Tracking complex primitives in an image sequence

Benedicte Bascle [◇], Patrick Bouthemy [◇◇], Rachid Deriche [◇], Francois Meyer [◇◇]

[◇] INRIA, B.P. 93, 06902 Sophia-Antipolis Cedex, France
[◇◇] IRISA/INRIA, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France
email: bascle@sophia.inria.fr

Abstract

This paper describes a new approach to track complex primitives along image sequences - integrating snake-based contour tracking and region-based motion analysis. First, a snake tracks the region outline and performs segmentation. Then the motion of the extracted region is estimated by a dense analysis of the apparent motion over the region, using spatio-temporal image gradients. Finally, this motion measurement is filtered to predict the region location in the next frame, and thus to guide (i.e. to initialize) the tracking snake in the next frame. Therefore, these two approaches collaborate and exchange information to overcome the limitations of each of them. The method is illustrated by experimental results on real images.

1 Introduction

Object tracking is an important clue to dynamic scene analysis. Among other methods, snake-based contour tracking [10] [6] [7] [15] [2] and region-based tracking (motion-based region segmentation) [5] [13] [14] are useful to track more complex and global primitives than points or line segments. Moreover these methods are quite complementary. Indeed snake-based contour tracking is rather quick and efficient due to the active behaviour of snakes. It is also precise in the extraction of edges. However, it requires a proper initialization and can thus treat only slow motions. Moreover these methods use only edge information. On the contrary, region-based tracking (motion-based region segmentation) exploits the full region information, and thus estimates region motion quite precisely. Besides, this multi-resolution method is not very sensitive to large displacements, nor to partial occlusion. Furthermore, it is able to detect moving objects without any initialization. However, it gives rather rough estimations of region boundaries.

This paper describes an original combination of these two approaches, which overcomes some of the limitations of each of them. First, the moving object is detected by a motion-based segmentation algorithm [4]. Then a snake-based contour tracking algorithm is used to track and segment the object along the image sequence. It is based on B-spline snakes with motion constraints [2]. Thereafter, the motion of the region delineated by the snake is estimated using a region-based motion analysis approach [13]. It consists in a dense and multi-resolution estimation of an affine velocity field over the region. Temporal filtering is then applied to predict the position of the region in the next frame and thus to initialize the snake tracker in the next image. As a result, snake-based contour tracking and region-based motion analysis strongly cooperate: motion estimation relies on the snake-based segmentation, whereas it provides to the tracking snake a prediction of the region location.

The paper is organized as follows: the first section describes the global framework of our tracking approach. The second and third sections present in more detail the modules of the approach: respectively object segmentation and tracking, and motion estimation and prediction. Finally the last section presents experimental results to illustrate and validate the approach.

2 Outline of the tracking approach

This section describes the main steps of our tracking approach - combining snake-based contour tracking and region-based motion analysis (see fig. 1):

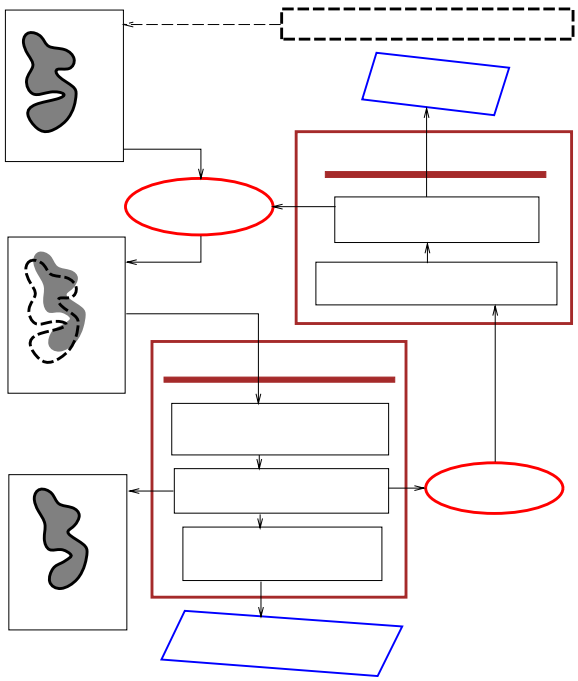


Figure 1: Combined tracking algorithm

Initialization on first image of the sequence: As snake-based tracking requires an initialization, motion-based segmentation [4] is used to detect a moving object and initialize tracking on the first image of the sequence. It generates a rough mask of the object. Then the mask outline is extracted by classical edge detection and linking. This first estimation of the object apparent contour (approximated by a B-spline curve) is improved by a free deformable B-spline

