

Application of line inspection algorithm for UAV remote sensing system in electricity maintenance

SONG WANG¹, FANGCHEN DONG², LIANGLIANG
BU², XIN ZHANG³

Abstract. Based on the traditional Fast Retina Keypoint(FREAK) feature description algorithm, this paper proposed a Gravity-FREAK feature description algorithm based on Micro-electromechanical Systems (MEMS) sensor to overcome the limited computing performance and memory resources of mobile devices and further improve the reality interaction experience of clients through digital information added to the real world by reality technology. The algorithm takes the gravity projection vector corresponding to the future point as its feature orientation, which saved the time of calculating the neighborhood gray gradient of each feature point, reduced the cost of calculation and improved the accuracy of feature extraction. In the case of registration method of matching and tracking natural features, the improved adaptive and generic corner detection based on the accelerated segment Test (AGAST) matching purification algorithm was used to eliminate abnormal matches, and Gravity Kaneda-Lucas Tracking (KLT) algorithm based on MEMS sensor can be used for the tracking registration of the targets and robustness improvement of tracking registration algorithm under mobile environment.

Key words. Mobile sensor, Kaneda-lucas tracking, Augmented reality, Virtual registration, Fast retina keypoint.

1. Introduction

With the rapid development of image processing and artificial intelligence, the conception can be realized through the combing use of different technologies and the augmented reality technology which focuses on virtual-real fusion emerged [1]. Different from the virtual reality technologies that focus on introducing users to virtual

¹School of Aeronautic Science and Engineering, BeiHang University, Beijing, 100191, China

²Robotics Institute, BeiHang University, Beijing, 100191, China

³Shenzhen Power Supply Co., Ltd., Shenzhen 518048, China

3D scenes, the augmented reality technology emphasizes how to accurately integrate the virtual information generalized by computer into the real-world environment so that to realize the simultaneous presentation of virtual information and the real environment for the supplementation and enhancement of the real environment. The relationship between the two parts is show as Figure 1:

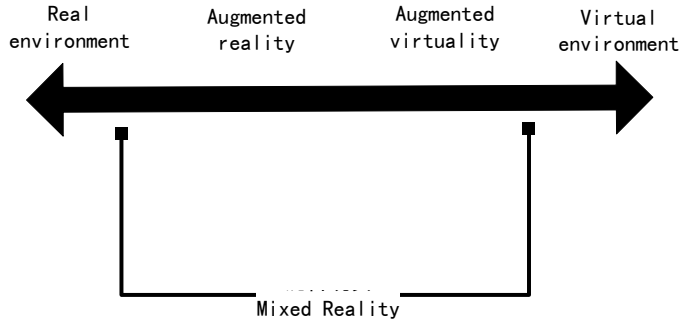


Fig. 1. Mixed reality

Generally, the augmented reality system is consisted of three parts: virtual-real fusion, real-time interaction and 3D registration [2]. Among the three parts, 3D registration, the accurate matching between virtual and real environments, is the key restraining factor of wider application of augmented reality technology. Most of traditional 3D registration methods were designed and proposed on the basis of PC [3]. They cannot be applied to mobile augmented reality systems directly as most of the mainstream mobile devices are not equipped with floating point processor (FPP), and the CPU speed and memory capacity are not able to support the devices efficiently to conduct feature extraction and position calculation of the target. Hence, it becomes an urgent matter to search a mobile 3D registration algorithm with better performance and lower resource occupation to popularize mobile augmented reality.

2. Related work

As the product of the constant development of virtual reality technology, the appearance of augmented reality can be traced back to the HMD (Head Mounted Display) invented by an American in 1965 [4]. Through the device, the user can visualize the superposition of real environment and 3D image. Until the early 1990s, the concept of augmented reality was first proposed by Caudelland Mizell[5] , scientists from Boeing Co. After that, the size of portable device became smaller and smaller, while the computing performance became stronger and stronger, which makes it possible to conduct image rendering and superposition on mobile devices. In 1997, Feiner et al. [6] designed the first prototype of mobile augmented reality system. The system can add 3D travel guide information onto the real built environment. By the end of 1990s, augmented reality became an independent and significant research field which attracted more and more researchers. Many AR related international conferences also emerged, such as IWSAR (International Workshop and Sympo-

sium on Augmented Reality), ISMR (International Symposium on Mixed Reality), DARE (Designing Augmented Reality Environments workshop), etc. Among all the research directions, the research on AR tracking registration technology is always the hotspot, which is also the key step in the application of AR. According to the registration method, AR system can be divided into sensor oriented system and machine vision oriented system.

2.1. Sensor oriented tracking registration

Sensor oriented tracking method has long ago been applied to AR registration field by researchers, including mechanical tracking registration, electromagnetic tracking registration, ultrasonic tracking registration, GPS tracking registration and inertial tracking registration, etc. The method relies on the related sensor function of the hardware device. With the accurate real-time data provided by the sensor, the method can obtain the position and direction information of the tracking target. The outdoor AR system designed by Feiner [6] et al. used sensors as GPS and angle instrument for tracking registration. However, the method has high requirements of hardware and environment. Many sensor oriented tracking registration methods are still in the experimental stage, which cannot be promoted to ordinary users.

2.2. Computer vision oriented tracking registration

Compared with sensor oriented tracking registration, the computer vision oriented method has a wider practicability. Based on different identification method, the computer vision oriented method can be divided into artificial identification oriented registration method and natural feature oriented registration method.

The artificial identification oriented registration method needs to install landmark with obvious identification feature in natural environment, which can be identified in the video image through matching algorithm. With regular geometrical features and known position in the natural environment coordinate system, the landmark can be captured easily and accurately by computer. Then the camera can calibrate the algorithm and obtain the accurate parameters of the projection matrix. In this way, the virtual 3D image can be superimposed on the position of the real environment landmark [7–11]. Without any requirement of rebuilding natural environment with small computational amount, this method is proper for the mobile devices with limited computing and main memory resources. However, due to the unsatisfactory performance of matching algorithm in light and shadow robustness, it is hard to be applied in real application environment on a large scale.

The natural feature oriented registration method uses natural feature to conduct 3D registration, mainly using vision positioning technology and image extraction algorithm to realize an accurate positioning and tracking of the target for 3D registration. Traditional vision feature tracking methods include: EKF (Extended Kalman Filter) [12] method and systems based on particle filter [13] and unscented Kalman filter [14]. On account of SLAM, Georg Klein et al. proposed the augmented reality system based on parallel tracking and mapping (PTAM) [15]. The method was

applied in mobile devices by Georg Klein et al. [16] in 2009.



Fig. 2. Augmented reality system based on PTAM

There are many methods of feature point extraction and description. The mostly common used methods include FAST, SIFT, and SURF, etc. In 2010, Wagner improved SIFT and Ferns [17,18] by replacing the DOG feature detecting algorithm in SIFT with FAST, reduced the number of feature vector dimensions and realized the 6-DoF real-time mobile augmented reality system on smartphones.

3. Gravity-FREAK feature extraction and description

On account of the performance problems on mobile devices of feature description algorithms as SIFT, SURF and ORB, this paper used AGAST algorithm to extract feature points and proposed an improved FREAK feature description algorithm combining with the inertial sensor commonly equipped on mobile devices[19,20,21, 22].

3.1. AGAST

As a feature point extraction algorithm proposed by Mair et al. [19,20], AGAST added scale information onto the feature point on the basis of FAST algorithm, which makes the algorithm more accurate in the positioning performance. In order to obtain high quality feature points with scale information, AGAST algorithm not only conducted the extremum search in its image space, but also used FAST algorithm to detect the extremum at each scale space.

