

Robot visual guide with Fourier-Mellin based visual tracking

Chao PENG, Danhua CAO (✉), Yubin WU, Qun YANG

School of Optical and Electronic Information, Huazhong University of Science and Technology, Wuhan 430074, China

© Higher Education Press and Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract Robot vision guide is an important research area in industrial automation, and image-based target pose estimation is one of the most challenging problems. We focus on target pose estimation and present a solution based on the binocular stereo vision in this paper. To improve the robustness and speed of pose estimation, we propose a novel visual tracking algorithm based on Fourier-Mellin transform to extract the target region. We evaluate the proposed tracking algorithm on online tracking benchmark-50 (OTB-50) and the results show that it outperforms other lightweight trackers, especially when the target is rotated or scaled. The final experiment proves that the improved pose estimation approach can achieve a position accuracy of 1.84 mm and a speed of 7 FPS (frames per second). Besides, this approach is robust to the variances of illumination and can work well in the range of 250–700 lux.

Keywords robot visual guide, target pose estimation, stereo vision, visual tracking, Fourier-Mellin transform (FMT)

1 Introduction

In the field of industrial robots, vision technology is crucial for target pose estimation and robot end pose control. This paper aims to implement a fast and reliable pose estimation approach for robot visual guide based on binocular stereo vision system.

A stereo vision algorithm usually includes the following steps: matching the left and right images to obtain the disparity, triangulating the disparity to obtain the depth, and combining the depth for further operations. In recent years, researches that introduce stereo vision to industrial robot automation have emerged one after another. For example, Ref. [1] proposed a weld detection and position-

ing system based on stereo vision, and Ref. [2] proposed a target pose estimation system based on binocular camera for robotic grabbing applications.

Although the principle of the stereo vision is simple, the stereo matching process is usually computationally intensive, which limits the speed of the entire stereo vision system. To solve this problem, researchers introduce the concept of the region of interesting (ROI). After extracting the ROI, the algorithm can only focus on the target area in the ROI and ignores the background. In Ref. [3], the ROI is pre-defined in image's center, and the pixel intensity in the ROI is used to define the threshold to segment target and background. Reference [4] also set the ROI at the center of the image and correspondingly place the target to ignore the background. However, in practical applications, it is difficult to ensure that the target is always in the center of the image. So in Ref. [5], the researchers proposed a method to automatically extract ROI, but this method is limited to weld inspecting on metal plates. Reference [6] proposed a ROI extraction algorithm based on target detection, which can detect target objects with different poses and positions in a complex background. In Ref. [7], a target detection algorithm using the histogram of gradient (HOG) feature is applied to recognize objects and estimate pose for a service robot. The above studies show that using target detection to extract ROI is more universal. However, most of the target detection algorithms perform the detection in the full image, which is computationally intensive. We introduce object tracking into stereo vision system to perform the ROI extraction, since target tracking usually has better real-time performance than object detection.

In the past few decades, various tracking algorithms have been proposed. It is worth mentioning that since deep learning has made great progress from 2013, several deep learning-based trackers have emerged, such as MDNet [8], and GOTURN [9]. But deep learning based trackers usually depend on a large amount of training data and high-performance computing device, such as GPU, which limits its application in industrial scenes. For the purpose of introducing object tracking is to accelerate the ROI

extraction, we focus on trackers which perform well in aspects of both speed and accuracy, while a comprehensive review of the existing tracking algorithms is not in the scope of this paper. Among the existing trackers, the correlation filter based trackers can basically meet our requirements. The kernelized correlation filter (KCF) tracker proposed by Henriques et al. in Ref. [10] can achieve an amazing speed of 273 FPS, but it cannot handle target's rotation and scale changes very well. In this paper, we improve the standard KCF tracker to handle these conditions more robustly. Our proposed method is inspired by fast Fourier transform based (FFT-based) image registration techniques [11]. First, we convert an image patch containing the target into the frequency domain with Fourier transform and then transform the converted image patch from Cartesian coordinate to log-polar coordinate. These two processes are collectively referred as the Fourier-Mellin transform. After transform, the rotation and scale changes in the spatial domain can be expressed as translation in the frequency domain. We construct a correlation filter to estimate the translation in the frequency domain and convert it to the rotation and scale changes in the spatial domain. Then, the original image can be corrected according to the obtained rotation and scale. And finally, the corrected image patch that only has translation is input into the standard KCF for tracking. The improved tracker is more robust to the rotation and scale changes of the object. Experiments on the OTB-50 benchmark [12] show that our tracker is more accurate and robust than other lightweight trackers and can still reach 50 FPS.

The rest of this paper is organized as follows: In Section 2, we will give an overview of our pose estimation approach. The proposed tracker is described detailedly in Section 3. The experimental results are presented and discussed in Section 4. And finally, the conclusion is presented in Section 5.

2 System overview

An overview of the robot guide system is shown in Fig. 1.

