7. R. Thoma, M. Bierling, “Motion Compensating Interpolation Considering Covered and Uncovered Background”, *Signal Processing: Image Communication*, Vol. 1, pp. 191-212, 1989.

8. M. Kilger, “A Shadow Handler in a Video-based Real-time Traffic Monitoring System”, *IEEE Workshop on Applications of computer Vision*, Palm Springs, CA, pp. 1060-1066, 1992.

9. P. Bouthémy, E. François, “Motion Segmentation and Qualitative Dynamic Scene Analysis from an Image Sequence”, *International Journal of Computer Vision*, Vol. 10, pp. 157-182, 1993.

10. T. Aach, A. Kaup, R. Mester, “Statistical Model-Based Change Detection in Moving Video”, *Signal Processing*, Vol. 31, pp. 165-180, 1993.

11. S. Gil, T. Pun, “Non-linear Multiresolution Relaxation for Alerting”, *Proceedings of the European Conference on Circuit Theory and Design*, pp. 1639-1644, Davos, Switzerland, 1993.

12. D. Geiger, J.A. Vlontzos, “Matching Elastic contours”, *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 602-604, New York, June 1993.

13. F. Leymarie, D. Levine, “Tracking Deformable Objects in the Plane Using an Active Contour Model”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, pp. 617-634, 1993.

14. F. P. Preparata, M. I. Shamos, *Computational Geometry: an Introduction*, Springer-Verlag, 1985.

15. O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, The MIT Press, London, UK, 1993.

16. J. Shi, C. Tomasi, “Good Features to Track”, *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 593-600, Seattle, USA, 1994.

The experimental results suggests that the method based on the bounding rectangle is quite accurate, since the four tracked corners have the same behavior. This method appears to be robust to measurement errors introduced by some defective frames having different image parity. The position error drops quickly, and remains bounded for the rest of the sequence. The trace of the motion vector covariance matrix decreases rapidly and remains low for all the image sequence. With respect to this method, the results provided by the center of gravity feature are less attractive. Although the initial frames provide an acceptable estimate of the motion parameters, the measurement error does not allow them to improve and converge. The position error and the covariance matrix trace reveal that one target recovers from the error, while the other one diverges. We found that small displacements (target far from the camera) combined with small position errors can produce a loss of stability of the motion parameters. This situation could be avoided either by having a simpler motion model for cars which are far away from the camera, or by measuring the position error in percentage of the vehicle size. The third method, based on the 2D pattern of the targets, seems to perform well in the tracking process: the motion parameters are correctly estimated before the measurement error (8-th frame in the case of the 2D pattern) and then diverge considerably afterwards. Although one of the vehicles is more affected by the measurement error, the system recovers the stability after it disappears, as shown by the decreasing curve of the position error. The trace of the motion covariance matrix indicates different levels of reliability for the two vehicles.

The processing time required for the whole tracking procedure have been reported for the extreme situations in which vehicles enter and exit the field of view of the camera. The different sizes of the vehicles at these two locations have been shown to highly influence the total computing time. However, for the two contour-based features, this dependence is roughly linear, which keeps the total time within a maximum of 133 ms/frame on a Unix workstation. For the region-based feature, however, the dependence is quadratic, which makes the approach unfeasible for vehicles that are at the bottom of the image. In any case, the processing time remains over 2 s. This means that the computations must be optimized through multiscale approaches, possibly using special-purpose, parallel hardware.

From these comparisons, the bounding-rectangle feature appears to be superior, in terms of both quality and efficiency. Indeed, it provides the highest stability, robustness to measurement errors, while requiring the shortest processing time. However, this method may not be optimal if different lighting conditions are considered (for instance, with rain or with dim light). In this cases, the correlation method may be used, since its tracking performance would not be as affected, at the price of a higher computing time.

8. ACKNOWLEDGEMENTS

The authors would like to thank Prof. J. Malik and Dr. D. Koller (Univ. of California, Berkeley) for useful comments and suggestions. The support of Prof. J. Feldman (Intl. Computer Science Institute, Berkeley) is also gratefully acknowledged. S. Startchick (Univ. of Geneva) helped in processing and analyzing experimental results. Some images were kindly provided by I.R.B.S.A. (Switzerland) and Dr. C. Labit (IRISA, Rennes, France).

9. REFERENCES

1. D. Koller, K. Daniilidis, H.H. Nagel, “Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes”, *International Journal of Computer Vision*, Vol. 3, pp. 257-281, 1993.
2. A. Gelb, *Applied Optimal Estimation*, The MIT Press, MA, and London, UK, 1974.
3. K. Baker, G. D. Sullivan, “Performance Assessment of Model-based Tracking”, *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pp. 28-35, Palm Springs, CA, 1992.
4. F. Meyer, P. Bouthémy, “Region-Based Tracking in an Image Sequence”, *Proceedings of the European Conference on Computer Vision*, pp. 476-484, Santa Margarita-Ligure, Italy, 1992.
5. A. Blake, R. Curwen, A. Zisserman, “A Framework for Spatiotemporal Control in the Tracking of Visual Contours”, *International Journal of Computer Vision*, Vol. 11, pp. 127-147, 1993.
6. D. Koller, J. Weber, J. Malik, “Robust Multiple Car Tracking with Occlusion Reasoning”, *Proceedings of the Third European conference on Computer Vision*, Vol. 1, pp. 189-199, Stockholm, Sweden, 1994.

tors. It must be noted that the update and the prediction steps are the same for all kind of features. Their major differences in computing time are thus due to the measurement step. For all contour- and region-based features, the time required to track a vehicle directly depends on its apparent size, that is, on the size of the search window. As a car moves away from the camera and its projection in the image decreases, the time required for its localization in the subsequent frame also decreases. Table 1 reports the time required by the whole tracking process of a vehicle for each frame, for the three features. This time includes disk access for data exchange through files. Given the dependency of the processing time on the apparent size of the vehicle, two values are reported, referring to cars that are at the top and at the bottom of the image (characterized by different values of their average *y* coordinate).

Table 1: Min-max CPU times for the three methods on a Sun Sparc10

Method	Min CPU (y = 12)	Max CPU (y = 78)
Bounding box	58 ms	121 ms
Center of gravity	66 ms	133 ms
Correlation	2202 ms	20726 ms

For the two contour-based features, the measurement step requires the computation of spatio-temporal gradients, the convex hull, both restricted to a search window. Either the bounding rectangle of the convex hull or its center of gravity are then computed. Since the latter steps are much less time-consuming than the previous ones, the total computing times remain comparable. For both features, the dependency of the processing time on the *y* position of the vehicle in the image is roughly linear. In the case of the region-based feature, the measurement step consists of the correlation between a window representing the target and a search window in the subsequent frame, followed by the search of the highest peak in the correlation surface. These operations are clearly quadratic in the size of the window, leading to a much higher dependency of the processing time on the vehicle's position on the image. Since these operations are currently performed in the spatial domain at the highest resolution using a general-purpose workstation, this method requires a total processing time larger than the previous ones by several orders of magnitude. It is clear that region-based features can be considered only using low-resolution images, using coarse-to-fine strategies for correlation and peak localization. In this case, they may be considered as a backup technique for situations that are critical to contour-based methods, such as the presence of high amounts of noise, or very small size vehicles.

