Quasi-simultaneous Motion Segmentation and Estimation using a Generalized Least Square method.

R. Montoliu , F. Pla
Computer Vision Group (www.vision.uji.es)
Universitat Jaume I
Campus Riu Sec, 12071 Castellón (SPAIN)
{montoliu, pla}@uji.es

Abstract – **This paper presents a quasi-simultaneous motion segmentation and estimation method based on a parametric model fitting algorithm. 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 a sequence of two consecutive images. The classification of the pixels is based on a measure of their likelihood under each model.**

Experiments using synthetic sequences are shown in order to illustrate the results obtained using the proposed algorithm. In addition, preliminary results using real sequences are also shown.

I. Introduction

Detection and segmentation of moving objects in image sequences have received a large attention in the literature of image sequence analysis. Some of the applications where motion estimation/segmentation techniques can be applied are image registration, moving object detection, surveillance, guidance of autonomous vehicles, efficient video compression, traffic monitoring and tracking.

Most of the early work has been concentrated on the estimation of the optical flow computed between image pairs, [1], [2]. These techniques can be used to detect multiple motions, however, they are very sensitive to the quality of the optical flow, that it is usually very inaccurate in the motion boundaries.

Another approach in motion estimation/segmentation is the use of parametric motion estimators, which describe the motion over a larger spatial region in terms of a parametric model [3]. These methods ([3], [4], [5]) are usually quite accurate since they have to estimate a small number of parameters (6 for the affine motion model) under a large set of constraints, one constraint for each pixel used to estimate the parameters in a region. The problem with the model-based method is that large image regions are often not well modeled by a single parametric model due to the complexity of the motion or the presence of multiple moving objects. When the region of analysis contains multiple moving objects it is would be more appropriate to have a method that can recover simultaneously the motion parameters and the points (pixels) that support these parameters.

In [6], Bober and Kittler presented a Hough transform based hierarchical algorithm for motion estimation and segmentation using the affine motion model. They used robust estimation techniques in order to reduce the influence of the outliers. The Bober and Kittler algorithm estimates first the parameters of the dominant motion present in the entire image. The pixels that support that model are removed, then the algorithm is applied recursively to the remaining pixels until all the motion models present have been extracted. The pixels are then classified depending on their likelihood under each model. The main drawback of these "dominant motion" algorithms ([6], [5]) is that usually the dominant motion is a mean of some of the different motions present in the sequence and therefore the motion parameters of the dominant motion extracted do not exactly correspond to any of the motions in the sequence. In addition the M-estimator works fine when the number of outliers are small, usually less than 50% of the total number of pixel and they do not deviate significantly from the true estimate [7], but in real world like traffic scenes there are object that do not hold this constraints.

Danuser and Stricker [8] presented a framework for parametric model fitting applied in lines and planes fitting problems. They use a quasi-simultaneous application of a generalized least squares algorithm to extract all the models (lines or planes). Their main idea for extracting multiple models, is that for one model the other models can be considered as outlier. For that, they use simultaneously with the parameter estimation, two distance measures that classify the pixels into supporting points to a model and no-supporting points to this model.

In this paper a generalized least squares fitting technique join as a pixel classification measure based in the likelihood of each pixel under a model, is applied in order to improve an coarse initial motion segmentation

Research supported in part by projects GV97-TI-05-27 from the *Conselleria d'Educaci'o, Cultura i Ci'encia, Generalitat Valenciana*, and CICYT TIC98-0677-C02-01 from the Spanish *Ministerio de Educaci'on y Cultura*

and to estimate accurately the affine motion parameters in a sequence of two consecutive frames.

The next section explains the motion estimation technique used for fitting the motion parameters of each model. The measurement used in the classification process is explained in Section 3. The scheme of the motion segmentation algorithm is shown in Section 4. In the last section a set of experiments are shown to illustrate the performance of this approach.

