Segmenting traffic scenes from gray level and motion information *

Jorge Badenas[(1)], Miroslaw Bober[(2)] and Filiberto Pla[(1)]

[(1)]Dept. Informática, Universitat Jaume I,

Castellón 12071 (SPAIN)

(badenas,pla)@inf.uji.es

[(2)]Dept. Elec. & Elec. Eng., University of Surrey,

Guildford, Surrey GU2 5XH,(U.K.)

M.Bober@ee.surrey.ac.uk

This paper is concerned with an efficient estimation and segmentation of 2-D motion from image sequences, with the focus on traffic monitoring applications. In order to reduce the computational load to achieve real-time implementation, the proposed approach makes use of simplifying assumptions that the camera is stationary and that the projection of vehicles motion on the image plane can be approximated by translation. We show that satisfactory results can be achieved even under such apparently restrictive assumptions. The use of 2D motion analysis and the pre-segmentation stage significantly reduces the computational load, and the region-based motion estimator gives robustness to noise and changes in the illumination conditions.

Key Words : Traffic monitoring, motion estimation, motion segmentation

*This work was partially supported by the projects ESPRIT PROJECT EP-21007, TIC98-0677-C02-01, Generalitat Valenciana GV97-TI-05-26 and GV97-TI-05-27.

1 Introduction

A traffic monitoring system collects traffic data, analyses it and reports on road state information by providing measures of traffic flow and density and detecting events such as traffic jams or accidents. Computer vision techniques can be useful in this context substituting conventional sensors, like magnetic loops, etc. One of the crucial elements of a traffic monitoring system is the motion analysis component, which segments vehicles from the scene and estimates their motion on the image plane. The task is difficult for several reasons: there are multiple moving objects, the objects of interest are usually small (in the image plane) and poorly textured, illumination conditions may be poor and change rapidly, and multiple occlusions are likely and the environment may be cluttered. But probably the most challenging is the requirement of real-time performance on relatively cheap hardware. These specific difficulties and constraints require that a standard off-the-shelf algorithm cannot usually be applied and special algorithms must be designed.

A great deal of work has been done in the area of motion estimation and segmentation in traffic monitoring. One of the early works [1] describes a system with motion analysis based on simple frame differencing. This simple approach provided poor results, but it showed that frame differencing can provide useful information on road scenes, mainly for classifying areas of the image as static or moving with a relatively low computational cost. This approach has been followed by recent works such as [2], [3] and [4]. Dubuisson and Jain [2] describe a technique that combines motion and colour segmentation. A subtraction approach produces a mask that represents the contour of the vehicle. This mask is subsequently refined by using colour information. Unfortunately, this approach can not cope with occlusions since it does not estimate motion for moving regions. In [3], the subtraction of images is combined with a multiresolution approach. In order to increase robustness to noise, this method uses sign changes in difference images to detect edges with higher accuracy. In [4], subtraction is used to construct a map that represents areas of the image where vehicles can be expected to be observed. This method classifies regions as static or moving by comparing the number of edgels that a region contains in the current frame and in a reference frame.

An important group of methods fits models to moving vehicles. These methods are suitable for very structured scenes, such as roads, but some of them present the model selection

problem. For a correct application of these methods it is necessary to choose a correct model for each vehicle: car, van, truck, etc. This selection can present problems when vehicles are grouped in the scene. On the other hand, they can provide richer information than simpler approaches, although their high computational cost reduces their usefulness in traffic monitoring. We can distinguish two main trends in these methods depending on the dimensionality that they use. The first trend fits three-dimensional models to the vehicles and tracks them along a sequence [5], [6], [7], [8], [9]. These methods classify the vehicle model and supply information about the three-dimensional pose and motion of each vehicle. As traffic monitoring systems do not require so precise information, the second trend of methods dramatically reduces the complexity of the model by using two-dimensional models. The method defined in [10] fits rectangular models to vehicles and can track one vehicle at frame rate. In [11] a polygonal model and Kalman filters are used in order to track the vehicle's position as well as its motion using an affine model. A similar approach is presented by Koller et al [12] which combines active contours with two separate Kalman filters. Remagnino et al [13] combines 2-D and 3-D models with a subtraction approach. Subtraction detects moving regions that are fitted with a three-dimensional or a two dimensional model depending on their size. Small regions are classified as pedestrian (2-D model), as bigger regions are assumed as belonging to vehicles (3-D model).

Beymer et al [14] present a method based on point correspondences that takes into account the rigid motion of the vehicles. Point features that are seen moving rigidly together are grouped into a single vehicle. Gil et al [15] introduce a method for combining contour-based and region-based approaches: one is based on the boundaring-box of the moving object and the other one uses the object's 2-D pattern. This work compares two combining approaches, a linear and a Kalman filter based approach. The results show the superiority of the second one. A different approach is presented in [16] which achieves real time performance by concentrating the method on a concrete task: counting the number of vehicles that cross a street junction. This method does not estimate motion and shape of the vehicles.

In order to overcome some of the drawbacks present in the above mentioned approaches, we present a motion segmentation method for a traffic monitoring system able to deal with occlusions, several objects with different size and real time using low cost hardware. Hence, we present a simple but accurate novel algorithm which combines several techniques

to image segmentation and motion estimation of gray level images. Although most of the employed techniques are well-known, the interest of the proposed method lies on the way in which they are combined to achieve satisfactory results with low computational cost. The proposed method uses a gray level segmentation and frame subtraction in order to distinguish moving from static regions, and carries out robust motion estimation to separate moving adjacent regions that belong to different objects. One of the premises of the developed method has been the use of simple techniques that provide satisfactory results with the aim of achieving a realistic method that can work in real time. Hence, instead of a complex motion model, a translational model has been used and instead of fitting vehicles' models to the image, the proposed method is based on motion analysis of regions.

Therefore, the rest of the paper has the following structure. Next section, presents an outline of the algorithm. Sections 3, 4 and 5 describe in greater detail three stages of processing, namely pre-segmentation, motion analysis and post-processing. Section 6 shows some experimental results on real-word sequences and, finally, conclusions are drawn is section 7.

2 Outline of the Algorithm

The algorithm (Figure 1) consists of three stages that are applied to every pair of consecutive frames: pre-processing involving gray-level based segmentation (static segmentation), motion analysis (including further segmentation if needed) and post-processing stage (motion segmentation), where regions are allowed to merge.

The static segmentation, that it is applied to the first frame of each pair of consecutive frames, is designed to group pixels of similar gray-levels. Since the road surface and the cars are usually poorly textured, the gray-level segmentation is likely to group pixels belonging to objects (cars) or background (road). It does not matter that the image can be segmented into too many regions, because what it is really important is to ensure that the pixels that are included in each region belong to a single object. In the subsequent stages of the

method, motion information will be used to merge regions that belong to the same object.

