Object Segmentation in Videos from Moving Camera with MRFs on Color and Motion Features

Rita Cucchiara, Andrea Prati, Roberto Vezzani
D.I.I.- University of Modena and Reggio Emilia - 41100 Modena, Italy
{cucchiara.rita,prati.andrea,vezzani.roberto}@unimore.it

Abstract

In this paper we address the problem of fast segmenting moving objects in video acquired by moving camera or more generally with a moving background. We present an approach based on a color segmentation followed by a region-merging on motion through Markov Random Fields (MRFs). The technique we propose is inspired to a work of Gelgon and Bouthemy [6], that has been modified to reduce computational cost in order to achieve a fast segmentation (about ten frame per second). To this aim a modified region matching algorithm (namely Partitioned Region Matching) and an innovative arc-based MRF optimization algorithm with a suitable definition of the motion reliability are proposed. Results on both synthetic and real sequences are reported to confirm validity of our solution.

1. Introduction

Moving object segmentation in videos is a key process in a large number of multimedia and computer vision applications. Most of the difficulties associated to this task are due to the presence of a relative motion of the observer with respect to the motion of objects and background. When the background is stationary and videos are acquired by fixed cameras, moving object detection is a relatively "easy" problem: well know methods as background suppression and frame differencing are implemented with success in many applications. Instead, the segmentation of moving objects becomes more critical when the video is acquired by a moving camera with an unconstrained and a priori unknown motion. More in general, we consider the framework of videos with moving foreground objects on a moving background. In this case, segmentation by computing visual motion only is not always reliable, and other features must be exploited such as color, shape, texture and so on.

Several researchers have approached the problem of motion detection on video acquired by moving camera. The proposed solutions can be mainly differentiated by several features such as whether they explicitly compute egomotion or not, which motion model (translational, affine, quadratic, etc.) is assumed, which features are used to segment the video (visual features as color, or motion, or both), and whether they achieve real-time performance or not.

Accordingly, we could group most of the proposals into three classes of approaches:

1. *based on egomotion computation*: these methods initially compute egomotion and, after the compensation of the camera motion, they apply an algorithm defined for fixed-camera. Examples are [2][8][4][9][11][15];

2. *based on motion segmentation*: the objects are mainly segmented by using the motion vector computed at pixel level, as in [3][7][10][13]

3. *based on region merging with motion*: the objects are obtained with a segmentation using several visual features, and then merged on motion parameters computed on a region-level ([5] use gray level as visual feature, [6] uses color, [12] exploits edges). Our approach belongs to this class.

Among the papers in the third class, a previous work of Gelgon and Bouthemy [6] proposes an initial color-based segmentation followed by a motion-based region merging with an affine motion model. The merging of regions is performed with Markov Random Fields (MRF, hereafter). An implicit tracking of the objects is also included, and it is obtained with the initialization of the MRF according with a label prediction. This approach is based on two basic assumptions on the objects: they must have i) different colors (the segmentation of the scene in objects is based only on color information), and ii) rigid motion (expressible with affine equations). This solution produces good results, but the computational complexity is too high to allow real-time implementation: authors in [6] state to be able to process a frame every 80 seconds (on a ULTRASPARC Machine).

Indeed, few proposals take into account the time requirements; among these, many use a special purpose hardware

[2][10], or a manual initialization [14], or are oriented to a specific application (as [4] for periodic model). [9] and [13] are general purpose but they do not assure the extraction of the entire moving objects because they make use only of the motion information in the segmentation process.

In this paper we present an approach inspired to the one of [6] substantially modified in order to abate the computational time. The goal is to provide a fast segmentation that could be adopted in an on-line video segmentation process, suitable for applications such as indoor/outdoor surveillance, video-conferencing, multimedia on-the-fly transcoding, and augmented reality. To this aim our algorithm assumes some simplifications with respect to [6], but also some important new features. The basic simplification is the hypothesis of a translational motion model instead of an affine one: this hypothesis limits the applicability but increases speed considerably.