II. Motion Estimation

A generalized least squares-based method is used to estimate the parameters of each model. A model consists of a set of pixels that follow a vector of motion parameters. For each pixel p_i belonging to the model an observation vector l_i is defined as follows: $l_i = (x_i, y_i, I_2(x_i, y_i))$, where x_i and y_i are the coordinates of pixel p_i and $I_2(x_i, y_i)$ is the grey level of this pixel in the second image of the sequence.

This method fits a set of functions f_i, each one associated to an observation vector l_i. All functions f_i have a common vector of unknown parameters χ

$$f_i(\chi, l_i) = I_1(x_i', y_i') - I_2(x_i, y_i) = 0, i : 1..r \qquad (1)$$

where r is the number of pixels of the model, χ is a vector of 6 model parameters $(a_1, b_1, c_1, a_2, b_2, c_2)$ that correspond to the affine motion model, and:

$$\begin{aligned} x_i' &= a_1 x_i + b_1 y_i + c_1 \\ y_i' &= a_2 x_i + b_2 y_i + c_2 \end{aligned} \qquad (2)$$

The estimation process is an iterative process which starts with an initial guess of the model parameters $\hat{\chi}_0$. The estimation process is stopped when the increment $\widehat{\Delta\chi_t} = \hat{\chi}_{t+1} - \hat{\chi}_t$ of the estimation $\hat{\chi}$ at the iteration t is smaller than a threshold. $\widehat{\Delta\chi_t}$ is calculated at each iteration as follows:

$$\begin{aligned} \widehat{\Delta\chi_t} &= \left(A^T Q_{ww}^{-1} A\right)^{-1} A^T Q_{ww}^{-1} w \\ Q_{ww} &= \left(BB^T\right) \qquad w = (-F(\chi, L))_{(r\times 1)} \\ A &= \left(\frac{\partial F}{\partial \chi}\right)_{(r\times 6)} \qquad B = \left(\frac{\partial F}{\partial L}\right)_{(r\times(r\times 3))} \end{aligned} \qquad (3)$$

Matrices A, B and vector w are calculated using χ_t. See ([9], [8], [10], [7]) for more details.

III. Pixel Classification

Least-squares estimation assumes that the noise corrupting the data is of zero mean and implicitly assumes that the entire set of data can be explained by only one parameter vector. As it is well known, least-squares estimators are vulnerable to the violation of these assumptions. Therefore a test for extracting noisy points, called outliers, is needed.

On the other hand, classical regression algorithms to extract outliers are very sensitive of initialization. They fit, extract outliers based on residuals, and repeat the same loop until some stop condition is reached. In our approach we add a step for collecting new supporting points. These points can be outliers to other models or points that had been classified into a wrong model, due mainly to the initialization.

In our approach, the outliers and inliers classification is based on a measure of the likelihood of a pixel of belonging to a model [6]. It is defined as:

$$l_k(p) = e^{-0.5\frac{R_k^2(x,y)}{\sigma_k^2}} \qquad (4)$$

where $R_k(x, y)$ is the residual of the function f_i (see 1) of the pixel (x,y) using the motion parameters of the model k. This measure can determine if a pixel belongs to a model and can be used to calculate if a pixel must be moved from one model to other.

IV. Motion Segmentation

The motion estimation algorithm and the classification method shown in the two previous sections are used in order to improve an initial motion segmentation.

The scheme of our approach is shown in Figure 1. The motion segmentation algorithm starts with a coarse segmentation of the image. This segmentation is used as starting point to the algorithm. For each initial model the initial vector of model parameters is set to no-motion, i.e $a_1 = 1$, $b_1 = 0$, $c_1 = 0$, $a_2 = 0$, $b_2 = 1$ and $c_2 = 0$. The first step is to fit all initial models in order to obtain the estimated motion parameters. The next step is a search for outliers using the likelihood shown in the previous section. For each model the outliers are extracted using their motion parameters. After outlier extraction a new fit is needed to re-estimate the motion parameters.

