

# Object Tracking Using a New Level Set Model

Nagi H. Al-Ashwal, Abdulmajid F. Al-Junaid

Electrical Engineering Department, Ibb University, Ibb, Yemen.  
Email: [nlashwal@yahoo.com](mailto:nlashwal@yahoo.com)

Received November 18<sup>th</sup>, 2013; revised December 18<sup>th</sup>, 2013; accepted December 25<sup>th</sup>, 2013

Copyright © 2014 Nagi H. Al-Ashwal, Abdulmajid F. Al-Junaid. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. In accordance of the Creative Commons Attribution License all Copyrights © 2014 are reserved for SCIRP and the owner of the intellectual property Nagi H. Al-Ashwal, Abdulmajid F. Al-Junaid. All Copyright © 2014 are guarded by law and by SCIRP as a guardian.

## ABSTRACT

**This paper proposes a novel implementation of the level set method that achieves real-time level-set-based object tracking. In the proposed algorithm, the evolution of the curve is realized by simple operations such as switching values of the level set functions and there is no need to solve any partial differential equations (PDEs). The object contour could change due to the change in the location, orientation or due to the changeable nature of the object shape itself. Knowing the contour, the average color value for the pixels within the contour could be found. The estimated object color and contour in one frame are the bases for locating the object in the consecutive one. The color is used to segment the object pixels and the estimated contour is used to initialize the deformation process. Thus, the algorithm works in a closed cycle in which the color is used to segment the object pixels to get the object contour and the contour is used to get the typical-color of the object. With our fast algorithm, a real-time system has been implemented on a standard PC. Results from standard test sequences and our real time system are presented.**

## KEYWORDS

**Object Tracking; Deformable Model; Level Set Method; HSI Color Space**

## 1. Introduction

Tracking is the problem of generating an inference about the motion of an object given a sequence of images. Good solutions to this problem have a variety of applications areas including surveillance [1-7], human-computer interfaces [3,4], and medical imaging [8,9].

Deformable models have been used in tracking tasks [8] and [10-12]. Deformable model is appealing for object tracking because the object in the image sequence can vary from frame to frame due to a change in the view point, the motion of the object, or its nonrigid nature. The principal challenges for object tracking using deformable models are to develop accurate deformable model of image variability and to design computationally efficient tracking algorithms which use this model. Also parametric deformable models have some difficulties in handling topological changes such as the merging and splitting of object regions. For these reasons, the level set method [13,14] is a more powerful technique. But the high

computational cost of the level set method has limited its popularity in real-time scenarios.

Tracking using level set methods can be classified as edge based or region based. Edge based models can be traced back to the snake model in [15]. Using the difference between the current frame and a reference background, a region based model was proposed in [16]. Considerable work has been done to improve the speed of level-set-based curve evolution. The basic idea has been to limit the solution of the level set PDE to a narrow band around the zero level set [17,18], but issues such as narrow band construction, reinitialization and step size control still make existing level set methods computationally too expensive for real-time tracking. In [19] the authors proposed a region based level set model for tracking. They used an energy functionality that contains two terms, fidelity term and smoothness term. The resultant evolution equation contains two speeds. The speed represents the competition between two regions, and the speed makes the curve smooth. For implementation of

their evolution equation they defined two lists of neighboring pixels  $L_{in}$  and  $L_{out}$  of the contour  $\Gamma(t)$ . Their level set is evolved by switching the neighboring pixels between the two lists  $L_{in}$  and  $L_{out}$ .

In this paper, we propose a novel approach to tracking using the level set methods. Our approach is based on that the implicitly represented curves can be moved by changing the sign of values of the level set function inside and outside the curve. Our approach requires simple addition operation and then changes the sign of some values inside or outside the curve to achieve the tracking. Our method does not contain any functional energy, so it does not need to solve any partial differential equations (PDEs), thus reducing the computation dramatically compared with other techniques proposed before. The proposed tracking technique tries to trace both contour and color variations in order to locate the object in the consecutive frames. With our approach, real-time level-set-based video tracking can be achieved by using a conventional PC.

The rest of the paper is organized as follows. In Section 2 we give a brief review of the level set methods. Section 3 presents the new level set technique for tracking. In Section 4 the proposed technique for object tracking is described. The experimental results are presented in the Section 5. The paper is concluded in Section 6.