- 1) Capture image: capture images with a binocular camera.
- 2) Preprocess and extract target: preprocess original images, such as stereo rectify and then extract the target

(the ROI).

3) Stereo match: calculate the disparity map of two ROIs by semi-global-matching (SGM).

4) Triangulation: reconstruct the target's point cloud by using the disparity map and calibration results.

5) Estimate pose: use normal distribution transformation (NDT) point cloud registration algorithm to estimate the target's pose.

2.1 Fast target extraction

In this work, object detection and object tracking are used to extract ROI. We train an AdaBoost classifier with the pyramidal aggregate channel feature [13] for a special target and apply the classifier in left image of the first frame to find the target. To find the target more quickly in the right image, we introduce the scale-invariant hypothesis and the epipolar constraint when searching the target. After the target is detected in the first frame, the object tracking algorithm can be applied to find the target in subsequent frames. Thanks to the proposed tracker that will be discussed in detail in the next section, the target extraction process can be executed in a faster speed.

2.2 Pose estimation

After the ROI is extracted, we perform a stereo matching to get the disparity map of the target. Accurate stereo matching is important for 3D reconstruction. We achieve this with the SGM [14] algorithm. The SGM needs the input images that have a known epipolar geometry. The epipolar geometry is obtained by stereo calibration, which can obtain the Euclidean transformation between the two cameras and the intrinsic parameters of each camera, and remove the effects of lens distortion. The calibration result is also used in triangulation process. We do an offline calibration [15] with MATLAB's calibration toolbox.

Once the matching has been found, the 3D point cloud of the target can be generated by a triangulation algorithm. The triangulation method is shown in Ref. [16]. Then, the NDT [17] point cloud registration algorithm is performed to get the target's 3D pose related to the reference point cloud.

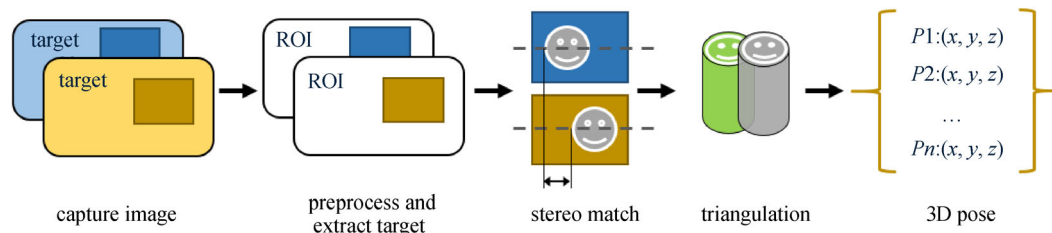


Fig. 1 Robot visual guide algorithm workflow

3 Tracker based on Fourier-Mellin transform

In this section, we first introduce Fourier-Mellin transform (FMT) and then show how the proposed KCF estimate rotation and scale. Finally, we combine the rotation-scale KCF with the standard KCF tracker to perform the whole tracking process.

3.1 Fourier-Mellin transform

Let $f_1(x,y)$ and $f_2(x,y)$ be two images differing in a 2D translation $t(x_0,y_0)$, scale factor a and rotation θ_0 . The image pairs are related as follow:

$$f_2(x,y) = f_1\left(\frac{x\cos\theta_0 + y\sin\theta_0 - x_0}{a}, \frac{-x\sin\theta_0 + y\cos\theta_0 - y_0}{a}\right). \quad (1)$$

Their Fourier transforms of them are related by

$$F_2(\xi,\eta) = \frac{e^{-j2\pi(\xi x_0 + \eta y_0)}}{a^2} \times F_1\left(\frac{\xi\cos\theta_0 + \eta\sin\theta_0}{a}, \frac{-\xi\sin\theta_0 + \eta\cos\theta_0}{a}\right). \quad (2)$$

We know that the high-frequency components can represent the edge in the image and the low-frequency components usually represent the plain regions in the image. And the energy of the low-frequency component of an image is typically a lot larger than the energy of the high-frequency component, which could cause numerical instability in the calculation. Therefore, we apply a high-pass filter in the Fourier domain to filter out low-frequency component, which is one of the main differences between our work and the work in Ref. [18]. The high-pass filter can be shown as follow:

$$\mathcal{H}(\xi,\eta) = (1.0 - \cos(\pi\xi)\cos(\pi\eta)) \times (2.0 - \cos(\pi\xi)\cos(\pi\eta)). \quad (3)$$

After filtering out the low-frequency signal, we need to calculate the Fourier magnitude spectra of the image, and the phase spectrum will be ignored because that the change in the phase spectrum represents the translation in the spatial domain, which can be estimated accurately by KCF. Letting M_1 and M_2 be the Fourier magnitude spectra of two images, and ignoring the multiplication factor $1/a^2$, we can get the following expression:

$$M_2(\xi,\eta) = M_1\left(\frac{\xi\cos\theta_0 + \eta\sin\theta_0}{a}, \frac{-\xi\sin\theta_0 + \eta\cos\theta_0}{a}\right). \quad (4)$$

The above expression can be expressed in log-polar coordinate as follow:

$$M_2(\log\rho,\theta) = M_1(\log\rho - \log a, \theta - \theta_0). \quad (5)$$

Then, we can write it as follow, where $\phi = \log\rho$, $d = \log a$,

$$M_2(\phi,\theta) = M_1(\phi - d, \theta - \theta_0). \quad (6)$$

Once the translation is got, the scale factor and rotation angle can be found.