The next step is a search for inliers, this step is needed because some pixels that were previously considered outliers to model k may now be considered as inliers given the new parameters. For each outlier p_i, $l_k(p_i)$ (see 4) is calculated using the motion parameters of model k; the pixel is then classified in the model with the highest probability value. After this step a new fit is performed.

The last step is an interchange of pixels between models. In this step all pixels of a model m_i are tested for classification in model m_j $(m_i \neq m_j)$ and vice versa. This step adjusts the possible deficiencies in the initial segmentation. If there is any change a new iteration starts, otherwise the algorithm stops.

The final output is a list of models, each one with a list of supporting pixels and a vector of affine motion parameters. In Addition a list of pixels that could not be classified to any model is also obtained.

V. Experiments and Results

In this section, experiments with synthetic and real sequences are shown in order to test the proposed approach.

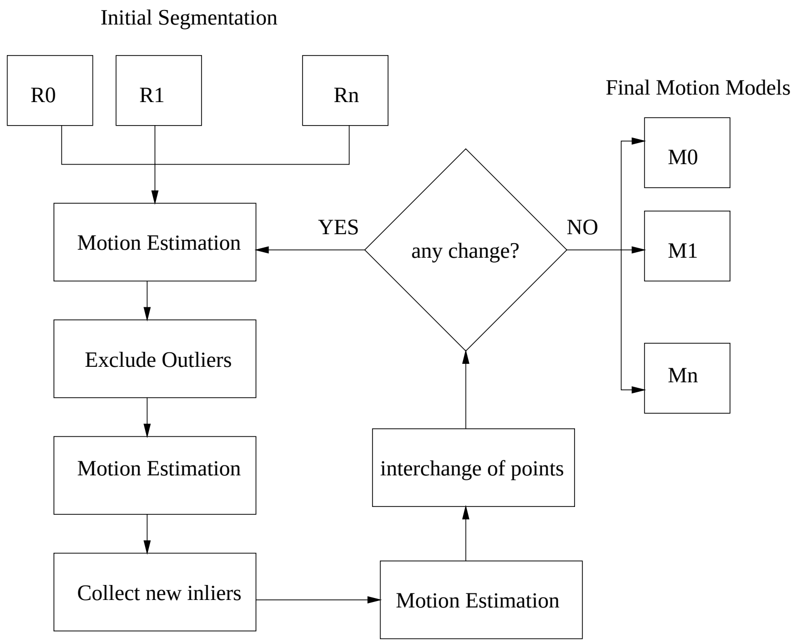


Figure 1 - The motion segmentation scheme

Grey level	50	100
a_1	0.9999	1.0000
b_1	0.0000	0.0000
c_1	1.4995	-0.0005
a_2	0.0000	0.0000
b_2	1.0000	0.9999
c_2	-0.0001	-0.7993
$NPoints$	9793	12211

