3D Object Recognition Method Based on Improved Canny Edge Detection Algorithm in Augmented Reality

Tianhan Gao
Dept. of Software Engineering
Northeastern University
Shenyang, China
e-mail: gaoth@mail.neu.edu.cn

Zhenhao Yang
Dept. of Software Engineering
Northeastern University
Shenyang, China
e-mail: yangzh@stumail.neu.edu.cn

Abstract—**Augmented reality (AR) superimposes computer-generated virtual objects on real scenes to gain immersive experience. Effective recognition of 3D objects in real scenes is the fundamental requirement in AR. The traditional Canny edge detection algorithm ignores the important boundary information about the object, thus decreasing the recognition accuracy. In this paper, we improve Canny to propose a novel 3D object recognition method, where median filtering is adopted in order to extract the contour of the object instead of Gaussian fuzzy. An operator based on wedge template is designed to improve the boundary detection effect of the corner. Local feature descriptors are then introduced to describe the local feature points of the object. Finally, SLAM technology is conducted to ensure that the virtual model is stably superimposed above the 3D object. The experimental results show that the proposed method is able to retain the edge information of the object well and can be combined with local feature descriptors to accurately recognize 3D objects.**

Keywords-Augmented Reality; Edge Detection; Median Filtering; Local Descriptor; SLAM

I. INTRODUCTION

Augmented Reality (AR) superimposes computer-generated virtual information and real scenes to present virtual information more intuitively, which has been widely used in education, medical treatment, architecture, game and other fields [1]. Effective recognition of 3D objects is the essentials to achieve AR applications [2]. Existing recognition methods can be divided into four categories: local feature-based methods [3], global feature-based methods [4], image matching-based methods [5], as well as machine learning-based methods [6]. The method based on local features can identify 3D objects by finding local information features. It does not need to segment the recognized objects, but extracts and compares local features such as key points, edges or patches of objects to complete the recognition. However, this method has lower recognition accuracy with few surface feature points [7]. The method based on global features needs to segment the target object from the background, and recognize the object by describing and comparing the significant geometric features of the object. While, the method suffers from the poor recognition effect on objects with smooth surface and irregular shape [8]. Image matching method decomposes point cloud data into basic shapes, expresses them with abstract points, and then uses topological maps to represent the proximity relationship between shapes to realize object recognition through the point cloud data. This method also has poor recognition effect on objects with few surface feature points [9]. Recently, machine learning method begins to be employed for the classification and recognition of 3D objects. Unfortunately, the method is hard to ensure real-time performance in AR applications due to the tedious calculation overhead [10].

In general, existing AR-oriented 3D object recognition methods have different degrees of defects in recognition accuracy, efficiency, or suitability. In this paper, an improved edge detection algorithm is proposed to detect the contour of 3D objects. The Gaussian filter in Canny detection is replaced by the median filter, which is able to reduce the loss of important information on the edge. A novel operator based on wedge template is designed to improve the edge detection effect of the corner. Combined with local feature descriptors, the 3D object is accurately recognized. Moreover, SLAM is adopted in order to ensure that the virtual model is stably displayed above the object. The experimental results show that the improved edge detection algorithm can better preserve the edge information about the object, and accurately recognize the 3D object with the help of local feature descriptors.

II. PRELIMINARIES

A. Canny Edge Detection Algorithm

Canny is a multi-level edge detection algorithm [11]. Compared with traditional edge detection algorithms, such as Sobel operator and Roberts operator [12], Canny owns better detection accuracy and higher signal-to-noise ratio. As shown in Figure 1, Canny firstly smoothes the image and calculates the gradient blessing and direction. Then the gradient blessing is suppressed non-maximally. The double threshold algorithm is further conducted to detect the edge information of the object. Finally, the edge detection is achieved by suppressing the isolated weak edges.

Canny uses Gaussian filtering to denoise. Firstly, the pixels in the image are scanned, where the gray values of all pixels in the neighborhood are added according to the weight to gain the weighted average value. The value is then assigned to the middle of the template. When expectation u

978-1-7281-6661-2/20/$31.00 ©2020 IEEE

tis equal to 0, the formula for the two-dimensional Gaussian distribution is as (1), where σ is the standard deviation.

$$\mathrm{f(x,y)} = \mathrm{f(x)f(y)} = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

However, Gaussian filtering denoising will lead to the weakness of the edge information, especially weak and isolated edges [13].