The first step of the motion analysis stage (second stage) attempts to reduce the number of regions in order to simplify the subsequent operations. This reduction is based on difference images analysis, and it allows merging the majority of the static regions.

Motion estimation is applied to the rest of regions. The motion estimator uses a translational motion model. Although more complicated motion models are probably more appropriate for the road-traffic sequences, the translational model is computationally less expensive and can still cope when the scaling effect is small compared to the translation. The estimation procedure minimises, for each cluster, the sum of Displaced Frame Differences (DFD) transformed by a robust kernel. As a result we obtain unbiased estimates and can segment regions with multiple motions. A multiresolution approach is applied in order to avoid local minima in the minimization process to estimate motion parameters.

The post-processing stage (motion segmentation), uses the spatial neighbourhood relations between regions to improve the final segmentation. If two neighbouring regions have similar motion parameters they are likely to belong to the same moving object. In a merging procedure, the parameters of the neighbouring regions are compared and if similar, the regions are merged. We can also correct erroneous motion parameters at this stage.

To this end we obtain a 2D segmentation map with a large background region and smaller foreground regions (cars). An accurate estimate of the translation is given to each region. The complete system should also include a high-level interpretation module, which is beyond of the scope of the work presented here. This module would be able to track each vehicle in time and calculate traffic parameters, such as mean velocity, number of vehicles, etc.

3 Gray-level based segmentation

The purpose of this step is to group pixels into regions that correspond to a single object. Here, we propose the segmentation method we have developed for our traffic monitoring system, however, it can be used for other applications taking into account that its properties

are:

- The method is based on unsupervised techniques.
- The final number of regions is not fixed, but the algorithm varies it depending on the image characteristics.
- Each resulting region belongs to a single object, although an object can consist of several regions.

Kottle and Sun [17] proposed a clustering method based on the classical k-means algorithm which employs a three-dimensional space of features: the two image coordinates and the pixels intensity. Pixel intensity allows the algorithm to separate pixels of the different objects and the consideration of the image coordinates concentrates the pixels of each region. This method has an important drawback, namely it requires that the number of regions (clusters) is provided as an input parameter. It is very difficult to predict reliably how many regions are needed, because it depends not only on the number of moving objects, but also on their size. This is because the clustering tends to favour a uniform distribution and the average size of the clusters depends on their number. If a small number of regions is used, the clustering may group pixels from different objects, but if a large number is used, it will cause over-segmentation on all the image, even in very smooth areas, and subsequently an unnecessary increase of both the computational load and the complexity of the problem.

Our approach removes these problems by introducing a multistage segmentation where the original image is segmented initially into a relatively small number of clusters, and then each cluster is considered for further segmentation. Since in traffic scenes the road and cars do not exhibit much texture, the decision as to whether a cluster should be further divided, is based on the cluster intensity variance σ_j.

Each static segmentation stage is an application of the k-means algorithm which subdivides an original cluster into other smaller clusters in an iterative process. At each iteration a pixel i of the original cluster is assigned to that new cluster j which minimises the following criterion:

$$E^{ij} = (\overline{p}^i - \overline{m}^j)W^j(\overline{p}^i - \overline{m}^j) \qquad \text{for } j \in \{1, 2, ...k\} \tag{1}$$

where $\overline{p}^i$ is a vector composed of the coordinates and the intensity of the pixel i, $\overline{m}^j$ is the vector which contains the mean coordinates and mean intensity of the cluster j, k is the number of clusters in the image and W^j is a weight matrix that makes the algorithm adapts itself to the image giving more importance to that set of features which characterize better to the properties of a particular cluster.

W^j has the following form:

$$W^j = \begin{bmatrix} w_x^j & 0 & 0 \\ 0 & w_y^j & 0 \\ 0 & 0 & w_z^j \end{bmatrix} \tag{2}$$

and

$$w_x^j w_y^j w_z^j = 1 \tag{3}$$

The weights w_x^j, w_y^j and w_z^j are calculated as:

$$w_x^j = c^j/\sigma_x^j, \;\; w_y^j = c^j/\sigma_y^j, \;\; w_z^j = c^j/\sigma_z^j \tag{4}$$

$$c^j = (\sigma_x^j \sigma_y^j \sigma_z^j)^{1/3} \tag{5}$$

$$(\sigma_x^j)^2 = \left(\frac{1}{N_j - 1}\right) \sum_{x=k^i}^{N_j} (x - m_x^j)^2 \tag{6}$$

where N_j is the number of pixels of cluster j, and σ_x^j, σ_y^j and σ_z^j are the typical deviations of the cluster (coordinate x, coordinate y and gray-level, respectively).

In the absence of noise, σ_z^j can be zero, in such a case the following weights will be assigned: $w_x^j = w_y^j = 0$ and $w_z^j = 1$.

Every cluster j has its matrix W^j. The use of the matrix makes the algorithm to adapt itself to the image, giving more importance to that set of features which characterize better to the properties of a particular cluster. Thanks to this mechanism the iterative process needs fewer iterations to reach a stable state. These matrices are recalculated each t iterations, usually between 10 to 15 iterations.

Since it is very difficult to estimate motion for a very small region reliably, a minimum region size μ is used to prevent over-segmentation. On the other hand, large regions with a small intensity variance are also not desirable, since they may contain a small region of different intensity. Thus, a maximum region size η is also specified. The cluster is divided if the following condition is fulfilled:

$$(N_j \geq \eta \text{ or } (N_j \geq \mu \text{ and } \sigma_j^2 \geq \sigma_t)),$$

where N_j is the number of pixels in cluster j and σ_t is the variance threshold. The selection of the parameter values is somewhat arbitrary, and will depend on the image size, and the minimum size of the object on the image plane that should be detected. The parameters are constant for a given system (e.g. fixed image resolution and camera location).

The proposed technique adapts the final number of clusters to the content of the images. We have to specify the initial number of clusters and the number of divisions per iteration, but the results are not sensitive to the value of these parameters, since they only affect to the number of iterations that are needed by the algorithm. Furthermore, the minimum size of the cluster is restricted, preventing extreme over-segmentation. In our experiments, we initially create 6 to 10 clusters and every cluster that passes the division test is again subdivided into 4 clusters. When the size of a cluster is smaller than two times η, then the cluster is only divided into 2 clusters. The technique does not guarantee that all pixels are spatially-connected and we need to perform connected component analysis afterwards.

