TRACKING REGIONS USING TEMPORAL INTEGRATION FOR TRAFFIC MONITORING

J. Badenas, J.M. Sanchiz and F. Pla *

Departamento de Informática. Universitat Jaume I de Castellón (SPAIN)
e-mail: badenas@inf.uji.es

Abstract

This paper describes a method for tracking regions in image sequences. Regions segmented from each frame by a motion segmentation technique are matched by using a relaxation procedure. Matching is based on measuring the similarity of the regions from the current frame and a list of regions corresponding to objects. A Kalman filter is used in order to estimate motion parameters. Filter uses a kinematic model which considers varying acceleration. This assumption allows the system to model the approach of objects to the camera.

Keywords: Tracking regions, Kalman filter, matching, traffic monitoring.

1 Introduction

Tracking techniques are based on matching tokens from the image. They are extracted along the sequence and are used as observations for the tracking algorithm. This algorithm basically determines the motion parameters that best fit to the set of observations. We can find in the literature several trends and tracking strategies. Most of these tracking methods can be divided into four groups:

1. Three-dimensional-based methods. They consist of precise geometrical representations of known objects. By using the knowledge about the geometry of the camera and the scene, a three-dimensional model is projected onto the image. This type of methods presents a tremendous computational load that can not be justified by any requirement of a traffic monitoring system. However, they have been applied for tracking individual vehicles by using expensive hardware [7], [8], [10], [17].

2. *Feature-based methods* track individual tokens such as points [19], [3], lines [5], [12] or curves [18], usually based on matching schemes. These methods present two main disadvantages [13], [15]: they do not provide explicit grouping of tokens moving with coherent motion, and are quite sensitive to occlusion.

*This work was partially supported by the projects ESPRIT PROJECT EP-21007 and Generalitat Valenciana GV97-TI-05-26.

3. *Deformable model-based methods* [4], [14], [6] fit models to the contours of the moving objects of the scene. Thereby, models are tracked from one frame to the next. They are very suitable for structured scenes, but exhibit initialization problems [3], [13]. When moving objects are partially occluded in the scene, as usually happens in traffic, initialization fails, since models can not be adapted to the real objects.

4. *Region-based methods* define groups of connected pixels that are detected as belonging to a single object that is moving with a different motion from its neighbouring regions [9],[11], [16]. Region tracking is less sensitive to occlusion due to the extensive information that regions supply. Characteristics such as size, shape or intensity can be directly obtained from them. Moreover, regions are very suitable for scenes with a stationary background, since once motion has been estimated for regions, this information allows to separate moving regions from stationary ones.

Our approach is based on a region tracking algorithm which can take advantage of characteristics found in traffic scenes. In traffic scenes, the background is stationary, and vehicles sometimes appear partially hidden behind other vehicles. Moreover, regions are suitable for tracking objects, since they determine completely the shape and location of objects. Like deformable models, regions present a reasonable computational cost, but without the initialization problem.

The tracking method presented here has been successfully applied to traffic monitoring tasks, where it connects other two computer vision based modules: motion segmentation and temporal integration. The first one analyses each pair of consecutive frames and extracts regions moving with different parameters from their neighbours, providing these motion parameters for each one. The temporal integration module integrates information provided by every segmentation in order to generate new segmentations that are based on the information recovered from the whole sequence. The purpose of this paper is to describe the tracking module. Information about motion segmentation and temporal integration modules can be found in [1] and [2].

The tracking module consists on two main parts: to match regions from different segmentations and to estimate motion parameters. The next two sections describe these techniques, Section 4 shows some results over several sequences and Section 5 draws some conclusions.

2 Matching regions

For every frame of the sequence, our method carries out a motion segmentation that is followed by a matching process. This process matches regions from a list called *master list* to regions segmented in the current frame. Regions from the current segmentation that have not been matched to any region of the *master list* are inserted into the list. The method assumes that these regions can correspond to new objects. When a region is not matched for several consecutive frames, it is removed since it is considered as an object that has gone out of the scene, or come from an erroneous segmentation.

Matching is an iterative procedure which repeats two steps: updating weights and matching.

1. Matching. Each master list region is matched to the most similar region considering five features: x and y region means, intensity mean and motion parameters.

In the case of the master list region, region means and motion parameters are predicted by a Kalman filter (see next section). The five features define a vector $V = (m_x, m_y, m_z, v_x, v_y)$ for each region that is used to look for that region that minimizes a weighted squared Euclidean distance measure:

$$D^{mn} = (V^m - V^n)^t W (V^m - V^n) \tag{1}$$

where V^m is the feature vector of the master list region R_m, V^n is the vector of the new region R_n, and W is a 5x5 diagonal matrix that contains a weight for each feature $(w_x, w_y, w_z, w_{v_x}, w_{v_y})$.

