International Journal of Automation and Computing
www.ijac.net

18(2), April 2021, 288-299
DOI: 10.1007/s11633-020-1265-9

A Tracking Registration Method for Augmented Reality Based on Multi-modal Template Matching and Point Clouds

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

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

Abstract: In order to overcome the defects where the surface of the object lacks sufficient texture features and the algorithm cannot meet the real-time requirements of augmented reality, a markerless augmented reality tracking registration method based on multi-modal template matching and point clouds is proposed. The method first adapts the linear parallel multi-modal LineMod template matching method with scale invariance to identify the texture-less target and obtain the reference image as the key frame that is most similar to the current perspective. Then, we can obtain the initial pose of the camera and solve the problem of re-initialization because of tracking registration interruption. A point cloud-based method is used to calculate the precise pose of the camera in real time. In order to solve the problem that the traditional iterative closest point (ICP) algorithm cannot meet the real-time requirements of the system, Kd-tree (k-dimensional tree) is used under the graphics processing unit (GPU) to replace the part of finding the nearest points in the original ICP algorithm to improve the speed of tracking registration. At the same time, the random sample consensus (RANSAC) algorithm is used to remove the error point pairs to improve the accuracy of the algorithm. The results show that the proposed tracking registration method has good real-time performance and robustness.

Keywords: Augmented reality, markerless, tracking registration, LineMod, iterative closest point (ICP) algorithm.

Citation: P. X. Cao, W. X. Li, W. P. Ma. A tracking registration method for augmented reality based on multi-modal template matching and point clouds. *International Journal of Automation and Computing*, vol.18, no.2, pp.288–299, 2021. http://doi.org/10.1007/s11633-020-1265-9

1 Introduction

Augmented reality[1] is a technology that seamlessly merges the real world with the virtual world. Users can achieve a sensory experience that transcends reality by superimposing computer-generated virtual information onto real scenes. In order to achieve this "seamless" fusion, it is necessary to use object recognition methods to detect enhanced objects and estimate the pose information of the camera relative to the real scene. The technique of accurately aligning virtual information with real scenes based on pose information is tracking registration technology[2]. Tracking registration technology is the core of developing augmented reality systems, and has become a key issue that needs to be solved urgently for augmented reality to be more widely used. The augmented reality systems at this stage are mainly implemented by the methods of pre-placement of artificial identification in real scenes. Those methods have the advantages of small amounts of calculation, fast execution speed, and do not require complicated hardware. However, the methods based on artificial identification have defects such as visual pollution and poor robustness[3].

With the development of science and technology and the improvement of computer computing power, markerless tracking registration methods have become a research hotspot in augmented reality technology and have made great progress[4]. But, there are still different degrees of defects. For example, the methods based on natural feature points[5, 6] are prone to tracking jitter or disturbance in the environment where the surface of the objects lack sufficient texture features; the methods based on edge contours[7] are more sensitive to the cluttered background and are less robust when the targets are partially blocked. The tracking registration methods based on simultaneous localization and mapping (SLAM)[8, 9] do not require prior knowledge of the scene and eliminate the training process, but those methods can only estimate the relative pose of the camera and cannot identify objects in the scene. They are not suitable for many augmented reality systems. In addition, those methods are easy to cause tracking failure in dynamic scenes. Model-

Research Article
Manuscript received June 22, 2020; accepted October 20, 2020; published online January 30, 2021
Recommended by Associate Editor Xian-Dong Ma
Colored figures are available in the online version at https://link.springer.com/journal/11633

based methods have been widely used in markerless augmented reality 3D tracking registration systems, but real-time processing of a large number of reference images collected from different perspectives will generate a huge search space and a huge amount of calculations, thereby greatly reducing the real-time performance and availability of those methods[10]. Tracking registration methods based on dense point clouds can show good robustness in scenes with low ambient light intensity and a lack of texture on the target surface[11]. Those methods generally use iterative closest point (ICP) registration algorithms to iteratively calculate all or part of the point cloud, which can reduce the requirements of the lighting conditions and the accuracy of pose estimation can also meet the requirements. However, when the depth sensor is moving quickly and the correct initial corresponding data point set cannot be obtained, the iterative process of the ICP algorithm may fall into a local optimum or even cause a frame drop situation, which may cause the interruption of the tracking registration process[12].

In order to overcome the difficulty of detecting objects due to a lack of sufficient texture features on the object surface, the LineMod algorithm is used to effectively detect the texture-less target in a complex background environment or when the target is partially occluded. At the same time, the LineMod algorithm was improved in order to have the characteristics of scale invariance. Aiming at the problem that the multi-modal template matching algorithm cannot meet the real-time requirements of augmented reality and the tracking registration methods based on point clouds can easy cause interruption in the tracking registration process, this paper proposes a tracking registration method for augmented reality based on multi-modal template matching and point clouds. First, the linear parallel multi-modal LineMod template matching method with scale invariance is used to detect the texture-less target to obtain the key frame and the initial pose. Then, the improved ICP algorithm is used to obtain the precise position of the camera. In the process of accurately calculating the camera pose, the improved LineMod template matching method will be restarted to obtain the key frame and the initial pose when the camera moves quickly and the tracking registration is interrupted. This method can meet the requirements of the augmented reality system for the accuracy and robustness of the tracking registration algorithm, and it can also be applied to the scene where the targets lack texture features.

2 System overview

The basic idea and work flow of the markerless augmented reality tracking registration method based on multi-modal template matching and point clouds is shown in Fig. 1. First, in the offline training phase, the multi-view target reference images are collected in a com-

