Motion and Intensity-based Segmentation and its Application to Traffic Monitoring*

Jorge Badenas[1], Miroslaw Bober[2], Filiberto Pla[1]

[1] Dept. Informática, Universitat Jaume I, Castellón 12071 (SPAIN)
[2] Dept. Elec. & Elec. Eng., University of Surrey, Guildford, Surrey GU2 5XH,(U.K.)

Abstract. 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 and facilitate 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 a good performance can be achieved even under such apparently restrictive assumptions. To further reduce processing time, we perform gray-level based segmentation that extracts regions of uniform intensity. Subsequently, we estimate motion for the regions. Regions moving with the coherent motion are allowed to merge. The use of 2D motion analysis and the pre-segmentation stage significantly reduces the computational load, and the region-based estimator gives robustness to noise and changes of illumination.

1 Introduction

Motion estimation and segmentation is one of the fundamental problems in image sequence processing. In recent years, as a result of advances in information technology both in terms of computational power and cost, it has become possible to use computer vision techniques to solving many everyday problems. A good example of such a task is traffic monitoring. Traffic monitoring system should be able to carry out many operations including estimation of traffic mean velocity, counting the number of moving vehicles, tracking individual vehicles and detecting whether traffic is moving freely or not.

One of the crucial components of the traffic monitoring system is the motion analysis algorithm. 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, and the camera may vibrate due to the weather conditions. The illumination conditions may be poor and may change rapidly. Multiple occlusions are likely and the environment may be cluttered. But probably the most challenging is the requirement of real-time or close to real-time performance on relatively cheap hardware. These specific difficulties and constraints require that

* The present work was supported in part by the projects ESPRIT PROJECT EP-21007 IOTA and CICYT TIC95-0676-C02-01

a standard of-the-shelf algorithm cannot usually be applied and a dedicated algorithm must be designed. In this paper we present a novel algorithm, which modifies and combines several known approaches to image segmentation and motion estimation. Before we explain the main premises behind our approach, we very briefly review the state of the art in motion estimation.

Motion analysis methods are usually divided into two groups: gradient-based methods and feature-based methods. *Gradient-based methods* exploit the relationship between the spatial and temporal gradients of intensity through the well-known *optic flow equation* [6]. These methods only work satisfactorily in regions where there are sufficient spatial intensity variations (texture) and where the motion between frames is relatively small. Unfortunately, in the case of traffic monitoring, objects exhibit little intensity variations and may be quite small, consequently ruling-out this class of approaches. *Feature-based methods* use features extracted from the frames such as interest points, corners, lines, zero crossing points or regions to determine a motion field by analysing the change of position of features through several frames. Here again, due to small size of the vehicle in the image plane, grey-level edges and corners are short and can be barely detected. Even if features are detected, a confusion during the feature-matching stage is quite likely as the environment is highly cluttered and there are many occlusions and disocclusions. Recently, Bober and Kittler [3] developed a *region-based* motion analysis technique called RHT. They combine the Hough Transform and Robust Statistical kernels. It has been shown that RHT can extract motion parameters (such as displacement or parameters of affine transformation) from two frames with excellent accuracy. Moreover, the technique offers simultaneous motion estimation and segmentation and is computationally efficient.

Great deal of work has been done in the area of motion estimation and segmentation to traffic monitoring. We shall very briefly mention several contributions that show the current trend. Dubuisson and Jain [4] describe a technique that combines motion and colour segmentation. In the paper [5] a system for traffic monitoring is described which fit 3-D wireframe models of generic vehicles to the vehicles projected onto the image plane. A similar approach is shown in [7] where the model of a generic car is projected from the 3-D scene onto the 2-D image. Methods for tracking of 2-D contours have been proposed in [2] and [1]. Weber at al [8], presented other method which represents contours by curves. It employs a contour tracker based on intensity and motion boundaries which uses two Kalman Filters.

The main premise behind our approach are that for the application at hand, an efficient and robust region-based motion analysis should be used. It is expected that region-based estimation is more robust that feature-based approaches, and is also likely to outperform the contour based trackers, which rely solely upon the outer contour of the tracked object. To further aid motion segmentation and reduce computational cost, we propose to apply a grey-level based segmentation as the pre-processing stage. Such approach removes motion-based ambiguity in regions of uniform grey-level and speeds-up motion segmentation by reducing

the number of passes in multi-pass stage.

The paper has the following structure. In next section, we present 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 the experimental results on real-word sequences and, finally, conclusions are drawn is section 7.

2 Outline of the Algorithm

The algorithm consists of three stages: pre-processing involving gray-level based segmentation, motion analysis (including further segmentation if needed) and post-processing stage, where regions are allowed to merge.

The segmentation of the reference image 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).

The first step of motion analysis stage attempts to reduce the number of regions in order to simplify the subsequent operations. This reduction is based on difference images analysis, and, at least, 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 postprocessing stage, 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.

