Combining Computer Graphics and Feature for 3D Camera Tracking Based on CAD Model

Linlin Wang, Guoyun Lv(✉), Ningxin Zhang, and Yanggege Yu

School of Electronics and Information, Northwestern Polytechnical University, Xi'an, China
534183837@qq.com, 178510887@qq.com, 2568117320@qq.com, 376151248@qq.com

Abstract. A real-time 3D tracking system based on CAD model is proposed in this paper. The strategy in the tracking process includes both template-based and keypoint-based approaches. Compared with traditional CAD model-based tracking, a more accurate initial camera pose can be provided for the following feature-based operation which can accelerate the convergence of camera pose estimation. Using template-based method reduces the demand for textures of the tracking object to improve the universality of the system. Furthermore, adaptive visual feature extraction within the feature-based tracking is adopted in the experiment, and feature homogenization and other methods is also chosen to enhance the robustness and interference immunity of the system. Finally, the effectiveness and stability of the methods proposed in this paper is verified through the experiment of tracking targets in the image sequence.

Keywords: Template-based tracking · Feature-based tracking
CAD model

1 Introduction

The recognition and tracking of objects have always been the hot issue in machine vision. With the rapid development of related research in recent years, many related studies have already been applied in various domains such as robotics, AR (Augmented Reality), industry applications and so on. The 2D plane tracking is to get the object location in the image coordinate system among continuous image sequence. While in 3D tracking, the goal is to obtain the position and orientation of object relative to the camera, which is also called camera pose. Nowadays, there are a variety of related methods to solve the problem of real-time 3D object tracking and camera pose estimation. The CAD model are also widely used as a priori knowledge in these methods. These methods can be divided into the following categories: *edge-based*, *keypoint-based*, *template-based* and *hybrid-based*. For most objects, feature points and edges can easily be extracted, so the 3D visual tracking based on feature is widely used.

In the methods of *edge-based* tracking, the estimation of the pose is achieved by aligning projective edges of 3D CAD line model and edges extracted from current frame. Harris proposed a matching algorithm which matches the projective edges of CAD with the corresponding edges in the image [1]. Drummond eliminate the

B. Li et al. (Eds.): ChinaCom 2017, LNICST 237, pp. 23–32, 2018.
https://doi.org/10.1007/978-3-319-78139-6_3

influence of outliers by using a robust estimator [2]. Goued proposed a real-time tracking system based on the method of Wuest [3] and get a more stable tracking result by using multi-constraint conditions [4]. In addition, a relatively stable initial camera pose should be ensured when introducing edge-based tracking. And if the pose has large deviation, it will be difficult to get the correct value in subsequent optimal estimation process.

In *keypoint-based* methods, pose estimation is performed by matching the corresponding feature points of the reference image and the current image. The corner detection algorithm delivered by Harris is highly efficient [5]. Lowe proposed an AR system built by using the scale-invariable feature point named SIFT (Scale-invariant Feature Transform) [6]. Bay proposed an optimization algorithm for SIFT, which could reduce the computation cost [6]. In addition, Rabaud thought about another new way on the 3D visual tracking, based on scale-invariable feature ORB [7]. The 3D CAD model of the object needs to be pre-built, and it is easy to estimate the camera pose when the correspondences between 3D points and the 2D image features are known.

Hybrid-based approach is a combination of methods that mentioned above. It is proposed to solve the instability caused by using a single method in practical applications. Vacchetti raised methods which fuse feature points and edges under multiple hypothesis [8]. Similarly, Marchand presented method that combine the texture information and edge tracking, and integrated M-estimator in the minimization process to improve system stability [9]. Christensen came up with an approach integrating Global Pose Estimation (GPE) with Local Pose Estimation (LPE) for tracking [10].

But each approach has its strengths and weakness. When the target has obvious edge features and poor textures, edge-based tracking method is very efficient and stable even under the influence of illumination and specular material. Since the edge features are not invariant, the approach cannot get superior results when the textures of the object are rich or the environment is complicated. Feature points have strong characteristic, which makes them invariant to both rotation and affine transformation. However, as illumination can change the gray-level of the image, the keypoint-based tracking is greatly affected. The blurred image caused by fast movement of camera influences the extraction and matching of feature points. Moreover, keypoint-based tracking is not always beneficial, because the amount of computation is expensive and it cannot achieve real-time effects in the application.