4 Motion Estimation

This stage consists of two steps. In a first step we perform a simple rough test on each of the pre-segmented regions to determine if they belong to the stationary background. The test is based on a simple observation that if there is an intensity edge between two regions and at least one of them is moving, then the frame difference for some pixels on the boundary is large (namely for the fractions of edges which are perpendicular to the direction of motion). Therefore, we count pixels that belong to the boundary of two regions and calculate the proportion of them with large frame difference.

At this step, two types of regions are merged: static regions and regions that were divided by the clustering algorithm but in fact form a single region without a substantial discrepancy in the intensity of their pixels.

Difference images are usually calculated by simple subtraction of corresponding pixels in two subsequent frames. This method presents two problems: noisy pixels are considered as moving pixels and some moving pixels are not detected due to the lack of contrast between neighbouring pixels. In order to avoid these problems as much as possible we calculate difference images by selecting groups of difference pixels $\mathcal{G}(\vartheta, \gamma)$. In order to reduce the effect of noisy pixels, the value of a difference pixel is calculated considering its neighbourhood. Thus, a difference pixel is the resulting value of subtracting the means of two groups of pixels (for example, four neighbouring pixels) that are corresponding in two subsequent frames. A difference pixel D_{ij} belongs to a group $\mathcal{G}(\vartheta, \gamma)$ if it satisfy the next condition:

$$\begin{aligned} &D_{ij} \in \mathcal{G}(\vartheta, \gamma) \text{ if } D_{ij} \geq \vartheta \text{ and } \exists D_{kl} \in \mathcal{G}(\vartheta, \gamma) \\ &/D_{kl} \in \{D_{(i+1)j}, D_{i(j+1)}, D_{(i-1)j}, D_{i(j-1)}\} \text{ and } \exists D_{mn} \in \mathcal{G}(\vartheta, \gamma)/D_{mn} \geq \gamma \end{aligned} \tag{7}$$

In order to detect as many moving pixels as possible we do not use only one threshold, but two (ϑ, γ). Difference Pixels with an absolute value larger than the smallest threshold (ϑ) are selected when they form a group of pixels in which there is at least a pixel with difference absolute value larger than the largest threshold (γ). As noisy pixels are isolated pixels with uncorrelated values, difference pixels that do not belong to a group of at least n pixels with values of a same sign (positive or negative) are discarded. For images of 192×144 pixels

$n = 20$ allows us to remove groups of difference pixels that are not produced by motion.

The second step (region-based motion estimation) involves finding the translation parameters that minimises the sum of displaced frame differences (DFD), transformed by a robust kernel ρ. The summation is done over all pixels from the region. In fact, we are minimising an error measure E defined as follows:

$$E_i(dx, dy) = \frac{1}{N_j} \sum_{(x,y) \in C_j} \rho(d(x, y, dx, dy), \alpha, \lambda) \tag{8}$$

$$d(x, y, dx, dy) = I_1(x, y) - I_2(x + dx, y + dy) \tag{9}$$

where C_j is the cluster j, $I_1(x, y)$, $I_2(x, y)$ are the pixel intensity values at location (x, y) in the segmented frame and its consecutive one respectively. $\rho(.)$ is the robust redescending function and α, λ are the function parameters.

Several authors [18] [19] have applied robust estimation to the motion estimation problem. When multiple motions are present within a region, the pixels that are not consistent with the dominant motion may bias the estimate. These pixels are referred to as outliers. Application of the robust kernel ρ reduces the influence of outliers so that they will not affect the value of motion estimate. We have used the following kernel due to its low cost:

$$\rho(x, \alpha, \lambda) = \begin{cases} \lambda |x| & if\ |x| < \frac{\alpha}{\lambda} \\ \alpha & \text{otherwise} \end{cases}$$

The original kernel (Truncated Quadratic) uses a square and not an absolute value of DFD (X). We have transformed this kernel because the absolute value is less expensive computationally.

In this kernel we set the parameter λ to 1. The constant α is the maximum contribution of a pixel to the error measure E. Outliers usually have intensity values very different from the rest of pixels of the region they belong. Thereby, the kernel limits their influence, because, in other case, it would be proportional to the difference between their gray value and the value of their corresponding pixel in the other frame.

Our approach to estimate function (8) is a variant of the steepest descent algorithm but it

requires less computations. At each iteration, the value of the error function $E_i(dx, dy)$ is compared to the values of E_i computed for eight modified displacements:

$$(dx + k * r, dy + l * r), k, l \in \{-1, 0, 1\}\ kl \neq 0,$$

where r is the current resolution. The procedure ends when the value of E_i cannot be further improved by modifications to (dx, dy).

To avoid the local minima of the E_i function and to accelerate the process, we use a multiresolution approach. A coarse value of the motion parameters is calculated from the image sequence at coarse resolution. This value is then used as a starting point for iterations at finer resolution. At the finest resolution we obtain the parameters of the translation to subpixel accuracy (0.1 pixel). Bilinear interpolation is used to approximate intensity values at inter-pixel locations.

5 Final Motion Segmentation

This final stage is based on a region merging approach. It attempts to merge regions with coherent motion. This stage is needed, because the initial gray level based segmentation may split an object into smaller regions. All pairs of adjacent regions are considered as candidates to be merged. Two regions, say A and B, are merged if at least one of the following conditions is satisfied:

$$(E_{AB} \leq Q_1) AND (E_{AB} \leq E_{AA} + Q_2) \tag{10}$$

$$(E_{BA} \leq Q_1) AND (E_{BA} \leq E_{BB} + Q_2) \tag{11}$$

where E_{XY} is the error function for the region X displaced with motion parameters calculated for region Y, and Q_1 and Q_2 are two positive numbers that represent, respectively, the confidence in the motion estimated for a region and the probability of uniting two adjacent regions. Values assigned to Q_1 and Q_2 for a region X are: $Q_1 = \alpha/2$ and $Q_2 = E_{XX} * \upsilon$. Where α is the constant used in the robust kernel and υ is a constant with value 0.18 which has been established experimentally. When error is large Q_1 prevents from merging

regions. On the other hand, when the new error is better or equal than the error due to its estimated motion, Q_2 allows the regions to be merged. This process is repeated for all pairs of adjacent regions until no pair can be merged.

