A contour tracking method of large motion object using optical flow and active contour model

Jin Woo Choi · Taeg Keun Whangbo · Cheong Ghil Kim

Abstract In this study, an object contour tracking method is proposed for an object with large motion and irregular shape in image sequence. To track object contour accurately, an active contour model was used, and the initial snake point of the next frame is set by defining feature points with changing curvature in the object tracked from the previous frame and calculating an optical flow at the location. Here, any misled optical flow due to irregular changes in shape or fast motion was filtered by producing a difference edge map from the previous frame, and as a solution to the energy shortage of objects with complex contour, a method of adding snake points by partial curvature was applied. Findings from experiments with real image sequence showed that the contour of an object with large motion and irregular shapes was extracted in a relatively precise way.

Keywords Object contour tracking · Optical flow · Active contour model · 2D-to-3D

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 demands for the 3D contents, available 3D contents are very rare due to limited production time and money [6, 12]. The 2D-to-3D technology that converts existing 2D images to 3D videos has been

J. W. Choi (✉)
Culture Technology Institute, Gachon University, San 65, Bokjeong-dong, Sujeong-gu, Seongnam-city, Gyeonggi, South Korea
e-mail: cjw49@paran.com

T. K. Whangbo
Department of Interactive Media, Gachon University, Seongnam, South Korea
e-mail: tkwhangbo@gachon.ac.kr

C. G. Kim
Department of Computer Science, Namseoul University, 21 Maeju-ri, Seonghwan-eub, Seobuk-gu, Cheonan-city, Choongnam, South Korea
e-mail: cgkim@nsu.ac.kr

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 [6].

- 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 [6]. It usually takes 6 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 image-processing and manual work has been emerging to reduce manual work necessary for video conversion and meet quality requirements [6]. To produce quality scene-based image sequence 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 image sequence for the 2D-to-3D conversion in a stable manner. Regarding this issue, we explain snake algorithm as related works in Section 2 and how to track the contour of objects using the proposed optical flow and active contour model in Section 3. In Section 4, we assess the performance of the proposed method through experiments, and Section 5 discusses the conclusion.

2 Related works

Precise tracking of an object in image sequence has been viewed as one of the challenges for researchers partly because of complex background and vague difference between the object and the background. There are two different methods of tracking an object in image sequence: one is existing object extraction algorithm with some changes, and the other is an object tracking algorithm [3].

Technologies to apply the object extraction algorithm to image sequence include the method of forming an energy function considering the relationship between time-dimensional nodes together with background subtraction and graph-cut algorithm [9, 10]. 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 [1]. 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 [2, 8, 13–15]. In this method, the contour of an object of the next frame is tracked based on contour energy information of the object from the results of extraction of the previous frame. In this study, we propose a method of continuously tracking the outermost

closed curve of a non-rigid object extracted from the first frame of image sequence by the operator in subsequent frames by 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 [2]. 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.

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

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.

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

Continuity energy $E_{continuity}(v_i)$, meaning the energy making the distance between snake points the same, normalizes the difference between mean distance of the entire snake points $\overline{d}$ and the distance of two snake points by dividing mean distance $\overline{d}$.

$$E_{continuity}(v_i) \approx \frac{\left| \overline{d} - |v_{i+1} - v_i| \right|}{\overline{d}} \tag{3}$$

$$\overline{d} = \frac{1}{N} \sum_{i=0}^{N-1} |v_{i+1} - v_i| \tag{4}$$

Curvature energy $E_{curvature}(v_i)$ pulls snake contour toward the side with smaller rate of change, causing it to move toward the object contour. The difference of vector addition of previous point of the current snake point (v_{i-1}) and the next point (v_{i+1}) is normalized to the value with the highest rate of change in the search area (c_{max}).

$$E_{curvature}(v_i) \approx \frac{|v_{i-1} - 2v_i + v_{i+1}|}{c_{max}} \tag{5}$$

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.

$$E_{external}(v_i) = \frac{-|\nabla f(v_i)|}{e_{max}} \tag{6}$$

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 [4]. The algorithm:

(1)   is highly dependent on the location and shape of initial snake points.
(2)   is unable to 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 in order to solve the problem with object tracking using existing snake algorithm. Figure 1 shows how the proposed method works. The goal is to track exactly contour of an object in the n+1th frame from object extraction data in the nth frame.

First of all, in the n+1th frame, optical flow of object contour feature points in the nth 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+1th 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 [5].

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 image sequence 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.

