proposed.

In 2010, Kalal *et al*. developed a Tracking-Learning-Detection framework (TLD) for long-term tracking [18]. The tracker and detector of TLD run in parallel. The learned model reacts on the tracker and detector to update them in real time. Based on the KCF tracker, Ma *et al*. [16] proposed a remarkable long-term tracking method (LCT), which is a representative long-term method in recent years. It consists of three parts: the displacement detection part, the scale detection part, and the target re-detection part. The correlation filter is used in the displacement detection part and a scale detection part, the target re-detection part is realized by the online classification of random forest. Inspired by the Atkinson-Shiffrin Memory Model, Hong *et al*. proposed MUlti-Store Tracker (MUSTer), a dual-component approach consisting of short- and long-term memory is used to process target appearance memories [17]. The integrated correlation filter is employed in the short-term store for short-term tracking. The integrated long-term component can interact with the long-term memory and provide additional information for output control. For the memory module, the target is re-detected by searching for the most similar image features that are memorized.

Although there is a large overlap between robot following and video tracking, they have some significant differences. First, the bounding box in the robot following task is selected manually at the beginning, whose quality is much worse than those in video tracking. Second, when the target is lost, compared to video tracking, robot following can re-detect the target by active movements. Besides, factors such as camera shaking, motion limitations, etc. should be considered in the target-following task. Finally, a target-following guider is a complete integrated system, which requires effective cooperation between the tracking part, the re-detection part, and the robot control part.

For most of the video tracking methods, the minimum bounding rectangle is used as the bounding box. The target is expected to fill the bounding box as much as possible. Good tracking performances are depended on the assumption that the feature intensity of the background is weaker than that of the target region. For non-rectangular targets, especially those with non-convex contours, the target may be smaller than half of the bounding box. In this case, it is difficult for the trackers to distinguish between the background and the target. The tracking performance will decline significantly. A typical case is to track a triangle target, where the target area will never be greater than half of the bounding box. In theory, if the area ratio of the target is less than one-half, the tracking task will hardly be completed if the feature intensity of the background is stronger than the target. This is more prominent in trackers based on correlation filter. A strategy that can intelligently distinguish the background and the target in the bounding box might be an effective solution to these problems. In this paper, a target contour band (TCB) based screening mask is conducted to distinguish the target and the background, which can improve the tracking performance.

In the re-detection part, the success rate is improved by fusing the offline trained detection module with the online learning module, which combines each other's advantages while avoiding their disadvantages. In addition, the concept of active re-detection strategy is proposed. Compared with the re-detection in video tracking methods, the active re-detection strategy can preferentially control the robot turn to the lost direction of the target.

The control part is also essential for the target-following task. The image-based visual servo method is used to guide the robot.

The main contribution of our paper can be summarized as follows. First, a target-following guider for a mobile robot is proposed. Secondly, a TCB-based feature screening method is proposed to distinguish the target area and the background area for correlation filter-based methods. This is not only useful for the robot following, but also for video tracking. Finally, a target re-detection method is proposed. Combining an offline trained detector with a correlation filter based online training module, the success rate of re-detection is improved.

The rest of the paper is organized as follows. Section II broadly introduces the target-following guider. Section III is about the feature screening method proposed in this paper and its application in correlation filter methods. In Section IV, an efficient target re-detection method is developed. Section V tells about the details of the visual servo part. Different

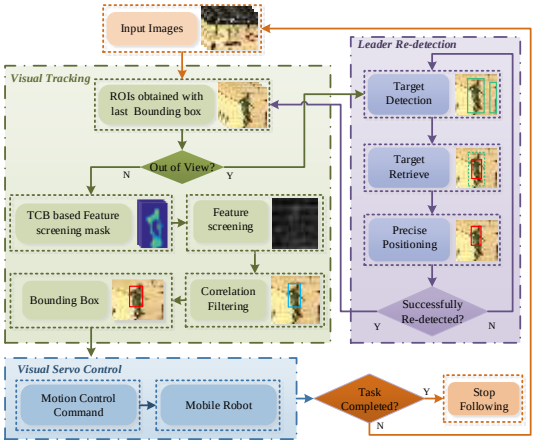


Fig. 1 Framework of the proposed target-following guider

experiments are designed to verify the effectiveness of the guider in section VI. Finally, section VII concludes the paper.

II. System Introduction

The structure of the target-following guider is shown in Fig.1. During following, the visual tracking part is used to capturing the target. The visual servo part is used to control the robot. If the target is in the field of view, the robot will be controlled to realize the synchronous motion with the target. If not, the robot will be controlled for active searching and the target re-detection part will be started. If the target is re-detected successfully, the visual tracking part will be continued.

A. The Visual Tracking Part

It is important to get enough training samples to ensure the tracking performance. Generating samples with circulant matrix is a good approximation of dense sampling, and correlation filter based trackers with circulant matrix have achieved good performances, such as KCF [21]. However, the traditional KCF tracker is not suitable for the long-term target-following task since the bounding box might easily drift from the target to the objects in the background, such as trees or pillars. This reflects the need that the background region and the target region should be further distinguished.

As is standard with correlation filters, the input image patches are weighted by a cosine window [29], [30]. Inspired by the idea of a weighted sliding window, an irregular sampling method based on TCB is proposed, which is the approximate contour area of the target. TCB can be gotten according to the feature stability map of samples. Unlike the precise contour which is consisted of edge chains, TCB is a band belongs to the approximate contour area that contains the real target contour or near it. More details are available in section III.

B. The Target Re-detection Part

The target-following guider needs but not limited to track the target and control the robot. Even humans cannot assure that the target will not be lost during the following. Once the target is lost, humans will actively move toward the lost direction and attempt to re-detect it. This experience can be taken as an imitation in the target-following guider.

An active re-detection strategy is proposed in this paper. The robot is controlled to search toward the loss direction proactively, and at the same time, the target re-detection method is started. The re-detection method includes two parts: a generic pedestrian detection part and a precise relocation part. More details are available in section IV.

C. The Visual Servo Part

An image-based visual servo control method is used in this part. Ultrasonic sensors are used as an auxiliary. The aim of the controller is to keep the target in the center of the view. Changes of target features reflect the changes of the relative position between the robot and the target. The visual servo part tries to make sure that the feature changes within a range and as small as possible. In general, the smaller the feature changes, the higher the robustness of the following is.

In our experiments, the interaction matrix of point features is used. The camera is installed in front of the robot. The original of the camera coordinate system is established at its optical center. Its z-axis is along the optical axis of the camera. The x-axis is parallel to the horizontal of the image. More details are introduced in section V.

