An Iterative Region-Growing Algorithm for Motion Segmentation and Estimation

Raul Montoliu,* Filiberto Pla†
Departamento Lenguajes y Sistemas Informáticos, Universitat Jaume I, 12071 Castellón, Spain

This article presents a new framework for the motion segmentation and estimation task on sequences of two gray images without a priori information of the number of moving regions present in the sequence. The proposed algorithm uses temporal information, by using an accurate Generalized Least-Squares motion estimation process, and spatial information, by using an iterative region-growing algorithm that classifies regions of pixels into the different motion models present in the sequence. The initial regions of pixels are obtained from a given gray-level segmentation process. The performance of the algorithm is tested on synthetic and real images with multiple objects undergoing different types of motion.

1. INTRODUCTION

Segmentation of moving objects in a video sequence is a basic task for several applications of computer vision, for example, a video monitoring system, intelligent-highway system, tracking, airport safety, surveillance tasks, and so on. In this article, motion segmentation, also called spatial-temporal segmentation, refers to labeling pixels that are associated with different coherently moving objects or regions in a sequence of two images. Motion estimation refers to assigning a motion vector to each region (or pixel) in an image.

Perfoming motion estimation and motion segmentation simultaneously usually mimics a *hen-and-egg* problem. This is due to the fact that data classification and parameter estimation strongly depend on each other. It is known that, on the one hand, if the data are well classified, that is, we know which pixels support which model, then it is easy to obtain accurate estimates for the parameters. On the other hand, if we know accurate estimates of the parameters, then it is straightforward to classify the pixels into the models.

The motion segmentation and estimation problem has been formulated in many different ways.[1–6] We choose to approach this problem as a multistructural para-

*Author to whom all correspondence should be addressed: e-mail: montoliu@uji.es.
†e-mail: pla@uji.es.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 20, 577–590 (2005)
 Published online in Wiley InterScience (www.interscience.wiley.com). • DOI 10.1002/int.20082

metric fitting problem. In this context, the segmentation problem is similar to a robust statistical regression. The main difference is that a robust statistical regression usually involves statistics for data having one target distribution and corrupted with random outliers. Motion segmentation problems usually have more than one population with distinct distributions and not necessarily with a population having absolute majority.

The problem of fitting an a priori known model to a set of noisy data (with random outliers) was studied in the statistical community for a number of decades. One important contribution was the Least Median of Squares (LMedS) robust estimator, but it has a breakdown point of 50%. This means that the LMedS technique needs the population recovered to have at least a majority of 50% (plus 1). Other robust estimators have been developed to overcome this problem, which is frequently encountered in different computer vision tasks. They are Adaptive Least *k*th Order residual (ALKS)[7] and Minimum Unbiased Scale Estimator (MUSE).[8] These techniques minimize the *k*th order statistic of the square residuals where the optimum value for the *k* is determined from the data. The problem of both techniques is that the estimation of the correct value of *k* suffers from high computation effort.

To overcome the computational complexity Bab-Hadiashar and Suter presented a method called the Selective Statistical Estimator (SSE),[9] which is a variation of the least *k*th order statistic data regression where the user proposes the value *k* as the lower limit of the size populations one is interested in. All the motion segmentation LKS-based algorithms start selecting an initial model using random sampling and classifying all the pixels into this model using a scale measure. With the remaining pixels the process is repeated until all the pixel have been classified. The main problem of these algorithms is that there are frequently pixels that more suitably belong to one model but they have been classified in an earlier model.

Danuser and Stricker[10] presented a similar framework for parametric model fitting. Their algorithm has a fitting step that is one component of the algorithm that also collect model inliers, detects data outliers, and determines the a priori unknown total number of meaningful models in the data. They apply a quasi-simultaneous application of a general least squares fitting while classifying observations in the different parametric data models. They applied their algorithm to multiple lines and planes fitting tasks. The most important advantages with respect to LKS-based algorithms are the use of an exchange step, which permits change of observation among models, and the use of an inliers/outliers classification process, which increases the accuracy of the segmentation.

In Ref. 11, a quasi-simultaneous motion segmentation and estimation method based on a parametric model fitting algorithm was presented. The method accurately estimates the affine motion parameters using a generalized least squares fitting process. It also classifies the pixels into the motion models present in two consecutive frames. This algorithm uses each pixel of the image as an observation. It suffers from problems of isolated points because it does not use neighborhood information and needs good initial models to obtain the final motion segmentation. Nevertheless, it indicates that the quasi-simultaneous application of the inliers/

outliers classification algorithm and the accurate motion estimator can be useful if applied in motion segmentation tasks.