A special problem arises when considering road regions. As we said, the lack of texture makes it possible that the values of the DFD function might be similar with very different pairs of motion parameters when they are applied to road regions. This problem is specially important due to the large size that road regions usually have. Therefore, if we try to merge a moving region with the static region corresponding to the road, the DFD gets worse but the effect of pixels that worsen the DFD is vanished within the large amount of pixels corresponding to the road. To avoid these situations, after motion estimation, regions are divided into static (motion near to zero) and non-static regions. When the DFD's of a non-static region and a static region are compared, the quantity Q_2 becomes $Q_2/2$ to reduce the probability of union. To reduce the probability of union, two adjacent regions of the same class (static or non-static) are more likely to belong to the same object than a pair of regions where one of both was estimated as static and the other one as non-static. This last consideration means that erroneous unions can be avoided, and that the number of comparisons can be reduced.

6 Results

The proposed method has been tested on several traffic scenes and image sequences. In this section we show some experimental results obtained on several of these image sequences. Figure 2 shows the segmentation process for a sequence in which two vehicles are moving away from the camera. Figure 3 presents the segmentation for a sequence in which 10 vehicles are moving in both directions. For each sequence, subfigure a is the reference frame, subfigure b is the result of the gray-level based segmentation, subfigure c is the result of the reduction of regions step, and subfigure d is the final motion segmentation, showing also the displacement vectors for each moving region.

For the first sequence, it can be seen that both vehicles are correctly segmented, whereas

for the second sequence, it can be seen that nine vehicles, and not ten, are detected. There is a car that is moving in the lane on the left that is united to the van that is hiding part of it. Due to the small relative velocity difference that exists between the velocities of both vehicles and their distant position with respect to the camera, this can not be considered as an error of the algorithm. For the rest of vehicles shown in Figure 3 we can see that they are correctly segmented, in spite of the fact that some of them hide parts of other vehicles. Therefore, as we can notice the use of a translational model and subpixel accuracy estimation can cope with small effects of scaling in these type of scenes.

In subfigures 2.c and subfigure 3.c the importance of the reduction of the number of regions step can be seen. This step allows grouping 60-80% of the static pixels of the image into one region. Thus, the subsequent operations are carried out more easily and using less computational time.

The illumination conditions and the size of the vehicles are very similar in all of these sequences. The images were taken with a similar perspective and distance from the camera to the objects. Therefore, the parameters of the algorithm have been the same for both sequences. In the gray-level segmentation stage, the parameters initial number of regions, typical deviation σ, minimum region size μ and maximum region size η were chosen as 6, 12.0, 150 and 2000 respectively, whereas, in the motion estimation stage, the thresholds ϑ and γ and the kernel parameter α were set to 6, 10 and 19 respectively.

Figure 4 shows the segmentation of four frames of the second sequence. In the four frames there is a problem with a small part of the nearest truck. This region is not correctly segmented due to its small decreasing size. The size of this region decreases at every frame due to the motion of the two neighbouring cars. This fact causes that the motion parameters for this region to be incorrectly estimated, although it is very close to the motion estimated for the truck. In any case, this region can be easily classified as a non-vehicle region since it has a very small size when it is compared with the size of the neighbouring vehicle regions.

The next figure (Figure 5) shows the effect that the algorithm produce on the resulting segmentations when we try different combinations of parameters. In general, we can observe the method has a good response when parameters are slightly modified. This means that

it is not necessary to look for the optimal parameters of each processed sequence, but there is a wide range of values that produce suitable segmentations. Although, in some figures we have used intentionally parameter values that are excessively inappropriate, in all the figures the main vehicles have been correctly segmented.

Subfigure 5.a is a segmentation using the following parameters: number of regions = 6, $\sigma = 14$, $\mu = 100$, $\eta = 2500$, $\alpha = 16$. The processed image is the same that appears in Figure 2, but we have introduced small modifications to almost all the parameters. The results in both cases are qualitatively similar. Small differences are observed with respect to the shape of vehicles, but they are due to the changes in the gray-level segmentation parameters.

In order to observe the effect of each parameter, the rest of subfigures of Figure 5 show modifications of a unique parameter with respect to the ones used in subfigure 5.a. The variation of the kernel parameter α in the range 15 - 25 does not produce significant changes in the final segmentation. Thus, in subfigure 5.b the parameter α is incremented to an unreasonable value ($\alpha = 40$) that produces an increase of the outliers influence. As a consequence, the algorithm miscalculates the motion parameters of a few regions, that, finally, are not correctly segmented. Subfigures 5.c and 5.d correspond to variations of the typical deviation used in the gray-level segmentation. This parameter affects gray-level segmentation: the larger this value is, the bigger regions are. We have observed that the variation of this parameter in the range 10 - 18 does not influence significantly the results obtained. Both subfigures (5.c and 5.d) are very similar. The main differences can be observed in subfigure 5.d, in which the typical deviation value, that is excessively large, produces that the gray-level segmentation does not carry out all the necessary subdivisions of clusters.

Subfigures 5.e and 5.f show two segmentations in which the initial number of regions for the gray-level segmentation are varied with respect to the subfigure 5.a. Results are very similar when varying this parameter, however there is an important difference: when the number of regions is increased, the computational cost also augments. In subfigure 5.e the number of regions is twice the one used in subfigure 5.a (6 initial regions), and the increment of computing time was 7.7%. In subfigure 5.f, the number of regions was 35 and

the increment of computing time was 61.5%.

In subfigure 5.g, subpixel motion estimation has been removed. Some adjacent regions that correspond to different objects are merged due to the lack of subpixel information that is needed to distinguish the different motions of these vehicles. In subfigure 5.h, the values of the parameters that control the calculation of difference pixels have been considerably augmented. Consequently, the quantity of difference pixels is widely reduced, with respect to subfigure 5.a, and some adjacent regions are merged.

Figure 6 presents the segmentation of several frames from a sequence of cars on a highway. In this sequence vehicles are moving with velocities close to 100km/h. The parameters used to process these images were the same that in the sequences of Figures 2 and 3, except for the minimum region size that was set to 100. Subfigure 6.b corresponds to the segmentation of a car. This is the first frame in which the car appears in the segmentation since in previous frames its motion in the image was too close to zero. Similarly, subfigure 6.d is the first segmentation in which two different vehicles are detected, when their motions are sufficiently different to zero. Subfigure 6.e is the 55th frame and subfigure 6.f is its corresponding segmentation. Finally, subfigure 6.h is the segmentation of the 100th frame.

Figure 7 corresponds to a city traffic scene where several cars are moving in a roundabout. The car speeds are not very quick (40-70 km/h). Some vehicles turn on the left following the roundabout, and other ones turn on the right, leaving the roundabout. For this sequence, images 1th, 10th, 20th and 30th are shown. Subfigures 7.b, 7.d, 7.f and 7.h contain their corresponding segmentations. This sequence presents the problem of little contrast since it was recorded at late afternoon. Thereby, we have used the same parameters that in the previous sequences, except for the typical deviation that was set to 9.