III. Irregular Sampling and Robust Tracking

The robustness of correlation filter based trackers is largely affected by background features. With the TCB, effective features can be screened. The target contour is approximate as a band area with a certain width. It can be used to optimize the sample window and improve tracking performance.

In order to facilitate the descriptions of follow-up sections, some necessary concepts are introduced first.

Regional feature: For an image point, the statistical characteristic of the region surrounding it can be defined as its regional feature. The statistical characteristic can be the mean value of the region color, spatial frequencies and so on. The region scale refers to the side length or radius of the region.

Feature stability: Feature stability is a value that indicates the stability of a regional feature when the region scale or frame changes. The feature stability is obtained by comparing the region features of the same point between adjacent frames and different scales, as shown (1):

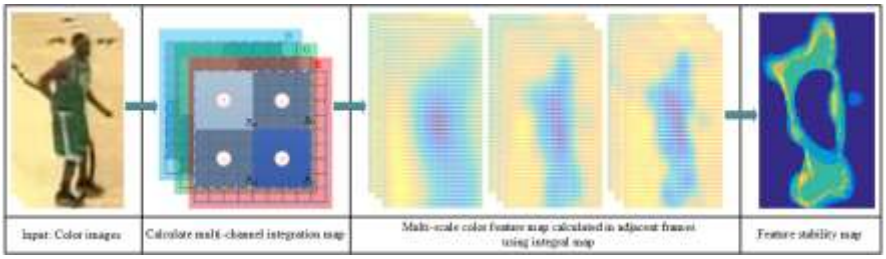


Fig. 2 Processing steps of scale stability map

$$D_F = -\sum_{n=-1}^{1}\sigma^n \cdot \sum_{l=1}^{L}\left\|\mathbf{F}_l^{t-n} - \frac{1}{L}\sum_{l=1}^{L}\lambda_l \mathbf{F}_l^{t-n}\right\| \quad (1)$$

where $\mathbf{F}_l^t$ is the feature vector at scale l in the t-th frame, λ_l is the corresponding weight factor, and L is the number of scales. σ^n is the weight factor of different frames. The feature used in our paper is the mean vector of the color feature.

Optimal path: Optimal path is a specific path connects two ridge points in the feature stability map. The length and average value are used to evaluate a path. For the length, the shorter, the better. For the average value, the larger, the better. All paths between a pair of ridge points form up a path set Ω_R. Mathematically, the optimal path R^* in Ω_R can be described as:

$$R^* = \arg\max_{R\in\Omega_R} \frac{1}{N_R^2}\sum_{n=1}^{N_R} D_F\left(u_{R,n}, v_{R,n}\right) \quad (2)$$

where N_R is the number of points in path R. $D_F(u_{Rn}, v_{Rn})$ is the scale stability value corresponding to the n-*th* point in R.

A. TCB Based Irregular Sampling

1) Feature Stability Map

In general, the scale term of feature stability in (1) is higher near region edges and is lower inside regions. The time term of (1) in the target region is higher than that in other regions, especially in the target-following task.

That is to say, values in the feature stability map are likely to be higher around the target-background border, and are lower in the background region. Values inside the target region are the lowest. According to this, effective information for target tracking can be conveniently screened.

The procedure of getting the feature stability map is shown in Fig. 2. In adjacent video frames, changes of regional features are calculated at different scales, which are used to determine the regional stability. It is known that the regional feature is statistical, which makes it robust to textures and noise, but sensitive to contours. That is to say, the feature stability of a regional feature has the following characteristics: (1) Regional features are more unstable near edge regions than other regions; (2) Near the target contour, feature stability values are continuous along the axial direction of the contour, while those along the normal direction decrease from inside to outside.

2) TCB Extraction

With property (1), we know that regions, where their feature stability values are lower than a certain threshold, can be considered as potential contour regions. It is known from the property (2) that the target contour should exist near ridges of the feature stability map. To extract the TCB, ridge points are selected in potential contour regions. A point chain C is formed with a series of ridge points and optimal paths between them.

It is not necessary to find the optimal path between every two ridge points, but only between the most possible adjacent ridge points, which can be called as each other's potential ridge point. The ridgeline can be used as the reference to search potential ridge points, which should meet the following constraints:

- Starting from the first selected ridge point, a fixed search direction should be selected as clockwise or counterclockwise;
- If there is no other ridge point on the path connecting the i-th ridge point p_i and the current ridge point p_c, then p_i can be considered as the adjacent ridge point of p_c;

$$\Omega_{ci}^l \bigcap \Omega_r = \{p_c, p_i\} \quad (3)$$

- The potential ridge point should be a point in the forward search direction of the current ridge point,

$$p_r \in \Phi_{se} \quad (4)$$

where Φ_{se} is a 180 °sector region in the search direction whose vertex is the current ridge point;

- The radius of the sector region r_Φ is determined by the closest adjacent ridge point of the current ridge point. If the distance between the two points is r, then $r_\Phi = 3r/2$.

In most cases, several point chains will be found, and constitute a point chain collection Ω_C. In Ω_C, the point chain that is closest to the target contour can be taken as the optimal point chain C^*. Since the target is the largest object in the sample image and is located near the center of the sample image, C^* should satisfy the following conditions:

(1) According to the length, point chains in Ω_C are divided into different levels. C^* should belong to the maximum level:

$$\begin{cases} Gra(C) = \left\lceil n\left(L(C) - L_{\min}\right)/\left(L_{\max} - L_{\min}\right)\right\rceil \\ C^* = \arg\max_{C\in\Omega_C} Gra(C) \end{cases} \quad (5)$$

where n is the total number of levels. $L(C)$ represents the number of pixel points of C. L_{max} is the maximum length of point chains in Ω_C, and L_{min} is the minimum one. $Gra(C)$ is the length level of C. $\lceil\ \rceil$ is the roundup operator.

(2) The point chain of the target should be able to form a loop by itself or with image boundaries. Inspired by isoperimetric inequality, (6) is selected as the screening criterion to get smooth enclosed areas and avoid interference features:

$$\begin{cases} f\left(A_C, L(C)\right) = 4\pi A_C^2 / L(C)^2 \\ C^* \in \arg\max_{C\in\Omega_C} f\left(A_C, L(C)\right) \end{cases} \quad (6)$$

where A_C is the connected region of C. When C is round, $f(A_C,L(C))$ has the largest value 1.