Before the extremum detecting, n octaves c_i and n intra-octaves d_i should be constructed to compose there solution pyramid needed for the research. Normally $i = \{0, 1, \dots, n - 1\}$, and $n = 4$. Octaves c_j can be obtained through semi-sampling of c_{j-1} , and c_0 is the original image. Interlayer d_i is located in the middle of c_i and c_{i+1} . The first interlayer was generated through the down sampling with the parameter of 1.5 of original image, and the rest interlayers were obtained through

the semi-sampling of the previous interlayer as shown in Formula 1:

$$\begin{cases} t(c_i) = 2^i, \\ t(d_i) = 2^i \cdot 1.5. \end{cases} \quad (1)$$

Where t refers to the image dimension.

In the extremum test, FAST 9-16 template is often used first to extract the potential interest points of each octave and the middle octave. As shown in Figure 3, the white dash line is a circle taking the key point to be tested p as the center and 3 pixels are the radius with totally 16 pixels. According to the requirement of the test standard, only if the brightness of at least 9 pixels of all the 16 pixels is higher or lower than that of the key point and exceeds the threshold value t , the key point can be determined as the potential feature point as shown in Formula 2:

$$\begin{cases} V = \max(\sum_{x \in S_{bright}} |I_{p \rightarrow x} - I_p| - t, \sum_{x \in S_{dark}} |I_p - I_{p \rightarrow x}| - t), \\ S_{bright} = \{x | I_{p \rightarrow x} \geq I_p + t\}, \\ S_{dark} = \{x | I_{p \rightarrow x} \leq I_p - t\}. \end{cases} \quad (2)$$

Where V is the FAST 9-16 score of the key point.

Then the obtained potential interest point needs to be conducted with non-maximum suppression. The real feature point needs to satisfy the following two conditions: (1) In the 3×3 area of the octave with the point as the center, its FAST score should be the highest among the 9 points; (2) In the corresponding upper and lower 3×3 areas, its FAST score should be the highest. The extremum test on the octave c_0 is an exception: FAST 5-8 template is used on c_0 for calculation as the FAST score on the virtual middle octave $d - 1$ needs to be captured.

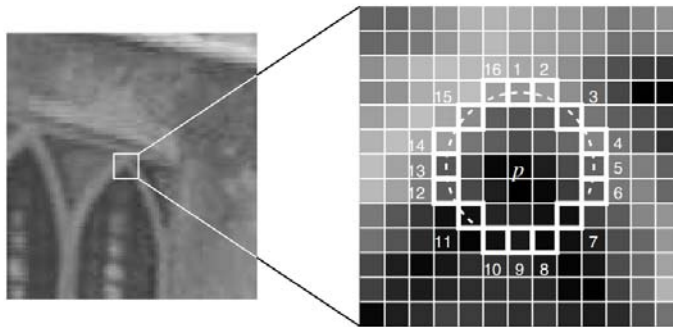


Fig. 3. FAST 9-16 working principle diagram

Considering the successive image conspicuousness in the dimension space, obtained feature points can be conducted with elaboration of the successive sub-pixel level dimension. In order to simplify the elaboration process, a two-dimensional quadratic function was used to proceed the least square fittings on the upper, middle and lower areas, and set the area size as

to reduce repeated sampling; Then use one-dimensional parabolic for the fitting of the elaborated score and obtain the final score and dimension of the feature point; At last, interpolate the position of the feature point to obtain more accurate interest point.

3.2. Traditional FREAK feature description

FREAK algorithm is a binary feature description algorithm proposed by Alexandre Alahi et al. in 2012. The core idea of FREAK algorithm is to use the sampling template of imitated retinal structure to conduct the descriptor structure of feature point. In the field of neurology, retina can be divided into 4 areas according to the distribution density of the ganglion cells on retina: foveal, fovea, parafoveal and perifoveal, as shown in Figure 4 (left). Each area plays an important role in the process of visual detection and recognition. For example, the foveal area in the density center captures high-resolution fine information and the perifoveal area with the most sparse ganglion cell distribution perceives the low-resolution vague information. To some extent, this division of labor can be considered as a rational allocation of human body resources. By imitating the physical structure of retina, FREAK sampling template uses different sizes of Gaussian kernel for the smoothing of sampling point. As shown in Figure 4 (right), each circle represents a sampling point, and the Gaussian smoothing kernel size of the point is in proportion to the diameter of the circle. The closer to the feature center, the smaller the Gaussian kernel will be, and the larger the sampling density will be. In addition, Alexandre Alahi et al. discovered that allowing the crossing and overlapping of the sampling area can help capturing and describing more information and improving the performance of descriptor. Hence, redundant information was also applied to the sampling of FREAK retina template.

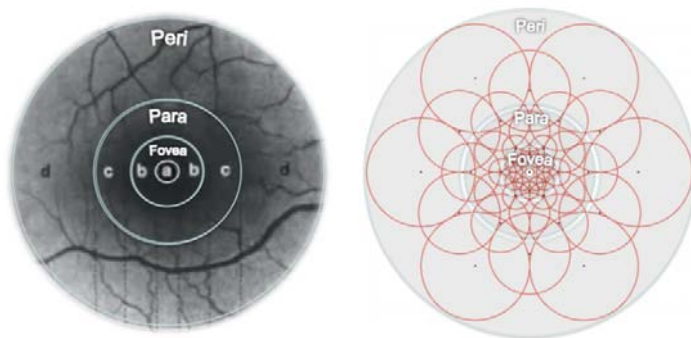


Fig. 4. Diagram of human retina structure (left) and FREAK sampling template (right)

Similar to the descriptor of image feature as ORB, FREAK constructs the binary descriptor F by calculating the difference of Gaussian convolution (DoG) at a

sampling point pair as shown in Formula 3:

$$F = \sum_{0 \leq a \leq N} 2^a T(P_a), \quad (3)$$

Where P_a refers to a sampling point pair, N stands for the bit length of descriptor, and $T(P_a)$ satisfies Formula 4:

$$T(P_a) = \begin{cases} 1 & \text{if } (I(P_a^{r1}) - I(P_a^{r2}) > 0) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Where $I(P_a^{r1})$ refers to the brightness of the first sampling point in P_a .

General speaking, if 43 sampling areas were selected in the neighboring area of the feature point as shown in Figure 4 (right), a single feature point would generate almost one thousand sampling point pairs. However, the factor is that only some of the point pairs can effectively describe the image information. Hence, FREAK designed an algorithm based on ORB descriptor to select the best sampling point pair from the training set. The concrete steps are shown follows:

(1) Based on the information of feature points, FREAK would first construct the matrix D . Each row in the matrix represents the maximum bit description of a feature point, referring the bit string generated by all the points to be constructed by the 43 sampling areas;

(2) Calculate the mean value of each column. To obtain a better discriminant effect, it is better to choose feature points with larger differences meaning the larger the variance between each column in the matrix the better. While in 0-1 distribution, the variance reaches the largest when the mean value is 0.5;

(3) For each column of the matrix D , the mean value should be arranged from largest to smallest;

(4) Choose the column with the largest mean value. After loop iteration, the Column with the smallest correlation with the selected column will be retained.

The 512 sampling point pairs selected by above algorithm already embraced the idea of arrangement from rough to fine of Gaussian difference. As shown in Figure 5, if the 512 sampling point pairs were divided into 4 groups with 128 pairs a group, the 4 groups can happen to be corresponding to the 4 areas of human retina. The first group of sampling point pairs is mostly located at lateral side corresponding to the perifoveal area of retina, while the last group of sampling point pairs is mostly located at center area corresponding to the foveal area of retina. Hence, in the target matching process, the approximate location of the target can be first estimated by taking advantage of the perifoveal area and then determined by using more sophisticate area, which accorded with the method of human searching for targets. Specifically, the first 16 bytes of the FREAK descriptor can be used for the coarse-grained comparison. If the matching distance is less than a threshold value, keep using the rest bytes for more sophisticated comparison. The cascade method greatly accelerated the matching speed of descriptor as 90% of the candidates have been abandoned in the previous 16 bytes screening.

