Fast Video Object Tracking using Affine Invariant Normalization

Paraskevi Tzouveli, Yannis Avrithis and Stefanos Kollias
National Technical University of Athens
School of Electrical and Computer Engineering
Iroon Polytexneiou 9, 15780, Athens, Greece
email: (tpar,iavr)@image.ntua.gr

Abstract. One of the most common problems in computer vision and image processing applications is the localization of object boundaries in a video frame and its tracking in the next frames. In this paper, a fully automatic method for fast tracking of video objects in a video sequence using affine invariant normalization is proposed. Initially, the detection of a video object is achieved using a GVF snake. Next, a vector of the affine parameters of each contour of the extracted video object in two successive frames is computed using affine-invariant normalization. Under the hypothesis that these contours are similar, the affine transformation between the two contours is computed in a very fast way. Using this transformation to predict the position of the contour in the next frame allows initialization of the GVF snake very close to the real position. Applying this technique to the following frames, a very fast tracking technique is achieved. Moreover, this technique can be applied on sequences with very fast moving objects where traditional trackers usually fail. Results on synthetic sequences are presented which illustrate the theoretical developments.

1 Introduction

Object tracking is a very common problem in computer vision and image processing applications. The localization of object boundaries in a video frame and its tracking in the next frames [1],[2],[8]-[13] is a crucial issue in the materialization of a tracking method. It is therefore important and challenging to develop an approach to track objects under geometric transformations. Such an approach can allow the tracking of fast moving objects.

In this paper, a fully automatic method for fast tracking of video objects in a video sequence using affine invariant normalization is proposed. Initially, the detection of a video object is achieved using a GVF snake [3], [5]. Next, a vector of the affine parameters of each contour of the extracted video object in two successive frames is computed using affine-invariant normalization [6]. Under the hypothesis that these contours are similar, the affine transformation between the two contours is computed in a very fast way.

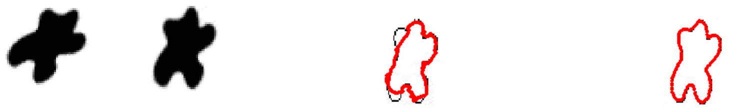
Using this transformation to predict the position of the contour in the next frame allows initialization of the GVF snake very close to the real position. Applying this technique to the following frames, a very fast tracking technique is achieved. Moreover, this technique can be applied on sequences with very fast moving objects where traditional trackers usually fail. Results on synthetic sequences are presented which illustrate the theoretical developments.

2 Problem Statement

The basic idea in active contour models or snakes is to evolve a curve, subject to constraints from a given image, in order to detect objects in that image [4]. For instance, starting with a curve around the object to be detected, the curve moves toward its interior normal and has to stop on the boundary of the object. Usually, using the contour of the previous frame, we can estimate the contour of the object of the next frame.

However, the main drawback of the active contours methods is the position estimation of the object contour if it is moved very fast. If the object is fast the contour can not be estimated and can be lost for the rest frames of the sequence. In addition, active contours methods still suffer from the sensitive of initial parameters while the computation is an expensive process which makes difficult the implementation of active contours in real time applications.

On the other hand, the active contours can also be initialized across the object boundary and if the initialization is closed to the real boundary, the object boundary is quickly localised. Our method uses this feature of active contours, having an initialization contour very close to the object boundary. The initialization contour can be acquired applying the proposed affine normalization transformation (Section 5) to the previous contour of a video sequence, in order to compute the initialization of the next contour. The initialization of the next contour can be applied very quickly, providing an accurate estimation of the object contour of the next frame.



a) Two successive frame of a sequence

b) Apply active contours method

c) Apply active contours and normalization affine transformation

Figure 1: Finding the contour of next frame knowing the previous contour

Figure 1a depicts two successive frames of a fast moving object of a video sequence. The contour is shown as dash line in Figure 1b and 1c image depicts the estimation of the next contour knowing the previous contour. It is obvious (Figure 1b) that using only an active contour method taking as input the contour of the previous frame, the estimation of the object is not accurate. Figure 1c represents the estimation of the contour according to affine normalization transformation and it is very close to the real contour of the object. This simple example proves that the proposed method can be used for very fast moving video objects tracking.