curve, so as to better fit image edges [1]. This provides an initialization to the snake-based tracking algorithm on the first image of the sequence. For the rest of the sequence, region segmentation is achieved by a snake-based contour tracking algorithm, which is more precise in the detection of edges than motion-based region segmentation.

Contour tracking by a snake with motion constraints: The region outline is thus simultaneously extracted and tracked along the image sequence by a deformable B-spline curve with motion constraints [2]. B-splines can describe various shapes and have regularization properties. And motion constraints - namely an affine displacement model for the curve - improve the speed of the deformable curve and its robustness to partial occlusion and large displacement. Moreover such a global model helps to perform a good point-to-point matching between the edges. Indeed free deformable curves have local fitting behaviours that produce erroneous matching between curves. Due to the motion model, the pointwise trajectography of an object in the image can be done. The affine 2D displacement model is interesting since it is a good approximation of the 2D apparent motion of a 3D rigid object if perspective is weak. Afterwards motion constraints are relaxed in order to refine the extracted curve and to deal with small perturbations with respect to the affine motion model (non-rigid deformations for instance). Thus this snake-based algorithm performs region segmentation and tracking. Then Kalman filtering is used to regularize the estimated trajectories of the region border points, and study the region deformations.

Region segmentation: Thus region segmentation and tracking are performed by a snake. Indeed edge-based segmentation is more precise than motion-based segmentation. However, the estimation of motion given by the snake relies only on edge information. Therefore region motion is more precisely estimated (in the lambertian hypothesis) using a region-based approach because it uses the information given by the interior of the region.

Region-based motion estimation: Given the region segmentation done by the snake, region motion is estimated using a region-based approach inspired by motion-based region segmentation [13]. A 2D affine displacement model is assumed. Its parameters are deduced from the parameters of a 2D affine velocity model. As the approach is multi-resolution and dense over the region (not limited to edge information), it is robust to noise and partial occlusion and can handle large displacements. The measures of the region affine motion thus obtained are smoothed through time using Kalman filtering.

Contour prediction: Moreover, this Kalman filtering on the region motion is used to predict the region motion and thus its position in the next frame. This prediction is then employed to initialize the snake - which tracks the region contour - in the next frame. This guides the deformable contour towards the image area where the region edge is to be expected. This improves the tracking performances of the snake algorithm, which is sensitive to initialization, especially if the contour displacement is large between two images. Thus, motion analysis, which relies on the region segmentation provided by snakes, operates back on snake tracking by supplying a prediction. Therefore our tracking approach truly and closely combines snake-based tracking and region-based motion analysis.

Remark: This algorithm is rather robust to partial occlusions. Indeed region-based motion estimation is not very sensitive to occlusion since it relies on a dense information. Moreover snakes with motion constraints perform global matching between contours, so that they are also rather robust to occlusions.

3 Object segmentation and tracking

Object segmentation and tracking is performed by a deformable B-spline curve with an affine motion constraint [2]. Indeed the B-spline curve model can describe most real-world shapes rather realistically. The affine motion constraint consists in imposing an affine displacement model on the deformable curve. Moreover the deformable B-spline curve is parametrized by the affine motion parameters [2], and not by its shape control points as done classically [12] [1]. This tracking approach performs reliable matching between the points of moving contours, due to the global motions constraints. The method also is rather quick and robust, due to the limited number of degrees of freedom. Tracking is finally completed by a refinement step, during which motion constraints are relaxed. This final step is necessary to perform accurate tracking if the affine displacement model does not exactly describe reality, and especially to deal with non-rigid objects. The equations of this deformable model are presented below:

