

캐니 에지 맵을 LOD로 변환한 맵을 이용하여 객체 소거를 위한 추적

장영대*, 박지현*

*홍익대학교 컴퓨터공학과

e-mail:ferthona@cs.hongik.ac.kr

Object Tracking for Elimination using LOD Edge Maps Generated from Canny Edge Maps

Young-dae Jang*, Jihun Park*

*Dept of Computer Science, Hong-Ik University

ABSTRACT

We propose a simple method for tracking a nonparameterized subject contour in a single video stream with a moving camera and changing background. Then we present a method to eliminate the tracked contour object by replacing with the background scene we get from other frame. Our method consists of two parts: first we track the object using LOD (Level-of-Detail) canny edge maps, then we generate background of each image frame and replace the tracked object in a scene by a background image from other frame that is not occluded by the tracked object. Our tracking method is based on level-of-detail (LOD) modified Canny edge maps and graph-based routing operations on the LOD maps. To reduce side-effects because of irrelevant edges, we start our basic tracking by using strong Canny edges generated from large image intensity gradients of an input image. We get more edge pixels along LOD hierarchy. LOD Canny edge pixels become nodes in routing, and LOD values of adjacent edge pixels determine routing costs between the nodes. We find the best route to follow Canny edge pixels favoring stronger Canny edge pixels. Our accurate tracking is based on reducing effects from irrelevant edges by selecting the stronger edge pixels, thereby relying on the current frame edge pixel as much as possible. This approach is based on computing camera motion. Our experimental results show that our method works nice for moderate camera movement with small object shape changes.

1. Introductions and Related Works

The tracking of moving subjects is a hot issue because of a wide variety of applications in motion capturing for computer animation, video coding, video surveillance, monitoring, and augmented reality. We mean complex because both tracked subject and background scene leave many edges after the edge detection. We assume our subject is never occluded by any background objects, but it occludes other objects in the background. Our background generation assumes all background objects are static. We can classify

the methods of representing a subject contour into two categories depending on the method used, parameterized contour or nonparameterized contour. In tracking a parameterized contour, a subject contour estimating the motion is represented by using parameters.

In the method of tracking a nonparameterized contour, a subject contour as a subject border is represented. The contour created by these algorithms is represented as a set of pixels. Paragios's algorithm[1] and Nguyen's algorithm[2] are popular in these approaches. Recently, Nguyen

proposed a method[2] for tracking a nonparameterized subject contour in a single video stream with a moving camera and a changing background. This paper has no big difference in object tracking of our old work[3] but we extend our research in tracked object elimination. This technique can be basic in editing a movie compared to a popular image editing. Our object elimination by background is basically similar to other works[4].

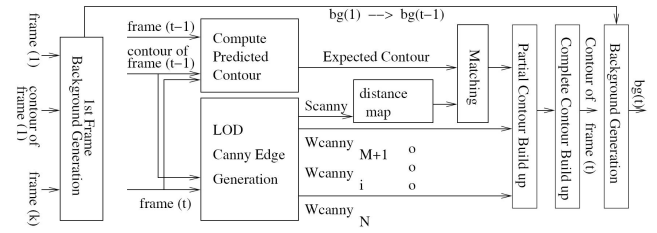
2. Our Approach

We propose a new method to increase the subject tracking accuracy by using LOD Canny edge maps in predicted contour normal direction. We compute a predicted contour as Nguyen does. But, we use two major approaches. First, in order to reduce side-effects caused by irrelevant edges, we generate Canny edge maps around the predicted contour in the contour normal direction. Second, we start our basic tracking contour using *simple (strong)* Canny edges generated from large image intensity gradients.

A *strong Canny edge map* is generated by a pixel-wise union of the simplest Canny edge maps out of various scaled Canny edge maps. Our new method selects only the Canny edges with large image intensity gradient values, *Scanny* edges. Working on Scanny has an effect of background removal. Our accurate tracking is based on reducing the effects from irrelevant edges by only selecting strongest edge pixels, and relying on the current frame edge pixels as much as possible contrary to Nguyen's approach of always combining the previous contour.

We consider *Scanny* edges around a predicted contour, computed from the previous frame contour, to likely be a part of the new contour. To make a closed contour, we do a final routing using the above segments of *partial contours* and *Scanny* edges around the predicted contour. We do a routing between two disconnected *Scanny* edge pixels using LOD *Wcanny* edge maps favoring stronger edge maps.