This paper has mentioned to achieve 3D object tracking and camera pose estimate-on by the way of combining template and feature points. A more accurate initial pose is obtained by using template-based initialization method. After initialization, keypoint-based method is employed to continue the tracking and estimation. The initial value of the object pose is estimated by matching the template image rendered from CG (Computer Graphics) with image sequence which only uses the geometric information of the object. Therefore, this method can acquire the accurate initial pose for object with rich or poor texture. The video frame which has been matched successfully with the CAD model rendered by CG is set as a reference frame. And then we can get the correspondences between 3D coordinates and 2D feature points in the reference frame based on the initial pose. In the subsequent process, as long as obtaining the transformation between the reference frame and the current video frame, the camera position can be obtained by updating the corresponding relationship between 2D feature points and 3D coordinates.

In the remainder of this paper, the theory basis is introduced in Sect. 2. The system implementation process is introduced in Sect. 3. The Sect. 4 describes the experiments and evaluation of the system. Section 5 summarizes the paper.

2 Theoretical Background

The essential of the pin-hole camera model used in this paper is a perspective project-ion model. The intrinsic matrix presents the relationship between the image coordinate and the camera coordinate. The coordination of any object can be transformed from a world coordinate to a camera coordinate through rigid body transformation, which is called camera extrinsic matrix ***M***. A series of 3D points $p_i = (x_i, y_i, z_i)^t$ that are non-coplanar in the world coordinate can be transformed to the points $p_i^{'} = (x_i^{'}, y_i^{'}, z_i^{'})^t$ in the camera coordinate by the formula (1).

$$p_i^{'} = rp_i + t \tag{1}$$

$r = (r_x, r_y, r_z)^t$ is a rotation vector, while $t = (t_x, t_y, t_z)^t$ is a translation vector. The rotation vector can be converted to the corresponding rotation matrix ***R*** by the Rodriguez transformation. The camera's extrinsic matrix is constituted of the rotation matrix R and the translation vector t.

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{R} & \mathrm{t} \\ 0^T & 1 \end{bmatrix} \tag{2}$$

These parameters are closely related to the problem of camera pose estimation. According to the theoretical relationship mentioned above, the relationship between the 2D point g_i of the image coordinate and its corresponding 3D point p_i in the pinch model is:

$$g_i = \boldsymbol{KM}p_i \tag{3}$$

The critical issue of pose estimation is to determine 6–DOF (Degree of Freedom) parameters, which means to find the correspondence between 3D and 2D points observed in image plane. The current pose of object, called camera pose, can be calculated by the pre-computed camera pose and the transformation between the reference frame and the current frame. Usually the camera pose is obtained by minimizing accumulation of errors, such as the least-squares method and the Gauss-Newton method (Fig. 1).

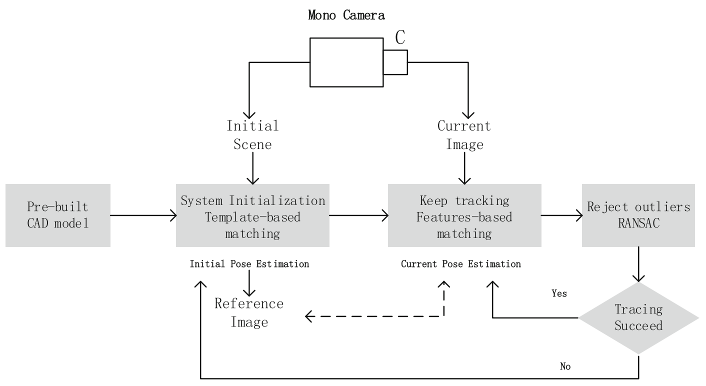

Fig. 1. The framework of tracking system. The initial pose of object is estimated by template-based matching. On the basis of the initial pose, the pose of the object is consecutively estimated by SURF key-points matching.

