Object Tracking using Generalized Gradient Vector Flow

Israa A. Alwan*

Received on: 2/11/2010
Accepted on: 7/4/2011

Abstract

The aim of an object tracker is to generate the trajectory of an object over time by locating its position in every frame of the video. In this research, we present an object contour tracking approach using Generalized Gradient Vector Flow (GGVF). GGVF active contour, or snake, is a dynamic curve that moves within an image domain to capture desired image features. Mostly, GGVF is not sensitive to initial conditions and converges to the optimal contour. Given an initial contour near the object in the first video frame, GGVF can iteratively converge to an optimal object boundary. In each video frame thereafter, the resulting contour in the previous video frame is taken as initialization so the algorithm consists of two steps. In the first step, the initial contour is applied to the desired object in first video frame. The resulting contour is taken as initialization of the second step, which applies GGVF to current video frame. To evaluate the tracking performance, we applied the algorithm to several real world video sequences. Experimental results are provided.

Keywords: Object tracking, Generalized gradient vector flow, Motion-based approach, Contour-based approach, Active contours, Snakes.

تعقب الجسم بأستخدام تدفق متجه الميل المعمم

الخلاصة

ان الهدف من متعقب الجسم (object tracker) هو توليد مسار الجسم المطلوب بمرور الوقت عن طريق تحديد موقعه في كل شريحه (frame) من الفديو. في هذا البحث, تم تنفيذ تعقب الجسم في الفديو بأستخدام خوارزمية المنحنيات النشيطة العاملة وفقا لتدفق متجه الميل المعمم (GGVF) . المنحنيات النشيطة العاملة وفقا لتدفق متجه الميل المعمم, او تسمى الافاعي, هي منحنيات ديناميكية او حركية تتحرك ضمن مجال الصورة بهدف تحديد او التقاط خصائص الصورة ذات الاهمية. في الغالب هذه المنحنيات غير حساسة لشروط الحالة الابتدائية للمنحنى وتستقر على الحدود المثلى للجسم. فعند اعطاء او تسليط المنحنى الابتدائي قرب حدود الجسم المطلوب في الشريحة الاولى من الفديو, فأن المنحنيات النشيطة العاملة وفقا لتدفق متجه الميل المعمم (GGVF) سوف تستقر على حدود الجسم المطلوب بشكل تكراري, وبالتالي, ولباقي شرائح الفديو فأن المنحنى الناتج (النهائي) في الشريحة الحالية يمكن ان يستخدم كمنحنى ابتدائي للشريحة التالية. لذلك خوارزمية تعقب الجسم تتألف من خطوتين. في الخطوة الاولى, يتم تسليط المنحنى الابتدائي (GGVF) قرب حدود الجسم المطلوب في الشريحة الاولى من الفديو. المنحنى الناتج في الخطوة الاولى يستخدم كمنحنى ابتدائي لتعقب نفس الجسم في الشريحة القادمة في الخطوة الثانية, وهكذا لبقية الشرائح. لغرض تقييم اداء الخوارزمية , فأنه تم تطبيقها على شرائح لفديوات مختلفة. النتائج التجريبية متوفره في البحث.

* Eduction College of Ibn Al-Haithem , University of Baghdad/ Baghdad

1. Introduction

Object tracking has been a hot topic in the area of computer vision. It is useful in a wide range of applications, such as military domain, perceptual user interface, augmented reality, video montage, motion-based recognition, automated surveillance, video indexing, human-computer interaction, traffic monitoring, and the vehicle navigation. All these applications require robust and efficient visual tracking which is still an open problem. In most of the applications, the camera keeps still so that the background is unchanged while the objects are moving within the field of view. This makes the tracking problem easier than the one with a changing background. However, there are still many difficulties even in this simplified scenario, such as loss of depth information, noise in images, complex object motion, non-rigid or articulated nature of objects, Partial and full object occlusions, complex object shapes, scene illumination changes, real-time processing requirements, and the presence of other moving objects [1].

Various tracking approaches have been proposed in computer vision literature. A common one is the motion-based approach. In this approach, moving object detection and tracking is done by using a static camera. Because the background is static and the desired object is in motion, a background model constructing is needed. The moving object (foreground) in incoming video frames is detecting by using subtraction algorithms [2]. Many different subtraction algorithms have been proposed over the recent years, most of them have the capabilities of modeling the changing illumination, noise, and the periodic motion of the background regions and, therefore, can accurately detect objects in a variety of circumstances. Moreover, these methods are computationally efficient. *In practice* although background subtraction algorithms can handle illumination changes but can easily be confused by rapid ones. Also, they provide incomplete object regions in many instances, that is, the objects may be spilled into several regions, or there may be holes inside the object since there are no guarantees that the object features will be different from the background features. These algorithms work on the entire domain of the video frame and this requires much time [3]. *The most important limitation of background subtraction algorithms is the assumption of camera stationarity* [4, 5].

In this research we regard the contour-based approaches, where the object detection can be formulated as matching problem between model contour parts and image edge fragments (Edge Track) [6]. In other words, they usually do not make use of the spatial and motion information of the entire object and rely only on the information closer to the boundary of the video object, so these approaches are particularly useful in segmenting and tracking a moving object without constructing a background model for a video scene, then no stationary camera. We used the active contours models, which are dynamic curves that move within an image domain to capture desired image features. There are two types of active contour models: parametric snakes and geometrical snakes. Parametric snakes are represented explicitly as parameterized contours and the snake evolution is carried out

on the predetermined spline control points only. Geometrical snakes, on the other hand are represented implicitly as the zero-level sets of higher dimensional surfaces, and the updating is performed on the surface function within the entire image domain. But the geometrical snakes have several drawbacks when compared with parametric snakes. Firstly, the level set scheme makes it difficult, if not impossible, to impose arbitrary geometric or topological constraints on the evolving contour indirectly through the higher dimensional hyper-surface. Secondly, user-defined external force would be very inconvenient to be added. Thirdly, geometrical snakes may generate shapes that have inconsistent topology with respect to the actual object, when applied to noisy images with significant boundary gaps [7], so in our research we used the parametric snakes.