All the parts of the algorithm have linear computational cost. The costs of clustering, motion estimation and motion segmentation stages are, respectively, $O(Nk^2)$, $O(Nt^2)$ and $O(Nl)$ where N is the number of pixels of the image. The constants k, t and l are small compared to N. After a long experimentation their values are estimated as varying in the following ranges: $1 \leq k \leq 4$, $1 \leq t \leq 8$ and $0 < l \leq 49$.

The algorithm has been tested on a personal computer with two Pentium II processors at

400Mhz, using sequences of images with 192×144 pixels and using an 'Activity Map' [20] that marks the pixels that are always static. These pixels are discarded since they will be never occupied by a vehicle. Thus the computational cost have been calculated for several sequences producing times that range from 0.11 to 0.18 seconds per frame, depending on the number and size of vehicles present on the image. Particularly, for the sequence showed in Figure 2 the mean cost per frame was 0.12 seconds. This allows the system to segment, at least, eight frames per second which would permit a tracker based on the present algorithm to monitor vehicles in real time.

7 Conclusions

We have presented an approach for motion analysis in traffic scenes, which segments moving vehicles and estimates their velocities in the image plane. Our approach is a novel combination of several techniques and algorithms which unites gray-level and motion segmentation with enough robustness and accuracy to be applied to traffic monitoring tasks and performed in real time.

The initial gray-level segmentation algorithm is an improved variant of the k-means method that adapts itself to different illumination conditions and groups pixels that uniquely belong to the same object. That segmentation method can be substituted by another one that accomplishes the same conditions (see Section 3): unsupervised, non-fixed number of regions and one object per region (although an object can consist of several regions). The technique based on difference images reduces in a way that is both fast and simple, the number of regions by merging almost all the static regions of the image into a stationary background.

The motion estimation process is based on finding the motion that minimizes the gray level difference between the motion-compensated pixels in the original frame and corresponding pixels in the consecutive frame. The estimation is performed with sub-pixel accuracy and uses a multi-resolution approach that means that the local minima of the Displaced Frame Difference function can be avoided and speeds up the computational process. In order to achieve a reliable motion estimation, we use a redescending robust kernel that allows the

system to reject outlier pixels which often appear in outdoor scenes. Sub-pixel accuracy allows the method to separate vehicles that appear overlapped on the image plane even if their relative motions are very similar.

We have demonstrated that a motion analysis based on a translational model is a valid technique for segmentation purposes when the scaling effect of the motion is small compared with translation, and therefore it is not necessary to employ a more complex model such as affine or perspective motion model.

The proposed method copes well with multiple moving objects. Unlike some other techniques it does not use feature points and it consequently works satisfactorily in a cluttered environment under different illumination conditions. Moreover the method can work in real time using standard low cost hardware according with the requirements for a traffic monitoring system.

Future work is directed at developing a tracker that takes the segmentation and motion estimates provided for the current algorithm and carries out surveillance of vehicles. This tracker should perform traffic monitoring tasks as vehicle counting, detection of traffic jams and accidents, calculation of medium velocity of traffic, etc.

References

1. Waterfall R, Dickinson K. Image procesing applied to traffic. Traffic Engineering Control 1984. 25:60–67.

2. Dubuisson M, Jain A. Contour Extraction of Moving Objects in Complex Outdoor Scenes. International Journal of Computer Vision 1995. 14:83–105.

3. Gil S, Pun T. Non-linear multiresolution relaxation for alerting. In Proceedings of the European Conference on Circuit Theory and Design. 1993 pp. 1639–1644.

4. Teal M, Ellis T. Spatial-Temporal Reasoning on Object Motion. In Proceedings of the British Machine Vision Conference. BMVA Press, 1996 pp. 465–474.

5. Gardner W, Lawton D. Interactive Model-Based Vehicle Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 1996. 11:1115–1121.

6. Koller D, Daniilidis K, Nagel H. Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes. International Journal of Computer Vision 1993. 10:257 –281.

7. Haag M, Nagel HH. Tracking of complex driving manoeuvres in traffic image sequences. Image and Vision Computing 1998. 16:517–527.

8. Haag M, Nagel HH. Combination of Edge Element and Optical Flow Estimates for 3D-Model-Based Vehicle Tracking in Traffic Image Sequences. International Journal of Computer Vision 1999. 35(3):295–319.

9. Tan T, Sullivan G, Baker K. Model-Based Localisation and Recognition of Road Vehicles. International Journal of Computer Vision 1998. 27(1):5–25.

10. Ferrier N, Rowe S, Blake A. Real-Time Traffic Monitoring. In Proceedings, 2nd IEEE Workshop on Applications of Computer Vision. 1994 pp. 81–87.

11. Meyer F, Bouthemy P. Region-Based tracking in an Image Sequence. In G Sandini, editor, Proceedings, Second European Conference on Computer Vision, (Santa Margherita Ligure). Springer-Verlag, 1992 pp. 476–484.

12. Koller D, Weber J, Malik J. Robust Multiple Car Tracking with Occlusion Reasoning. In JO Eklundh, editor, Proceedings, 5th European Conference on Computer Vision, Berlin. Springer-Verlag, 1994 pp. 189–196.

13. Remagnino P, Baumberg A, Grove T, et al. An integrated Traffic and Pedestrian Model-Based Vision System. In Proceedings of the British Machine Vision Conference. BMVA Press, 1997 pp. 380–389.

14. Beymer D, McLauchlan P, Coifman B, et al. A Real-Time Computer Vision System for Measuring Traffic Parameters. In Computer Vision and Pattern Recognition. 1997 pp. 495–501.

15. Gil S, Milanese R, Pun T. Combining Multiple Estimates for Vehicle Tracking. In JO Eklundh, editor, 7th European Conference on Computer Vision. Springer-Verlag, Cambridge, 1996 pp. 307–320.

16. Fathy M, Siyal M. Measuring traffic movements at junctions using image procesing techniques. Pattern Recognition Letters 1997. 8:493–500.

17. Kottke D, Sun Y. Motion Estimation Via Cluster Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 1994. 16:1128–1132.

18. Bober M, Kittler J. Estimation of general multimodal motion: an approach based on robust statistics and Hough transform. Image and Vision Computing 1994. 12(12):661–668.

19. Black M, Anandan P. The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields. Computer Vision and Image Understanding 1996. pp. 75–104.

20. Steward B, Reading I, Thomson M, et al. Adaptative Lane Finding In Road Traffic Image Analysis. In Proceedings of the 7th International Conference on Real Traffic. 1994 pp. 133–136.