3 Proposed Approach

3.1 System Initialization Using Optimized Template-Based Matching

Tracking through the visual features of an object,the correctness of subsequent tracking is based on the accurate initial pose. The method of common initialization includes not only the manual calibration of corresponding 2D and 3D points, but also feature-based matching. These methods are suitable for objects with rich texture or for simple background situation, otherwise, it might influence the subsequent tracking because of inaccurate initial pose caused by mismatching. We use CAD model rendered by Computer Graphics matched with the current image. There is only geometric information in the current frame after processed, which does not contain the texture information and background of the object. Using OpenGL to render the CAD model of the object, as the camera intrinsic parameters $\boldsymbol{K}$ and extrinsic parameters $\boldsymbol{M}$ are known, the template image can be rendered. The initial pose is estimated by matching the current image with the template image.

In this paper, a CAD model is pre-built and a camera has been calibrated in advance, the intrinsic matrix $\boldsymbol{K}$ has been fixed and the value in extrinsic matrix $\boldsymbol{M}$ can be set while rendering. We can get the rendering image T using OpenGL as the first row in Fig. 2.

Simultaneously, we can capturing a series of images I' using camera with a series of unknown extrinsic parameter matrixes $\boldsymbol{M}'$ In order to obtain the extrinsic parameter matrix $\boldsymbol{M}'$ of the image currently captured by the camera, we need to minimizing the difference between image $T(\boldsymbol{M})$ and the image of the current frame $I(\boldsymbol{M}')$:

$$\Delta = T(\boldsymbol{M}) - I'(\boldsymbol{M}') \tag{4}$$

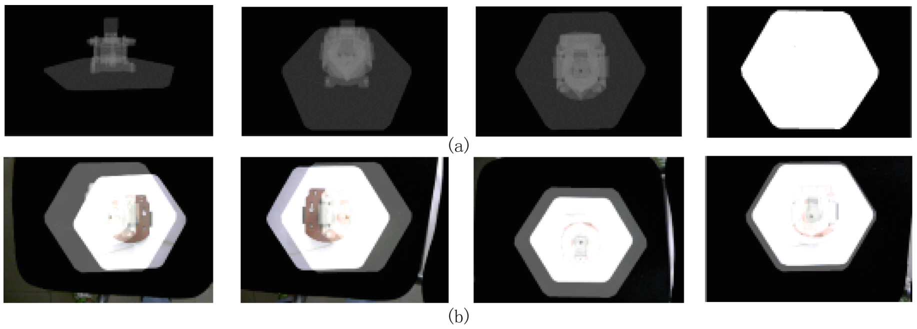

Fig. 2. Template-based matching using CG images. The images rendered by different extrinsic parameter shown in the first row (a). The last image in first row show the preprocessed image which eliminate the texture information from input image. The second row (b) shows the image difference between the template and the current image from a real camera.

The similarity of the matching is usually described by using the squared difference of the gray value of the video frame sub-region and the template image.

$$\boldsymbol{D}(i,j) = \sum_{m=1}^{M}\sum_{n=1}^{N}[I_{ij}(m,n) - T_{ij}(m,n)]^2((i,j) \in l(i,j)) \tag{5}$$

In the above equation, the $\boldsymbol{D}(i,j)$ is similar matrix, and $l(i,j)$ is the upper left corner coordinate of the region with the most similarity. The best matching pose found in $\boldsymbol{D}$ is the point $\varepsilon_1 = \min\{d_{ij}\}$ with smallest value. To ensure the accuracy of the matching, it is found in the experiment that the correctness of choosing points fallen in $\varepsilon_1 \sim 2\varepsilon_1$ matching is up to 98%. Therefore, the modified similar matrix $\boldsymbol{D}'(i,j)$ can be determined by using $\varepsilon_2 = 2\varepsilon_1$ as the upper bound of the similarity of the matching point. The value d'_{ij} is:

$$d'_{ij} = \begin{cases} 1 & (d_{ij} > \varepsilon_2) \\ d_{ij} & (d_{ij} \le \varepsilon_2) \end{cases} \tag{6}$$

The modified similarity matrix can accelerate the convergence rate. We found that when the rendered image is similar with specific region of the current frame, the value will converge quickly. We take advantage of this distribution characteristic to carry out the coarse scan to lock the approximate area, and then using square error to describe the similarity of two regions that matched, the time for matching will decrease significantly. The method mentioned above is usually an iterative process. When the error is less than the threshold, the two are thought to be matching successfully.