7. DISCUSSION AND CONCLUSIONS

This paper describes a system for tracking moving vehicles in traffic scenes. A motion detection algorithm based on a multiresolution relaxation is first used to segment moving objects from the static background. This provides a set of coarse binary mask for each vehicle. A refinement process is then applied on the masks in order to obtain an accurate description of the target's shape and position. The shape of the vehicles is computed through convex polygons approximating their contours. A set of three features is proposed in order to represent the targets. Two of them are contour-based features which depend on the convex polygon approximation of the target: the bounding box and its the center of gravity. The last feature is region-based, consisting of the 2D pattern of the vehicle itself. A tracking system based on two Kalman filters is used in order to estimate the feature positions, as well as their motion parameters. The feature motion is approximated by an affine model which takes into account the translations along the *x* and *y* axes and the scale change of the features. An acceleration term is introduced in the prediction equations of the motion state vector, in order to take into account the changes in the apparent motion of the vehicles due to the wide-angle effect of the camera, as well as the curvature of the road. We found that a model including an acceleration term in the state vector rather than in the prediction equation would be more suitable for this type of camera/road configuration. Another acceleration parameter in the motion state vector could be introduced to account for the road curvature. A comparison has been made for the three types of features, in terms of their tracking performance and the required computing time on a highway real image sequence.

proposed sequence, the cars move away from the camera, so that their actual displacements monotonically decrease, eventually to the subpixel level. For each vehicle it is thus unavoidable that, after a certain amount of time, measurement noise will predominate in the velocity estimates, causing some instability. In order to understand the rapid recovery from the errors introduced in the measurement vector by the parity change, it is appropriate to consider the covariance matrix P_{1k} of the velocity state vectors (see section 5.1). The diagrams on the right of figure 4 show the evolution of the trace of this matrix, whose value is inversely proportional to the "reliability" of the model, i.e. its capability to describe the evolution of the state vectors (see equations 7-9). It can be seen that, for all methods, the trace of P_{1k} continuously decreases in time, meaning an increasing confidence in the system's model.

For the bounding rectangle feature, two points are tracked for each vehicle: the upper left corner of the bounding rectangle and the lower right one. Since two cars are followed at each frame, two pairs of corners are tracked. This yields the four curves represented in the diagram showing the evolution of the position error in figure 4. It can be observed that the motion vector is well predicted for all the frames, since the error (with the exception of the defective frames 6 and 7) generally remains within 6 pixels. At the defective frames, the rapid and strong change of the position error prevents the state vectors from being updated, preserving correct predictions. This is why, after these frames, the position error drops again towards zero and remains bounded. Another interesting observation is that the four corners of the bounding rectangle seem to follow the same coherent motion, even if they are tracked independently. Their velocity, or motion parameters have very similar behaviors. In this case, the negative velocity values are produced by the *y* component of $\hat{\underline{v}}_k(+)$, since the cars move towards the top of the image, corresponding to $y = 0$. The diagram at the upper right part of figure 4 represents the trace of the motion parameters of the four tracked corners. These curves clearly reflect the consequence of the position errors. Indeed, the highest curve corresponds to the upper left corner of the closest (largest) vehicle to the camera. This point is the most directly influenced by the measure errors and thus its reliability remains low, compared to the other ones. Conversely, the trace of the covariance matrix corresponding to the lower right corner of the furthest (smallest) vehicle remains low, which reflects the reliability of the state vectors parameters.

For the method based on the center of gravity, one point per vehicle is considered. Each of the two tracked cars produces one curve for the position error, which are illustrated on the left of figure 4. Let us study the position error for each vehicle separately. The closest (largest) car presents a low position error in the first frames of the sequence. It increases significantly when the measurement error occurs (6th and 7th frames) and does not recover after the error disappears, diverging at the end of the sequence. However, this vehicle can still be tracked thanks to the large search window around the predicted window, which compensates for erroneous predictions. For the farthest (smallest) vehicle, its position error decreases slowly after the initialization step and is less affected by the measurement error. The velocity vectors do not seem to follow a clear asymptotic behavior, as was the case for the previous feature. Their values strongly approach zero in the first frames, and then tend to somewhat oscillate. The trace of its covariance motion matrix shows low values which reflect a reliability increase. For vehicles approaching the top of the image ($y \cong 0$), displacements drop to a sub-pixel level. In such situations, even if the position error is relatively low, repeated updates may lead to important changes in the motion state vector, and thus produce small errors. This situation can produce a loss of stability of the motion parameters, although it does not influence significantly the position error of the feature. To overcome this problem, it would be interesting to represent the position errors in percentage of the vehicle size. Also, a simplified motion model could be proposed in the cases of extremely small displacements.

Finally, the bottom row of figure 4 represents the evolution of the tracking for the correlation-based method. Both curves representing the position errors for the two vehicles slowly decrease with time. The effects of the defective frames are clearly visible. After a few frames, the error is again bounded to low values. These results are comparable to the ones obtained for the bounding rectangle method, shown above. Most of the velocity parameters have a correct behavior. Two of them, produced by the farthest vehicle, diverge after the error has disappeared and seem to fluctuate when displacements become close to a pixel. The traces of the covariance matrices indicate that the measurement error which happened for the defective frame mainly affects one vehicle.

6.2 Time performances

In this section we compare the three different features in terms of the computing time required for tracking a vehicle. This includes the prediction of the position and the motion state vectors, the measurement steps and the update of both state vec-