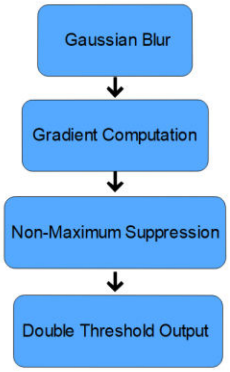

Figure 1. The Process of Canny

B. SLAM

Simultaneous Localization and Mapping (SLAM) provide space location function for AR applications, which include the following steps: sensor data processing, camera position and attitude estimation, back-end nonlinear optimization, loop detection, and 3D point cloud map construction. SLAM can be further divided into Radar SLAM and Visual SLAM according to the hardware equipment. Visual SLAM has the advantages of low cost, large amount of information obtained, and wide application range. Visual SLAM calculates the position and posture information about the camera through adjacent image matching, and carries out triangulation from two viewing angles to obtain the depth information about the corresponding points, thus realizing positioning and mapping through iteration.

III. The Proposed Method

A. Improved Edge Detection Algorithm

1) Selective blurring based on median filtering

The improved algorithm replaces the Gaussian filter in Canny with median filter, which is a typical nonlinear filtering method that can preserve the edge information of the image during denoising.

The median filtering uses a template to slide in the image so that the center of the template coincides with each pixel point of the image. When the center of the template coincides with the pixel points, the median value of the pixel points is computed according to the gray value around the center of the template to replace the gray value of the center position of the original template. Consequently, the gray value around each pixel point is close to the expected gray value for denoising. As shown in Figure 2, in a 3×3 sliding window, the gray value of each pixel point is set to X_1, X_2... X_9 respectively and is sorted from small to large to obtain the median M of the pixel points. Afterward, M is assigned to the center point of the sliding window.

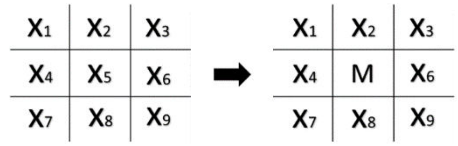

Figure 2. Using Median Filter to Replace Pixel Values at Matrix Center Points

2) Improved operator based on wedge template

We divide the circle into multiple wedges to make the angle between the convolution kernels of every two wedges small enough. Since the boundary probability values generated by the convolution kernels are the same regardless of whether the points (x, y) are at the edge of the straight line or at the corner, the boundary detection effect at the corner is thus improved. Let θ_1 and θ_2 be the angles of two wedge convolution kernels respectively. At the corners, the gradient direction $Pb(x,y,\bar{\theta})$ of the actual edge and the boundary probability value $\bar{\theta}$ are calculated as (2) and (3) according to [15].

$$\bar{\theta} = \frac{Pb(x,y,\theta_1)\cdot(\theta_1 \mp \pi) + Pb(x,y,\theta_2)\cdot(\theta_2 \overline{=} \pi)}{Pb(x,y,\theta_1) + Pb(x,y,\theta_2)} \tag{2}$$

$$Pb(x,y,\bar{\theta}) = \frac{\sqrt{Pb^2(x,y,\bar{\theta}_1) + Pb^2(x,y,\bar{\theta}_2) + 2Pb(x,y,\bar{\theta}_1)Pb(x,y,\bar{\theta}_2)\cos|\theta_1 - \theta_2|}}{2} \tag{3}$$

B. Local Feature Descriptor

After determining the location and range of local feature points, it is necessary to describe the local feature points to express the local structural information of the image, such as color, texture, shape, etc. [16]. The Key points-based Surface Representation (KSR) method is adopted to describe the local information [17]. First, we should detect the key points on the input 3D surface. Then the 3D feature points are extracted for the calculation of the distance and relationship between the feature points. The geometric distance measurement between each designated key point P_i and other key points on the 3D surface is derived as (4).

$$d(P_i, P_N) = \sqrt{\sum_{N=2}^{N} (P_i - P_N)^2} \tag{4}$$

Afterwards, a subset of feature points is selected to calculate KSR. Then, the two key points P_{n1} and P_{n2} that are the shortest from the key point P_i are found, and the subset α_i where P_i is located can be obtained.