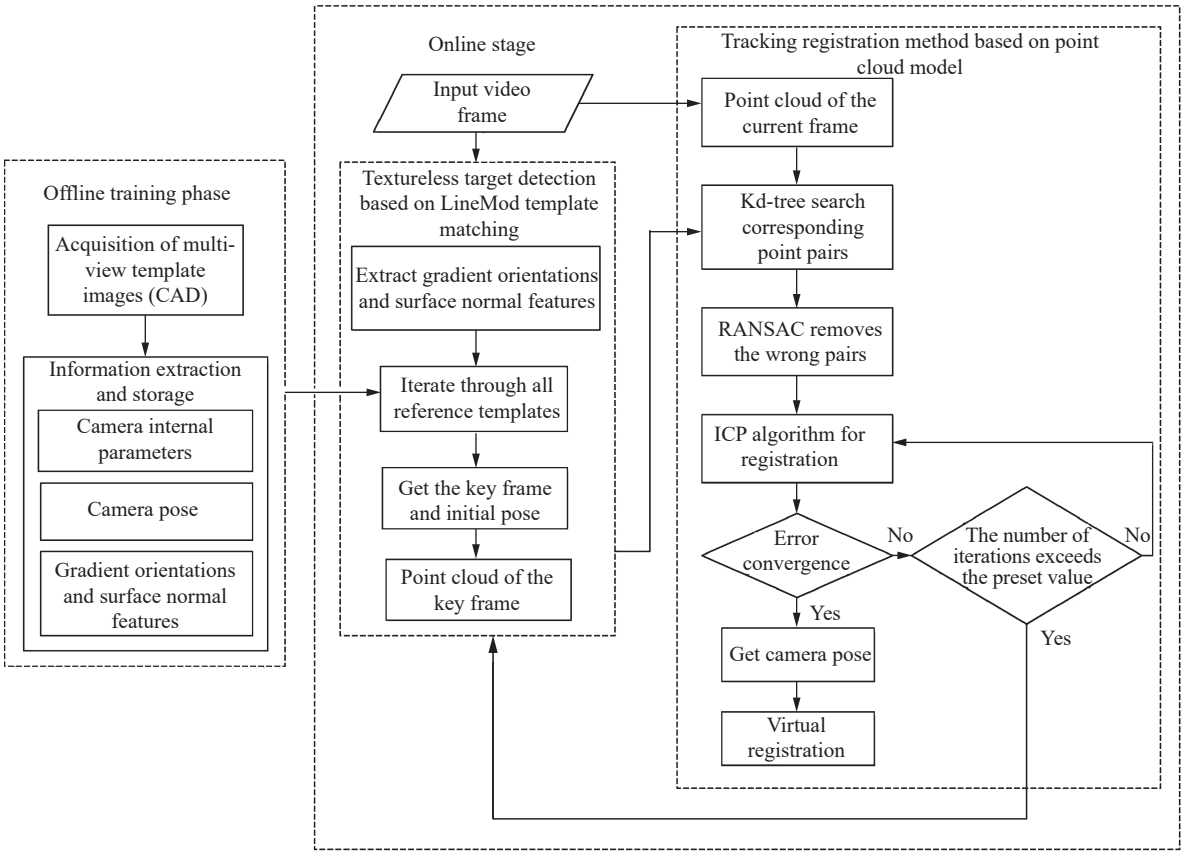


Fig. 1 Algorithm ideas and work flow

puter-aided design (CAD) environment, and the camera′s collection position and posture are recorded during the collection process. At the same time, the gradient orientations and surface normal features of the reference images are extracted and stored. In the online stage, video frames of real objects are obtained, and the gradient orientations and surface normal features of each frame of the image are extracted. Then the improved LineMod template matching method is used to quickly obtain the key frame that is most similar to the current video frame and the corresponding camera pose. Finally, the tracking registration method based on the point cloud model is used to calculate the current pose of the camera. In the process of tracking registration based on the point cloud model, in view of the shortcomings of the classic ICP algorithm, the Kd-tree (k-dimensional tree) is used to search for the nearest neighbor point pairs of the key frame point cloud and the current frame point cloud under the graphics processing unit (GPU), and the random sample consensus (RANSAC) algorithm is used to remove error point pairs. In the process of using the improved ICP algorithm to obtain the camera pose, if the average distance of the matched point pairs is too large and the registration fails, then the improved LineMod template detection algorithm will be re-run to obtain the key frame.

3 Textureless target detection based on improved LineMod template matching

The LineMod algorithm is an efficient multi-modal template matching algorithm proposed by Hinterstoisser et al.[13] in 2011. This algorithm uses red-green-blue (RGB) images and depth images to identify textureless objects in cluttered scenes. Similar to the general template matching algorithm, the process of using LineMod for object recognition can be roughly divided into two stages: the offline training stage and online detection stage. In the offline training phase, the template images of the target object are collected under multiple perspectives and different sizes, and then the features are extracted from these template images and saved as a database file. In the online detection stage, the features of the scene image and the template image are extracted and compared in a sliding window. If the threshold is greater than a preset threshold, the part of the window is considered as the detected target object. The general template matching methods are based on the low-level features of the image to design similarity measurement criteria. The LineMod multimodal template matching algorithm uses the gradient orientations of the RGB image and the surface normal features of the depth image as the basis for template matching to define the template T, as shown below:

$$T = (\{O_m\}_{m\in M}, P(r, m))$$

where $\{O_m\}_{m\in M}$ is a series of reference images of the target. M includes a set of modalities which represents RGB image or depth image. P is a list of pairs (r, m) made of the locations r of a discriminant feature in modality m. The template T records the features at the position r in the picture O. We consider the multi-modal template features defined by LineMod. In the RGB image, due to the good robustness of the gradient orientations under illumination and noise conditions, only the main gradient orientations on the contour of the target object are retained. In the depth image, since the normal vectors on the boundary of the target object in the depth image cannot often be estimated reliably, the surface normal features inside the contour are mainly considered. The template features of the LineMod multimodal template matching algorithm can be represented as shown in Fig. 2.

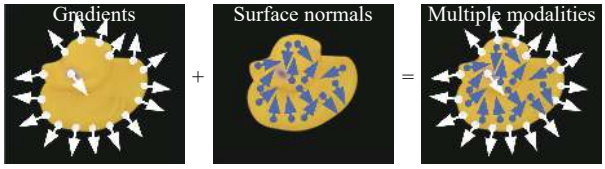


Fig. 2 Template features of LineMod multimodal template matching algorithm

The method of LineMod template matching detects the object in the scene image according to the defined similarity criterion in the form of a sliding window. For the input image I and the template T, the similarity measure at the position c is calculated as shown in (1):

$$\varepsilon(\{I_m\}_{m\in M}, T, c) = \sum_{(r,m)\in P} \left(\max_{t\in R(c+r)} f_m(O_m(r), I_m(t)) \right) \tag{1}$$

where the function $f_m(O_m(r), I_m(t))$ calculates the similarity score for modality m between the template image at the location r and the input image at the location t. $t \in R(c+r) = \left[c+r-\frac{\tau}{2}, c+r+\frac{\tau}{2}\right] \times [c+r-\frac{\tau}{2}, c+r+\frac{\tau}{2}]$ represents the neighborhood of size τ in the input image with the location $c+r$ as the center, and c is the center point on the input image corresponding to the template image.