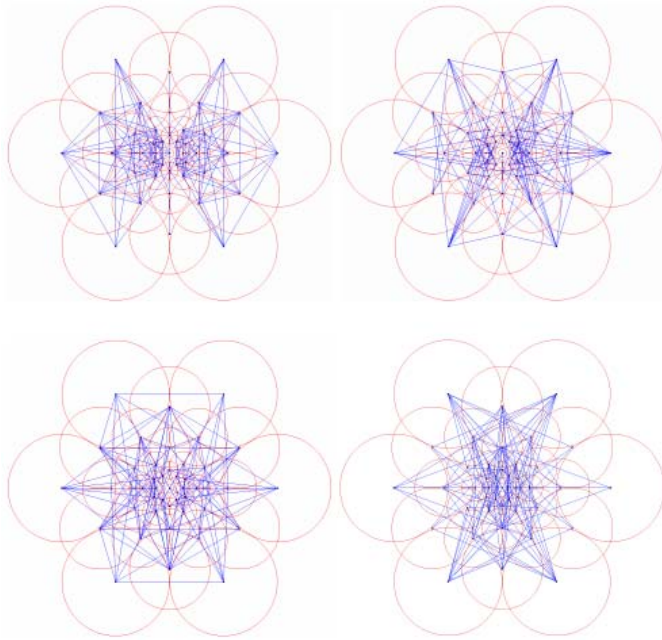


Fig. 5. Sampling point pair from coarse to fine

In addition, to estimate the direction of feature point, FREAK used similar descriptor method as BRISK to calculate the local gradient. The difference is the FREAK uses centrosymmetric sampling area as shown in Figure 6:

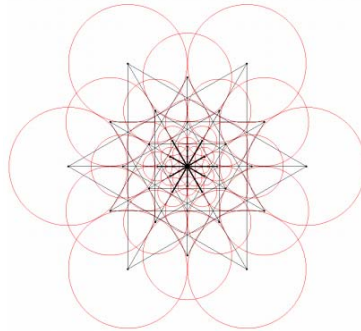


Fig. 6. Sampling point pair of calculating feature direction

G is assumed as the set consisted by all the sampling point pairs calculating the local gradient. The calculating formula of feature point direction is shown as Formula 5:

$$O = \frac{1}{M} \sum_{P_o \in G} (I(P_o^{r_1}) - I(P_o^{r_2})) \frac{P_o^{r_1} - P_o^{r_2}}{\|P_o^{r_1} - P_o^{r_2}\|}. \quad (5)$$

Where M refers to the element number of set G ; $P_o^{r_i}$ represents the two-dimensional vector of sampling center.

Comparing with the requirement of more than 100 sampling point pairs of BRISK, the calculation of FREAK feature direction only needs 45 point pairs. Besides, FREAK retina sampling template has a larger sampling area in perifoveal area, which can better calculate the deviation from redundant direction and has lower memory occupation.

3.3. FREAK description based on gravity

At present, mainstream mobile devices are all equipped with gravity sensor. Combined with computer vision and MEMS sensor, the direction gesture information obtained by AR system is more accurate and stable. As early as 2000, S. You et al. [23] already used the mixed method of inertial sensor and computer vision to conduct target tracking. In order to improve the feature matching of panoramic picture, J.C. Bazin et al. used gyroscope to measure the direction and posture variation when the camera shooting two adjacent frames. The second image will be affine transformed to align with the first image. Then the SIFT feature point of the two images will be extracted for the matching.

(1) Gravity model of mobile device

In order to obtain the current gravity direction of a point in the image, gravity sensor will be used to test and record the position and posture of the camera when shooting. Take Apple iOS system for example, the Core Motion Framework frame already encapsulated the related operations of the acceleration sensor into the UI Acceleration category and eliminated the impacts of user's shaking on current posture of the device by combining with the data from the gyroscope. The instance object of the category can be created to directly read x, y, z attributes to estimate the current position and posture. As shown in Figure 7, x, y, z represents the instantaneous acceleration component of the mobile device in the three directions respectively.

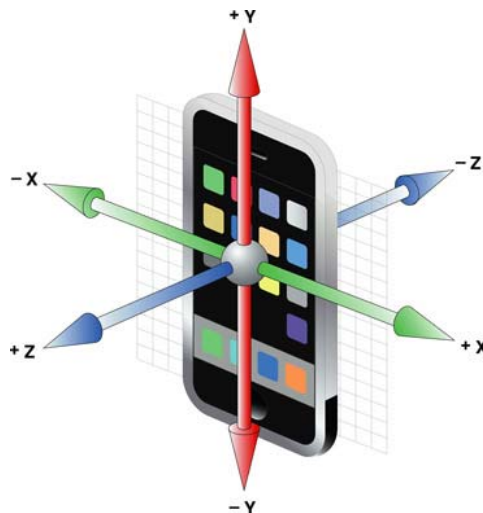


Fig. 7. Diagram of the three acceleration component direction of smartphone

Acceleration component is a double type data type, with the value range from -1 to +1 and gravity unit of g. For example, if the value is 1.0, the acceleration of the device along the direction equals to a gravity unit. When an iPhone was placed still on a horizontal surface with the screen upward, the current output acceleration value approximates as $(0, 0, -1)^T$.

As the gravity effect at each point on the imaging plane is uneven, the 2D projection of the 3D space gravity on the image plane needs to be calculated to accurately mark the gravity direction of the current image feature point. Given the normalized vector of 3D space gravity $g = [g_x, g_y, g_z]^T$, g satisfying the condition of $\|g\| = 1$, and given the camera internal parameter K (3×3 matrix), then 2D projection vector $d = [d_u, d_v, 0]^T$ of the actual environmental gravity at the pixel point $p = [u, v, 1]^T$ can be calculated through Formula 6:

$$d = p' - p, \quad (6)$$

Where $p' = [u', v', 1]^T$, can be obtained by Formula 7:

$$[(1 + g_z)u', (1 + g_z)v', (1 + g_z)]^T = p + Kg. \quad (7)$$

As the length of vector d and g can be any value, then:

$$d \propto [g_z(p_u - u) + f_u g_x, g_z(p_v - v) + f_v g_y, 0]^T, \quad (8)$$

Where $[p_u, p_v]^T$ refers to the main coordinate, f_u and f_v respectively represents the focus length of the camera in horizontal and vertical direction. The direction angle θ of the feature point can be expressed finally as:

$$\theta = \arctan\left(\frac{d_v}{d_u}\right). \quad (9)$$

(2) Gravity improved FREAK description

The direction computing method of traditional FREAK descriptor is similar to BRISK descriptor. Both of them need to conduct the accumulated operation on the local gradient of the point pair selected at feature neighborhood. The gravity sensor on the mobile device not only can improve the calculating efficiency of feature extraction but also can solve the mismatching of feature points with physical similarities to some extent to improve the accuracy of feature matching.

Take the condition described in the Figure 8 as example, if the angular point in the image is described by traditional FREAK algorithm, it would rotate the feature point to current dominant gradient orientation during the normalization process. Although the feature points of the four window corners respectively represented four different physical points in the reality scene, their descriptors are basically the same and the feature points are unable to be distinguished according to the descriptors. However, the gravity improved FREAK description algorithm has a completely different effect. For the four feature points in Figure 8, their feature directions are all straight down gravity directions from the window. During the normalization, each feature point will be rotated to the gravity direction as shown

in Figure 8 (right). Obviously they have strong discriminative characteristics. Hence, the generated feature description vectors will be also different.