3.2 Feature-Based Tracking

After initialization, we can get the correspondences between the 2D features in the reference frame and the 3D coordinates. The main target of subsequent tracking is to generate a new 3D-2D correspondences to update pose, based on the matching SURF features extracted from reference image and current image. And the RANSAC algorithm is used to remove the distortions. The new correspondences of 3D-2D can be generated according to the pre-computed camera pose and the new 2D transformation determined from feature-based matching. $\boldsymbol{H}_{ij}$ is a homography matrix, d_i and d_j is the feature points extracted from the reference frame and the current frame. The relationship among the feature points can be defined as:

$$d_i = \mathbf{H}_{ij} d_j \tag{7}$$

As for the follow-up feature-based tracking, richness of texture among objects impacts the quality and number of extracted features, which is the key factor in pose calculation and follow-up matching. Therefore, for the object with different texture, the sufficient feature points should be obtained by self-adaptive, which increases the robustness of the system and can also achieve better performance for various objects. The value of Hessian determinant is the primary gist for filtering the feature points. Therefore, we proposed self-adaptive method for extracting SURF features. Since the amount of the feature points can be obtained in the current image $F_{num}(I_i)$, and the quantitative range of feature points $\delta_1 \sim \delta_2$ is achieved in accordance with the test in advance, the method to adaptively extract SURF feature points from different objects is as follows:

$$\begin{cases} \rho_{+}(\det(Hessian)_{thre}) & F_{num}(I_i) > \delta_2 \\ \rho_{-}(\det(Hessian)_{thre}) & F_{num}(I_i) < \delta_1 \end{cases} \tag{8}$$

ρ_{+} and ρ_{-} are the functions used to dynamically adjust the Hessian determinant threshold. In order to decrease the computing time, the threshold is only adjusted at the beginning of feature-based tracking, and the acceptable maximum times of adjustment is set as well. When the times of iteration are over to maximum times but do not meet the conditions, the current threshold will be deemed to be the parameters of extracted SURF features.

Because the texture information of an object is concentrated in a small area, when the region is disturbed by the environment, the tracking and subsequent calculation would be affected seriously. The situation such as partial cover or outflow boundary of the object is inevitable during tracking process. And when it occurs, we hope that the remaining part can be used for tracking. As we know, the extracted feature points tend to focus on the richly-textured regions of the object, so a method to solve this problem is to weaken the dense distribution of feature points. Only when the feature points are scattered, the remaining parts of the object which is sheltered or out of boundary can also have feature points to continue tracking. In order to solve the problem above, we propose an algorithm on the basis of SURF algorithm which are obtained by segmenting the image and using multi-threshold.

The current image is divided into N pieces, respectively named as P_1, P_2, …, P_N. The number of the pieces is usually set as 4 or 9. Then the feature points can be detected from the image which has been segmented. Its descriptors need to be extracted and merged:

$$F = \{f_1\} \cup \{f_2\} \ldots \cup \{f_n\} \tag{9}$$

The distribution of feature is more dispersed after segmentation. But if each input frame is processed, it will undoubtedly increase computation. After initialization and self-adaptive feature extraction of the object, the number of features in the current frame can be used as the threshold. Once the area with rich textures on the object is affected, the number of feature points will decrease sharply. When the number of feature points in the current frame and the previous frame meet the (10), the input image will be processed by segmentation and feature extraction by multi-threshold:

$$F_{\mathrm{num}}(I_j^{'}) < \sigma F_{num}(I_j) \tag{10}$$

σ is related to the proportion of texture-rich area in total area, and the better results can be achieved when the value of σ is 0.6.

3.3 Robustness of System

Comparing 3D tracking with 2D tracking, the visual features are different while the viewpoint changes. It's necessary to change reference image. When the quantity of matching points is lower than the threshold we set, we can say that the current frame has not been tracked correctly. The video frame which can get the correct camera posture before tracking failure ought to be saved as the new reference frame. Moreover, if the amount of matching points in the next 20 frames still cannot reach the threshold, the process of tracking failed, and the system will return to the initial state and restart. To sum up, the methods proposed in this paper can be utilized for a variety of objects, and they have strong anti-jamming in the tracking process. Situations such as image blur caused by high movement, object covered, out of bond and so on, the system can run without manual intervention.