Let $C_0(u) = (x_0, y_0)^T(u)$ be the B-spline curve corresponding to the contour of the first image. It is obtained from motion-based segmentation [4], which generates a mask of the moving object in the first image. The mask outline is extracted by an edge detection algorithm. Then it is approximated, in the least-square sense, by a B-spline curve and optimized by a deformable B-spline curve, so as to better fit image edges [12] [1]. This contour is the template shape. As a B-spline, it can be writen as a linear combination of basis functions $B_i^k(u)\big|_{i=0..m}$ - where k is the degree (plus one) of the B-spline - and control points $(Vx_{i0}, Vy_{i0})|_{i=0..m}$, as follows: $x_0(u) = \sum_{i=0}^{m} Vx_{i0} B_i^k(u)$ and $y_0(u) = \sum_{i=0}^{m} Vy_{i0} B_i^k(u)$.

The contour is then tracked - from its initial position C_0 to its position $C(u)$ in the next image - by a deformable B-spline with an affine motion model. Therefore, $C(u)$ is described as an affine transform of the template shape C_0:

$$C(u) = \begin{bmatrix} x(u) \\ y(u) \end{bmatrix} = A \begin{bmatrix} x_0(u) \\ y_0(u) \end{bmatrix} + T \qquad (1)$$

The template shape $C_0(u)$ being fixed, $C(u)$ is parametrized by the affine transform parameters. These are optimized by energy minimization until the curve $C(u)$ fits the image edge. To this end, and as the deformable curve seeks edges, its energy is written as the mean of the intensity gradient along the curve (with a minus sign): $E = -\frac{1}{|C|}\int_C |\nabla I|(x,y)du$. The energy minimization is carried out using Euler-Lagrange dynamics, the system being massless and embedded in a viscous medium. Therefore the parameters a_j of the affine transform (A, T) have the following evolution equations:

$$\gamma_{a_j} \frac{da_j}{dt} = -\frac{\partial E}{\partial a_j} = \frac{1}{|C|}\int_C \left(\frac{\partial |\nabla I|}{\partial x}\frac{\partial x}{\partial a_j} + \frac{\partial |\nabla I|}{\partial y}\frac{\partial y}{\partial a_j}\right)du \qquad (2)$$

Finally the affine motion constraints are relaxed in order to refine the estimated contour, so that the B-spline

curve becomes fully deformable, like in [12] [1]. This refinement step improves the precision of contour detection when part of the object deformations (like non-rigid deformations) cannot be described by the affine displacement model. During this refinement, errors can occur in case of partial occlusion.

The final curve extracted in this frame becomes the new template shape to be tracked to the next frame. Thus the template shape is continuously updated during tracking.

Furthermore, the trajectories of the control points - parameterizing the region contour and measured by the snake - are smoothed using Kalman filtering, so as to diminish the influence of noise. Each coordinate x of the control points is filtered independently using an α, β tracker [8]. It is a steady-state Kalman filter with a constant velocity model. For more details, see [8]. The estimate of a control point coordinate x and its velocity $\underline{x}$ are given by: $x_{t/t} = -(\alpha + \beta - 2)x_{t-1/t-1} - (1-\alpha)x_{t-2/t-2} + \alpha v_t + (-\alpha+\beta)v_{t-1}$ and $\underline{x}_{t/t} = -(\alpha+\beta-2)\underline{x}_{t-1/t-1} - (1-\alpha)\underline{x}_{t-2/t-2} + \beta(v_t - v_{t-1})$ - where v is the instantaneous measure of x.