To this end we obtaine a 2D segmentation map with the large background region and smaller foreground regions (cars). An accurate estimate of the translation is given to each region.

3 Segmentation of the Reference Frame

The purpose of the first stage is to obtain groups of pixels with similar intensity. Motion will be estimated for these regions, so it is very important that the pixels which are grouped are likely to belong to a single object.

The clustering algorithm used is a modification of the technique developed by Kottle and Sun and described in [9]. This technique is an adaptation of the classical k-means algorithm, employing a three-dimensional space of features: the two image coordinates and the pixels intensity. Pixels are assigned to one of the clusters in an iterative process. At each iteration a pixel i is assigned to a 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 to adapt itself to the image.

The original algorithm 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 too small number of regions is present, the clustering may group pixels from different objects, but too large a parameter used will cause over-segmentation and increase in computational load.

Our approach removes this problem by introducing a multistage segmentation where the original image is segmented initially into a relatively small number of clusters, and each cluster in turn is considered for further segmentation. Since in traffic scenes the road and cars do not exhibit much texture, the decision if a cluster should be further divided is based on the cluster intensity variance σ_j. In addition, 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. Therefore, a maximum region size η is also restricted. The cluster is divided if the following condition is fulfilled: $(N_j \geq \eta \; or \; (N_j \geq \mu \; and \; sigma_j^2 \geq \sigma_t))$ where N_j is the number of pixels in cluster j and σ_t is the variance threshold. The following values were used in experiments: $\sigma_t = 12$, $\mu = 1500$ and $\eta = 150$ (image size 192x144 pixels). 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 (eg fixed image resolution and camera location).

The proposed modification makes the technique more adaptable to the content of the images, and the final number of clusters is no longer fixed. We still 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. Furthermore, the minimum size of the cluster is now restricted, preventing from extreme over-segmentation. 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.

4 Motion Estimation

Since motion estimation is computationally heavy, one does not want to apply it to stationary regions. Therefore, in the first stage we perform a simple and crude test on each of the pre-segmented regions to determine if they are stationary

or not. 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). We follow pixels along the boundary and calculate what proportion of them has large frame difference.

In this step, static regions are merged and those 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.

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

$$E_i(dx, dy) = \frac{1}{N_j} \sum_{(x,y) \in Cluster_j} \rho(I_1(x, y) - I_2(x + dx, y + dy), \alpha, \lambda) \qquad (2)$$

where $I_1(x, y)$, $I_2(x, y)$ are the pixel intensity values at location (x, y) in the reference and consecutive frames respectively. $\rho()$ is the robust redescending function and α, λ are the function parameters. 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:

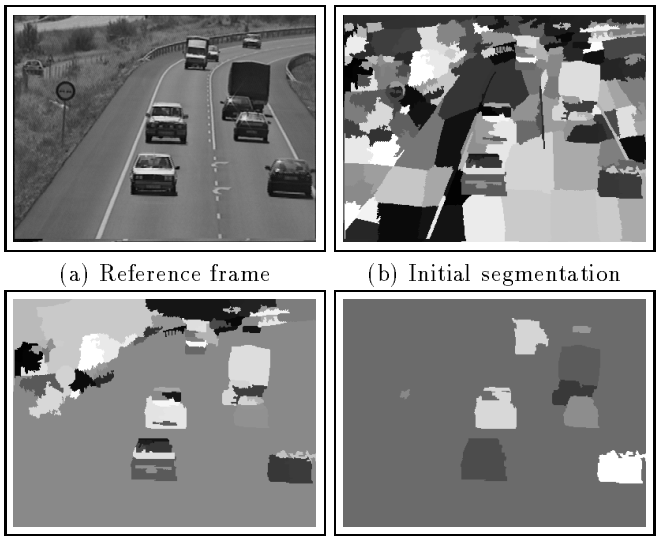
$$\rho(x, \alpha, \lambda) = \begin{cases} \lambda \, |x| \; if \; |x| < \frac{\alpha}{\lambda} \\ \alpha \quad \text{otherwise} \end{cases}$$

Our approach 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 terminates 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 value at inter-pixel locations.

5 Final Motion Segmentation

This final stage attempts to merge regions moving with coherent motion. This stage is needed, since the initial greylevel based segmentation may split object into smaller regions. All pairs of adjacent regions are considered as candidates



(a) Reference frame (b) Initial segmentation

(c) Merging regions by using difference images (d) Segmented regions

Fig. 1. Segmentation process of a sequence with nine vehicles.

to be merged. The two regions, say A and B, are merged if at least one of the following conditions is satisfied:

$$(E_{AB} \leq Q_1) AND (E_{AB} \leq E_{AA} + Q_2) \tag{3}$$

$$(E_{BA} \leq Q_1) AND (E_{BA} \leq E_{BB} + Q_2) \tag{4}$$

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 constants. Values assigned to Q_1 and Q_2 are: $5 \leq Q_1 < 9$ and $0 \leq Q_2 \leq 1$. This process is repeated for all pairs of adjacent regions until no pair can be merged.

