

KAZE Algorithm Applied in Augmented Reality

Qing Zhu, Zhenglong Lei

School of Software Engineering, Beijing University of Technology, Beijing 100124, China

Keywords: Augmented Reality; KAZE; features detection; match.

Abstract. The Augmented Reality system based on KAZE algorithm is to do nonlinear diffusion filtering by the additive operator splitting algorithm. In this way, the problem of blurred boundaries and detail missing can be solved. A stable nonlinear scale space is constructed by using arbitrary step to search the Hessian local maximum value point after different scales normalizing to detect feature points. Meanwhile, it combines KLT algorithm when tracking the target feature in order to improve the efficiency. Experiment show that, since the KAZE algorithm matching feature points are better and the positioning accuracy is higher, therefore this AR system has good effect of real-time virtual and actual fusion in special environments such as insufficient light, occlusion and jittering.

1. System framework

The system executes feature's extraction and matching by KAZE algorithm. First, it improves the efficiency and quality of the key frame image's feature extraction by preprocessing the image of the scene. Then it builds scale space by nonlinear diffusion, and detect key features by seeking extreme points. Then use a method similar to the surf Algorithm to determine the location and orientation of the key points and generate the feature description vectors. Furthermore, the system matches with the key feature vector of the target image. We use RANSAC to purify mismatching feature points in order to improve the success rate of matching. Finally, combine KLT algorithm with optical flow tracking algorithm to track the success matched feature points. The system obtains internal and external parameters to get real-time update of real scenes estimated pose, and to achieve sub-pixel level tracking by using camera calibration. On this basis, through the 3D virtual scene rendering, enables seamless integration of three dimensions virtual objects and real-world. Specific processes are shown in Figure 1.

2. Kaze algorithm

In ECCV2012, appears a more stable feature detection algorithm, KAZE, which is better than SIFT. KAZE, commemorate the pioneers of scale-space analysis—Japanese scholars, Iijima. KAZE is a homonym of Japanese "wind". It means the images cause the flow of nonlinear diffusion way to be processed just like wind. This is the KAZE [1].

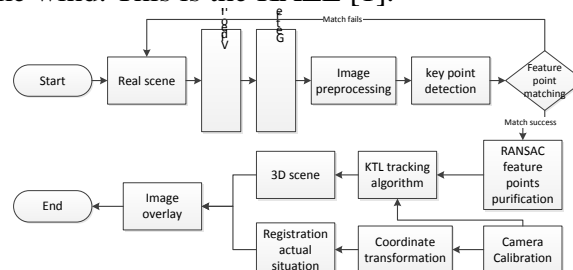


Figure 1 system flow chart

Traditional SIFT, SURF and other feature detection algorithms are based on linear multi-scale Gaussian pyramid decomposition to eliminate noise and extract significant feature points. But Gauss decomposition is sacrificed at the expense of partial accuracy, likely to cause fuzzy boundaries and loss of details. Nonlinear scale decomposition is expected to solve this problem. However, when the traditional method which is based on the forward Euler scheme solves nonlinear diffusion equation, the step size of iterations converges are too short. Meanwhile, time-consuming and high

computational complexity are still its weakness [1]. Thus, the author of KAZE algorithm proposed that we should use nonlinear diffusion filtering by Additive Operator Splitting algorithm (AOS).

Nonlinear diffusion filter. Nonlinear diffusion [1] filtering method is to consider the image brightness's changing at different scales as a form of flow function's divergence. It can be described by nonlinear partial differential equations:

$$\frac{\partial L}{\partial t} = \text{div}(c(x, y, z)\nabla L) \quad (1)$$

Diffusion can be made adaptive to the local structure of the image by setting the appropriate conduction function $c(x, y, t)$. Conduction can be scalar functions, and can also be tensor. Time t is the scale parameter, the greater it is, the simpler representation of the image is. Perona and Malik proposed construction method of conduction function:

$$c(x, y, t) = g(|\nabla L_\delta(x, y, t)|) \quad (2)$$

The ∇L_δ in the function is the gradient of the image L_δ Gaussian smoothed. Function $g()$ is in the form of the following categories:

$$g = \frac{1}{1 + \frac{|\nabla L_\delta|^2}{k^2}} \quad (3)$$

The parameter k is the contrast factor that controlling the level of diffusion. It decides how much edge information is retained. The greater value it is, the less the edge information is retained. In KAZE algorithm, the parameter k is a value of 70% percentile of the gradient image ∇L_δ 's histogram.