4 Motion estimation and prediction

Given the segmentation performed by the snake , region motion is then estimated using a region-based approach inspired from motion-based region segmentation [13] [4]. Indeed such a region-based approach to motion estimation is likely to be more stable than a contour-based approach. A 2D affine displacement model is used to capture the motion of the tracked region between two successive frames. The computation of this affine displacement relies on the spatio-temporal derivatives of the intensity function and is embedded in a multi-resolution scheme:

The region evolution model As before, the region R is assumed to undergo a 2D affine displacement, with parameters $(\mathbf{A}, \mathbf{T})$. Hence every point $(x(t), y(t)) \in R$ at time t is located at $(x(t+1), y(t+1))$ at time $t+1$, with :

$$\begin{pmatrix} x \\ y \end{pmatrix}(t+1) \quad = \quad \mathbf{A}(t) \begin{pmatrix} x \\ y \end{pmatrix}(t) \quad + \mathbf{T}(t) \qquad (3)$$

The regions have been delineated by a snake and the affine displacement parameters $(\mathbf{A}, \mathbf{T})(t)$ now need to be determined. To this end, the 2-D velocity field within R is also approximated with an affine model $(\mathbf{M}, \mathbf{b})$ - such a model describes a large class of motions, and conveys information about 3D motion and structure [13]:

$$\forall (x,y) \in R, \qquad \begin{pmatrix} \underline{x} \\ \underline{y} \end{pmatrix}(t) \quad = \quad \mathbf{M}(t) \begin{pmatrix} x \\ y \end{pmatrix}(t) \quad + \mathbf{b}(t) \qquad (4)$$

If $(x,y)^T(t+1)$ is expanded in Taylor series to the first order, it gives $x(t+1) = x(t) + \delta t.\underline{x}(t)$ and $y(t+1) = y(t) + \delta t.\underline{y}(t)$ (where δt is the time step between two successive frames). From (3), (4) and above, it comes that the 2D affine displacement $(\mathbf{A}(t), \mathbf{T}(t))$ can be deduced from the 2D affine velocity field $(\mathbf{M}, \mathbf{b})(t)$ as follows (with $\mathbf{I}_2$ being the 2×2 identity matrix): $\mathbf{A}(t) = \mathbf{I}_2 + \delta t \mathbf{M}(t)$ and $\mathbf{T}(t) = \delta t \mathbf{b}(t)$.

Multi-resolution estimation of the motion parameters The motion parameters $(\mathbf{M}, \mathbf{b})(t)$ are estimated using a multi-resolution scheme[13] . First a rough estimate of the parameters is obtained at the lowest resolution. Then it is refined using the higher resolution images. This method provides accurate and robust estimates of the motion parameters, even in the case of large displacements.

Two Gaussian pyramids are built, for the images at time t and $t + \delta t$. Each level image in the pyramid is a blurred and sub-sampled (of a factor 2) version of its predecessor. The pyramid provides the delineations of the segmented region at each level.

First, the six parameters $(\mathbf{M}, \mathbf{b})$ of the region afine velocity $\mathbf{v}(x,y)$ are estimated at the lowest resolution level L, with a least-squares fit to normal flows. To do this, we use the well-known image flow constraint equation (5) [9] that relates the motion field $\mathbf{v}(x,y)$ to the spatial and temporal derivatives of the image intensity $\nabla \mathbf{I}(x,y)$ and $I_t(x,y)$. This constraint is applied at the lowest resolution L, since it assumes that image motion is small.

$$\nabla \mathbf{I}(x,y).\mathbf{v}(x,y) + I_t(x,y) = 0 \qquad (5)$$

Then the estimate of $(\mathbf{M}, \mathbf{b})(x,y)$ is refined using the higher resolution levels, using equation (6). It is established as follows (time variable t is dropped wherever possible):

Let $\mathbf{p}^l$ be a point within a region at a given level l. Let $\delta \mathbf{p}^l$ be the displacement at location $\mathbf{p}^l$. Let $\mathbf{v}_{(\mathbf{M},\mathbf{b})^l}(\mathbf{p}^l) \stackrel{\Delta}{=} \mathbf{M}^l.\mathbf{p}^l + \mathbf{b}^l$ be the affine velocity at location $\mathbf{p}^l$. Since $\delta \mathbf{p}^l = \mathbf{v}_{(\mathbf{M},\mathbf{b})^l}(\mathbf{p}^l).\delta t$, we wish to estimate the displacement $\delta \mathbf{p}^l$.

