ROBUST 3-D MOTION ESTIMATION AND DEPTH LAYERING

N. Komodakis and G. Tziritas
Institute of Computer Science - FORTH, and,
Department of Computer Science, University of Crete
P.O. Box 1470, Heraklion, Greece
E-mails: komod@csd.uch.gr, tziritas@csi.forth.gr

Abstract: Two problems are addressed in this paper: camera 3-D motion estimation and image depth layering. For the first problem, the method we adopt is the use of an hierarchy of motion models in combination with robust estimation methods. A dense motion vector field is assumed to be granted. For the second, an iterative deterministic relaxation algorithm is used and the extraction of the image depth layers is based on both optical flow and luminance information.

1 Introduction

One of the major areas in computer vision research is 3-D motion analysis. Two general methodologies exist for achieving 3-D motion estimation based on image sequences. According to the first of these approaches, the 3-D motion parameters are obtained through calculations based on a previously estimated 2-D apparent motion vector field [1] [7]. The second approach tries to evaluate these 3-D motion parameters directly through the use of the spatio-temporal derivatives of the intensity function [6].

The method we adopt in this paper is the former. This method is based on a scheme consisting of two stages. During the first stage a dense 2-D motion vector field is computed. During the second stage, the 3-D motion parameters are identified by equations linking the projected 2-D motions and 3-D motions inside the image sequence. We will concentrate on the second stage. In this paper, when we refer to 3-D motion estimation, we mean dominant motion estimation (or camera motion estimation).

One main problem for the right estimation of the parameters of the camera motion, is the fact that the optical flow field may contain a set of noisy and partially incorrect data (outliers) [3]. The set of incorrect data can be even larger, if we consider the case of the existence of independent motions throughout the image sequence. The negative effects of this set of outliers to the motion estimation, increase with the complexity of the motion model which is used to describe the camera motion. Therefore, the approach we adopt is the use of an hierarchy of motion models. This hierarchy of models can be structured in a tree form. A criterion is defined to reject or accept models. Simpler models are first tested, and then more complex models are considered. This way we hope to find the simplest possible model (the one with the fewest parameters), which can adequately describe the camera motion. At each stage of the algorithm, a robust estimation method is used to cope with the set of outliers.

After the estimation of camera motion the problem of depth layering is considered (Section 4), and experimental results are given in Section 5.

2 Robust estimation of 3-D motion

In this paper the optical flow motion analysis concerns with the perspective projection of 3-D rigid body motions onto a 2-D image domain. Without any loss of generality the focal length is assumed to be known, and for simplification equal to the length unit. The 2-D motion vector (u, v) at an image point (x, y) can be expressed using the instantaneous 3-D translation vector (T_X, T_Y, T_Z) and the instantaneous 3-D rotation vector $(\Omega_X, \Omega_Y, \Omega_Z)$ of the kinematic screw associated with the moving object [9]

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{T_X - xT_Z}{Z} - \Omega_X xy + \Omega_Y(1+x^2) - \Omega_Z y \\ \frac{T_Y - yT_Z}{Z} - \Omega_X(1+y^2) + \Omega_Y xy + \Omega_Z x \end{bmatrix} \quad (1)$$

Using equations (1) to throw the depth Z away, this leads to the following equation:

$$\begin{aligned} &\Omega_X + \beta\Omega_Y - (\Omega_X + \alpha\Omega_Z)x - (\Omega_Y + \beta\Omega_Z)y + \\ &(\beta\Omega_Y + \Omega_Z)x^2 + (\alpha\Omega_X + \Omega_Z)y^2 - \\ &(\alpha\Omega_Y + \beta\Omega_X)xy + \alpha v - \beta u = xv - yu \end{aligned} \quad (2)$$

where $\alpha = T_X/T_Z$ and $\beta = T_Y/T_Z$, assuming $T_Z \neq 0$. If we have managed to find the eight essential motion parameters of the above linear equation, then the rotation and the translation, within a scale factor, could be obtained. Therefore, the motion vectors of at least eight points are needed. However, in practice the observed optical flow vectors may be corrupted by noise. In addition to that, it is quite possible that there are additional 3-D motions in the image, apart from the motion of the camera. Consequently, a robust estimation method must be used to determine the camera's 3-D motion parameters.

Robust methods provide tools for statistics problems in which underlying assumptions are inexact [5]. A robust procedure should be insensitive to departures from underlying assumptions caused by outliers. That is, it should have a good performance under the underlying assumptions and the performance deteriorates gracefully as the situation departs from the assumptions. An important characteristic of robust estimators is their breakdown point, which may be defined as the smallest amount of outlier contamination that may force the value of the estimate

outside an arbitrary range. There are several types of robust estimators: M-estimator, LMedS-estimator and others. We are going to be concerned with the M-estimator.

