

A Study on Object Contour Tracking with Large Motion Using Optical Flow and Active Contour Model

Jin-Woo Choi, Taek-Keun Whangbo and Nak-Bin Kim

Abstract In this study, an object contour tracking method is proposed for an object with large motion and irregular shapes in video sequences. To track object contour accurately, an active contour model was used, and the initial snake point of the next frame is set by calculating an optical flow of feature points with changing curvature in the object contour tracked from the previous frame. Here, any misled optical flow due to irregular changes in shapes or fast motion was filtered by producing an edge map different from the previous frame, and as a solution to the energy shortage of objects with complex contour, snake points were added according to partial curvature for better performance. Findings from experiments with real video sequences showed that the contour of an object with large motion and irregular shapes was extracted precisely.

Keywords Object contour tracking · Optical flow · Active contour model

1 Introduction

With the growing interest in the 3D videos, displays such as 3D TV and broadcasting technologies that make it possible to watch 3D videos have been developed and gaining popularity in a quick pace recently. However, despite ever-growing

J.-W. Choi (✉)
CT Research Institute, Seongnam-si,
Republic of Korea
e-mail: cjw49@paran.com

T.-K. Whangbo · N.-B. Kim
Department of Interactive Media, Gachon University, Seongnam-si,
Gyeonggi-do, Republic of Korea
e-mail: tkwhangbo@gachon.ac.kr

N.-B. Kim
e-mail: nabkim@gachon.ac.kr

demands for the 3D contents, available 3D contents are very rare due to limited production time and money [1, 2]. The 2D-to-3D technology that converts existing 2D images to 3D videos has been drawing attention lately as a solution to this. Although there are a number of ways of realizing such a technology, generally, 2D-to-3D conversion works in the following manner [1].

- Object Extraction or Segmentation
- Depth Map Assignment or Generation
- Rendering or Occlusion
- Re-touch

In this technology, conversion is conducted by means of (1) manual conversion, (2) non real-time software automatic conversion, and (3) real-time hardware automatic conversion, depending on the use of contents, costs, and the quality of videos [1]. It usually takes six months and 300 staffers to carry out manual conversion to attain the level of movie-like quality, and a hybrid conversion technology that combines software conversion based on an image-processing function and manual work has been emerging to reduce manual work necessary for video conversion and meet quality requirements [1]. To produce quality scene-based video sequences that are properly classified, the operator intervenes to deliberately extract individual objects and subtract the background for the first frame before producing depth map through geometric analysis. The next frames are converted automatically within a reasonable time on the basis of data used for the first frame. Here, it is highly important for the operator to correctly track the contour of an object subtracted from the first frame in subsequent frames, and the more irregular the shape of the object is and the larger its movement becomes, the harder tracking gets.

In this study, we propose a method of tracking contour of an object with larger motion and irregular shapes in video sequences for the 2D-to-3D conversion in a stable manner. Regarding this issue, we explain existing object contour tracking technologies in [Chap. 2](#) and how to track the contour of objects with larger motion using the proposed optical flow and active contour model in [Chap. 3](#). In [Chap. 4](#), we assess the performance of the proposed method through experiments, and [Chap. 5](#) discusses the conclusion.

2 Existing Object Contour Tracking Technologies

Object tracking methods of video sequences are classified largely into the method of using modified object extracting algorithm of the past and the one of using object tracking algorithm [3]. Technologies to apply the object extraction algorithm to video sequences include the method of forming an energy function considering the relationship between time-dimensional nodes together with background subtraction and graph-cut algorithm [4, 5]. These methods often cause errors in the adjacent areas with complex background. In object tracking algorithm, features such as point, kernel, and silhouette are extracted from the previous

frames, with the object extracted from the following frame by matching them [6]. Designed mainly to situate an object, this method needs additional algorithms to ensure accurate extraction of the shape of an object and has disadvantages such as occlusion of the object or vulnerability for objects with large motion [3].

In recent years, extensive studies have been made on an active contour model-based method that effectively captures the modification of an object and divides in flexible lines surrounding the object [7–11], a method to find the shape of an object based on its contour energy data. In this study, the outermost closed curve of an irregular object extracted from the first frame of video sequences by the operator are being tracked continuously in the following frames using the active contour model.