3.2 Fourier-Mellin based kernelized correlation filter

By converting the Fourier magnitude spectrum to the Fourier-Mellin domain, scaling and rotation detection have been transformed to translation detection. Hence, we can construct a correlation filter in the Fourier-Mellin domain to estimate target scales and rotation. In Ref. [18], the researchers proposed a similar correlation filter to do this, but that correlation filter is linear. We introduced a KCF to estimate scale and rotation. We call this filter as KCF_{sr}. Compared to the linear correlation filter in Ref. [18], the proposed KCF_{sr} can represent features more effectively in higher dimension.

We train KCF_{sr} by a set of target patches x_i ($i = 1, 2, \dots, n$) and corresponding Gaussian shaped outputs g_i ($i = 1, 2, \dots, n$). Training patches x_i are target templates that have been scaled and rotated. The generated outputs g_i peak centered at the target patch in the Fourier-Mellin domain. So we make correlation filter h regresses all the training samples x_i to g_i and learns the minimum sum of squared errors expressed as

$$\operatorname{argmin} \sum_i (K(x_i, h) - g_i)^2 + \lambda \|h\|^2. \quad (7)$$

In our case, K represents the kernel correlation operation. According to our evaluation, the polynomial kernel is able to use complex model for fitting the training input. The training samples x_i are the training images' Fourier magnitude spectrum represented in log-polar

By solving Eq. (7), a close form solution of h is found as following [10]:

$$\hat{h} = \frac{\hat{g}}{\hat{k}^{xx} + \lambda}, \quad (8)$$

where symbol $\hat{\cdot}$ denotes the Fourier transform operation. Where the training patch x conduct a kernel correlation result k^{xx} , the \hat{g} represents the Fourier transform of output g , and \hat{h} denotes the Fourier transform of the correlation filter h . Fast detection can also be performed in Fourier domain according to Eq. (9). \hat{y} is the detection response of the input patch z .

$$\hat{y} = \hat{k}^{xz} \odot \hat{h}, \quad (9)$$

where \hat{k}^{xz} represents the kernel correlation of x and z . By searching the maximum value of response map y , the translation $(\Delta\phi, \Delta\rho)$ in Fourier-Mellin domain can be obtained to recover scale and rotation.

$$(\Delta\phi, \Delta\rho) = \operatorname{argmax}(y(\phi, \rho)). \quad (10)$$

In the training stage, the filter is updated by Eq. (11),

$$H_t = \eta H + (1 - \eta)H_{t-1}, \quad (11)$$

where η is the learning rate, H is equal to h , which is the Fourier transform of computed kernel correlation filter in current frame.

3.3 Improved KCF tracker

The FMT and KCF_{sr} can be collectively referred to as Fourier-Mellin correlation filter (FMCF). By employing FMT, scale and rotation of the sample can be detected. With the detected rotation and scales, samples can be corrected. And, by taking the advantage of KCF tracker, fast training and fast detection can be performed in Fourier domain with the corrected samples. We call the proposed tracker that combining the standard KCF tracker and FMCF as the scale-and-rotation robust kernelized correlation filter tracker.

The workflow of the proposed tracker is given as Fig. 2. We define the symbol P as a cropped image patch, and $P(I_t, \sigma, \phi, \ell)$ represents the patch in image I with the location ℓ , scale σ and rotation ϕ . Symbol t represents the time-stamp. Our algorithm can be described as follows.

Algorithm Our tracker: iteration at time t

Input: Current image I_t , location ℓ_{t-1} , scale factor σ_{t-1} and rotation ϕ_{t-1} in previous frame

Output: Target location ℓ_t , scale factor σ_t and rotation ϕ_t , updated filters KCF_t and $FMCF_t$

$t = 0$, initialize the first frame.

Calculate initial filters KCF_0 and $FMCF_0$ with current frame I_0

$t \geq 1$, iterates to solve σ_t, ϕ_t, ℓ_t , as Fig. 2.

Step I Estimate current scale σ_t and rotation ϕ_t from $P(I_t, \sigma_{t-1}, \phi_{t-1}, \ell_{t-1})$ using $FMCF_{t-1}$.

Step II Estimate current location ℓ_t from scaled and rotated patch $P(I_t, \sigma_t, \phi_t, \ell_{t-1})$ using KCF_{t-1} .