## 2. Level Set Methods

The main idea behind the level set formulation is to represent an interface  $\Gamma(t)$  bounding a possibly multiply connected region in  $R^n$  by a Lipschitz continuous function  $\phi$ , changing sign at the interface, *i.e.*

$$\begin{cases} \phi(x,t) > 0, & \text{if } x \text{ is inside } \Gamma(t), \\ \phi(x,t) = 0, & \text{if } x \text{ is at } \Gamma(t), \\ \phi(x,t) < 0, & \text{if } x \text{ is outside } \Gamma(t). \end{cases} \quad (1)$$

In numerical implementations, often regularity is imposed on  $\phi$  to prevent the level set function of being too steep or flat near the interface. This is normally done by requiring  $\phi$  to be the signed distance function to the interface

$$\begin{cases} \phi(x,t) = d(\Gamma(t), x), & \text{if } x \text{ is inside } \Gamma(t), \\ \phi(x,t) = 0, & \text{if } x \text{ is at } \Gamma(t), \\ \phi(x,t) = -d(\Gamma(t), x), & \text{if } x \text{ is outside } \Gamma(t), \end{cases} \quad (2)$$

where  $d(\Gamma(t), x)$  denotes Euclidian distance between  $x$  and  $\Gamma(t)$ . We emphasize that requiring (2) is a technicality to prevent instabilities in numerical implementations.

The interface  $\Gamma(t)$  is implicitly moved according to the nonlinear PDE

$$\frac{\partial \phi}{\partial t} + v(\phi) \cdot \nabla \phi = 0, \quad (3)$$

where  $v(\phi)$  is a given velocity field. This vector field can depend on geometry, position, time and internal or external physics. Usually only the velocity normal to the interface  $v_N$  is needed, and  $\phi$  is then moved according to the modified equation

$$\frac{\partial \phi}{\partial t} + v_N(\phi) |\nabla \phi| = 0. \quad (4)$$

## 3. Tracking Level Set Model

In this paper, we use a region-based level set model for the object tracking. The tracking process is depending on the quality of the matching between the object contour and the interface of the level set function,  $\Gamma(t)$ . The tracked object is segmented then the tracking process is started.

The used level set function,  $\phi$  is be given by:

$$\phi(r,c) = \begin{cases} 1 & \text{if } (r,c) \text{ is inside } \Gamma(t), \\ -2 & \text{if } (r,c) \text{ is outside } \Gamma(t), \\ 0 & \text{if } (r,c) \text{ is in } \Gamma(t), \end{cases} \quad (5)$$

where  $(r,c)$  are the row and column coordinates of the current point. From (5) we can conclude that  $\phi$  contains two regions, inside region with value “+1” and outside region with value “-2”, see **Figure 1**.

Our method does not need to solve any PDE, so regularity is not important. Therefore  $\phi$  is not required to be a distance function. The tracked object is segmented and given the value “1”, the value “0” is reserved for the background, see **Figure 2**.

## Deformation

The deformation process is started by adding both the segmented image,  $I$ , see **Figure 2**, and the current level set function  $\phi^n$ , see **Figure 1**. The result is put in a new image called  $\psi$ ,

$$\psi = I + \phi^n. \quad (6)$$

Every intersection between the object region, and level set regions has a unique value in the resultant image,  $\psi$ . It is clear that any point in  $\psi$  will take its value based on the following:

$$\psi(r,c) = z, \quad z = \{-2, -1, 1, 2\}. \quad (7)$$

The number of points carrying any value,  $z$ , in  $\psi$  gives information about the amount of overlap between the level set function and the object, we have four states, see **Figure 3**:

1) The number of occurrence of the value “-1” in  $\psi$ , results from addition of the object and the exterior region

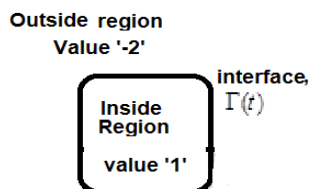


Figure 1. Proposed level set model,  $\phi$ .

