

Tracking Registration Algorithm for Augmented Reality Based on Template Tracking

Peng-Xia Cao Wen-Xin Li Wei-Ping Ma

Lanzhou Institute of Physics, China Academy of Space Technology, Lanzhou 730000, China

Abstract: Tracking registration is a key issue in augmented reality applications, particularly where there are no artificial identifier placed manually. In this paper, an efficient markerless tracking registration algorithm which combines the detector and the tracker is presented for the augmented reality system. We capture the target images in real scenes as template images, use the random ferns classifier for target detection and solve the problem of reinitialization after tracking registration failures due to changes in ambient lighting or occlusion of targets. Once the target has been successfully detected, the pyramid Lucas-Kanade (LK) optical flow tracker is used to track the detected target in real time to solve the problem of slow speed. The least median of squares (LMedS) method is used to adaptively calculate the homography matrix, and then the three-dimensional pose is estimated and the virtual object is rendered and registered. Experimental results demonstrate that the algorithm is more accurate, faster and more robust.

Keywords: Tracking registration, augmented reality, markerless, random ferns, Lucas-Kanade (LK) optical flow.

1 Introduction

Augmented reality^[1] (AR) is a kind of technology that superimposes virtual information such as computer-generated 3D models, texts, images, and videos into real scenes. Tracking registration technology is a key technology that determines the performance of augmented reality systems. Its goal is to quickly and accurately calculate the pose information of the camera relative to the real scene, and precisely align the virtual information with the real scene based on the pose information^[2]. Based on different identification methods, the methods of tracking and registration based on augmented reality can be divided into artificial identification tracking registration methods and markerless tracking registration methods. The most representative of the artificial identification tracking registration methods are ARToolKit^[3] and ARTag^[4]. This type of methods has the advantages of small amount of calculation, fast execution speed, and no need for complicated hardware equipment. However, the artificial identification oriented registration tracking methods need to install artificial identifiers with obvious identification features in natural environments, and these artificial identifiers can be identified in the video image through matching algorithms. Due to the placement of artificial identifiers in real scenes, the artificial identification oriented registration tracking methods cannot solve

the problem of environmental illumination changes and occlusion of artificial identifiers, and had the disadvantage of poor robustness. At the same time, placing artificial identifiers in real scenes also brings a problem of visual contamination. In these cases, the markerless tracking registration methods must be used to solve the registration problem of virtual reality scenes in augmented reality^[5]. Therefore, the markerless tracking registration method is the main direction of current research and development.

The markerless tracking registration methods can estimate the camera pose directly from the relationship of the natural features between the template image and the current frame image. There are mainly two methods: a method based on template feature tracking and a method based on template image matching. On account of the method based on template feature tracking, Shi and Tomasi^[6] proposed the tracking algorithm based on Kanade-Lucas-Tomasi (KLT). The method was widely used due to its real-time advantages and was applied in the real tracking registration process of augmented reality by Li et al.^[7] and Yuan et al.^[8]. However, the KLT tracking algorithm has the disadvantage of being greatly affected by the illumination conditions, and there is a problem of tracking failure when the target transforms faster or the target is largely occluded. In addition, once a tracking failure occurs, the general target tracking algorithms will not be reinitialized. The tracking registration method based on template image matching mainly solves the problem of wide baseline matching^[9]. In the traditional wide baseline matching algorithms, the scale-invariant feature transform (SIFT) is widely used in the fields of

Research Article

Manuscript received January 3, 2019; accepted July 24, 2019; published online September 13, 2019

Recommended by Associate Editor Jangmyung Lee

© Institute of Automation, Chinese Academy of Sciences and Springer-Verlag GmbH Germany, part of Springer Nature 2019

pattern recognition and image matching because of its strong robustness^[10]. In 2004, Lowe^[11] applied the SIFT operator to the tracking registration of the AR system for the first time. However, the traditional wide baseline matching algorithms have the characteristics of computational complexity, and there are difficulties in meeting the real-time requirements of the augmented reality system. In response to this problem, Ozuysal et al.^[12] regarded wide-baseline image matching as a classification problem, and constructed a random ferns classifier based on naive Bayes, which puts the computationally intensive part into the classifier's offline training process to improve the real-time performance of the algorithm.

We can solve the problems of tracking registration failure due to changes in ambient lighting or occlusion of targets by using wide baseline matching based on random ferns classifier to detect the target of each frame and applying it to AR tracking registration. At the same time, the real-time performance of the algorithm is improved compared with the traditional wide baseline matching algorithms. However, the real-time requirements of the AR system still cannot be satisfied by only using the original random ferns algorithm. Therefore, we proposed an augmented reality tracking registration algorithm based on template tracking by combining detector and tracker. The target detection is performed using a random ferns classifier as a detector, and the detection output is used as the tracking area of the tracker to narrow the tracking range. The detector is capable of resolving the problem of not being able to reinitialize after a registration failure due to ambient lighting changes or target occlusion tracking. After detecting the target, the target is tracked by the pyramid Lucas-Kanade (LK) optical flow method to improve the real-time performance of the algorithm. This method was applied to the markless augmented reality system and had good applicability.

2 Random ferns detector

Using the random ferns classifier as a detector to detect the target mainly includes offline training and target detection. The flow chart was shown in Fig. 1. Offline training starts by extracting a certain number of keypoints of the template image. Then, the set of stable keypoints and training samples are generated. Finally, the training samples are put into random ferns of a certain scale to obtain a random ferns classifier. After getting the current frame image, corresponding patches of the keypoints are put into the random ferns classifier for rough matching. Then, random sampling consistency (RANSAC) is used to eliminate the mismatch and calculate the corresponding homography matrix.