3. Overview of Our System



(Figure 1) Overview of our single frame tracking and background generation

Figure 1 shows an overview of our system for tracking and eliminating an object (to make a background image) in a single image frame. First, we generate the first frame background scene. Then we compute a tracked object contour for the next frame. As inputs to compute an object contour, we get a previous image frame, denoted as *frame (t-1)* and the corresponding tracked subject contour of input *frame (t-1)*, and a current image frame, denoted as *frame (t)*. From *frame (t-1)*, contour of *frame (t-1)*, and *frame (t)*, we compute a predicted contour, $\partial\Omega^{(p,t)}$, for *frame (t)* using subject motion[2]. Then, we generate various detailed levels of modified Canny edge image maps for the input *frame (t)*. We select *Scanny* edges from the LOD Canny edge maps. From a *Scanny* edge map, we derive a corresponding distance map. Using the predicted contour, the best matching is then found between the predicted contour and the *Scanny* distance map. *Scanny* edge pixels matching with the predicted contour become the frame of the contour build up. We call these pixels *selected Scanny contour pixels*. *Selected Scanny contour pixels*, generated using *Scanny* and predicted contour, are the most reliable (but not closed) contour pixels to start building a closed tracked contour, and are stored in the *selected Scanny found list*. We then route a path to connect adjacent *selected Scanny contour pixels* in the found list. If we finish connecting every adjacent *selected Scanny contour pixel* pair,

we get a set of *partial contours* although not guaranteed to be the *best* closed contour.

4. LOD Canny Edge Maps and Matching for Selecting Reference Contour Pixel

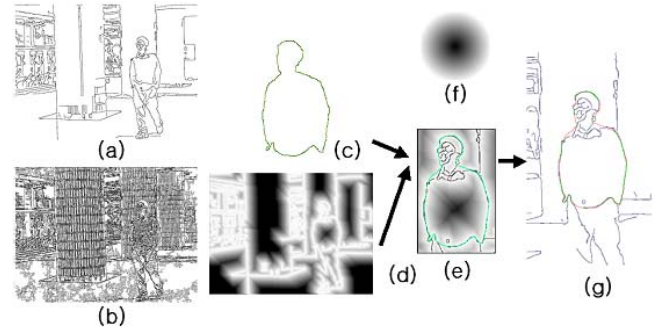
Our new method selects only the Canny edges with large image intensity gradient values, *Scanny* edges. A *Scanny* edge map does not have noisy background edges and looks simple, meaning there are less edges in the Canny edge map of the scene. Working on *Scanny* has an effect of background removal. Our accurate tracking is based on reducing the effects from irrelevant edges by only selecting strongest edge pixels, and relying on the current frame edge pixels as much as possible contrary to Nguyen's approach of always combining the previous contour. For Canny edge maps generated with smaller image intensity gradient values, we call $Wcanny_i$ used in computing *Scanny* edge map. $Wcanny_{M+1}$ has the simplest Canny edges generated from a set of large (*strongest*) intensity gradient value edges. $Wcanny_N$ has the most detailed Canny edges generated by an accumulation from largest (*strongest*) till to the smallest (*weakest*) intensity gradient valued edges.

Let $\Phi_i^{(I,t)}$, where $i=1, \dots, N$, be a totally ordered set of Canny edge maps of an input image *frame* (t). The ordering is done by counting the number of edge pixels. $\Phi_1^{(I,t)}$ has the smallest number of edge pixels while $\Phi_N^{(I,t)}$ has the largest number of edge pixels. Then, we take the top 10 percent to 30 percent of the simple Canny edge maps and union into pixel-level to make a *Scanny* edge map, $S\Phi^{(I,t)}$. M is the total number of Canny edge maps used to make a $S\Phi^{(I,t)}$. The rest of the Canny edge maps are used to generate $Wcanny_i, W\Phi_i^{(I,t)}$.

$$S\Phi^{(I,t)} = \bigcup_{i=1}^M \Phi_i^{(I,t)}$$

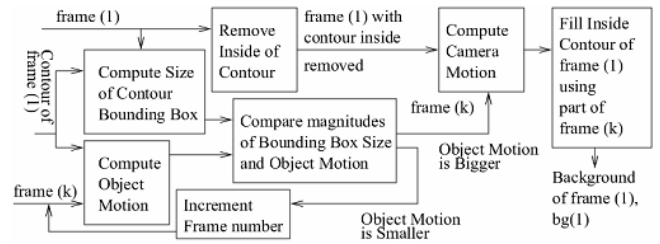
$$W\Phi_i^{(I,t)} = S\Phi^{(I,t)} \cup \left(\bigcup_{j=M+1}^i \Phi_j^{(I,t)} \right), i = (M+1), \dots, N$$

where \cup is pixel-wise union of Images.