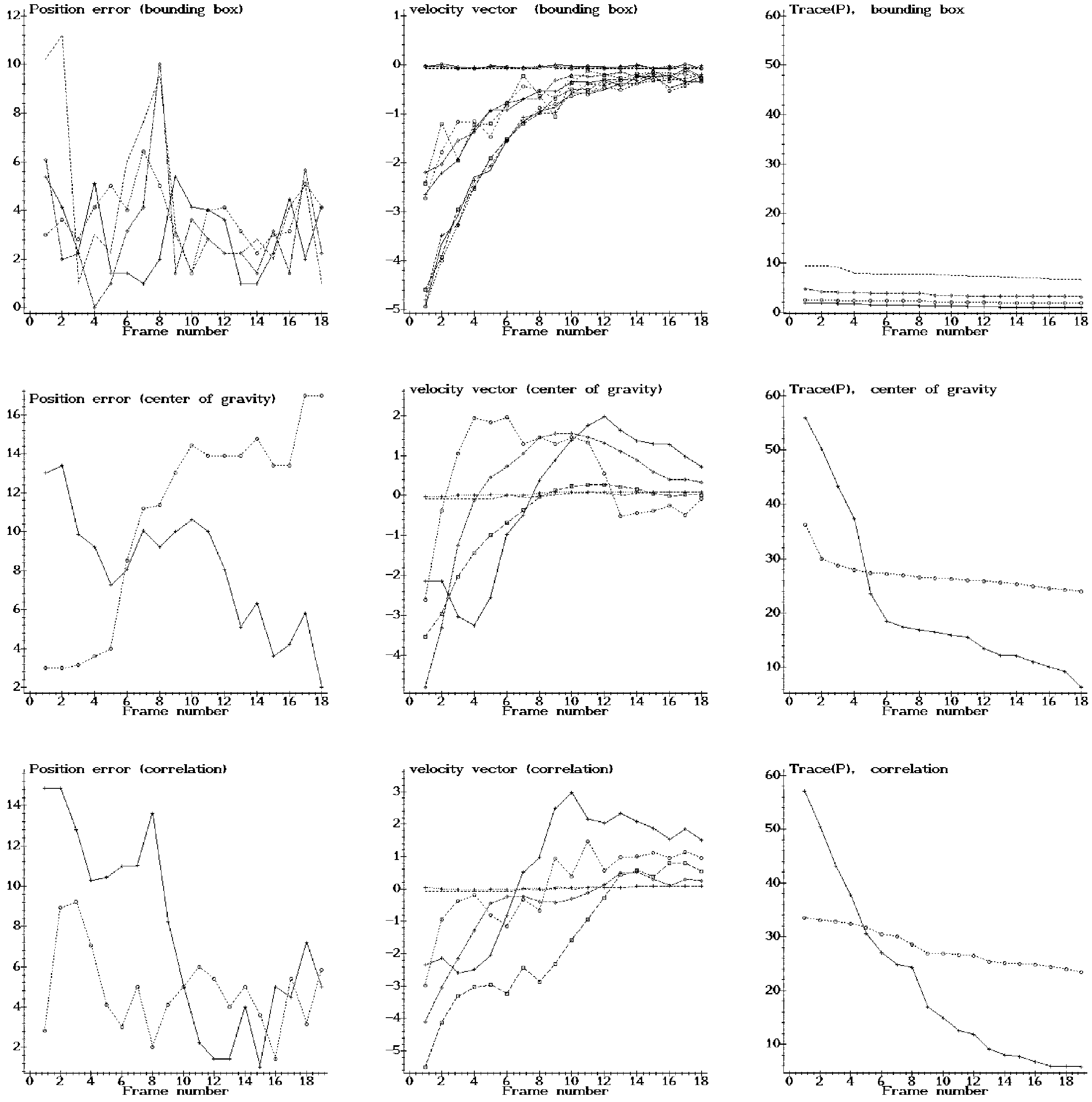


Figure 4: Results of the tracking algorithm for the whole highway sequence. Each row of diagrams shows the performances of different features: bounding rectangle (upper row), center of gravity (middle row), correlation method (lower row). In each row are represented: the evolution of the error measures $w_{1k} = \left\|\underline{z}_k - \hat{\underline{x}}_k\right\|$ (left); the evolution of the motion parameters $\hat{\underline{v}}_k(+)$ (middle); the evolution of the trace of the covariance matrix P_{1k} of the position vectors (right).

The diagrams in the central column of figure 4 show the evolution of the estimated velocity, or motion vector parameters $\hat{\underline{v}}_k(+)$. It can be seen that some of these estimates become unstable for large values of *k*. This is due to the fact that, for the

6. EXPERIMENTAL RESULTS

In this section, experimental results obtained by the tracking system are reported for a highway sequence acquired during daylight. Each frame is 8 bits, 128x256 pixels. The results of all contour- and region-based methods described in the previous section are presented and compared. For all methods, the same initial conditions are applied, using the results of the motion-detection algorithm (see section 5.3).

A common step for all methods is the detection of the convex hull from the spatio-temporal gradients, providing the measurement vector $\underline{z}_k$. The results of this step are reported in figure 3, superposed on the original images. It is interesting to note that the measurements obtained for the lower vehicle in the 1st, 6th and 7th frames tend to be shifted to the right. This is caused by an image parity change due to an error in the image video sampling, which produces an artificial high temporal gradient at the location of the lane mark. This problem would have been present also for other approaches, such as those based on background substraction. Besides this error caused by a manual digitization process, in general the measurements are quite satisfactory, in terms of localization as well as in the shape of the object.

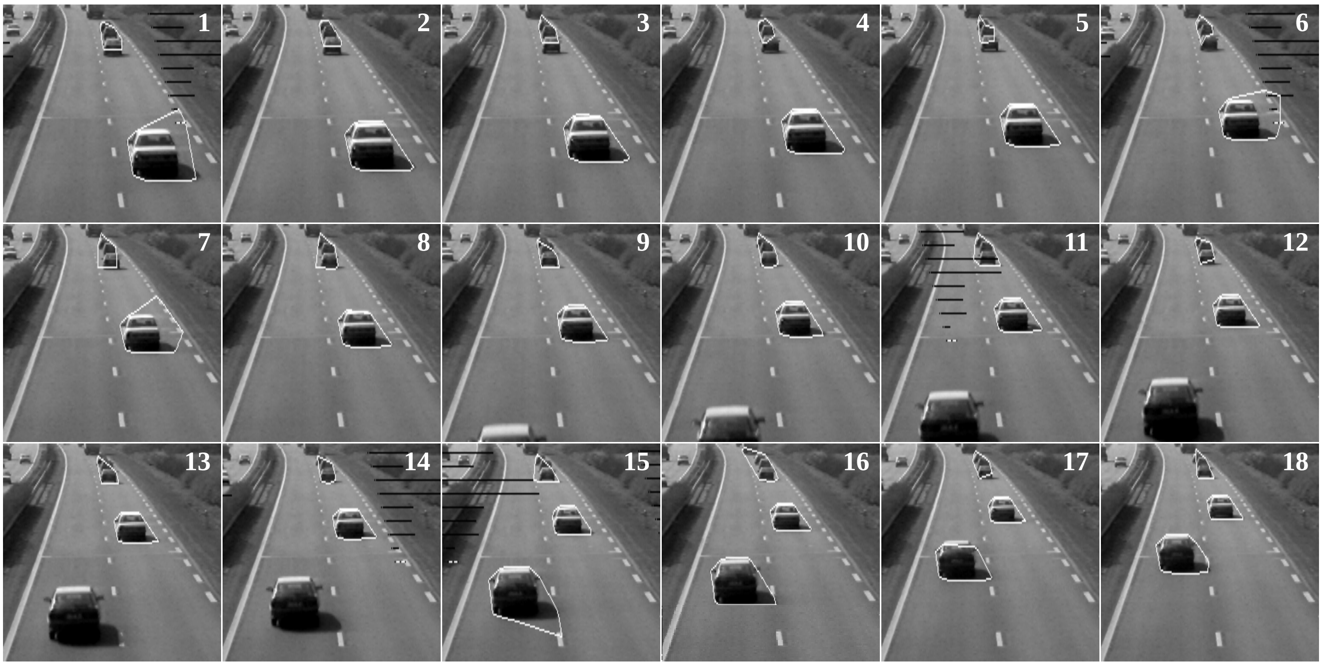


Figure 3: Results of the tracking algorithm for 18 frames of the highway sequence.