$$\alpha_i = \begin{bmatrix} x_{P_i} & x_{P_{n1}} & x_{P_{n2}} \\ y_{P_i} & y_{P_{n1}} & y_{P_{n2}} \\ z_{P_i} & z_{P_{n1}} & z_{P_{n2}} \end{bmatrix} \tag{5}$$

Finally, the geometric relation between the two subsets α_i and α_{i+1} is expressed by τ_{α_i}, and the KSR between the subsets is gained. The conversion mode between the subsets is as (6) and (7).

$$\alpha_{i+1} = \tau_{\alpha_i}\alpha_i \tag{6}$$

$$\tau_{\alpha_i} = \alpha_{i+1}(\alpha_i)^{-1} \tag{7}$$

The Gaussian difference edge detection operator is further employed to measure the key points of the 3D surface. The image is convolved with the Gaussian function to obtain the low-pass filtering result [18], which can filter out the areas with small gray value change range to enhance the denoising effect. The calculation steps are as follows: (1) Convolution is carried out by using two Gaussian check images with the same size and different standard deviations; (2) The difference of convolution results is calculated to normalize the results. The Difference of Gaussian (DOG) is shown in (8), Where A and B are constants, and σ_1 and σ_2 are standard deviations of two Gaussian functions, respectively.

$$\mathrm{DoG}(x, y) = \frac{A}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{B}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{x^2+y^2}{2\sigma_2^2}} \qquad (8)$$

C. Feature Point Matching

Based on the above operations, the features of two images of the same target in a specific scene are matched to determine the positional relationship.

As for feature point matching, firstly, the detection operator is extracted to find the pixel points that can be easily recognized in the two images. Then the detected feature points are described according to the extracted descriptor. Afterwards, the corresponding relationship is judged by the description operator of each feature point. Finally, noise reduction is carried out to remove the wrong matching points. Figure 3 is the result of feature point matching for the two input images.

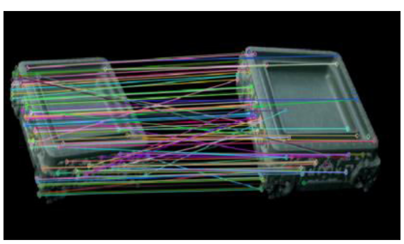

Figure 3. Feature Point Matching

Due to the strong edge response to DoG function, the key points located at the edge are easily affected by noise. If the principal curvature of the extreme value of the Gaussian difference operator of a key point at the transverse edge is obviously smaller than that at the vertical edge, then the key point can be discarded. Let the principal curvature be H, which can be expressed by the Hessian matrix as shown in (9).

$$\mathrm{H} = \begin{bmatrix} Dxx & Dxy \\ Dyx & Dyy \end{bmatrix} \qquad (9)$$

In addition, we should judge whether the key point is at the edge since it may be affected by serious noise. Let A and B be the corresponding eigenvalues of the key points at the transverse edge and the vertical edge respectively. If A is a larger eigenvalue and is r times of B, there are:

$$\mathrm{Tr(H)} = D_{xx} + D_{yy} = \alpha + \beta \qquad (10)$$

$$\mathrm{Det(H)} = D_{xx}D_{yy} - {D_{xy}}^2 = \alpha\beta \qquad (11)$$

$$\mathrm{ratio} = \frac{Tr(H)^2}{Det(H)} = \frac{(r+1)^2}{r} \qquad (12)$$

Set the threshold R. When ratio $\geq \frac{(R+1)^2}{R}$ holds, the key point is considered to be at the edge.

IV. Experiment and Analysis

A. Description of the Experiment

The experiment is divided into two parts. The first part verifies whether the improved edge detection algorithm (I-Canny) can retain more edge information than original Canny. For each target object, Canny and I-Canny are executed to detect the edge information respectively. Then the proportion W of white pixel values in the whole image is calculated. The higher W value, the more edge information will be stored. The second part of the experiment is to combine I-Canny with KSR descriptor for object recognition to observe the effectiveness and stability of the proposed method.

The operating system version in the experiment is Ubuntu 16.04.Relevant software libraries include ORB_SLAM2, OpenCV3.0, and Eigen3.1.