The M-estimation problem could be expressed as follows: given a set of data samples d_i and x_i, where $d_i = f(x_i, \theta) + e_i$, estimate the vector of parameters θ under noise e_i. The only underlying assumption is that the noise obeys a symmetric, independent, identical distribution. The M-estimate $\hat{\theta}$ is defined as the minimum of a global error function:

$$\hat{\theta} = \arg\ \min \sum_i g(d_i - f(x_i, \theta)) \tag{3}$$

A data weighting function could be obtained from the error penalty function g as follows $h(e) = \frac{g'(e)}{2e}$. In the Least Squares regression, all data points are weighted equally with $h(e) = 1$. In robust M-estimation, the function h provides adaptive weighting. In this paper we use the Tukey's M-estimator (or biweight estimator) with the following weighting function

$$h_c(e) = \begin{cases} (1 - (\frac{e}{c})^2)^2 & |e| \leq c \\ 0 & |e| > c \end{cases} \tag{4}$$

The c parameter in the above function is a scale parameter, which plays a crucial role in the success of the M-estimator. For the Tukey's biweight estimator, the parameter c can be chosen to be: $c = c_0 \text{median}(|e_i|)$, where c_0 is a normalizing constant in the range between 6 and 9.

3 Hierarchy of 3-D motion models

The M-estimator has the theoretical breakpoint of $1/(p+1)$, where p is the number of unknown parameters to be estimated. What this implies is, that the more the unknown parameter are, the less robust the M-estimator becomes. Therefore, if we succeed in reducing the number of parameters to be found, then we can expect better results. Therefore, it would seem interesting to try to find that 3-D model, which can describe the observed motion adequately, and with as few parameters as possible. Apart from avoiding large estimation bias, this will also lead to a major speed-up of the M-estimator. On the other hand, if a too simple 3-D motion model is chosen for an image in which the physically observed motions are complex, this will naturally lead to a poor motion estimation. The approach we adopt in this paper is to use an hierarchy of motion models [7]. This hierarchy of motion models can be structured in a tree form. Once this model hierarchy has been determined, one needs to define a criterion, which will decide which one among the available motion models will be chosen to describe the camera's motion. The criterion could be the minimization of the mean squared displaced frame difference associated with the given model, with a possible additional penalty term for complex models.

Two possible methodologies for the effective use of this hierarchy of motion models are the following:

- all motion models are tested in parallel and the best one is chosen
- the motion models are examined sequentially in a pre-defined order, one after another. The order of the models will correspond to a traversal of the hierarchy tree. Only if the current model fails to describe the 3-D motion, we examine the next one in the path. The simplest models are examined first, and then progressively more complex models are considered, hoping this way that the simplest model that matches the camera motion will be chosen.

The methodology we adopted in this paper was the latter. Models of approximately the same complexity are placed at the same level of the hierarchy, while deeper levels in the tree contain models that can describe more complex motions.

In the general case, the camera's motion can contain six degrees of freedom (3-D translation and 3-D rotation). However in most cases and considering certain assumptions regarding the relative distancing of the various objects present in the scene in relation to the small rotation angles of the camera, one can categorize the camera motions to three basic classes [4] [7]

- motions which contain only translation parallel to the image plane (panning)
- motions which contain only a change to the focal length (zooming) or tranlation along the optical axis
- and motions which contain only rotation about only one axis

With these observations in mind, the hierarchy of models given in Figure 1 was created.

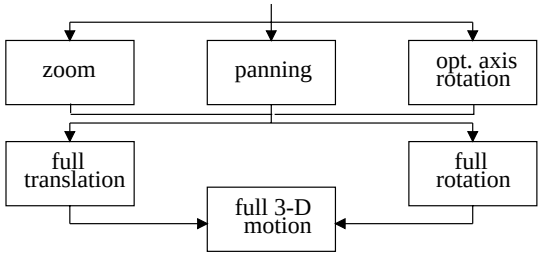

Figure 1. Motion models hierarchy

For all motion models, apart from the estimation of the camera motion, the points of the image which are consistent with that motion (motion inliers) are detected. For the simple models of the first level, only acception/rejection tests are performed without the estimation of motion parameters. This could also be useful in image analysis applications where the type of motion and not the exact motion estimation is wanted. Also, it must be mentioned that the above hierarchy could be enriched with even more motion models. In the following paragraphs, each motion model in the tree of the figure above will be described in more detail.