6.1 Tracking performances

In this section, the three features used in the tracking process are compared by studying the errors in the position vectors, the stability of their motion parameters as well as the evolution in the trace of the motion parameters covariance matrix. In figure 4, these results are described in a quantitative way for the whole sequence of 18 frames.

The first diagrams (on the left) show the evolution of the norm of the error vector $w_{1k} = \|\underline{z}_k - \hat{\underline{x}}_k\|$. It can be seen that, at the beginning, the absolute value of these errors is relatively high, due to an imperfect initialization step. After a few frames, however, the errors are greatly reduced. A new source of error occurs at the 6th and 7th frames (see figure 3). Here, the digitization problem described above causes incorrect measurements. However, it is interesting to notice that, while the center-of-gravity method is highly affected by this error, the bounding box and the correlation methods quickly recover from this problem, since the following values of w_{1k} converge again towards zero.

$$P_{2k}(+) = \left[Id_3 - K_k \cdot H_{2k}\right] \cdot P_{2k}(-) \tag{16}$$

$$\underline{\hat{v}}_k(+) = \underline{\hat{v}}_k(-) + K_k \cdot \left[\underline{z}_k - \left(\left(H_k \cdot \underline{\hat{v}}_k(-)\right) + \underline{\hat{x}}_k(-)\right)\right] . \tag{17}$$

5.2 Measurement step

The measurement step provides new information on the state vectors, by detecting the position of the N target features in the new frame. As described in section 4, the tracking process has been evaluated on three types of features. The first type of features is contour-based, in that they are obtained from the smallest convex polygon enclosing high spatio-temporal gradients, within a search window (see algorithm introduced in section 3.1 for the mask-refinement step). At each step *k*, the search window is obtained by translating the previous bounding rectangle of the target's convex hull, according to the predicted motion. A tolerance margin is added to the window size for safety. This can compensate for possible errors, for instance those introduced by an incorrect estimation of the predicted motion. Another source of errors degrading the measurement step is the image parity change (resulting from an error in the image video sampling). This error can produce artificial temporal gradients at the location of a high spatial gradient, causing a deterioration in the shape of the convex hull (see section 6 for some examples). Given the search window, a new convex hull is obtained from the thresholded spatio-temporal gradients. From the resulting polygon, only two parameters are extracted, and used for tracking ($N = 2$): the upper left and the lower right corners of its bounding rectangle. This approach leads to a considerable information compression, and avoids the problem of tracking feature vectors of varying size (the number of vertices of the convex polygon is generally not constant through successive frames).

The second contour-based method is based on a similar algorithm, although only the centroid of the convex hull approximation is retained as the feature to track ($N = 1$). This approach represents a further reduction of information, leading to an even lower spatial localization. A priori, it presents the advantage of being more robust for tracking, since it is the result of an averaging process.

The third class of features is region-based, since they correspond to 2D the pattern of the target. The measurement step estimates the target's position in the new image through a normalized correlation technique. The maximum peak in the correlation surface is selected as the position feature vector in the new image ($N = 1$). In order to improve the results of this method, an adaptive approach has been used. At each step *k*, the pattern used in the correlation is not fixed, but selected from the image *k - 1*. This allows to account for the slow changes in illumination and in the target's size (which typically gets smaller for an outgoing car flow). A considerable improvement in system's performance in terms of both computing time and error rate is obtained by selecting a window enclosing the target, rather than using the whole image. To this end, a template containing the target is defined at each step. Correlation operations are thus limited to the smallest rectangular window enclosing this template (surrounded by zeroes). Once the peak $\underline{z}_k = \left(x_k, y_k\right)^T$ has been selected, a new template is computed, to be used as the pattern for the next frame *k+1*. This is done by placing the previous mask at location $\underline{z}_k$, and by computing the convex hull of the resulting points, according to the algorithm introduced in section 3.1.

5.3 Initial conditions

The dynamic equations defining the tracking process require an initialization step. This is obtained by analyzing the binary masks issued by the motion-detection subsystem (see sections 2-3) on two subsequent frames, as soon as the moving object enters the image (typically from the lower end, where figure-ground separation is easier). The correspondence between the masks in the two frames are obtained by simply comparing their spatial coordinates. Given a pair of corresponding masks m', m'', the center of gravity of m'' (the most recent one) is used as the initial value for the position vector $\underline{\hat{x}}_0(+)$ for the region-based method as well as for the second contour-based approach. For the remaining contour-based method, the bounding rectangle of m'' is used to initialize the window enclosing the moving object's convex hull.

For all methods, the displacement between the centers of gravity of m', m'' is employed as the initial velocity vector $\underline{\hat{v}}_0(+)$. The third parameter defining the velocity vector is the scaling factor s_k, defining an magnification/reduction factor of the target between two successive frames. In the initialization step, the value of s_0 is defined according to the ratio between the areas of m', m''.

5.1 Position and velocity filters

The Kalman filters are used to track a set of N visual features of a moving target. For each feature $n = 1, ..., N$ we thus use two state vectors, namely $\underline{x}_k^n$, and $\underline{v}_k^n$, to represent respectively their position and instantaneous velocity. For each of these state vectors, a set of equations similar to equations 4-9 is defined. In the case of the position, the state vector refers to the estimated 2-D coordinates of the N features in the image, that is: $\hat{\underline{x}}_k = (x_k^1, y_k^1, x_k^2, y_k^2, ..., x_k^N, y_k^N)^T$. The measurements $\underline{z}_k$ are the positions of the features, as computed from the k-th image frame. Therefore, there is a straightforward correspondence between the measurements and the position state vector:

$$\underline{z}_k = \hat{\underline{x}}_k(-) + \underline{w}_{1k}, \tag{10}$$

where $\underline{w}_{1k}$ is the measurement error in the position vector. The use of a velocity vector $\hat{\underline{v}}_k^n,\ n = 1, ..., N$ in conjunction to the position vector, allows to define an affine motion model, which takes into account the translations along the x and y axes, as well as the scaling factor s_k representing the shrink of the target as it moves away from the camera: $\hat{\underline{v}}_k^n = \left(u_k^n, v_k^n, s_k\right)^T$. The *observation matrix* H_2 for the velocity state vector for a given feature is defined as:

$$H_2 = \begin{bmatrix} 1 & 0 & \hat{x}_k(-) - x_{ck} \\ 0 & 1 & \hat{y}_k(-) - y_{ck} \end{bmatrix}. \tag{11}$$

The last column of the matrix H_{2k} represents the vector joining the center of gravity of the target $\underline{x}_{ck} = \left(x_{ck}, y_{ck}\right)^T$ to one of the corners of the bounding box containing the target. A change in this vector indicates a change in the scale of the target, and allows the estimation of the scale factor s_k parameter which is responsible for it.

