Using Temporal Integration for Tracking Regions in Traffic Monitoring Sequences

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[1] 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. This filter uses a kinematic model which considers varying acceleration. This assumption allows the system to model the movement when objects are approaching to the camera to the camera. The tracking method presented here has been successfully applied to traffic monitoring tasks, where it connects to other two computer vision based modules: motion segmentation and temporal integration.

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 use precise geometrical representations of known objects. This type of methods presents a considerable 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], [9].

2. *Feature-based methods* [5], [3] track individual tokens such as points, lines or curves. These methods present two main disadvantages [11]: they do not provide explicit grouping of tokens moving with coherent motion, and are quite sensitive to occlusion.

3. *Deformable model-based methods* [4], [6] fit models to the contours of the moving objects of the scene. They exhibit initialization problems [3], [11]. 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 [8],[10]. 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.

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. As opposed to usual methods, our method not only uses segmentation to achieve object tracking, but the tracking is

[1]This work was partially supported by the projects TIC98-0677-C02-01, Generalitat Valenciana GV97-TI-05-26 and GV97-TI-05-27.

used for improving the segmentation. This approach stabilizes the shape of the vehicles along the sequence, and corrects segmentation errors. This avoids both appearance and disappearance of regions or parts of regions from one frame to the next one. Moreover, it enables us to store vehicle shapes and maintain it even when a vehicle stops.

The system provides useful facilities for traffic monitoring, such as counting vehicles at each lane, surveillance of trajectories, estimation of the mean velocity of the traffic and detection of stopped vehicles.

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], respectively.

The tracking module consists of 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: weights update and matching.

1. *Matching*. Each master list region is matched to the most similar region in the current frame considering five features: region centroid, intensity mean, and velocity. In the case of the master list region, region means and motion parameters are predicted by a Kalman filter (see next section).These five features define a vector $\mathbf{v} = (m_x, m_y, m_g, v_x, v_y)^t$ for each region, that is used to look for the region that minimizes a weighted squared Euclidean distance measure:

$$d^{mn} = (\mathbf{v^m} - \mathbf{v^n})^t \mathbf{W} (\mathbf{v^m} - \mathbf{v^n}) \tag{1}$$

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

2. *Weights update*. For every iteration, the weights are recalculated in order to minimize the total sum of the distances d^{mn}. To avoid the uncontrolled increasing of weights, the constraint $w_x w_y w_g w_{v_x} w_{v_y} = 1$ is introduced and used for calculating weights by applying Lagrange Multipliers. The function to minimize is:

$$d_T = \sum_{m=1}^{k^m} d^{mn} + \lambda (w_x w_y w_g w_{v_x} w_{v_y} - 1) \tag{2}$$

   where R_m and R_n are two corresponding regions, and k^m is the number of regions that we are matching. After applying Lagrange Multipliers method, the obtained weights expression is:

$$\begin{aligned} w_x = {} & \sqrt[5]{(\textstyle\sum_{m=1}^{k^m} (m_y^m - m_y^n)^2} \sqrt[5]{(\textstyle\sum_{m=1}^{k^m} (m_g^m - m_g^n)^2} \\ & \sqrt[5]{(\textstyle\sum_{m=1}^{k^m} (m_{v_x}^m - m_{v_x}^n)^2} \sqrt[5]{(\textstyle\sum_{m=1}^{k^m} (m_{v_y}^m - m_{v_y}^n)^2} \\ & ((\textstyle\sum_{m=1}^{k^m} (m_x^m - m_x^n)^2)^{-4/5} \end{aligned} \tag{3}$$

$$w_k = \frac{w_x \sum_{m=1}^{K^m} (m_k^m - m_k^n)^2}{\sum_{m=1}^{K^m} (m_x^m - m_x^n)^2}, \quad k = \{y, g, v_x, v_y\} \tag{4}$$

The 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. The observations are the motion estimates computed for each image by the motion segmentation module, more precisely, it provides four measures for each region: centroid coordinates (x, y) and velocity (v_x, v_y). As most works, we use a recursive estimator, the Kalman filter, which estimates the best value, in a least-square sense, of a state vector from a set of noisy measures.

We should use a motion model that considers the velocity increment that the vehicles projections undergo when they approach to the camera. The model assumes varying velocity ($\mathbf{v}_k = \{v_{xk}, v_{yk}\}$) and acceleration ($\mathbf{a}_k = \{a_{xk}, a_{yk}\}$). The variation of acceleration ($\mathbf{c}_k = \{c_x, c_y\}$) is assumed constant. Therefore, the kinematic model for the evolution of tokens is defined by equations (5-8), in which Δt is the time increment, which is assumed as 1:

$$\mathbf{x}_k = 1/6 \mathbf{c}_{k-1} \Delta t^3 + 1/2 \mathbf{a}_{k-1} \Delta t^2 + \mathbf{v}_{k-1} \Delta t + \mathbf{x}_{k-1} \tag{5}$$

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

$$\mathbf{a}_k = \mathbf{a}_{k-1} + \mathbf{c}_{k-1}\Delta t \tag{7}$$

$$\mathbf{c}_k = \mathbf{c_{k-1}} = \mathbf{c} \tag{8}$$

Equation (8) defines a planar motion with an acceleration that increases uniformly. According to equations (5) to (8), we can separate this system into two independent subsystems, each one for a different coordinate (x, y). Thus, we define two Kalman filters with different state vectors ($\mathbf{x}$ and $\mathbf{y}$):

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ v_{xk} \\ a_{xk} \\ c_{xk} \end{bmatrix} \qquad \mathbf{y}_k = \begin{bmatrix} y_k \\ v_{yk} \\ a_{yk} \\ c_{yk} \end{bmatrix} \tag{9}$$