3.1 Panning

In this case only one translation component parallel to the image plane exists (say T_X). Using equations (1) we conclude that $u = T_X/Z$ and $v = 0$. Since the depth

Z is unknown, T_X cannot be computed. We can only confirm or not the existence of this kind of 3-D motion. Let us call t the angle between the optical flow vector and the horizontal axis. The test could be designed on the measures of this angle. Therefore, to determine the image points that are consistent with the camera motion we only need to minimize the following quantity with respect to t, using a M-estimator technique:

$$\sum_i g(t - \arctan(\frac{v_i}{u_i}))$$

where the sum is taken over all image points. If the solution that will be found is close to zero and the number of inliers is greater than the half of the image points, then this motion model is accepted, else it is rejected.

3.2 Zooming

In this case the rotation vector is zero, while the translation vector contains only the T_Z component. As in the case of panning, we cannot estimate T_Z exactly, but only check if zooming exists or not, and also find the motion inliers. From equations (2) we see that for any image point it is true that: $xv - yu = 0$, where the image coordinates are relative to the center of the image. This means that the motion field converges to the image center. Therefore to find the inliers of the camera motion, we will have to minimize (with respect to t) the following sum:

$$\sum_i g(t - \frac{x_i v_i - y_i u_i}{\sqrt{x^2 + y^2}\sqrt{u^2 + v^2}})$$

3.3 Rotation around optical axis

In this case the translation vector is zero, while the rotation vector contains only one component (Ω_Z). From equations (2) we see that for any image point $xu + yv = 0$. Therefore this case is similar to the previous one.

3.4 Full translation

In this case equation (2) could be written as follows ($T_Z \neq 0$)

$$\alpha v - \beta u = xv - yu$$

Therefore, we can estimate the ratios T_X/T_Z and T_Y/T_Z using a robust regression algorithm, like Tukey's biweight estimator.

3.5 Full rotation

Because the depth is not involved in the expression of the 2-D motion vector, the three components of the rotation can be robustly estimated directly from the two equations in (1).

3.6 Full 3-D motion

This is the most general case. In this case equation (2) should be used with eight unknown parameters in a linear form, or five parameters in nonlinear form.

4 Depth layering

After the camera motion has been estimated and the part of the image that is consistent with the camera motion has been determined, we can try to extract the depth layers for that part of the image [10] [8]. Obviously the extraction of the depth layers is not possible in the case of pure rotational motion. Therefore, only the case of pure translational motion will be considered here. The case of full 3-D motion could be reduced to the case of pure translational motion.

Let us consider only one component of the 2-D motion field, say u. It can be seen from equation (1) that $u/(\alpha - x)$ is constant at a constant depth. In particular in the case of panning, u should be constant, if the depth were constant. Therefore, for this particular case, to find the depth layers inside the image, we only need to segment the optical flow field into regions of constant value. However, optical flow field estimation usually contains errors due to occlusions, noise, etc. Therefore, apart from optical flow we will also make use of luminance information [8]. The set of labels that the segmentation algorithm will use, will consist of labels contributed both by optical flow and luminance. Although this makes the algorithm more robust, it will also have as a result that not all the regions generated, will correspond to different depth layers. For this purpose, redundant regions will be eliminated at a later stage.

The algorithm for the depth layering consists of three stages: 1) motion and luminance data clustering, 2) combined motion and luminance segmentation, and 3) region merging.

4.1 Stage 1: Motion and luminance data clustering

Let us call L_0, L_1 the labels due to optical flow and luminance respectively. The set of labels given as an input to the segmentation algorithm will be the cartesian product of the above two sets. Therefore, the first problem to be addressed is that of determining labels L_0, L_1 and their corresponding energy functions.

- **Determining L_0, the set of labels due to optical flow**. We assume that the first component u of the optical flow field can be decomposed into a mixture of n Gaussian distributions. The number n and the parameters of the Gaussian distributions are determined by the analysis of the optical flow histogram. The energy function for a label i determined by parameters (μ_i, σ_i^2) is

$$f_i(u) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{(u-\mu_i)^2}{2\sigma_i^2}}$$

- **Determining L1, the set of labels due to luminance**. In this case, in order to reduce the computational cost, the quantization of the image intensity is performed. For this purpose a Lloyd-Max quantizer was used. As in the case of the mixture of Gaussian distributions, the number of quantization levels is determined empirically. The energy function for label j represented by q_j is: $f_j(x) = |x - q_j|$.

4.2 Stage 2: Combined motion and luminance segmentation

A global segmentation criterion will be used based on a Markov random field model. For the minimization of the resulting energy function an iterative deterministic

relaxation algorithm is used, known as the *Highest Confidence First (HCF)* algorithm [2]. The HCF algorithm is completely defined, if the number of labels and an energy function for each label are given. Each label will correspond to a different depth layer combined with luminance information. The energy function corresponding to the label i for motion and j for luminance will be the sum of the corresponding energy functions. The connected components of the regions that HCF generated are passed as input to the next stages of the algorithm.