Since the measurements $\underline{z}_k$ are the features position computed from the k-th frame, the position and velocity state vectors are inter-related in the measurement equation:

$$\underline{z}_k = H_2 \cdot \hat{\underline{v}}_k(+) + \hat{\underline{x}}_k(+) + \underline{w}_{2k}, \tag{12}$$

which is another way of writing equation (10) by substituting $\hat{\underline{x}}_k(-)$ by its value given in equation (13).

The prediction equation for the position vector takes now into account the velocity state vector $\hat{\underline{v}}_k = \left(u_k, v_k, s_k\right)^T$, yielding the following expression:

$$\hat{\underline{x}}_{k+1}(-) = \hat{\underline{x}}_k(+) + s_k \cdot \left(\hat{\underline{x}}_k(+) - \underline{x}_{ck}\right) + (u_k, v_k)^T + \underline{w}'_{1k}. \tag{13}$$

The prediction equation for the velocity vector $\hat{\underline{v}}_{k+1}(-)$ incorporates a constant deceleration factor α, which for the image sequence analyzed in this paper is fixed to a value smaller than 1. This coefficient allows to compensate for the change in apparent motion of the vehicles while they cross the camera's field of view:

$$\hat{\underline{v}}_{k+1}(-) = \alpha \cdot \hat{\underline{v}}_k(+) + \underline{w}'_{2k}. \tag{14}$$

This is mainly due to the relative angle between the road and the projective plane, and may also be amplified by the presence of curves in the road, and by the use of wide-angle lenses.

To conclude, the update equations are given below. For the position state vector, they are simply obtained from equations 7-9 by substituting the observation matrix by Id_2. For the velocity vector, they are again obtained from equations 7-9 with some slight modifications:

$$K_k = P_{2k}(-) \cdot H^T{}_{2k} \cdot \left[H_{2k} \cdot P_{2k}(-) H^T{}_{2k} + R_k\right]^{-1} \tag{15}$$

fail. Indeed, if the target size is small, contour-based approaches tend to fail, since the spatio-temporal gradients of the target become low, and the target displacement gets to the sub-pixel level. Contour- and region-based approaches thus appear to be complementary.

In our work, both types of representations will be considered. The contour features are based on the convex polygon approximation of the target's profile. Two features are taken into account: the bounding box of the convex polygon and its center of gravity. Both features are derived from approximating convex polygons using a variable number of vertices, but are represented by a fixed number of parameters (bounding box: two pair of coordinates, center of gravity: one pair of coordinates). The region-based feature is the spatial pattern of the target, which is stored in a rectangular window containing either the target or zeros.

5. THE KALMAN FILTERS

Kalman filters are recursive filters which provide an unbiased, minimum-variance and consistent estimate $\hat{\underline{x}}_k$ of a state vector $\underline{x}_k$. The index *k* represents the discrete time. Kalman filtering consists of a three-steps strategy named prediction, measurement and update. The prediction computes a first estimate of the state vector $\hat{\underline{x}}_{k+1}(-)$ and of the covariance matrix P_k defined as $P_k = \mathrm{E}[\tilde{\underline{x}}_k, \tilde{\underline{x}}_k^T]$, where $\tilde{\underline{x}} = \underline{x}_k - \hat{\underline{x}}_k$ (capital letters are used to denote matrices). According to the dynamic systems notation presented in[2], $\hat{\underline{x}}_k(-)$ denotes the prediction vector before measurement and $\hat{\underline{x}}_k(+)$ refers to the updated vector after the measurement. Prediction equations are based on previous realizations of the updated vector $\hat{\underline{x}}_k(+)$ and the updated matrix $P_k(+)$:

$$\hat{\underline{x}}_{k+1}(-) = f\left(\hat{\underline{x}}_k(+)\right) + \underline{w}'_k, \tag{4}$$

$$P_{k+1}(-) = P_k(+) + Q_k, \tag{5}$$

where Q_k is the covariance matrix of the *model noise* $\underline{w}'_k$: $Q_k = \mathrm{E}[\underline{w}'_k, \underline{w}'^T_k]$. Q_k reflects the adequacy of the model to the studied physical system. The measurement step consists of the computation, through image processing routines, of visual features named the measurements: $\underline{z}_k$. Measurements are related to the state vector through the *observation equation*:

$$\underline{z}_k = H \cdot \hat{\underline{x}}_k + \underline{w}_k, \tag{6}$$

where *H* is the *observation matrix* and $\underline{w}_k$ is a measurement error modeled through an uncorrelated noise. The goal of the observation matrix *H* is to relate the measurement to the state vector. The final update step modifies the state vector according to the measurement $\underline{z}_k$, thus providing an updated estimate $\hat{\underline{x}}_k(+)$. The equations describing the update step modify the state vector and the covariance matrix through the following equations:

$$K_k = P_k(-) \cdot H^T{}_k \cdot \left[H_k \cdot P_k(-) H^T{}_k + R_k\right]^{-1} \tag{7}$$

$$P_k(+) = \left[I - K_k \cdot H_k\right] \cdot P_k(-) \tag{8}$$

$$\hat{\underline{x}}_k(+) = \hat{\underline{x}}_k(-) + K_k \cdot \left[\underline{z}_k - \left(H_k \cdot \hat{\underline{x}}_k(-)\right)\right] . \tag{9}$$

R_k represents the covariance matrix of the *measurement noise* $\underline{w}_k$: $R_k = E[\underline{w}_k, \underline{w}_k^T]$. The matrix K_k described in equation (7) is also called the *Kalman gain* and has the role of modulating the update state vector $\hat{\underline{x}}_k(-)$ into $\hat{\underline{x}}_k(+)$ by appropriately weighting the measurement error $\underline{w}_k$. In order to qualitatively describe equation (9), let us consider a matrix norm such as the trace. If the traces of R_k and P_k are respectively small and large, then K_k becomes large according to equation (7), allowing the update of $\hat{\underline{x}}_k(+)$. This happens when there is a small amount of measurement noise, thus justifying the update. Conversely, if the trace of R_k is large and that of P_k is close to zero, then the elements of K_k converge to zero too, preventing from changing $\hat{\underline{x}}_k(+)$ and locking its value to previous estimations which were not corrupted by noise.

from the top of the pyramid down to its bottom until the full-resolution image is reached. A straightforward method for this projection procedure is the search of the maximum spatio-temporal gradient in a window defined by the 2x2 neighborhood: $(2x + i, 2y + j)$, $i, j \in \{0, 1\}$ where (x,y) is a polygon vertex at the coarse level. However, this method is not satisfactory, since the resulting contour at the lower level tends to slide to neighboring and even static contours of the image, thus deteriorating the quality of the mask shape. In order to avoid this problem, the search is limited to the window obtained by scaling the refined mask of level *l* to the higher resolution *l-1*. The search window thus will contain the spatio-temporal gradient limited to the regions of the refined mask of the higher level. Figure 2 (b) and (c) shows the results of the contour propagation through different resolution levels for different image sequences.