This article presents a motion segmentation and estimation algorithm that, instead of using the pixel as an observation, uses regions of pixels. The use of regions made the segmentation more spatially consistent. In addition, the algorithm uses neighborhood constraints to collect new inliers to the model; only regions that are neighbors of the model are considered to be inliers. This algorithm also erases the need of a previous good segmentation of the models, and allows extracting the model without a priori information of the number of moving regions present in the sequence.

Our algorithm has been designed to be applied to general purpose motion segmentation problems, without a priori information about the origin of the images. In more specific problems, the knowledge of some properties of the scene can help in obtaining accurate segmentation. For instance, in traffic scenes the background (the road) usually is static and therefore can be removed, simplifying the segmentation process. However, this assumption cannot always be made in other problems. For this reason, our algorithm has been designed to be applied to all kind of motion segmentation problems. No specific information about the scenes, like the existence of static regions, the size and the shape of the objects, the motion of the sensor used to captured the images, and so forth, is given.

The rest of the article is organized as follows: Section 2 explains the complete motion segmentation and estimation algorithm. Section 3 presents a set of experiments to verify the results obtained with our approach. Finally, some conclusions drawn from this work are described.

2. ALGORITHM OUTLINE

In this article the following terminology is used:

- **Model** as a structure with two elements; the first is a parametric motion vector and the second is a list of regions of the image that support the parametric motion vector.
- **Region** as a set of pixels with gray-level coherence.
- **Inlier** as an observation that supports the motion of a model, that is, it has a very high likelihood of performing the motion of the model.
- **Outlier** as an observation that does not support the motion of a model, that is, it has a very low likelihood of performing the motion of the model.

The inputs of the algorithm are two images of a sequence, the first one I_1 (called the reference image) captured at time t and the second one I_2 (called the test image) captured at time $t+1$. The outputs of the algorithm are a motion-based segmentated image I_s and a list of motion parameters corresponding at each model in I_s. In I_s all the pixels belonging to each motion model are labeled using the same color.

For the sake of clarity, we describe the proposed algorithm in six steps:

(1) **Preliminaries**: In this step, I_2 is segmented using a given gray-level segmentation algorithm. The regions obtained are used as input for the algorithm. An adjacency graph of the previous segmentation is created. In addition, the spatial derivatives of the images I_1 and I_2 are estimated.

The purpose of the gray-level segmentation process is to classify the pixels into regions. Our motion segmentation algorithm requires that each segmented region should not have pixels belonging to more than one final motion model. Any gray-level segmentation algorithm that fulfills the previous constraint can be used. A sieve-based gray-level segmentation algorithm[12] has been used, because it produces a hierarchical representation of the image with different segmentations that differ in the region size. A segmentation with small regions must be used to fulfill the constraint.

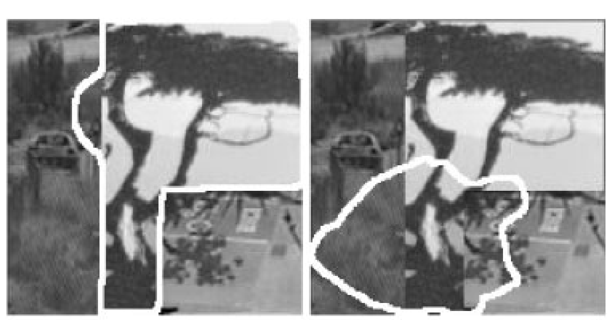
(2) **Get initial model**: The aim of this process is to find the best possible start point for the global motion segmentation and estimation algorithm. A good initial model is made up of a set of regions that have a high likelihood of belonging to the same model. The process starts selecting a region randomly. A model with this region and its neighbors is formed. The motion is estimated for this model using the process in subsection 2.2. A goodness measure *GM* is calculated for this model. The previous step is repeated q times. The model with the best goodness measure is selected as the initial model.

The goodness measure is calculated using the following expression:

$$GM = ((1 - l_{avg}) * 2 + (l_{best} - l_{worst})) \tag{1}$$

where l_{avg} is the average of the likelihood $L_{M_n}(R)$ for each region R using the motion model M_n (see subsection 2.1), l_{best} is the highest likelihood of the regions, and l_{worst} is the lowest likelihood of the regions. Therefore, the best initial model is the one that has the least *GM*.