3.1 Processing the RGB image

For the calculation of the gradient orientations of the RGB image, we calculate and normalize the gradient orientations on the three channels at each pixel location and then the maximum one is taken as the gradient orientation at that location. That is, the gradient orientation at the location x of the RGB color input image I can be computed as

$$I_g(x) = ori(\hat{C}(x)) = \frac{\partial \hat{C}}{\partial x} \tag{2}$$

where $\hat{C}(x) = \underset{C\in\{R,G,B\}}{\arg\max} \left\| \frac{\partial C}{\partial x} \right\|$, R, G, B represent the three channels of red, green, and blue of the color image. Therefore, the similarity score in the RGB image is

$$f_g(O_g(r), I_g(t)) = \left| O_g(r)^{\mathrm{T}} I_g(t) \right| \tag{3}$$

where O_g represents the gradient orientations of the reference image at the location r. I_g represents the gradient orientations of the input image at the location t. In order to reduce the influence of illumination changes and occlusion on target detection, the absolute value of the cosine is taken for the difference in gradient orientations. Therefore, in the RGB image, the similarity measure between the input image I and the template image I is

$$\varepsilon(I, T, c) = \sum_{(r,m)\in P} \left(\max_{t\in R(c+r)} f_m(O_m(r), I_m(t)) \right) = \sum_{r\in P} \max_{t\in R(c+r)} \left| \cos(ori(O, r) - ori(I, t)) \right| \tag{4}$$

where $ori(O, r)$ is the gradient orientations of the reference image O at the location r. $ori(I, t)$ is the gradient orientations of the input image at the location t. P is a list of r. $T = (O, P)$.

In order to reduce the amount of calculation, the gradient orientation is discretized to obtain n discrete spaces. The gradient orientation of any pixel position is determined by the pixels in the domain. As shown in Fig. 3, take the collinear direction as one direction, and unity all the gradient directions into the interval $0° - 180°$. Then, the gradient direction space in the interval is divided into 5 equally, and the binaries are used to represent the directions. The pink direction in Fig. 3 is closest to the second discrete azimuth, so it is represented by 00010. Moreover, in order to improve the robustness of the quantized gradient to noise, the direction of the quantized gradient is propagated to a 3×3 neighborhood. The gradients at each pixel position whose norms are greater than a threshold were retained.

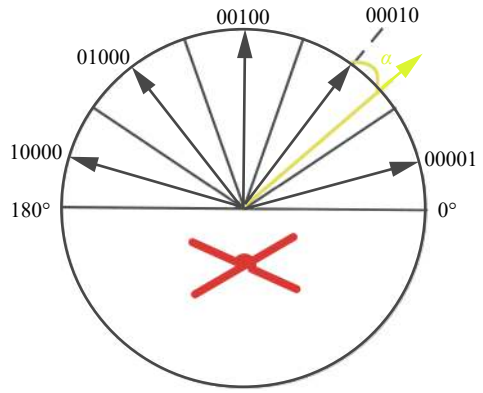


Fig. 3 Quantizing the gradient orientations

3.2 Processing the depth image

For the calculation of the surface normal features of the depth image, around each pixel position x, we consider the first-order Taylor expansion of the depth function $D(x)$, as shown in (5).

$$D(x + \mathrm{d}x) - D(x) = \mathrm{d}x^{\mathrm{T}} \nabla D + h.o.t. \tag{5}$$

In each pixel neighborhood, $\mathrm{d}x$ always satisfies the constraints of the formula. Then, the least square method is used to find the optimal gradient $\widehat{\nabla D}$. The depth normal vector at this pixel location corresponds to a three-dimensional plane, which passes through three points X, X_1 and X_2.

$$\begin{cases} X = \overrightarrow{v}(x) D(x) \\ X_1 = \overrightarrow{v}(x + [1, 0]^{\mathrm{T}}) D(x + [1, 0] \widehat{\nabla D}) \\ X_2 = \overrightarrow{v}(x + [0, 1]^{\mathrm{T}}) D(x + [0, 1] \widehat{\nabla D}) \end{cases} \tag{6}$$

where $\overrightarrow{v}(x)$ is the line-of-sight vector calculated from the depth camera's internal parameters via pixel x. The surface normal of pixel x can be obtained by cross-multiplying the vectors $\overrightarrow{XX_1}$ and $\overrightarrow{XX_2}$, and then be normalized. However, due to the defects of the acquired depth images, the numerical values are often abrupt. Therefore, the calculation of the normal vectors at the edge of the object are often inaccurate, and the normal vectors obtained by directly expanding the first-order Taylor function at each pixel point are often invalid. Considering the similar processing method of the bilateral filter[14], LineMod ignores the neighboring pixels whose depth difference of the center pixel is greater than the set threshold when calculating the normal vector. When the normal vectors are quantized, this method effectively smooths the noise of the surface of the depth image, and can also estimate the surface normal vector when the depth is not continuous. $f_m(O_m(r), I_m(t))$ is used to calculate the similarity score, so the similarity score in the depth image is

$$f_D(O_D(r), I_D(t)) = O_D(r)^{\mathrm{T}} I_D(t) \tag{7}$$

where $O_D(r)$ is the normalized normal of the template image at location r. $I_D(t)$ is the normalized normal of the input image at location t. In order to make the algorithm have the characteristics of scale invariance, the depth information of the image is introduced into the calculation process of the similarity measure. Therefore, the similarity measure formula can be[15]:

$$\begin{cases} \varepsilon(I, T, c'_I) = \sum\limits_{(c'_O + r')\in p} \max\limits_{t\in R(S_I(c'_I, r'))} \left| \cos[ori(O, S_O(c'_O, r')) - ori(I, t)] \right| \\ S_x(c'_x, y) = c'_x + \dfrac{D(c'_x)}{D(c'_O)} y \end{cases} \tag{8}$$

where

$$R(S_I(c'_I, r')) = \left[c'_I + \frac{D(c'_I)}{D(c'_O)}r' - \frac{\tau}{2}, c'_I + \frac{D(c'_I)}{D(c'_O)}r' + \frac{\tau}{2}\right] \times \left[c'_I + \frac{D(c'_I)}{D(c'_O)}r' - \frac{\tau}{2}, c'_I + \frac{D(c'_I)}{D(c'_O)}r' + \frac{\tau}{2}\right] \tag{9}$$

$ori(O, S_O(c'_O, r'))$ is the gradient radian of the template image O at location $c'_I + \frac{D(c'_I)}{D(c'_O)}r'$. $R(S_I(c'_I, r'))$ represents the area with $c'_x + \frac{D(c'_x)}{D(c'_O)}r'$ as the center and τ as the neighborhood. $D(c'_x)$ is the depth at point c'_x. $D(c'_O)$ represents the distance from the coordinate origin of the reference image O to the camera coordinate origin during offline training. Since the position of the object during training is fixed and is known, the zoom scale of the template can be obtained from the online depth information according to (8). Then, a sliding window is used in the input image. In the sliding window, all similarities at the position list p are added to determine the similarity between the input image and the reference image. All reference images are traversed, and the image with the highest similarity to the current input image is used as the key frame to determine the initial pose of the current camera.