2.1 Snake Algorithm

The snake algorithm, as the most common active contour model, was first proposed in 1987 by Kass [7]. In this algorithm, initial snake points are set around the object to extract from the input video, and the contour of the object is extracted in the course of minimizing the energy function by moving snake points iteratively according to the defined energy function.

Snake energy function is the sum of internal energy determining the shape of snake contour and external energy that serves to pulling snake points toward the object contour.

$$E_{\text{snake}}(v) = \sum_{i=0}^{N-1} (E_{\text{internal}}(v_i) + E_{\text{external}}(v_i)) \quad (1)$$

where snake point is set as v_i , v_i is $v_i = (x_i, y_i)$, which means the i th snake point, and N refers to the number of snake points.

Internal energy consists of continuity energy that determines the distance between snake points and curvature energy that determines the movement of snake points.

$$E_{\text{internal}}(v_i) = E_{\text{continuity}}(v_i) + E_{\text{curvature}}(v_i) \quad (2)$$

External energy functions to attract snake contour to features or contour of an object. Features of frequently used videos include gradient meaning the boundary of the object where brightness changes sharply. External energy at individual points of 9-neighbor including the current control point is calculated to move to the location with a larger gradient.

$$E_{\text{external}}(v_i) = \frac{-|\nabla f(v_i)|}{e_{\text{max}}} \quad (3)$$

2.2 Problems with Snake Algorithm

Since it can extract the object contour in a simple and effective manner, snake algorithm is being used broadly and has following problems although there have been many minimization algorithms proposed [12]. The algorithm

- (1) is highly dependent on the location and shape of initial snake points.
- (2) cannot extract the contour of an object with complex shape by means of functions of internal energy function.
- (3) takes enormous time due to the limited range where fixed points move at a time.

As this study aims at applying to the non real-time 2D-to-3D conversion, the problem as in (3) is not taken into account seriously. By calculating optical flow with the previous frame and setting the location of initial snake points for object tracking of the current frame, the problem of dependence on the location and shape of initial snake points as mentioned in (1) is solved. As far as the problem described in (2) is concerned, object contour, no matter how complex, is effectively extracted by inserting new snake points after calculating partial curvature between snake points.

3 The Proposed Method

This study proposes a new tracking method that follows in order to solve the problem of object tracking using existing snake algorithm. The proposed method is designed to track exactly object contour in the $n + 1$ th frame from object extraction result data in the n th frame. In the $n + 1$ th frame, optical flow of object contour feature points in the n th frame is calculated to set the initial snake points. In this case, any optical flow as a result of wrong algorithm due to irregular object or large motion is filtered, compared to the result of morphology algorithm of difference edge map between two frames. Afterwards, activation contour is converged to the target object in the $n + 1$ th frame in the course of finding activation contour solution in the snake algorithm. To solve the problem of energy shortage caused by complex contour of the object to track, we took into consideration of the method of adding snake points using partial curvature of Lee [13] (Fig. 1).

3.1 Calculation of Optical Flow

In the case of non real-time 2D-to-3D conversion, objects are classified generally by the operator for the first frame ($n = 0$) in video sequences of the scene to convert. From the object contour classified in this way, end points of horizontal, vertical, and diagonal components are set as feature points of object contour.