4. FEATURES TO TRACK

Once the target has been accurately isolated from the background, some features must be chosen, in order to be tracked over successive image frames. Several choices are possible for target features, such as the target's color, its contour or a pattern defining its spatial layout. Two tendencies appear to emerge in the existing literature, corresponding to a representation of the target's contour and on its description as a region. Both representations have advantages and drawbacks. Contour-based approaches[15] are fast, since they are based on the (efficient) detection of spatio-temporal gradients. Their major drawback is that the contours of an object in an image not always have a physical meaning. Indeed, contour extraction depends on the local intensity variation between an object and the background, so that changes in their relative intensity may cause a contour to disappear. This type of features is thus reliable only when the contrast between the target and the background is sufficiently constant.

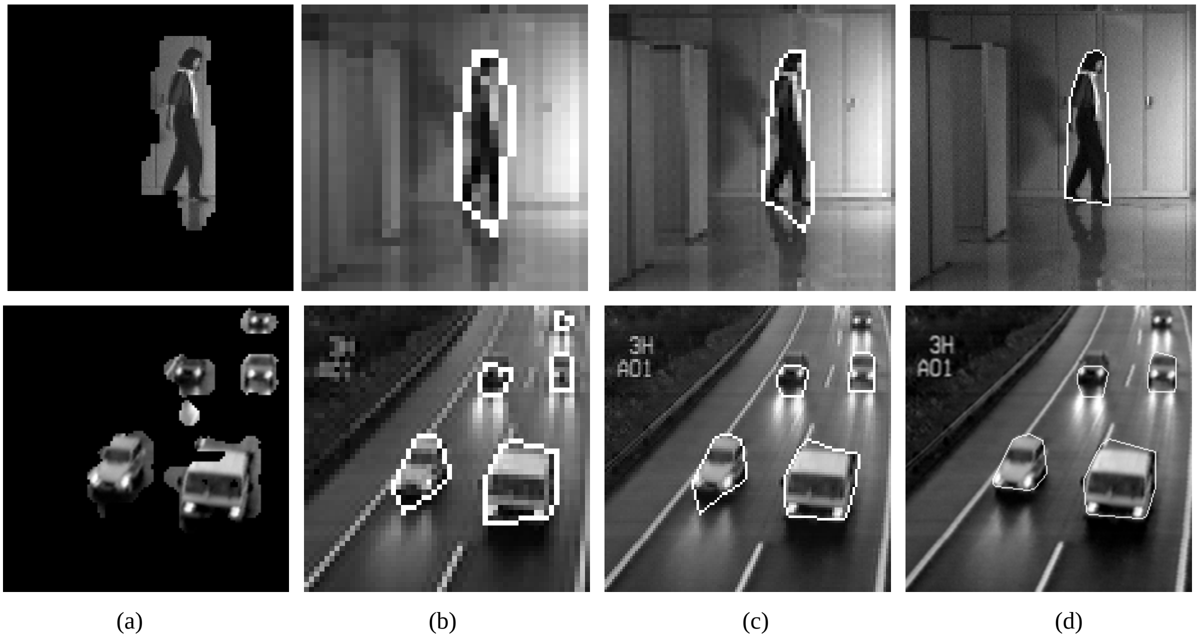
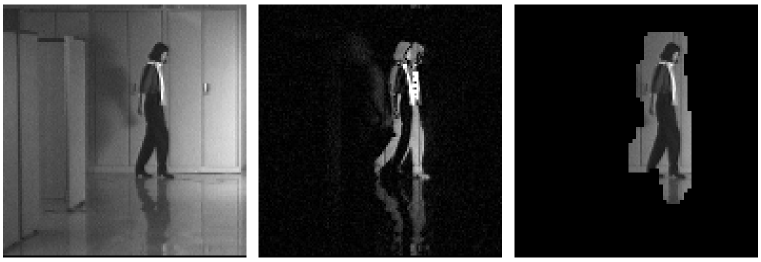


Figure 2: Mask refinement and propagation through the pyramid levels; (a) mask provided by the motion detection algorithm superposed to the one frame of the sequence; propagation of the refined mask (b) to middle (c) and high (d) resolution images.

On the other hand, region-based approaches[16] represent the target through a 2D-pattern; they are quite accurate and do not depend on the background. Their drawbacks are the computing time required for their manipulation (such as pattern matching) and the sensitivity of pattern matching techniques to changes in scale and rotation. Their use is most appropriate when the target size is small, when a low resolution approximation of the target is available or when other representations



(a) (b) (c)

Figure 1: Results of the motion detection module on the “walking” sequence: (a) first frame of the sequence $I_{x,y}(t=0)$; (b) estimate $E^{l=0}_{x,y}$; (c) resulting binary mask superposed to the original image.

3. MASK REFINEMENT AND PROPAGATION

3.1 Mask refinement

The masks provided by the motion detection module represent a coarse description of the moving object, that must be refined. The strategy proposed in this section refines the initial mask boundaries according to the magnitude of spatio-temporal derivatives in the proximity of the initial mask. A similar approach has been adopted in the field of medical imaging for tracking contours of moving organs and cells. Several solutions of different complexity have been proposed. Geiger and Vlontzos [12] present a method to match the inner and outer boundaries of a moving heart wall, by defining a cost function to be minimized. In this context the two contours are known in advance. The cost function thus only takes into account a smoothness constraint on the motion field and a penalty factor for large unmatched arcs of boundaries, while ignoring the image intensity or gradient. Although this approach is not applicable in our context, (the final contour is not known in advance), the regularization terms apply in both situations. Leymarie and Levine [13] describe the tracking mechanism of moving cells by means of active contours (snakes). In their case, an initial snake is matched in the following image using a potential surface which takes into account the image intensity. Low-pass and band-pass pyramids of the potential surface are constructed in order to let the snake evolve while preventing local minima. Snakes are a suitable representation for objects that undergo non-rigid motion, and in the context of vehicles, some simplifications can easily be applied.