Step III Train correlation filters with current target, where the updated filter $FMCF_t$ and KCF_t are trained by $P(I_t, \sigma_t, \phi_t, \ell_t)$.

4 Experiment results and discussion

4.1 Experiment system

Our experiment system is shown in Fig. 3. It consists of a stereo camera, a computer, and a KUKA industrial robot. The transform between the camera coordinate and the robot coordinate is obtained through a hand-eye calibration. The robot is responsible for holding the target and placing it in different positions. Since the pose of the robot's end is known and the motion accuracy of the robot can reach 0.03 mm, which is quite high, we regard the target pose obtained from robot as the true pose, and regard the pose measured by the stereo vision system as the measured pose.

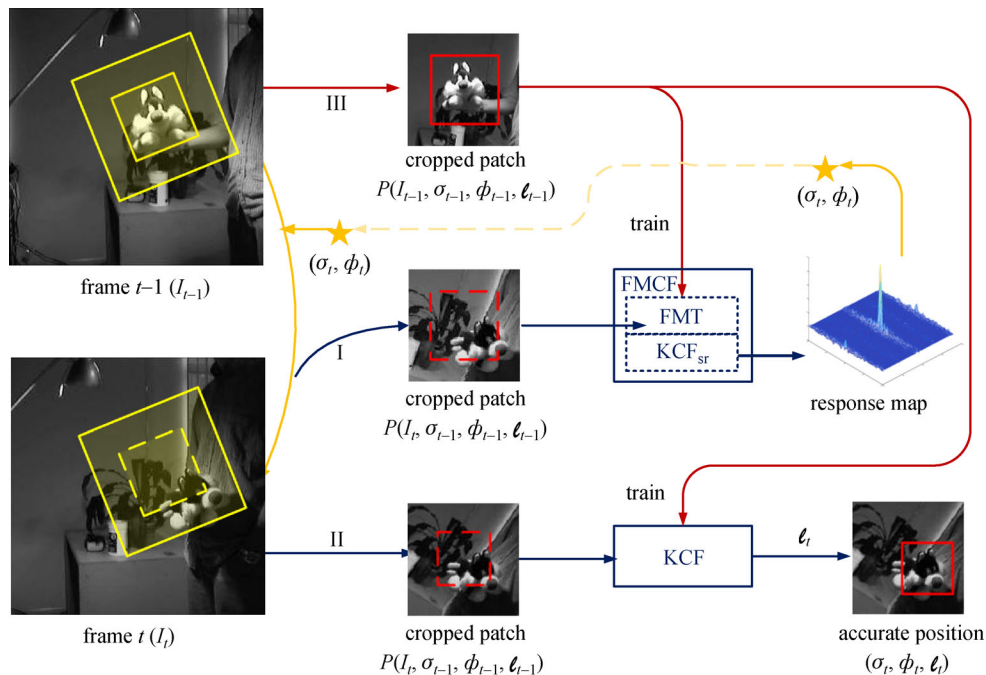


Fig. 2 Flowchart of our tracker, red line (III) represents training stage of last frame, blue line (I, II) represents detect stage of current frame

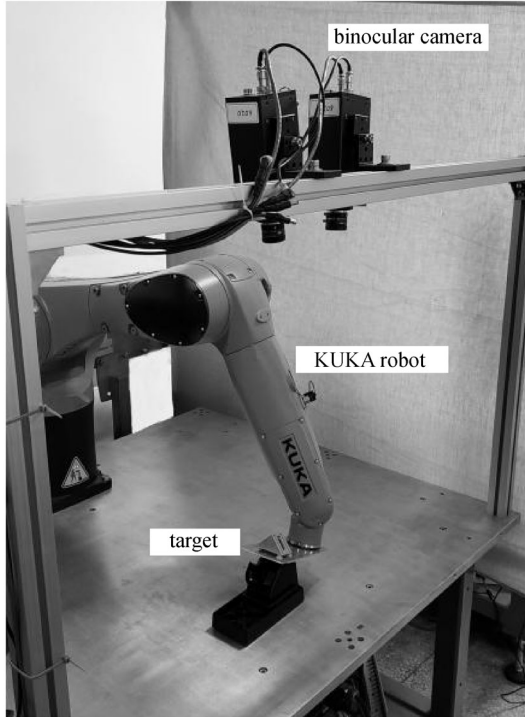


Fig. 3 Robot visual guide experiment system

4.2 Tracking results

We test our tracker on OTB-50 benchmark. In order to measure the impact of scale changes, the overlap score [12] is selected as the evaluation metric in this paper. Given a tracked bounding box r_t and the ground-truth bounding extent r_{gt} of a target object in a frame, the overlap score S in this frame is defined in Eq. (12), where \cup and \cap represent the intersection and union operators. As to a video sequence, average overlap score (AOS) and success plot

are usually used to perform evaluation [12]. We use success plot and AOS to evaluate our tracker's performance, since success plot tends to be more intuitive and AOS to be more accurate.