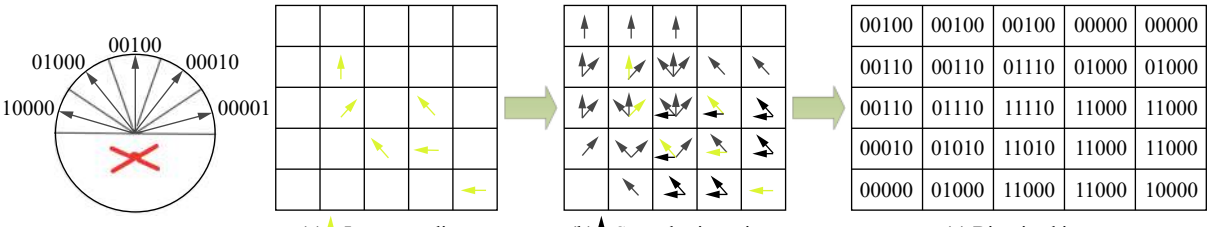
Similar to the processing of gradient directions in the RGB image, the direction of the normal vector of the depth image is usually quantized into N spaces. In order to improve robustness, the normal vector of the pixel is propagated to a 5×5 neighborhood. The final normal of the pixel is determined by the largest number of normal vectors in its neighborhood.

3.3 Spreading the orientations

In (1), in order to calculate the similarity measure between the template image and the input image, the Max operation is used to find the maximum modality. In order to avoid the Max operation for each match and improve the matching speed, the modalities of each position are expressed in a binary manner. LineMod uses multiple templates to match the input image at the same time. For an RGB-D scene image, we firstly calculate the gradient orientations or surface normals of each pixel position and quantize to N spaces. Then, each quantized direction is encoded as a binary string. Taking the gradient orientation as an example[16], its direction diffusion is shown in Fig. 4. After the feature direction is quantized, in order to make the match have a certain degree of fault tolerance, the features of each pixel position on the image are propagated to other neighborhoods. Therefore, one pixel position of the image contains multiple features propagated by the neighborhood. At the same time, the original image is expressed in a new way. LineMod creates N corresponding lookup tables for each discrete orientation in advance. Each corresponding lookup table is represented in a binary manner, and the index value corresponding to the index number of each lookup table is the cosine value of the pixel point and the direction of the discrete feature. When the template features are used to match the sampled sliding window of the scene image, LineMod computes the similarity measure of the sliding window's features with the template features through binary encoding. Therefore, the feature matching in the entire process is converted into a search method.

4 Tracking registration method based on point clouds

After the key frame is obtained by the texture-less object detection method based on improved LineMod template matching, the training pose of the key frame template is used as the initial pose $M_{init} = [R_{init}|t_{init}]$. At the same time, the depth image of the key frame is mapped into three-dimensional target template point cloud data combined with the camera internal parameters. Define the target template point cloud as $P = \{p_1, p_2, \cdots, p_{N_p}\}$. The camera internal parameters are used to map the input depth image into 3D environment point cloud data $X = \{x_1, x_2, \cdots, x_{N_x}\}$. Afterwards, a method based on point cloud models is used for tracking registration, i.e., the ICP algorithm is used to register the environmental point cloud with the model point cloud, and the real-time pose of the camera is estimated to complete the tracking registration. In order to further improve the speed of the algorithm, this process uses Kd-tree to perform the nearest neighbor search to increase the search speed of the corresponding point. At the same time, the graphics processor GPU is used to accelerate the point cloud registration process. In addition, in order to improve the accuracy of the algorithm, the RANSAC



00100	00100	00100	00000	00000
00110	00110	01110	01000	01000
00110	01110	11110	11000	11000
00010	01010	11010	11000	11000
00000	01000	11000	11000	10000

Fig. 4 Spreading the gradient orientations

algorithm is used to eliminate erroneous point pairs. If the average distance of the matched point pairs is too large and the registration fails, then the improved LineMod template detection algorithm is re-run to obtain the key frame.

4.1 Point cloud generation

Let each pixel at time i in the depth image be $u = (x, y)$. The computer unified device architecture (CUDA) parallel computing platform of NVIDIA is used to back-project to the depth camera space coordinate system to obtain the vertex map. The back-projection result is

$$V_i(u) = D_i(u)M^{-1}_{\text{int}\_D}[u, 1] \tag{10}$$

where $D_i(u)$ is the depth image at time i and M_{int_D} is the internal parameters of the depth camera.

From Section 3, the initial camera pose at time i is $M_{init} = [R_{init}|t_{init}]$, where R_{init} is the rotation matrix of 3×3 and t_{init} is the three-dimensional translation vector. Equation (10) can be transformed into the global coordinate space through the initial camera pose.

$$V_{3D}(u) = V_i(u)M_{init}. \tag{11}$$

At the same time, the normal vector $N_{3D}(u)$ corresponding to each vertex is obtained by using adjacent projection points and doing the following operations.

$$N_{3D}(u) = (V_{3D}(x+1, y) - V_{3D}(x, y)) \times (V_{3D}(x, y+1) - V_{3D}(x, y)).$$

The normal vector is normalized to get the unit normal vector $N_{3D}(u)$. In order to improve the registration speed, the voxel grid method is used to down-sample the original point cloud data to obtain the template 3D point cloud data $P = \{p_1, p_2, \cdots, p_{N_p}\}$. The world coordinate origin and camera origin are set to coincide and the same method is used to get the environment point cloud $X = \{x_1, x_2, \cdots, x_{N_x}\}$. Fig. 5 shows an example of model point cloud generation. Then, an improved ICP algorithm is used to register the template point cloud and the environment point cloud to obtain a precise pose.

4.2 Classic ICP algorithm

The ICP registration algorithm is an iterative closest point registration algorithm, which is a registration algorithm based on free-form surface[17]. The essence of the ICP algorithm is to continuously rotate and translate by matching between points, and the least square method is used as a measure until the distance between points reaches a preset threshold. The core idea of the process of the ICP algorithm is: We know two point cloud data sets

(a) Color map of the model (b) Depth map of the model

(c) Model point cloud (d) Model point cloud after voxelization

Fig. 5 Model point cloud generation

P and X to be registered and select a point set p_i in P, where p_i is a subset of P. We need to find the point set x_i with the shortest Euclidean distance from p_i in the point cloud X, and p_i and x_i are used as the corresponding point set pairs to obtain the transformation matrix. Through continuous iteration, the error function $f(R, T)$ in (12) is minimized to obtain the optimal transformation matrix R and T[18].

$$f(R, T) = \frac{1}{k}\sum_{i=1}^{k} \|x_i - (Rp_i + T)\|^2 \tag{12}$$

where R is the rotation matrix, T is the translation matrix.