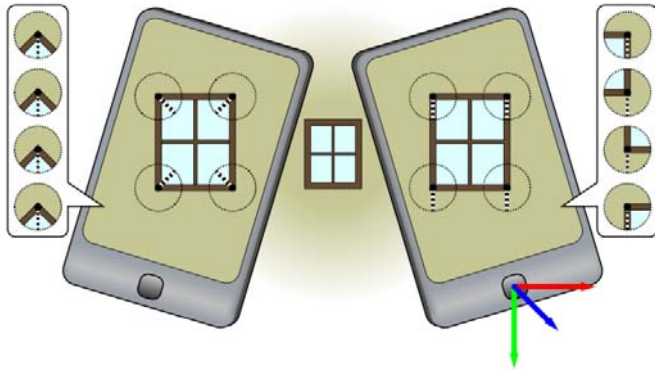


Fig. 8. Feature point aligned in gray gradient direction (left) and gravity direction (right)

The theory of taking gravity direction as the feature direction can be understood as: when users are experiencing mobile augmented reality system, they will proceed with the augmentation with smartphone aligned with nearly vertical scene objective. First, phone camera will shoot the first frame of the scene image and the device sensor will record the present gravity and its direction $G1$. That is the phone describes the feature point in the first frame of picture along with the gravity components of x, y, z axles in Figure 7 with its projection of $g1$ on the image; due to the rotation of phone, the projection of $g2$ and $g1$ on the image of the gravity vector $G2$ pointed in different directions when shooting the second frame of image. So the second frame of image can be rotated until $g2$ and $g1$ have the same direction, which means the gesture and angle of the phone shooting the first frame of image was restored so that to maintain the rotation invariance of the feature point as shown in Figure 9:

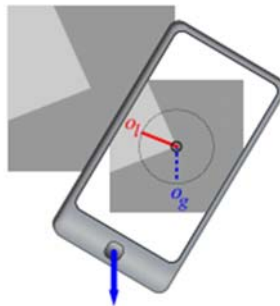


Fig. 9. FREAK feature using gravity as feature direction

Based on the above theory analysis, traditional FREAK description algorithm can be improved by using gravity angle instead of traditional feature point angle to calculate descriptor.

Compared with original FREAK algorithm, improved FREAK algorithm can accomplish the assignment of all the feature point directions in the same image only by using the build-in sensing data interface of mobile platform system once, which greatly improved the description efficiency of the feature point. In addition, due to the accuracy requirement of the sensor itself, the feature direction information obtained by gravity sensor is more accurate than that obtained by computer vision method, which can help feature point accomplish more sophisticated matching. Meanwhile, the experiment showed that traditional FREAK algorithm can preferably maintain the rotation invariance of feature point when the mobile device rotating around Z axle in Figure 7. The matching accuracy will be greatly decreased when the device rotates around other axles. While improved FREAK algorithm can better adapt the multi-directional rotating shooting and is more flexible.

4. Mobile AR image matching and tracking registration

In the mobile augmented reality system, the obtaining of the feature point successfully matched in referential image and target image showed that homography relation can be created between the two images and the registration of the augmented information can be accomplished. However, it is hard to avoid the mismatching in the real situation. Mismatching feature would affect the accuracy of 3D registration tremendously. Meanwhile, if every frame of image in the scene video is to be conducted with feature extraction and matching, it will consume the limited system resource on the mobile platform and affect the real-time characteristics of the mobile augmented reality system. Hence, to deal with the mismatching problem, RANSAC algorithm is used to eliminate exterior points and improve the final matching accuracy; meanwhile, the inherent continuity of video image will be fully excavated and the tracking registration of the video will be conducted through the improved KL algorithm based on inertial sensor to assure the fluency of mobile augmented reality system.

4.1. Matching purification

In the image field, it is common to face the problem of estimating the image transformation matrix with matching feature points. How to obtain stable model parameter is the key point of conducting the matrix estimation. However, in the real feature matching, it is hard to assure the accuracy of all the matching data. Hence, it is of high priority to solve the problem of estimating correct model parameters by using the samples containing abnormal data.

(1) Random sample consensus matching algorithm

First proposed by Fischler et al., random sample consensus matching algorithm (RANSAC) is an optimized algorithm used to estimate stable model parameters [26]. Based on RANSAC algorithm, Chum et al. [24] proposed local optimization in 2003 to improve RANSAC algorithm and increase efficiency through accelerating the algorithm convergence. In 2004, Matas and Churn et al. [25] proposed the pre-testing model of $T_{d,d}$, the speed of which has been improved 50% compared

with typical RANSAC algorithm. On this basis, Chen Fuxing and Wang Runsheng [27] raised the pre-testing model of $T_{c,d}$ in 2005, which is more generalized. After conducted a number of experiments, Capel [28] proved in the articles that the adding of local optimization and pre-testing model of RANSAC algorithm can also maintain the previous accuracy and reliability. Considering the limited resource of the mobile platform, this paper chose to use the PERANSAC algorithm proposed by Chen Fuxing et al. to improve the overall efficiency [27].

(2) Pre-testing rapid random sample consensus matching algorithm

Based on RANSAC algorithm, PERANSAC algorithm increased a pre-testing model. First, part of sample data was chosen to evaluate current model parameters. Only those models that have passed the pre-testing are supposed to be used for the evaluation of all the rest samples. Besides, in order to maintain the algorithm reliability, PERANSAC increased the sampling data. The specific algorithm procedure is shown as follows:

1) Given the testing data number n , sample abnormal rate ε and minimum pre-testing pass rate, calculate n_f and P_f according to Formula 10; C_n^i refers to the combination of i samples selected, P_f represents the test pass rate, n_f stands for the domain point number needed to pass the test;

$$1 - \sum_{i=0}^{n_f-1} C_n^i \varepsilon^{(n-i)} (1 - \varepsilon)^i = P_f. \quad (10)$$

2) According to Formula 11, select the sample number M needed for the model parameter evaluation, m represents the minimum sample number needed for the model estimation;

$$1 - (1 - (1 - \varepsilon)^m P_f)^M = P. \quad (11)$$

3) Select a group of random samples and estimate the model parameters that the samples satisfy;

4) Select n random data, test the model parameters obtained from (3), and loop perform (3) (4) until all the parameters were tested. If all the parameters failed in the test, restart from (1). If the obtained model parameters still can't pass the pre-test after many times of repeated operation, increase the sample abnormal rate ε and estimate again;

5) Estimate the succeeded models with all the sample data and write down the corresponding number of interior points;

6) Select the best model parameters under the standard of interior points number and error variance;

7) Based on the interior points included in the selected model parameters, estimate the model parameter that is the final results of the algorithm.

4.2. Gravity-KLT improved tracking registration algorithm based on MEMS sensor

(1) Kanade-Lucas Tracking algorithm

KL(Kanade-Lucas)Tracking algorithm uses window image registration technol-

ogy for the tracking of feature. First proposed by Lucas and et al. [29] and improved by Baker and Matthews [30], this algorithm has been a typical algorithm in the target tracking field. The early Lucas-Kanade algorithm aimed to realize the registration of template image $T(\mathbf{x})$ and input image $I(\mathbf{x})$, where $\mathbf{x} = (x, y)^T$ is the 2D column vector representing the pixel coordinates. Specific to optical flow computation, template $T(\mathbf{x})$ represents the subdomain of the image at the time of t (e.g. 5×5 image window); $I(x)$ refers to the image at the time of $t + 1$. If $W(\mathbf{x}, \mathbf{p})$ stands for the parameterized set of all reasonable deformation of the image, where $\mathbf{p} = (p_1, \dots, p_n)^T$ refers to the parameter vector. Then the parameterized set can map the pixel x in the coordinate system of template image T onto the $W(\mathbf{x}, \mathbf{p})$ position in the coordinate system of the image I . Taking the optical tracking method for example, $W(\mathbf{x}, \mathbf{p})$ can be expressed as:

$$W(\mathbf{x}, \mathbf{p}) = \begin{pmatrix} x + p_1 \\ y + p_2 \end{pmatrix}. \quad (12)$$

The vector parameter $\mathbf{p} = (p_1, p_2)^T$ refers to the optical flow vector. If the target to be tracked is an image block which can move freely in the 3D space, then $W(x, p)$ can be assumed to represent affine transformation, that is:

$$\begin{aligned} W(\mathbf{x}, \mathbf{p}) &= \begin{pmatrix} (1 + p_1) \cdot x + p_3 \cdot y + p_5 \\ p_2 \cdot x + (1 + p_4) \cdot y + p_6 \end{pmatrix} \\ &= \begin{pmatrix} 1 + p_1 & p_3 & p_5 \\ p_2 & 1 + p_4 & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \end{aligned} \quad (13)$$

Where $\mathbf{p} = (p_1, p_2, p_3, p_4, p_5, p_6)^T$ denotes Vector parameter. Obviously, under different tracking environment, $W(\mathbf{x}, \mathbf{p})$ can be any complicated target tracking template with variable dimension of the parameter vector.