B. Comparison Analysis

As shown in Figure 4, three pictures with noise added were selected as experimental subjects. Image 1 has a pure white background and the edge information about the object is simple. Image 2 has a pure black background and complicated object edge information. The background color of Image 3 is complex, and the edge information about the object is more complex.

Canny and I-Canny are executed for contour extraction of the objects in the 3 images respectively.

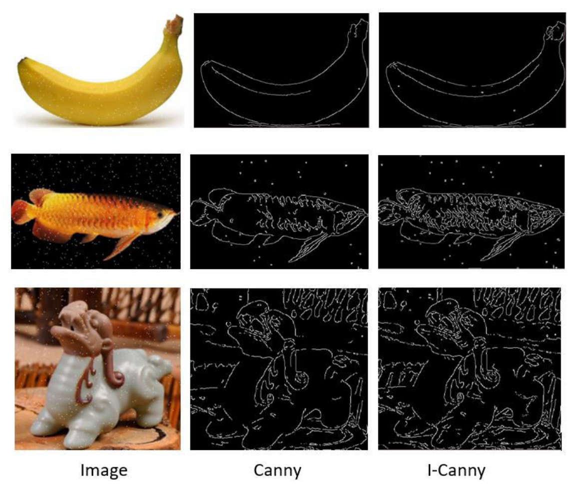

Figure 4. Comparison of Segmentation Effects of the Two Edge Detection Algorithms

As shown in Figure 4, the picture on the left are the subjects, the middle and right are the segmentation effect of Canny and I-Canny respectively. It can be seen that I-Canny performs better on preserving the edge information of the objects than Canny.

Moreover, we further binarize the output from the two algorithms to obtain the proportion W of white pixel points respectively, where W means the edge information obtained after the edge detection algorithm is executed. The higher the value of W is, the more edge information is reserved.

TABLE I. THE PROPORTION OF WHITE PIXEL POINTS

Object	Proportion of Edge Information	
	Canny	***I-Canny***
Image 1	1.585%	2.101%
Image 2	4.286%	6.898%
Image3	6.176%	8.299%

It can be seen from Table I that the proportion of white pixels is increased in the images where I-Canny is used. It also means that I-Canny preserves more edge information than Canny.

C. *Recognition Effect*

Through the combination of I-Canny and KSR descriptor, the code is fused into the AR front-end part of ORB_SLAM2 open source library. Figure 5 shows the AR effect from a monocular camera. The images on the left are 3D objects, and the images on the right are the AR images.

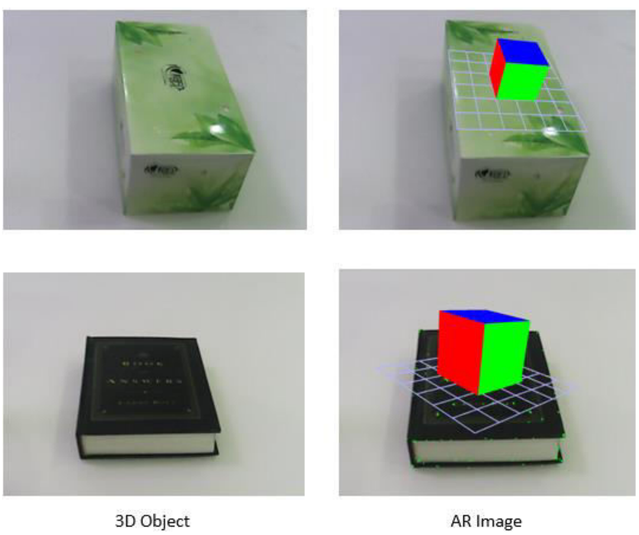

Figure 5. Recognition of 3D Objects

As shown in Figure 5, the proposed method is able to accurately recognized the 3D objects and make the virtual model stably displayed above them.

TABLE II. THE NUMBER OF FEATURE POINTS

Object	Number of Feature Points	
	N1	***N2***
Box 1	126	143
Box 2	151	160

Three-dimensional object detection is performed on the two objects in Figure 4 using and optimized and unoptimized SLAM front-end codes respectively. Let the number of recognized feature points is N_1 and N_2, and the values of N_1 and N_2 are shown in Table II. It can be seen that the optimized SLAM front-end code recognizes more feature points, which shows that the improved algorithm can perform better in recognizing feature points of three-dimensional objects.