2.1 Basic random ferns algorithm

The random ferns algorithm was originally proposed

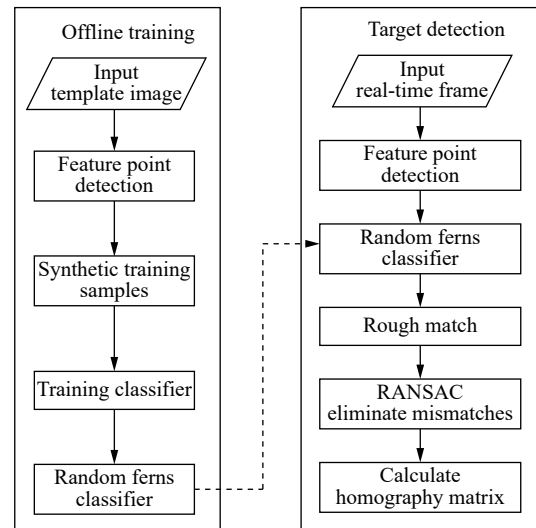


Fig. 1 Flow chart for target detection using a random ferns detector

by Lepetit et al.^[13], which is a simplified form of the random forests algorithm with better performance than the random forests algorithm. The hierarchical characteristics of the random forests are changed to non-hierarchical characteristics by selecting the same decision for each node. This change transforms the tree structure into a relatively single fern structure.

The basic idea of the random ferns classifier is similar to the feature matching based on random forests. The template image is described as H feature points. Let $K = \{k_1, \dots, k_H\}$ be the set of the feature points. Take the set of all possible appearances of the image patch $p(k_i)$ surrounding the feature point k_i as a class. Let $c_i, i = 1, \dots, H$ be the set of classes. Therefore, given the patch surrounding a feature point detected in an image, our task is to assign it to the most likely class.

Let $f_j, j = 1, \dots, N$ be the set of binary features about the patch surrounding the input image feature point, and the size of the $p(k^{input})$ is $L \times L$ (generally $L=32$). The value of each binary feature f_j only depends on the gray value $I_{d_{j1}}$ and $I_{d_{j2}}$ of two pixel locations d_{j1} and d_{j2} randomly generated by the image patch $p(k^{input})$ in the classifier training stage^[14]. We therefore write

$$f_j = \begin{cases} 1, & I_{d_{j1}} < I_{d_{j2}} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Define the category of the patch $p(k^{input})$ surrounding the input image feature point as \hat{c}_i . Therefore, we are looking for

$$\hat{c}_i = \arg \max_{c_i} P(C = c_i | f_1, f_2, \dots, f_N) \quad (2)$$

where C is a random variable that represents the class. Bayes' formula yields

$$P(C = c_i | f_1, f_2, \dots, f_N) = \frac{P(f_1, f_2, \dots, f_N | C = c_i) P(C = c_i)}{P(f_1, f_2, \dots, f_N)}. \tag{3}$$

Assuming that the prior probability $P(C)$ is an uniform distribution, since the denominator of the above formula is independent from the class, formula (2) can be converted into

$$\hat{c}_i = \arg \max_{c_i} P(f_1, f_2, \dots, f_N | C = c_i). \tag{4}$$

Due to the independence between f_j , we therefore write

$$P(f_1, f_2, \dots, f_N | C = c_i) = \prod_{j=1}^N P(f_j | C = c_i). \tag{5}$$

In order to reduce the storage of (5) and ensure the correlation between f_j , the random ferns algorithm uses a Semi-Naive Bayesian approach to patch recognition^[15]. The features f_j are divided into M groups of size $S = \frac{N}{M}$. These groups are what we define as ferns. Under the condition of Semi-Naive Bayesian classifiers, we believe that different ferns are independent of each other and that there is a correlation between nodes in the same fern. Thus, the conditional probability becomes

$$P(f_1, f_2, \dots, f_N | C = c_i) = \prod_{m=1}^M P(F_m | C = c_i) \tag{6}$$

where $F_m = [f_{\sigma(m,1)}, f_{\sigma(m,2)}, \dots, f_{\sigma(m,S)}]$, $m = 1, \dots, M$ represents the m -th fern and $\sigma(m, j)$ is a random permutation function with range $1, \dots, N$. Hence, the category $p(k^{input})$ becomes

$$\hat{c}_i = \arg \max_{c_i} \prod_{m=1}^M P(F_m | C = c_i). \tag{7}$$

In order to solve the above formula, it is only necessary to calculate the fern F_m and the conditional probability $P(F_m | C = c_i)$.

2.2 Offline training

The offline training phase estimates the class conditional probability $P(F_m | C = c_i)$ for each fern F_m and class c_i ^[16]. We assume that at least one image of the object to be detected is available for training. We call any such image as a template image. The key of offline training is to obtain stable key points and training samples. These are done by affine transformation. Using the template image as the front view, the affine images are obtained by affine transformation, and the affine images simulate the current frame images of different perspect-

ive transformations. Assuming that the point \mathbf{x} in the template image is the same as the point \mathbf{x}' in the current frame image, the relationship between them can be approximated by an affine transformation.

$$\mathbf{x}' = \mathbf{H}_A \mathbf{x} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}. \tag{8}$$

The linear matrix \mathbf{A} can be decomposed by singular value.

$$\mathbf{A} = \mathbf{U} \mathbf{D} \mathbf{V}^T = (\mathbf{U} \mathbf{V}^T) (\mathbf{V} \mathbf{D} \mathbf{V}^T) = \mathbf{R}(\theta) \mathbf{R}(\phi) \mathbf{D} \mathbf{R}(-\phi) \tag{9}$$

$$\mathbf{D} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \tag{10}$$

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \tag{11}$$

$$\mathbf{R}(-\phi) = \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix} \tag{12}$$

where $\mathbf{R}(-\phi)$, $\mathbf{R}(\phi)$ and $\mathbf{R}(\theta)$ are rotation transformation matrix, \mathbf{D} is unequal scaling matrix along the direction x and y , and $\mathbf{t} = [t_x, t_y]^T$ is translation matrix. The affine transformation is represented by $\mathbf{A}(\phi, \theta, \lambda_1, \lambda_2, t_x, t_y)$.