The essence of the ICP algorithm is to calculate the transformation matrix of the source point cloud and the target point cloud, and the registration error of the two point clouds is minimized by the way of rotation and translation to achieve the best registration effect. In this paper, $P = \{p_1, p_2, \cdots, p_{N_p}\}$ is the template point cloud and $X = \{x_1, x_2, \cdots, x_{N_x}\}$ is the environmental point cloud. The following part introduces the registration process of two point clouds using the ICP algorithm[19].

1) Sample the template point cloud P and take the point set $p_i \in P$.

2) For each point in p_i, the method of shortest Euclidean distance is used to search for the corresponding closest point from X to get the point set x_i.

3) Use algorithms or limited conditions to remove the incorrect correspondence.

4) Calculate the transformation matrix of the two point set which is obtained in Step 2). Through continuous iteration, the error function $f(R, T)$ in (12) is minimized to obtain the optimal transformation matrix R and

T. Then, we can obtain the new point set $p_i' = \{p_i' = Rp_i + T, p_i \in P\}$.

5) Determine whether the iteration is terminated according to $d = \frac{1}{n}\sum_{i=1}^{n} \left\|p_i' - x_i\right\|^2$. If d is greater than the set threshold τ, then, return to the process (2) to continue the iteration. If d is less than the set threshold τ, the registration is successful. If the number of iterations reaches the maximum number of iterations, the registration fails and the registration algorithm is exited. At the same time, the LineMod template matching algorithm is used to reacquire the key frame.

4.3 Improved ICP algorithm

Compared with other registration algorithms, the ICP algorithm has many advantages. It can directly register through the corresponding point without segmenting the point cloud and other processing, and its registration effect can also meet the required accuracy requirements[20]. However, the ICP algorithm also has deficiencies. First, the ICP algorithm enables each point in the point cloud to participate in the search for the corresponding point. When there are many points in the point cloud, the registration process takes a long time. Secondly, incorrect point pairs may appear in the search process of corresponding points of the ICP algorithm. In view of the above deficiencies, this article uses Kd-tree to replace the part of the original ICP algorithm to find the nearest point under the GPU, which can accelerate the registration process of the traditional algorithm and greatly improve the efficiency of the algorithm[21]. For the situation where the search for the corresponding point may be wrong during the registration process, the RANSAC algorithm is used to remove the erroneous point pairs through the setting of the distance threshold.

Kd-tree is a kind of data structure that divides a k-dimensional data space. It is mainly used in the search of key data in multi-dimensional space. It is a special case of a dividing tree in binary space[22]. The Kd-tree algorithm is divided into two parts, the first part is to create a Kd-tree, and the second part is to search for the nearest neighbor node. The creation of the Kd-tree is divided into a recursive method and non-recursive method. In order to improve efficiency, this paper uses non-recursive methods to create a Kd-tree, and the search process also uses non-recursive methods. Therefore, this paper divides the search process into two steps: first, it searches down to the leaf node to find the first nearest neighbor reference node, and then the parent node information is used to perform a retrospective search for the nodes that meet the conditions. The data structure of Kd-tree greatly facilitates the search of a single datum in k-dimensional space. In order to further improve the efficiency of the algorithm, the Kd-tree algorithm is transplanted to the GPU[23], and the purpose of further improving the efficiency of the algorithm is achieved by searching multiple data in parallel. In order to conveniently transfer the Kd-tree node data to the GPU, the Kd-tree nodes are stored in an array, and the child nodes of the current node are stored in an integer array to store their serial numbers for an easy search. In order to facilitate backtracking to find possible nearest neighbors, an integer data bit is set to indicate the serial number of the parent node. The use of an array structure to store Kd-tree nodes is mainly considered to meet the management of global memory under the GPU. For establishing a Kd-tree under the GPU, first, the input sample set is used to establish a Kd-tree under the CPU. Then, the Kd-tree nodes are converted into an array storage form under the CPU. Finally, the Kd-tree is transferred to the GPU in the global memory search.

RANSAC can estimate the parameters of the mathematical model from a set of observation data containing outliers through an iterative method[24]. The algorithm is an uncertain algorithm that calculates reasonable results based on a certain probability. The RANSAC algorithm first randomly selects a sample subset from the data set and uses the minimum variance estimation algorithm on the mathematical model of the subset. Then, the remaining data points are substituted into the model and the error is calculated. When the error is less than the preset threshold, this point is the inner sample point, otherwise it is the outer sample point. By comparing the number of sample points in each estimated model, the optimal model parameters are determined. The RANSAC algorithm obtains an effective data model by repeatedly selecting a set of random subsets in the point cloud data, eliminating erroneous or abnormal data points, and achieving robustness requirements[25]. In this paper, the algorithm is used to filter the corresponding points in the point cloud, remove the incorrect corresponding point pairs, and improve the calculation efficiency. The specific process is the following.

1) Supposing the sample data set is Q, the number of points in Q is $Num(Q) > n$. Randomly select n points in Q as a subset S and estimate the initial model M of the subset.

2) Substitute the remaining points in the sample data set Q into the model M and calculate the error between these points and the model M. The points whose error is less than the preset threshold and subset S form a new point set S_0. Then S_0 is the inner sample set of the initial estimation model.

3) If the number of points in S_0 is not less than the minimum number of internal sample points contained in the accurate model, that is $Num(S_0) \geq N$, then the model parameters are reasonable and the point set S_0 can be used to estimate the new model M_0.

4) Re-select a new subset S. Repeat the above steps, and compare the estimated model before and after. If the new model has too few internal points and it is not as

good as the previously estimated model, then the model will be discarded. If the new model has more internal points than the previous model, the old model will be replaced by the new model.

5) When the algorithm reaches the preset number of iterations, the final model is estimated if the error threshold is met.

5 Experimental results and analysis

The experiments on the algorithms in this paper were implemented in the environment of VS2017, OpenCV 3.4.1 and Python 3.7.3. The PC is Intel (R) Core (TM) i3-2370M, CPU frequency is 2.4 GHz and ROM 4 GB. The graphics card model is NVIDIA Geforce 610 M. A Kinect was used to capture images and videos. Taking public data sets as an example, its RGB sensor resolution is 640×480 and its depth sensor resolution is 640×480.