Since first parametric model introduced by Kass *et al.* [8] in 1987, numerous methods have been developed attempting to improve the performance of the parametric snakes, which includes overcoming the two major drawbacks: *small capture range and lack of topological flexibility***.** A number of methods have been proposed to address the capture range problem, including the distance potential forces, pressure forces, multi-resolution methods, and gradient vector flow (GVF) [9]. Among the above mentioned remedies, the GVF snakes [9] and GGVF snakes [10] proposed by Xu and Prince have gained tremendous popularity due to their ability of attracting the active contour toward object boundary from a sufficiently large distance and the ability of moving the contour into object cavities. But GVF snake can't deal with long, thin boundary indentations [10]. Previously, we experienced this shortcoming when we used the GVF snake in the medical images segmentation. So, in this research, we shall use its modified model which, it is called the GGVF snake in another informational media which, it is the video for objects tracking.

2. Object Tracking Algorithm

Every tracking algorithm requires an object detection and correspondence mechanisms. These tasks can either be performed separately or jointly. In the first case, possible object regions in every video frame are obtained by means of an object detection algorithm, and then the tracker corresponds objects across video frames. In the latter case, the object region and correspondence is jointly estimated by iteratively updating object location and region information obtained from previous video frames. In this research the tracking algorithm as the latter case, where first, the object detection (segmentation) is performed by using active contour model (GGVF snake) and the correspondence mechanism is performed by using the projection principle, and this will be explained in the following sections.

2.1 Generalized Gradient Vector Flow (GGVF)

All facts and equations that relating to traditional snake (Kass snake) are explained in [8]. The most important fact is that the Kass snake has two problems. *First, it has limited capture rang; second, it can't deal with topological changes like concave region.* Xu and Prince replaced the external force of Kass snake by the GVF and produced the GVF snake.

GVF solved these problems where, it has many desirable properties as an external force for snakes [9]. But it still has difficulties, however, *forcing a snake into long, thin boundary indentations* [10]. Authors Xu and Prince hypothesized that this difficulty could be caused by excessive smoothing of the field near the boundaries, governed by the coefficient μ in following equation:

$$v_t = \mu\nabla^2 v - (v - \nabla f)|\nabla f|^2 \quad \text{..... (1)}$$

They reasoned that introducing a spatially varying weighting function, instead of the constant μ, and decreasing the smoothing effect near strong gradients, could solve this problem. In the following formulation, which they have termed *Generalized GVF* (GGVF), they replace both μ and $|\nabla f|^2$ in Eq. (1) by more general weighting functions [10].

The authors define GGVF as the equilibrium solution of the following vector partial differential equation:

$$v_t = g(|\nabla f|)\nabla^2 v - h(|\nabla f|)(v - \nabla f) \quad \text{..... (2)}$$

The first term on the right is referred to as the *smoothing term* since this term alone will produce a smoothly varying vector field. The second term is referred as the *data term* since it encourages the vector field ***v*** to be close to ∇f computed from the data. The weighting functions $g(\cdot)$ and $h(\cdot)$ apply to the smoothing and data terms, respectively. Since these weighting functions are dependent on the gradient of the edge map which is spatially varying, the weights themselves are spatially varying, in general. Since the vector field ***v*** must be slowly varying (or smooth) at locations far from the edges, but to conform to ∇f near the edges, $g(\cdot)$ and $h(\cdot)$ should be monotonically non-increasing and non-decreasing functions of $|\nabla f|$, respectively.

Equation (2) reduces to that of GVF when:

$$g(|\nabla f|) = \mu \quad \text{..... (3)}$$

And

$$h(|\nabla f|) = |\nabla f|^2 \quad \text{..... (4)}$$

Since $g(\cdot)$ is constant here, smoothing occurs everywhere; however, $h(\cdot)$ grows larger near strong edges, and should dominate at the boundaries. Thus, GVF should provide good edge localization. The effect of smoothing becomes apparent, however, when there are two edges in close proximity, such as when there is a long, thin indentation along the boundary. In this situation, GVF tends to smooth between opposite edges, losing the forces necessary to drive an active contour into this region.

To address this problem, weighting functions can be selected such that $g(\cdot)$ gets smaller as $h(\cdot)$ becomes larger. Then, in the proximity of large gradients, there will be very little smoothing, and the effective vector field will be nearly equal to the gradient of the edge map. The weighting functions for GGVF are computed as follows:

$$g(|\nabla f|) = e^{-(|\nabla f|/k)} \quad \text{..... (5)}$$

$$h(|\nabla f|) = 1 - g(|\nabla f| \quad \text{..... (6)}$$

The GGVF field computed using this pair of weighting functions will conform to the edge map gradient at strong edges, but will vary smoothly away from the boundaries. *The specification of **K** determines to some*

extent the degree of tradeoff between field smoothness and gradient conformity [10]. The solution remains the same as discussed previously under the subheading "GVF Snakes" in [9].

2.2 Object Contour Tracking

The object tracking process (correspondence) is performed by using the projection principle which, it is as follows:-

We first apply the initial contour on the desired object in the first video frame, when the segmentation of the object is completed, by opening the next video frame and loading or projecting this resulting contour, the contour will appear at the same coordinate positions as in the previous video frame, and it can then be iterated on the new video frame. Hence, no contour initialization is needed for the second video frame. This results in time saving and simplicity of operation.