Afterwards, referring to the n+1th frame, optical flow of 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 [7]. Among many methods of tracking the motion of pixels, the Lucas-Kanade [11] 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:
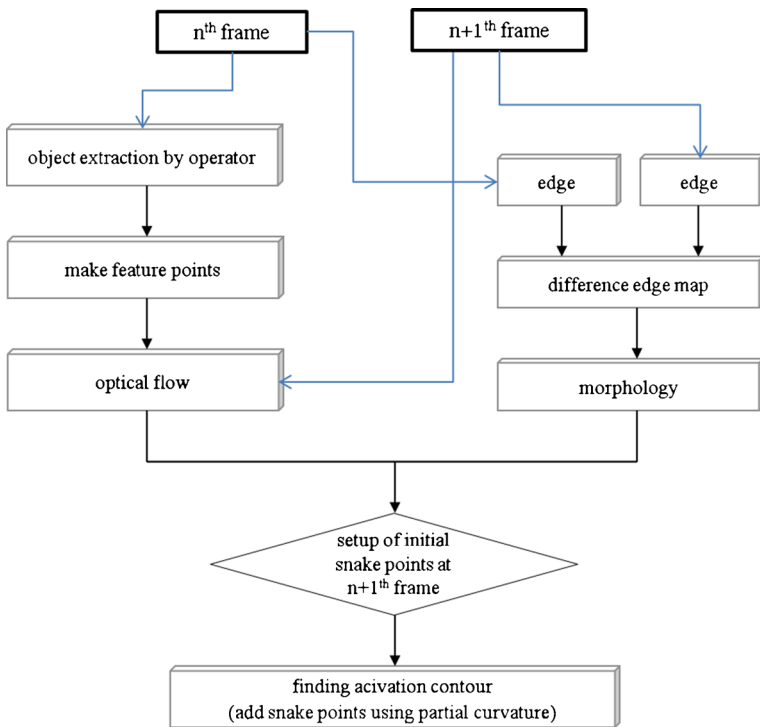
Fig. 1 Flow of the proposed object contour tracking

$$f(x,t) \equiv I(x(t),t) \;=\; I(x(t+dx), t+dt) \tag{7}$$

Equation (7) can be put again as in Eq. (8) by adding the partial differential chain rule and speed component of axle x and axle y:

$$
\begin{aligned}
I_{x_1} V_x + I_{y_1} V_y &= -I_{t_1} \\
I_{x_2} V_x + I_{y_2} V_y &= -I_{t_2} \\
&\;\;\vdots \\
I_{x_n} V_x + I_{y_n} V_y &= -I_{t_n}
\end{aligned}
\tag{8}
$$

V_x and V_y in Eq. (8) refer to speed component of each axle, and I to brightness degree of each pixel. Lastly, spatial coherence means that adjacent points in a space are more likely to be included in the same object and show the same motion. From this, Eq. (8) can be converted into Eq. (9):

$$
\begin{bmatrix}
I_{x_1} & I_{y_1} \\
I_{x_2} & I_{y_2} \\
\vdots & \vdots \\
I_{x_n} & I_{y_n}
\end{bmatrix}
\begin{bmatrix}
V_x \\
V_y
\end{bmatrix}
=
\begin{bmatrix}
-I_{t_1} \\
-I_{t_2} \\
\vdots \\
-I_{t_n}
\end{bmatrix}
\tag{9}
$$

By applying the method of least square, Eq. (9) is converted to Eq. (10) that produces speed component of each axle (V_x, V_y):

$$
\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum I_{x_i}^2 & \sum I_{x_i} I_{y_i} \\ \sum I_{x_i} I_{y_i} & \sum I_{y_i}^2 \end{bmatrix} \begin{bmatrix} \sum I_{x_i} I_{t_i} \\ \sum I_{y_i} I_{t_i} \end{bmatrix} \tag{10}
$$

Figure 2 shows the results of motion of feature points following the calculation of optical flow with the n+1th frame, once feature points are created along the object contour extracted by the operator from the nth frame.

3.2 Setup of initial snake points

The 15frame/s images 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 nth frame and the n+1th 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 $D_{edge}(x,y)$ for the two frames is defined as follows:

(a) Object extracted from the nth 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+1th frame

Fig. 2 Creation of feature points and calculation of optical flow from object contour

$$D_{edge}(x,y) = \begin{cases} S_{n+1}(x,y) - S_n(x,y) & if \quad S_{n+1}(x,y) - S_n(x,y) > I_{th} \\ 0 & otherwise \end{cases} \tag{11}$$

Where $S_n(x,y)$ and $S_{n+1}(x,y)$ refer to edge extraction result of using the Sobel operator of the nth frame and the n+1th 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 3(a) and (b) show the results of the Sobel edge extraction for the nth frame and the n+1th frame, respectively, and Fig. 3(c) 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. 3(d).