In the present work, advantage is taken of the geometric simplicity of vehicles by approximating their profile by a convex polygon in a similar way to [9] and [6]. The convexity assumption is not restrictive in the case of vehicles because in most projections their profiles are pretty compact and are thus well approximated by a convex polygon. This assumption considerably simplifies the matching step required by the tracking procedure, since it allows to by-pass problems such as contour regularization. Furthermore, an extensive literature exists on the topic of convex hull computation[14]. For each resolution level l, the binary mask $M_f^l(x, y)$ contains a number of regions R_1,..., R_N representing moving targets. Due to the properties of the relaxation process, these regions tend to be slightly larger than the underlying objects. For this reason, the refining process uses each region R_i as a search window for the smallest convex polygon P_i containing the set of key points $\{\left(x_1^i, y_1^i\right), ..., \left(x_m^i, y_m^i\right)\}$. Each key point $\left(x_j^i, y_j^i\right)$ within a regions R_i is defined as a point where the spatio-temporal derivatives $D^l\left(x_j^i, y_j^i\right)$ exceed a certain threshold. These operations are actually limited to low resolution images in order to save computation time. Figure 2 shows the coarse initial masks issued from the motion detection algorithm (2.a) versus the contours of the final mask after the refinement process (2.b).

3.2 Mask propagation to higher resolutions

The propagation of the refined mask to higher resolutions is performed by iteratively repeating the projection procedure changing external conditions (e.g. clouds). An example of this method is given in [8] in which a Kalman filter is used to update the background image. This method requires a certain number of frames until a reliable background is available, but its adaptability is a very attractive feature.

Other methods such as [9] exploit motion coherence through MAP techniques in order to separate objects undergoing different motions. This method minimizes, through a deterministic relaxation procedure, an energy function which combines a regularization term and a measure of match between spatio-temporal derivatives and the motion assigned to each region. This method, however, requires the computation of motion parameters and therefore does not meet our requirements described above. MAP techniques have also been used in order to compute global thresholds that segment images into *static* and *dynamic* areas[10]. Global thresholds are first computed according to the noise probability density function (pdf) of the difference images. Since segmentation through global thresholding does not provide well-segmented masks, local refinements are then applied on this preliminary data, based on the MAP criterion. However, this method still leads to oversegmentation, i.e. to many isolated masks which are actually part of the same moving object. Also, a major drawback of MAP techniques is the large processing power they require.

2.2 Motion detection with multiscale relaxation

In contrast to the previous approaches, our method is based on the simple difference of subsequent frames and requires only two frames in order to provide satisfactory results (see[11] for more details). The aperture problem, at the basis of the oversegmentation artifacts, is solved by the use of a multiresolution pyramid $I^l_{x,y}(t)$, for each frame $I_{x,y}(t)$ of the input sequence. At each level *l* of the pyramid ($l = 0, \ldots, \log_2 \text{image\_size}$), first estimates of motions are obtained by computing temporal image differences:

$$D^l_{x,y}(t) = I^l_{x,y}(t) - I^l_{x,y}(t-1) \quad . \tag{1}$$

Local differences $D^l_{x,y}(t)$ provide two motion contributions, through their magnitude, and through the locations of sign changes. These two factors are locally combined together to form the first motion estimates $E^l_{x,y}(t)$ (see Figure 1.b). High-resolution levels of $E^l_{x,y}(t)$ have a better spatial localization, but may only yield information at the object boundaries. Lower-resolution levels help solve the aperture problem, by filling in the interior of moving objects having constant grey level.

Multiple-resolutions motion estimates $E^l_{x,y}(t)$ are combined through a coarse-to-fine pyramidal relaxation process. Its goal is to locally propagate the pixel values *horizontally* within each level, as well as *vertically*, across contiguous levels of the pyramid. The "horizontal" component consists of a diffusion process within each pyramid level, to fill in gaps and reduce noise. The "vertical" component of the relaxation process combines information at location (*x,y*) of level *l* with that at locations $(2x+i, 2y+j)$, $i, j \in \{0, 1\}$ at the higher resolution level *l*-1 of the pyramid. The updating rule of the vertical component is defined by a multiplicative factor $\gamma^l_{x,y} \cdot \Delta^l_{x,y}$, in which $\gamma^l_{x,y}$ is a scaling coefficient.

The increment $\Delta^l_{x,y}$ is defined as a function of the difference image $D^{l+1}(t)$. That is, if the value of $D^{l+1}_{x/2, y/2}(t)$ is smaller than a threshold ξ (proportional the estimated image noise), then $\Delta^l_{x,y}$ is the quadratic term f_1, and otherwise it is given by f_2 :

$$f_1 = -k_1 \cdot (D^{l+1}_{x,y} - \xi)^2 \ , \tag{2}$$

$$f_2 = g \cdot \left(D^{l+1}_{x,y} - k_2 \cdot \xi \right) , \tag{3}$$

where $g(\alpha)$ is a sigmoidal function of the type $1/\left(1+e^{-\alpha}\right)$, and k_1, k_2 are positive constants. This algorithm corresponds to pushing the values of the estimates $E^l_{x,y}$ further towards either 0 or 1.

After the application of this algorithm, the full-resolution image at the bottom of the pyramid contains a binary mask of the moving objects $M^0(x,y)$. Due to the diffusion component of the relaxation process, the shape of these regions tends to be "convex", and to adapt to the shape of the underlying objects. Figure 1 presents the results on the sequence "walking". Despite the shadows and other reflecting surfaces, the resulting masks correspond well to the shape of the moving object.

Surveillance of urban and highway scenes has been widely studied in the past five years, thus providing a large amount of literature. One of the most popular methods, called model-base tracking, uses a 3-D model of a vehicle and is structured in two steps: (i) computation of scale, position and 3-D orientation of the modeled vehicle, also called pose recovery, and (ii) tracking of the vehicle by fitting the model in subsequent frames by means of maximum-a-posteriori (MAP) techniques[1] or Kalman filters[2, 3]. The vehicle model being quite detailed (3-D model including the shadow), model-based tracking provides an accurate estimate (or recovery) of the vehicles 3-D position which might not be needed for most applications. A simplified model of the vehicle is proposed in [4] where it is represented through a polygon, with fixed number of vertices, enclosing the convex hull of some vehicle features. This model dramatically reduces the vehicle model complexity. In [4] Kalman filters are used in order to track the vehicle's position as well as its motion using an affine model which allows for translation and rotation. The fixed number of polygon vertices, however, allows little variations on the objects shape. Some improvements on this point are proposed in [5] through the use of dynamic contours instead of polygons with a fixed number of vertices. Cubic B-splines are fitted on a set of control points (vertices) belonging to the target and so providing a smooth parametric curve approximating its contour. In this case, a Kalman filter is used in order to track the curve in subsequent frames with a search strategy guided by the local contrast of the target in the image, i.e. with no use of the motion information. In the context of traffic scenes, especially in the case of highways, vehicle's motion should be a powerful cue in order to direct the search for the target position in subsequent frames. Another system that combines active contours model with Kalman filtering has been presented in [6]. In this case, the use of separate filters for the vehicle position and other motion parameters (affine model: translation and scale), has been shown to provide better results.