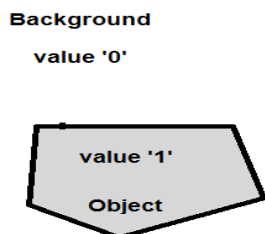


Figure 2. Segmented image model,  $I(r,c)$ .

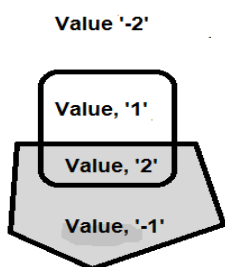


Figure 3. The resultant image after the addition,  $\psi(r,c)$ .

of  $\phi$ , represents the area in which the object misses the interior region of  $\phi$ .

2) The number of occurrence of the value “2” represents the area in which the object hits the interior region of  $\phi$ .

3) The number of occurrence of the value “1”, results from the addition of the background and the interior region of  $\phi$ , represents the area in which the interior region of  $\phi$  misses the object.

4) The number of occurrence of the value “-2”, results from addition of the background and exterior region of  $\phi$ , represents the area in which the exterior region of  $\phi$  hits the background.

The next step is the deformation process which is done by simply changing some values in the resultant image,  $\psi$ . The values are changed to force the interface contour,  $\Gamma(t)$ , to catch the object’s boundary. By considering the object-mask arrangement shown in Figure 3, we can observe that to force  $\Gamma(t)$  to catch the object the values of  $\phi$  should be changed based on the following rules, see Figure 4:

1) The pixels of values “-2” and “2” remain without change,

2) the pixels of value “1” is regarded as outside region so it should be changed to the value “-2”,

3) the pixels of value “-1” is regarded as inside region

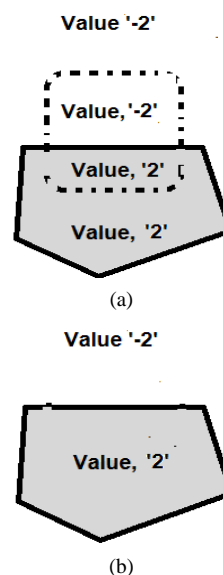


Figure 4. Deformation process, (a) changing values in  $\psi$  to catch the object, (b) the new level set function,  $\phi$  after catching the object.

so it should be changed to the value “2”. The above steps can be achieved by:

$$\phi^{n+1}(r,c) = \begin{cases} 2 & \text{if } \psi(r,c) = -1, \\ -2 & \text{if } \psi(r,c) = 1, \\ \psi(r,c) & \text{otherwise.} \end{cases} \quad (8)$$

4) Finally, to get the level set model in (5) the pixels of the exterior region are changed to the value “-1”,

$$\phi^{n+1}(r,c) = \begin{cases} 1 & \text{if } (r,c) \text{ is inside } \Gamma(t), \\ -2 & \text{if } (r,c) \text{ is outside } \Gamma(t). \end{cases} \quad (9)$$

## 4. Proposed System

In this section our algorithm for object tracking is described. Before the start of tracking process there are two necessary steps. The first step, Display, is to select the object for tracking. Second step, Initialization, is carried out to locate the object contour.

### 4.1. Display

The captured images are previewed and the initial level set function,  $\phi$ , is created by hand and given values as shown in Figure 1. The process is done by putting the interface  $\Gamma(t)$  over the object needed to be tracked, see Figure 5 in the experimental result section.

### 4.2. Initialization

The current frame is converted from RGB color system into HSI color space that decouples the brightness information from the color information. Under this represent-