Let point $\mathbf{p}^{l+1}$ be the father of $\mathbf{p}^l$ at level $l+1$. The displacement $\delta \widehat{\mathbf{p}}^{l+1}$ estimated at the coarser level $l+1$ at location $\mathbf{p}^{l+1}$ is projected on level l and gives an initial estimate $2\delta \widehat{\mathbf{p}}^{l+1}$ of the displacement $\delta \mathbf{p}^l$ at level l. Let $\delta^2 \mathbf{p}^l \stackrel{\Delta}{=} \delta \mathbf{p}^l - 2\delta \widehat{\mathbf{p}}^{l+1}$ be the incremental estimate to be computed at level l. Let $\widehat{(\mathbf{M}, \mathbf{b})}^{l+1}$ be the estimate of $(\mathbf{M}, \mathbf{b})^{l+1}$ obtained at level $l+1$. And let $\Delta \mathbf{M}^l \stackrel{\Delta}{=} \mathbf{M}^l - \widehat{\mathbf{M}}^{l+1}$, and $\Delta \mathbf{b}^l \stackrel{\Delta}{=} \mathbf{b}^l - 2\widehat{\mathbf{b}}^{l+1}$ be the refinement of the motion parameters to be estimated at level l.

A lambertian reflection is assumed, i.e. $I(\mathbf{p}^l + \delta \mathbf{p}^l, t + \delta t) = I(\mathbf{p}^l, t)$. This equation is equivalent to (5). Expanding I to the first order about $\mathbf{p}^l + 2\delta \mathbf{p}^{l+1}$ gives:

$$\begin{array}{l} \nabla \mathbf{I}(\mathbf{p}^l + 2\delta \widehat{\mathbf{p}}^{l+1}, t + \delta t).(\Delta \mathbf{M}^l \mathbf{p}^l + \Delta \mathbf{b}^l)\delta t \\ + I(\mathbf{p}^l + 2\delta \widehat{\mathbf{p}}^{l+1}, t + \delta t) - I(\mathbf{p}^l, t) = 0 \end{array} \qquad (6)$$

As equation (6) is linear with respect to $\Delta \mathbf{M}^l$ and $\Delta \mathbf{b}^l$, least-squares estimates of these quantities can easily be obtained. Then we have: $\widehat{\mathbf{M}}^l = \sum_{k=l}^{L-1} \Delta \widehat{\mathbf{M}}^k + \widehat{\mathbf{M}}^L$ and $\widehat{\mathbf{b}}^l = \sum_{k=l}^{L-1} 2^{k-l} \Delta \widehat{\mathbf{b}}^k + 2^{L-l} \widehat{\mathbf{b}}^L$

Rather than incrementally warping the image at time t towards the image at $t+1$ like [3], an "incremental" version (6) of the image flow constraint equation is used.

Recursive estimation of the motion parameters The multi-resolution method gives instantaneous measurements of the motion parameters. These are then filtered to generate more accurate and stable estimates. Moreover temporal filtering is useful to propagate the estimation of the motion

parameters even when no measurements are available (in case of total occlusion) [11].

In the absence of a known model, the temporal evolution of the motion parameters $(\mathbf{M}, \mathbf{b})(t)$ can be described, in first approximation, by a constant velocity model. In practice, this model is more robust than the (second-order) constant acceleration model. Instantaneous measurements $\tilde{a_i}(t)$ of the motion parameters $a_j(t)$ are given by the multi-resolution algorithm. Since the six components $a_j(t), j = 1, \ldots, 6$ of $(\widehat{\mathbf{M}, \mathbf{b}})(t)$ are usually weakly coupled, six decoupled filters are used. Using this model, a standard Kalman filter generates recursive estimates of each motion parameter. The initialization of the Kalman filter is discussed in [13].