On the other hand, we introduce some important novelties:

- the "Partitioned Region Matching": a new motion estimation algorithm based on region matching but emphasizing advantages of both region matching and block matching;
- a measure of "motion reliability" in the MRF model;
- a new minimization algorithm of the MRF function that approximates the classical approach with a search based on arcs instead that on nodes. In this way we have defined a technique for very fast segmentation reaching up to 10 frames per second (on a dual Pentium III 1000 MHz with 512 MB of RAM) in frames of 100x100 pixels.

2. The Proposed Approach

Our approach can be described as follows: each frame of the sequence is segmented into regions by using color as in [6]. For this task a region growing algorithm is applied. According with color segmentation, a *Spatial Region Graph (SRG)*, also called adjacency graph, is created. In SRGs, nodes represent regions, whereas arcs represent topological adjacencies. We consider an attributed graph: area size, the coordinates of the bounding-box and the centroid are associated with each node. Then, *motion estimation* is computed with the adoption of the translational motion model. To accomplish to this task there are basically two ways: the first requires *pixel-level motion estimation* (e.g., with optical flow or block matching) followed by a statistical function (e.g., mean or mode) over regions; the second (adopted in our system) directly exploits computation of the *region-level motion vectors* through region matching (and in particular with *Partitioned Region Matching*, described in the following). In addition to motion vector a measure of *motion reliability* is defined and computed for each region, based on the presence of "strong" gradients (that allow correct estimation of the motion) and a good match. Computed information are then stored in the spatial graph. Then SRG is the starting point of a Markov Random Field (MRF) framework, where regions are merged according with an energy function influenced by the motion parameters (velocity and reliability). The largest region is assumed as background and its motion as egomotion, while other regions are classified as moving objects.

2.1. Region Segmentation and Motion Estimation

According with [6], we mix color feature and motion and exploit MRF to optimize the spatial region graph in order to extract objects as regions with similar motion parameters. Nevertheless, three substantial differences have been introduced in all sub-tasks and will be described in the following.

First, we changed the color segmentation method. In [6] color segmentation is performed at the first frame only and then refined frame by frame by means of the MRF. Instead, focusing on speed, MRF optimization has been substituted by a color (re-)segmentation at each frame. This last solution has proved to be less accurate but much faster.

Thus, color segmentation is implemented with a region growing algorithm. Since our goal is moving object detection, we work principally on an object-level and a perfect color segmentation at pixel level is not required. Region growing is provided as in [1]: starting from a seed point, the region is grown by merging 4-connected pixels with a "similar" color. The similarity between a pixel $\mathbf{x}$ and the region R is computed with the simple Manhattan distance in the RGB space. An α parameter is used as distance upper-bound to guide the merging phase: by relaxing α, larger regions (minor precision) are created and the further steps of the process result to be faster. Conversely, this simple growing algorithm could create small regions in presence of strong edges or noise that are included in adjacent regions in a second step of refinement.

The second modification to the approach presented in [6] is in the motion model and its estimation. We have adopted the translational motion model, less realistic than the affine one but characterized by a much lower complexity. However, when camera and objects are moving slowly the simplification results acceptable. With translational model, fast correlation based algorithms could be sufficient for motion estimation, such as block or region matching.

When a previous color segmentation is available, as in this case, region-matching techniques produce often better results than block matching in a relative short time. To this aim, we present here a variation of the classic region matching. Differently from the block matching algorithms, that compare fixed sized blocks, region matching exploits

the whole regions extracted with segmentation as comparing patterns and a function defined over the entire regions as matching value (i.e. distortion rate). For example, the *SAD* (Sum of Absolute Differences) can be computed and evaluated over a region.

Given a region R, being $R(\mathbf{v})$ the set of pixel corresponding to the region R after a shift of a vector $\mathbf{v} \triangleq (v_x, v_y)$, we can define:

$$SAD_R(\mathbf{v}) = \sum_{\mathbf{x} \in R} dist(\mathbf{x}, \mathbf{x}(\mathbf{v})) \quad (1)$$

where $\mathbf{x}(\mathbf{v})$ is the $\mathbf{x}$ point after a shift $\mathbf{v}$, and $dist$ is a Manhattan distance function.