Firstly, for targets lacking sufficient texture on the surface, the improved LineMod template matching algorithm with scale invariance is used for target detection to obtain the key frame and initial pose. According to Fig. 6, the textureless target can be accurately detected under different viewing angle sizes, scales and illumination conditions. In particular, the textureless target can still be accurately detected when the target was partially occluded.

In this paper, the LineMod algorithm of literature[13] is improved, so that the improved LinMod algorithm has scale invariance. The improved LineMod algorithm can effectively detect the target in a complex background environment and when the target is partially occluded. If the improved LineMod algorithm is used to obtain the pose and directly perform registration rendering, the result is shown in Fig. 7. We can see that there is a large error between the green target registered according to the obtained pose and the actual target, which means that directly applying the improved LineMod template matching algorithm to augmented reality cannot meet the requirements of augmented reality for pose accuracy. Therefore, this paper uses the improved ICP algorithm to obtain a more accurate pose. Directly using the ICP algorithm for tracking and registration requires a higher initial pose, otherwise it is easy to fall into a local optimum. Therefore, we combine the two algorithms, the improved multi-modal LineMod template matching algorithm is used for textureless target detection to obtain the initial pose, and then the improved ICP algorithm is used to implement the augmented reality tracking registration process, as shown in Figs. 8 and 9. Fig. 8 annotates the pose information of the tracking registration target, and Fig. 9 shows the registration results obtained according to the algorithm in this paper. From Figs. 9(a)–9(c), it can be seen that under different perspectives and scales, the registered green target and the actual tracked target coincide well. From Fig. 9(d), it can be seen that even when the target is partially occluded, a good registration effect can still be obtained. This is because the improved ICP algorithm can accurately obtain the pose of the camera. Even if the tracking registration fails, the method in this

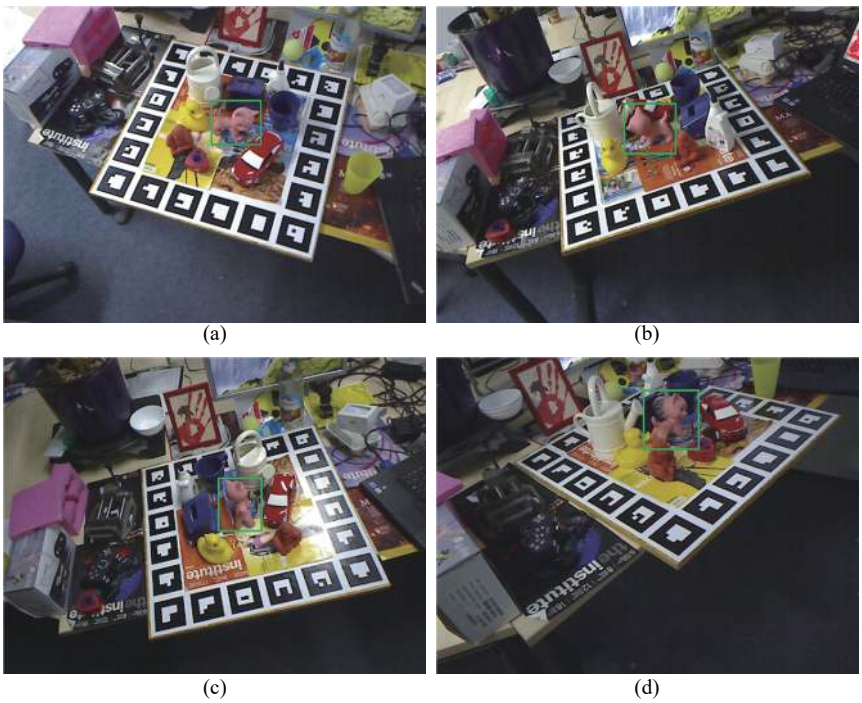


Fig. 6 Textureless target detection results based on improved LineMod template matching: (a)–(c) Textureless target detection results under different perspectives and scales and different lighting conditions; (d) Textureless target detection result during partial occlusion.

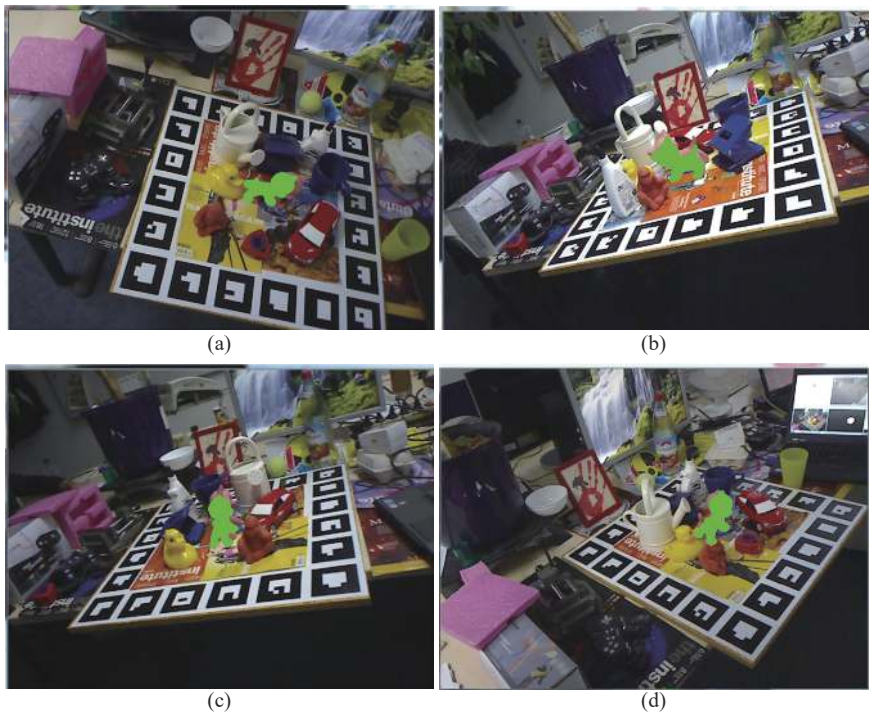


Fig. 7 Tracking registration results of the improved LineMod: (a)–(c) Registration results of the improved LineMod under different perspectives and scales; (d) Registration result of the improved LineMod under partial occlusion.