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 the state and measurement equations. The state equation defines the transition from the state $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$:

$$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{w}_k \tag{10}$$

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

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{11}$$

In (10) y (11), variables $\mathbf{w}_k$ and $\mathbf{v}_k$ represent the process and measurement noises. It is assumed that both are uncorrelated, with normal probability distributions.

$$P(\mathbf{w}) \longrightarrow N(0, \mathbf{Q})$$
$$P(\mathbf{v}) \longrightarrow N(0, \mathbf{R})$$

Each Kalman filter uses a different measurement vector: $\mathbf{z}_{xk}$ and $\mathbf{z}_{yk}$.

$$\mathbf{z}_{xk} = \begin{bmatrix} x_k \\ v_{xk} \end{bmatrix} \qquad \mathbf{z}_{yk} = \begin{bmatrix} y_k \\ v_{yk} \end{bmatrix}, \tag{12}$$

Matrix $\mathbf{A}_k$ $(n \times n)$ relates the state at instant k with the state at instant $k+1$. Matrix $\mathbf{H}_k$ $(m \times n)$ in (11) relates the state at instant k with the measurement at the same instant. According to (5-8), $\mathbf{A}_k$ and $\mathbf{H}_k$ are defined as:

$$\mathbf{A}_k = \begin{bmatrix} 1 & 1 & 0.5 & 1/6 \\ 0 & 1 & 1 & 0.5 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{13}$$

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{14}$$

Introduction of a large 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 measurements that are generated previously to the introduction of the real measurement. We use these predictions to detect gross errors in the position and motion measures. Thus, the measures x_k, y_k, v_{xk} and v_{yk} are substituted by x'_k, y'_k, v'_{xk} and v'_{yk} by using function ρ:

$$u' = \rho(u, \hat{u}, \sigma_u) = \begin{cases} u - 2\sigma_u & If\ u < \hat{u} - 2\sigma_u \\ u & If\ \hat{u} - 2\sigma_u \leq u \geq \hat{u} + 2\sigma_u \\ u + 2\sigma_u & If\ z_i > \hat{u} + 2\sigma_u \end{cases}$$
$$u' = \{x'_k, y'_k, v'_{k_x}, v'_{k_y}\}$$
$$x'_k = \rho(x_k, \hat{x}_k, \sigma_{x_k}),\ y'_k = \rho(y_k, \hat{y}_k, \sigma_{y_k}),$$
$$v'_{k_x} = \rho(v_{k_x}, \hat{v}_{k_x}, \sigma_{v_{k_x}}),\ v'_{k_y} = \rho(v_{k_y}, \hat{v}_{k_y}, \sigma_{v_{k_y}})$$

where $\hat{x}_k$, $\hat{v}_{xk}$, $\hat{y}_k$ and $\hat{v}_{yk}$ are the predictions of each measure, and σ_{x_k}, $\sigma_{v_{xk}}$, σ_{y_k} and $\sigma_{v_{yk}}$ are the typical deviations.

4 Results

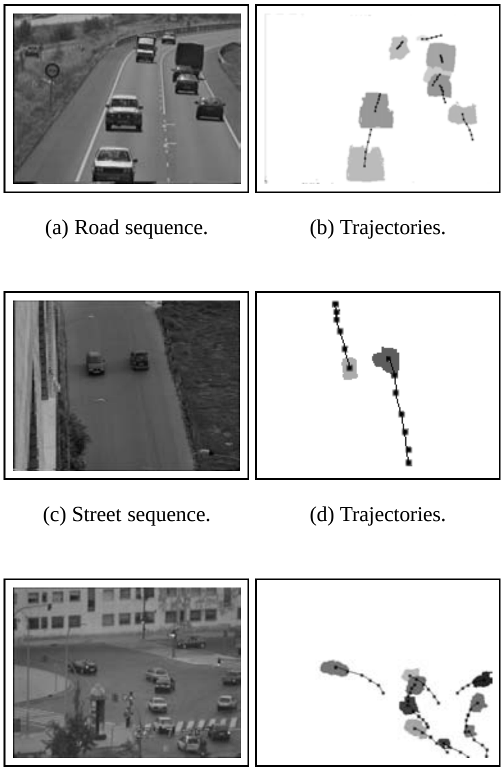
In this section we show some results of the proposed method over the sequences that can be observed in Figure 1. Subfigures 1.a and 1.b show a sequence of 13 frames in which eight vehicles are moving onto a road. Frames have been recorded at a rate of ten per second.

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

Subfigure 1.b 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 the first iterations, errors produced in the measuring process (motion estimation and segmentation) influence seriously the estimates provided by the Kalman filter. After these iterations the influence of the errors begins to be decreased, and estimates are closer to the correct trajectory.

Subfigures 1.c and 1.e present two different urban sequences, and subfigures 1.d and 1.f are the segmentations and the trajectories of the vehicles. These sequences are



(a) Road sequence. (b) Trajectories.

(c) Street sequence. (d) Trajectories.

(e) Roudabout seq. (f) Trajectories.

Figure 1. Tracking in traffic sequences.

longer than the previous one (956 and 120 frames, respectively), thus we have used them to test the traffic monitoring facilities that our system provides. Probably, the most important one is counting vehicles, since it allows the system to calculate measures such as traffic density. The system detected 100% and 95% of vehicles at each sequence.

The total system, including motion segmentation and temporal integration, has been tested on a PC based system 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 were 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. It has been tested on several traffic sequences (road and city sequences). It can process from 5 to 10 frames per second, working in real-time for most traffic sequences.

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*, pages 999–1001. 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] 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.

[9] 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.

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

[11] 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.