AOS Algorithm. The general method of iterative solution is numerical analysis, because nonlinear partial differential equations have no analytical solution. The traditional solution method, that using explicit difference schemes, can only use small step. The convergence of it is slow. To this end, discrete the following equations into implicit difference scheme:

$$\frac{L^{i+1} - L^i}{T} = \sum_{i=1}^m A_i(L^i)L^{i+1} \quad (4)$$

This solution method is valid for any time step T . In the above formula, A_i is tridiagonal and diagonally dominant matrix? This linear system can be quickly solved by the Thomas algorithm.

KAZE feature detection and description

KAZE feature detection step is as follows:

1) Construct a nonlinear scale space by AOS diffusion algorithm and variable conductance diffusion [2] method at first.

2) Detecting interested feature k points by calculating partial maxima number.

3) Calculate the main direction of the feature points, and extract description vector which has the scale and rotation invariant basing on the first derivative image.

The structure of nonlinear scale space. KAZE's structure of scale spatial is familiar with SIFT [3]. The scale level is increasing logarithmically, it total concludes O group octaves, and each octave has S sub-level. KAZE adopt the resolution that same with the original image in each level, which is different with SIFT, whose every new octave down sample layer by layer. Different octave and sub-level respectively marked by number o and s , and correspond with scale parameters σ by the following formula:

$$\delta_i(o, s) = \delta_o 2^{o+s/S} \quad (5)$$

δ_o Is the initial reference value of scale parameters. $N=O*S$ is the total number of images of the entire scale space contains. Known from the preceding description, nonlinear diffusion filtering model is based on units of time, therefore we need to scale parameter δ_i pixels converted to time units. In the case of the Gaussian scale space, the convolution of an image with a Gaussian of standard deviation σ (in pixels) is equivalent to filtering the image for some time $t=\sigma^2/2$. Thus, we get the mapping formula that converted scale parameter δ_i to units of time:

$$t_i = \frac{1}{2} \delta_i^2, i = \{0 \dots N\} \quad (6)$$

t_i Is called evolution time. It is worth being noted that this mapping is only used to obtain one set of evolutionary time value. It constructs nonlinear scale space through these time values. In general, in the nonlinear scale space at each filtered image t_i , the resulting image does not correspond with the convolution of the original image with a Gaussian of standard deviation δ_i . But as long as the conduction function g identically equal to 1 (i.e., g is a constant function), non-linear scale space is equivalent to a Gaussian scale space. Moreover, with the scale enhance the level, except for those image edge pixels corresponding to the target contour, most conduction function values corresponding to the pixels that will tend to a constant value.

Feature point detection. KAZE's feature point detection is similar with SIFT. It is achieved by looking for Hessian partial maxima which is normalized in different scales [2]. Hessian matrix is calculated as follows:

$$L_{Hessian} = \delta^2(L_{xy}L_{yy} - L_{xy}^2) \quad (7)$$

δ is the integer value of the scale parameter δ_i . When looking for extreme point, each pixel compare with all its neighboring point. When it is larger than all of its neighboring points both in image domain and scale domain, it is the extreme point. Three rectangular windows with the size $\delta_i \times \delta_i$ in current scale, the former scale and the next scale is the range of theoretically comparison. However, in order to speed up the search, the window size is fixed at 3×3 , so search space is a side length of 3 pixel cube. Intermediate detection point compare with 26 point that includes 8 neighboring point having same scale, and 9×2 points which is correspondence with the adjacent scale up and down. In this way, it can detect the extreme points both in scale space and two-dimensional image space.