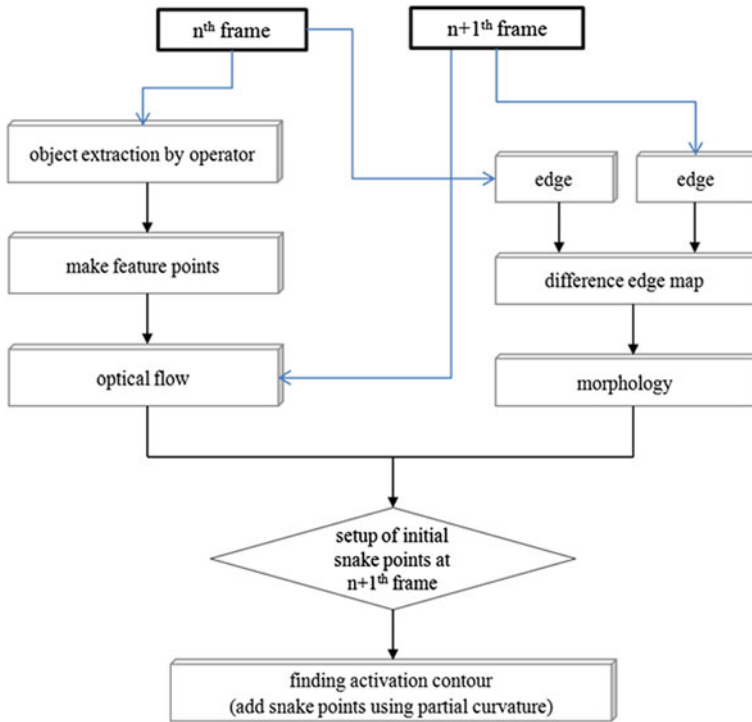


Fig. 1 Flow of the proposed object contour tracking

Afterwards, referring to the $n + 1$ th frame, optical flow off feature points is calculated. Optical flow means the motions of individual pixels created by 3D movement of the object in the video or the camera represented by vector field [14]. Among many methods of tracking the motion of pixels, the Lucas-Kanade [15] algorithm, most widely known, was used in this study. The Lucas-Kanade algorithm is based on three hypotheses: brightness constancy, temporal persistence, and spatial coherence. Brightness constancy means that brightness values among video frames never change, while temporal persistence means that compared to motion of an object in a video, time change faster, causing less motions of the object between frames. Under these two hypotheses, the optical flow between time t and $t + \Delta t$ is expressed as follows:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum I_{x_i}^2 & \sum I_{x_i} I_{y_i} \\ \sum I_{x_i} I_{y_i} & \sum I_{y_i}^2 \end{bmatrix} \begin{bmatrix} \sum I_{x_i} I_t \\ \sum I_{y_i} I_t \end{bmatrix} \quad (4)$$

V_x and V_y in Eq. (4) refer to speed component of each axle, and I to brightness degree of each pixel. Figure 2 shows the results of motion of feature points following the calculation of optical flow with the $n + 1$ th frame, once feature points are created along the object contour extracted by the operator from the n th frame.



Fig. 2 Creation of feature points and calculation of optical flow from object contour. **a** Object extracted from the n th frame by the operator, **b** Feature points created from the object contour in **(a)**, **c** Results of tracking feature points of **(b)** along optical flow in the $n + 1$ th frame

3.2 Setup of Initial Snake Points

The 15frame/s videos were used for this study and have tracking points where as a result of creating optical flow, quickly moving irregular objects are unable to move precisely to the object contour as shown in Fig. 2. Therefore, these tracking points cannot be used for snake algorithm as they are due to higher dependence on the location and shape of initial snake points set in 2.2, so it is impossible to track object contour properly. To solve this problem, a difference edge map between frames is created for filtering purposes. To make a difference edge map, edge maps of the n th frame and the $n + 1$ th frame are made first of all. To reduce the impacts of surrounding noises and obtain thick gradient data, the Sobel operator is used for an edge map. The difference edge map $D_{\text{edge}}(x, y)$ for the two frames is defined as follows:

$$D_{\text{edge}}(x, y) = \begin{cases} S_{n+1}(x, y) - S_n(x, y) & \text{if } S_{n+1}(x, y) - S_n(x, y) > I_{th} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $S_n(x, y)$ and $S_{n+1}(x, y)$ refer to edge extraction result of using the Sobel operator of the n th frame and the $n + 1$ th frame, respectively, and 10 is used for I_{th} as a threshold to prevent the mixture of noises. Figure 3 is the result of creating difference edge map. Figure 3a, b show the results of the Sobel edge extraction for the n th frame and the $n + 1$ th frame, respectively, and Fig. 3c is the result of creating difference edge map. Fixed background and still object in a scene where the camera does not move are set to values close to 0. Then, considering the scope of filtering, dilation algorithms as a morphological algorithm are performed many times to obtain a map for final snake point filtering as in Fig. 3d.