Figure 1 shows an illustrative example of two possible initial models for a sequence with three different motion models: static (left part of the image) and two translational motions (the part of the image showing a tree and the bottom right part). The limits of two possible initial models are drawn with a continuous white line. Note that in Figure 1a, the majority of the pixels perform the same motion (the model of the three) and only a small area performs a different motion. Therefore, its *GM* will have a very small value. In addition, its *GM* will be lower than in the case of Figure 1b, where there is not a majority of pixels performing the same motion.



(a) (b)

Figure 1. Two examples of initial models.

(3) **Improve the model**: After an initial model has been obtained, an iterative classification process (see subsection 2.1) is started to find the inliers and to reject outliers between the k regions that make up the initial model. With the set of resulting regions, we start another classification process with the neighbors of the last inserted regions not yet processed. This classification step continues until there are no more new neighbor regions to be processed.
(4) **Exchange of regions**: If a valid model M_n has been extracted, then a region exchange procedure is started. The goal of this procedure is to reclassify regions that have been captured by an early model M_m where $m < n$. A region is moved if it lies closer to the new extracted model and there is a neighbor relationship between the region and the new model. If all the regions of the model M_m lie closer to the new model M_n then the model M_m is deleted. When for each region of model M_m we cannot decide whether it lies closer to the model M_m or to the model M_n, then the models are merged, that is, it is considered that both models have similar motion parameters.
(5) **Repeat**: Go to step 2 and repeat the same process with another initial model, if any. If there is any problem estimating the motion of some model, for instance, not enough texture information, not enough numbers of observations, and so on, the regions of this model are moved to a set called *regions with problems* (RWP).
(6) **End**: When all possible models have been extracted, the models that only have one region are tested in order to try to merge them with their neighbor models. In addition, each region in the RWP set is tested in order to move it into some of the models in its neighborhood.

At the end of the algorithm, a set of N_m motion models have been extracted. Each motion model is made up of a vector of motion parameters and a set of regions that support the motion.

2.1. Inliers/Outliers Region Classification

The aim of this process is to classify the regions of a model (according to its motion parameters) into two sets, inliers—regions that support the motion parameters—and outliers—regions that do not support them. The loop of this classification process consists of the following:

(1) Estimate the motion parameters using all the pixels belonging to the regions of the model (see subsection 2.2).
(2) Look for outliers in the regions of the model; if there are outliers, improve the motion parameters using only the remaining regions. A region R is considered an outlier (with respect to model M_n) if the likelihood of region R belonging to a model M_n is lower than a threshold.
(3) Test each outlier to see whether it can be now considered an inlier according to the new estimated parameters. If there are new inliers, the parameters are improved again. A region R is considered an inlier (with respect to model M_n) if the likelihood of the region R belonging to a model M_n is higher than a threshold.
(4) Go to step 2 and repeat until there are no changes in the set of regions of the model.

To estimate the likelihood of a region R belonging to a model M_n, the next expressions are used:

$$L_{M_n}(R) = \left(\sum_{p_i \in R} L_{M_n}(p_i)\right) \Big/ N_R$$

$$L_{M_n}(p_i) = e^{-0.5*[F^2_{M_n}(p_i)/\sigma_2]} \tag{2}$$

where N_R is the number of pixels of the region R. For each pixel p_i belonging to the region R the likelihood $L_{M_n}(p_i)$ of the pixel belonging to a model M_n is calculated. This likelihood[4] has been modeled as a Gaussian-like function where $F_{M_n}(p_i)$ is the residual for the pixel p_i of the objective function using the motion parametric vector of the model M_n. A region is considered as an inlier when this measure is higher than a threshold and it is considered as an outlier when its measure is lower than a threshold.

2.2. Motion Estimation

The Generalized Least Squares (GLS) algorithm is used to obtain the motion parameters of a model. In our approach the GLS estimator is used instead of an ordinary least squares estimation algorithm because GLS is able to reach more accurate estimates in the presence of outliers.[13,14]

The GLS estimation technique is based on minimizing an objective function O over a set S of r observation vectors, $S = \{L_1, \ldots, L_r\}$:

$$O = \sum_{L_i \in S} (F_i(\chi, L_i))^2 \tag{3}$$

where $\chi = (\chi^1, \ldots, \chi^p)$ is a vector of p parameters and L_i is an observation vector of n components $L_i = (L_i^1, \ldots, L_i^n)$, $i = 1 \ldots r$.