(3) The connection area of C^* should contain the center point of the sample image:

$$(u_t, v_t) \subset A_{C^*} \quad (7)$$

where (u_t, v_t) is the center point of the sample image. Points in C^* do not have to be the contour points, but should at least near them. To make sure that the contour points are included in the contour band as many as possible, a TCB is formed by checking points along the normal direction of C^*. The searching along the normal direction will stop, once there is a point whose feature stability value cannot meet the threshold ε_D. The average width of the filling area can be taken as the TCB width if it is larger than the threshold ε_w, otherwise, ε_w will be taken. The TCB is represented by its main curve and bandwidth.

In Fig.3, a(1) is the accurate target contour, a(2) is the approximation of the target contour that is acceptable for tracking. The point chains collection Ω_C is shown in b(1). Local remarkable points are uniformly selected as ridge points in the feature stability map. According to the constraints mentioned above, the optimal point chain C^* can be screened from Ω_C, as shown in b(2). The TCB then can be filled according to C^*. The main curve of the TCB is shown in c(1). It can be seen that the

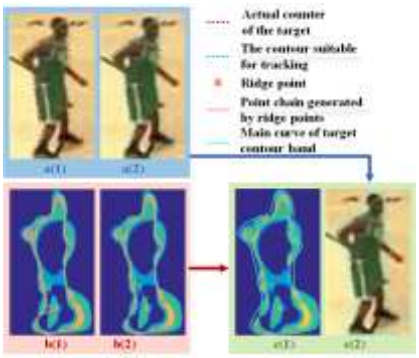


Fig. 3 Details of getting the target contour band

target contour is smoothly surrounded by the TCB in c(2).

The target contour changes as the target moves, so the TCB should be updated if necessary. Taking the main curve of the current TCB as a reference, the updating of TCB can be done by finding the nearest local maximum along the normal direction of the main curve. As shown in Fig.4, the feature stability map is meshed first and the main curve is divided into discrete segments. Along the normal direction of a segment, the grid's maximum response points are checked one by one. The nearest unchecked ridge point of maximum response points can be considered as the update point. When all checks are done, a new TCB region can be obtained. For tasks like the target-following that the shapes of the target are similar, TCB can be obtained in advance and be used universally.

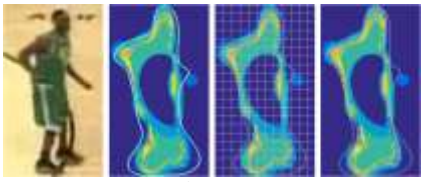


Fig. 4 Updating of the target contour band

B. Tracking with Irregular Sampling

Since the TCB is gotten, a weight assignment strategy can be used to approximate irregular sampling. With different feature weights, a screening mask can be structured, which is denoted as $\mathbf{m}_p$. Features in the target region have higher weights while those in the background region have lower weights. Weights in the TCB decrease from the inside to the outside.

For a sample $\mathbf{x}$, its screening matrix $\mathbf{m}_p$ is gotten by now. For samples around $\mathbf{x}$, the closer their locations are, the greater their response is. Responses $\mathbf{y}_i$ of these samples can be modeled as:

$$\mathbf{y}_i = \boldsymbol{\omega}^T \left(\mathbf{m}_p \odot \mathbf{x}_i \right) \tag{8}$$

where i=1, 2, …, n, n is the number of samples. $\boldsymbol{\omega}$ is the parameter matrix, which is trainable. To train $\boldsymbol{\omega}$, an optimization function is constructed as (9). The circulant matrix is used to obtain more training samples. According to the distance between the target center and the sample center, $\mathbf{y}_i$ meets Gaussian distribution.

$$\min_{\omega} \left\| \boldsymbol{\Gamma}\boldsymbol{\omega} - \mathbf{y} \right\|^2 + \lambda \left\| \boldsymbol{\omega} \right\|^2 \tag{9}$$

Where $\boldsymbol{\tau}_i = \mathbf{m}_p \odot \mathbf{x}_i$, $\boldsymbol{\Gamma}$ is a circulant matrix based on $\boldsymbol{\tau}_0$. $\boldsymbol{\tau}_i$ is the i-th sample in the screened matrix with i shifts. $\mathbf{y}_i$ is the response value of the i-th sample. λ is the regularization parameter. The Lagrange multiplier method is used to solve (9):

$$\boldsymbol{\omega} = \boldsymbol{\Gamma}^T \left[\lambda^{-1} \left(\mathbf{y} - \boldsymbol{\Gamma}\boldsymbol{\omega} \right) \right] \tag{10}$$

Let $\boldsymbol{\alpha}=\lambda^{-1}(\mathbf{y}-\boldsymbol{\Gamma}\boldsymbol{\omega})$, $\boldsymbol{\omega}$ can be expressed as:

$$\boldsymbol{\omega} = \boldsymbol{\Gamma}^T \boldsymbol{\alpha} = \sum_{i=1}^{n} \alpha_i \boldsymbol{\tau}_i \tag{11}$$

Then (9) can be rewritten as:

$$\min_{\boldsymbol{\alpha}} \left\| \mathbf{K}_{\tau}^{\mathbf{x}\mathbf{x}_0} \boldsymbol{\alpha} - \mathbf{y} \right\|^2 + \lambda \boldsymbol{\alpha}^T \mathbf{K}_{\tau}^{\mathbf{x}\mathbf{x}_0} \boldsymbol{\alpha} \tag{12}$$

where $\boldsymbol{\alpha}$ is the parameter vector. $\mathbf{K}_{\tau}^{\mathbf{x}\mathbf{x}_0}$ is the kernel matrix, $\mathbf{K}_{\tau}^{\mathbf{x}\mathbf{x}_0}=C(\mathbf{k}_{\tau}^{\mathbf{x}\mathbf{x}_0})$. $\mathbf{k}_{\tau}^{\mathbf{x}\mathbf{x}_0}$ is the kernel correlation vector of $\mathbf{x}_0$ and its circulant sample. For the linear kernel, $\mathbf{k}_{\tau}^{\mathbf{x}\mathbf{x}_0}$ has elements:

$$k_{\tau,i}^{\mathbf{x}\mathbf{x}_0} = \kappa_{\tau} \left(C\left(\mathbf{m}_p \odot \mathbf{x}_0 \right)_i, \mathbf{m}_p \odot \mathbf{x}_0 \right) = \left\langle C\left(\boldsymbol{\tau}_0 \right)_i, \boldsymbol{\tau}_0 \right\rangle \tag{13}$$

where $C(\boldsymbol{\tau}_0)$ is the circulant matrix corresponding to $\boldsymbol{\tau}_0$. $\boldsymbol{\tau}_0$ is the original sample, that is, it is obtained with no shift. $C(\boldsymbol{\tau}_0)_i$ is the i-th sample in $C(\boldsymbol{\tau}_0)$ with i shift. The solution to the kernelized version of (12) is given by:

$$\boldsymbol{\alpha} = (\mathbf{K}_{\tau}^{\mathbf{x}\mathbf{x}_0} + \lambda \mathbf{I})^{-1} \mathbf{y} \tag{14}$$