$$S = \frac{r_t \cap r_{gt}}{r_t \cup r_{gt}}. \quad (12)$$

We use 13 state-of-the-art existing trackers (according to Ref. [12], the top 10 tracker: Struck, SCM, ASLA, CSK, L1APG, OAB, VTD, VTS, DFT, LSK; the tracker which performs well on SV/IPR/OPR (Scale Variation/In Plane Rotation/Out Plane Rotation) subset but not contained in the top 10: CXT; the tracker which is fast but not contained in the top 10: FCT [19]; our basis tracker: KCF) as a baseline. Figure 4 shows the obtained one-pass evaluation (OPE) success plots ranking by area under curve (AUC) score. Our tracker achieves the best overall performance for all 50 sequences and the best performance for 7 of the 11 subsets. Especially, our tracker can perform well on subsets with scale variation and rotation. The AUC score of our tracker has a gain of 4.6% compared with the standard KCF tracker.

In order to evaluate these performances of our tacker in aspects of accuracy, robustness and speed more accurately, we compare it with trackers that perform excellently on SV/IPR/OPR subset according to Ref. [12] on several typical challenging sequences. The results are in Table 1. The first row and the first column in Table 1 are the compared trackers and the typical challenging sequences, respectively. We use the AOS and FPS as the evaluation criteria and the AOS data in Table 1 is in percent. It can be seen that our tracker has no obvious deficiency in all 10 sequences. Besides, our tracker can reach a quite fast speed of 50 FPS on an Intel i5 3.5 Hz CPU. Although SCM outperform our tracker, the speed of SCM is only 0.51 FPS, which is too slow to be applied to the robot vision

Table 1 Comparison of average overlap score

sequence	ours	KCF	CXT	VTD	Struck	SCM	ASLA	CSK	L1APG	attributes
Walking	0.67	0.53	0.17	0.61	0.57	0.71	<u>0.77</u>	0.54	0.75	SV
Walking2	0.62	0.40	0.37	0.33	0.51	<u>0.80</u>	0.37	0.46	0.76	SV
Car2	0.79	0.68	0.87	0.80	0.69	<u>0.92</u>	0.87	0.69	0.91	SV
Car24	0.73	0.43	0.79	0.35	0.14	<u>0.88</u>	0.80	0.38	0.78	SV
Car4	<u>0.76</u>	0.48	0.31	0.36	0.49	0.76	0.75	0.47	0.25	SV
Jogging1	0.66	0.19	<u>0.77</u>	0.15	0.17	0.18	0.18	0.18	0.17	OPR
Singer1	0.67	0.35	0.49	0.49	0.36	<u>0.86</u>	0.79	0.36	0.28	SV OPR
Human6	<u>0.59</u>	0.21	0.15	0.18	0.21	0.35	0.38	0.21	0.25	SV OPR
Tiger2	<u>0.63</u>	0.35	0.36	0.30	0.54	0.09	0.15	0.17	0.24	IPR OPR
Doll	0.74	0.53	0.75	0.65	0.55	<u>0.83</u>	0.83	0.32	0.45	SV IPR OPR
FPS	50	275	15.3	5.7	20.2	0.51	8.5	362	2	-

Notes: The underline fonts indicate the best result; the *italic-block* fonts indicate the second best ones. SV indicate scale variation, IPR indicate in-plane rotation, OPR indicate out-plane rotation