Essentially, the ultimate goal of Lucas-Kanade algorithm is to reduce the mean-square error between the template image T and the transformed image I to the most:

$$\sum_x [I(W(\mathbf{x}; \mathbf{p})) - T(x)]^2. \quad (14)$$

As shown in Formula 14, tracking problem was transformed to a nonlinear optimization problem so that to calculate the vector \mathbf{p} to satisfy the target of the algorithm. In order to obtain the best solution of Formula 14, Lucas-Kanade algorithm assumes the estimation of \mathbf{p} is known, and calculate $\Delta \mathbf{p}$ and conduct the iterative updates of \mathbf{p} until its convergence, as shown in Formula 15 and Formula 16:

$$\sum_x [I(W(\mathbf{x}; \mathbf{p} + \Delta \mathbf{p})) - T(x)]^2. \quad (15)$$

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta \mathbf{p}. \quad (16)$$

Whether \mathbf{p} is convergent or not is usually decided when the template of the vector

$\Delta \mathbf{p}$ is lower than a threshold value.

In the $\Delta \mathbf{p}$ computing process, in order to linearize Formulas 15, $I(W(\mathbf{x}; \mathbf{p} + \Delta \mathbf{p}))$ will be conducted with Grade 1 Taylor Series unfolding and Formulas 17 can be obtained:

$$\sum_x [I(W(\mathbf{x}; \mathbf{p})) + \nabla I \frac{\partial W}{\partial \mathbf{p}} \Delta \mathbf{p} - T(x)]^2. \quad (17)$$

Where $\nabla I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$ refers to the grade value of image I at $W(\mathbf{x}; \mathbf{p})$. Normally, ∇I will be calculated first at the coordinates of image I and then mapped to the $W(\mathbf{x}; \mathbf{p})$ position of template image T . $\frac{\partial W}{\partial \mathbf{p}}$ is the Jacobian determinant of $W(\mathbf{x}; \mathbf{p})$. If $W(\mathbf{x}; \mathbf{p}) = (W_x(\mathbf{x}; \mathbf{p}), W_y(\mathbf{x}; \mathbf{p}))^T$, then:

$$\frac{\partial W}{\partial \mathbf{p}} = \begin{pmatrix} \frac{\partial W_x}{\partial p_1}, \frac{\partial W_x}{\partial p_2} \cdots \frac{\partial W_x}{\partial p_n} \\ \frac{\partial W_y}{\partial p_1}, \frac{\partial W_y}{\partial p_2} \cdots \frac{\partial W_y}{\partial p_n} \end{pmatrix}. \quad (18)$$

Traditionally, the result of column vector calculating partial derivatives can be described by vector. The chain rule can change to matrix multiplication for calculation and the Jacobian determinants of Formula 13 can be:

$$\frac{\partial W}{\partial \mathbf{p}} = \begin{pmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{pmatrix}. \quad (19)$$

If the partial derivative related to $\Delta \mathbf{p}$ is calculated on the basis of Formula 17:

$$2 \sum_x [\nabla I \frac{\partial W}{\partial \mathbf{p}}]^T [I(W(x; \mathbf{p})) + \nabla I \frac{\partial W}{\partial \mathbf{p}} \Delta \mathbf{p} - T(x)], \quad (20)$$

$\nabla I \frac{\partial W}{\partial \mathbf{p}}$ is regarded as the steepest gradient descent image. If Formula 20 = 0, then:

$$\Delta \mathbf{p} = H^{-1} \sum_x [\nabla I \frac{\partial W}{\partial \mathbf{p}}]^T [T(x) - I(W(x; \mathbf{p}))] \quad (21)$$

H refers to the Hessian matrix of $n \times n$:

$$H = \sum_x [\nabla I \frac{\partial W}{\partial \mathbf{p}}]^T [\nabla I \frac{\partial W}{\partial \mathbf{p}}]. \quad (22)$$

If $\sum_x [\nabla I \frac{\partial W}{\partial \mathbf{p}}]^T [T(x) - I(W(x; \mathbf{p}))]$ is considered as the steepest gradient descent parameter update, then Formula 22 means $\Delta \mathbf{p}$ is actually the multiplication of steepest gradient descent parameter update and Hessian matrix inverse matrix. Therefore, Lucas-Kanade algorithm can repeat the iteration of Formula 16 and Formula 17 until convergence occurs or exceeds the iteration times. The specified procedure is shown follows:

Repeated iteration:

- (1) Calculate the transformation $I(W(\mathbf{x}; \mathbf{p}))$ of image I according to current parameter \mathbf{p} ;
 - (2) Calculate the error image $T(x) - I(W(\mathbf{x}; \mathbf{p}))$;
 - (3) Use $W(\mathbf{x}; \mathbf{p})$ to transform the gradient image ∇I ;
 - (4) Estimate the Jacobian determinants $\frac{\partial W}{\partial \mathbf{p}}$ at position $(\mathbf{x}; \mathbf{p})$;
 - (5) Calculate the steepest gradient descent image $\nabla I \frac{\partial W}{\partial \mathbf{p}}$;
 - (6) Calculate Hessian matrix according to Formula 22;
 - (7) Calculate the steepest gradient descent parameter update $\sum_x [\nabla(8) I \frac{\partial W}{\partial \mathbf{p}}]^T [T(x) - I(W(x; \mathbf{p}))]$;
 - (9) Calculate $\Delta \mathbf{p}$ according to Formula 21;
 - (10) Update parameter \mathbf{p} according to the Formula 16;
 - (11) Until $\|\Delta \mathbf{p}\| \leq \varepsilon$
-

After analyzing the procedure of the above algorithm, as parameter vector \mathbf{p} changes with the increase of iteration, steps from (1) to (9) need to be recalculated at each iteration. In order to reduce the time complexity of algorithm, some scholars proposed to use the method of reverse combination to conduct the parameter update. The main idea of the method is to exchange the roles of image I and template T , and use current deformation matrix and reverse matrix of augmented matrix to compose a new deformation matrix so that to replace the method of updating through incremental parameter vector \mathbf{p} as shown in Formulas 23 and Formulas 24:

$$\sum_x [T(W(x; \Delta \mathbf{p})) - I(W(x; \mathbf{p}))]^2. \quad (23)$$

$$W(x; \mathbf{p}) \leftarrow W(x; \mathbf{p}) \circ W(x; \Delta \mathbf{p})^{-1}. \quad (24)$$

Unfold the Formulas 23 at Level I Taylor series to obtain Formula 25:

$$\sum_x [T(W(x; 0)) + \nabla T \frac{\partial W}{\partial \mathbf{p}} \Delta \mathbf{p} - I(W(x; \mathbf{p}))]^2. \quad (25)$$

Without loss of generality, $W(x; 0)$ can be assumed as the identical transformation, saying $W(x; 0) = x$. Then the solution of the minimum square problem is:

$$\Delta \mathbf{p} = H^{-1} \sum_x [\nabla T \frac{\partial W}{\partial \mathbf{p}}]^T [I(W(x; \mathbf{p})) - T(x)]. \quad (26)$$

Where H refers to the Hessian matrix which replaced I with T :

$$H = \sum_x [\nabla T \frac{\partial W}{\partial \mathbf{p}}]^T [\nabla T \frac{\partial W}{\partial \mathbf{p}}]. \quad (27)$$

As current Hessian matrix has nothing to do with parameter vector \mathbf{p} , it is unnecessary to repeat the calculation at each iteration but calculate once and store in

advance, which improved the efficiency of the algorithm.