According to the properties of the circular matrix, (14) can be quickly calculated in the Fourier domain as follows:

$$\hat{\boldsymbol{\alpha}} = \frac{\hat{\mathbf{y}}}{\hat{\mathbf{k}}_{\tau}^{\mathbf{x}\mathbf{x}_0} + \lambda} \tag{15}$$

where $\wedge$ denotes the Discrete Fourier Transform (DFT) of the corresponding vector. For a new sample $\mathbf{z}$, we should evaluate the response $\mathbf{y}_i$ for every possible sample from every possible image locations to find the actual location. These samples can be gotten by cyclic shifts. $\mathbf{k}_{\tau}^{\mathbf{x}\mathbf{z}}$ is the kernel correlation vector of $\mathbf{x}$ and $\mathbf{z}$, respectively, each element of $\mathbf{k}_{\tau}^{\mathbf{x}\mathbf{z}}$ is given by $\kappa_{\tau}(C(\boldsymbol{\tau}_0)_i, \mathbf{m}_p \odot \mathbf{z})$. For the new sample $\mathbf{z}$, its response vector is:

$$\hat{\mathbf{y}} = \hat{\mathbf{k}}_{\tau}^{\mathbf{x}\mathbf{z}} \odot \hat{\boldsymbol{\alpha}} \tag{16}$$

$\mathbf{y}$ is the corresponding matrix of $\hat{\mathbf{y}}$ in the time domain. The position of the largest element in $\mathbf{y}$ can be considered as the displacement between the target position in the current frame and the previous frame.

From (15) and (16), we know that the calculation of $\mathbf{k}_{\tau}$ is the core part to train the tracker. The solution of the linear kernel function is described with (13). However, in most cases, the linear kernel cannot meet our need, so some other kernels are used to map the samples to a nonlinear feature space $\varphi(\boldsymbol{\tau})$.

$$\kappa_{\tau} \left(\boldsymbol{\tau}, C\left(\boldsymbol{\tau} \right)_i \right) = \left\langle \varphi\left(\boldsymbol{\tau}_j \right), \varphi\left(\boldsymbol{\tau}_i \right) \right\rangle \tag{17}$$

Dot-product kernels have the form $\kappa_{\tau}(\boldsymbol{\tau}, C(\boldsymbol{\tau})_i) = g(\boldsymbol{\tau} \cdot C(\boldsymbol{\tau})_i)$ for some function g. (18) can be written in vector form:

$$g\left(\boldsymbol{\tau} C\left(\boldsymbol{\tau} \right)_i \right) = g\left(\left(\mathbf{x}_i \odot \mathbf{m}_p \right) \left[C\left(\mathbf{x} \right) \odot C\left(\mathbf{m}_p \right) \right]_i \right) \tag{18}$$

What is perhaps the most amazing is the fact that all circulant matrices are made diagonal by the DFT, regardless of the generating vector, that is to say, (18) can be rewritten as:

$$\mathbf{k}_{\tau}^{\mathbf{x}\mathbf{x}} = g\left(F diag\left(\mathbf{x} \odot \mathbf{m}_p \right) F^H \left(\mathbf{x}_i \odot \mathbf{m}_p \right) \right) \tag{19}$$

where F is a constant matrix that does not depend on $\mathbf{x}$. Some commonly used kernels that meet the form of function g are the linear kernel, polynomial kernel, and Gaussian kernel.

In particular, for a polynomial kernel $\kappa_{\tau}(\boldsymbol{\tau}, C(\boldsymbol{\tau})_i) = (\boldsymbol{\tau}^T C(\boldsymbol{\tau})_i + a)^b$, it can be written as:

$$\mathbf{k}_{\tau}^{xx} = \left(\mathcal{F}^{-1}\left[\left(\mathbf{x} \odot \mathbf{m}_p^{*} \right) \odot \mathbf{x}_i \odot \mathbf{m}_p \right] + a \right)^{b} \quad (20)$$

For a Gaussian kernel $\kappa_{\tau}(\boldsymbol{\tau}, C(\boldsymbol{\tau})_i)=\exp(-\|\boldsymbol{\tau}^T - C(\boldsymbol{\tau})_i\|^2/\sigma^2)$, it can be written as:

$$\mathbf{k}_{\tau}^{xx} = \exp\left(-\frac{1}{\sigma^2}\left(\|\mathbf{x} \odot \mathbf{m}_p\|^2 + \|\mathbf{x}_i \odot \mathbf{m}_p\|^2 - 2\mathcal{F}^{-1}\left(\left(\mathbf{x} \odot \mathbf{m}_p^{*} \right) \odot \mathbf{x}_i \odot \mathbf{m}_p \right) \right) \right) \quad (21)$$

Now, the trackers with different kernel functions can be trained with irregular-like samples.

C. Scale Estimation

We have known from [25] that it is necessary to have an independent correlation filter dedicated to scale evaluation. [26] tells us that responses of other scales near the optimal scale are approximately distributed in a quadratic curve-like form. Based on this, an optimal scale estimation method is constructed. Considering the needs of the target-following task, the estimation method in [26] has been streamlined as:

$$y_s = as_t^2 + bs_t + c \quad (22)$$

where s_t is the target scale, and y_s is the corresponding response.

Responses of different scales near the current scale are put into the scale correlation filter model. At least 3 pairs of (s_t, y_s) are needed to calculate the parameters in (22). The scale where the derivative is zero can be considered as the best scale. For security, if the estimated scale is greater or less than 20% of the previous one, then the scale with the largest response will be selected as the optimal target scale.

IV. Target Re-detection

The processing progress of the proposed target re-detection method is shown in Fig.5. First, a generic detection method is used to search all potential targets. Then, an online learning module and some rules are used to identify the real target from the potential targets. If the target is successfully identified, the tracking area will be precisely re-located further.

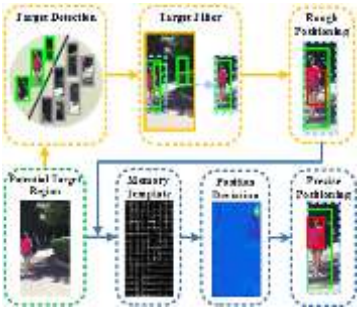


Fig.5 Target re-detection process

A. Target Searching

First, the generic detection module is performed. Because the target is usually human, many detection methods can meet the requirements, such as SVM, AdaBoost and so on. Results of these methods are commonly represented as rectangles including targets. These rectangles form up a set Ω. To choose the real target out of Ω, an online learning module is necessary.