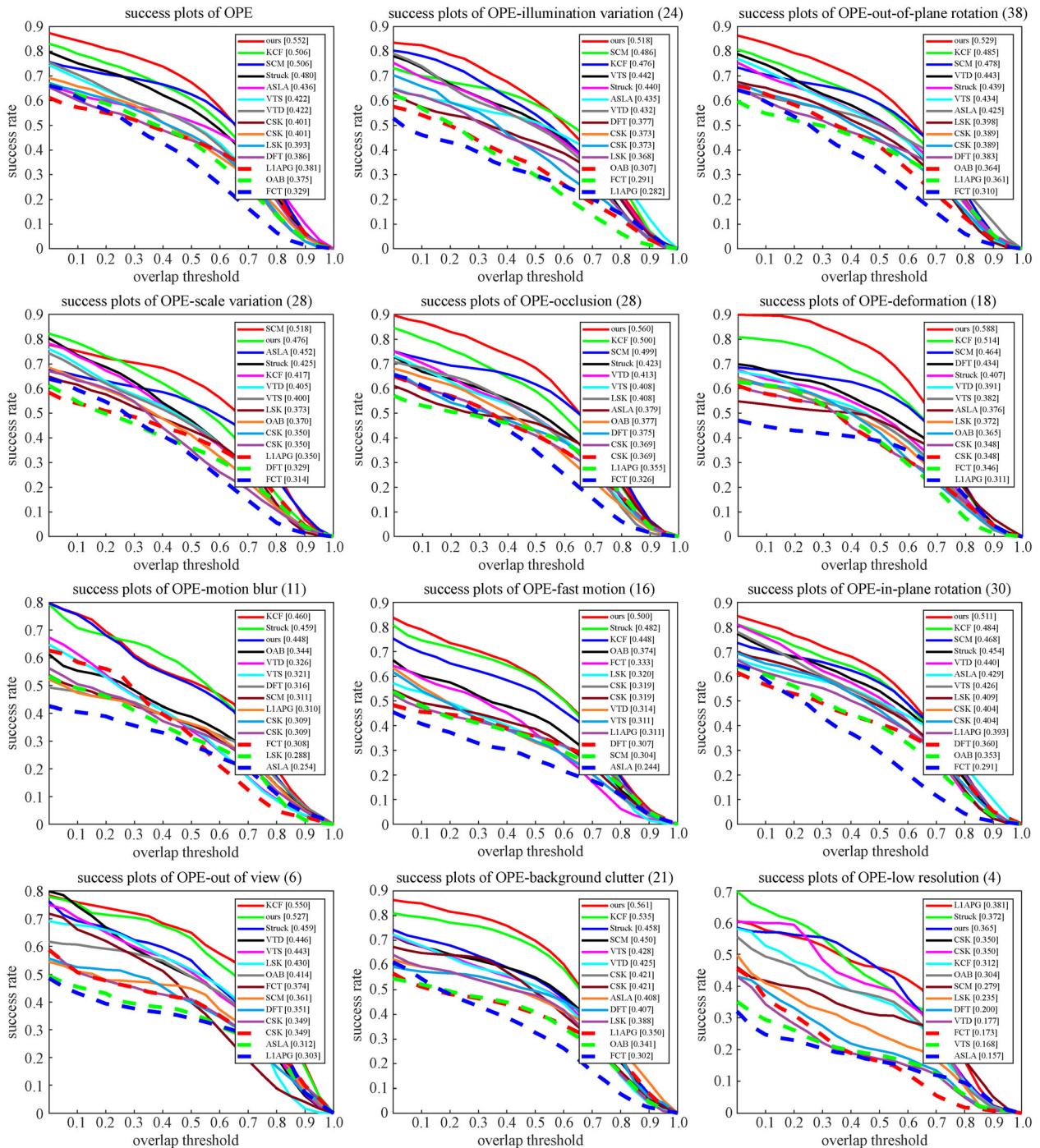


Fig. 4 Success plots on OTB-50, the AUC scores are shown in the legend

guide. Among trackers that can run in real-time, our tracker has the best performance.

For giving an intuitive impression, several snapshots of KCF, ASLA [20], CXT [21] and our tracker on two challenging sequences are shown in Fig. 5. In Walking2 sequence, the scale of target changes rapidly. And in Tiger2 sequence, the posture of target changes drastically. In these cases, our tracker is still able to track the target.

4.3 Accuracy and speed

To evaluate our system's accuracy, we carry out a total of 163 different pose estimations under illumination of 250–300 lux. We use the position error as the evaluation criteria of accuracy. In addition, the position error can be used for determining whether a pose estimation is successful. If the error is smaller than a certain threshold (e.g., to 1 mm), we say it is successful. As the threshold

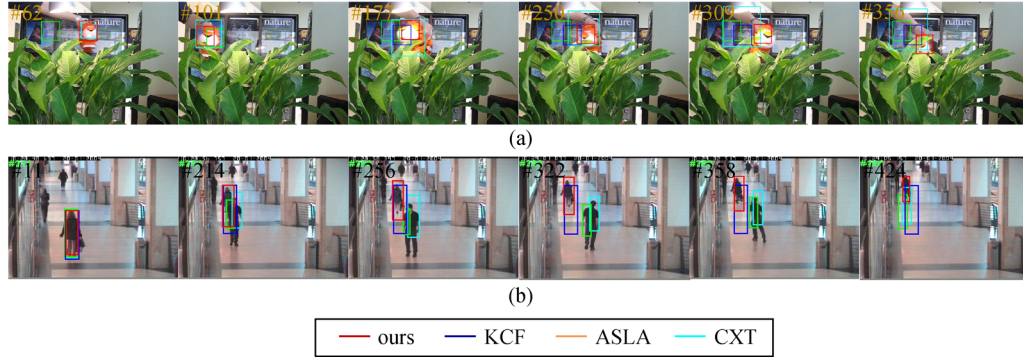


Fig. 5 Snapshot of our tracker. (a) Tiger2; (b) Walking2

varies, we can get a success rate curve in Fig. 6. And Table 2 shows that our system can achieve an average position error of 1.84 mm and a 2.5 mm success rate of 100%.