Figure 1. Flow Chart of the implemented algorithm.

Figure 2. Segmentation process of a sequence with two vehicles. (a) Reference frame, (b) Initial segmentation, (c) Merging regions by using difference images, (d) Segmented regions.

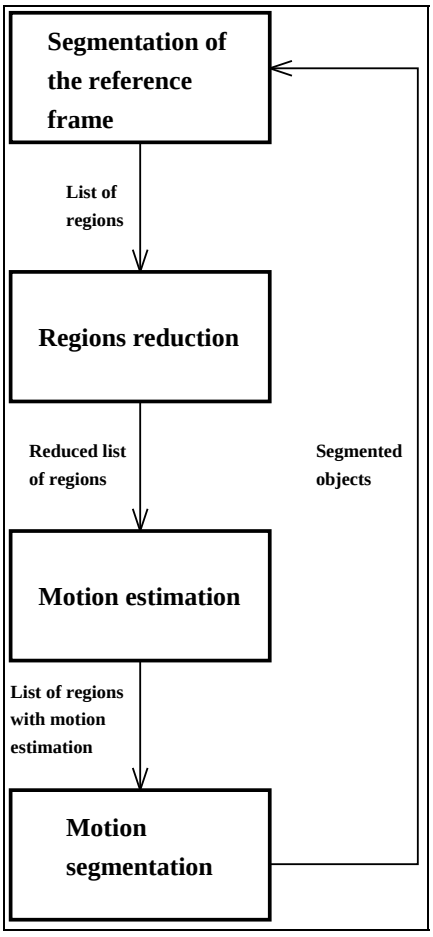
Figure 3. Segmentation process of a sequence with ten vehicles. (a) Reference frame, (b) Initial segmentation, (c) Merging regions by using difference images, (d) Segmented regions.

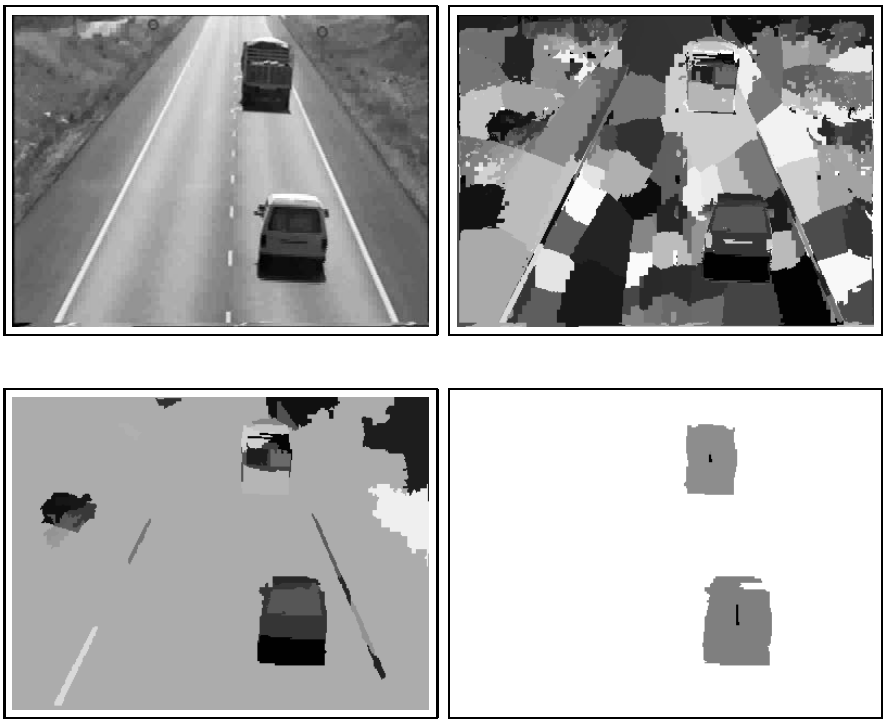
Figure 4. Segmentation of several frames of the sequence in Figure 3. (a) First frame segmentation, (b) Forth frame segmentation, (c) Seventh frame segmentation, (d) Tenth frame segmentation.

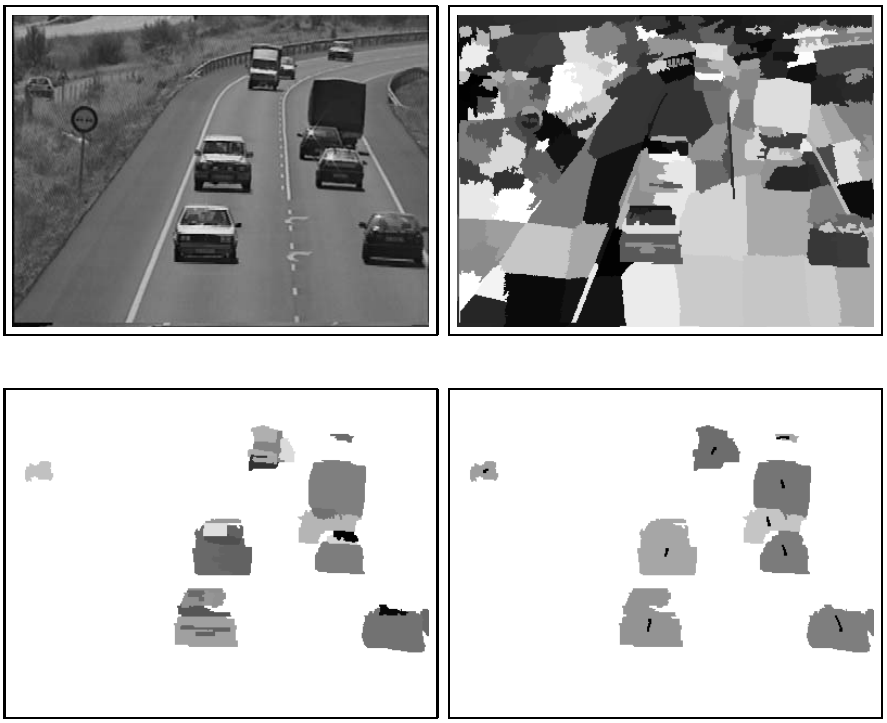
Figure 5. Segmentation results with different combinations of parameters. (a) Number of regions= 6, $\sigma = 14$, $\mu = 100$, $\eta = 2500$, $\alpha = 16$. (b) Variation of the robust kernel parameter $\alpha = 40$. (c) Variation of the typical deviation $\sigma = 9$. (d) $\sigma = 20$. (e) Variation of the initial number of regions for the gray-level segmentation. Number of regions = 12. (f) Number of regions = 35. (g) Execution of the algorithm without subpixel estimation. (h) Variation of the parameters ϑ and γ for the calculation of difference pixels. $\vartheta = 10$ and $\gamma = 18$.