Offline training starts by selecting a subset of the keypoints detected on the template image. According to $\mathbf{A}(\phi, \theta, \lambda_1, \lambda_2, t_x, t_y)$ randomly selecting the affine parameters, the template image I_0 is performed affine transformation to obtain the affine image I' . The process is cyclic to obtain affine images N_{total} . The detected times of the same keypoint in all affine images is recorded as $N_{detected}$. Therefore the probability that the keypoint k is detected can be obtained as $P(k) = \frac{N_{detected}}{N_{total}}$. The keypoints that are found most often are assumed to be the most stable and retained. These stable keypoints are assigned a unique class c_i , $i = 1, \dots, H$.

Then, the training set for each class is generated based on the set of stable keypoints of the template image. After obtaining the set of stable keypoints, training set B_{train} is obtained by projecting each stable keypoint to the corresponding point in the affine view, taking the corresponding point as the center, intercepting the pixel patch as the training patch. We warped the template image using such deformations computed by randomly choosing ϕ , θ in the $[0 : 2\pi)$ range and λ_1 , λ_2 in the $[0.6 : 1.5]$ range. For each class c_i , $i = 1, \dots, H$, we used 30 random affine deformations per degree of rotation to produce 10800 training sample images.

After we got the initial class c_i and training set B_{train} , the offline training process can be performed. Randomly generating $M \times S$ random ferns that contains M ferns and each fern contains S nodes. Each node can

generate a judgment function. The judgment function selects a pair of pixel position d_{j1} and d_{j2} for each training patch in the range of $L \times L$ (take $L = 32$) pixels. For each initial class c_i , the value of the binary features f_i of the M random ferns is calculated by the formula (1) according to the gray values of the random pixel position d_{j1} and d_{j2} . The conditional probability $P(F_m|C = c_i)$ of the class c_i and each random fern F_m in (7) can be calculated based on the values f_i .

2.3 Detection matching

The detection matching starts by extracting the Lepetit keypoints from the current frame. The patch $patch^{input}$ of the keypoint is put into the random ferns classifier for classification. From the basic random ferns algorithm, it can be known that the classification of the patch $patch^{input}$ is to find the corresponding c_i according to formula(7). $p_{x,c_i} = P(F_m = x|C = c_i)$ can be obtained through offline training. Therefore, if \hat{c}_i is obtained, the patch $patch^{input}$ of the keypoint can be considered to match the template image.

After a rough match between the current frame and the template image by the random ferns classifier, the random sampling consistency (RANSAC) algorithm is used to eliminate mismatches. Usually, the larger the size of the random ferns, the higher the correct rate of matching, but the longer the running time. Figs. 2 and 3 show the trend of matching points increases with the number

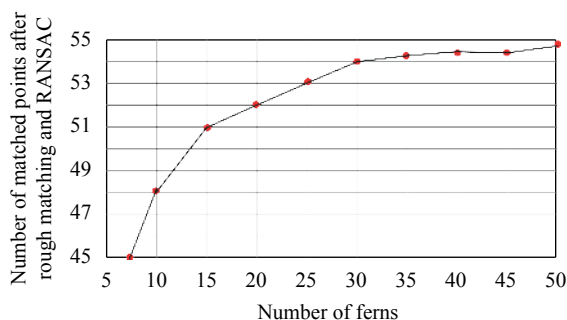


Fig. 2 Trend of matching points increases with the number of random ferns

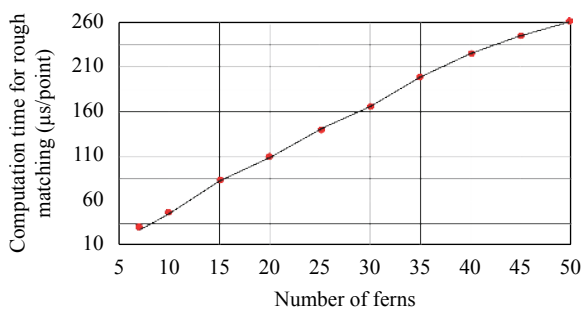


Fig. 3 Trend of single point matching time increases with the number of random ferns

of random ferns and the average matching time of each Lepetit keypoint increases with the number of random ferns. Therefore, we take the random ferns $M=30$.

3 Pyramid LK optical flow tracker

The LK optical flow algorithm is a sparse optical flow algorithm because it only needs to calculate the optical flow vector of a specific pixel. These specific pixels have certain characteristics that reflect the characteristics of the frame image well. Applying the LK optical flow algorithm to the target tracking process had the advantages of small calculation amount and fast calculation of optical flow. In this paper, the Lepetit keypoints on the target detected by the random ferns detector are used as the feature points of the target tracking. The tracking area is much smaller than the original frame image, so the tracking speed can be further improved.