In general, the function F_i can be nonlinear. In the GLS method, instead of using linear functions F_i, the objective function O is linearized using the Taylor expansion and neglecting higher order terms.[10] This implies that an iterative solution has to be found. At each iteration, the algorithm estimates $\Delta\chi$ to update the parameters as follows: $\chi_{t+1} = \chi_t + \Delta\chi$. The increment $\Delta\chi$ is calculated (see Ref. 10) based on the partial derivatives of the function with respect to the parameters and the observations using the following expressions:

$$\Delta\chi = (A^T(BB^T)^{-1}A)^{-1}A^T(BB^T)^{-1}E \tag{4}$$

where

$$B = \begin{pmatrix} B_1 & 0 & 0 & 0 \\ 0 & B_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & B_r \end{pmatrix}_{(r\times(r\times n))} \qquad A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_r \end{pmatrix}_{(r\times p)} \qquad E = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_r \end{pmatrix}_{(r\times 1)}$$

$$B_i = \left(\frac{\partial F_i(\chi_t, L_i)}{\partial L_i^1}, \ldots, \frac{\partial F_i(\chi_t, L_i)}{\partial L_i^n} \right)_{(1\times n)}$$

$$A_i = \left(\frac{\partial F_i(\chi_t, L_i)}{\partial \chi^1}, \ldots, \frac{\partial F_i(\chi_t, L_i)}{\partial \chi^p} \right)_{(1\times p)}$$

$$\epsilon_i = -F_i(\chi_t, L_i) \tag{5}$$

In our motion estimation problem the GLS estimator allows us to directly use the Brightness Constancy Assumption (BCA); that is, the changes in gray levels between the reference image and the transformed one are only due to motion. Therefore, the objective function is expressed as follows:

$$O = \sum_{L_i \in S} (F_i(\chi, L_i))^2 = \sum_{L_i \in S} (I_1(x_i', y_i') - I_2(x_i, y_i))^2 \tag{6}$$

where $I_1(x_i', y_i')$ is the gray level of the reference image at the transformed point (x_i', y_i'), and $I_2(x_i, y_i)$ are the gray levels of the test image at point (x_i, y_i). S is related to the set of pixels that belong to the model.

The reference image (I_1) is used as given data and the test image (I_2) as observation data. Here, for each pixel i, the observation vector $L_i = (x_i, y_i, I_2(x_i, y_i))$ has three components $(n = 3)$: column, row, and gray level of test image at these coordinates. The gray level of the test image has been selected as a member of the observation vector because it is the observation that we want to match with the given gray level in the reference image using the BCA. The spatial coordinates have been also selected because their measurement can be inaccurate.

To fill the matrices A, B, and E, the partials of the function F_i with respect to the parameters and with respect to observation must be calculated. The partials of the functions $F_i = (I_1(x_i', y_i') - I_2(x_i, y_i))$ with respect to a parameter χ^j, $(j = 1 \ldots p)$ is calculated using the chain rule as follows:

$$\frac{\partial F_i}{\partial \chi_j} = \frac{\partial I_1(x_i', y_i')}{\partial x_i'} \frac{\partial x_i'}{\partial \chi^j} + \frac{\partial I_1(x_i', y_i')}{\partial y_i'} \frac{\partial y_i'}{\partial \chi^j} - \frac{\partial I_2(x_i, y_i)}{\partial \chi^j} \tag{7}$$

with

$$\frac{\partial I_1(x_i', y_i')}{\partial x_i'} = I_x^1(x_i', y_i')$$

$$\frac{\partial I_1(x_i', y_i')}{\partial y_i'} = I_y^1(x_i', y_i')$$

$$\frac{\partial I_2(x_i, y_i)}{\partial \chi^j} = 0 \tag{8}$$

where $I_x^1(x_i', y_i')$, $I_y^1(x_i', y_i')$ are the gradients of the reference image at the pixel (x_i', y_i') in the x and y directions. Therefore, Equation 7 can be simplified as follows:

$$\frac{\partial F_i}{\partial \chi^j} = I_x^1(x_i', y_i')\,\frac{\partial x_i'}{\partial \chi^j} + I_y^1(x_i', y_i')\,\frac{\partial y_i'}{\partial \chi^j} \tag{9}$$

where $\partial x_i'/\partial \chi^j$ and $\partial y_i'/\partial \chi^j$ must be calculated using a specific motion model.