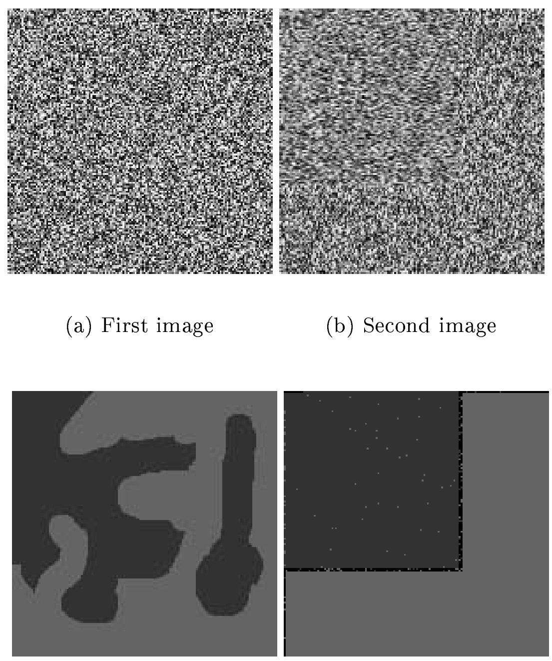
Table I

Motion parameters estimated for sequence 2m-rand

A. Synthetic sequences

Two experiments using synthetic sequences are shown in order to test the proposed algorithm. In the first experiment (labeled as *2m-rand*) a random generate image is tested. The sequence has two translational movements, the first one is a translation of 1.5 pixel in x, and the second it is a translation of -0.8 pixel in y. In this case the initial segmentation is very bad (see Sub-Figure 2c). Our algorithm can improve this bad segmentation , thanks to the interchange step, and estimates the motion parameters with high accuracy, see Figure 2 and Table I. The real motion parameters used to generate this sequence were: $\hat{p}_{50} = (1, 0, 1.5, 0, 1, 0)$ and $\hat{p}_{100}(1, 0, 1, 0, 1, -0.8)$

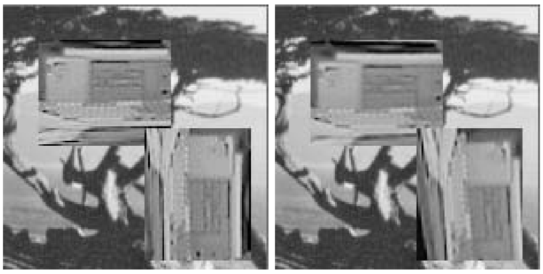
In the second synthetic experiment a sequence(labeled as *3m-scale-rot*) with more complex movement is tested. This sequence has three motion model, the first performs a rotational movement of 3 degrees, the second has no-motion and the third performs a change of scale of 0.98 in x and 0.95 in y. In Figure 3 the two images of the sequence and the initial segmentation and final segmentation are shown. Table II shows the motion parameters estimated. The algorithm segmentate the rotational and change of scale movements with high accuracy, even in motion discontinuities. The motion parameters are estimated with a high accuracy. There are some pixels that have been classified into model 1 (grey level 100) but should have been classified into model 0 (grey level 50) and 2 (grey level 150). This pixels are in areas of the image with a low texture



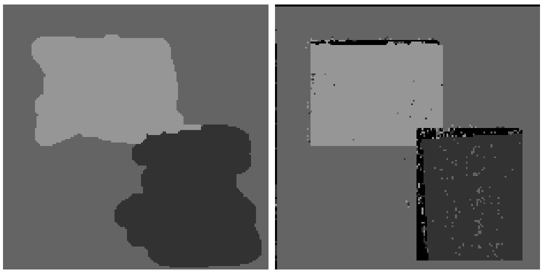
(a) First image (b) Second image

(c) Initial Segmentation (d) Final Segmentation

Figure 2 - 2m-ran sequence and results



(a) First image (b) Second image



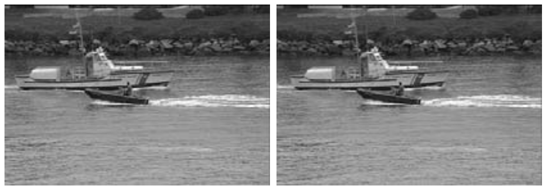
(c) Initial Segmentation (d) Final Segmentation

Figure 3 - 3m-scale-rot sequence and results

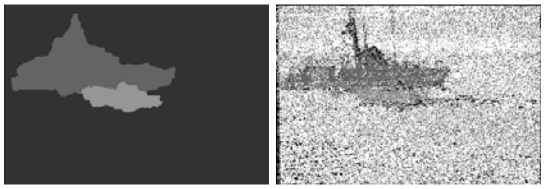
Grey level	50	100	150
a_1	0.9985	1.0000	0.9800
b_1	-0.0523	0.0000	0.0000
c_1	1.3136	0.0000	0.5993
a_2	0.0522	0.0000	0.0000
b_2	0.9986	1.0000	0.9549
c_2	-9.8598	-0.0002	-1.1500
$NPoints$	3864	13527	4225

