CONTOUR BASED OBJECTS POSE ESTIMATION USING DIFFERENTIAL EVOLUTION ALGORITHM

Tao Ngoc Linh, Hiroshi Hasegawa

Graduate School of Engineering and Science, Shibaura Institute of Technology

taongoclinh@gmail.com, h-hase@shibaura-it.ac.jp

ABSTRACT - This paper proposes a novel object-tracking algorithm to estimate three dimensions position of an object using a web camera and its given 3D CAD model. The method integrates differential evolution algorithm with an efficient chamfer matching method, which provides an error function for searching algorithm to find the most suitable position of objects in Euclidean space. We also implement parallel framework with multi-core computer processors to accelerate error calculation, our algorithm to track objects in reasonable time.

1. INTRODUCTION

In the last decade, object detection and recognition have gained significant improvement by using keypoint features [1]. Since geometric transformations and illumination changes has no effect on finding keypoints, they have been widely used for matching images from slightly different viewpoints [2]. Keypoint-based approaches work well in textured objects but texture-less objects. Textured objects have various keypoints, those have high potential appearing on both images. After finding keypoints, sample consensus such as RANSAC [3] calculates the most suitable transformation of the object from reference position to current position. The more matched keypoints, the more accurate the transformation is. On texture-less objects lack of keypoint repeatability and stability on textureless regions neither reduces the accuracy of sample consensus method nor leads to wrong results. Like keypoints, edges are also invariant to general geometric transformations and illumination changes [4]. Using edges are more suitable as a general approach even with texture-less objects.

In early computer vision research, to find the best alignment between two edge maps, a given priori set of edge templates compare their suitability to the current edge maps to draw the mot suitable transformation. The current proposed method of chamfer distance matching [5] enhances the cost functions enable for applying global searching algorithm into object tracking problem.

Harris [5] and various proposed edge-based tracking system [6], [7] used edges and contours for visual tracking task. One drawback of using edges is that they are not distinctive enough to provide effective discrimination in complex background or occlusions, there have been efforts to enhance the previous one by unifying interest points or considering multiple but limited hypotheses on edge correspondences. For consideration of multiple hypotheses in a more general sense global searching algorithm should come into consideration.

We propose an approach of using Differential Evolution (DE) [8] as the global searching method to continuously search the 3D position of object in camera coordinate. A parallel calculating of chamfer matching error is considered to increase the speed and get higher accuracy.

2. METHODOLOGY

Initialization is the most important step, a subject of the paper's interest, in tracking algorithm. Following steps presents the implementation pipeline of the initialization:

- Narrow the searching region by using background subtraction
- Searching initial position by using searching algorithm at large boundary.

After initialization, narrower searching boundary is used to get the accurate results at on-line speed. If the cost function goes large, initialization is required.

2.1 Narrow searching region

a) Background subtraction

To gain the first 3D position of objects, the algorithm employs a background subtraction method to narrow the searching area. Recently, new stable background subtraction algorithms are available even in hard condition of illumination changing. As focusing on 3D positioning only, our algorithm merely uses ready-to-use method from OpenCV library [9].

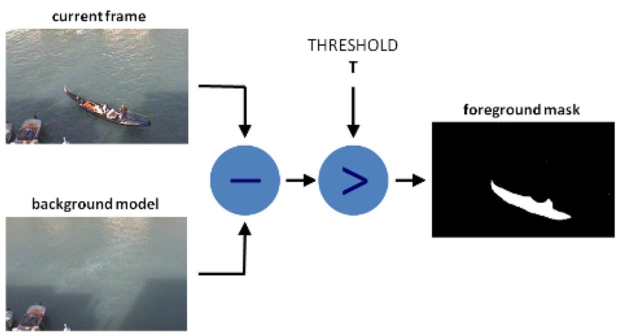

Fig1. Background subtraction mechanism

b) Biggest region searching

After background subtraction, results images have grain points. To get rid of those points, erosion-and-dilation scheme is used. Fig2 show example of the scheme. To gain an even more stable subtraction area, we use the biggest contour, which is a boundary of the object.

Fig2. Erosion and Dilation [10], two morphology transformations

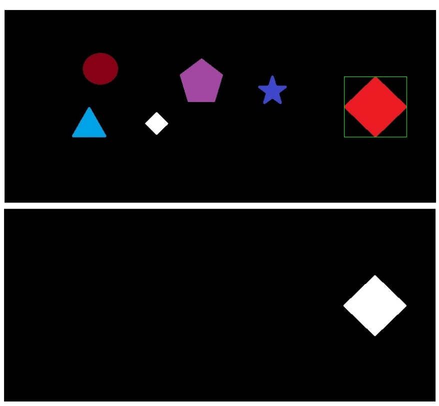

Fig3. Biggest contour finding example [11]

c) Edges image for comparison

The white region as in Fig3 is the input for Canny [12] edge detection method. Canny method includes five different steps:

1. Apply Gaussian filter to smooth the image in order to remove the noise.
2. Find the intensity gradients of the image
3. Apply non-maximum suppression to get rid of spurious response to edge detection
4. Apply double threshold to determine potential edges.
5. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