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.
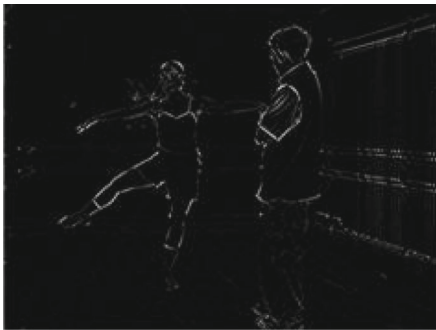
(a) Result of edge extraction in the nth frame
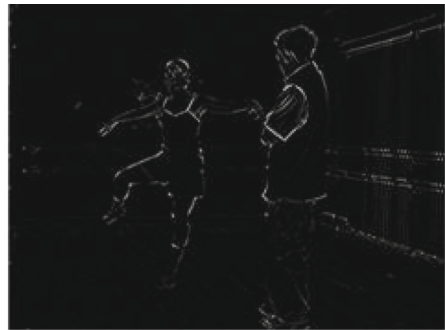
(b) Result of edge extraction of the n+1th frame

(c) Difference edge map

(d) Dilation for (c)

Fig. 3 Creation of difference edge map to set reference snake points

(a) Filtering method of feature points   (b) Filtering results

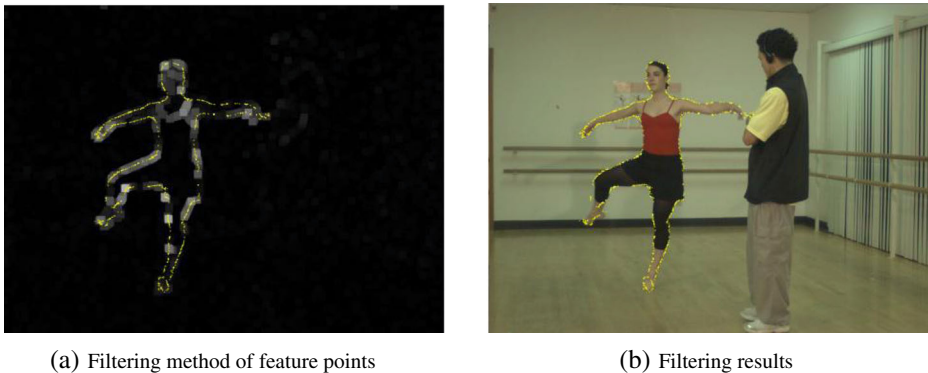**Fig. 4** Filtering of feature points according to difference edge map

3.3 Finding activation contour solution

Feature points filtered by difference edge map are set as reference snake points in the n+1th 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 [5] 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 greatly . Discrete curvature k_d is calculated using three snake points v_{i-1}, v_i, v_{i+1} as follows:

$$\overrightarrow{T_{d1}} = v_{i-1} - v_i \quad , \quad \overrightarrow{T_{d2}} = v_{i+1} - v_i \tag{12}$$

$$\cos\theta = \frac{\overrightarrow{T_{d1}} \bullet \overrightarrow{T_{d2}}}{\left\|\overrightarrow{T_{d1}}\right\| \cdot \left\|\overrightarrow{T_{d2}}\right\|} \tag{13}$$

$$k_d = \frac{2\sin\theta}{d} \quad \text{where} \quad d = v_{i+1} - v_{i-1} \tag{14}$$