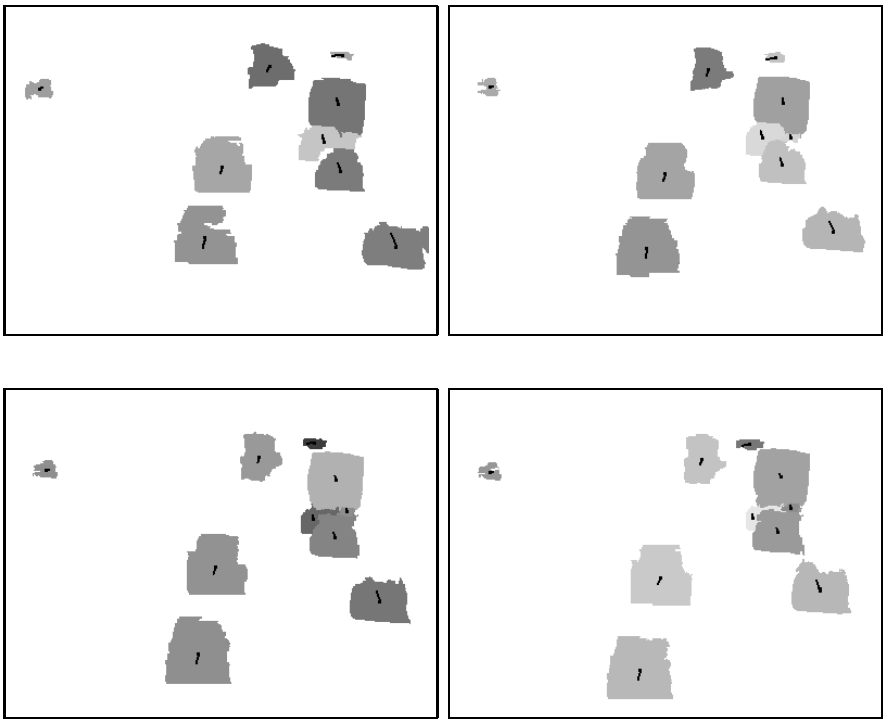
Figure 6. Segmentation of several frames of a highway sequence. (a) First frame, (b) Segmentation of the first frame, (c) 46th frame, (d) Segmentation of the 46th frame, (e) 55th frame, (f) Segmentation of the 55th frame, (g) 100th frame, (h) Segmentation of the 100th frame.

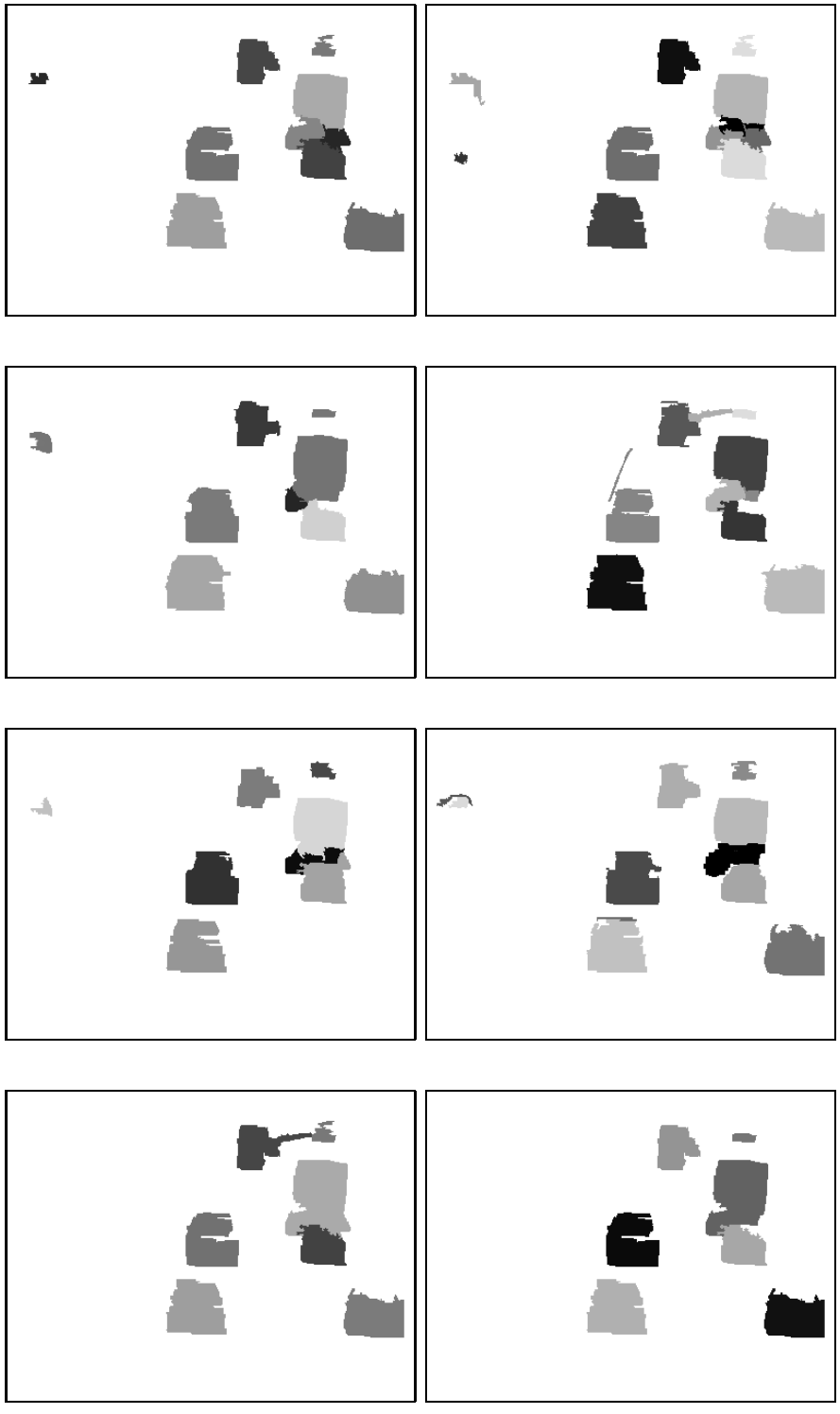
Figure 7. Roundabout scene. Segmentation of several frames of a street sequence. (a) First frame, (b) Segmentation of the first frame, (c) 10th frame, (d) Segmentation of the 10th frame, (e) 20th frame, (f) Segmentation of the 20th frame, (g) 30th frame, (h) Segmentation of the 30th frame.