(2) Improved KL algorithm based on MEMS sensor

There are many restrictions when applying traditional KL feature tracking algorithm. The algorithm assumes the constant target brightness and allows only small movements of the object in adjacent frames. The first restriction is to assure the mean square error of target image and template image not influenced by illumination, and the later part is to assure KL algorithm could find out the solution to satisfy the threshold condition. However, in the actual application of mobile augmented reality, it is common when the device needs to move in a large scale or rotate rapidly, which makes traditional KL algorithm unable to accurately track the rapid and large-scale moving target scenes.

Considering the regular application scenes of mobile augmented reality, this paper uses inertial sensor to help KL algorithm select parameters and takes the affine illumination model proposed by H.Jin et al. [31] as the tracking model of the algorithm. This model contains 8 parameters: $p = (a_1, \dots, a_6, \alpha, \beta)$. In the model, affine matrix (A, b) is used to describe the movement of the device in the space, such as translation and rotation; proportion and migration model (α, β) is used to deal with the light contrast change of the template image; Alpha proportion is used to compensate the change of environment light and Beta migration is used to compensate the change of direct light. The whole model can be expressed as:

$$T(x; p) = (\alpha + 1)T(Ax + b) + \beta, \quad (28)$$

In which

$$A = \begin{bmatrix} 1 + a_1 & a_2 \\ a_3 & 1 + a_4 \end{bmatrix}, b = \begin{bmatrix} a_5 \\ a_6 \end{bmatrix}. \quad (29)$$

During the tracking progress, most of the introduction of light stream comes from the change of camera posture. Hence, three-axis gyroscope sensor can be used to record the instantaneous rotation q_t^{t+1} of current camera and p_{t+1}^0 can be predicted according to the current deformation parameter vector p_t as shown in Figure 10.

$$I_t I_{t+1} w(p_t) x_t w(p_t) \hat{x}_{t+1} x_t w(p_{gyro}) w(p_{t+1}^0) K R(p_{gyro}) K^{-1} \omega_2 \omega_3 \omega_1.$$

Suppose the difference between image I_t and image I_{t+1} is caused by the rotation R_t^{t+1} of camera, then under the premise of given the calibration parameter matrix K of camera, all the movements of feature points can be described by a 2D homography matrix H :

$$H = K^{-1} R_t^{t+1} K \quad (30)$$

Homography H is more advanced than the affine transformation (A, b) used for tracking target change in the model. So (A, b) can be obtained directly from Homography matrix. First, homography matrix needs to be standardized to be $H3X3$. The linear transformation part A pred is the same as upside 2×2 submatrix of matrix $H3X3$, describing the affine parts as rotation and dimension, etc. Essentially, b pred is the position offset caused by 2D homography matrix such as the position offset $x_H \equiv H[x \ 1]^T$ generated by point x through H . The illumination parameters

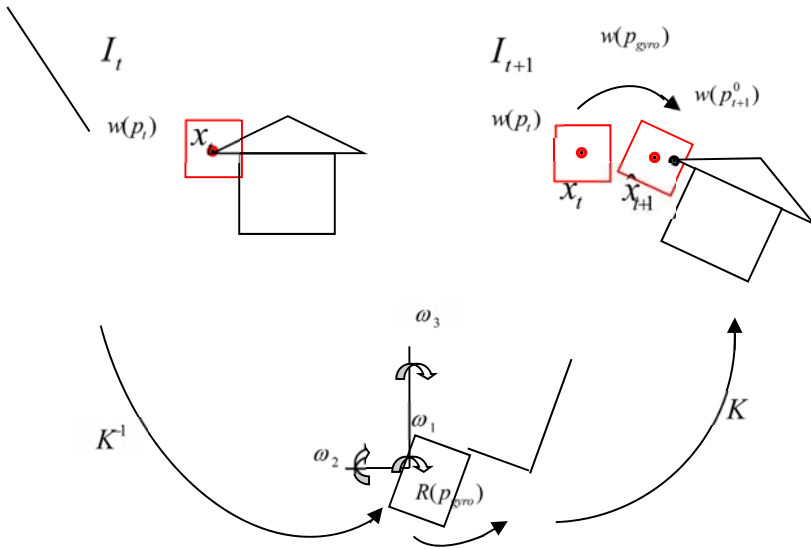


Fig. 10. Diagram of KL tracking algorithm based on inertia

in the model will not be influenced, so:

$$\begin{cases} A_{pred} = H_{2 \times 2} \\ b_{pred} = x_H - x \\ \alpha_{pred} = \beta_{pred} = 0 \end{cases} \quad (31)$$

In conclusion, the initial parameter of image I_{t+1} can be obtained through the forward combination of sensor prediction parameter p_{pred} and current parameter p_t , that is:

$$\begin{aligned} W(x; p_{t+1}^0) &= W(x; p_{pred}) \circ W(x; p_t) \\ &= W(W(x; p_{pred}), p_t) \end{aligned} \quad (32)$$

Guided by the test data from the sensor, p_{t+1}^0 is more easily to drop in the convergence interval C compared with p_t .

5. Design and realization of mobile augmented reality system

In the experiments, MEMS sensor data is applied to realize FREAK feature extraction and to improve algorithms as KL tracking registration.

5.1. System environment

Since iPhone is selected to perform the establishment process of the mobile augmented reality system in this study, the development tasks are mainly finished

in MacOS X 10.9 Mavericks. The main body of the system is programmed with Objective-C an object-oriented extension of C Programming Language with perfect downward compatibility with the existing image algorithm' C Language Library as OpenCV. The system project is compiled and published with Apple XCode 4.5, and then finally operated in iPhone5.

5.2. System process and structure

As shown in Figure 11, the system presented in this paper is mainly composed of two parts, namely offline stage and online stage. In offline stage, feature extraction and description are conducted for the reference image to form a feature data package, which will be applied as a matching basis in online stage.

In offline stage, every reference image will be described in two different ways, which are the traditional FREAK description algorithm and the gravity-based FREAK description algorithm. Since these procedures can all be completed in PC offline, they would not affect the registration efficiency of the mobile terminal.

The online procedures are the acquisition of video stream, the judgment of scene information, the estimation of camera pose and the display of model rendering, which are mainly finished on iPhones. Before the follow-up procedures, the video frame obtained with iPhone5 HD camera should be adjusted to the size of 480x640 to reducing the computation complexity in iPhone5.

The process of scene information is the core of the entire registration flow, which includes the matching and tracking of video features. Detailed procedures are shown in Figure 12.