In the comparison step, points get higher score when they come closer to the edge point. In order to get that score map, we apply a blur filter to the edge maps after Canny edge detection.

d) Camera model and edges from CAD

From a camera with prior-known configuration and object CAD model, we are able to archive ideal visible edges of objects by using camera model matrix. This matrix convert a point with coordinate of (x, y, z) in real coordinate to a image point (u, v) as Equation 1.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x/_z \\ y/_z \\ 1 \end{bmatrix} \quad (1)$$

To determine visibility of object edges we use edge features based method from [13]. Figure 4 shows an edge between adjacent faces $A =< v_0, v_1, v_2 >$ and $A =< v_3, v_1, v_2 >$ with unit face normals.

$$n_A = normalize([v_1 - v_0] \times [v_2 - v_0])$$
$$n_B = normalize([v_3 - v_0] \times [v_1 - v_0])$$

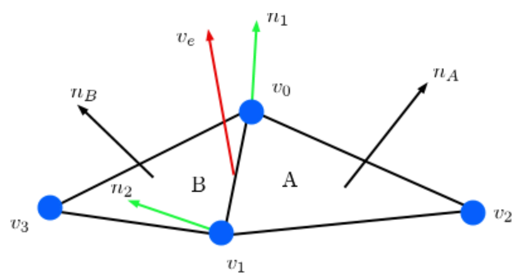

Fig4. Edge of two surfaces

To determine the visibility of edge $E =< v_0, v_1 >$, we uses additional vector v_e with direction from v_0 to the camera center. E is visible edge if cross manipulation value of v_e with n_A or v_e with n_B positive.

2.3 Initial pose searching

2.3.1 Cost function calculation

The cost function for global searching algorithm is a comparison result between edge maps from image and edge maps from CAD model. To gain a equivalent between ideal edge maps, a re-sampling step is applied, so the number of edge points in different ideal maps is set equally at N=200 points.

The error cost function is calculated as in Equation 2,

$$F(R,t) = f(n)\frac{1}{n^2}\sum_1^n (E_i - M_i)^2 \quad (2)$$

$$f(n) = 1 - {}^{n}/_{N} \quad (3)$$

where $f(n)$ is function depended on number of inlier n, E_i is value of real edge images at inlier i, M_i is value of CAD model edge images at inlier i.

By using OpenMP [13], error of each point in N points in is dependently evaluated.

2.3.2 Differential Evolution

Differential evolution (DE), proposed by Storn and Price, is a very popular EA. Like other EAs, DE is a population-based stochastic search technique. It uses mutation, crossover and selection operators at each generation to move its population toward the global optimum minimum.

a) Initialization in DE

The initial population was generated uniformly at random in the range lower boundary (LB) and upper boundary (UB).

$$X_{i,j}^{G=0} = lb_j + rand_j(0,1)(ub_j - lb_j) \quad (4)$$

where $\mathrm{rand}_j(0,1)$ a random number in [0,1].

b) Mutation operation

In this process, DE creates a mutant vector, $X_i^G = (X_{i,1}^G, X_{i,2}^G, \dots, X_{i,n}^G)$. For each individual at each generation X_i^G is a target vector in the current population.

There are several variants of DE based on mutation schemes, which are: DE/rand/1, DE/best/1, DE/current to best/1, DE/rand/2, DE/best/2, DE/rand to best/1.

b) Crossover operation

After mutation process, DE performs a binomial crossover operation on X_i^G and V_i^G to generate a trial vector $U_i^G = (U_{i,1}^G, U_{i,2}^G, \dots, U_{i,n}^G)$ for each particle i as shown as Equation 5,

$$U_i^G = \begin{cases} V_{i,j}^G & \text{if } \mathrm{rand}_j \le CR \text{ or } j = j_{rand} \\ X_{i,j}^G & \text{otherwise} \end{cases} \quad (5)$$

where i=1, …, NP. j=1,…, D is randomly chosen integer in $[1, D]$, $CR \in [0,1]$ is the crossover control parameter.

c) Selection

The selection operator is performed to select the better one between the target vector $X_{i,j}^G$ and the trial vector $U_{i,j}^G$ to enter the next generation.

$$X_i^{G+1} = \begin{cases} U_i^G & if\ f(U_i^G) \le f(X_i^G) \\ X_i^G & otherwise \end{cases} \quad (6)$$

where i=1, …, NP, $X_i^{G+1}$is target vector in the next generation.

3. EXPERIMENT AND RESULTS

3.1 Boundary searching results

To implement the algorithm, the system hardware included a iBuffalo BSW20KKM11BK camera. We used small box as a tracking object. All code is implemented on C++ code on a standard Desktop Computer powered with Intel Core i7-4790 CPU 3.6x8.