The estimates of the 2D affine velocity model delivered by the filter are then used to compute the 2D affine displacement $(\mathbf{A}, \mathbf{T})$ of the region. Furthermore, a prediction of the region affine displacement in the next frame is calculated. It is then used to initialize in the next frame the position of the snake tracking the region contour. This helps solving tracking ambiguities that the snake may encounter. In conclusion, the two main modules of our approach, contour tracking and motion estimation, truly cooperate, since motion estimation relies on the segmentation provided by tracking, whereas tracking is guided in each new image by the motion prediction.

5 Experimental results

The first sequence shows cars on a highway (fig. 2 and 3). Tracking was initialized on the first image by motion-based region segmentation (see section 2). It provides a map of the moving objects (fig. 2a), from which chains of edge points can be extracted (fig. 3b). One of them describes the car, but very unprecisely. It was approximated by a B-spline and optimized using a fully deformable B-spline curve, so as to better fit image edges (fig. 2c). This curve was then used to initialize tracking. Tracking is good, though the car displacement between two frames is rather large (see fig. 2g). Indeed the use of motion constraints and a prediction (based on region-based motion analysis) increase the robustness of snake-based tracking. The approach is also rather robust to partial occlusion (fig. 3f), the final refinement step being turned off (see section 2). Figure 3h displays the filtered trajectories of the car contour points, which are realistic and coherent.

The second sequence shows a moving human head (see fig. 4 and 5). Here the final refinement step is turned on. Tracking is good, despite a complicated motion - scaling, change of motion direction, temporary occlusion of the ears - and a cluttered background - it disturbs edge detection a bit in the 14th image (fig. 4c) but the error is corrected afterwards. The prediction of the head position in the next frame (see fig. 5d) is rather good and it greatly helps tracking since the head motion is sometimes large and the background is cluttered. Fig. 4g and 5h show the contours tracked during motion. Fig. 5f displays the filtered trajectories of the outline points, which are realistic and regular.

6 Conclusion

The experimental results illustrate the good tracking performances and robustness of the approach. This is due to the combination of snake-based contour tracking and region-based motion analysis. Motion-based segmentation detects the moving objet. Then a snake tracks the region outline and thus performs segmentation. The trajectories of the region border points are also estimated. Motion analysis over the segmented region estimates motion and predicts the region location in the next frame. This prediction guides the snake during tracking and helps to solve ambiguities.

In the future, it is planned to treat total occlusion, and to look at several applications, such as the estimation of time-to-collision, the spatio-temporal surface generated by the tracked contour or 3D structure and/or motion.

7 Acknowledgements

This research was funded by the GDR-PRC "Man-Machine Communication" Computer Vision program.

References

[1] B. Bascle and R. Deriche. Features extraction using parametric snakes. In *Proceedings of 11th IAPR Int. Conf. on Pattern Recognition (ICPR'92), The Hague, The Netherlands*, volume 3, pages 659–662, August 30 - September 3 1992.

[2] B. Bascle and R. Deriche. Energy-based methods for 2d curve tracking, reconstruction and refinement of curves of 3d curves and applications. *Proceedings of Geometric Methods in Computer Vision II, San Diego, July*, 1993.

[3] Bergen, J.R. and Anandan, A. and Hanna, K. and Hingorani, R. Hierarchical model-based motion estimation. *Proc. of ECCV'92, S. Margherita Ligure, Italy, pages 237–252, Springer-Verlag*, 1992.

[4] P. Bouthemy and E. François. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *Intern. J. Comput. Vis.*, 10(2):157–182, 1993.

[5] T.J. Broida and R. Chellappa. Estimation of objects motion parameters from noisy images. *IEEE Trans. PAMI*, Vol.8, No.1:90–99, Jan. 1986.

[6] I. Cohen, N. Ayache, and P. Sulger. Tracking points on deformable objects using curvature information. In *Proc. of 2nd European Conf. on Computer Vision (ECCV'92), Santa Margherita Ligure, Italy*, pages 458–466, May 1992.