3 Active Contours and Gradient Vector Flow

The automatic localization of objects of interest in an image or video sequence is a challenging task. Objects of interest are presented in many techniques with active contours [3]-[5]. In the proposed method, we use the GVF snake [5] in order to extract the objects of interest from a video sequence (Video Object) and accelerate the proposed VO tracking procedure.

The basic idea in active contour models is to evolve a curve, subject to constraints from a given image, in order to detect objects in it. Starting with this curve within the image domain and moving it under the influence of internal and external forces derived from image data, we can acquire the boundaries of the objects of interest.

A new external force for active contours, called Gradient Vector Flow, has been proposed in [5], trying to tackle problems that are associated, with initialization and poor convergence, to boundary concavities. The GVF snake begins with the calculation of a field of forces, called the GVF forces, over the image domain. The GVF forces are used to drive the snake, modeled as a physical object having a resistance to both stretching and bending, toward the boundaries of the object.

The GVF forces are calculated by applying generalized diffusion equations to both components of the gradient of an image edge map. Because the GVF forces are derived from a diffusion operation, they tend to extend very far away from the object so that snakes can find objects that are quite far away from the snake's initial position. This same diffusion creates forces which can pull active contours into concave regions.

4 Affine invariant normalization

Normalization is a procedure that enables comparison between different images of the same object, as well as of different objects, since distances are always measured in a normalized frame [7],[7] and [12]. In the proposed method, affine-invariant normalization is applied to the object curves in order to make them affine invariant, and thus appropriate for curves matching. For this purpose, a set of transformations is applied to each point of the contour composing the object curve.

For the sake of simplicity, let us assume a synthetic video sequence, the first frame of which is depicted in Figure 2. Figure 1 also illustrates its contour which has been extracted using GVF snake (section 2).

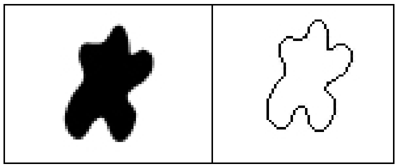


Figure 2: First frame of the synthetic video sequence and its contour

Firstly, the obtained contour is re-sampled in order to be constituted of a fixed number of equidistant points without losing its original shape. Equation 1 represents the N points of the contour of the kth frame of the synthetic sequence:

$$c_k = [x_i, y_i]^T,\ \ i = 0,1,\ldots N-1,\ \ k = 1,2\ldots K \qquad \text{(Eq.1)}$$

For each contour, the (p-q) order moments are given by Eq. 2:

$$m_{pq}(c) = \frac{1}{N}\sum_{i=0}^{N-1}(x_i^p y_j^q) \qquad \text{(Eq. 2)}$$

Moments of order up to two are used for the construction of the normalized curve. The orthogonalization procedure comprises a set of linear operations (translation, scaling, and rotation) that do not depend on the selected starting point of the closed curve. The orthogonalization of the curve can be acquired:

$$n_a(c) = R_2 \cdot B \cdot R_1 \cdot A \cdot (c - T) \quad \text{(Eq. 3)}$$

where $T = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix} = \begin{bmatrix} m_{10}(c) \\ m_{01}(c) \end{bmatrix}$ is the translation of the initial resample curve, $A = \begin{bmatrix} \sigma_{1x} & 0 \\ 0 & \sigma_{1y} \end{bmatrix} = \begin{bmatrix} 1/m_{20}(c_I)^{1/2} & 0 \\ 0 & 1/m_{02}(c_I)^{1/2} \end{bmatrix}$ is the scaling factor which is applied to the translated curve, while $R_1 = \begin{bmatrix} \sin(r_{1x_2}) & -\cos(r_{1y_2}) \\ \cos(r_{1x_2}) & \sin(r_{1y_2}) \end{bmatrix}$ is the rotation factor applied to the curve, already scaled using factor σ_1. In the proposed method, $r_1 = \pi/4$. The matrix $B = \begin{bmatrix} \sigma_{2x} & 0 \\ 0 & \sigma_{2y} \end{bmatrix} = \begin{bmatrix} 1/m_{20}(c_{II})^{1/2} & 0 \\ 0 & 1/m_{02}(c_{II})^{1/2} \end{bmatrix}$ denotes the second scaling factor, applied in the curve after the counterclockwise by π/4 rotation while $R_2 = \begin{bmatrix} \sin(r_{2x_4}) & -\cos(r_{2y_4}) \\ \cos(r_{2x_4}) & \sin(r_{2y_4}) \end{bmatrix}$ is the rotation matrix, applied to the curve, already scaled using the factor σ_2 with $r_2 = [(a_1 + a_{N-1})/2]\bmod\pi$ (where $a_1 = (x_1, y_1)$ and $a_{N-1} = (x_{N-1}, y_{N-1})$ the average value of Fourier phases).