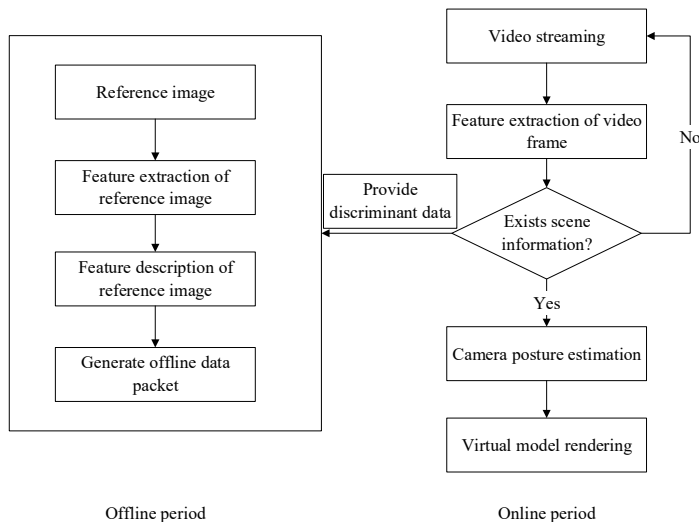


Fig. 11. Flowchart of mobile augmented reality system

First, obtain a frame of video image and extract its feature description. On augmented or approximate horizontal planes, mobile phones cannot produce significant

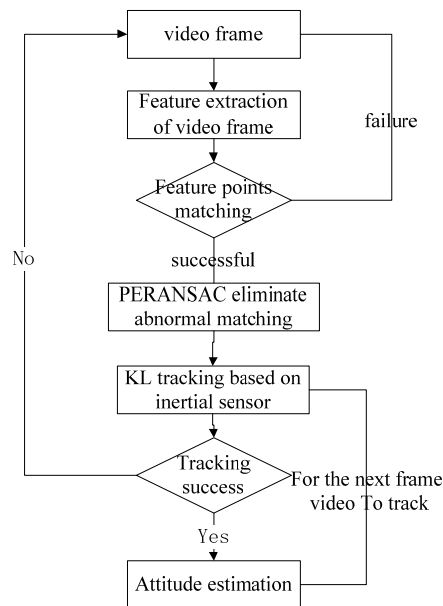


Fig. 12. Flowchart of scene detection and registration tracking

project-vector on imaging plane due to gravity factor. In this case, the system can shift to the traditional FREAK description algorithm and use the gray level of feature points' adjacent area to set the feature direction. This is also the reason why two algorithms are required in offline stage.

Then, change the algorithm according to the current pose of the mobile phone estimated by the built-in MEMS sensor of iPhone. If it similar to vertical pose, Gravity-FREAK algorithm will be applied to conduct the feature point description, otherwise FREAK algorithm will be selected.

Match the obtained feature points with those of the reference image and use PERANSAC algorithm to screen the matching features. Remove the abnormal data that cannot meet the requirements. If the rest matching feature is larger than a certain threshold value (e.g. 20), it indicates that the objective scene do exist in this image frame. In this case, the rest matching points can be used to evaluate the projection matrix between the reference image and the current image.

Meanwhile, in order to make full use of the continuity of two adjacent video frames, the projection matrix from the reference image to image t will be recorded when the objective scene is detected in a certain image frame t . Gravity-KL will then utilized to track the scene. If the scene is successfully traced, the transition matrix of this image frame relative to the former image frame can also be obtained; iterate it to image t to calculate the projection matrix from the reference image to this image frame, which marks the completion of the track registration. If the track failed, return to the scene detection stage and restart the feature matching. Change the projection of the virtual model in line with the projection matrix from the reference image to the current video image; render and superimpose it to the

right position in the current image to acquire a fusion image, thus realizing the augmentation of the reality scene. The tree-dimensional model used in this paper is created with 3dMax and rendered with OpenGL after format conversion.

5.3. Experimental for Gravity-KLT algorithm

In this study, iPhone5 is chosen as the carrier of the experimental program; scene videos in the size of 640X480 resolutions and the frequency of 30Hz are applied as the input source to test the tracking results of the original and improved KL algorithms. The rotation movements of iPhone5 along X , Y and Z axis are recorded in the scene video, and the current rotation parameters are also recorded with the gyroscope for mobile phone.

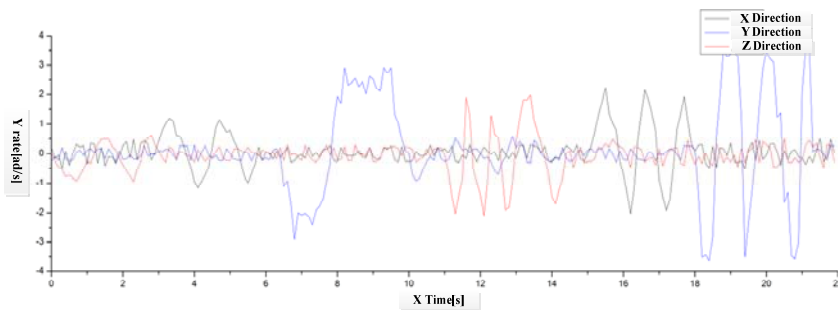


Fig. 13. Diagram of iphone5 angular velocity change

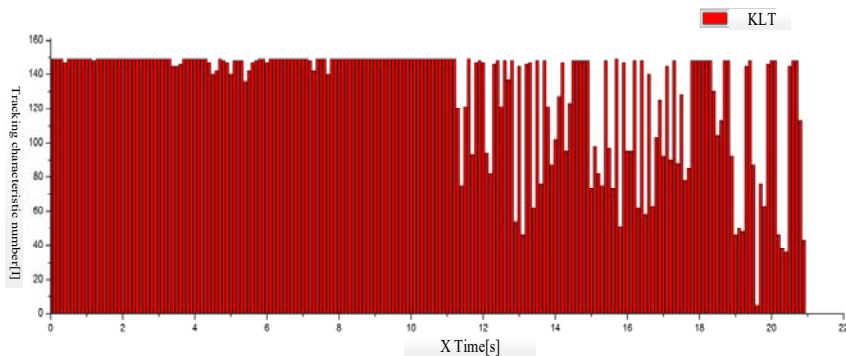


Fig. 14. Diagram of tracking feature change of traditional KLT algorithm

In the scene video as illustrated in Figure 13, the mobile phone firstly rotates along X , Y and Z axis separately at low speed (0~11s), and then at high speed (12~22s). According to Figure 14 and 15, both algorithms have achieved good tracking results in slow movements. When the mobile phone starts to rotate at high speed, the traditional KLT algorithm presents big fluctuation as the angular velocity changing violently. It indicates that the traditional KLT algorithm fails to acquire the tracking features in rapid movements, which would further result in the loss of

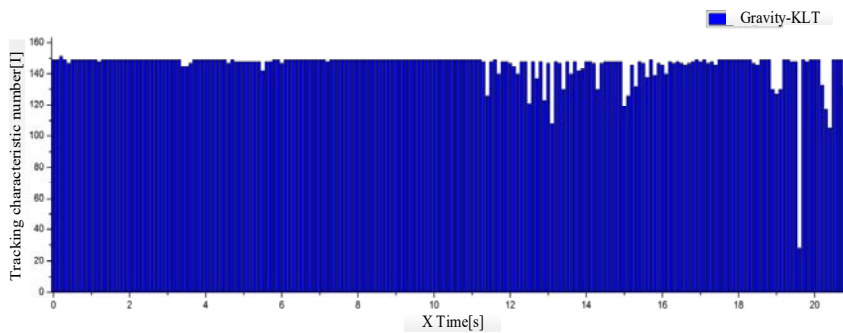


Fig. 15. Diagram of tracking feature change of improve KLT algorithm



Fig. 16. Power line inspection

targets. On the contrary, the improve algorithm can predict the positions of feature points with the rotation parameters obtained through the sensor, and then starts the search from these positions. As shown in Figure 16, this kind of prediction is quite precise due to the support of the measured data of the sensor. In other words, the improved KLT algorithm can achieve good tracking effects when the mobile phone is rotating at high speed.

6. Conclusion

With the widespread use of smartphones in recent years, AR application has been transited from PC system to mobile system. Its mobility and real-time problem also brings a lot of challenges. Based on this, this paper proposed the Gravity-FREAK feature extraction algorithm and Gravity-KLT tracking registration algorithm based on MEMS sensor to improve the classic AR algorithm based on PC. Compared with traditional FREAK algorithm, Gravity-FREAK feature description algorithm only needs to use the system interface to read the sensor data when calculating directions, which can save the time of calculating the gray gradient at feature point neighbor-

hood and improve the operating speed of the algorithm. Meanwhile, the description on the feature will be more accurate due to the accuracy requirement of the sensor itself. Regarding the defect of traditional KL algorithm's sensitivity on rapid rotation and a wide range of movement, MEMS sensor can provide initialization parameters for KLT algorithm which would enable the increase of algorithm convergence rate when the camera moves fast and improve the robustness of the registration tracking of mobile augmented reality. In the future, deeper improvements can be made based on this paper. First, improvements can be made on the algorithm effectiveness when the device is positioned at horizontal or approximately horizontal scene. Secondly, on the mechanism of offline registration, dependency on PC offline registration can be further solved by using cloud computing server combined with smartphones to realize remote online registration. Finally, improvements can be made on virtual registration object and visual fusion effect of real scene, including adding more real shadow and illumination effect, restoring the obstruction effect as well as the consistent experience of real world and virtual world to improve the overall user experience of mobile AR system.