In the online learning module, the continuously acquired target samples during tracking should be used to train the online detector. For the commonly used detectors, such as SVM and AdaBoost, online learning will seriously affect the real-time performance. In comparison, the calculation of the correlation filter based tracker is less expensive. It is known that trackers based on correlation filter are discriminative. These trackers solve the tracking problem by separating the target from the background. Therefore, the correlation filter based tracker is used as an online learning detector in our method. Integrated with some rule set, the re-detection accuracy is improved.

An independent correlation filter dedicated to screening targets is modeled in our paper. The parameter vector $\boldsymbol{\alpha}$ and history sample $\mathbf{x}$ are used, which are obtained during following. For each potential target, there is a corresponding sample $\mathbf{z}_j$, j=0, 1, 2,..., R. R is the number of elements in Ω. Its response vector $\mathbf{y}_j$ can be computed with (20) and (21). The maximum element value in $\mathbf{y}_j$ is expressed as y_j^*.

It is reasonable that the sample with the response y_j^* is the sample containing the real target. However, the sample with the highest response does not qualifying stand for the sample of the target. Under this strategy, if there is no real target in view, a false detection will be caused. Therefore, some reasonable assumptions are summarized to avoid it: the movements of the pedestrian are generally limited to the ground area; the shape of the target will not change largely in a short time. Based on these, some rules are proposed. The element in Ω who meets all the rules can be fully considered as containing the real target.

Rule1 If the target is out of view in a short time, the center point (u_l, v_l) of the target will not change largely when it is re-detected, that is:

$$(u_l, v_l) \subset \Omega^{j*} \quad (23)$$

where Ω^{j*} represents the j*-th element in Ω. Ω^{j*} is where the real target is located.

Rule2 The response of the element who contains (u_l, v_l) should be the highest in Ω:

$$j* = \arg\max_{j \in [1,R]} \left(y_j^* \right) \quad (24)$$

where y_j^* is the maximum element in $\mathbf{y}_j$.

Rule3 The maximum response y_{j*}^* should meet threshold ε_j:

$$y_{j*}^* - \varepsilon_j \geq 0 \quad (25)$$

ε_j can be set according to different applications.

B. Precise Target Re-locating

The accuracy of a tracking area directly affects the robustness of tracking. Therefore, it is necessary to precisely relocate the target and get an appropriate location for the tracking area. From section III, we know that the sampling window of the correlation filter trackers is larger than the actual tracking area, which introduces much background information. However, when it comes to relocating, the larger sampling window can make the tracking area be included as much as possible, which improves the accuracy.

It is able to roughly locate the tracking area and get the sample $\mathbf{z}$ with Ω^{j*}. The precise position of the tracking area then needs to be searched in $\mathbf{z}$. The dense samples gotten with the circulant matrix can greatly help to improve the re-locating accuracy. Using the parameter vector $\boldsymbol{\alpha}$ and the sample $\mathbf{x}$ memorized during tracking, the response vector $\hat{\mathbf{y}}_{j*}$ of sample $\mathbf{z}$ can be calculated as:

$$\hat{\mathbf{y}}_{j*} = \hat{\mathbf{k}}_{\tau}^{xz} \odot \hat{\boldsymbol{\alpha}} \quad (26)$$

The position (Δu, Δv) of maximum element $\hat{y}_{j*}^*$ in $\hat{\mathbf{y}}_{j*}$ can be

considered as the positional deviation between the real tracking center and the current sample center. The precise location of the tracking area then can be calculated in (27).

$$\left(u^*, v^*\right) = \left(u_l, v_l\right) + \left(\Delta u, \Delta v\right) \tag{27}$$

where (u_l, v_l) is the current sample center, (u^*, v^*) is the real tracking center.

It's important to note that the target here is unique and irreplaceable. If the leader is lost intermittently, the guider can re-detect the target with the re-detection part. If the target cannot be found, our guider will not look for an alternative target. The robot will stop moving for safety's sake, and the following task is interrupted.

V. IMAGE-BASED VISUAL SERVO CONTROL

A. Kinematic Model of the Mobile robot

A two-wheeled differential drive robot is used in our experiments. When the reference coordinate system {W} is established, its kinematic model can be described as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix} \tag{28}$$

where x, y are the coordinates of the robot kinematic center in{W}, θ is the orientation angle. The original point of the robot coordinate system {R} is the robot's kinematic center and its X-axis along the forward direction, as shown in Fig.6. v and ω are linear and angular velocities of the robot.

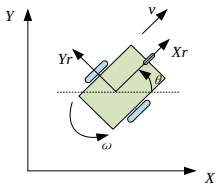


Fig.6 A two-wheel differential drive robot model

B. Image Jacobian Matrix Estimation

In the target-following guider, four vertices of the bounding box can be used as feature points. Assuming the lens distortion of the camera is small enough to be ignored, and the intrinsic parameters of the camera can be described using the pinhole model. For a point (x_n, y_n, z_n) in the camera's frame, its imaging point is denoted as $(x_{cn}, y_{cn}, 1)$. Generally, (x_n, y_n, z_n) is unknown in monocular vision. But x_{cn} and y_{cn} can be calculated from the image coordinates and the camera's intrinsic parameters, n=1, 2, 3….

The variation of a feature point on the normalized imaging plane is given in (29) when the camera moves [28]:

$$\begin{bmatrix} \dot{x}_{cn} \\ \dot{y}_{cn} \end{bmatrix} = J \begin{bmatrix} v_p \\ \omega_p \end{bmatrix} \tag{29}$$

where J is the interaction matrix.

$$J = \begin{bmatrix} -\frac{1}{z_n} & 0 & \frac{x_{cn}}{z_n} & x_{cn}y_{cn} & -\left(1+x_{cn}^{\ 2}\right) & y_{cn} \\ 0 & -\frac{1}{z_n} & \frac{y_{cn}}{z_n} & \left(1+y_{cn}^{\ 2}\right) & -x_{cn}y_{cn} & -x_{cn} \end{bmatrix} \tag{30}$$

where $v_p = [v_x, v_y, v_z]^{\mathrm{T}}$ is the translational of the camera, $w_p = [w_x, w_y, w_z]^{\mathrm{T}}$ is the angular velocity of the camera.