The most important advantage of region matching with respect to block matching is the possibility to estimate motion of uniform regions without texture, by using the shape. In fact, the presence of at least two not-parallel gradients is required to correctly compute a motion vector (aperture problem). The adoption of regions in substitution of blocks make more probable that the previous requirement will be satisfied, both for the bigger size and for the presence of the object border inside the region. At the same time, problems with region matching based on SAD can arise when the motion is not strictly translational or when the shape varies over time (for example, in presence of occlusions). To reduce this problem, we define a new matching criterion that we call *Partitioned Region Matching* (PRM in the following), that ensembles features of both block and region matching.

PRM is defined as follows. First, a region R is partitioned in a number n of disjoint sub-regions SR^k, with $k = 1, ..., n$. For each sub-regions SR^k the function $SAD_{SR^k}(\mathbf{v}_j)$ is evaluated. Then, an *a posteriori* probability function $P(\mathbf{v})$ is associated to the region and it is computed as the product of the number of occurrences of each motion vector with the *a priori* probability.

$$P(\mathbf{v}) = \frac{\frac{\sum_{k=1}^{n} \delta(\mathbf{v}, \mathbf{v}^k)}{n} \cdot P_{priori}(\mathbf{v})}{z} \quad (2)$$

where $\delta(\mathbf{v}, \mathbf{v}^k)$ is 1 if $\mathbf{v} = \mathbf{v}^k$ and 0 otherwise, and z is the normalization factor. The *a priori* probability could be computed taking into account the motion in the previous frame, the physical constraints, and the knowledge on the scene or application. To reduce computational complexity, in the experiments reported in this paper the *a priori* probability has been set identically to $\frac{1}{|\mathbf{V}|}$.

Finally, the motion vector $\mathbf{MV}_R = (MVx_R, MVy_R)$ of the whole region R is assumed to be the vector $\mathbf{v}^j$ that maximizes $P(\mathbf{v})$.

The defined PRM reduces the problems of occlusion typical of region matching, since occlusions affect only a limited number of sub-regions that are not significant due to the equation 2; moreover, it resolves the drawback of motion vector locality of block matching, working on larger patterns.

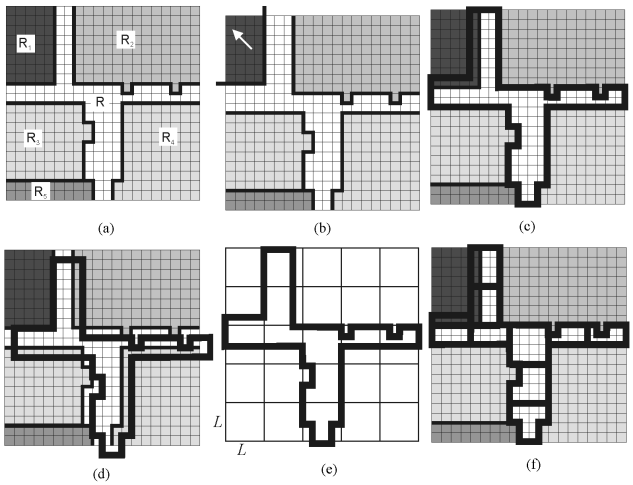


Figure 1. Example of PRM

	R_1	R_2	R_3	R_4	R_5		
$dist(R, R_j)$	235	55	25	15	205	$\widetilde{SAD}_R$	$P(\mathbf{v})$
$\#overl(R, R_j)$ $\mathbf{v_1} = (0,0)$	12	0	0	0	0	2820	0.7
$\#overl(R, R_j)$ $\mathbf{v_2} = (1,1)$	0	7	7	18	0	830	0.1

Table 1. Results on the example of Fig. 1

To better understand and appreciate the power of this solution we can consider the two consecutive frames in Figg. 1(a) and 1(b), where only R_1 is in motion. Moreover, to help in understanding this example we consider that the regions are characterized by an uniform color. Under these assumptions, the SAD of Eq. 1 could be approximated in the following way:

$$\widetilde{SAD}_R(\mathbf{v}) = \sum_{j=1}^{r} \#overl(R_i, R_j) \cdot dist(\overline{R}_i, \overline{R}_j) \quad (3)$$

where $\#overl(R_i, R_j)$ is the number of pixels overlapped between region R_i and R_j.