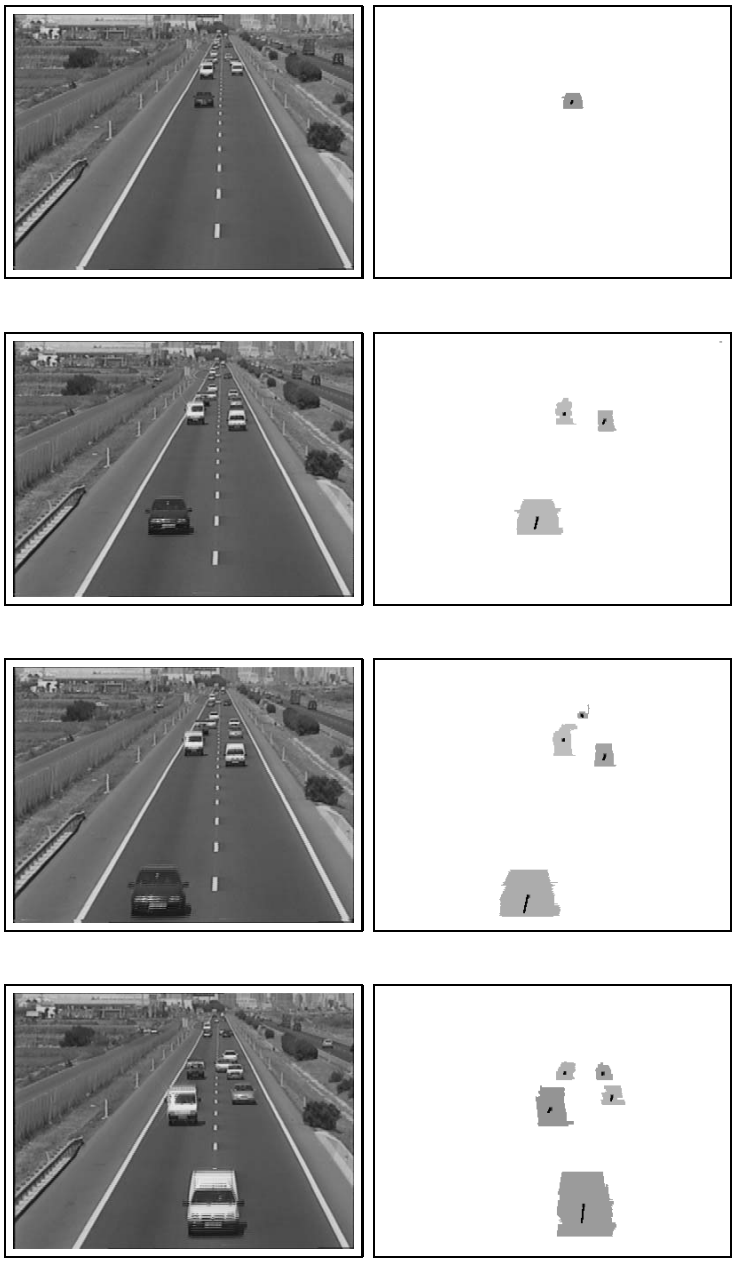

Segmentation of the reference frame
List of regions
Regions reduction
Reduced list of regions
Segmented objects
Motion estimation
List of regions with motion estimation
Motion segmentation

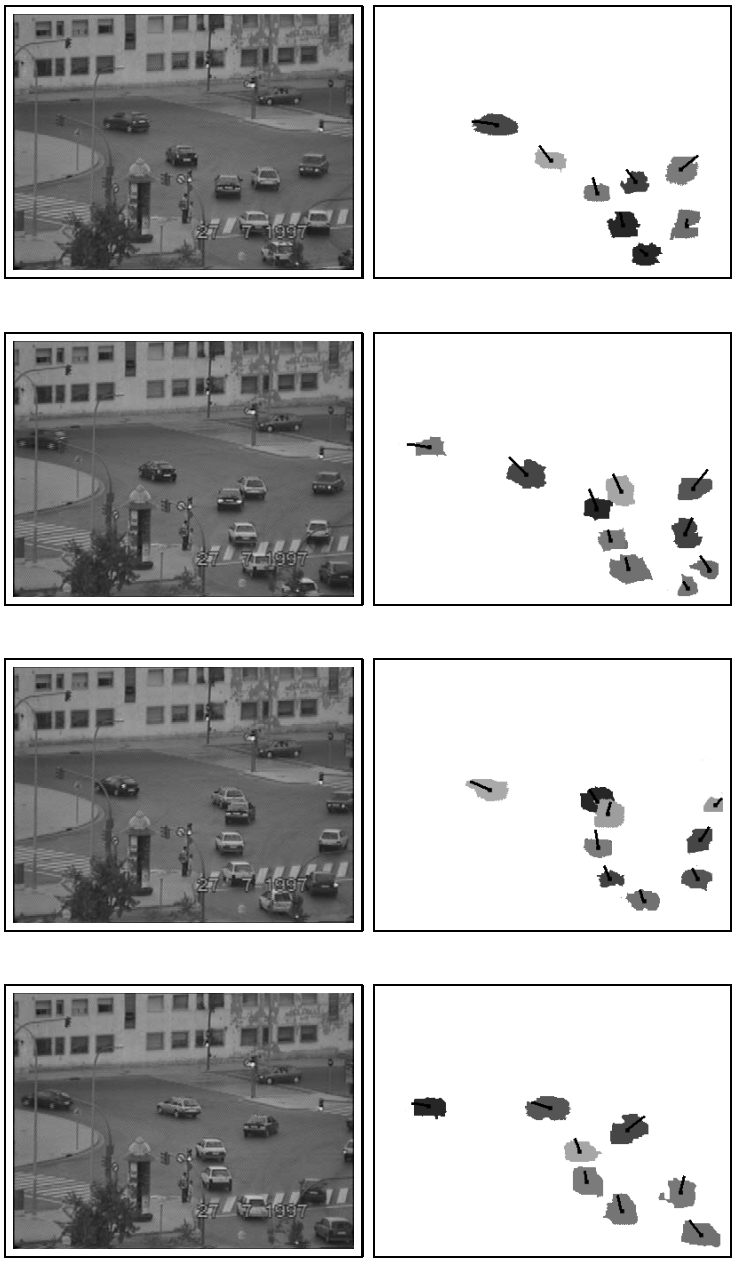











27 7 1997
27 7 1997
27 7 1997
27 7 1997