At this point we have achieved the reduction of affine transformations to orthogonal ones and we need a transformation invariant to rotation and reflection. The overall normalization (orthogonalization and normalization) is now affine invariant and the starting point normalization is necessary since the rotation normalization depends on the starting point. Thus, a standard circular shift is defined using the first and last Fourier phases: $p(z) = [(N/4\pi)\cdot(a_1 - a_{N-1})]\bmod(N/2)$ and the opposite shift is applied in order to normalize the curve $n_p(z) = S_{-p(z)}(z)$.

The presented normalization method transforms the object contours in order to make them affine invariant 5. To sum up, an affine normalization of a contour can be achieved applying normalization after the orthogonalization of the contour:

$$n_p(n_a(c)) = S_{-p}(R_2 \cdot B \cdot R_1 \cdot A \cdot (c - T)) = S(R_2 \cdot B \cdot R_1 \cdot A \cdot (c - T), -p) \quad \text{(Eq. 4)}$$

5 Tracking VO using normalization affine transformation

The tracking method that we propose in this section provides a fully automatic affine invariant method for fast tracking video object in a video sequence. Initially, a GVF snake is implemented in the first frame of the sequence in order to provide an initial estimation of the video object contours (c_1). Having as pattern for the GVF snake the contour of first frame of the video sequence, the contour c_2, of the second frame can then be acquired.

In the next step, applying affine normalization (Eq. 4) to each contour, the $a_1(c_1) = n_p(n_a(c_1))$ and $a_2(c_2) = n_p(n_a(c_2))$ for contour c_1 and c_2 respectively, is obtained. Now, we can define the vector of normalized affine parameters which can be represented as $P = \{D, g, s\}$ where $D = R_2 \cdot B \cdot R_1 \cdot A$ is the rotation-scaling deformation matrix while parameter $g = -D \cdot T$ represents the video object translation and $s = -p$ is a shifting parameter. The overall normalization affine is:

$$a(c) = n_p(n_a(c)) = S((Dc + g), -s) \qquad \text{(Eq.5)}$$

After normalization of each contour of the first two sequential video objects, accepting that these contours are almost equal, we can assume that

$$a_1(c_1) = a_2(c_2) \xrightarrow{\text{Eq.5}} S(D_1c_1 + g_1), -s_1) = S(D_2c_2 + g_2), -s_2)$$
$$\Rightarrow D_2c_2 + g_2 = S(S(D_1c_1 + g_1), s_1 - s_2)$$
$$\Rightarrow c_2 = D_2^{-1} \cdot (S((D_1c_1 + g_1),\ s_1 - s_2) - D_2^{-1}g_2) \ \text{ with } D_2 \neq 0$$
$$\Rightarrow c_2 = \cdot S((D_2^{-1}D_1c_1 - D_2^{-1}(g_1 - g_2)),\ s_1 - s_2)) \quad \text{(Eq.6)}$$

Now, we can define the transformation:

$$a_2^{-1} \circ a_1 = \left\{\ D_2^{-1}D_1,\ g_1 - g_2,\ s_1 - s_2\ \right\} = a_{1\to 2} \quad \text{(Eq. 7)}$$

Eq.7 will be examined in order to verify that knowing the previous contour, the computation of the next contour of the video object is possible by applying this transformation. For this purpose, the steps of our algorithm are presented.