4 Experiments and Results

In this paper, not only the method proposed has been tested in practice, but also the results of the tracking and pose estimation have been evaluated. The hardware platform used in this experiment is Intel Core i7-4720 2.6 GHz and software platform is visual studio 2010, with OpenCV and OpenGL as additional Dependencies. The Gsou USB monocular camera is adopted in the experiment, whose frame rate of image sequence is 25 frame/s, and resolution of video image is 640 $\times$ 480 pixels.

Thus we will translate CAD model data into a format of OBJ text file, in order to complete the system through the language C more conveniently.

4.1 Initialization Test

Through this experiment, it can be verified the initial pose is achieved by matching the video frame with the template image, which is a CAD model rendered by CG. The object we chose is a pedestal of mechanical arm in the experiment. The camera's six degrees of freedom parameters $E = [r_x, r_y, r_z, t_x, t_y, t_z]$ can be used to represent the position of the camera, where r_x, r_y and r_z respectively represent rotation angles around the x, y and z axes. Similarly t_x, t_y and t_z respectively represent translation.

We set the initial pose in the matrix $E_0 = [0, 0, 0, 0, 0, 0]$, which is used to render the CAD model to get the template image. After the video sequence frame matches the template image successfully, it can be regarded as initializing successfully and that the location of the object L_0 is recorded meanwhile. Afterwards, we keep the location of camera and move the object until 200 frames. On the premise of correct tracking, the new camera pose E_{end} and the final position of the object L_{end} should be recorded by the system, then finish tracking. We keep the camera and object motionless, the camera pose E_{end} is used to reinitialize the system. At this moment, the template image obtained by rendering the CAD model in the current state must match the input video frame successfully.

With the location of camera still fixed, we move the object to the initial position, and get the corresponding pose $E_0' = [-0.1, 0.4, 0, 0.3, -0.5, 0.6]$, which has small deviation with the initial given pose. Therefore, the method of initialization based on the template matching can complete estimating the initial pose.

4.2 Tracking Results and Re-initialization Test

In the experiment, the camera pose, which is estimated above, is used to render a virtual cube model. The accuracy of tracking and pose estimation was determined by judging whether the cube is fit with the being tracked object and the movement trend is consistent with the actual situation. Two different video sequences are used to verify the system stability of the subsequent tracking. In the first video sequence, the camera and the tracked object are always static. The Fig. 3 respectively express rotation and translation parameters calculated by the system according to the sequence. In the case of fixed position, the trend of the six parameter values is approximately referred as a straight line in the Fig. 3(a), which meets the physical situation. In addition, the pictures in Fig. 4(a) are the 50th, 80th, 150th and 180th frame after rendering the virtual cube in the first sequence and we find that the state of virtual cube model has not changed. The second video sequence is obtained by rotating, translating and scaling the tracked object. We can find the 6-DOF parameters in Fig. 3(b) without burr which prove the estimation is accurate. Separately, the pictures in Fig. 4(b) are the 50th, 150th, 200th and 240th frame after rendering the virtual cube by 6-DOF parameters in second sequence. In the experiment, the virtual cube is basically accord with the state of the object, even with the problems like out of boundary and partial occlusion.

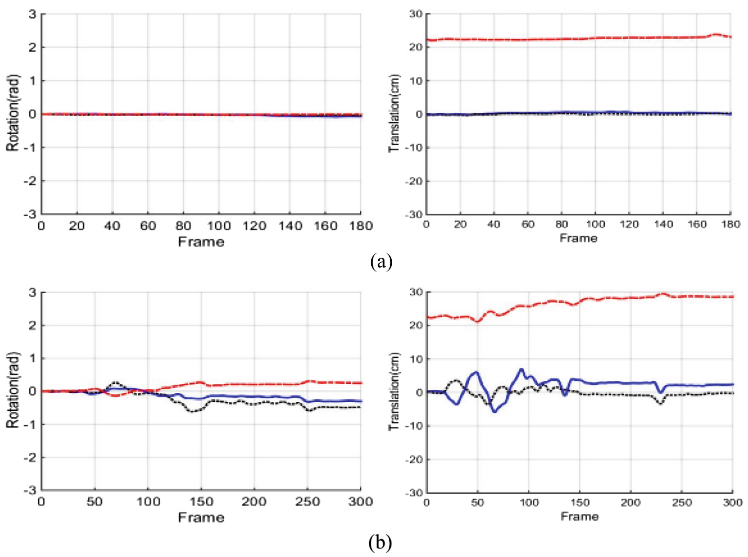

Fig. 3. 6-DOF pose of the pedestal of mechanical arm in tracking test. The first row (a) shows the rotation and translation of the first sequence while the camera and object are always static. The second row (b) shows the parameters of the second sequence while the motion of object is random. (blue solid = x-axis, black dotted = y-axis, red dashed = z-axis) (Color figure online)