3. General Flow of the Tracking Algorithm

The general execution of the tracking algorithm traces the path that is shown in Fig (1).

4. Implementation of the Proposed Object Tracking Algorithm

The tracking algorithm has been tested on grey level Audio Video Interlaced (AVI) sequences, and for estimating the performance of the algorithm, it was applied on variant *simple* and *complex* objects, as clarified in the following sections.

4.1 Simple Objects (Progressive in Complexity)

The object tracking algorithm was applied on three simple objects which are a line drawing starfish (cartoon), the sun as an object has a regular shape, and a jellyfish as an object has irregular shape. In all experiments of this research the implementation of the tracking algorithm involves two steps. The first step involves loading the desired video frame, computing the edge map after removing the noise (only with grey level video frames, with line drawing video frames mostly there is no need), computing the GGVF field, giving the initial contour by the user, inputting the appropriate values of the GGVF snake's parameters, and performing the GGVF snake deformation for detecting the desired object in the current video frame. The second step involves taking the resulting contour in the previous video frame as initial contour to the current video frame.

With respect to a line drawing starfish, the algorithm succeeded in detecting and tracking it, see Fig (2) and Fig (3). The values of GGVF snake's parameters of the tracking operation are shown in Table (1).

With grey level video frames, the algorithm also succeeded in detecting and tracking the sun and the jellyfish. See Fig (4), Fig (5), Fig (6), and Fig (7). The values of GGVF snake's parameters of the tracking operation of the sun and the jellyfish are shown in Table (2) and Table (3) respectively.

4.2 Complex Objects

The object tracking algorithm was applied on three complex objects, which are a fish, a rally car, and a human. The algorithm succeeded in detecting and tracking the three objects. See Fig (8), Fig (9), Fig (10), Fig (11), Fig (12), Fig (13), Fig (14), and Fig (15). Fig (11) and Fig (12) show the tracking operation of the rally car in two different scenes, where Fig (11) shows the tracking operation of the rally car in its straight movement, and Fig (12) shows that the tracking algorithm can handle the rotation of the rally car, where the

appearance and shape are changing. Fig (14) and Fig (15) also show the tracking operation of the human in two different scenes, where Fig (14) shows the tracking operation of a human but no entire body, and Fig (15) shows that algorithm somewhat can track the entire human body. The values of GGVF snake's parameters of the tracking operation of the fish, the rally car, and the human are shown in Table (4), Table (5), Table (6), Table (7) and Table (8).

5. Important Issues about the Implementation of Tracking Algorithm

This research used the Object Tracking Using Generalized Gradient Vector Flow focuses on the following items:-

5.1 Comparison between Motion and Contour-Based Approaches

As mentioned before, tracking object using motion-based approach requiring fixed camera, making a background model, and then detecting the moving object for each incoming video frame by using the subtraction methods. In addition to the previously mentioned shortcomings of the subtraction methods, acquiring the background model, on the other hand, is more complicated. The most straightforward approach would be to simply set up the camera, empty the scene of any moving objects, and take a snapshot. Although this approach is simple, it is always impractical in real scenes because backgrounds can change over time, it can be difficult to empty a scene, lighting can change subtly, and the camera position can drift. In contour-based approach, we don't experience the aforesaid problems, where the initial contour is applied on the desired object in first video frame. The resulting contour is taken as initialization of the next video frame and so on. Fig (16) and Fig (17) show the difference between the two approaches.

5.2 Tracking Objects in the Frames that Have Significant Motion

In section (4), the object tracking algorithm succeeded in detecting and tracking the simple and complex objects. But as shown in the figures of that section, the frames of all experiments were successive. *So, Can this algorithm track the objects just in the frames that have significant motion*?.

The object tracking algorithm was applied on the same objects (grey level frames) of section (4). With simple objects the algorithm performance was perfect; see Fig (18) and Fig (19). The values of GGVF snake's parameters of the tracking operations are shown in Table (9) and Table (10) respectively. With complex objects, the algorithm performance was not promising, so the video frames must be successive, but the promotive thing is, the contour iteration no. greatly shrinks in each iteration, and the rest of GGVF snake's parameters mostly don't change as previously shown in the Tables (4, 5, 6, 7, and 8) of section (4.2).

5.3 Effects of Simplicity and Complexity of the Background

When applying the initial contour around the desired object, the algorithm works well for cases where the background is simple. When the background is complex (the background together with object or there are strong gradients near the desired object), the algorithm provides false contour, see Fig (20). *This problem can be solved by making the initial contour very close to the*

boundary of the desired object, as previously shown in Fig (15).

6. Conclusions

It is concluded that the Motion-based approach is a full-automatic and relies on very strong assumptions like static camera with moving objects of interest or certain class of objects identifiable by templates. The lack of these strong assumptions makes it harder to achieve a high tracking quality in some cases so that more user interaction is required. GGVF snake is a semi-automatic; objects tracking using it avoid us these assumptions in addition to the making of the background model and its requirements, and the problems of the subtraction methods. Also it is found that the most important advantage , the GGVF snake don't work on the entire domain of the video frame but only the information closer to the boundary of the video object, so it provides more time.

Also it is found that GGVF snake efficiency relies on two important sides. The first side, is the quality of the video frame, where with the high quality video frame, the algorithm gives amazing tracking results. The second side is the GGVF snake's parameters, when the user gives the appropriate values, the snake works very well. With respect to the experiments of this research, the values of the parameters were as follows:-

With line drawing video frames, the value of K was 0.04 and with gray level video frames, was 0.16. The GGVF iteration no. with all experiments was 80. The Alpha (α) value with simple objects was 0.05 and with complex objects was (0.01 to 3.5). The Beta (β) value in all experiments was 0, and the Contour iteration no. depends on the shape of object.