2. Updating weights. For every iteration, the weights are recalculated in order to minimize the total addition of the distances D_n^m, where R_m and R_n are two corresponding regions, and K^m is the number of regions that we are matching.

$$\min_{w_x, w_y, w_z, w_{v_x}, w_{v_y}} D_T = \sum_{m=1}^{K^m} D^{mn} \tag{2}$$

To avoid the uncontrolled increasing of weight values the constraint $w_x w_y w_z w_{v_x} w_{v_y} = 1$ is introduced and used for calculating weights by applying Lagrange Multipliers.

$$D_T = \sum_{m=1}^{K^m} D^{mn} + \lambda(w_x w_y w_z w_{v_x} w_{v_y} - 1) \tag{3}$$

By calculating the value of λ we obtain the values of w_x and w_k $(k = \{y, z, v_x, v_y\})$.

$$w_x = \sqrt[5]{(\textstyle\sum_{m=1}^{K^m}(m_y^m - m_y^n)^2} \sqrt[5]{(\textstyle\sum_{m=1}^{K^m}(m_z^m - m_z^n)^2} \sqrt[5]{(\textstyle\sum_{m=1}^{K^m}(m_{v_x}^m - m_{v_x}^n)^2} \\ \sqrt[5]{(\textstyle\sum_{m=1}^{K^m}(m_{v_y}^m - m_{v_y}^n)^2} \, ((\textstyle\sum_{m=1}^{K^m}(m_x^m - m_x^n)^2)^{-4/5} \tag{4}$$

$$w_k = \frac{w_x \sum_{m=1}^{K^m}(m_k^m - m_k^n)^2}{\sum_{m=1}^{K^m}(m_x^m - m_x^n)^2} \tag{5}$$

These two steps of the matching process are repeated until a minimum is reached for D_T. In order to speed up the process, regions with centroids separated by a distance larger than a threshold e are discarded before calculating D^{mn}.

3 Estimating motion parameters

The goal of this stage is to determine the motion parameters of each region that best fit the observations set. These observations are the motion estimates carried out for each image by the motion segmentation module. In order to speed up the estimation, this module assumes a translational motion model for the vehicle projections. Thus, it provides four measures for each region: centroid coordinates (x, y) and motion parameters (v_x, v_y).

$\vec{v}_k = \{v_{kx}, v_{ky}\}$ are measures of the velocity of regions in the instant k. We should use a motion model that considers the velocity increment that projections of vehicles undergo when they go near to the camera. Thus, the model assumes varying velocity and acceleration. As most of works, we use a recursive estimator (Kalman filter). It estimates the best value, in the least-square sense, of a state vector from a set of Gaussian noisy measures.

The variation of acceleration ($\vec{a}_k = \{a_{kx}, a_{ky}\}$) is assumed constant, and represented by vector $\vec{c} = \{c_x, c_y\}$. Therefore, the kinematic model for the evolution of tokens is defined by the equations (6-10), in which Δt is the time increment:

$$\vec{x}_k = 1/6\vec{c}\Delta t^3 + 1/2\vec{a}\Delta t^2 + \vec{v}_{k-1}\Delta t + \vec{x}_{k-1} \tag{6}$$

$$\vec{v}_k = 1/2\vec{c}\Delta t^2 + \vec{a}_{k-1}\Delta t + \vec{v}_{k-1} \tag{7}$$

$$\vec{a}_k = \vec{a}_{k-1} + \vec{c}_{k-1}\Delta t \tag{8}$$

$$\vec{c}_k = \vec{c}_{k-1} \tag{9}$$

$$\Delta t = 1 \tag{10}$$

Equation (9) defines a planar motion with an acceleration that increases uniformly. According to equations (6) to (9), we can separate this system into two independent subsystems, each one for a different coordinate (x, y). Thus, we construct two Kalman filters with different state vectors ($\vec{X}$ and $\vec{Y}$). These vectors are defined as:

$$\vec{X}_k = \begin{bmatrix} x_k \\ v_{kx} \\ a_{kx} \\ c_{kx} \end{bmatrix} \qquad \vec{Y}_k = \begin{bmatrix} y_k \\ v_{ky} \\ a_{ky} \\ c_{ky} \end{bmatrix} \tag{11}$$