On the other hand, the partial of the functions F_i with respect to a observation L_i^j, $(j = 1 \ldots n)$ is calculated using the chain rule as follows:

$$\frac{\partial F_i}{\partial L_i^j} = \frac{\partial I_1(x_i', y_i')}{\partial x_i'}\,\frac{\partial x_i'}{\partial L_i^j} + \frac{\partial I_1(x_i', y_i')}{\partial y_i'}\,\frac{\partial y_i'}{\partial L_i^j} - \frac{\partial I_2(x_i, y_i)}{\partial L_i^j} \tag{10}$$

with

$$\begin{aligned}
\frac{\partial I_1(x_i', y_i')}{\partial x_i'} &= I_x^1(x_i', y_i') \\
\frac{\partial I_1(x_i', y_i')}{\partial y_i'} &= I_y^1(x_i', y_i') \\
\frac{\partial I_2(x_i, y_i)}{\partial x_i} &= I_x^2(x_i, y_i) \\
\frac{\partial I_2(x_i, y_i)}{\partial y_i} &= I_y^2(x_i, y_i) \\
\frac{\partial I_2(x_i, y_i)}{\partial I_2(x_i, y_i)} &= 1.0
\end{aligned} \tag{11}$$

where $I_x^2(x_i, y_i)$, $I_y^2(x_i, y_i)$ are the gradients of the test image at the pixel (x_i, y_i) in the x and y directions. Therefore, Equation 10 can be simplified as follows:

$$\frac{\partial F_i}{\partial L_i^j} = I_x^1(x_i', y_i')\,\frac{\partial x_i'}{\partial L_i^j} + I_y^1(x_i', y_i')\,\frac{\partial y_i'}{\partial L_i^j} - \frac{\partial I_2(x_i, y_i)}{\partial L_i^j} \tag{12}$$

where $\partial x_i'/\partial L_i^j$ and $\partial y_i'/\partial L_i^j$ will be calculated using a specific motion model.

Using the affine motion model, which is able to cope with translations, scaling, rotation, and shear of images, the vector of parameters is $\chi = (a_1, b_1, c_1, a_2, b_2, c_2)$, $(p = 6)$. The transformed coordinates x_i', y_i' are related to the original ones x_i, y_i as follows:

$$\begin{aligned}
x_i' &= a_1 x_i + b_1 y_i + c_1 \\
y_i' &= a_2 x_i + b_2 y_i + c_2
\end{aligned} \tag{13}$$

Therefore, B_i, A_i, and ϵ_i are expressed as follows:

$$\begin{aligned}
B_i &= (a_1 I_x^1 + a_2 I_y^1 - I_x^2,\, b_1 I_x^1 + b_2 I_y^1 - I_y^2, -1.0)_{(1\times 3)} \\
A_i &= (x_i I_x^1, y_i I_x^1, I_x^1, x_i I_y^1, y_i I_y^1, I_y^1)_{(1\times 6)} \\
\epsilon_i &= -(I_1(x_i', y_i') - I_2(x_i, y_i))
\end{aligned} \tag{14}$$

where I_x^1, I_y^1, I_x^2, and I_y^2 have been introduced to simplify notation as $I_x^1 = I_x^1(x_i', y_i')$, $I_y^1 = I_y^1(x_i', y_i')$, $I_x^2 = I_x^2(x_i, y_i)$, and $I_y^2 = I_y^2(x_i, y_i)$.

Note that the GLS motion estimation framework can be easily extended to be used with more a complex motion model such as projective and quadratic motion models.

2.3. Refining Segmentation

Our motion segmentation approach requires that each region from the given gray-level segmentation should not have pixels belonging to more than one final motion model. A gray-level segmentation with a small region has been used to deal with this constraint. However, it is very likely that some regions will not fulfill this constraint. For problems requiring high accuracy in the segmentation of the motion, a refining process can be performed. The aim of this process is to refine the classification of the pixels without taking into account the initial classification in regions from the given gray-level segmentation. Now, we use the term *outlier* for a pixel that does not support the model and *inlier* for a pixel that supports the model.