6 Results

Figure 1 presents the segmentation for a sequence in which 9 vehicles are moving in both directions. 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.

For this sequence, it can be seen that eight vehicles, and not nine, are detected. There is a car that is moving onto 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 exits between the velocities of both vehicles and their distant position in relation to the camera, we believe that this can not be considered as a defect of the algorithm. In any case, the algorithm separates both vehicles after a few frames.

For the rest of vehicles shown in figure 1 we can see that they are correctly segmented, in spite of some of them hide parts of other vehicles.

In both sequences the camera was vibrating slightly, provoked by the wind. However, the algorithm can cope with these situations. Note that the method has successfully segmented the moving vehicles.

In subfigure 1.c the importance of the reduction of 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 time.

In the first stage, the parameters initial number of regions, σ, μ and η were chosen as 6, 12.0, 300, and 4000, respectively. In the motion segmentation stage, the parameters Q_1 and Q_2 received the values 8.0 and 1.0, respectively.

Table 1 represents the computational costs of the different parts of the algorithm when using diferent configurations for this sequence. These times have been measured on a Hewlett Packard workstation Apollo Model 725/75 with a processor PA-RISC 7100 (75 MHz).

The first row of the table shows the costs when the size of the image is 384x288. We can see that the computational cost of the two first stages is large in comparison with the third stage. The total cost is 47.8 seconds per frame. The algorithm can be speeded up by calculating an 'Activity Map' [10] that marks the pixels that are always static. These pixels are discarded since they never will be occupied by a vehicle. The time of computation of the algorithm when the 'Activity Map' is used is shown in the second row. Now, it has been reduced to 26.63 seconds. We can see that the cost of the *clustering stage is significatively reduced. However, the reduction in the* Motion Estimation stage is not so large, because the cost of the static pixels is small compared with the cost of estimating the motion for the moving pixels.

	Clustering	*Motion estimation*	*Motion segmentation*	*Total*
384x288	15.87	21.69	9.88	47.8
384x288 and A.M.	5.84	14.54	6.25	26.63
192x144 and A.M.	1.55	5.45	1.03	8.03

Table 1. Computational cost (seconds) of the algorithm.

Since 384x288 can be considered as a big image size, and the algorithm does not require so much resolution, we can reduce the computational cost of the algorithm by reducing the size of the images. The third row of the table 1 shows the computational costs of the algorithm when using a 192x144 image size and an 'Activity Map' (A.M.). Now, the total time of computation is 8.03 seconds.

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 existing techniques and algorithms.

The clustering algorithm is an improved variant of the method presented in [9] in which we have carried out modifications. With these modifications is not necessary to indicate the final number of regions. We have also developed a method based on difference images which reduce the number of regions by uniting 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 allows to avoid the local minima of the *Displaced Frame Difference* function and speeds up the computational process. In order to achieve a reliable motion estimation, we use a *redescending robust kernel*. The final motion segmentation is achieved by merging regions moving with coherent motion.

We have demonstrated that a motion analysis based on a translational model is sufficient for the segmentation purpose when 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.

The proposed method proved to be robust to camera vibrations and it copes well with multiple moving objects. Unlike some other techniques it does not use feature points and it consequently works well in a cluttered environment.

References

1. A. Blake, R. Curwen, and A. Zisserman. Affine-invariant contour tracking with automatic control of spatiotemporal scale. In *Proceedings of the Fourth International Conference on Computer Vision, Berlin, May 1993*, pages 66–75, 1993.
2. A. Blake, R. Curwen, and A. Zisserman. A framework for spatio-temporal control in the tracking of visual contours. pages 127–45, 1993.
3. M. Bober and J. Kittler. Estimation of general multimodal motion: an approach based on robust statistics and Hough transform. *Image and Vision Computing*, 12(12):661–668, 1994.
4. M.P. Dubuisson and A.K. Jain. Contour extraction ofmoving objects in complex outdoor scenes. *International Journal of Computer Vision*, 14:83–105, 1995.
5. J.M. Ferryman, A. D. Worrall, G.D. Sullivan, and K.D. Baker. A generic deformable model for vehicle recognition. In *BMVC95*, pages 128–136, 1995.
6. B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
7. D. Koller, K. Daniilidis, and H.H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. *I.J.Computer Vision*, 10:257–281, 1993.
8. D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In Jan-Olof Eklundh, editor, *Proceedings, 5th European Conference on Computer Vision, (Berlin, 1994)*, pages 189–196. Springer-Verlag, 1994.
9. Kottle and Sun. Motion estimation via cluster matching. *PAMI*, 16, 1994.
10. B.D. Steward, I. Reading, M.S. Thomson, T.D. Binnie, K.W. Dickinson, and C.L. Wan. Adaptative lane finding in road traffic image analysis. pages 133–136. IEEE Conference Publications, 1994.

This article was processed using the LaTeX macro package with LLNCS style