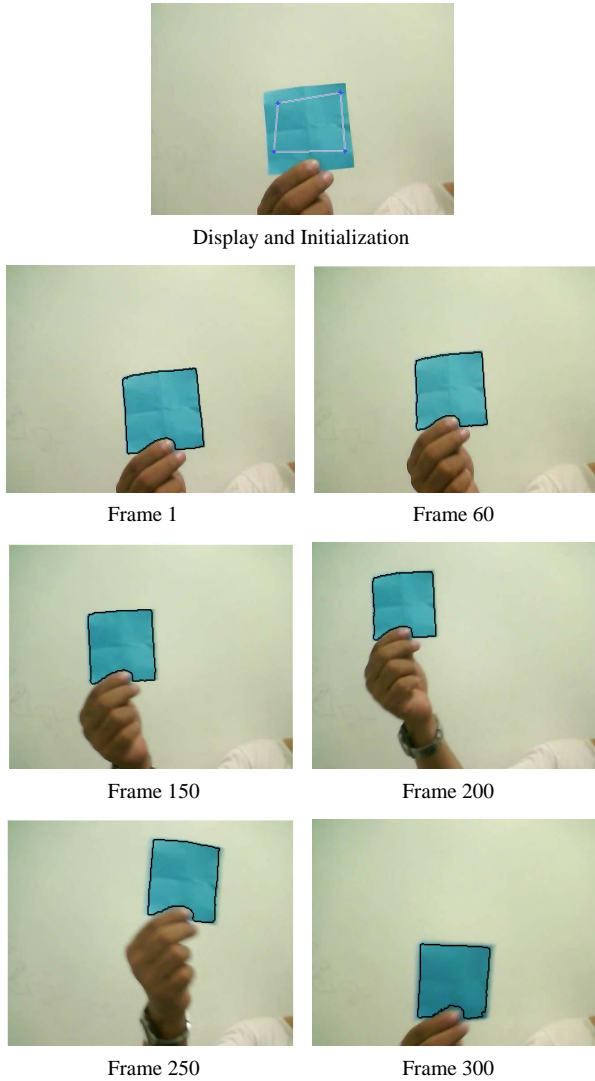


Figure 5. Example 1 of the tracking of an object.

tation we could easily segment the object by assuming that the changes of the object hue or saturation are much smaller than the expected changes in the intensity. Thus the object is a blob in the image with limited variation in hue and saturation and slightly larger variation in the intensity. The object segmentation is performed in two steps: 1) get the typical-color of the object, see [Algorithm 1](#), and 2) use the typical-color to threshold the current frame to extract the object, see [Algorithm 2](#), the resultant image,  $I$ , is binary image with object and background values as shown in [Figure 2](#).

Now we have  $\phi$  and  $I$  so (6) can be used to find the new image,  $\psi$ . The new  $\phi^{n+1}$ , which catches the object, is then found by (7) and (8).

### 4.3. Tracking-Target

The tracking is a sequence of operations; the first step is to use the level set function,  $\phi$ , from the previous frame,

#### Algorithm 1. Get the typical color of the object.

<b>Get Hsi Thresholds</b>	Find the average of values of $H_{AV}$ , $S_{AV}$ , $I_{AV}$ of pixels in the region which encircled by a contour
Inputs	Image, $I$ , Leve set function, $\phi$
Outputs	the average values $H_{AV}$ , $S_{AV}$ , $I_{AV}$
Steps	<pre> Initialize <math>H_{AV} = 0</math>; <math>S_{AV} = 0</math>; <math>I_{AV} = 0</math>; <math>No\ Of\ Pixels = 0</math>; <b>for</b> <math>I = 0</math> <b>to</b> Image Rows <b>do</b>   <b>begin</b>     <b>for</b> <math>j = 0</math> <b>to</b> Image Columns <b>do</b>       <b>begin</b>         <b>if</b> <math>image(i,j)</math> is inside the region which encircled by         model <b>then</b>           <b>begin</b>             <math>H_{AV} = H_{AV} + image(i,j, HUE)</math>             <math>S_{AV} = S_{AV} + image(i,j, SATURATION)</math>             <math>I_{AV} = I_{AV} + image(i,j, INTENSITY)</math>             <math>No\ Of\ Pixels = No\ Of\ Pixels + 1</math>           <b>end</b>         <b>end</b>       <b>end</b>     <b>end</b>     <math>H_{AV} = H_{AV}/No\ Of\ Pixels,</math>     <math>S_{AV} = S_{AV}/No\ Of\ Pixel,</math>     <math>I_{AV} = I_{AV}/No\ Of\ Pixels</math> </pre>