The LK optical flow algorithm is based on the three assumptions^[17]: constant brightness, continuous slow motion, and spatial uniformity. The classical optical flow constraint equation can be obtained from the assumption of constant brightness and continuous slow motion. The equation can be expressed as an expression.

$$\mathbf{I}_x u + \mathbf{I}_y v + \mathbf{I}_t = 0 \quad (13)$$

where \mathbf{I}_x and \mathbf{I}_y are the partial derivative of the image, \mathbf{I}_t is the derivative of the image over time, u and v are the speeds in the x direction and the y directions respectively. Because there are two unknowns u and v for any one pixel, but only one constraining equation, there is no unique solution to (13). In order to address this aperture problem, the third hypothesis is proposed. The spatially consistent hypothesis satisfies the consistency of pixel motion in a local region. A total of 25 contradictory equations can be established by using the neighborhood 5×5 around the feature point, and the equations can be solved by the idea of the least square method. The equations can be expressed as below.

$$(\mathbf{A}^T \mathbf{A}) \mathbf{d} = \mathbf{A}^T \mathbf{b} \quad (14)$$

where \mathbf{A} is a coefficient matrix containing \mathbf{I}_x and \mathbf{I}_y , \mathbf{A}^T is the transpose of \mathbf{A} , \mathbf{d} is the velocity matrix containing u and v , and \mathbf{b} is the matrix of \mathbf{I}_t . When $\mathbf{A}^T \mathbf{A}$ is reversible, the texture of the image exists at least in two directions. Therefore, the equations have solutions.

Although the LK optical flow tracking algorithm has good real-time performance, it is not ideal in the case of large-scale motion. This problem can be solved using image pyramids^[18]. The optical flow is calculated from the highest layer of the image pyramid, and the calculation result of this layer is used as the starting value of the calculation of the next layer. The process is repeated until the bottom layer of the pyramid. This process minimizes the possibility of not meeting the assumption of continu-

ous slow motion. Thus, pyramid optical flow tracking algorithm can track targets with faster motion and larger scale changes.

4 Tracking registration algorithm for augmented reality based on template tracking

Performing template target detection on each frame of image using a random ferns detector, then calculating the three-dimensional pose of the target and performing virtual registration is feasible in practice. The part with a large amount of calculation is completed offline in the training process of the classifier. Compared with the traditional matching algorithm, this method has certain advantages, but it is still difficult to meet the real-time requirements of the augmented reality system. In addition, the method of detecting each frame does not fully utilize the similarity between successive frames. It is also feasible to apply the target tracking algorithm to the real tracking registration process, but it is easy to track failure and cannot be reinitialized. In this paper, we combine the detector and tracker and applied it to the mark-less augmented reality tracking registration process. The flow chart was shown in Fig. 4. This method mainly includes three modules: target detection, target tracking and virtual registration. The target images in the real scene are used as template images. A random ferns detector is used for target detection according to the template image. The pyramid LK optical flow tracker is used for the target tracking process to track Lepetit keypoints on the target area in real time. At the same time, we use the update optical flow tracking strategy to determine if the tracking area needs to be reinitialized, and use the least median of squares (LMedS) algorithm to adaptively

calculate the homography matrix to estimate the three-dimensional pose. OpenGL is used for the virtual registration process to perform virtual rendering according to the estimation of the pose value of each frame.

4.1 Updating the optical flow tracker

In the process of optical flow tracking, tracking error may occur due to the target moving too fast or out of the scene range, and camera shake, etc., which leads to fewer and fewer correct tracking points, and finally leads to tracking failure. For these cases, we judge whether to use the random ferns detector to reinitialize according to the number of optical flow tracking points in the target area. The specific strategy is that when the optical flow tracking point of the target area loses more than 30%, the tracking error is considered to be large, and the target detection is performed again by using the random fern detector to re-determine the tracking area. The strategy of updating the optical flow tracking is to re-adjust the tracking error in the case of a large cumulative tracking error to ensure stable tracking registration, and to reinitialize in the case of tracking failure to achieve long-term tracking.

4.2 Homography matrix estimation

The homography matrix reflects the transformation relationship between the template image and the current frame target. The estimation accuracy of the homography matrix directly affects the effect of three-dimensional registration. For this problem, although RANSAC^[19] method can obtain the homography matrix results relatively simply, there are still some shortcomings. First, the RANSAC method needs to set the dis-

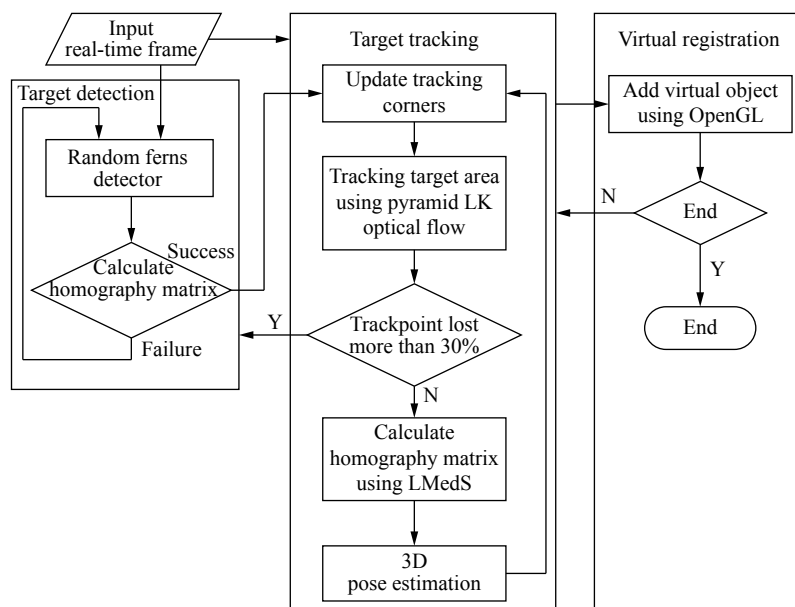


Fig. 4 Flow chart of augmented reality tracking registration algorithm based on template tracking

tance deviation threshold. If the threshold is set too small, the number of iterations will increase. Conversely, if the threshold is set too large, it will have a large impact on the estimation results. In addition, since the fewer the inner points, the higher the number of iterations, RANSAC is not suitable for the case where the interior point rate is low. However, the LMedS (least median of squares) method automatically processes according to the median value of the minimum distance deviation. This method can obtain more accurate and stable results because no thresholds need to be set^[20].