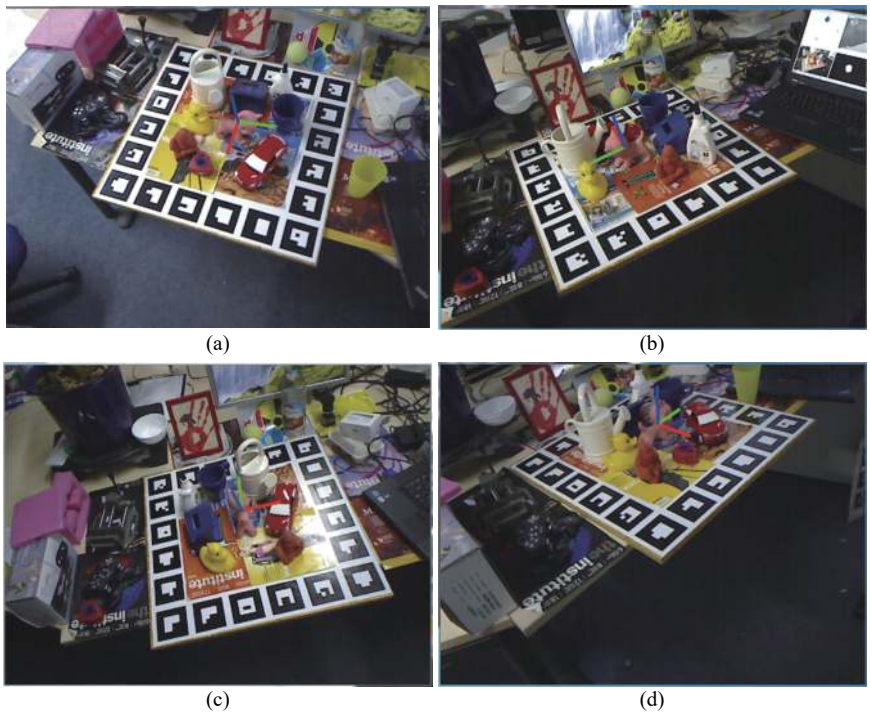


Fig. 8 Pose information obtained by the method in this article: (a)–(c) Pose information obtained by the method in this article under different perspectives and scales; (d) Pose information obtained by the method in this article under partial occlusion.

paper can reactivate the improved LineMod multi-modal template matching algorithm to retrieve the key frame and the initial pose. Therefore, it will not affect the tracking registration effect. In addition, Table 1 lists the time consumed by each key step in the process of using this method. Among them, the template matching process is marked as Process 1 and the tracking registration process is marked as Process 2. It takes 51.08 ms to obtain the initial pose using the improved multi-modal LineMod template matching algorithm, which is mainly

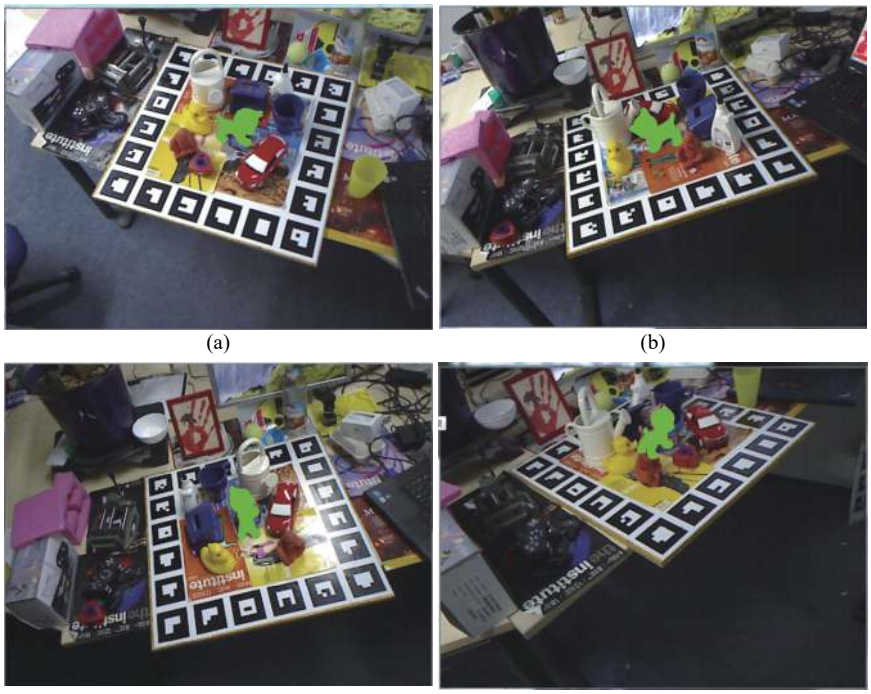


Fig. 9 Registration results of the method in this article: (a)–(c) Registration results of the method in this article under different perspectives and scales; (d) Registration result of the method in this article under partial occlusion.

Table 1 Average running time of each frame of this method

Process	Main steps	Time (ms)	Total time (ms)
1	Video stream input	1.02	51.08
	Feature extraction	6.47	
	Detection by LineMod	43.59	
2	Video stream input	1.02	29.82
	Point cloud generation and preprocessing	9.17	
	Improve ICP tracking registration	19.63	

due to the longer time required to match the input image with all reference images one by one. After the key frame and initial pose are obtained, the time required for tracking registration using the improved ICP algorithm is 29.82 ms per frame. In addition, the method in this paper is mainly in the tracking registration process based on the point cloud. The template matching process will be executed again only when the initial state or tracking registration fails. Therefore, this method can meet the real-time requirements of augmented reality.

6 Conclusions

In this paper, the methods of multi-modal template matching and point clouds are effectively combined, and a new markerless tracking registration method for augmented reality is proposed for textureless targets. First, the improved multi-modal LineMod template matching method is used to verify the similarity between the current perspective image and the reference images. Meanwhile, the key frame which is most similar to the current perspective image and the initial pose of the camera are obtained.

Then we use the tracking registration method based on the point cloud to calculate the precise pose of the camera in real time. In view of the shortcomings of the traditional ICP algorithm, Kd-tree is used to replace the part of finding the nearest point of the original ICP algorithm under the GPU to improve the efficiency of the algorithm. At the same time, the RANSAC algorithm is used to remove the error point pairs to improve the accuracy of the algorithm. When the camera moves too fast and the process of tracking registration fails, the method in this paper can reactivate the multi-modal LineMod template matching algorithm to retrieve the key frame and the initial pose without affecting the tracking registration effect. Finally, public data sets are used to verify the tracking registration effect of this method, and its real-time performance is analyzed. The results show that the proposed method has good real-time performance and robustness.