#### Algorithm 2. Extract the object.

<b>Extract Object</b>	Extract the object by set most of its pixels to value "1" and most of other pixels in the image to value "0"
Inputs	HSI Image, $H_{AV}$ , $S_{AV}$ , $I_{AV}$ , $t_H$ , $t_S$ , $t_I$ , $t_H$ is tolerance of plane $H^*$
Outputs	Segmented binary Image $BW(r,c)$ in which most of the object pixels are "1" and most of the background pixels are "0"
Steps	<pre> <math>H_{upper} = H_{AV} + t_H</math>, <math>H_{lower} = H_{AV} - t_H</math> <math>S_{upper} = S_{AV} + t_S</math>, <math>S_{lower} = S_{AV} - t_S</math> <math>I_{upper} = I_{AV} + t_I</math>, <math>I_{lower} = I_{AV} - t_I</math> <b>for</b> <math>i = 0</math> <b>to</b> Image Rows <b>do</b>   <b>begin</b>     <b>for</b> <math>j = 0</math> <b>to</b> Image Columns <b>do</b>       <b>begin</b>         <math>BW(r,c) = \begin{cases} 1 &amp; \text{if } (H_{lower} \leq I(r,c,H) \leq H_{upper}) \\ &amp; \text{and } (S_{lower} \leq I(r,c,S) \leq S_{upper}) \\ &amp; \text{and } (I_{lower} \leq I(r,c,I) \leq I_{upper}) \\ 0 &amp; \text{Otherwise} \end{cases}</math>       <b>end</b>     <b>end</b>   <b>end</b> </pre>

in order to get the typical-color in the current frame. The second step is to use this color for object segmentation, *i.e.* getting the image  $I$ . Finally we use  $I$  and  $\phi$  to deform the model towards a better fit using the steps in Section 3.A. By this the interface of the level set will move to catch the moved target.

### 4.4. Advantages and Disadvantages

Some of methods presented in the literature have a matching problem in which the systems mistake the object. This is because the objects may change color due to environmental or camera changes. Kinematics-based



algorithms may not work properly when the objects abruptly change their motion from one frame to the next [20,21]. Such situations can be due to collisions or sharp turns. Also the algorithms that detect motion in a sequence of frames by extracting tokens such as edges, corners, interest points, etc. [22,23] mistake objects when there is a change in their 2-dimensional shape [24,25]. The solution of the above problems is to use high frame rate. By maintaining the interframe time small, we maintain both shape and color variation to minimum. High frame rate requires a high speed tracking algorithm to segment and track the object in the small interframe time.

The proposed method does not contain any PDE and the tracking process is done by simple addition and change values operations, see (6) and (8). Thus, the speed of deformation is the strongest point in the proposed system. It is capable of tracking objects at 30 frames/sec on regular PC (CORE 2 Duo). Each of the frames is  $480 \times 640$  RGB 24-bit pixels. This speed keeps the color position and shape variation in the interframe time small. Consequently the reliability of the tracking process increases.

The most important limitation of the proposed system is that it loses track of the object when the object is partially occluded by similar color object. In this case the deformation algorithm tends to combine both similar color objects within a single contour. As one of the object moves away from the other the algorithm will follow both objects. Also, the algorithm gives poor results if there is no color difference or intensity contrast between the object and the background as with all tracking algorithms.

## 5. Experimental Results

In this section two examples are given to demonstrate the system capability for real time tracking. In all experiments we used CORE 2 Duo Laptop with 2.2 GHz speed and 4 GB memory.

### 5.1. Example 1

The used video is taken using a webcam at 30 frames per second. The sequence consists of frames each of ( $480 \times 640$  pixels). As shown in **Figure 5**, frame 0 shows the object (blue paper) with the initial model being place on it. The initial contour is determined manually using the mouse. In the initialization the deformation process is used to get the contour of the object. As can be clearly shown in some subsequent frames in **Figure 5**, the proposed system is able to continuously track the object real time. The motion of the contour is due to the change of the location of the object.