To compute motion of R, we search the best match between the second and the first frame. In Figg. 1(c) and 1(d) are represented two possible matches, obtained with $\mathbf{v_1} = (0,0)$ and $\mathbf{v_2} = (1,1)$, respectively. Even if the first is the correct match, $SAD_R(\mathbf{v_1})$ is higher than $SAD_R(\mathbf{v_2})$ since the distance between points of R and R_1 is high (see Table 1). Thus, the classical SAD-based approach fails, since the lowest SAD is associated with $\mathbf{v_2}$. This problem often occurs in real sequences, when the foreground moving

objects have very different colors from the background. In this case the background's motion may be wrongly evaluated as equal to the motion of a foreground occluded object. With PRM, instead, we can estimate the correct motion vector. The region R is partitioned in 10 sub-regions as in Fig. 1(e). Among them, only 3 sub-regions (Fig. 1(f)) are affected by the occlusion of R_1, while most of the sub-regions can correctly estimate the motion vector. Consequently, the probability $P(\mathbf{v})$ associated with $\mathbf{v_1} = (0,0)$ is correctly the highest.

For each region R we also define a *motion reliability* as:

$$\rho_R = \frac{\sum_{(x,y)\in R}\left[\sum_{i\in\{R,G,B\}}\left(\left|\frac{\partial I(x,y)_i}{\partial x}\right| + \left|\frac{\partial I(x,y)_i}{\partial y}\right|\right)\right]}{SAD_R(\mathbf{MV}_R)+1} \quad (4)$$

ρ_R takes into account two factors: the gradient in the color image and the goodness of the best match found, evaluated as the SAD increased by 1 to prevent division by 0. If a region presents a low texture, the choice of the best match could be ambiguous, therefore the reliability of the motion estimation is low too. Reliability is low also if the best match presents an high distortion and thus an high SAD (e.g., in presence of occlusions and deformations). Where the motion can not be reliably estimated the motion reliability is set to 0.

2.2. Markov Random Field Computation

A Markov Random Field framework can be exploited to merge regions with similar motion. In this subsection we first discuss the approach directly derived from the proposal reported in [6]; then, in subsection 3.2 a new formulation of the MRF will be presented to improve computational time.

As proposed in [6], the energy function can be decomposed into three terms: one based on motion (U_1), one used to perform a geometrical regularization (U_2), and one based on the number of assigned labels (U_3):

$$U(e,o) = U_1(e,o) + U_2(e) + U_3(e) \quad (5)$$

where e and o are respectively the labels and the observation fields.

Differently from [6] we also included motion reliability in the energy term based on motion:

$$U_1(e,o) = \sum_{(s,t)\in\Gamma} V_1(e_s,o_s,e_t,o_t) \quad (6)$$

$$V_1(e_s,o_s,e_t,o_t) = \begin{cases} 0 & e_s \neq e_t \\ \dfrac{c_1 \cdot \sqrt{(MVx_s - MVx_t)^2 + (MVy_s - MVy_t)^2}}{\sqrt{\rho_s \cdot \rho_t}} & e_s = e_t \end{cases} \quad (7)$$

where Γ is the set of two-dimensional cliques and s and t are two sites of a clique (corresponding to regions); MVx and MVy are the two components of the motion vector, ρ is the motion reliability and c_1 is a positive constant.