V. CONCLUSIONS

In the AR-oriented 3D object recognition process, the traditional Canny edge detection algorithm filter out the important edge information about the object, resulting in the decrease of recognition accuracy. In this paper, an improved edge detection method is proposed. The Gaussian filter in Canny is replaced by the median filter, which reduces the loss of edge information. Moreover, the local object recognition technology is integrated to recognize 3D objects, so that the virtual model can be stably displayed.

REFERENCES

[1] D. Van Krevelen and R. Poelman, “A survey of augmented reality technologies,” applications and limitations. International journal of virtual reality, vol.9, pp. 1-20, 2010.

[2] M. K. Mokhtar, F. Mohamed, M. S. Sunar, A. A. Abd Aziz, M. A. Arshad, and M. K. M. Sidik, “Image Features Detection and Tracking for Image Based Target Augmented Reality Application,” Paper presented at the 2019 IEEE Conference on Graphics and Media (GAME), pp. 38-43, November 2019.

[3] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan , and N. M. Kwok, “A comprehensive performance evaluation of 3D local feature descriptors. International Journal of Computer Vision,” vol.116, pp. 66-89, 2016.

[4] Q. CAI, J. HAO, J. CAO and H.-s. LI, “Salient detection via local and global feature,” Optics Precision Engineering, vol.25, pp. 772-778, 2017.

[5] F. Zhou and F. De la Torre, “Factorized graph matching,” IEEE transactions on pattern analysis machine intelligence, vol.38, pp. 1774-1789, 2015.

[6] M.Wu, and L.Chen, “Image recognition based on deep learning,” Paper presented at the 2015 Chinese Automation Congress (CAC), pp. 542-546,November 2015.

[7] F. Marra, G. Poggi, C. Sansone and L. Verdoliva, “A study of co-occurrence based local features for camera model identification,” Multimedia Tools Applications, vol.76, pp. 4765-4781, 2017.

[8] Y. Lee, K. Lee and S. Pan, “Local and global feature extraction for face recognition,” Paper presented at the International Conference on Audio-and Video-Based Biometric Person Authentication , pp. 219-228,July 2005.

[9] J. Yan, X.-C. Yin, W. Lin, C. Deng, H. Zha and X. Yang, “A short survey of recent advances in graph matching,” Paper presented at the Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, pp. 167-174,June 2016.

[10] Y. Gal, R. Islam and Z. Ghahramani, “Deep bayesian active learning with image data,” Paper presented at the Proceedings of the 34th International Conference on Machine Learning,Vol.70, pp. 1183-1192,August 2017.

[11] A. L. Kabade and V. G. Sangam, “Canny edge detection algorithm,” International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE), Vol.5, pp. 1292-1295, 2016.

[12] N. D. Hoang, and Q. L. Nguyen, "Metaheuristic optimized edge detection for recognition of concrete wall cracks: a comparative study on the performances of roberts, prewitt, canny, and sobel algorithms," Advances in Civil Engineering, PT.10, pp.1-16, November 2018.

[13] Q. Xu, S. Varadarajan, C. Chakrabarti, and L. J. Karam, "A distributed canny edge detector: algorithm and FPGA implementation," IEEE Transactions on Image Processing, Vol.23, pp.2944-2960, 2014.

[14] Bechtel, J. H., Sherman, T. B., & McMillan, D. G. (2018). U.S. Patent No. 10,044,991. Washington, DC: U.S. Patent and Trademark Office.

[15] L. Gun, Z.Ran, and C. Honglei, "A contour detector with improved corner detection," Multimedia Tools and Applications, Vol.76, pp. 5965-5984, 2017.

[16] B. Zhao and J. Xi, "Efficient and accurate 3D modeling based on a novel local feature descriptor," Information Sciences, Vol. 512, pp. 295-314, February 2020.

[17] S. A. A. Shah and M. Bennamoun and F. Boussaid, "Keypoints-based surface representation for 3D modeling and 3D object recognition.," Pattern Recognition, Vol.64, pp. 29-38, 2017

[18] H. A. E. F. El-Sennary, M. E. Hussien and A. E. M. A. Ali, "Edge Detection of an Image Based on Extended Difference of Gaussian.,"American Journal of Computer Science and Technology, Vol. 2, pp. 35-47, 2019.