(Figure 2) Scanny edge map (a), LOD edge map (b), predicted contour from (c), distance map generated from Scanny (d), matching between predicted contour and Scanny distance map (e), circular distance map used in matching (f), final routing result favoring *Scanny* (g)

5. Object Elimination & Background Generation

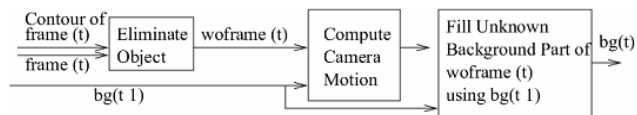


(Figure 3) Process of generating the first frame background

Figure 3 shows a process to determine the first frame background given a sequence of video stream. The process to determine the exact frame to fill occluded part is as follows. First we try with an arbitrary frame, say *frame* (k). In order to verify that the frame actually contains the missing background part of the first image frame, we compute object motion between *frame* (1) and *frame* (k) [2]. If the object motion magnitudes in both x and y direction are bigger than the width and height of the bounding box of the first frame respectively, we are done in finding the exact frame to fill the missing part of *frame* (1). Otherwise we try with the next image frame until the exact frame is found. Then

we compute *camera motion* between the first frame and the k th frame, and the computation result is used in generating a background image of the first frame denoted as $bg(1)$. To fill the occluded part of $woframe(1)$, we use computed *camera motion* and take corresponding image part from $frame(k)$. Consequently, we get the background image of the first frame.

Figure 4 shows a process to determine the t th frame background. Using $woframe(t)$ and $bg(t-1)$, we compute the *camera motion* between the $frame(t-1)$ and the $frame(t)$. Using the computed *camera motion*, we fill the occluded part of $woframe(t)$ using $bg(t-1)$. As a result, we get the background image for $frame(t)$.



(Figure 4) Process of generating the t th frame background using $(t-1)$ th frame background

4. Conclusion

We start by selecting a boundary edge pixel from the simple (strong) Canny edge map, referring to the most detailed edge map to get edge information along the LOD Canny edge maps. Our basic tracking frame is determined from the strong Canny edge map, and the missing edges are filled by the detailed Canny edges along the LOD hierarchy. Even though detailed Canny edges are noisy, our basic tracking frame is determined from the Scanny, and is not disturbed by noisy edges. This has an effect of Nguyen's background noisy edge removal. Another major contribution of our work is not accumulating tracking errors.

A new contour is determined by mixing the current image edge map with the previous contour. If there is no edge present, we may have a tracking error for the part. Whenever we get Scanny edge information, the tracking error

disappears, and we can restart accurate tracking for the erroneous part. By using our novel method, our computation is not bothered by noisy edges resulting in a robust tracking. Our experimental results (Figure 5) show that our tracking approach is reliable enough to handle a sudden change of the tracked subject shape in a complex scene.



(Figure 5) The result of generating background of the first frame

Reference

- [1] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.22, pp. 266-280, 2000.
- [2] H. T. Nguyen, M. Worring, R. van den Boomgaard, and A. W. M. Smeulders, "Tracking nonparameterized object contours in video," IEEE Trans. on Image Processing, Vol.11, pp. 1081-1091, September 2002.
- [3] M. Roh, T. Kim, J. Park, and S. Lee, "Accurate Object Contour Tracking Based on Boundary Edge Selection," Pattern Recognition, Vol. 40, No. 3, pp. 931-943, March 2007.
- [4] A. Bartoli, N. Dalal, B. Bose and R. Horaud, "From Video Sequences to Motion Panoramas," Proceedings of Workshop on Motion and Video Computing, pp. 1-7, 2002.

This work was supported by 2006 Hongik University Research Fund.