The robot in our experiments has two degree-of-freedoms in Cartesian space, its velocity vector is expressed as $[v_z, w_y]^{\mathrm{T}}$. Therefore, the interaction matrix in our paper can be written as:

$$\begin{bmatrix} \dot{x}_{c1} & \dot{y}_{c1} & \cdots & \dot{x}_{c4} & \dot{y}_{c4} \end{bmatrix}^{\mathrm{T}} = J_p \begin{bmatrix} v_z & \omega_y \end{bmatrix}^{\mathrm{T}} \tag{31}$$

$$J_p = \begin{bmatrix} \frac{x_{c1}}{z_1} & \frac{y_{c1}}{z_1} & \cdots & \frac{x_{c4}}{z_4} & \frac{y_{c4}}{z_4} \\ -\left(1+x_{c1}^{\ 2}\right) & -x_{c1}y_{c1} & \cdots & -\left(1+x_{c4}^{\ 2}\right) & -x_{c4}y_{c4} \end{bmatrix}^T \tag{32}$$

In (32), the calculation of x_{cn} and y_{cn} are calculated with the camera's intrinsic parameters and their image coordinates u and v. The camera's intrinsic parameters should be calibrated in advance. z_n corresponds to the target depth. It is noteworthy that, the target depth can be gotten with the ultrasound information and the scale estimation from the visual tracking part.

Therefore, the robot motion can be computed according to variations of the feature points on the normalized imaging plane and the interaction matrix J_p:

$$\begin{bmatrix} v_z & \omega_y \end{bmatrix}^T = -\lambda J_P^+ \begin{bmatrix} \Delta x_{c1} & \Delta y_{c1} & \cdots & \Delta x_{c4} & \Delta y_{c4} \end{bmatrix}^T \tag{33}$$

where J_p^+ is the pseudo-inverse matrix of J_p, λ is an adjustment factor.

C. Robot Control with Vision Feedback

Assuming that the current position of the feature points is $f(t)$, t he desired position is $f^*(t)$, then the system error $f(t)$-$f^*(t)$ on the normalized image plane can be defined as:

$$e_g(t) = \left[\dot{x}_{c1}(t), \dot{y}_{c1}(t), \cdots, \dot{x}_{c4}(t), \dot{y}_{c4}(t)\right]^T \tag{34}$$

The robot manipulated variable is [U(t)=v, W(t)=ω], the PI controller can be designed as:

$$\begin{bmatrix} \mathrm{U}(t) \\ \mathrm{W}(t) \end{bmatrix} = -\lambda J_P^+ \left[k_1 e_g(t) + k_2 \sum \Delta e_g\right] \tag{35}$$

where k_1, k_2 are the proportion coefficient and the integral coefficient. The control block diagram is shown in Fig.7.

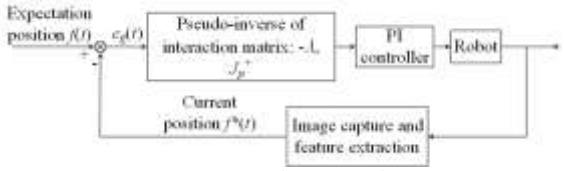
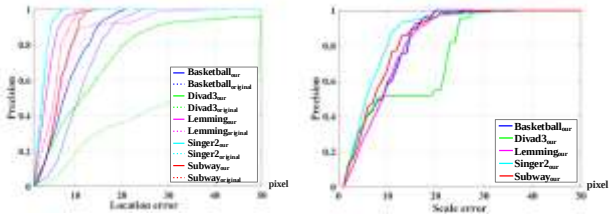


Fig. 7 Motion control part of the target-following guider

VI. EXPERIMENTS

Performances of the target-following guider are tested, including the individual performance and the overall performance. Before overall tests, independent tests are conducted. Some videos from the OTB dataset are used. In addition, videos from the robot perspective are collected for further experiments. For comparison, some state of the art methods, namely CFCF [30], GOTURN [29], TLD [22] are tested. Among them, TLD is specifically geared toward target re-detection and long-term tracking. We also migrated our system to robotic platforms for onboard tests. In total, over 30 videos are tested, including handheld video data and robot perspective video data. Experiments are introduced below.

A. Experiments with Database



(a) Precision plots (b) Scale estimation plots

Fig.8 T Performance curve on some OTB videos

The visual tracking part is tested first. To facilitate peer review, videos from the OTB dataset are used. As discussed in [27], the success plot is chosen to show the tracking performance. To intuitively show the tracking performances on different videos, the success plot is drawn in Fig.8(a). The KCF tracker [21] is chosen as a comparison method. Different colors represent for different videos. Solid lines indicate the effects of the proposed method while dotted lines represent that of the KCF method. Through experiments, we found that when the location error is greater than 20 pixels, it is difficult to ensure the effectiveness of the following task. Therefore, the tracking can be considered a success when the overall location error does not exceed 20 pixels. In Fig.8(a), almost all videos can be handled well within the required location error. The minimum accuracy of our method at the 20-pixel position is 0.8, and the average accuracy is 0.94. It shows that our method has higher accuracy than the original method.

The scale estimation performance of our method is also evaluated. The scale evaluation plot in Fig.8(b) demonstrates the effeteness of the scale estimation method. The scale error threshold represents its differences from the true scale. During following, for safety reasons, the tracking distance is generally controlled between 2m $\pm$ 0.5m. In other words, as long as the estimated error does not exceed 25 pixels, the scale estimation can be considered success. As can be seen from Fig. 8(b), when the error tolerance is located at 25 pixels, the success rate of our method can reach up to 95%.

To compare our method with CFCF, GOTURN, and TLD, representative videos from OTB are used. Some experiments are shown in Fig. 9. It can be seen that the correlation filter based methods, such as TLD and our method, are effective in dealing with occlusion, collision and illumination changes. Better features are used in GOTURN because of DCNN, but its performance is significantly reduced due to the confusion caused by other targets. Although CFCF uses features extracted by DCNN, it is still disturbed by the background clutter. In general, our method and LCT perform better.

Fig.9 Tracking performances on some OTB videos

Real-time performance determines whether the method can be applied to the mobile robot. As mentioned earlier, methods based on the correlation filter have good real-time performances. From Table I, it can be seen that the real-time and accuracy of our method are outstanding. The performance of LCT is also good, but it performs badly in videos based on the robot perspective, as shown in Fig. 10 and 11. Although the real-time of GOTURN is guaranteed compared to other DCNN-based methods, its accuracy is quite poor. CFCF performs well with good features from DCNN, but its real-time is unacceptable. Its average frame rate is about 1FPS, which is far from the requirement of the mobile robot.

Table I
Performance metrics on the all tested videos

Algorithm	Feature optimization	Mean precision	Mean scale accuracy	Mean FPS
Our method	yes	95%	94%	34
GOTURN[29]	no	87%	72%	22
CFCF[30]	yes	93%	93%	1
LCT[22]	no	91%	89%	27

The reported quantities are averaged over all tested videos in our experiments. Reported speeds include feature computation.