In consideration of this previous work, the approach described in this paper is based in the following points. First, advantage is taken from the simplicity of the targets profile (man-made vehicles), which can be well approximated by simple geometric models such as convex polygons; no restriction on the vertices number should be needed. Motion information in terms of an affine model (translation and scale) is used, as well as local contrast, in order to locate the vehicle in subsequent frames, by means of two separate Kalman filters. Finally, multiple features are tracked in the same image sequence and their performances are compared in terms of robustness, CPU time, and error measures. The rest of this paper is organized in the following way: Section 2 presents a motion detection system which discriminates between static background and dynamic objects and provides a set of binary masks coarsely representing the moving objects. Once moving objects are isolated, their mask shape is refined until their boundary accurately matches their contour (Section 3). After the mask refinement is accomplished, a set of features, such as the mask contour, the pattern describing the target itself, and its center of gravity, are computed for each vehicle, in order to be tracked in subsequent frames (Section 4). In section 5, the tracking procedure is described. Results are presented in Section 6, followed by a discussion. Finally, conclusions are presented in Section 7.

2. THE MOTION DETECTION SYSTEM

The goal of the motion detection module is to perform a segmentation between static and dynamic regions in an image sequence by providing a set of binary masks which coarsely represent the shape and the position of the moving objects. The method is required to be fast since it represents a preprocessing step for motion computation and tracking. For this purpose, it operates on low-level data such as spatio-temporal derivatives or image differences rather than an optical flow information.

2.1 Related work

Motion detection has been studied in different contexts such as video coding, surveillance, or traffic control. Differential methods are based on the substraction of subsequent frames in order to get rid of the constant background and process only the moving regions of the image. An example of this method is described in[7]: after performing the difference between successive frames, a 2-D median filter is applied on the difference image in order to smooth the mask boundaries; finally small regions are eliminated. This strategy is strongly affected by the aperture problem, when moving objects contain large regions of uniform gray-level. In this case, part of these objects are considered static and the resulting masks, despite the median regularization, appear oversegmented. A related approach, called the background method, aims at reconstructing the background using the spatial and temporal derivatives. When an accurate approximation of the background is available, it is subtracted from each frame in order to enhance moving objects. The background image has to be updated to account for

This paper has been accetpted at the SPIE Intl. Symp. on Intelligent Vehicle Highway Systems, Boston, Octobre 31 - Novembre 4 1994

Feature selection for object tracking in traffic scenes

Sylvia Gil[1]
Ruggero Milanese
Thierry Pun

Computer Science Department
University of Geneva, Switzerland
E-mail: *gil@cui.unige.ch*

ABSTRACT

This paper describes a motion-analysis system, applied to the problem of vehicle tracking in real-world highway scenes. The system is structured in two stages. In the first one, a motion-detection algorithm performs a figure/ground segmentation, providing binary masks of the moving objects. In the second stage, vehicles are tracked for the rest of the sequence, by using Kalman filters on two state vectors, which represent each target's position and velocity. A vehicle's motion is represented by an affine model, taking into account translations and scale changes. Three types of features have been used for the vehicle's description state vectors. Two of them are contour-based: the bounding box and the centroid of the convex polygon approximating the vehicles contour. The third one is region-based and consists of the 2-D pattern of the vehicle in the image. For each of these features, the performance of the tracking algorithm has been tested, in terms of the position error, stability of the estimated motion parameters, trace of the motion model's covariance matrix, as well as computing time. A comparison of these results appears in favor of the use of the bounding box features.

Keywords: traffic scenes, motion detection, Kalman filter, tracking, feature comparison.

1. INTRODUCTION

Computer vision techniques can be useful in traffic control in order to increase safety and obtain road state information of monitored areas. For instance, the possibility to extract complex, high-level road information such as congestion, accident or fluid traffic allows to efficiently plan a path through the road network, to quickly bring rescue where needed or to deviate the traffic. In order to extract this type of information it is first necessary to segment moving objects from the scene. In this way, vehicles can be counted, and their trajectory, as well as their velocity and acceleration can be determined. Moreover, statistics can be collected from kinematic parameters in order to make a classification between safe, fluid, congestioned or dangerous state of traffic.

One of the major difficulties of monitoring traffic scenes, along with the real-time requirement, is the variety of light conditions of outdoor scenes. Indeed, the system should be reliable day and night, even though at night only vehicle lights are visible. Weather conditions also bring additional difficulties, such as the presence of the vehicle shadow in sunny days (shadows can prevent from correctly segmenting nearby vehicles) or a change in the contrast between the road and the vehicles when raining (a wet road is darker and generally dries irregularly). Thus, it is necessary to have a system able to adapt to these different lighting conditions by exploiting different visual features according to their reliability under such conditions. This paper presents a comparison of the ability of different features to be recovered and tracked, in an image sequence.

[1] Part of this research was conducted while the first author was a visiting scholar at the International Computer Science Institute, Berkeley, California, thanks to a grant from the Swiss National Fund for Scientific Research (4023-027036).

INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center St. • Suite 600 • Berkeley, California 94704-1198 • Phone (510) 643-9153 • FAX (510) 643-7684

Feature selection for object tracking in traffic scenes

Sylvia Gil
Ruggero Milanese
Thierry Pun[1]

TR-94-060

November 1994

Abstract

This paper describes a motion-analysis system, applied to the problem of vehicle tracking in real-world highway scenes. The system is structured in two stages. In the first one, a motion-detection algorithm performs a figure/ground segmentation, providing binary masks of the moving objects. In the second stage, vehicles are tracked for the rest of the sequence, by using Kalman filters on two state vectors, which represent each target's position and velocity. A vehicle's motion is represented by an affine model, taking into account translations and scale changes. Three types of features have been used for the vehicle's description state vectors. Two of them are contour-based: the bounding box and the centroid of the convex polygon approximating the vehicles contour. The third one is region-based and consists of the 2-D pattern of the vehicle in the image. For each of these features, the performance of the tracking algorithm has been tested, in terms of the position error, stability of the estimated motion parameters, trace of the motion model's covariance matrix, as well as computing time. A comparison of these results appears in favor of the use of the bounding box features.

[1] Thierry Pun is with the Dept. of Computer Science, University of Geneva, Switzerland.