The terms U_2 and U_3 are kept unchanged with respect to [6] and are defined as follows:

$$U_2(e) = \sum_{(s,t)\in\Gamma} V_2(e_s,e_t) \quad (8)$$

$$V_2(e_s,e_t) = \begin{cases} 0 & e_s \neq e_t \\ -c_2 \dfrac{\xi_{s,t}}{\xi_{s,t} + \sqrt{(G_x^s - G_x^t)^2 + (G_y^s - G_y^t)^2}} & e_s = e_t \end{cases} \quad (9)$$

where c_2 is a positive constant, ξ is th length of the shared border and $\mathbf{G} = (G_x, G_y)$ is the region centroid. This term enhances the fusion of two adjacent regions with a long shared border and a small distance between centroids.

The last term takes into account the number of labels assigned, i.e. the cardinality of e ($\#e$). Thus, $U_3 = c_3 \cdot \#e$, where c_3 is a positive constant.

In conclusion, the motion based term tries to keep separate regions with different motion, geometrical term decrements the weight of motion distance if there is a strong adjacency and U_3 represents a sort of motion difference threshold under which two regions will be merged.

The optimization of the energy function is performed with a multi-scale iterative algorithm. A label and a binary stability flag (preset to "unstable") are assigned to each node. The algorithm extracts one by one the regions with "unstable" flag and evaluates which label among the old label, the labels of the adjacent nodes and an outlier label minimizes the energy function. With the new label the node becomes "stable" but if the label has been changed, the adjacent nodes become "unstable". When all the nodes become stable, nodes with the same label are grouped together and the algorithm is iteratively run on the new graph.

After MRF optimization, we obtain a final graph with each node corresponding to an object (or a blob containing objects with the same motion). The largest one is assumed to be the background. Moreover, the background could be separated in more parts by interposed objects. To prevent this, every region with the same motion of the background is merged with it, even if not adjacent.

3. Experimental Results

3.1. Efficacy Analysis

To evaluate the performance in terms of efficacy we use ground truth images to compare with. Results are summarized in Table 2, where the average false positives (F_P) and false negatives (F_N) are reported (numbers in brackets in third and fourth rows report the percentage of false positives/negatives with respect to the whole image, while the

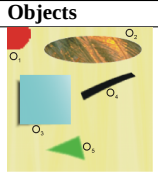
	Objects	Indoor	Hand
Snapshot			
Type	Syntetic	Real	Real
Info	20 frames (200x200)	170 frames (350x275)	90 frames (240x200)
F_P	0.03% (0.01%)	5.52% (0.66%)	20.03% (4.01%)
F_N	7.10% (1.44%)	12.8% (1.14%)	5.02% (0.50%)
% of mov. obj.	20.54%	10.49%	17.03%

Table 2. Efficacy results on the benchmark.

fifth refers only to objects in motion and not to the background). Thus, false negatives are moving objects' points that are not detected and false positives are points of still objects or of the background labeled as belonging to moving objects. In the synthetic sequence (with objects of different textures, shadowed and smoothed) the false negatives are mainly due to the assignment to the background of the border's pixels of shadowed or vanishing objects.

The "Indoor" sequence (where a people is moving in opposite direction w.r.t. to the camera) produces good average results. By observing Fig. 2, we can note the presence of false positives, caused by a wrong motion estimation of the background region at the right of the man. Instead, the false negatives are caused by the very slow motion of the person.

Lastly, in "Hand" sequence a hand is moving over a desk and the camera is in motion too. This sequence is very hard to analyze, because the background is composed by a lot of small regions (of the mouse pad) with some large and fine-textured ones (of the desk) in addition. Motion estimation is difficult on both: on the small regions because of the occlusion of the hand and on the white desk for the absence of gradients. In this case the false positives reported in Table 2 are some details of the mouse pad attached to the hand.

3.2. Improving Computational Costs

All the experiments testify that the efficacy results are acceptable when the motion can be assumed translational. Similar results were reported in [6] for different test sequences. However, their approach is known to be computationally expensive.

Fig. 3 reports the processing time in dependence of the number of regions. Note that color segmentation, spatial graph creation and motion estimation are almost independent from the number of regions.