B. Experiments with Videos from the Robot Perspective

To evaluate the guider, many experiments are conducted, and over 12’016 frames are used. To be clear, at the beginning of all the following tasks, the tracking area is manually select.

1) Visual following experiments

Experiments with datasets are very consultative, but before transferred to a mobile robot, it is necessary to test the system on videos from the robot perspective first. As shown in Fig. 10, performances of GOTURN and LCT are not ideal. The bounding box drifts obviously, and the scale estimate is not accurate. In contrast, CFCF has better effects. However, it still has problems in drifting and scale estimation. Although the features obtained with DCNN are used to improve the tracking performance, it cannot suppress the disturbance of background features. In general, our method has better performance. It cannot only follow the target accurately but also estimate the

Fig.10 Tacking performances on long-term videos

scale change well. More importantly, our method has good real-time performance and is more suitable for a mobile robot.

Fig.11 Outdoor target-following experiment

To be more persuasive, the video from the perspective of a gait robot is taken as an example. The control of the gait robot is out of the scope of this paper. Therefore, the gait robot is controlled with a remote control device in this experiment. Only the target tracking part and re-detection part are executed. Fig.11 shows the performance of the target-following guider. It can be seen that, during the following, our method can achieve the effective following by overcoming interferences, like robot swaying, illumination changing, and so on. Fig.11(a) and (b) respectively represent images of the third-person perspective and the robot perspective.

2) Target re-detection experiments

To ensure the comprehensiveness of the re-detection experiments, videos containing different scenes are collected. In those videos, different targets are used, and they may be lost occasionally or habitually.

Fig.12 Experimental results of target re-detection

The target re-detection experiments are shown in Fig. 12. As can be seen, LCT cannot determine whether the target is out of view. Meanwhile, our method can accurately determine whether the target is out of view and start the re-detection part in time. Our method can re-detect the target with high accuracy and position the tracking area precisely in different situations. Specifically, in the first row of Fig. 12, the experiment is conducted in an indoor environment. Portrait posters and other disturbances are overcome effectively by our method, and the target is successfully re-detected from both sides of the view. As for LCT, it is disturbed by the portrait posters, and the bounding box drifts. In the second and third row of Fig.12, the scale of the target changes when it is lost and there are also disturbances from shrubs and other pedestrians. Still, our method can correctly locate to the real target and make a reasonable scale adjustment. Meanwhile, LCT fails on both scale estimation and target re-detection. Generally speaking, our target re-detection method has better performance.

C. Experiments with Mobile Robot

In order to provide comprehensive experiments, onboard experiments are carried out with a mobile robot. The Sony FCB-EX11DP Camera module is used. Its image size is 640 pixel × 480 pixel. The onboard computing system is equipped with an i5 processor with 2.66 GHz and an 8 GB RAM. Experiments are conducted with different targets in different situations, including indoor scenes, outdoor scenes.

To demonstrate the adaptability of the guider, an indoor experiment is taken as an example in Fig.13. It can be seen that there are plants, pillars, revolving doors and backlighting in the indoor environment. The mobile robot can smoothly and steadily follow the target with our guider.

Fig.13 Indoor target-following with a mobile robot

The long-term following experiments are carried out outdoors. One of the following trajectories of the mobile robot is shown in Fig.14. The image in the middle of Fig.14 is a satellite map that shows the overall experiment scene. A mobile phone is used to obtain real-time satellite positioning. The 16 images around the robot's trajectory map are the corresponding third-person perspective images in those positions. As can be seen, the robot can smoothly follow the target for about 648 meters. The average walking speed of the target is about 0.35m/s, and the total following time is about 30 min.

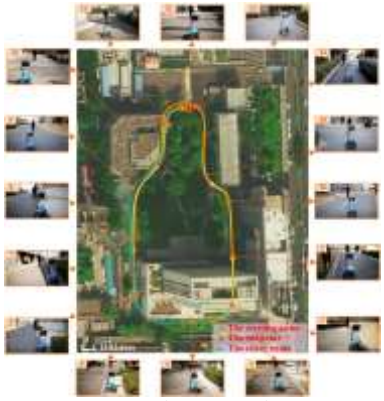


Fig.14 A trajectory of the target-following guider

VII. CONCLUSION

In this paper, a target-following guider is proposed for a mobile robot. It consists of three parts: the visual tracking part, the target re-detection part, and the visual servo part. In the visual tracking part, a TCB based sampling strategy is proposed to enhance the performance of the tracker. With the target re-detection part, the success rate of target-following is greatly improved. A generic detection method, the online learning model and some reasonable assumptions ensure the accuracy of target re-detection. In the visual servo part, the point interaction matrix of the bounding box is calculated to achieve reliable robot control. Independent tests of different parts are first conducted. Then, comprehensive experiments on the robot platform are conducted indoors and outdoors respectively. Videos from gait robot are used to demonstrate the robustness of our guider. To test the versatility of the guider, a wheeled robot is used in the onboard tests. In our experiment, the robot robustly follows the target for about 648 meters, which strongly proved the validity of the target-following guider.

REFERENCES

[1] F. Lin, X. Dong, B. M. Chen, *et al*, "A robust real-time embedded vision system on an unmanned rotorcraft for ground target following." *IEEE Trans. Industrial Electronics*, vol. 59, no. 2, pp. 1038-1049, 2012.

[2] A. Jevtić, G. Doisy, Y. Parmet, *et al*, "Comparison of interaction modalities for mobile indoor robot guidance: direct physical interaction, person following, and pointing control." *IEEE Trans. Human-Machine Systems*, vol. 45, no. 6, pp. 653-663, 2015.

[3] E. J. Jung, J. H. Lee, B. J. Yi, *et al*, "Marathoner tracking algorithms for a high speed mobile robot." *Proc. 24th IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 3595-3600, 2011.

[4] G. Doisy, A. Jevtic, E. Lucet, *et al*, "Adaptive person-following algorithm based on depth images and mapping." *Proc. 25th IEEE/RSJ Int. Conf. IROS Workshop on Robot Motion Planning*, pp. 43-48, 2012.

[5] D. Ristić-Durrant, G. Gao, and A. Leu, "Low-level sensor fusion-based human tracking for mobile robot." *Facta Universitatis, Series: Automatic Control and Robotics*, vol. 1, no. 1, pp. 17-32, 2016.

[6] E. Petrović, A. Leu, D. Ristić-Durrant, *et al*, "Stereo vision-based human tracking for robotic follower." *International Journal of Advanced Robotic Systems*, vol. 10, no. 5, pp. 230-240, 2013.