Characterization vector. In order to realize image rotation invariance, need to determine the main direction according to the partial image structure of feature points. The method is that if the scale parameter of the feature point is δ_i , then set the search radius as $6\delta_i$, which is familiar with SURF. To make the larger contribution of response that is close to the feature points, the less contribution of response this is farer to the feature points, Gaussian-weighted all first order differential value L_x and L_y in the search circle. Consider all these differential value as the set of points in the vector space; vector superimposed the point in an angle of 60° fan-shaped sliding window, let it traverse the entire circular area. The main direction of the vector gets the maximum angle [2].

The following is the results of feature point detection. It is the KAZE feature map of face images which is according to the above steps. The length of mark circle's radius is related with the scales that feature point lying. The radius is the feature point's direction. As shown in figure:



Figure 2 result of Feature point detection

3. Feature tracking based on klt

KLT fundamentals. Because of the KAZE algorithm is more complicated than other algorithms, it can't be directly used in scene change's real-time tracking. It needs to detect all target frames' pixels. The larger the image resolution is, the more time consuming to detect. So, it needs to combine the KLT tracking algorithm when tracking the target feature points. In this way not only can reduce the time cost, but also can ensure efficient matching performance.

KLT (Kanade-Lucas-Tomasi) is an image feature tracking algorithm that based on the optimal estimation. It defines the next feature's coarse position by forecast estimation [8].

KLT algorithm is a fast feature point tracking method, its accuracy can meet the real-time requirements of augmented reality even when in the scene that owns less iterations [4]. This algorithm is basing on translational motion model. Translational motion model's theory is that its sport has little change in the frame image stream of the video sequence adjacent two frames. Assuming on time t, there is a point I(x, y) on the image move to I(x - ε, y - η) after time T. Maked $d = (\epsilon, \eta)$ called translational motion parameter vector, and assume the later image frame point J can get from the former image frame point I translating d, and their gray values are approximately equal, that:

$$J(x) = I(x - d) + n(x) \quad (8)$$

In the above formula, J(x) is the gray value that point I throwing time T, n(x) is the corresponding noise, so the error function is:

$$\xi = \int w[I(x - d) - J(x)]^2 w dx \quad (9)$$

In the above formula, w is the different pixel's weighting equation in the feature region. In order to make the double integral's residuals within some feature window around feature point I(x, y) as small as possible, need to choose the best motion parameter vector d. So, calculate the best motion parameter vector d by iteration, thereby tracking feature window. Using the feature key frame as the refresh condition makes the refresh of the smoothing feature point faster and makes the probability of the error rate when updating declines. Detect the number of the feature points in real time. When the number is less than the threshold value (now set 20), do refresh the KLT's feature points. The figure shows that the result of experiment.

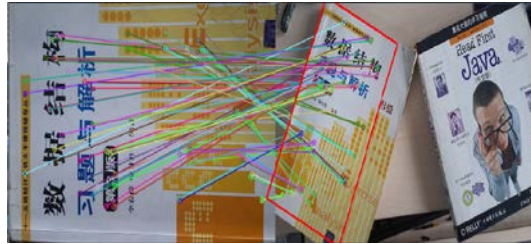


Figure 3 result of the matching

4. Feature tracking based on klt

Camera calibration is defined as a progress that solving corresponding geometric model parameters for determining the relationship between dimensional position space objects and its points in the flat image. The number separated into two parts, one is intrinsic internal parameters of camera (focal length f, lens distortion coefficients, coordinate distortion factor, coordinate origin), and the other is external parameters changing following with the target scene (rotation vector R, the translation vector T). For realizing the real-time tracking of target object, it goes through three-step conversion which is world coordinate system---camera coordinate system---imaging coordinate system---screen coordinates when projecting the success matched object coordinates (x_w, y_w, z_w) into two-dimensional screen coordinates (u, v). Conversion relation is shown at formula. Sees from right to left; it is followed by the camera coordinate system (x_c, y_c, z_c) which is conversed by homogeneous coordinates from the world coordinate system (x_w, y_w, z_w) , and then left multiply the camera coordinate system (x_c, y_c, z_c) which is rotated, translation converted. Then use the triangle similarity principle, $x = f \times X_c / Z_c, y = f \times Y_c / Z_c$ to get (x, y) which is in the imaging plane coordinate system. Then convert to the screen coordinates, get (u, v) in two-dimensional plane, during that time, it should do normalization process because the effect of tilt factor s' which is at the camera imaging coordinate's axes.