The computational time of MRF optimization phase depend on the number of region obtained with the color segmentation. Calling n the number of nodes of the graph, i.e. number of regions, the minimization algorithm proposed in

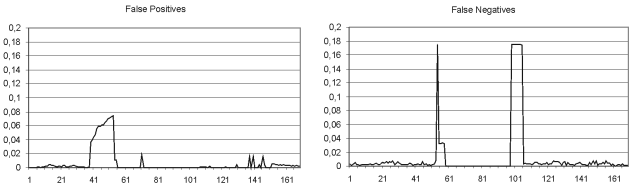


Figure 2. False positives and negatives over the Indoor sequence

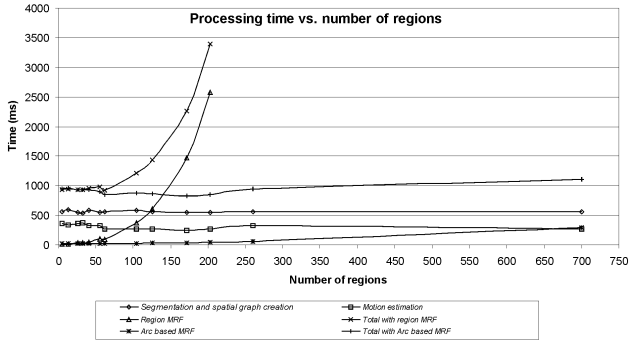


Figure 3. Processing time vs. number of regions (on object sequence 200x200) compared with the arc based MRF optimization.

section 2 has a complexity of $O(n^4)$. In fact, each iteration analyzes all the n nodes; for each of them it computes the energy value on the n arcs linking it with the others. If the state of the analyzed node changes, all the other n nodes may be re-evaluated. Supposing that each node is evaluated n times, we obtain a complexity of $O(n^3)$. Then a multi-scale optimization is performed, that consists, in the worst case, in n iterations.

As indicated by the total time costs in Fig. 3 (Region MRF), we are far from satisfying real time constraints. We propose a sub-optimal MRF energy minimization algorithm based on the analysis of the arcs (and pairs of linked nodes) only, instead of the analysis of each node and all its adjacent nodes. For this aim, the energy function previously presented has been modified; in particular, the term U_3 is now defined as the number of *broken arcs* (i.e. pairs of regions geometrically adjacent but with different labels). Consequently, analyzing the definitions of the three energy terms we can define a unified local term as:

$$U(r,o) = \sum_{(s,t)\in\Gamma} U_{s,t}(e_s, e_t, o_s, o_t) \tag{10}$$

where:

$$U_{s,t}(e_s, e_t, o_s, o_t) = \begin{cases} V_1(e_s, o_s, e_t, o_t) + V_2(e_s, e_t) & e_s = e_t \\ V_3(e_s, e_t) & e_s \neq e_t \end{cases} \tag{11}$$

$$V_3(e_s, e_t) = \begin{cases} 1 & e_s \neq e_t \\ 0 & e_s = e_t \end{cases} \quad (12)$$

The main advantage of Eq. 11 consists on the possibility of a faster optimization. The algorithm here proposed is composed by two scanning phases: the first on the arcs and the second on the nodes. The computational complexity becomes quadratic $O(n^2)$ (evaluation of n^2 arcs plus n nodes).

For each arc (s, t) we evaluate whether it is more convenient to break it or not, by considering only the sub-graph composed by the arc and the two connected nodes (s and t), that is:

$$V_1(e_s, o_s, e_t, o_t) + V_2(e_s, e_t) \geq V_3(e_s, e_t) \Rightarrow (s, t)\ broken \quad (13)$$

After that, the nodes of the graph are labeled using a modified region growing algorithm. Starting from the first node, all the nodes connected with non-broken arcs are merged with the same label.