[7] R. Curwen and A. Blake. *Active Vision*, chapter Dynamic Contours: Real-time Active Splines, pages 39–57. MIT Press, 92.

[8] R. Deriche and O.D. Faugeras. Tracking line segments. In *First European Conference on Computer Vision, p. 259-268, Antibes, France*, April 1990.

[9] Horn, B.K.P. and Schunck, B.G. Determining optical flow. *Artificial Intelligence, Vol.17:pp 185–203*, 1981.

[10] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *First International Conference on Computer Vision*, pages 259–268, June 1987.

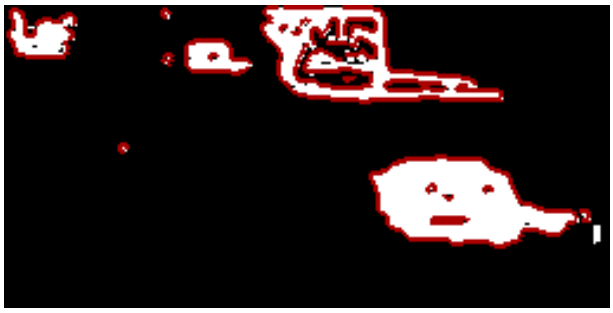
[11] Meditch, J.S. Stochastic optimal linear estimation and control. *McGraw-Hill*, 1969.

[12] A. Menet, P. Saint-Marc, and G. Medioni. Active contour models: Overview, implementation, and applications. In *IEEE Conf. Syst. Man. Cyb. L.A.*, Nov 90.

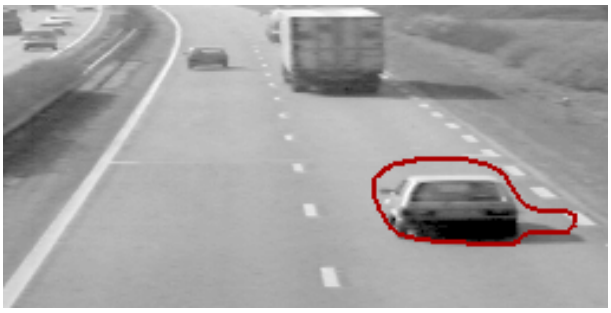
[13] F. Meyer and P. Bouthemy. Region-based tracking in an image sequence. In G. Sandini, editor, *Proc. of 2nd European Conference on Computer Vision (ECCV'92), Santa Margherita Ligure, Italy*, pages 476–484. Springer-Verlag, May 1992.

[14] Schalkoff, R.J. and McVey, E.S. A model and tracking algorithm for a class of video targets. *IEEE Trans. PAMI, Vol.PAMI-4, No.1:2–10*, Jan 1982.

[15] D. Terzopoulos and R. Szeliski. *Active Vision*, chapter Tracking with Kalman Snakes, pages 3–20. MIT Press, 92.