References

[1]. Prem Kumar A., Rickesh T. N., Venkatesh Babu R., and Hariharan R., "***Object tracking using Radial basis function networks***", Indian Institute of Technology Bombay, National Institute of Technology Karnataka, Surathkal, Video analytics consultant, and Junior scientist, Flosolver, 2009.

[2]. Ratheesh K., Santhosh Kumar G., and Rao S., "***Motion Segmentation and Meanshift Assisted Contour Refinement for Airborne Video***", Cochin University of Science and Technology, and Honeywell Technology Solutions Lab, 2008.

[3]. Pilet J., Strecha C., and Fua P., "***Making Background Subtraction Robust to Sudden Illumination Changes***", Ecole, Polytechnique Federale de Lausanne, Switzerland, 2008.

[4]. Sheikh Y., Kanade T., and Javed O., "***Background Subtraction for Freely Moving Cameras***", Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, and Object Video Inc., Reston, VA 20191, 2009.

[5]. Yilmaz A., Javed O., and Shah M., "***Object Tracking: A survey***", ACM Computing Surveys, Vol 38, No. 4, Article 13, Dec. 2006.

[6]. Yang X., Latecki L. J., and Liu H., "***Contour-Based Object Detection as Dominant Set Computation***", Temple University, Philadelphia, and National University of Singapore, 2010.

[7]. Li C., Liu J., and Fox D. M., "***Segmentation of Edge Preserving Gradient Vector Flow: An Approach Toward***

Automatically Initializing and Splitting of Snakes", IEEE Trans. On Computer Vision and Pattern Recognition, 1063-6919, 2005.

[8]. Kass M., Witkin A., and Terzopoulos D., "***Snakes: Active Contour Models***", International Journal of Computer Vision, 3, 321-331, 1987.

[9]. Xu C. and Prince J. L., "***Snakes, Shapes, and Gradient Vector Flow***", IEEE Trans. On Image Processing, vol. 7, No. 3, 359-369, Mar. 1998.

[10]. Xu C. and Prince J. L., "**Generalized *Gradient Vector Flow External Forces for Active Contours***", Elsevier Science, Signal Processing 71, 131-139, 1998.

Table (1): GGVF snake's parameters of the starfish tracking operation with K=0.04, GGVF iteration no. = 80, and α = 0.05.

Frame no.	Contour iteration no.
85	60
86	20
87	10
88	8
89	3
90	3

Table (2): GGVF snake's parameters of the sun tracking operation with K=0.16, GGVF iteration no. =80, and α = 0.05.

Frame no.	σ	Contour iteration no.
361	1.5	27
362	1.5	10
363	1	5
364	1	4
365	1	3
366	1	2

Notice: The value of β in all experiments of this research was zero, so, it isn't mentioned in its tables.

Table (3): GGVF snake's parameters of the jellyfish tracking operation with σ=1, K=0.16, GGVF iteration no. =80, and α = 0.05.

Frame no.	Contour iteration no.
25	40
26	15
27	12
28	12
29	25
30	15

Table (4): GGVF snake's parameters of the fish tracking operation with K=0.16, GGVF iteration no. =80, and α = 0.5.

Frame no.	σ	Contour iteration no.
525	2.5	9
526	1	5
527	0	2

Table (5): GGVF snake's parameters of the rally car tracking operation during its straight movement with K=0.16, and GGVF iteration no. =80.

Frame no.	σ	α	Contour iteration no.
1	3.5	0.5	10
2	1	0.01	3
3	1	0.5	2

Table (6): GGVF snake's parameters of the tracking operation of the rally car during its rotation with σ=3.5, K=0.16, GGVF iteration no. =80, and α = 0.25.

Frame no.	Contour iteration no.
37	35
38	5
39	5

Table (7): GGVF snake's parameters of the human tracking operation (no entire body) with K=0.16,GGVF iteration no. =80, and α = 0.05.

Frame no.	σ	Contour iteration no.
610	2.5	9
611	1.5	1
612	2.5	2

Table (8): GGVF snake's parameters of the tracking operation of the entire human body with σ=3.5, K=0.16, GGVF iteration no. =80, and α = 3.5.

Frame no.	Contour iteration no.
44	9
45	3
46	2

Table (9): GGVF snake's parameters of the sun tracking operation in the video frames that have significant motion with σ=1.5, K=0.16, GGVF iteration no. =80, and α = 0.05.

Frame no.	Contour iteration no.
361	27
369	9
377	5
385	3
393	2
401	2

Table (10): GGVF snake's parameters of the jellyfish tracking operation in the video frames that have significant motion with K=0.16, GGVF iteration no. =80.

Frame no.	σ	α	Contour iteration no.
58	2.5	0.25	15
62	1	0.25	8
66	1	0.05	15
70	1	0.05	15
74	1.5	0.05	25
78	1	0.05	50