4.3 Stages 3: Region merging

To eliminate the redundant regions from the previous stage of the algorithm, region merging must be introduced. The merging of regions will be based solely on optical flow information. Therefore a metric distance between two regions should be defined. In this work the absolute difference of the mean values of the velocity of two regions is defined to be their distance. At each iteration, we merge that pair of regions with the minimum distance. The merging continues as long as that minimum distance is below a certain threshold, otherwise the algorithm stops. All possible region pairs are sorted into a stack. Pairs with a small distance are placed at the top of the stack, while pairs with a big distance are placed at the bottom of the stack. At each iteration the pair on top is removed. In order to reduce the computational cost, for regions with a very small size, only adjacent regions are examined for merging.

5 Experimental results

The algorithm for the camera motion estimation has been tested on the *Stefan* sequence. The only independent motion inside the image is that of the tennis player. The camera motion is translational along the X and Z axes. From the set of motion models in the hierarchy tree, those of simple panning, zooming and rotation around optical axis were rejected. The full 3-D translation motion model was selected. The inliers to the camera motion according to this model, are displayed in Figure 2.

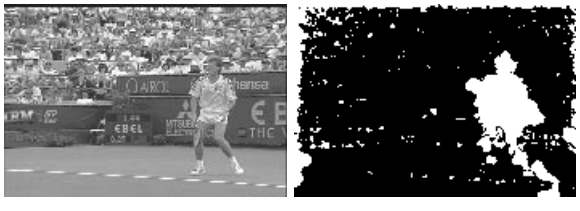

Figure 2. Stefan: camera motion outliers

The algorithm for the depth layering of an image was tested on the *Flower Garden* sequence. Before applying the algorithm to this image sequence, the existence of panning was confirmed through the use of the hierarchy models. The final depth layers extracted are those given at Figure 3.

6 Summary

In this paper, we described methods for solving two important problems in 3-D motion analysis: camera motion estimation and image depth layering. Both methods assume that a dense optic flow field is granted. Instead of having a fixed and predefined motion model, an hierarchy of models was used to estimate the camera motion. From the group of models inside this hierarchy, the one with the fewest parameters that can adequately describe the camera motion is selected. This way the effect of the incorrect optical flow data is minimized. Concerning the second problem, an iterative deterministic relaxation algorithm was used. In this case, to overcome the problem of the possible existence of outliers in the apparent motion field, luminance information was also used for the extraction of the depth layers. The algorithms have been tested on real image sequences. The results obtained were satisfactory and proved the robustness of these methods.

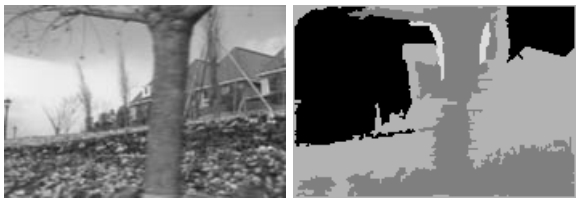

Figure 3. Flower Garden depth layers

References

[1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. on Pattern Analysis Machine Intelligence*, 7:384–401, 1985.

[2] P. Chou and C. Brown. The theory and practice of bayesian image labeling. *International Journal of Computer Vision*, 4:185–210, 1990.

[3] Y. Huang *et al.* Optic flow field segmentation and motion estimation using a robust genetic partitioning algorithm. *IEEE Trans. on Pattern Analysis Machine Inteligence*, 17:1177–1190, Dec. 1995.

[4] M. Hoetter. Differential estimation of the global motion parameters zoom and pan. *Signal Processing*, 16:249–265, 1989.

[5] P. Huber. *Robust statistics*. Wiley, 1981.

[6] A. N. Netravali and J.Salz. Algorithms for estimation of three-dimensional motion. *ATT Technical Journal*, 64, 1985.

[7] H. Nicolas. *Hiérarchie de modèles de mouvement et méthodes d'estimation associées*. Ph.D. thesis, Université de Rennes I, 1992.

[8] F. Pedersini, A. Sarti, and S. Tubaro. Combined motion and edge analysis for a layer-based representation of image sequences. *Proc. IEEE Conf. on Image Processing*, pages 921–924, Sept. 1996.

[9] G. Tziritas and C. Labit. *Motion analysis for image sequence coding*. Elsevier, 1994.

[10] J. Wang and E. Adelson. Representing moving images with layers. *IEEE Trans. on Image Processing*, 3:625–638, 1994.