Using the difference edge map obtain in this way, the feature points produced through optical flow in 3.1 are filtered. Specifically, any optical flow surpassing a set value (here 5 set from many experiments) becomes subject to filtering, while the one found not included in the value as a result of comparison with the difference edge map is removed. Figure 4 shows that almost all of the feature points not included in the object contour have been removed.

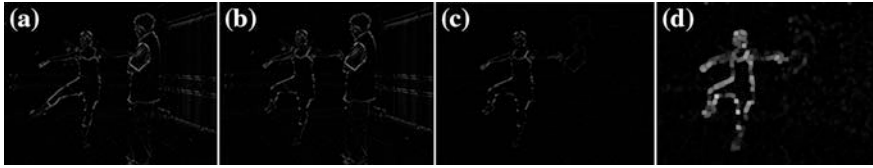


Fig. 3 Creation of difference edge map to set reference snake points. **a** Result of edge extraction in the n th frame, **b** Result of edge extraction of the $n + 1$ th frame, **c** Difference edge map, **d** Dilation for (c)

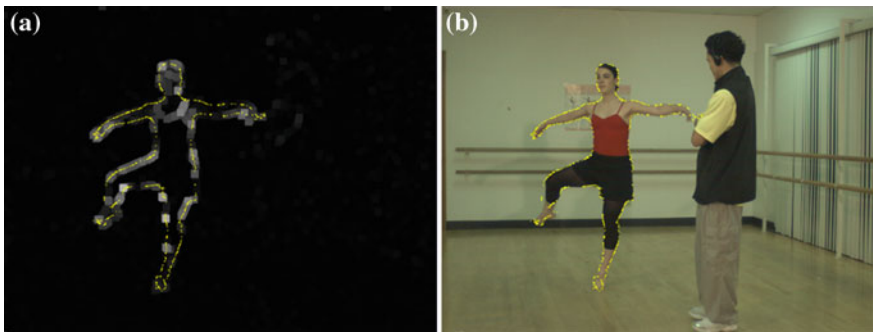


Fig. 4 Filtering of feature points according to difference edge map. **a** Filtering method of feature points, **b** Filtering results

3.3 Finding Activation Contour Solution

Feature points filtered by difference edge map are set as reference snake points in the $n + 1$ th frame and converged to object contour with gradient of grey image as external energy. For an object with irregular shape, we considered a method of adding snake points using partial curvature of Lee [13] in the stage of renewing individual snake points in order to detect the contour in the areas where the curvature among and between snake points changes severely. Discrete curvature k_d is calculated using three snake points v_{i-1}, v_i, v_{i+1} as follows:

$$\vec{T}_{d1} = v_{i-1} - v_i, \quad \vec{T}_{d2} = v_{i+1} - v_i \tag{6}$$

$$\cos \theta = \frac{\vec{T}_{d1} \bullet \vec{T}_{d2}}{\|\vec{T}_{d1}\| \cdot \|\vec{T}_{d2}\|} \tag{7}$$

$$k_d = \frac{2 \sin \theta}{d}, \text{ where } d = v_{i+1} - v_{i-1} \tag{8}$$

In the above equation, \bullet means inner area of two vectors, and $\| \cdot \|$ means the norm of vector. Since the curvature value more than the critical value means complex object contour, two new snake points are inserted into $v_{i-1} + v_i/2$ and $v_i + v_{i+1}/2$.

4 Results and Discussion

To test the proposed method, a ballet video (1024*768, 15frame/s) [16] and a foreman video (352*288, 50frame/s) containing irregularly changing object were used. To assess the performance, mean error of actual contour point coordinates and estimated contour point coordinates was calculated. Actual contour point coordinates were obtained from the object contour extracted manually.

Figure 5 shows the experiment of the foreman video with a sufficient number of frames and smaller motion of the object. Tracking was conducted in the 159th frame for the object manually extracted from the 158th frame. Weights for continuity energy, curvature energy, and external energy of the snake algorithm were set to 0.2, 0.3, and 0.5, respectively, through many experiments, and convergence condition was that the rate of change of the snake points was below 10 %. The mean error was 2.23 pixels, almost similar to the actual object contour.