We use $D = \{(x_1, x'_1), (x_2, x'_2), \dots, (x_n, x'_n)\}$ to represent the set of all corresponding point pairs of the template image and the current frame target image, where x_1, x_2, \dots, x_n and x'_1, x'_2, \dots, x'_n are the corresponding coordinate points of the template image and the current frame target image. If d_i is used to indicate the projection distance of the i -th pair of matching point pairs, we therefore write.

$$d_i = dist(Hx_i, x'_i). \tag{15}$$

The LMedS method records the distance deviation median Med_i and the homography matrix H calculated by the i -th iteration.

$$Med_i = median\{d_1, d_2, \dots, d_n\}. \tag{16}$$

After M iterations, the smallest $MinMed$ is selected from distance deviation median, and the estimated value of the corresponding model parameter is the final homography matrix H .

4.3 Three-dimensional pose estimation

As the correct tracking for target motion, the correspondence between the current frame target and the template can be obtained by pyramid LK optical flow tracking and homography matrix estimation. Thus we have

$$(X_{1, \dots, N}, t) = H^n(X_{1, \dots, N}, t_0) \tag{17}$$

where H^n is the corresponding relationship between the current frame target point set and the template point set, that is the homography matrix. The position and orientation of the tracking target in camera coordinates can be estimated from homography matrix. The relationship between the camera transmission projection model and the target tracked by pyramid LK optical flow tracker according to the template is shown in Fig. 5. In order to simplify the projection equation, the world coordinate is defined as the tracking target's coordinate^[21].

As shown in Fig. 5, (x_p, y_p) is the true coordinate of any point on the current frame target, (x_0, y_0) is the coordinate of the corresponding template point in the projection plane of the camera, and (x_n, y_n) is the coordin-

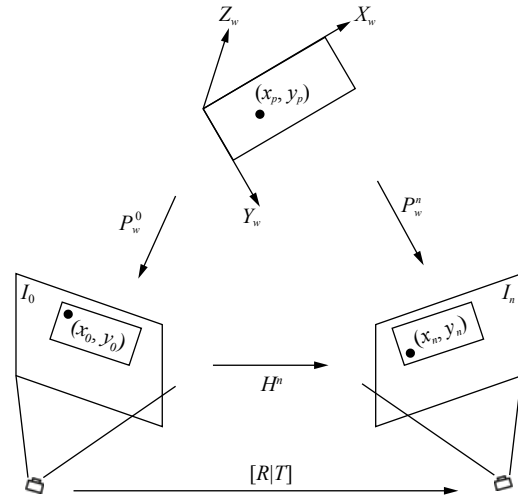


Fig. 5 Relationship between current frame target and camera in tracking process

ate of the current frame target point in the projection plane of the camera. Therefore, the relationships among them can be written as

$$\begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix} = P_W^0 \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} \tag{18}$$

$$\begin{pmatrix} x_n \\ y_n \\ 1 \end{pmatrix} = P_W^n \begin{pmatrix} x_p \\ y_p \\ 1 \end{pmatrix} \tag{19}$$

$$\begin{pmatrix} x_n \\ y_n \\ 1 \end{pmatrix} = H^n \begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix}. \tag{20}$$

The visual transmission projection equation is as follows^[22].

$$\begin{pmatrix} x_n \\ y_n \\ 1 \end{pmatrix} = \lambda K [R|T] \begin{pmatrix} x_p \\ y_p \\ z_p \\ 1 \end{pmatrix} \tag{21}$$

where λ is the scale factor, K is the intrinsic parameters of the camera, $[R|T]$ is the extrinsic parameters of the camera. $R = [r_1 \ r_2 \ r_3]$ is the rotation matrix and T is translation matrix. Because the world coordinate has been defined with respect to the tracking target, $[R|T]$ is the three-dimensional pose that we need to solve for tracking and registration. Let the target plane be the Z plane in the world coordinate and $z_p = 0$. Substituting the three equations (18), (19) and (20) into the visual transmission projection (21), we have

$$H^n = P_W^n (P_W^0)^{-1} = \lambda K [r_1 \ r_2 | T] (P_W^0)^{-1}. \tag{22}$$

It is known from the characteristics of the rotation matrix that $|r_1| = |r_2| = |r_3|$, and r_1, r_2, r_3 are perpendicular to each other. So r_3 can be obtained after finding r_1, r_2 . The projection matrix P_W^0 can be calculated by (18) (the number of calculating points ≥ 4). Thus, the three-dimensional pose $[R|T]$ of the target can be got by computing (22).