The rest of equations are equal for both Kalman filters, so we only describe the one for coordinate x. The estimation process is controlled by state and measurement equations. The state equation defines the transition from the state $\vec{X}_k$ to $\vec{X}_{k+1}$:

$$\vec{X}_{k+1} = A_k\vec{X}_k + W_k \tag{12}$$

The measurement equation relates the observations to the state of the process:

$$\vec{z}_k = H_k\vec{X}_k + V_k \tag{13}$$

In (12) y (13), variables W_k and V_k represent the noises of the process and measurement. It is assumed that both are independent, with normal probability distributions.

$$P(W) \longrightarrow N(0, Q)$$

$$P(V) \longrightarrow N(0, R)$$

Matrix $(n \times n)$ A_k relates the state at instant k with the state at instant $k+1$. Matrix $(m \times n)$ H_k in (13) relates the state at instant k with the measurement at the same instant. According to (6) to (10), A_k and H_k are defined as:

$$A_k = \begin{bmatrix} 1 & 1 & 0.5 & 1/6 \\ 0 & 1 & 1 & 0.5 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad (14)$$

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (15)$$

Introduction of a gross error in a measure can disturb the estimates produced by the Kalman filter in the following iterations. Therefore, it is important to avoid errors or, at least, to reduce their influence. At each iteration, the Kalman filter provides predictions of the state vectors that are generated previously to the introduction of the measurement. We use predictions to detect gross errors in the position and motion measures. Thus, the measure vectors used for each filter are:

$$\vec{z}_{xk} = \begin{bmatrix} \rho(x_k, \hat{x}_k, \sigma_{x_k}) \\ \rho(v_{kx}, \hat{v}_{kx}, \sigma_{v_{kx}}) \\ 0 \\ 0 \end{bmatrix} \qquad \vec{z}_{yk} = \begin{bmatrix} \rho(y_k, \hat{y}_k, \sigma_{y_k}) \\ \rho(v_{ky}, \hat{v}_{ky}, \sigma_{v_{ky}}) \\ 0 \\ 0 \end{bmatrix}, \qquad (16)$$

where $\hat{x}_k$, $\hat{v}_{kx}$, $\hat{y}_k$ and $\hat{v}_{ky}$ are the predictions of each measure, and σ_{x_k}, $\sigma_{v_{kx}}$, σ_{y_k} and $\sigma_{v_{ky}}$ are the respective typical deviations. Function ρ corrects abrupt changes in the measures and has the following definition:

$$\rho(z_i, \hat{z}_i, \sigma) = \begin{cases} z_i - 2\sigma & If\ z_i < \hat{z}_i - 2\sigma \\ z_i & If\ \hat{z}_i - 2\sigma \leq z_i \geq \hat{z}_i + 2\sigma \\ z_i + 2\sigma & If\ z_i > \hat{z}_i + 2\sigma \end{cases}$$

4 Results

In this section we show some results of the proposed method over the sequence that can be observed in Figure 1. This figure shows a sequence of 13 frames in which eight vehicles are moving onto a road. Frames have been recorded at a rate of ten per second.

Subfigure 1.a contains the first frame of the sequence, and subfigure 1.b is the last one. Subfigure 1.c is the segmentation of the first frame. Regions in this subfigure are tracked along the sequence and the estimated trajectories are shown in subfigure 1.d.

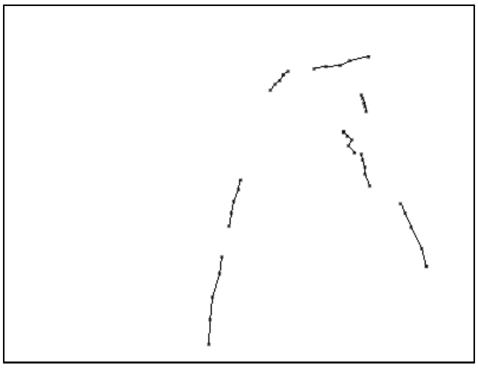
Subfigure 1.d represents the evolution of the centers of the regions. We have presented the estimated positions in five frames of the sequence: frames 1, 4, 7, 10 and 13. The positions have been estimated by using 13 frames with Kalman filter. Lines connect the centers that correspond to the projections of the same vehicle in each frame. Connections have been established keeping the order of the frames, to follow the motion of the regions.