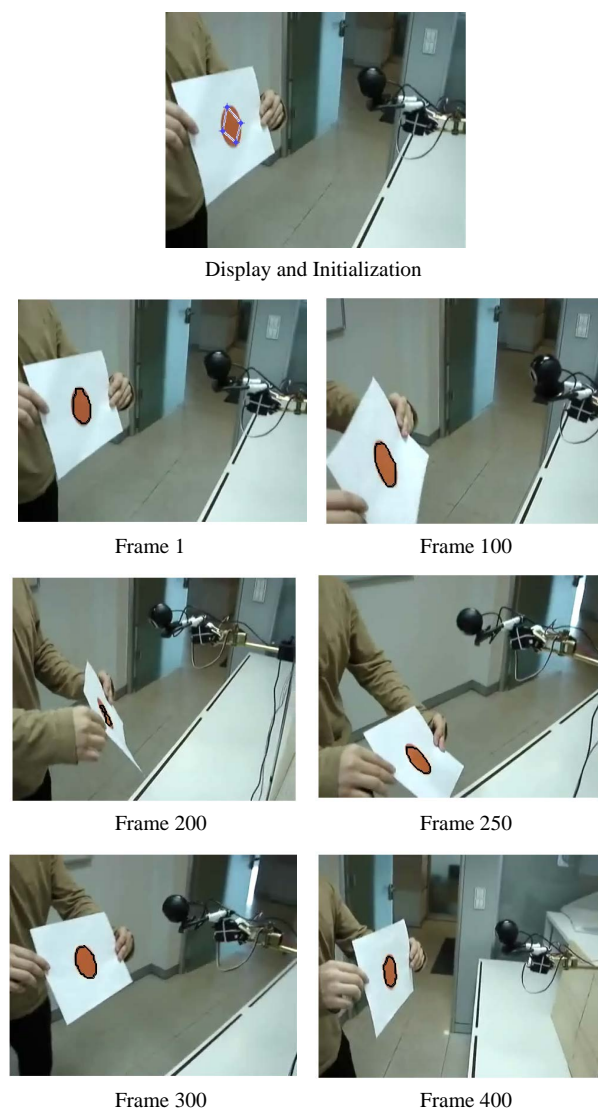
### 5.2. Example 2

The proposed system is applied to track an object in a

video downloaded from the internet. The downloaded video has 30 frames per second each is  $480 \times 640$  pixels. **Figure 6** shows the tracking results of the object. The object color and contour are used for tracking. Despite the motion and shape changes of the object, the deformable model reasonably delineates the contour of the object throughout the whole sequence.

## 6. Conclusion

This paper introduced a new tracking algorithm that is fast enough to track an object in real time. The used definition of an object is any connected area in the captured frame which has a consistent feature. Although arbitrary complex feature could be used, the one selected is the HSI representation of color which practically fits a wide range of applications. A deformable model which depends on the level set methods has been pursued because



**Figure 6.** Example 2 of the tracking of an object.

of its speed and simplicity. The deformation requires only two-image addition to obtain the contour of the target object. Using this algorithm, a real time system was constructed using a standard CORE 2 Duo Laptop. Many experiments are made on real videos with frame rate of 30 frames/sec. The result showed that the proposed technique can track an object in real time.