5 Experimental results

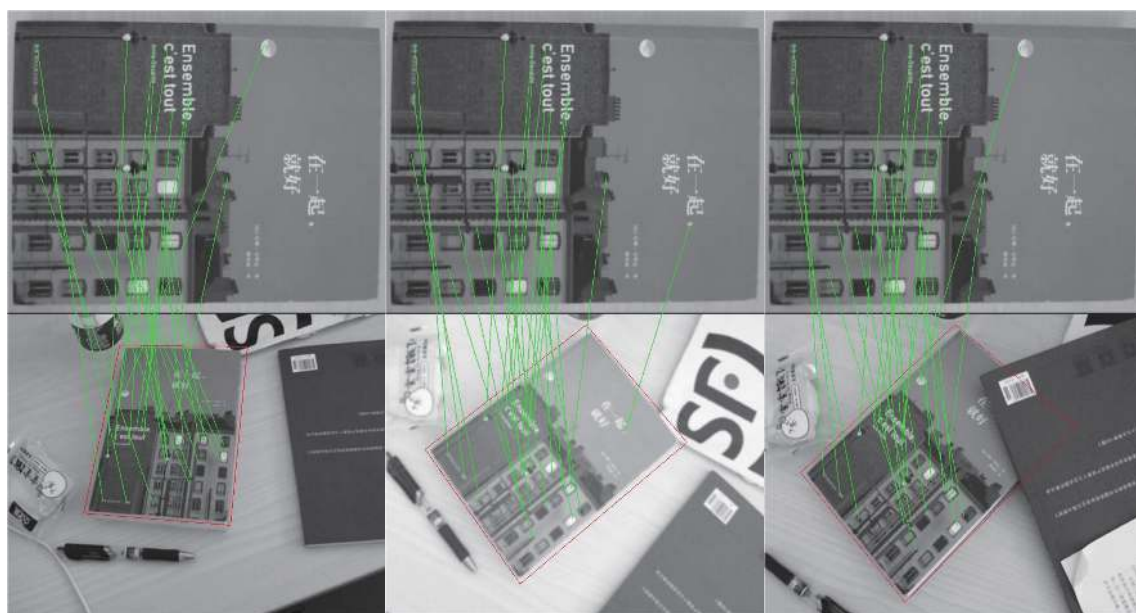
The experiments on the algorithms in this paper were implemented in the environment of VS2012 and OpenCV2.4.9. PC is Pentium(R) Dual-core T4400@2.20 GHz, ROM 2GB. The normal camera Logitech C525 was used to capture images and videos. The resolution of the template image is 640×480 .

In the experiment, the wide baseline matching target detection was firstly performed using the random ferns algorithm based on naive Bayesian classification. According to Fig. 6, the target can be accurately detected under different angles and different illumination environments. In particular, the target can still be accurately detected when the target was partially occluded.

After the target was successfully detected, the pyramid LK optical flow tracker was used to track the Lepetit keypoints on the target area in real time. Meanwhile, the homography matrix was adaptively calculated using LMedS. As shown in Fig. 7, we had tested the tracking effects of the pyramid LK and the algorithm of this paper in different environments such as different scales, changes in lighting conditions, and partial occlusion of the target. When the target moves too fast, the target is occluded, and the camera shakes, it will result in fewer and fewer correct tracking points, and may even cause tracking fail-

ure. For example, the third graph in Fig. 7(a) has shown a cumulative tracking error and the fourth graph in Fig. 7(a) has a tracking failure phenomenon. However, the random ferns detector can be used in this paper to re-adjust the tracking error in the case of a large cumulative tracking error to ensure stable tracking registration, and to reinitialize in the case of tracking failure to achieve long-term tracking. The fourth picture in Fig. 7(b) was the result of reinitialization using a random ferns detector after tracking failure.

After obtaining the target of each frame image using the random ferns detector and the pyramid LK optical flow method, the three-dimensional pose estimation was performed. The three-dimensional registration can be realized by using OpenGL drawing. Fig. 8 shows the experimental effects of using the algorithm of this paper to perform virtual and real registration for targets under different conditions. Therefore, the tracking registration algorithm for augmented reality based on template tracking can solve the problem of tracking registration failure under the condition that the lighting condition changes and the target is partially occluded. At the same time, for the same video stream, the time spent in the detection process and the tracking process of the algorithm are tested. The average single frame time of the detection process is about 47ms, and the average single frame time of the tracking process is about 10ms. In addition, the augmented reality tracking registration process using the algorithm of this paper is mostly in the tracking state, and only in the initial state and a small amount of reinitialization is required to be in the detection state. Therefore, the real-time performance of the algorithm has been greatly improved, and it can meet the real-time require-



(a) Viewing angle change detection (b) Illumination change detection (c) Partial occlusion detection

Fig. 6 Detection results based on random ferns broad baseline matching

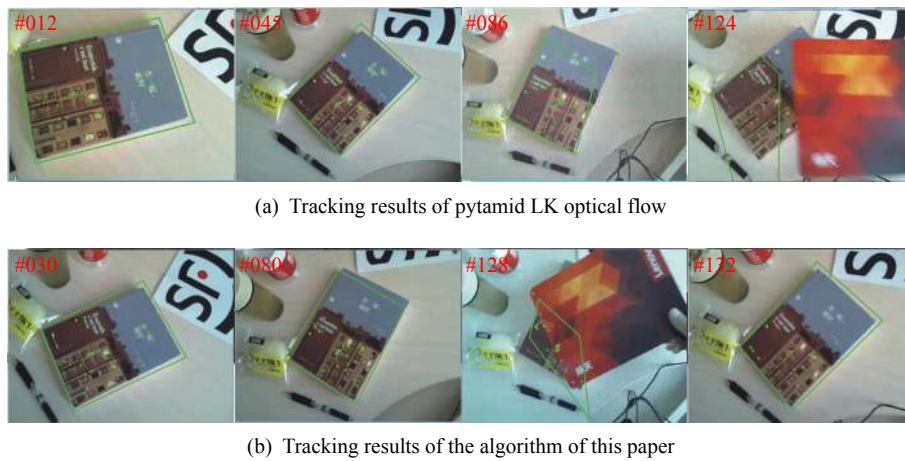


Fig. 7 Tracking results of pyramid LK optical flow and the algorithm of this paper