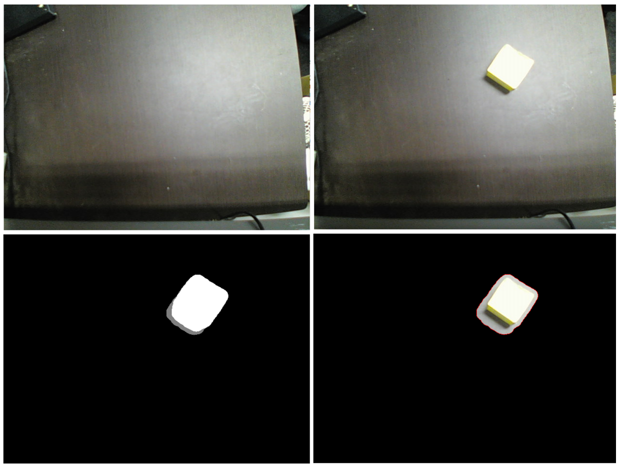

Fig5. Background subtraction result

Fig5 shows results from background subtraction method.

Currently, we used these results for edge detection, yet for limiting the searching boundary. With this shortage, our method still gives promising results. Fig6 shows an example of boundary search of the box. The box boundary is in blue color in the left image.

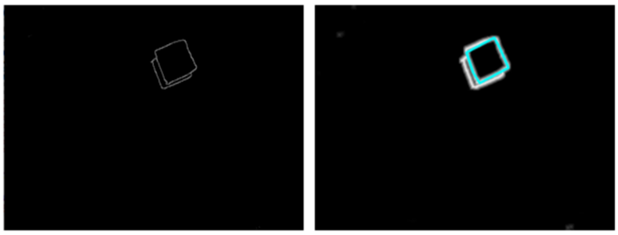

Fig6. Results of tracking

3.2 Runtime.

Without using OpenMP, each runtime is at about 1 second. This is for in initialization task. In normal thinking, with parallel programing, the task is divided and runtime is reduced. However, our experiment showed that, runtime is significantly larger than without parallel programing, at about 2 seconds. This can be explained by the cost of memory exchange between cores are even bigger than computing itself.

DISCUSSION

Object tracking has been always a challenging task in computer vision. Recently, evolution based global searching methods have proved its potential of tackling the tracking problem with ability of finding robust and accurate global optima solutions.

We proposed a novel approach of using differential evolution as a global searching method to find the best 3D position of objects. The experimental results show promising results.

Currently, the algorithm has shortage of not limiting the searching region; the cost function is without interactive of real edge maps back to ideal map.

In the future work, cost function would be improved together with narrower searching area for more accuracy but smaller generation of searching. By doing so, we expect to reduce the runtime but remains the accuracy and robustness.

REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant key- points," IJCV, vol. 60, no. 2, pp. 91–110, 2004.

[2] K. Grauman and T. Darrell, "The pyramid match kernel: Discrimina- tive classification with sets of image features," in ICCV, vol. 2, 2005, pp. 1458–1465 Vol. 2.

[3] M. A. Fischler, R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, Vol 24, pp 381-395, 1981.

[4] Luc Vosters, Caifeng Shan, Tommaso Gritti, Real-time robust background subtraction under rapidly changing illumination conditions, Image and Vision Computing, Volume 30, Issue 12, December 2012, Pages 1004-1015, ISSN 0262-8856, http://dx.doi.org/10.1016/j.imavis.2012.08.017.

[5] H. Barrow, J. Tenenbaum, R. Bolles, and H. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," in IJCAI, 1977, pp. 659–663.

[6] C. Harris, "Tracking with Rigid Objects". MIT Press, 1992.

[7] A. I. Comport, E. Marchand, and F. Chaumette, "Robust model-based tracking for robot vision," in IROS, vol. 1, 2004.

[8] Storn, Rainer "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces", Journal of Global Optimization, doi: 10.1023/A:1008202821328, 2007.

[9] Bradski, G. "opencv_library", Dr. Dobb's Journal of Software Tools, 2000.

[10] Erosion and Dialation with OpenCV, http://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html

[11] Biggest contour extraction with OpenCV, http://harismoonamkunnu.blogspot.jp/2013/06/opencv-find-biggest-contour-using-c.html

[12] Canny method, http://docs.opencv.org/master/da/d22/tutorial_py_canny.html#gsc.tab=0

[13] Morgan McGuire and John F. Hughes, "Hardware-Determined Feature Edges", Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering, 2004, doi: http://doi.acm.org/10.1145/987657.987663

[14] OpenMP parallel computing library, http://openmp.org/wp/

Tao Ngoc Linh received his B.E. (2010) at Hanoi University of Science and Technology, Vietnam and M.E. (2013) from Taiwan University of Science and Technology, Taiwan. He is pursuing Dr. Eng. degree at Shibaura Institute of Technology, Japan. His research interests includes 3D computer vision and intelligent algorithm.

Hiroshi Hasegawa received his B.E. (1992) and M.E. (1994) from Shibaura Institute of Technology, Japan. He received Dr. Eng. (1998) in Mechanical Engineering from Tokyo Institute of Technology, Japan. He has been working at Shibaura Institute of Technology, and currently is a Professor. He is a member of JSEE, JSME, ASME, JSCES and JSST. His research interests include Creativity of Design and Systems Engineering.