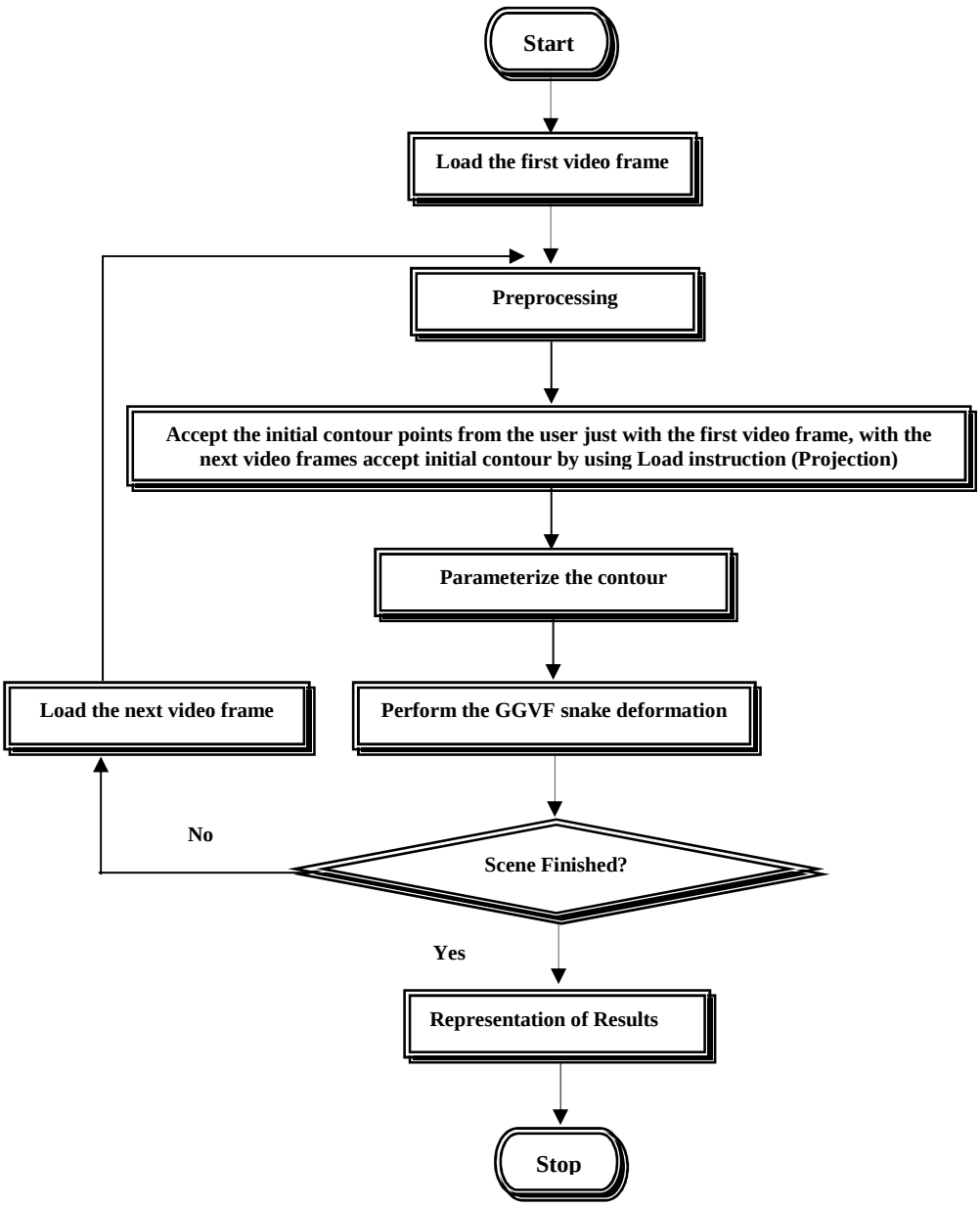


Figure (1): Overview of the implemented algorithm.

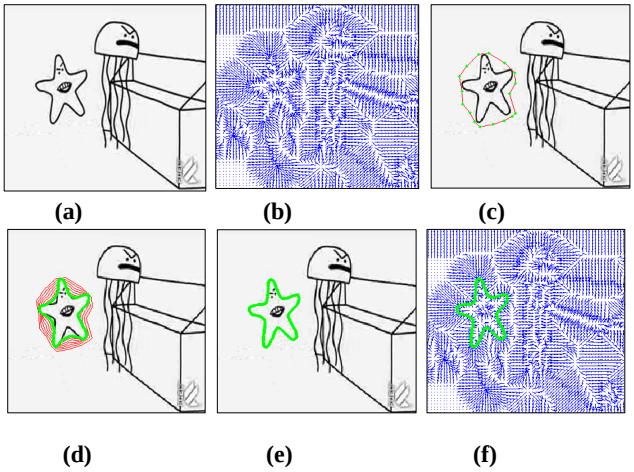


Figure (2): Detection operation of the starfish in the first frame of the scene, (a) is the first frame of the scene, (b) is the GGVF force field, (c) is the initial contour, (d) is the initial contour deformation, (e) and (f) are the final contour.

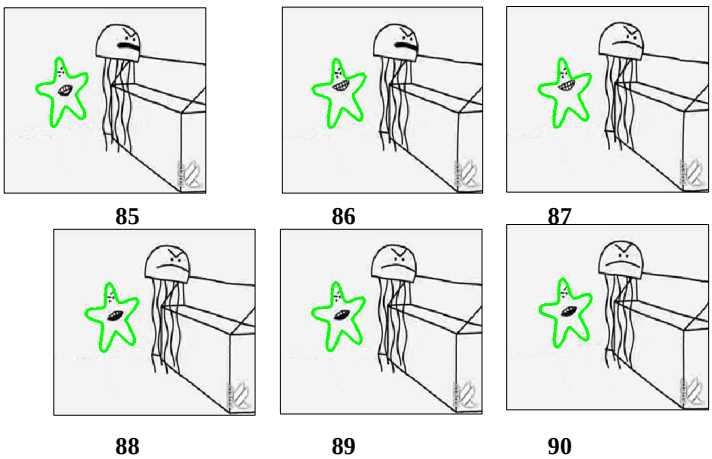


Figure (3): The tracking operation of the starfish in the scene.