In the above equation, • means inner area of two vectors, and ‖‖ 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 as criteria for assessment were obtained from the object contour extracted manually.
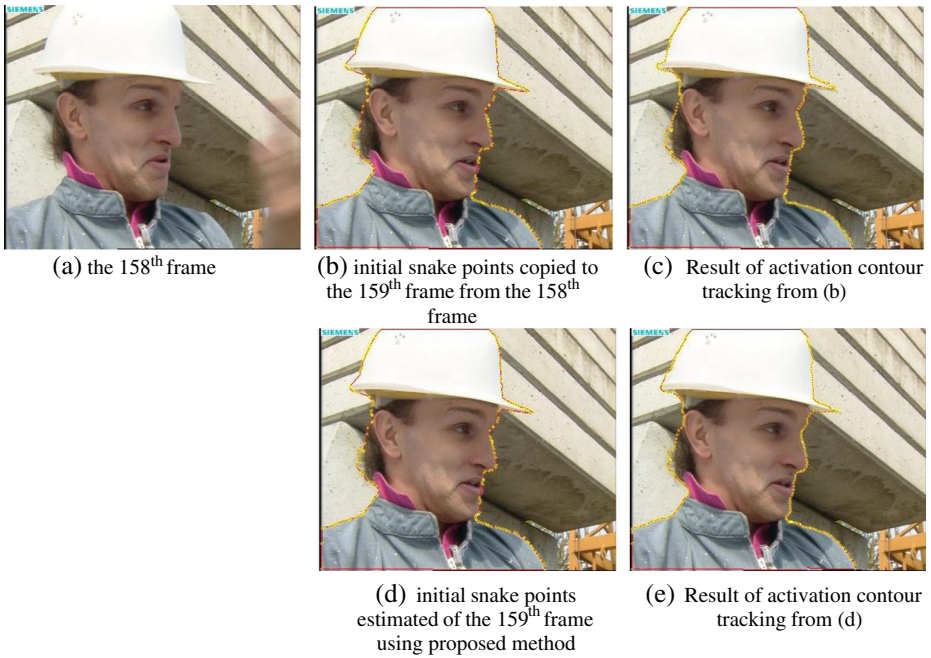
(a) the 158th frame

(b) initial snake points copied to the 159th frame from the 158th frame

(c) Result of activation contour tracking from (b)

(d) initial snake points estimated of the 159th frame using proposed method

(e) Result of activation contour tracking from (d)

Fig. 5 Results of experiments with the foreman video

The first experiment is with 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
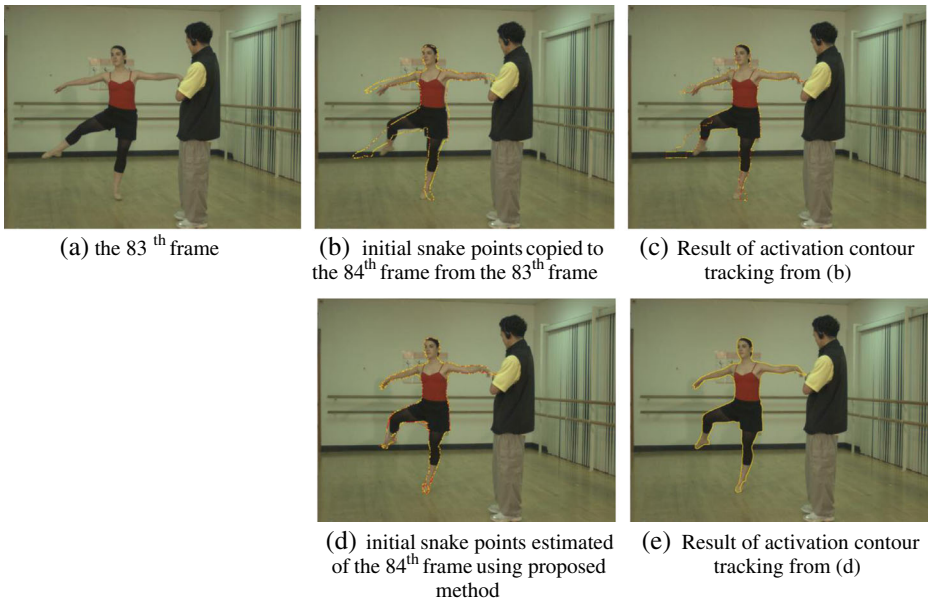

(a) the 83th frame

(b) initial snake points copied to the 84th frame from the 83th frame

(c) Result of activation contour tracking from (b)

(d) initial snake points estimated of the 84th frame using proposed method

(e) Result of activation contour tracking from (d)

Fig. 6 Results of experiments with the ballet video

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 %. Figure 5 compares the results of the existing method of using feature points of the previous frame as they are with those of the proposed method to set initial snake points in the 159th frame. The mean errors were 2.91 and 2.23 pixels, respectively, which indicate that while both methods tracked actual contour of the object almost similarly, the proposed method worked better.

The second experiment is with the ballet video with less frames and large motion. Tracking was performed in the 84th frame for the object extracted manually from the 83th frame. Weights of continuity energy, curvature energy, and external energy of the snake algorithm are 0.2, 0.15, and 0.65, respectively. Figure 6 shows how the results of using feature points of the previous frame differ from those of the proposed method to set initial snake points. The mean errors were 23.33 and 5.61 pixels, respectively. These findings reveal that by setting initial snake points in the proposed method, the contour of the object was tracked more exactly. However, in Fig. 6(e), some points near the left hand and left foot are not converged properly to the contour of the object. Motion blur is considered to make the object contour blurred near the left hand, while external energy shortage due to highly similar brightness of the object and the background appears to have caused the problem near the left foot. In general, the boundary of the object was tracked exactly in the remaining areas.