The input of the refining process is the output of our algorithm, that is, a set of motion models, each one made up of a vector of motion parameters and a set of regions that support the motion. The refining process consists of the following:

(1) **Find Outliers**: For each extracted model M_n $(n = 1 \ldots N_m)$, find all the pixels that can be considered as outliers. They are the pixels p_i for which their likelihood with respect to the model M_n, $L_{M_n}(p_i)$ is less than a threshold. All the outlier pixels are included in a set, together with the pixels belonging to the region that have been considered outliers at the end of the original algorithm.
(2) **Improve parameters**: The motion parameters for the motion models that have new outliers are improved (see subsection 2.2).
(3) **Find Inliers**: For each outlier, test whether it can be included in some of the motion models. A pixel p_i will be included in the model with the greatest likelihood $L_{M_n}(p_i)$, $n = 1 \ldots N_m$, if it is bigger than a threshold and there is a neighborhood relationship between the pixel p_i and the model M_n. The pixel p_i is neighbor of the model M_n if any pixel into a window of 5×5 centered in p_i belongs to the model M_n.
(4) **Improve parameters**: The motion parameters for the motion models that have new inliers are improved (see subsection 2.2).
(5) **Repeat**: Repeat 1 to 4 while there are changes in the set of pixels.

At the end of the refining step the pixels have been classified into the different motion models corresponding to the moving objects in the scene. The pixels that could not be included in any model will be considered as outliers.

3. EXPERIMENTAL RESULTS

To show the performance of the approach presented, two types of experiments have been done. In the first experiment, synthetic sequences have been used, where the results of the motion segmentation and the motion parameters of each model are known. In this synthetic sequence three different motion models can be found. The first is the background, which does not perform motion, that is, it is static. The second motion model performs a change of scale, and the third corresponds to a rotational motion.

In the second experiment real scenes are used, where the final motion segmentation and the motion parameters of each model are unknown. The main motions of the real scene are the background produced by the camera motion, the motion of the car, and the motion of the wheels.

Figure 2 shows both images of the synthetic sequence, the initial gray segmentation used and the final segmentation obtained, where each final motion model has been labeled with a different RGB color. Figure 3 shows both images of the real sequence, the initial gray segmentation used and the final segmentation obtained. White pixels in Figures 2d and 3d are the ones that have not been classified in any model. These regions correspond mainly to regions belonging to occluded areas due to the motion and to regions that do not fulfill the requirement of belonging only to a model, that is, some pixels belong to a model and some others belong to a different model.

Figures 4 and 5 show the optic flow for both sequences. They have been computed using the motion parameters of each model in all the pixel belonging to them. They are presented to illustrate the motion models estimated.

To test the accuracy of the model, two measures have been calculated. P_{WS} is the percentage of pixels that have been well classified with respect to an ideal

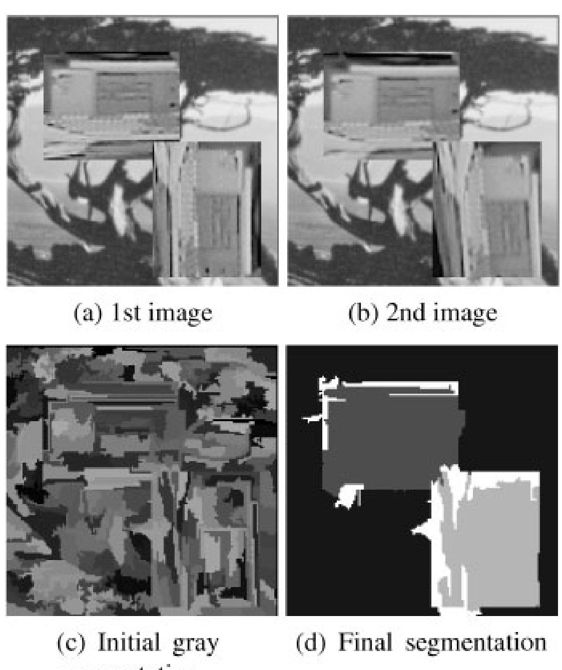


Figure 2. Both images of the synthetic sequence and results.

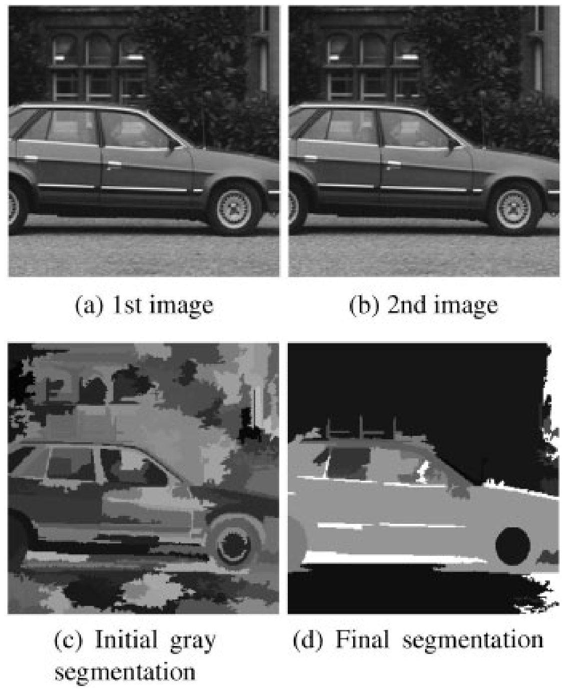


Figure 3. Both images of the real sequence and results.