Acknowledgement

The Project was supported by the National Natural Science Foundation of China (No. 61402386, No. 61305061 and Grant No. 61402389). And we wish to thank the anonymous reviewers who helped to improve the quality of the paper.

References

- [1] WU YUEZHOU; LIU CHANGJIANG; LAN SHIYONG: (2015) *Real-Time 3D Road Scene Based on Virtual-Real Fusion Method*, IEEE Sensors Journal, FEB Vol.15, Issue.2, pp(750-756).
- [2] VERYKOKOU STYLIANI; IOANNIDIS CHARALABOS; KONTOGIANNI GEORGIA: (2014) *3D Visualization via Augmented Reality: The Case of the Middle Stoa in the Ancient Agora of Athens*, 5th EuroMed International Conference, 2014, Vol.8740, Lecture Notes in Computer Science, pp(279-289)
- [3] NOWICKI MICHAL; SKRZYPCZYNSKI PIOTR: (2013) *Robust Registration of Kinect Range Data for Sensor Motion Estimation*, 8th International Conference on Computer Recognition Systems (CORES 2013) , 2013, Vol. 226, Advances in Intelligent Systems and Computing, pp(835-844)
- [4] AZUMA R, BAILLOT Y, BEHRINGER R, ET AL.: (2001) *Recent advances in augmented reality*[J]. Computer Graphics and Applications, IEEE, 21(6): 34-47.
- [5] CAUDELL T P, MIZELL D W: (1992) *Augmented reality: An application of heads-up display technology to manual manufacturing processes*[C]//System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on. IEEE, 2: 659-669.
- [6] FEINER S, MACINTYRE B, HÖLLERER T, ET AL.: (1997) *A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment*[J]. Personal Technologies, 1(4): 208-217.
- [7] KATO H: (2002) *ARToolKit: library for Vision-Based augmented reality*[J]. IEICE, PRMU, 2002: 79-86.

- [8] WAGNER D, SCHMALSTIEG D: (2003) *Artoolkit on the pocketpc platform*[C]//IEEE International Augmented Reality Toolkit Workshop. Darmstadt, Germany, 2003: 14-15.
- [9] HUYNH D N T, RAVEENDRAN K, XU Y, ET AL.: (2009) *Art of defense: a collaborative handheld augmented reality board game*[C]//Proceedings of the 2009 ACM SIGGRAPH symposium on video games. ACM, 2009: 135-142.
- [10] LEE JIANN-DER; HUANG CHUNG-HSIEN; HUANG TSING-CHE: (2012) *Medical augmented reality using a markerless registration framework*, Expert Systems With Applications , APR 2012, Vol.39, Issue.5, pp(5286-5294)
- [11] SAMIR CHAFIK; KURTEK SEBASTIAN; SRIVASTAVA ANUJ: (2014) *Elastic Shape Analysis of Cylindrical Surfaces for 3D/2D Registration in Endometrial Tissue Characterization*, IEEE Transactions On Medical Imaging, MAY 2014, Vol.33, Issue.5, pp(1035-1043)
- [12] DAVISON A J, REID I D, MOLTON N D, ET AL.: (2007) *MonoSLAM: Real-time single camera SLAM*[J]. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on, 29(6): 1052-1067.
- [13] PUPILLI M, CALWAY A: (2005) *Real-Time Camera Tracking Using a Particle Filter*[C]//BMVC.
- [14] CHEKHLOV D, PUPILLI M, MAYOL W, ET AL.: (2007) *Robust real-time visual slam using scale prediction and exemplar based feature description*[C]//Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE, 2007: 1-7.
- [15] KLEIN G, MURRAY D: (2008) *Improving the agility of keyframe-based SLAM*[M]//Computer Vision—ECCV 2008. Springer Berlin Heidelberg, 2008: 802-815.
- [16] KLEIN G, MURRAY D: (2009) *Parallel tracking and mapping on a camera phone*[C]//Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on. IEEE, 2009: 83-86.
- [17] WAGNER D, REITMAYR G, MULLONI A, ET AL.: (2010) *Real-time detection and tracking for augmented reality on mobile phones*[J]. *Visualization and Computer Graphics*, IEEE Transactions on, 16(3): 355-368.
- [18] WAGNER D, SCHMALSTIEG D, BISCHOF H: (2009) *Multiple target detection and tracking with guaranteed framerates on mobile phones*[C]//Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on. IEEE, 2009: 57-64.
- [19] MAIR E, HAGER G D, BURSCHKA D, ET AL.: (2010) *Adaptive and generic corner detection based on the accelerated segment test*[M]//Computer Vision—ECCV 2010. Springer Berlin Heidelberg, 2010: 183-196.
- [20] ROSTEN E, DRUMMOND T: (2006) *Machine learning for high-speed corner detection*[M]//Computer Vision—ECCV 2006. Springer Berlin Heidelberg, 2006: 430-443.
- [21] BIMBER O, FROHLICH B: (2002) *Occlusion shadows: Using projected light to generate realistic occlusion effects for view-dependent optical see-through displays*[C]//Mixed and Augmented Reality, 2002. ISMAR 2002. Proceedings. International Symposium on. IEEE, 2002: 186-319.
- [22] DEBEVEC P: (2008) *Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography*[C]//ACM SIGGRAPH 2008 classes. ACM, 2008: 32.
- [23] YOU S, NEUMANN U, AZUMA R: (1999) *Hybrid inertial and vision tracking for augmented reality registration*[C]//Virtual Reality, 1999. Proceedings., IEEE. IEEE, 1999: 260-267.
- [24] CHUM O, MATAS J, KITTLER J: (2003) *Locally optimized RANSAC*[M]//Pattern Recognition. Springer Berlin Heidelberg, 2003: 236-243.
- [25] CHUM O, MATAS J, OBDRZALEK S: (2004) *Enhancing RANSAC by generalized model optimization*[C]//Proc. of the ACCV. 2: 812-817.
- [26] SCHNABEL R.; WAHL R.; KLEIN R.: (2007) *Efficient RANSAC for point-cloud shape detection*, Computer Graphics Forum, 2007, Vol.26, Issue.2, pp(214-226)
- [27] CHEN FU-XING; WANG RUN-SHENG: (2005) *Fast RANSAC with Preview Model Parameters Evaluation*, Journal of Software, 2005, Vol.16, No.8, pp(1431-1437)

- [28] CAPEL D P: (1981) *An Effective Bail-out Test for RANSAC Consensus Scoring*[C]//BMVC. 2005.
- [29] LUCAS B D, KANADE T.: *An iterative image registration technique with an application to stereo vision*[C]//IJCAI. 81: 674-679.
- [30] BAKER S, MATTHEWS I: (2004) *Lucas-kanade 20 years on: A unifying framework*[J]. International Journal of Computer Vision, 56(3): 221-255.
- [31] JIN H, FAVARO P, SOATTO S: (2001) *Real-time feature tracking and outlier rejection with changes in illumination*[C]//Computer Vision, IEEE International Conference on. IEEE Computer Society, 1: 684-68
- [32] LIEBERKNECHT S, BENHIMANE S, MEIER P, ET AL.: (2009) *A dataset and evaluation methodology for template-based tracking algorithms*[C]//Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on. IEEE, 2009, pp(145-151).

Received May 7, 2017