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{d_r} & s' & u_o \\ 0 & \frac{1}{d_r} & v_o \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R \\ T \\ 1 \end{bmatrix} \quad (10)$$

5. Experimental result

The figure shows that under the condition of Gaussian Blur, angle rotating, scale transformation and intensity roughness, the system still has a good match.



Figure 4 results of the matching

These figures are the fusion effect after target scene image's grading scale, moving (a), rotation (b), light's intensity changing(c), and occlusion conditions (d)'s tracking register.

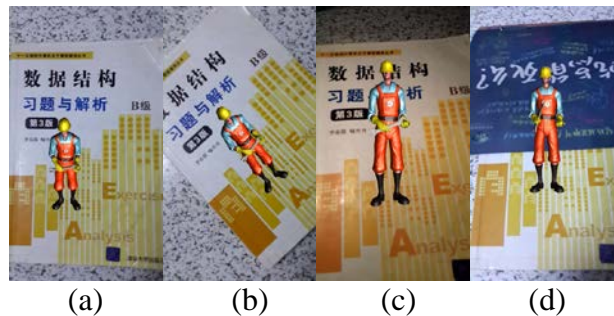


Figure 5. The system results in different environment

The figure (a), (b), (c), (d)'s good actual situation fusion effect in different complicated environment indicates the feasibility of the method. The research result proves that the KAZE has better performance even if under the condition of Gaussian Blur, angle rotating, scale transformation and intensity roughness. It is a solid foundation of coming tracking and actual situation integration that improving the efficiency of whole system running.

6. Conclusions

Augmented reality has been listed as one of the top 10 most promising future technology by authority agency such as American Times and Gartner because of its natural human-computer interaction and strong practical immersion. Augmented reality has been widely applied in digital marketing, design and simulation, science education and mobile navigation, etc. This article is augmented reality system which research and develop that basing on part of KAZE algorithm. The system has better real-time performance and robustness after partial improvement of system algorithm. It is a digital platform shows that applying the complex changing outdoor environment. Because of the partial limited capacity of features describing, the middle patch that combining the deep learning will be a hot research in the future. On the other hand, the RGB-D data that obtained combining with somatosensory camera will have a broader space development in human-computer interaction.

References

- [1] Alcantarilla PF, Bartley A, et al. 2012.KAZE features. Computer Vision – ECCV 2012.
- [2] Alcantarilla PF, Bergasa LM, Davison AJ. 2013. Gauge-SURF Descriptors. Image and Vision Computing. 31(1):103–116.
- [3] Agrawal M, Konolige K, Blas MR. 2008.Census: Center surround extreme as for real-time feature detection and matching. Computer Vision-ECCV 2008.5305:102–115.

- [4] Bay H, Ess A, and Tuytelaars T, Gool LV. 2008.SURF: Speeded up robust features. *Computer Vision and Image Understanding*. 110: 346–359.
- [5] Jianbo Shi and Carlo Tomasi.1994.Good Features to Track. *IEEE Conference on Computer Vision and Pattern Recognition*, pages: 593-600.
- [6] Lindeberg T.1998.Feature detection with automatic scale selection. *International Journal of Computer Vision*. 30(2):77–116.
- [7] Pintaric T. 2003. An adaptive thresholding algorithm for the augmented reality toolkit. *IEEE International Augmented Reality Toolkit Workshop*: 71.
- [8] Weickert J, Ishikawa S, Imiya A. 1999.Linear scale-space has first been proposed in Japan. *Journal of Mathematical Imaging and Vision*.10 (3): 237–252.