The algorithm requires, as input, the contours of the video object contained in the first and second frame. It can be computed applying a GVF snake to the first and second frame of the video sequence.

Then, the affine normalization (Eq. 4) is applied to the first and second frame contour, taking $a_1(c_1) = n_p(n_a(c_1))$ and $a_2(c_2) = n_p(n_a(c_2))$ respectively. The vectors of normalized affine parameters of the video object from first and second frame can now be computed: $P_1 = \{D_1, g_1, s_1\}$ and $P_2 = \{D_2, g_2, s_2\}$.

Having the vectors P_1, P_2 of normalized affine parameters, the transformation $a_2^{-1} \circ a_1$ can be computed using Eq. 7.

Supposing that the contour of the video object of the third frame, follows approximately the same transformation ($a_{1\to 2}$) as the first and second contour, the application of this transformation places the contour of the third video object close enough to the real position. Then, applying a GVF snake which takes as input the contour that has been achieved applying the transformation to the second contour, the right estimation of the third contour can be achieved very quickly: $c_3 \approx s(a_{1\to 2}(c_2))$ (Eq.8)

Having the next contour c_3 the transformation $a_{2\to 3}(c_3)$ is computed. Then, applying the GVF snake to the next video frame a very close estimation of the video object boundary that the fourth frame included can be achieved $c_4 \approx s(a_{2\to 3}(c_3))$.

Following the same procedure for the next frame c_i, the transformation $a_{i-1\to i}(c_i)$ for the video object of next frame is computed and estimation of the position of the next video object is calculated applying the GVF snake $s(a_{i-1\to i}(c_i))$.

Applying the proposed method to the following frames, a fast tracking technique can be achieved. The method discussed in this paper was tested on several video sequences with very fast moving objects and a series of experiments has been performed. Firstly, the accuracy of the affine normalization method is examined using a synthetic video sequence.

Figure 3 illustrates the first five successive frames of a synthetic video sequence. The video object of the second frame has been rotated by $\varphi = 30°$ while in third frame it has been translated. In fourth frame the video object has been scaled by 5% and in the last frame it has been rotated by $\varphi' = -30°$ and translated.

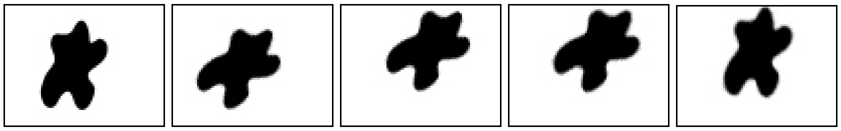


Figure 3: Synthetic video sequence

According to the proposed method, using the GVF snake method, the contours of the first two frames can be extracted. Then, we compute the vectors P_1, P_2 of normalized affine parameters as well as the transformation $a_{1\to 2} = a_2^{-1} \circ a_1$. In order to measure the similarity between curves $c_i = s(a_{i-2\to i-1}(c_{i-1}))$ $i \geq 3$ and c_{real_i}, we use the Euclidean distance is used. The value of similarity of the contours approaches 97%.

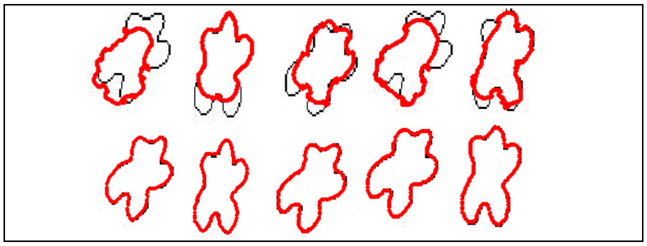


Figure 4: Fast video object tracking with or without affine invariant normalization

In Figure 4, $c_i = s(c_{i-1})$ is depicted with a dash-line in first line using only GVF snake while the estimation of $c_i = s(a_{i-2 \to i-1}(c_{i-1}))$ is depicted in the second line with a dash-line contour too. It is obvious that the estimation of the contours which uses GVF snake and affine normalization is closer to thin-line c_{real_i} that uses only GVF snake.

The result that the comparison between the proposed method and the GVF snake method gives is depicted in Figure 5. It can be seen that our algorithm reduces the number of GVF snake iteration more than 60%. In this experiment are used twenty successive frames of the synthetic video sequence.