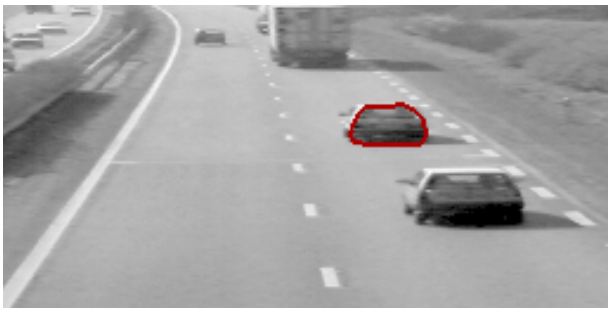


a. Motion-based segmentation and extracted contours for 1st image

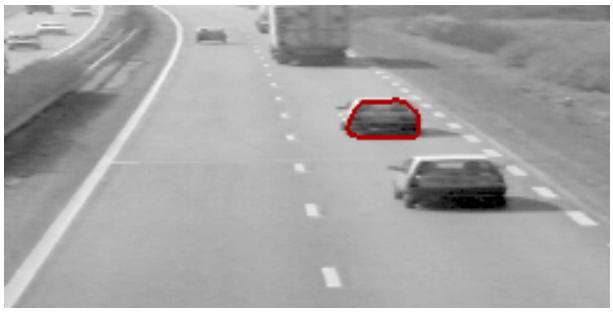


b. Contour of 1st image extracted from motion-based segmentation

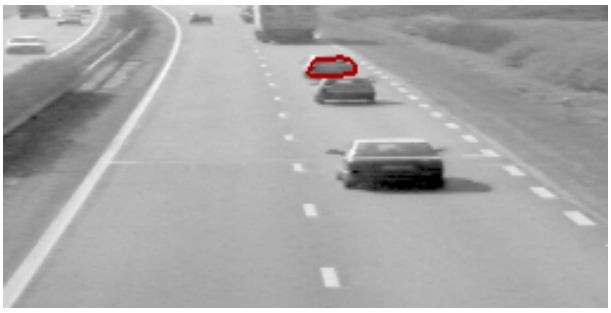
c. Contour of 1st image optimized by a snake starting from the contour extracted from motion-based segmentation



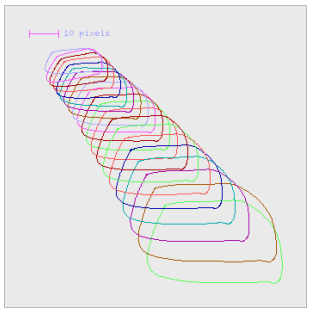
d. Contour predicted on 9th image



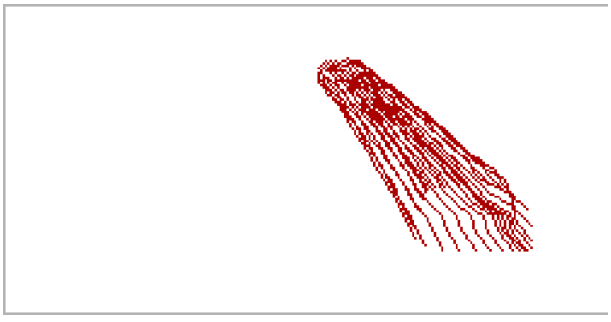
e. Contour tracked on 10th image



f. Contour tracked on 22th image



g. Contours tracked on the image sequence



h. Estimated trajectories of edge points

Figure 2: 1st example - tracking of a car

Figure 3: 1st example - tracking of a car

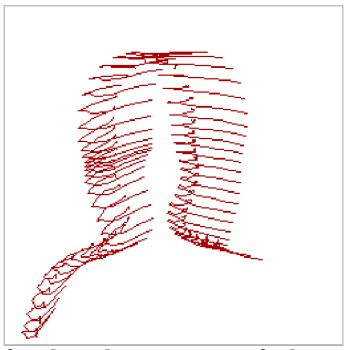
a. Contour extracted on 1st image

b. Contour tracked on 13th image

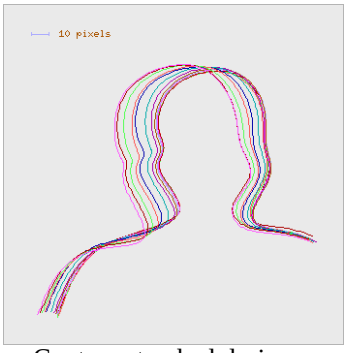
c. Contour tracked on 14th image

d. Prediction on 15th image

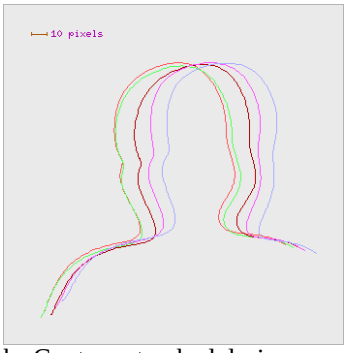
e. Contour tracked on 18th image



f. Filtered trajectories of edge points



g. Contours tracked during leftward motion



h. Contours tracked during rightward motion

Figure 4: 2nd example - tracking of a head

Figure 5: 2nd example - tracking of a head