[7] Y. Isobe, G. Masuyama, and K. Umeda, "Target tracking for a mobile robot with a stereo camera considering illumination changes." *Proc. 8th IEEE/SICE Int. Sympo. System Integration*, pp.702-707, 2015.

[8] B. Ilias, S. A. Shukor, S. Yaacob, *et al*., "A Nurse Following Robot with High Speed Kinect Sensor." *ARPN Journal of Engineering and Applied Sciences*, vol. 9, no. 12, pp. 2454-2459, 2014.

[9] M. A. Mekhtiche, Z. A. Benselama, M. A. Bencherif, *et al*. "Visual tracking in unknown environments using fuzzy logic and dead reckoning." *International Journal of Advanced Robotic Systems*, vol. 13, no. 2, pp. 53-61, 2016.

[10] Y. C. Chou, M. Nakajima, "Particle filter planar target tracking with a monocular camera for mobile robots." *Intelligent Automation & Soft Computing*, vol. 23, no. 1, pp. 117-125, 2017.

[11] C. H. Hu, X. D. Ma, and X. Z. Dai. "A robust person tracking and following approach for mobile robot." *Proc. 3th IEEE Int. Conf. Mechatronics and Automation*, pp. 3571-3576, 2007.

[12] A. Ess, B. Leibe, K. Schindler, *et al*., "A mobile vision system for robust multi-person tracking." *Proc. 26th IEEE Int. Conf. Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[13] D. S. Bolme, J. R. Beveridge, B. A. Draper, *et al*, "Visual object tracking using adaptive correlation filters." *Proc. 28th IEEE Int. Conf. Computer Vision and Pattern Recognition*, pp. 2544-2550, 2010.

[14] J. F. Henriques, R. Caseiro, P. Martins, *et al*, "Exploiting the circulant structure of tracking-by-detection with kernels." *Proc. 16th IEEE Euro. Conf. Computer Vision*, pp. 702-715, 2012.

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

[16] C. Ma, X. Yang, C. Zhang, *et al*, "Long-term correlation tracking." *Proc. 33th IEEE Int. Conf. Computer Vision and Pattern Recognition*, pp. 5388-5396, 2015.

[17] Z. Hong, Z. Chen, C. Wang, *et al*. "Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking." *Proc. 33th IEEE Int. Conf. Computer Vision and Pattern Recognition*, pp. 749-758, 2015.

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

[19] M. Danelljan, G. Häger, F. Khan, *et al*., "Accurate scale estimation for robust visual tracking." *Proc. 25th British Machine Vision Conf.*, pp. 1-11, 2014.

[20] M. Tang, J. Feng, "Multi-kernel correlation filter for visual tracking." *Proc. 15th IEEE Int. Conf. Computer Vision*, pp. 3038-3046, 2015.

[21] Y. Wu, J. Lim, and M.-H. Yang. "Online object tracking: A benchmark." *Proc. 31th IEEE Int. Conf. Computer Vision and Pattern Recognition*, pp. 2411–2418, 2013.

[22] D. Xu, J. Y. Lu, P. Wang, *et al*, "Partially decoupled image-based visual servoing using different sensitive features", *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 8, pp: 2233-2243, 2017.

[23] H. David, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks", *Proc. 14th IEEE Euro. Conf. Computer Vision*, pp. 749-765, 2016.

[24] E. Gundogdu, and A.A. Alatan, "Good features to correlate for visual tracking", *IEEE Trans. Image Processing*, vol. 27, no. 5, pp. 2526-2540, 2018.

[25] F. Husain, A. Colomé, B. Dellen, G. Alenya and C. Torras, "*May. Realtime tracking and grasping of a moving object from range video*", *Proc. 15th IEEE Int. Conf. Robotics and Automation*, pp. 2617-2622, 2014.

[26] J.H. Jean and F.L. Lian, "*Robust visual servo control of a mobile robot for object tracking using shape parameters*", *IEEE Trans. control systems technology*, vol. 20, no. 6, pp.1461-1472, 2012.

[27] Stein, Procopio Silveira , *et al*. "Navigating in Populated Environments by Following a Leader." *Proc. 22th IEEE Int. Conf.* Robot and Human Interactive Communication, PP. 527-532, 2013.

[28] Irki, Zohir , *et a*l. "A fuzzy UV-disparity based approach for following a leader mobile robot." *Proc. 19th IEEE Int. Conf.* Advanced Robotics, PP. 170-175, 2015.

[29] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," *Proc. 28th IEEE Int. Conf. Computer Vision and Pattern Recognition*, pp. 2544–2550, 2010.

[30] R. C. Gonzâlez and R. E. Woods, *Digital image processing. Prentice Hall*, 2008.

Mingyi Zhang received the B.Sc. degree from China University of Geosciences (Beijing), Beijing, China, in 2014, in electrical engineering and automation. She is currently working towards Ph.D. degree at the Institute of Automation, Chinese Academy of Sciences (IACAS), Beijing, China, in control sci-ence and engineering. Her current research interests include computer vision, service robot, visual servo, and reinforcement learning.

Xilong Liu received the B.S. degree from Beijing Jiaotong University, Beijing, China, in 2009 and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2014. He is currently an Associate Professor with the Re-search Center of Precision Sensing and Control, Institute of Automa-tion, Chinese Academy of Sciences (IACAS). His current re-search interests include image processing, pattern recognition, visual measurement and visual scene cognition.

De Xu (M'05–SM'09) received the B.Sc. and M.Sc. degrees from the Shandong University of Technology, Jinan, China, in 1985 and 1990, respectively, and the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2001, all in control science and engineering. He has been with Institute of Automation, Chinese Academy of Sciences (IACAS) since 2001. He is currently a Professor with the Research Center of Precision Sensing and Control, IACAS. His current research interests include robotics and automation such as visual measurement, visual control, intelligent control, welding seam tracking, visual positioning, microscopic vision, and micro-assembly.

Zhiqiang Cao (SM'14) received the B.S. and M.S. degrees from the Shandong University of Technology, Jinan, China, in 1996 and 1999, respectively, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2002. He is currently a Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences (IACAS). His current research interests include embedded vision and visual scene cognition, multi-robot systems, and net-worked robotic system.

Junzhi Yu (SM'14) received the B.E. degree in safety engineering and the M.E. degree in precision instruments and mechanology from the North University of China, Taiyuan, China, in 1998 and 2001, respectively, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences (IACAS), Beijing, China, in 2003.

Dr. Yu serves as an Associate Editor for the IEEE Transactions on Robotics and the IEEE/ASME Transactions on Mechatronics.