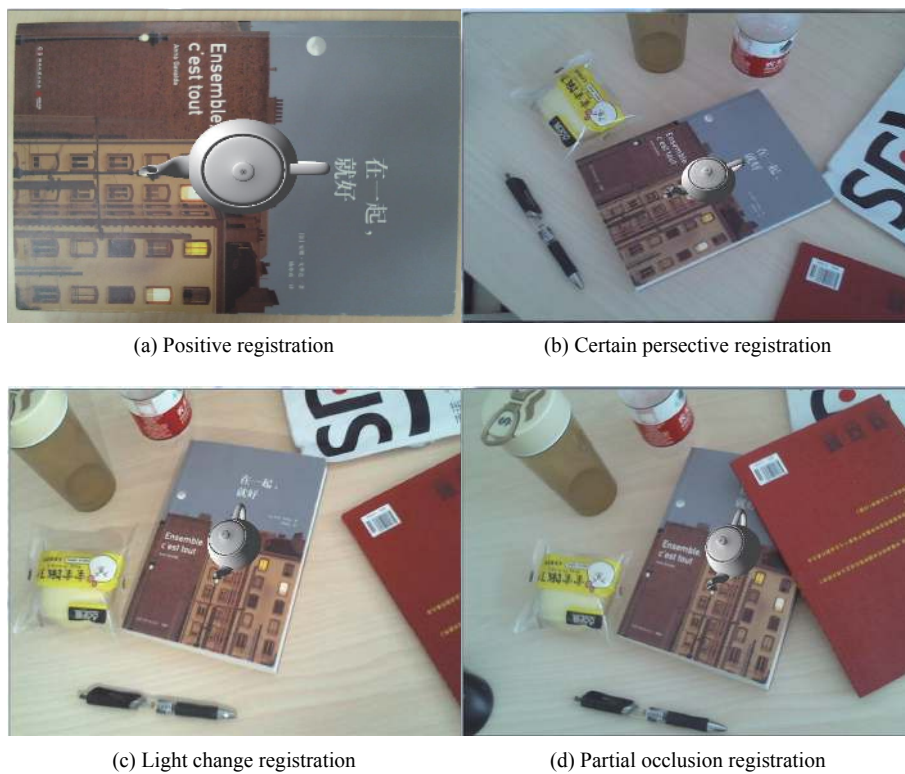


Fig. 8 Tracking registration results of markless augmented reality based on template tracking

ments of the augmented reality system.

6 Conclusions

In this paper, we had proposed an augmented reality tracking registration algorithm based on template tracking. This method combined random ferns detector with the pyramid LK optical flow tracking algorithm for augmented reality tracking registration. According to the template image, the target was detected by the random ferns classifier. The detection result was used as the input of the pyramid LK optical flow tracking, and the keypoints of the target area were tracked in real time to im-

prove the real-time performance of the algorithm. The random ferns detector was used to reinitialize the target area when the optical flow tracking points of the target area lost more than 30%. By repeating this process, long-term tracking registration can be achieved. The method of this paper can meet the requirements of real-time performance of the augmented reality system while ensuring the accuracy of registration.

Acknowledgements

This work was supported by National Natural Science Foundation of China (No. 61125101).