5 Conclusion

In this study, we proposed a method of tracking object contour with large motion and irregular shape in image sequence 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 image sequence 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, further research is necessary to solve the problem of motion blur or energy shortage due to a small brightness difference from the background. In addition, subsequent studies will focus on how to apply a background extraction method or a mean-shift algorithm to extract edges of an object more exactly.

References

1. Javed O, Rasheed Z, Shafique K, Shah M (2003) Tracking across multiple cameras with disjoint views. In: Proc. IEEE International Conference on Computer Vision, Nice, France, vol 2, pp 952–957. doi:10.1109/ICCV.2003.1238451
2. Kass M, Witkin A, Terzopoulos D (1988) Snakes: active contour models. Int J Comput Vision 1(4):321–331. doi:10.1007/BF00133570

3. Kim J, Lee J, Kim C (2011) Video object extraction using contour information. J Inst Electron Eng Korea 48(1):33–45
4. 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
5. Lee JH (2009) A study on an improved object detection and contour tracking algorithm based on local curvature. Dissertation, Paichai University
6. Lee YS (2011) The trends and prospects of 2D-3D conversion technology. Mag Inst Electron Eng Korea 38(2):37–43
7. Lee JW, You S, Neumann U (2000) Large motion estimation for omnidirectional vision. In: Proc. IEEE Workshop on Omnidirectional Vision, Hilton Head Island, USA, pp 161–168. doi:10.1109/OMNVIS.2000.853824
8. 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. doi:10.1109/34.216733
9. Li Y, Sun J, Shum HY (2005) Video object cut and paste. ACM Trans Graph 24(3):595–600. doi:10.1145/1073204.1073234
10. Li B, Yuan B, Yunda S (2006) Moving object segmentation using dynamic 3D graph cuts and GMM. In: Proc. IEEE International Conference on Signal Processing, Beijing, China, vol 2, pp 16–20. doi:10.1109/ICOSP.2006.345658
11. Lucas BD, Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: Proc. International Joint Conference on Artificial Intelligence, Vancouver, Canada, pp 674–679
12. Okino T, Murata H, Taima K, Iinuma T, Oketani K (1996) New television with 2D/3D image conversion technologies. In: Proc. SPIE 2653 Stereoscopic Displays and Virtual Reality Systems III: 96–103. dio:10.1117/12.237421
13. Pi L, Fan J, Shen C (2007) Color image segmentation for objects of interest with modified geodesic active contour method. J Math Imaging Vision 27(1):51–57. doi:10.1007/s10851-006-9797-3
14. Xiong B, Yu W, Charoensak C (2004) Face contour tracking in video using active contour model. Proc Int Conf Image Process 2:1024. doi:10.1109/ICIP.2004.1419475
15. Xu C, Prince JL (1997) Gradient vector flow: A new external force for snakes. In: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, pp 66–71. doi:10.1109/CVPR.1997.609299
16. Zitnick CL, Kang SB, Uyttendaele M, Winder S, Szeliski R (2004) High-quality video view interpolation using a layered representation. ACM Trans Graph 23(3):600–608. doi:10.1145/1015706.1015766

Jin Woo Choi received the M.S. and Ph.D. degree in Computer Science from Gachon University, Korea in 2007 and 2012, respectively. He is currently working as a Postdoctoral Researcher at Culture Technology Institute in Gachon University, Korea. His research focuses on image processing, computer vision and 3D Stereo Conversion.

Taeg Keun Whangbo received the M.S. degree from City University of New York in 1988 and the Ph.D degree both in Computer Science from Stevens Institute of Technology in 1995. Currently, he is a professor in the Department of Interactive Media, Gachon University, Korea. He is also the Dean of Research Affairs in Gachon University. Before he joined the Gachon University, he was the software developer in Q-Systems which is located in New Jersey from 1988 to 1993. He was also the researcher in Samsung Electronics from 2005 to 2007. From 2006 to 2008, he was the president of the Association of Korea Cultural Technology. His research areas include Computer Graphics, HCI and AR.

Cheong Ghil Kim received the M.S. and Ph.D. degree in Computer Science from Yonsei University, Korea, in 2003 and 2006, respectively. Currently, he is a professor at the Department of Computer Science, Namseoul University, Korea. His research areas include Multimedia Embedded Systems and AR.