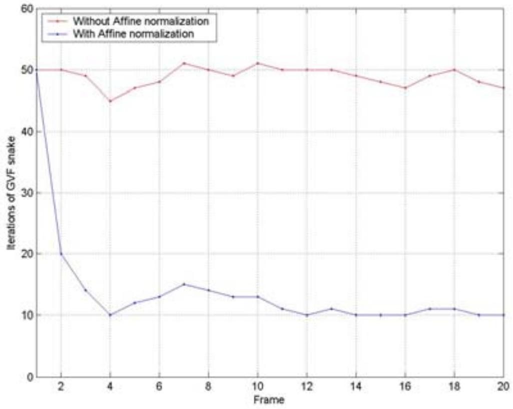


Figure 5: Iterations of GVF snake with and without affine normalization

Conclusion

In this paper, we have proposed a method for tracking using affine invariant normalization for very fast moving objects. The video object detection for the two first successive frames is achieved using a GVF snake. Next, using affine-invariant normalization, a vector of the affine parameters of each extracted video object is computed. Assuming that these contours are similar, we compute the affine transformation between these contours. Using this transformation to predict the position of the contour in the next frame allows initialization of the GVF snake very close to the real position. Experimental results suggest that our algorithm have great potential for tracking video objects with very fast moving objects where traditional trackers usually fail.

Reference

1. J. Guo, J. Kim, and C. Kuo, “An interactive object segmentation system for MPEG video,” IEEE Proceedings of International Conference on Image Processing, Kobe, Japan, 1999.
2. C. Gu and M. Lee, “Semiautomatic segmentation and tracking of semantic video objects,” IEEE Trans. Circuits and Systems for Video Technology, Vol. 8, No. 5, pp.574-584, September, 1998.
3. M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active Contour models,” Int’l J. Computer Vision, vol. 1, pp. 312-333, 1988.
4. H. S. Ip and S. Dinggang, “An Affine-Invariant Active Contour Model (AI-Snake) for Model-Based Segmentation,” Image and Vision Computing, 16(2), pp. 135-146, 1998.
5. C. Xu and J.L.Prince, “Gradient Vector Flow: A New External Force for Snakes” IEEE Proceedings Conference on Computer Vision and Pattern Recognition, pp 66-71, 1997
6. Y. Avrithis, Y.Xirouhakis, S. Kollias, “Affine-invariant curve normalization for object shape representation, classification, and retrieval,” Machine Vision and Applications, 13, pp. 80-94, 2001.
7. Y.S. Abu-Mostafa and D. Psaltis, “Image Normalization by Complex Moments,” IEEE Trans. Pattern Analysis and Machine Intelligence vol. 7, pp. 46-55, Jan. 1985.
8. F. Leymarie and M. D. Levine, “Tracking deformable objects in the plane using an active contour model,” IEEE Trans. Pattern Anal. Machine Intell., vol. 15, pp. 617–634, 1993
9. A. Blake, M. Isard and D. Reynard, “Learning to track the visual motion of contours”, Journal of Artificial Intelligence, vol. 10, pp. 323-380, 1997
10. C. Tomasi T. Kanade “Shape and Motion from Image Streams: a Factorization Method”, Full Report on the Orthographic Case, March 1992, Cornell TR 92-1270 and Carnegie Mellon CMU-CS-92-104A.
11. J. Shi and C. Tomasi, “Good features to track”, IEEE Proceedings Conference on Computer Vision and Pattern Recognition, pp 593-600, 1994.
12. Y. Avrithis, Y. Xirouhakis and S. Kollias, "Affine-Invariant Curve Normalization for Shape-Based Retrieval, " in Proc. of 15th International Conference on Pattern Recognition (ICPR '00), Barcelona, Spain, September 2000, pp. 1015-1018.
13. Y. Xirouhakis, Y. Avrithis and S. Kollias, "Image Retrieval and Classification Using Affine Invariant B-Spline Representation and Neural Networks," in Proc. of IEE Colloquium on Neural Nets and Multimedia, London, UK, October 1998, pp. 4/1-4/4.