Acknowledgements

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

References

[1] P. X. Cao, W. X. Li, W. P. Ma. Tracking registration

method based on improved random ferns. *Computer Applications and Software*, vol. 36, no. 11, pp. 51–56, 2019. DOI: 10.3969/j.issn.1000-386x.2019.11.009. (in Chinese)

[2] Q. B. Tang, S. M. Hou. Research on tracking and registration based on improved ORB algorithm and pose estimation. *Application Research of Computers*, vol. 33, no. 12, pp. 3905–3908, 2016. DOI: 10.3969/j.issn.1001-3695.2016.12.086. (in Chinese)

[3] S. M. Hou, J. Han, Y. D. Zhang, Z. Q. Zhu. Survey of vision-based augmented reality 3D registration technology. *Journal of System Simulation*, vol. 31, no. 11, pp. 2206–2215, 2019. DOI: 10.16182/j.issn1004731x.joss.19-FZ0286. (in Chinese)

[4] J. Chen, Y. Sun. System algorithm based on FAST keypoints for markerless augmented reality applications. *Transactions of Beijing Institute of Technology*, vol. 35, no. 4, pp. 421–426, 2015. DOI: 10.15918/j.tbit1001-0645.2015.04.017. (in Chinese)

[5] Y. Dong, L. Ji, S. Wang, P. Gong, J. Yue, R. Shen, C. Chen, Y. Zhang. Accurate 6DOF Pose Tracking for Texture-Less Objects. *IEEE Transactions on Circuits and Systems for Video Technology*, published online, 2020. DOI: 10.1109/TCSVT.2020.3011737.

[6] P. X. Cao, W. X. Li, W. P. Ma. Tracking registration algorithm for augmented reality based on template tracking. *International Journal of Automation and Computing*, vol. 17, no. 2, pp. 257–266, 2020. DOI: 10.1007/s11633-019-1198-3.

[7] N. Payet, S. Todorovic. From contours to 3D object detection and pose estimation. In *Proceedings of International Conference on Computer Vision*, IEEE, Barcelona, Spain, pp. 983–990, 2011. DOI: 10.1109/ICCV.2011.6126342.

[8] L. H. Pan, F. Q. Tian, W. J. Ying, Q. J. Qian. Survey on direct-method visual simultaneous localization and mapping. *Application Research of Computers*, vol. 36, no. 4, pp. 961–966, 2019. DOI: 10.19734/j.issn.1001-3695.2018.01.0123. (in Chinese)

[9] L. Chen, W. Tang, N. W. John, T. R. Wan, J. J. Zhang. SLAM-based dense surface reconstruction in monocular Minimally Invasive Surgery and its application to Augmented Reality. *Computer Methods and Programs in Biomedicine*, vol. 158, pp. 135–146, 2018. DOI: 10.1016/j.cmpb.2018.02.006.

[10] Y. Wang, S. S. Zhang, W. P. He, X. L. Bai. Model-based marker-less 3D tracking approach for augmented reality. *Journal of Shanghai Jiaotong University*, vol. 52, no. 1, pp. 83–89, 2018. DOI: 10.16183/j.cnki.jsjtu.2018.01.013. (in Chinese)

[11] Y. Park, V. Lepetit, W. Woo. Texture-less object tracking with online training using an RGB-D camera. In *Proceedings of the 10th IEEE International Symposium on Mixed and Augmented Reality*, IEEE, Basel, Switzerland, pp. 121–126, 2011. DOI: 10.1109/ISMAR.2011.6092377.

[12] Y. Wang, S. S. Zhang, X. L. Bai. A 3D tracking and registration method based on point cloud and visual features for augmented reality aided assembly system. *Journal of Northwestern Polytechnical University*, vol. 37, no. 1, pp. 143–151, 2019. DOI: 10.3969/j.issn.1000-2758.2019.01.021. (in Chinese)

[13] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, V. Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *Proceedings of International Conference on Computer Vision*, IEEE, Barcelona, Spain, pp. 858–865, 2011. DOI: 10.1109/ICCV.2011.6126326.

[14] C. Tomasi, R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the 6th International Conference on Computer Vision*, IEEE, Bombay, India, pp. 839–846, 1998. DOI: 10.1109/ICCV.1998.710815.

[15] Y. Wang, S. S. Zhang, S. Yang, W. P. He, X. L. Bai, Y. F. Zeng. A LINE-MOD-based markerless tracking approachfor AR applications. *The International Journal of Advanced Manufacturing Technology*, vol. 89, no. 5–8, pp. 1699–1707, 2017. DOI: 10.1007/s00170-016-9180-5.

[16] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, V. Lepetit. Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 876–888, 2012. DOI: 10.1109/TPAMI.2011.206.

[17] T. Ji, F. D. Li, Y. Q. Yang, T. P. Zhang. Monocular vision pose measurement based on improved ICP algorithm and CAD model. *Journal of Yangzhou University* (*Natural Science Edition*), vol. 22, no. 1, pp. 50–54, 61, 2019. DOI: 10.19411/j.1007-824x.2019.01.011. (in Chinese)

[18] K. S. Wang, X. Li, H. W. Lei, X. R. Zhang. An ICP algorithm based on block path closest point search. *Journal of Physics*: *Conference Series*, vol. 887, Article number 012063, 2017. DOI: 10.1088/1742-6596/887/1/012063.

[19] W. Guan, W. T. Li, Y. Ren. Point cloud registration based on improved ICP algorithm. In *Proceedings of Chinese Control and Decision Conference*, IEEE, Shenyang, China, pp. 1461–1465, 2018. DOI: 10.1109/CCDC.2018.8407357.

[20] H. Yu, H. J. Du, Y. H. Cai. Object point cloud modeling method based on improved SIFT-ICP algorithm. *High Technology Letters*, vol. 29, no. 344, pp. 24–31, 2019.

[21] J. Liu, J. W. Zhu, J. L. Yang, X. L. Meng, H. Zhang. Three-dimensional point cloud registration based on ICP algorithm employing K-D tree optimization. In *Proceedings of the 8th International Conference on Digital Image Processing*, Chengdu, China, Article number 100334D, 2016. DOI: 10.1117/12.2248362.

[22] L. J. Hu, S. Nooshabadi. Massive parallelization of approximate nearest neighbor search on Kd-tree for high-dimensional image descriptor matching. *Journal of Visual Communication and Image Representation*, vol. 44, pp. 106–115, 2017. DOI: 10.1016/j.jvcir.2017.01.013.

[23] M. Zhang. Study of Kd-tree based GPU ray tracing, Master dissertation, Chang′an University, China, 2014. DOI: 10.7666/d.D558268. (in Chinese)

[24] M. D. Li, S. P. Jiang, H. P. Wang. A RANSAC-based stable plane fitting method of point clouds. *Science of Surveying and Mapping*, vol. 40, no. 1, pp. 102–106, 2015. DOI: 10.16251/j.cnki.1009-2307.2015.01.022. (in Chinese)

[25] E. Arias-Castro, J. Wang. RANSAC algorithms for subspace recovery and subspace clustering. [Online], Available: https://arxiv.org/abs/1711.11220, 2017.

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 Technology (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