The time expense of our system is shown in Table 3. After introducing our tracking algorithm, the time consumption of ROI extraction is reduced from 61.21 to 32.32 ms. In addition, we further accelerate the algorithm

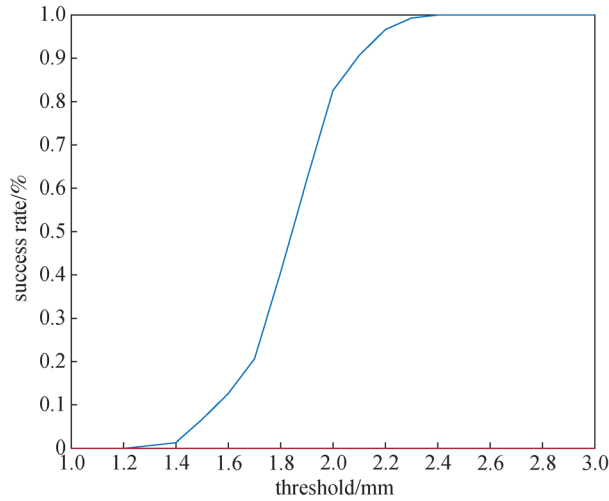


Fig. 6 Success rate plot

Table 2 Average distance and success rate in three typical threshold

2.0 mm	2.3 mm	2.5 mm	average position error
82.7%	99.3%	100%	1.84 mm

Table 4 Detail information of the illumination adaptability experiments

number	1	2	3	4	5	6	7
illumination/lux	98	125	143	186	217	238	248
5 mm success rate	10%	57%	61%	50%	62%	88%	100%
number	8	9	10	11	12	13	14
illumination/lux	276	298	340	432	534	703	823
5 mm success rate	100%	100%	100%	100%	98%	98%	54%

Table 3 Time expense

algorithm	time expense/ms	
	before apply tracking	after apply tracking
stereo rectify	7.46	7.46
ROI extract	61.21	32.32
stereo matching	14.18	14.18
pose estimation	83.88	83.88
total	166.73	137.84

by reducing the image size and the point cloud density, using the epipolar constraint to reduce the search range, etc. In the end, our algorithm can achieve a speed of 7 FPS to process 480×640 image pairs on an Intel i5 3.5 Hz CPU.

4.4 Illumination adaptability

To evaluate our system's robustness, we perform an illumination suitability test. We use position error and 5 mm success rate as the evaluation criteria. When the illumination changes from 98 to 823 lux, we measure the position error and calculate the success rate. Table 4 and Fig. 7 show the results. To depict the data distribution more fully, we plot a boxplot of the error with the maximum, minimum, outlier, median, 25%, and 75% quantile values for each set of measurements in Fig. 7. It can be seen that the system can achieve a 5 mm success rate of upper than 98% when the illumination is in a range of 250–700 lux.

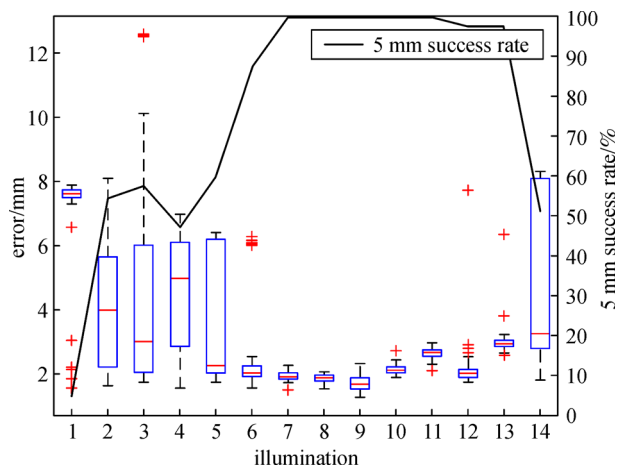


Fig. 7 Illumination adaptability experiments

5 Conclusions

We have presented a stereo vision system in this paper to solve the pose estimation problem in robot visual guide. We proposed a novel real-time target tracking algorithm which is robust to target's rotation and scale variation, and we applied this tracking algorithm to the stereo vision system to accelerate the ROI extraction. Our proposed tracking algorithm can run at a speed of 50 FPS and achieve the best accuracy among existing lightweight tracking algorithms, according to the experiment result on OTB-50 benchmark. After applying the tracking algorithm, our system's speed increases from 6 to 7 FPS. The accuracy and speed of our system can basically meet the requirement.