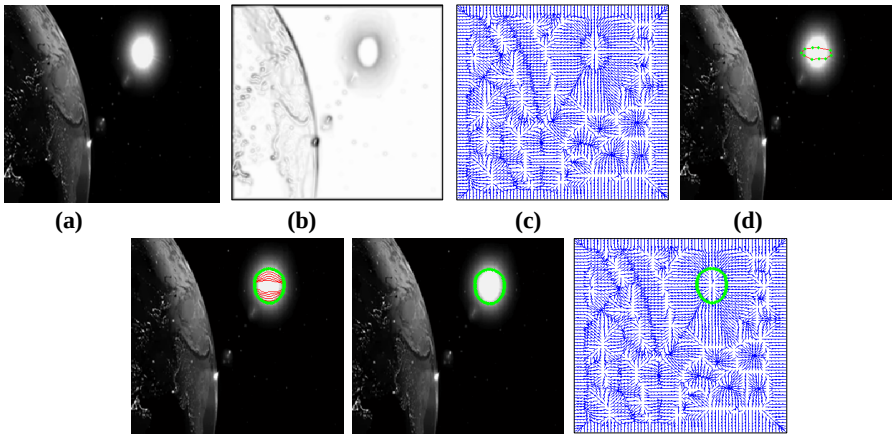


Figure (4): Detection operation of the sun in the first frame of the scene, (a) is the first frame of the scene, (b) is the edge map of the first frame, (c) is the GGVF force field, (d) is the initial contour, (e) is the initial contour deformation, (f) and (g) are the final contour.

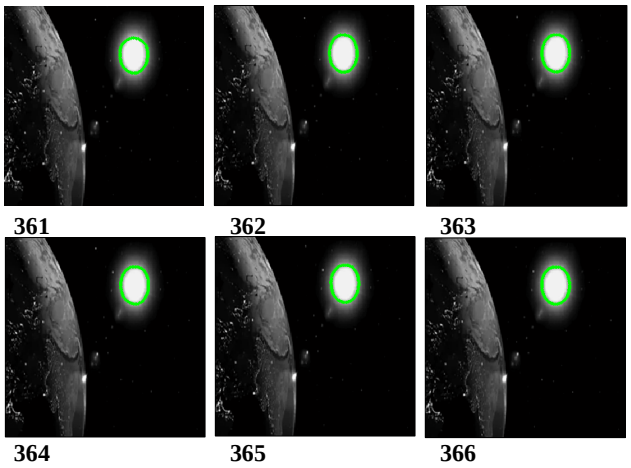


Figure (5): The tracking operation of the sun in the scene.

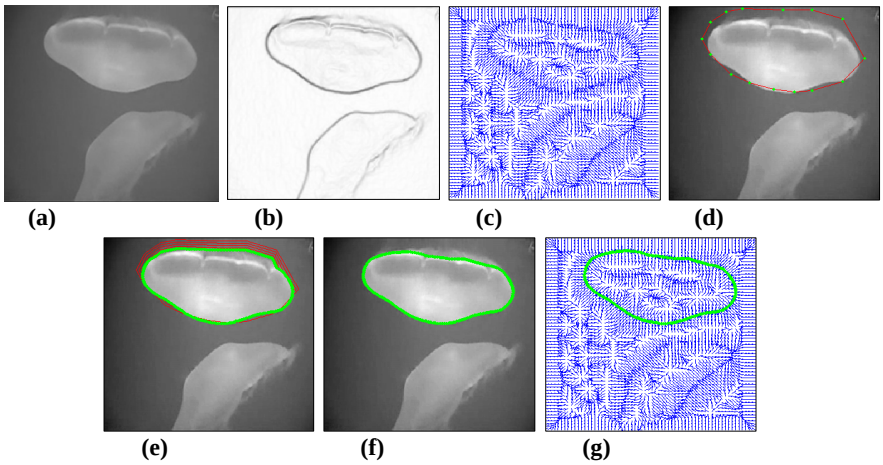


Figure (6): Detection operation of the jellyfish in the first frame of the scene, (a) is the first frame of the scene, (b) is the edge map of the first frame, (c) is the GGVF force field, (d) is the initial contour, (e) is the initial contour deformation, (f) and (g) are the final contour.