segmented image. P_{WME} is the percentage of pixels in which the motion has been well estimated. For this purpose, the second image of the sequence is compared with a new image generated from the first image of the sequence using the motion parameters of each motion model found. So, P_{WME} is the percentage of pixels in which the difference of gray level in both images is less than a threshold, that is, the percentage of static pixels.

For the synthetic sequence, $P_{WS} = 91.5\%$ and $P_{WME} = 99.7\%$. The three motion models have been accurately segmentated and their corresponding motion parameters are also accurately estimated. The main difficulties in the synthetic sequence are the regions that have pixels belonging to more than one model and the regions in areas occluded due to the motion. They have been correctly classified as member of the outliers set.

For the real sequence, $P_{WME} = 88.1\%$. The main motions of this sequence have been segmentated; they are the background and the motion of the car. The main difficulties with the real scene are the motion of the wheels, because although our method has detected a rotational motion, it has less magnitude than in reality. Nevertheless, interesting results have been obtained in the windows, detecting the motion of the background and the motion of the driver. The outliers are also mainly detected in regions that have pixels belonging to more than one final model and in the regions in occluded areas.

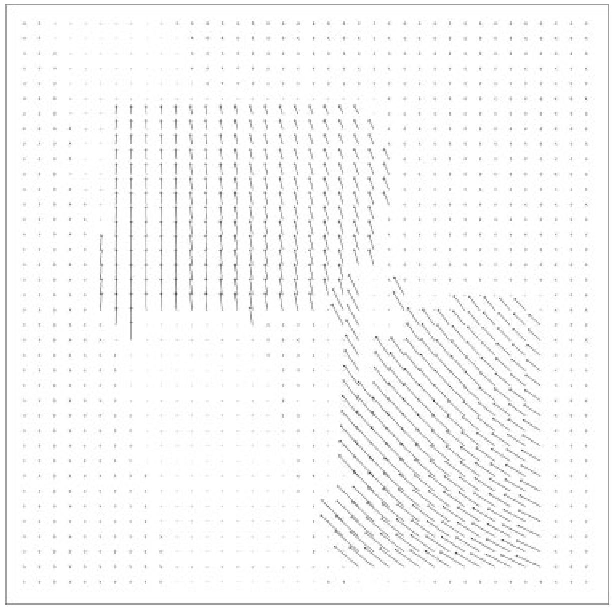


Figure 4. Optic flow computed from results of the synthetic sequence.

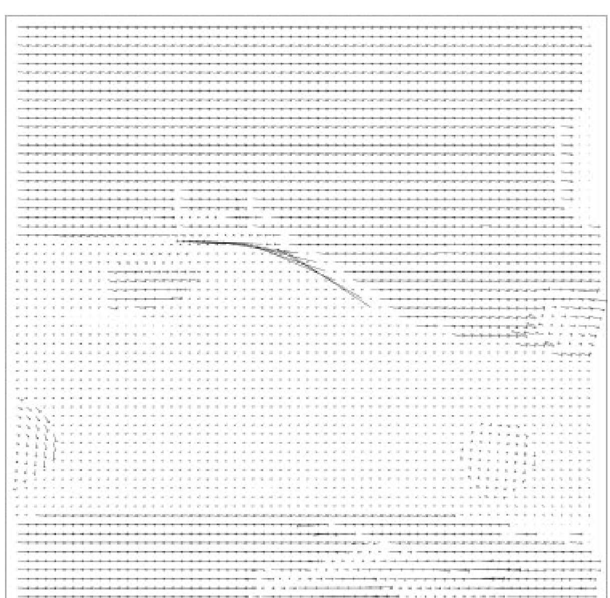
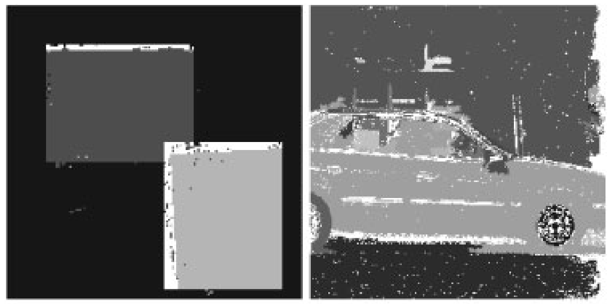


Figure 5. Optic flow computed from results of the real sequence.