Table II

Motion parameters estimated for sequence 3m-scale-rot



(a) First image (b) Second image



(c) Initial Segmentation (d) Final Segmentation

Figure 4 - Coast sequence and results

Grey level	255	100	150
a_1	0.9999	1.0002	1.0028
b_1	0.0025	-0.0087	-0.0188
c_1	-2.2524	-3.2215	1.5790
a_2	0.0000	0.0000	-0.0031
b_2	0.9994	1.0008	1.0134
c_2	-0.0191	-0.1352	-1.2045
$NPoints$	48511	19392	12811

Table III

Motion parameters estimated for coast sequence

information, and it is quasi-impossible without using space or neighbourhood information classify them into the correct model.

B. Real sequences

Preliminary experiments have been tested using real sequences. In Figure 4 the real *Coast* sequence is shown. This scene have two clear motion model, one the big ship and the second the small ship. The third motion is due to a camera movement. The Table III shows the motion parameters estimated for the three motion models. In this example the motion of two ship has been well segmentated, that can be shown in the boundaries of both ships. The third motion (white color) has problems due to the lack of texture information and the irregular motion of the water. Therefore there are pixels not correctly classified into the two boats motion model. The pixels like the water wake of the small boat do not correspond to any of three motion models, and therefore it is well classify as outliers (black points)

VI. Conclusion

A motion segmentation and estimation algorithm has been presented. It can improve a coarse initial segmentation to a high accurate segmentation, including motion boundaries, while estimates the accurate affine motion parameters. The algorithm includes a interchange of pixels between models for reduce the influence of the initial segmentation.

Future work must focus on using spatial constraints, like considering a neighborhood of pixels as a point to the model, in order to improve the segmentation in areas without texture information. In addition, hierarchical techniques could be used to accelerate the computing time.

References

[1] B.K.P. Horn and B.G. Schunck, "Determinig optical flow", in *MIT AI*, 1980, A.I. Memo No 572.

[2] J. Barron, D. Fleet, and S. Beauchemin, "Permormance of optical flow techniques", *Int. J. Computer Vision.*, vol. 12, pp. 43–77, 1994.

[3] S. Ayer, P. Schroeter, and J. Bigun, "Segmentation of moving objects by robust motion parameter estimation over multiple frames", in *ECCV94*, 1994, pp. B:316–327.

[4] S. Periaswamy, J. B. Weaver, D. M. Healy, D. N. Rockmore, Peter J. Kostelec, and Hany Farid, "Differential affine motion estimation for medical image registration", in *SPIE's 45th annual Meeting, The international Symposium on Optical Science and Technology*, 2000.

[5] M. Irani, B. Rousso, and S. Peleg, "Computing occluding and transparent motion", *IJVC*, vol. 12, no. 1, pp. 5–16, February 1994.

[6] M. Bober and J. Kittler, "A hough transform based hierarchical algorithm for motion segmentation and estimation", in *4th International workshop on Time-Varying image Processing and Moving Object Recognition*, V. Cappellini, Ed., 1993, pp. 335–342.

[7] Z. Zhang, "Parameter-estimation techniques: A tutorial with application to conic fitting", *Image and Vision Computing*, vol. 15, no. 1, pp. 59–76, 1997.

[8] G. Danuser and M. Stricker, "Parametric model-fitting: From inlier characterization to outlier detection", *PAMI*, vol. 20, no. 3, pp. 263–280, March 1998.

[9] F. Pla R. Montoliu, "Parametric motion model extration and estimation", in *Proccedings of the IX Spanish Symposium on Artificial Inteligence*, 2001, pp. 725–734.

[10] M. Trobina, "Error model of a coded-light range sensor", Tech. Rep., Communication Technology Lab, Image Science Group, ETH, September 1995.