References

1. Agrawal A, Sun Y, Barnwell J, Raskar R. Vision-guided robot system for picking objects by casting shadows. *International Journal of Robotics Research*, 2010, 29(2–3): 155–173
2. Taryudi, Wang M S. 3D object pose estimation using stereo vision for object manipulation system. In: *Proceedings of International Conference on Applied System Innovation*. Sapporo: IEEE, 2017, 1532–1535
3. Dinham M, Fang G, Zou J J. Experiments on automatic seam detection for a MIG welding robot. In: *Proceedings of International Conference on Artificial Intelligence and Computational Intelligence*. Berlin: Springer, 2011, 390–397
4. Ryberg A, Ericsson M, Christiansson A K, Eriksson K, Nilsson J, Larsson M. Stereo vision for path correction in off-line programmed robot welding. In: *Proceedings of IEEE International Conference on Industrial Technology*. Vina del Mar: IEEE, 2010, 1700–1705
5. Dinham M, Fang G. Autonomous weld seam identification and localisation using eye-in-hand stereo vision for robotic arc welding. *Robotics and Computer-integrated Manufacturing*, 2013, 29(5): 288–301
6. Yun W H, Lee J, Lee J H, Kim J. Object recognition and pose estimation for modular manipulation system: overview and initial results. In: *Proceedings of International Conference on Ubiquitous Robots and Ambient Intelligence*. Jeju: IEEE, 2017, 198–201
7. Dong L, Yu X, Li L, Hoe J K E. HOG based multi-stage object detection and pose recognition for service robot. In: *Proceedings of 11th International Conference on Control Automation Robotics & Vision*, Singapore: IEEE, 2010, 2495–2500
8. Nam H, Han B. Learning multi-domain convolutional neural networks for visual tracking. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas: IEEE, 2016, 4293–4302
9. Held D, Thrun S, Savarese S. Learning to track at 100 FPS with deep regression networks. In: *Proceedings of European Conference on Computer Vision*. Berlin: Springer, 2016, 749–765
10. Henriques J F, Rui C, Martins P, Batista J. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, 37(3): 583–596
11. Stricker D. Tracking with reference images: a real-time and markerless tracking solution for out-door augmented reality applications. In: *Proceedings of Conference on Virtual Reality, Archeology, and Cultural Heritage*. Glyfada: DBLP, 2001, 77–82
12. Wu Y, Lim J, Yang M H. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, 37(9): 1834–1848
13. Yang B, Yan J, Lei Z, Li S. Aggregate channel features for multi-view face detection. In: *Proceedings of IEEE International Joint Conference on Biometrics*. Clearwater: IEEE, 2014, 1–8
14. Hirschmuller H. Accurate and efficient stereo processing by semi-global matching and mutual information. In: *Proceedings of IEEE Computer Society Conference on Computer Vision & Pattern Recognition*. San Diego: IEEE, 2005, 807–814
15. Zhang Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000, 22(11): 1330–1334
16. Li J, Zhao H, Jiang T, Zhou X. Development of a 3D high-precise positioning system based on a planar target and two CCD cameras. In: *Proceedings of International Conference on Intelligent Robotics and Applications*. Berlin: Springer, 2008, 475–484
17. Magnusson M, Andreasson H, Nüchter A, Lilienthal A J. Appearance-based loop detection from 3D laser data using the normal distributions transform. *Journal of Field Robotics*, 2010, 26(11–12): 892–914
18. Li Y, Liu G. Learning a scale-and-rotation correlation filter for robust visual tracking. In: *Proceedings of IEEE International Conference on Image Processing*. Phoenix: IEEE, 2016, 454–458
19. Zhang K, Zhang L, Yang M H. Fast Compressive Tracking. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2014, 36(10):2002-2015.
20. Lu H, Jia X, Yang M H. Visual tracking via adaptive structural local sparse appearance model. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Providence: IEEE, 2012, 1822–1829
21. Dinh T B, Vo N, Medioni G. Context tracker: exploring supporters

and distracters in unconstrained environments. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Colorado Springs: IEEE, 2011, 1177–1184



Chao Peng received his B.E. degree in 2015 from the School of Optical and Electronic Information, Huazhong University of Science and Technology, where he is currently pursuing his M.E. degree. His research interests include machine vision, image processing, 3D vision and machine learning.

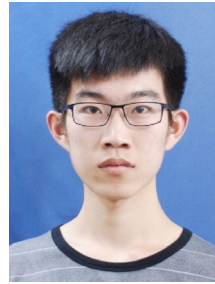


Danhua Cao is a professor in the School of Optical and Electronic Information, Huazhong University of Science and Technology. She received her Ph.D. degree in electronic physics and devices from Huazhong University of Science and Technology in 1993. She received her B.E. degree in measuring and control technology and instrumentations from Huazhong University of Science and Technology in 1987. She is a permanent member of the Professional Committee of Opto-electronic Technology in the Chinese Optical Society. Her research interests include optoelec-

tronic sensing and signal processing as well as machine vision algorithms and systems.



Yubin Wu is an associate professor in the School of Optical and Electronic Information, Huazhong University of Science and Technology. He received his M.E. degree in optical engineering from Institute of Optics and Electronics of the Chinese Academy of Sciences in 1987. He received his B.E. degree in optical instruments from Huazhong University of Science and Technology in 1984. His research interests include optoelectronic sensing and signal processing, machine vision, and development of high-tech products.



Qun Yang received his B.E. degree in 2017 from the School of Optical and Electronic Information, Huazhong University of Science and Technology, where he is currently pursuing his M.E. degree. He was interested in the field of 3D vision, machine learning and robotics.