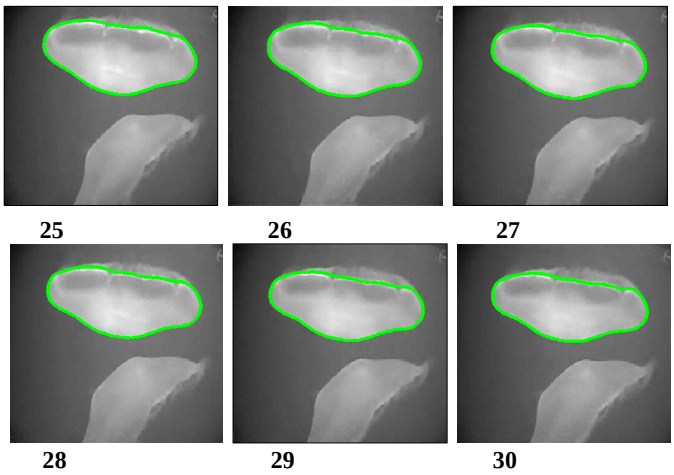


Figure (7): The tracking operation of the jellyfish in the

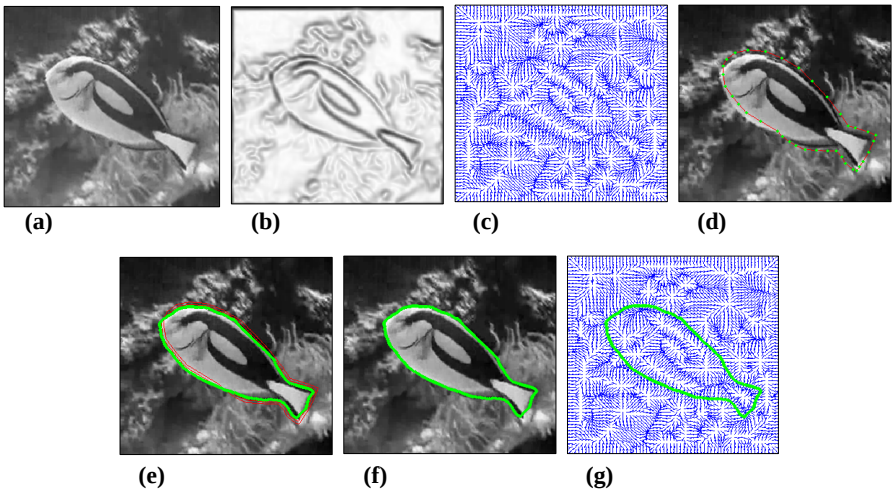


Figure (8): Detection operation of the fish in the first frame of the scene, (a) is the first frame of the scene, (b) is the edge map of the first frame, (c) is the GGVF force field, (d) is the initial contour, (e) is the initial contour deformation, (f) and (g) are the final contour.

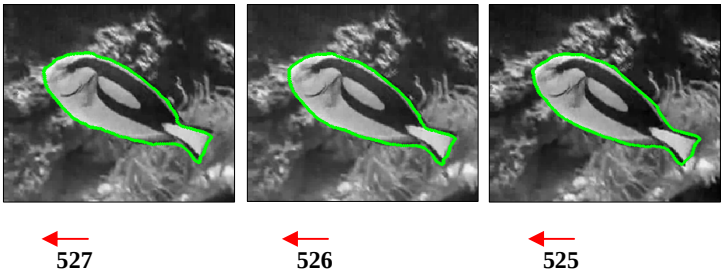


Figure (9): The tracking operation of the fish in the scene (red arrows indicate the motion direction).

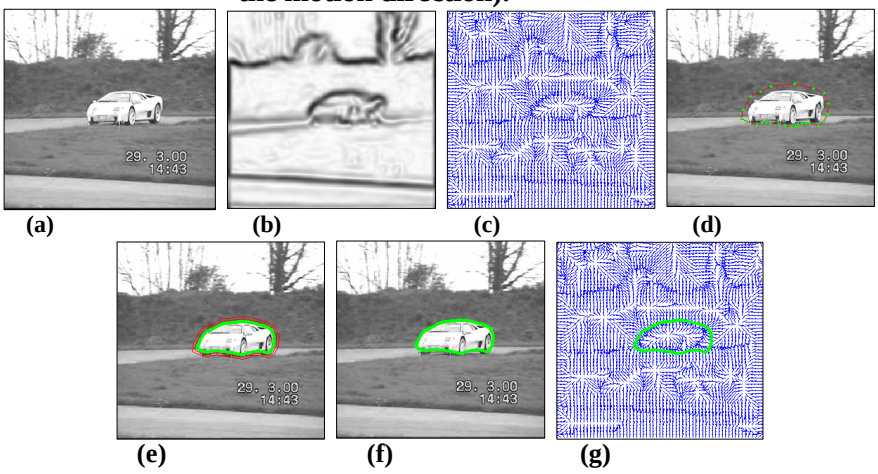


Figure (10): Detection operation of the rally car in the first frame of the scene, (a) is the first frame of the scene, (b) is the edge map of the first frame, (c) is the GGVF force field, (d) is the initial contour, (e) is the initial contour deformation, (f) and (g) are the final contour.

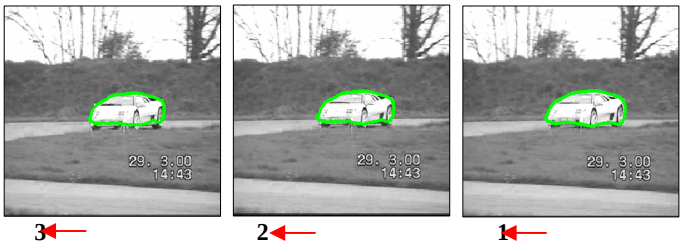


Figure (11): The tracking operation of the rally car in the scene, (red arrows indicate the motion direction).

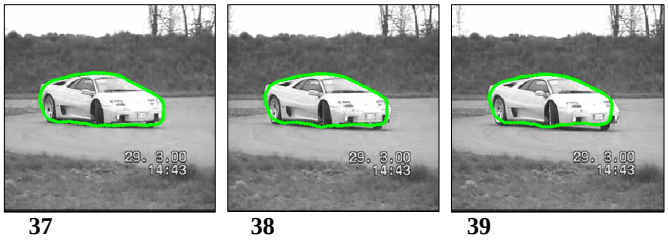


Figure (12): The tracking operation of the rally car during its rotation in the scene.