(a) Synthetic (b) Real

Figure 6. Refined segmentation for test sequences.

Figure 6 shows the results obtained from the two sequences after the refining process. Note that segmentation has been improved in the motion boundaries. Now white pixels are the ones considered as outliers. They are mainly pixels belonging to occluded areas due to the motion and pixels where our algorithm could not estimate the motion due to lack of texture or to the presence of motions that were too large.

4. CONCLUSIONS

In this article, a motion segmentation and estimation algorithm has been presented that can extract different moving regions of the scene quasi-simultaneously and without a priori information about the number of moving objects. The main properties of our approach are:

- A GLS motion estimation algorithm is used, which produces accurate estimation of the motion parameters.
- The classification process, which collects inliers, rejects outliers, and exchanges regions among models, allows us to improve motion segmentation.
- It uses regions of pixels instead of pixels as observations and neighbor information, which improves the spatial consistency.
- After motion models have been obtained a refining process can be used to improve segmentation in regions from the initial gray-level segmentation that has pixels belonging to more than one final model.

Acknowledgments

This work has been partially supported by project CTIDIB/2002/333 from the Conselleria de Educació Cultura i Ciència, Generalitat Valenciana. The authors thank Alireza Bab-Hadiashar, Miroslaw Bober, Javier Ruiz Hidalgo and Gaudenz Danuser, who provided valuable explanations about their papers.

References

1. Irani M, Rousso B, Peleg S. Computing occluding and transparent motion. Int J Visual Comput 1994;12:5–16.
2. Odone F, Fusiello A, Trucco E. Robust motion segmentation for content-based video coding. In: RIAO 2000, 6th Conf on Content-Based Multimedia Information Access, College de France, Paris, 2000. pp 594–601.
3. Kim JB, Kim HJ. Efficient region-based motion segmentation for a video monitoring system. Pattern Recogn Lett 2003;24:113–128.
4. Bober M, Kittler JV. Estimation of complex multimodal motion: An approach based on robust statistics and Hough transform. Image Vision Comput 1994;12:661–668.
5. Badenas J, Sanchiz JM, Pla F. Motion-based segmentation and region tracking in image sequences. Pattern Recogn 2001;34:661–670.
6. Kälviäinen H, Oja E, Xu L. Motion detection using randomized Hough transform. In: Proc 7th Scandinavian Conference on Image Analysis, Aalborg, Denmark, 1991. pp 72–79.
7. Lee K-M, Meer P, Park R-H. Robust adaptive segmentation of range images. IEEE Trans Pattern Anal Mach Intell 1998;20:200–205.
8. Miller JV, Stewart CV. MUSE: Robust surface fitting using unbiased scale estimates. In: IEEE Computer Vision and Pattern Recognition, June 18–20, 1996. pp 300–306.
9. Bad-Hadiashar A, Suter D. Robust motion segmentation using rank ordering estimators. In: Proc 3rd Asian Conference on Computer Vision ACCV98, Hong Kong, 1998. pp 599–606.
10. Danuser G, Stricker M. Parametric model-fitting: From Inlier characterization to outlier detection. IEEE Trans Pattern Anal Mach Intell 1998;20:263–280.
11. Montoliu R, Pla F. Multiple parametric motion model estimation and segmentation. In: ICIP01, 2001 Int Conf on Imaging Processing, October, 2001, vol. II. pp 933–936.
12. Bangham JA, Ruiz Hidalgo J, Harvey R, Cawley G. The segmentation of images via scale-space trees. In: Carter JN, Nixon NS, editors. Proc British Machine Vision Conference, Southampton, UK, September 1998. pp 33–43.
13. Montoliu R, Pla F. Comparing brightness constancy assumption and optic flow equation in motion estimation algorithms. In: Proc 2003 IASTED Int Conf on Visualization, Imaging, and Image Processing (VIIP 2003), September 2003. pp 90–95.
14. Montoliu R, Pla F. Robust techniques in least squares-based motion estimation problems. In: Progress in Pattern Recognition, Speech and Image Analysis. Lecture Notes in Computer Science 2905. Berlin: Springer Verlag; 2003. pp 62–70.