Since the sequence only represents 1.3 seconds, it is expected that the trajectories of the vehicles present a shape very close to a straight line. These results are obtained in almost all the regions, however we can observe small errors in some trajectories. When it happens, the points that do not follow the straight line are always the two first ones (frames 1 and 4). This is due to the fact that the Kalman filter needs some feedback iterations. In these iterations, errors produced in the measuring process (motion estimation and segmentation) influence seriously the estimates provided by the Kalman filter. After

(a) First frame of the sequence

(b) 13th frame of the sequence

(c) Segmentation of the first frame of the sequence



(d) Trajectories of the vehicles

Figure 1: Tracking vehicles in a traffic sequence.

these iterations the influence of the errors begins to be decreased, and estimates are closer to the correct trajectory.

The total system, including motion segmentation and temporal integration, has been tested on a personal computer with four Pentium Pro processors at 200Mhz. We have been able to process 5 frames per second that is enough for city traffic scenes, in which vehicles usually move slower than 70Km/h. At present, computers which double the speed of the one we have used can be easily found in the market. Thus, it is reasonable to say that the frame-rate can be speeded up noticeably. Therefore, it would be possible to cope with scenes where vehicles where moving faster.

5 Conclusions

This paper has shown a method for tracking regions. It is based on establishing correspondences between regions along a sequence. Members of a list are matched to the regions from the current segmentation, allowing the algorithm to follow the behavior of the regions. A Kalman filter is used to estimate the motion parameters of the regions.

The method here presented is part of a traffic monitoring system that works in real-time. The system provides facilities such as tracking vehicle trajectories, measuring traffic density, detection of traffic jams, etc.

References

[1] J. Badenas, M. Bober, and F. Pla. Motion and intensity-based segmentation and its application to traffic monitoring. In *Proceedings, International Conference on Image Analysis and Processing ICIAP'97, Florence, Italy*, pages 502–509. IAPR, 1997.

[2] J. Badenas and F. Pla. Segmentation based on region-tracking in image sequences for traffic monitoring. In *14th International Conference on Pattern Recognition*. ICPR, 1998.

[3] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In *Computer Vision and Pattern Recognition*, pages 495–501, 1997.

[4] A. Blake, R. Curwen, and A. Zisserman. A framework for spatio-temporal control in the tracking of visual contours. *International Journal of Computer Vision*, 11(2):127–145, 1993.

[5] R. Deriche and O. Faugeras. Tracking line segments. *Image and Vision Computing*, 8(4):261–270, 1990.

[6] N.J. Ferrier, S.M. Rowe, and A. Blake. Real-time traffic monitoring. In *Proceedings, 2nd IEEE Workshop on Applications of Computer Vision*, pages 81–87, 1994.

[7] T. Frank, M. Haag, H. Kollnig, and H-H. Nagel. Tracking of occluded vehicles in traffic scenes. In Jan-Olof Eklundh, editor, *7th European Conference on Computer Vision*, pages 485–494, Cambridge, April 1996. Springer-Verlag.

[8] W.F. Gardner and D.T. Lawton. Interactive model-based vehicle tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1115–1121, 1996.

[9] S. Gil, R. Milanese, and T. Pun. Feature selection for object tracking in traffic scenes. In *International Symposium on Smart Highways*, Boston, Massachusetts, 1994. SPIE.

[10] D. Koller, K. Daniilidis, and H.H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10:257 –281, 1993.

[11] D.P. Kottke and Y. Sun. Motion estimation via cluster matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:1128–1132, 1994.

[12] Y. Liu and T.S. Huang. Determining straight line correspondences from intensity images. *Pattern Recognition*, 24(6):489–504, 1991.

[13] A. Mitiche and P. Bouthemy. Computation and analisys of image motion: a synopsis of current problems and methods. *International Journal of Computer Vision*, 19(1):29–55, 1996.

[14] K. Stark and T. Ihle. Visual tracking of solid objects based on an active contour model. In *Proceedings of the British Machine Vision Conference*, pages 640–649. BMVA Press, 1997.

[15] A.M. Tekalp. *Digital video processing*. Prentice Hall, 1995.

[16] L. Wixson. Illumination assessment for vision-based traffic monitoring. In *13th International Conference on Pattern recognition*, pages 56–62, Viena, August 1996. IEEE Computer Society.

[17] A.D. Worral, R.F. Marslin, G.D. Sullivan, and K.D. Baker. Model-based tracking. In *Proceedings of the British Machine Vision Conference*, pages 310–318. BMVA Press, 1991.

[18] Z. Zhang and O.D. Faugeras. Three-dimensional motion computation and object segmentation in a long sequence of stereo frames. *International Journal of Computer Vision*, 7(3):211–241, 1992.

[19] Q. Zheng and R. Chellappa. Automatic feature point extraction and tracking in image sequences for arbitrary camera motion. *International Journal of Computer Vision*, 15:31–76, 1995.