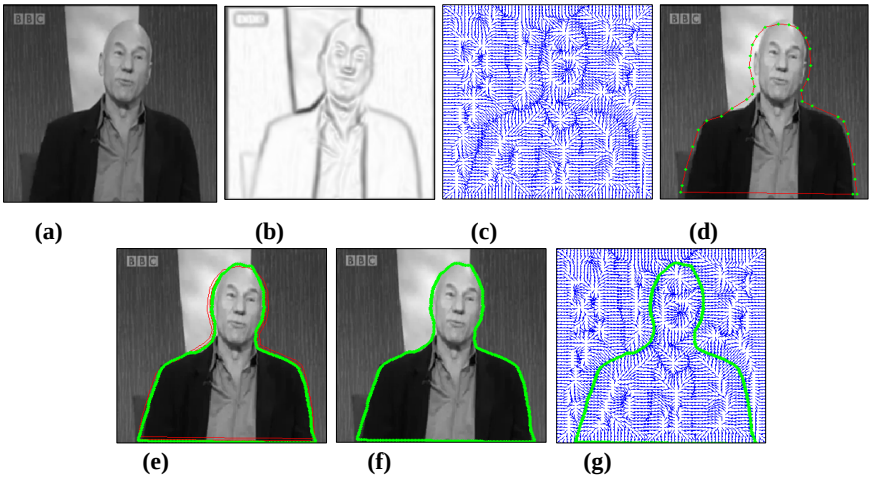
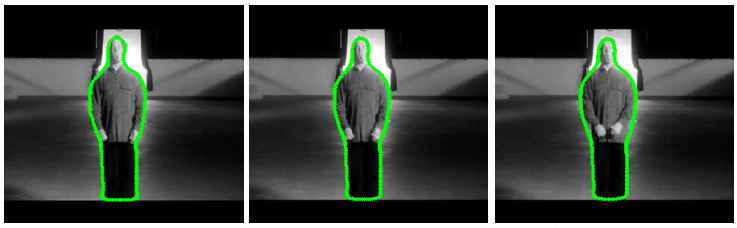


Figure (13): Detection operation of the human in the first frame of the scene, (a) is the first frame of the scene, (b) is the edge map of the first frame, (c) is the GGVF force field, (d) is the initial contour, (e) is the initial contour deformation, (f) and (g) are the final contour.

610 611 612

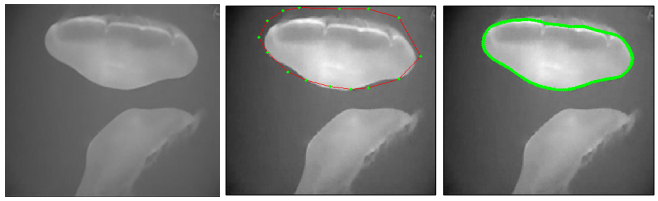
Figure (14): The tracking operation of the human (no entire body) in the scene.



44 45 46

Figure (15): The tracking operation of the entire human body in the scene.

Figure (16): The tracking operation of the jellyfish using Motion-based approach, (a) is the first frame of the scene, (b) is the background model (inside water), and (c) is the detection of the jellyfish using color-based subtraction.



(a) (b) (c)

Figure(17): The tracking operation of the jellyfish using Contour-based approach, (a) is the first frame of the scene, (b) is the initial contour, and (c) is the final contour.

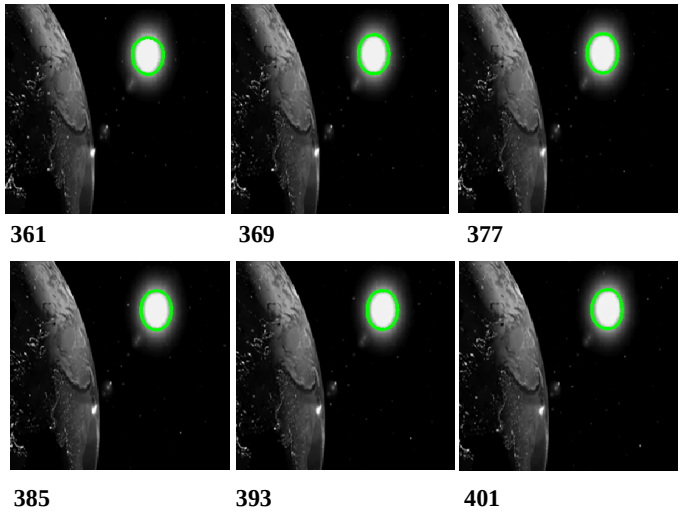


Figure (18): Tracking operation of the sun in the video frames that have significant motion.

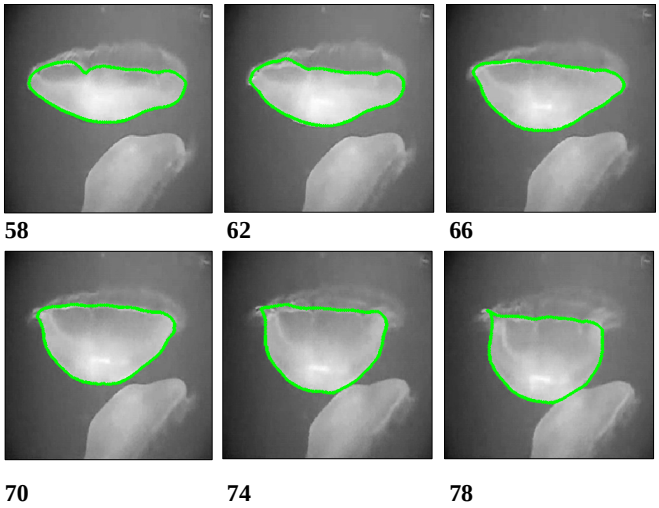


Figure (19): Tracking operation of the jellyfish in the video frames that have significant motion.

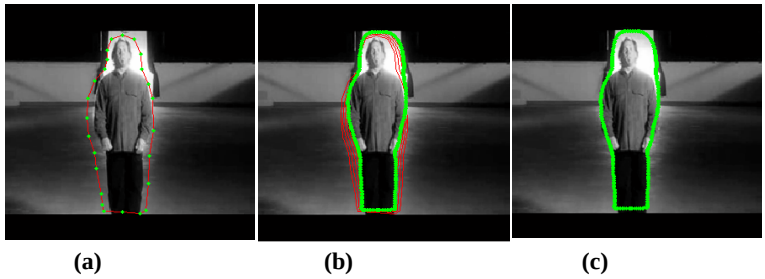


Figure (20): Effects of the strong gradients on the contour deformation, (a) is the initial contour, (b) is the initial contour deformation, and (c) is the false resulting contour because of the strong gradients found near the man's head.