## REFERENCES

- [1] H. Meyerhoff, F. Papenmeier and M. Huff, "Object-Based Integration of Motion Information during Attentive Tracking," *Perception*, Vol. 42, No. 1, 2013, pp. 119-121. <http://dx.doi.org/10.1068/p7273>
- [2] A. Feldman, M. Hybinette, T. Balch and R. Cavallaro, "The Multi-ICP Tracker: An Online Algorithm for Tracking Multiple Interacting Targets," *Journal of Field Robotics*, Vol. 29, No. 2, 2012, pp. 258-276. <http://dx.doi.org/10.1002/rob.21402>
- [3] S. B. Gokturk, C. Tomasi, B. Girod and J.-Y. Bouguet, "Model-Based Face Tracking for View-Independent Facial Expression Recognition," *5th IEEE International Conference on Automatic Face and Gesture Recognition*, Washington, 21-21 May 2002, pp. 287-293.
- [4] M. Wang, Y. Iwai and M. Yachida, "Expression Recognition from Time-Sequential Facial Images by Use of Expression Change Model," *3rd IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, 14-16 April 1998, pp. 324-329. <http://dx.doi.org/10.1109/AFGR.1998.670969>
- [5] O. Javed and M. Shah, "Tracking and Object Classification for Automated Surveillance," *The 7th European Conference on Computer Vision (ECCV 2002)*, Copenhagen, 2002, pp. 343-357.
- [6] R. Howarth and H. Buxton, "Visual Surveillance Monitoring and Watching," *Proceedings of European Conference on Computer Vision*, Vol. 2, 1996, pp. 321-334.
- [7] T. Frank, M. Haag, H. Kollnig and H.-H. Nagel, "Tracking of Occluded Vehicles in Traffic Scenes," *Proceedings of European Conference on Computer Vision*, Vol. 2, 1996, pp. 485-494.
- [8] N. Ray and S. T. Acton, "Active Contours for Cell Tracking," *5th IEEE Southwest Symposium on Image Analysis and Interpretation*, Sante Fe, 7-9 April 2002, pp. 274-278.
- [9] E. Bardinet, L. Cohen and N. Ayache, "Tracking Medical 3D Data with a Deformable Parametric Model," *Proceedings of European Conference on Computer Vision*, Vol. 1, 1996, pp. 317-328.
- [10] G. Unal, H. Krim and A. Yezzi, "Active Polygon for Object Tracking," *1st International Symposium on 3D Data Processing Visualization and Transmission (3DPVT 02)*, 2002, p. 696.
- [11] Y. Q. Chen, T. Huang and Y. Rui, "Mode-Based Multi-Hypothesis Head Tracking Using Parametric Contours," *5th IEEE International Conference on Automatic Face and Gesture Recognition*, Washington, 20-21 May 2002, pp. 112-117.
- [12] Y. Zhong, A. K. Jain and M.-P. Dubuisson-Jolly, "Object Tracking Using Deformable Templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 5, 2000, pp. 544-549. <http://dx.doi.org/10.1109/34.857008>
- [13] S. Osher and N. Paragios, "Geometric Level Set Methods in Imaging, Vision and Graphics," Springer Verlag, 2003.
- [14] J. Sethian, "Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanisms, Computer Vision, and Material Science," Cambridge University Press, 1999.
- [15] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active Contour Models," *International Journal of Computer Vision*, Vol. 1, No. 4, 1988, pp. 321-331. <http://dx.doi.org/10.1007/BF00133570>
- [16] S. Besson, M. Barlaud and G. Aubert, "Detection and Tracking of Moving Objects Using a New Level Set Based Method," *Proceedings of ICPR*, Vol. 3, 2000, pp. 1100-1105.
- [17] D. Adalsteinsson and J. Sethian, "A Fast Level Set Method for Propagating Interfaces," *Journal of Computational Physics*, Vol. 118, 1995, pp. 269-277.
- [18] N. Paragios and R. Deriche, "Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, 2000, pp. 266-280.
- [19] Y. G. Shi and W. C. Karl, "Real-Time Tracking Using Level Sets," *Proceedings of IEEE Conference on Computer Vision Pattern Recognition (CVPR '05)*, Vol. 2, 2005, pp. 34-41.
- [20] H. Shariat and K. E. Price, "Motion Estimation with more than Two Frames," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 5, 1990, pp. 417-432. <http://dx.doi.org/10.1109/34.55102>
- [21] B. K. P. Horn and B. Schunck, "Determining Optical Flow," *Artificial Intelligence*, Vol. 17, No. 1-3, 1981, pp. 185-203. [http://dx.doi.org/10.1016/0004-3702\(81\)90024-2](http://dx.doi.org/10.1016/0004-3702(81)90024-2)
- [22] X. Zhuang, T. S. Huang, N. Ahuja and R. M. Haralick, "A Simplified Linear Optic Flow-Motion Algorithm," *Computer Vision, Graphics and Image Processing*, Vol. 42, No. 3, 1988, pp. 334-344. [http://dx.doi.org/10.1016/S0734-189X\(88\)80043-4](http://dx.doi.org/10.1016/S0734-189X(88)80043-4)
- [23] L. S. Davis, Z. Wu and H. Sun, "Contour-Based Motion Estimation," *Computer Vision, Graphics and Image Processing*, 1982, pp. 313-326.
- [24] R. Bergevin and M. D. Levine, "Extraction of Line Drawing Features for Object Recognition," *Pattern Recognition*, Vol. 25, No. 3, 1992, pp. 319-334. [http://dx.doi.org/10.1016/0031-3203\(92\)90113-W](http://dx.doi.org/10.1016/0031-3203(92)90113-W)
- [25] C. Cédras and M. Shah, "Motion-Based Recognition: A Survey," *Image and Vision Computing*, Vol. 13, No. 2, 1995, pp. 129-155.