References

- [1] J. P. Lima, R. Roberto, F. Simões, M. Almeida, L. Figueiredo, J. M. Teixeira, V. Teichrieb. Markerless tracking system for augmented reality in the automotive industry. *Expert Systems with Applications*, vol. 82, pp. 100–114, 2017. DOI: [10.1016/j.eswa.2017.03.060](https://doi.org/10.1016/j.eswa.2017.03.060).
- [2] R. Frikha, R. Ejbali, M. Zaid. Camera pose estimation for augmented reality in a small indoor dynamic scene. *Journal of Electronic Imaging*, vol. 26, no. 5, Article number 053029, 2017. DOI: [10.1117/1.JEI.26.5.053029](https://doi.org/10.1117/1.JEI.26.5.053029).
- [3] F. P. Vista IV, D. J. Lee, K. T. Chong. Remote activation and confidence factor setting of ARToolkit with data association for tracking multiple markers. *International Journal of Control and Automation*, vol. 6, no. 6, pp. 243–252, 2013. DOI: [10.14257/ijca.2013.6.6.23](https://doi.org/10.14257/ijca.2013.6.6.23).
- [4] M. Fiala. ARTag, An Improved Marker System Based on ARToolkit[J], NRC 47166/ERB-1111, National Research Council Canada, 2004. DOI: [10.4224/5763247](https://doi.org/10.4224/5763247).
- [5] Y. Y. Du, Z. J. Miao, Y. G. Cen. Markless augmented reality registration algorithm based on ORB. In *Proceedings of the 12th International Conference on Signal Processing*, IEEE, Hangzhou, China, 1236–1240, 2015. DOI: [10.1109/ICOSP.2014.7015197](https://doi.org/10.1109/ICOSP.2014.7015197).
- [6] J. B. Shi, C. Tomasi. Good features to track. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, 593–600, 1994. DOI: [10.1109/CVPR.1994.323794](https://doi.org/10.1109/CVPR.1994.323794).
- [7] J. Li, R. Laganier, G. Roth. Online estimation of trifocal tensors for augmenting live video. In *Proceedings of the 3rd IEEE and ACM International Symposium on Mixed and Augmented Reality*, IEEE, Arlington, USA, 182–190, 2004. DOI: [10.1109/ISMAR.2004.42](https://doi.org/10.1109/ISMAR.2004.42).
- [8] M. L. Yuan, S. K. Ong, A. Y. C. Nee. Registration using natural features for augmented reality systems. *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 569–580, 2006. DOI: [10.1109/TVCG.2006.79](https://doi.org/10.1109/TVCG.2006.79).
- [9] B. Kang, P. Ren. Natural texture-based tracking algorithm for augmented reality. *Systems Engineering and Electronics*, vol. 31, no. 10, pp. 2480–2484, 2009. DOI: [10.3321/j.issn:1001-506X.2009.10.0440](https://doi.org/10.3321/j.issn:1001-506X.2009.10.0440). (in Chinese)
- [10] Y. Hao, Z. J. Xu, Y. Liu, J. Wang, J. L. Fan. Effective Crowd Anomaly Detection Through Spatio-temporal Texture Analysis. *International Journal of Automation and Computing*, vol. 16, no. 1, pp. 27–39, 2019. DOI: [10.1007/s11633-018-1141-z](https://doi.org/10.1007/s11633-018-1141-z).
- [11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. DOI: [10.1023/B:VISI.0000029664.99615.94](https://doi.org/10.1023/B:VISI.0000029664.99615.94).
- [12] M. Ozuysal, M. Calonder, V. Lepetit, P. Fua. Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 3, pp. 448–461, 2010. DOI: [10.1109/TPAMI.2009.23](https://doi.org/10.1109/TPAMI.2009.23).
- [13] V. Lepetit, J. Pilet, P. Fua. Point matching as a classification problem for fast and robust object pose estimation. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Washington DC, USA, pp. 244–250, 2004. DOI: [10.1109/CVPR.2004.1315170](https://doi.org/10.1109/CVPR.2004.1315170).
- [14] Y. Zhao, J. J. Li, H. P. Li, D. Yang. Real-time tracking and registration algorithm of scenarios of augmented reality based on improved random fern. *Journal of Northeastern University (Natural Science)*, vol. 37, no. 5, pp. 614–618, 2016. DOI: [10.3969/j.issn.1005-3026.2016.05.002](https://doi.org/10.3969/j.issn.1005-3026.2016.05.002). (in Chinese)
- [15] C. H. Yang, S. D. Liu, Z. M. Wang, Y. F. Guo, H. Li. Real-time vehicle matching based on random ferns. *Journal of Xiamen University (Natural Science)*, vol. 53, no. 2, pp. 206–211, 2014. DOI: [10.6043/j.issn.0438-0479.2014.02.011](https://doi.org/10.6043/j.issn.0438-0479.2014.02.011). (in Chinese)
- [16] M. Villamizar, J. Andrade-Cetto, A. Sanfeliu, F. Moreno-Noguer. Bootstrapping boosted random ferns for discriminative and efficient object classification. *Pattern Recognition*, vol. 45, no. 9, pp. 3141–3153, 2012. DOI: [10.1016/j.patcog.2012.03.025](https://doi.org/10.1016/j.patcog.2012.03.025).
- [17] Y. Liu, D. G. Xi, Z. L. Li, Y. Hong. A new methodology for pixel-quantitative precipitation nowcasting using a pyramid Lucas Kanade optical flow approach. *Journal of Hydrology*, vol. 529, pp. 354–364, 2015. DOI: [10.1016/j.jhydrol.2015.07.042](https://doi.org/10.1016/j.jhydrol.2015.07.042).
- [18] T. D. Chen, J. Hu, D. Wu. Dynamic target detection and tracking based on fast computation using sparse optical flow. *Journal of Image and Graphics*, vol. 18, no. 12, pp. 1593–1600, 2013. DOI: [10.11834/jig.20131207](https://doi.org/10.11834/jig.20131207). (in Chinese)
- [19] Y. Xie, X. D. Yang, Z. Liu, S. N. Ren, K. Chen. Method for visual localization of oil and gas wellhead based on distance function of projected features. *International Journal of Automation and Computing*, vol. 14, no. 2, pp. 147–158, 2017. DOI: [10.1007/s11633-017-1063-1](https://doi.org/10.1007/s11633-017-1063-1).
- [20] J. S. Xu, Y. J. Wang, X. Cheng, S. Li, S. Y. Chen. Adaptive method for homography matrix estimation. *Computer Engineering and Applications*, vol. 52, no. 5, pp. 160–164, 2016. DOI: [10.3778/j.issn.1002-8331.1409-0357](https://doi.org/10.3778/j.issn.1002-8331.1409-0357). (in Chinese)
- [21] J. Lin, Y. T. Wang, Y. Liu, K. Yang. Real-time markerless registration algorithm for augmented reality based on template tracking. *Journal of Image and Graphics*, vol. 13, no. 9, pp. 1812–1819, 2008. DOI: [10.11834/jig.20080929](https://doi.org/10.11834/jig.20080929). (in Chinese)
- [22] A. T. Erdem, A. O. Ercan. Fusing inertial sensor data in an extended Kalman filter for 3D camera tracking. *IEEE Transactions on Image Processing*, vol. 24, no. 2, pp. 538–548, 2015. DOI: [10.1109/TIP.2014.2380176](https://doi.org/10.1109/TIP.2014.2380176).



Peng-Xia Cao received the B.Eng. degree in communication engineering from Hunan International Economics University, China in 2011, and the M.Eng. degree in circuits and systems from Hunan Normal University, China in 2015. Currently, she is a Ph.D. degree candidate in space electronics at Lanzhou Institute of Physics, China Academy of Space Techno-

logy (CAST), China.

Her research interests include space electronic technology, computer vision, and augmented reality.

E-mail: 316657294@qq.com (Corresponding author)

ORCID iD: 0000-0002-3020-1650



Wen-Xin Li received the M.Eng. degree in applied mathematics from Northwestern Polytechnical University, China in 1993, and the Ph.D. degree in automatic control from Northwestern Polytechnical University, China in 2011. Currently, he is a researcher at Lanzhou Institute of Physics, CAST, China.

His research interests include space electronic technology, software reuse technology, system simulation and reconstruction technology.

E-mail: lwxcast@21cn.com



Wei-Ping Ma received the B.Eng. and M.Eng. degrees in electronic information science and technology from Xi'an University of Science and technology, China in 2011 and 2015, respectively. Currently, she is a Ph.D. degree candidate in space electronics at Lanzhou Institute of Physics, CAST, China.

Her research interests include space electronic technology, computer vision and intelligent robotics.

E-mail: 498938802@qq.com