Figure 6 indicates the experiment of the ballet video with less frames and larger motion. Tracking was performed in the 84th frame for the object manually extracted from the 83th frame. Weights of continuity energy, curvature energy,



Fig. 5 Results of experiments with the foreman video. **a** The 158th frame, **b** Reference snake points estimated from the 159th frame, **c** Result of activation contour tracking of the 159th frame



Fig. 6 Results of experiments with the ballet video. **a** The 83th frame, **b** Reference snake points estimated from the 84th frame, **c** Results of activation contour tracking of the 84th frame

and external energy of the snake algorithm are 0.2, 0.15, and 0.65, respectively. The mean error was 5.61 pixels, almost similar to the actual object contour. Figure 6c reveals that some contour points near the left hand and foot are not converged properly to the object. Motion blur is considered to cause the object contour blurred near the left hand, while external energy shortage due to highly similar brightness of the object and the background seemed to cause the problem near the left foot. Nevertheless, most are found to have been tracked as the boundary of the object.

5 Conclusion

In this study, we proposed a method of tracking object contour with large motion and irregular shape in video sequences in a stable manner using the optical flow and active contour model. Setting optical flow of feature points along the object contour extracted by the operator from video sequences as snake reference for the next frame, activation contour tracking was conducted with addition of snake points by partial curvature. Findings from experiments with actual videos indicate that irregular objects with large motion are easier to track. To apply this method to a non real-time 2D-to-3D conversion where precise tracking of object is important, however, research is necessary to solve the problem of motion blur or energy shortage due to smaller brightness difference from the background.

Acknowledgments This research is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research and Development Program [R2012030006].

References

1. Lee YS (2011) The trends and prospects of 2D-3D conversion technology. *J. Korean Inst Electron Eng* 38(2):37–43
2. Okino T, Murata G, Taima K, Inuma T, Oketani K (1996) New television with 2D/3D image conversion technologies. *Proc SPIE* 2653:96–103
3. Kim J, Lee J, Kim C (2011) Video object extraction using contour information. *J Korean Inst Electron Eng* 48(1):33–45
4. Li Y, Sun J, Shum HY (2005) Video object cut and paste. *J ACM Trans Graph* 24(3):595–600
5. Li B, Yuan B, Sun Y (2006) Moving object segmentation using dynamic 3D graph cuts and GMM. *IEEE Int Conf Signal Process* 2:16–20
6. Javed O, Rasheed Z, Shafique K, Shah M (2003) Tracking across multiple cameras with disjoint views. *IEEE Int Conf Comp Vis* 2:952–957
7. Kass M, Witkin A, Terzopoulos D (1988) Snakes - active contour models. *Int J Comp Vis* 1(4):321–331
8. Bing X, Wei Y, Charoensak C (2004) Face contour tracking in video using active contour model. *Int Conf Image Process* 2:1024–1024

9. Chenyang X, Prince JL (1997) Gradient vector flow: a new external force for snakes. In: IEEE computer society conference on computer vision and pattern recognition, pp 66–71
10. Leymarie F, Levince MD (1993) Tracking deformable object in the plane using an active contour model. *IEEE Trans Pattern Anal Mach Intell* 15(6):617–634
11. Ling P, Fan J, Shen C (2007) Color image segmentation for objects of interest with modified geodesic active contour method. *J Math Imaging Vis* 27(1):51–57
12. Kim D, Lee D, Paik J (2007) Combined active contour model and motion estimation for real-time object tracking. *J Inst Electron Eng Korea* 44(5):64–72
13. Lee JH (2009) A study on an improved object detection and contour tracking algorithm based on local curvature. Master's Thesis, Paichai University, Daejeon, Korea
14. Lee JW, You S, Neumann U (2000) Large motion estimation for omnidirectional vision. In: *Proceedings of IEEE Workshop on Omnidirectional Vision*, pp 161–168
15. Lucas BD, Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: *Proceedings of 1981 DARPA Imaging Understanding Workshop*, pp 121–130
16. Zitnick CL, Kang SB, Uyttendaele M, Winder S, Szeliski R (2004) High-quality video view interpolation using a layered representation. In: *Proceedings of ACM SIGGRAPH and ACM transaction on graphics*, Los Angeles, CA, pp 600–608