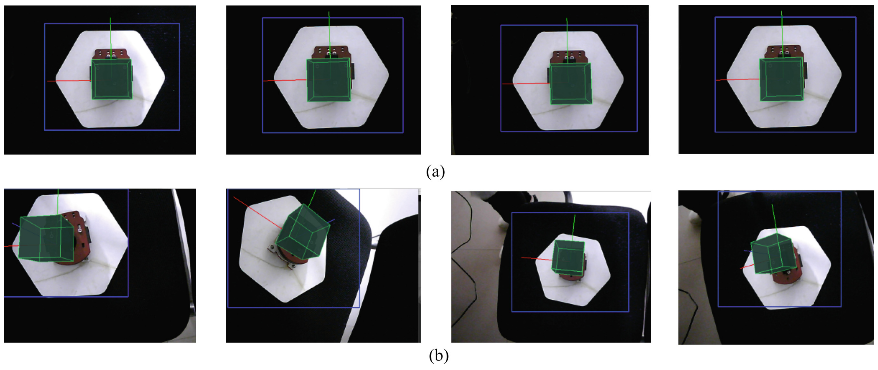

Fig. 4. Results on different input sequences. The first row (a) from the first sequence shows the virtual cube is changeless. The second row (b) from the second sequence the virtual cube is consistent with the movement of object.

5 Conclusions

In this paper, we have proposed a robust tracking system for 3D object recognition and tracking based on the combination of two different approaches. By using template-based matching in initialization, we can give an accurate initial pose for marker-less visual tracking. Based on the initialization, we use SURF descriptors in remainder tracking to achieve the system robust against the changes of rotation and scale. The system has been tested on real scene and show good results. There is still some work that we should be done to reduce the computation. We hope that the system can be applied to more scenes.

Acknowledgments. The work is sponsored by the Seed Foundation of Innovation and Creation for Graduate Students in Northwestern Polytechnical University (No. Z2017139).

References

1. Harris, C.: Tracking with Rigid Objects. MIT Press, Cambridge (1992)
2. Drummond, T., Cipolla, R.: Real-time visual tracking of complex structures. IEEE Tran. Pattern Anal. Mach. Intell. **24**(7), 932–946 (2002)
3. Wuest, H., Stricker, D.: Adaptive line tracking with multiple hypotheses for augmented reality. In: Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 62–69 (2005)
4. Gouet, B.L.V.: SAP: a robust approach to track objects in video streams with snakes and points. In: 15th British Machine Vision Conference, vol. 2, pp. 737–746 (2004)
5. Harris, C., Stephens, M.: A Combined corner and edge detector. In: Proceedings of the Fourth Alvey Vision Conference, pp. 147–151 (1988)
6. Bay, H., Tuytelaars, T.: SURF: speeded up robust features. Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
7. Rubble, E., Rabaud, V.: ORB: an efficient alternative to SIFT or SURF. In: International Conference on Computer Vision, vol. 11, no. 58, pp. 2564–2571 (2012)
8. Vacchetti, L., Lepetit, V., Fua, P.: Combining edge and texture information for real-time accurate 3D camera tracking. In: International Symposium on Mixed and Augmented Reality, pp. 48–57 (2004)
9. Marchand, E., Chaumette, F.: Real-time markerless tracking for augmented reality: the virtual visual servoing framework. IEEE Trans. Vis. Comput. Graph. **12**(4), 615–628 (2002)
10. Choi, C., Christensen, H.I.: Real-time 3D model-based tracking using edge and keypoint features for robotic manipulation. In: IEEE International Conference on Robotics and Automation (ICRA), Alaska, USA (2010)