Thus, the new optimization allows us to achieve real-time performance by preserving segmentation results. The new system was tested on the same sequences used previously. In the "Objects" sequence we obtained identical results. Instead, in the case of the "Hand" sequence the output is slightly worst with the new optimizer, because wrong motion values are assigned to part of small regions affected by occlusion. Anyway, the results are still satisfactory. Similar performance results in efficacy has been tested in the "indoor sequence" and in other videos, but they are not reported due to space limitations. The graph in Fig. 3 shows the improvement of speed obtained with the arc based optimization instead of the region one.

4. Conclusions

We proposed a fast approach based on color and motion segmentation with MRF for moving object detection in video acquired by moving camera or more generally with a moving background. The method achieves satisfactory results when the motion can be approximated with a translational model. A new Partitioned Region Matching has been proposed to perform a good motion estimation also in presence of occlusion or shape variation; moreover, a motion reliability value in the MRF allows to weight differently the motion or the geometry of regions in case of low texture or large shape changes.

Then, a new sub-optimal energy minimization algorithm applicable on this particular case of Markov Random Fields has been defined. With its introduction, we can process a sequence composed by frame of 100x100 pixels with a frame rate of about 10 fps on a PC platform.

This research is funded by the project "Domotics for disability" granted by the Fondazione CRM.

References

[1] R. Adams and L. Bischof. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, June 1994.

[2] S. Araki, T. Matsuoka, H. Takemura, and N. Yokoya. Real-time tracking of multiple moving objects in moving camera image sequences using robust statistics. In *Proceedings of Int'l Conference on Pattern Recognition*, volume 2, pages 1433–1435, 1998.

[3] M. Chang, A. Teklap, and M. Sezan. Simultaneous motion estimation and segmentation. *IEEE Transactions on Image Processing*, 6(9):1326–1333, Sept. 1997.

[4] R. Cutler and L. Davis. Robust real-time periodic motion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):781–796, Aug. 2000.

[5] F. Dufaux, F. Moscheni, and A. Lippman. Spatio-temporal segmentation based on motion and static segmentation. In *Proceedings of IEEE Int'l Conference on Image Processing*, volume 1, pages 306–309, 1995.

[6] M. Gelgon and P. Bouthemy. A region-level motion-based graph representation and labeling for tracking a spatial image partition. *Pattern Recognition*, 33:725–740, 2000.

[7] C. Hennebert, V. Rebuffel, and P. Bouthemy. A hierarchical approach for scene segmentation based on 2d motion. In *Proceedings of Int'l Conference on Pattern Recognition*, volume 1, pages 218–222, 1996.

[8] S. Jehan-Besson, M. Barlaud, and G. Aubert. Region-based active contours for video object segmentation with camera compensation. In *Proceedings of IEEE Int'l Conference on Image Processing*, volume 2, pages 61–64, 2001.

[9] K. Lee, S. Ryu, S. Lee, and K. Park. Motion based object tracking with mobile camera. *Electronics Letters*, 34(3):256–258, Mar. 1998.

[10] Y. Ninomiya, S. Matsuda, M. Ohta, and Y. Harata. A real-time vision for intelligent vehicles. In *Proceedings of IEEE Intelligent Vehicle Symposium*, pages 315–320, 1996.

[11] N. Paragios, P. . Perez, G. Tziritas, C. Labit, and P. Bouthemy. Adaptive detection of moving objects using multiscale techniques. In *Proceedings of IEEE Int'l Conference on Image Processing*, volume 1, pages 525–528, 1996.

[12] P. Smith, T. Drummond, and R. Cipolla. Segmentation of multiple motions by edge tracking between two frames. In *Proceeding of British Machine Vision Conference*, pages 342–351, 2000.

[13] S. Smith and J. Brady. Asset-2: Real-time motion segmentation and shape tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):814–820, Aug. 1995.

[14] A. Techmer. Contour-based motion estimation and obejct tracking for real-time applications. In *Proceedings of IEEE Int'l Conference on Image Processing*, volume 3, pages 648–651, 2001.

[15] T. Tian, C. Tomasi, and D. Heeger. Comparison of approaches to egomotion computation. In *Proceedings of IEEE Int'l Conference on Computer Vision and Pattern Recognition*, pages 315–320, 1996.