USER CHOICES FOR EFFICIENT 3D MOTION AND SHAPE EXTRACTION FROM ORTHOGRAPHIC PROJECTIONS

Yiannis Xirouhakis, Gabriel Tsechpenakis and Anastasios Delopoulos

Computer Science Div., Dept. of Electrical and Computer Engineering,
National Technical University of Athens,
9 Iroon Polytechniou str., Athens GR-15773, GREECE
e-mail: jxiro@image.ece.ntua.gr

ABSTRACT

The extraction of structure-from-motion emerges in several research fields such us computer vision, video coding, biomedical engineering and human-computer interaction. Present work focuses on the algorithmic approach of structure-from-motion extraction under orthography providing, at the same time, guidelines in matters of implementation. Relative principles, constraints and stability are discussed. The employed algorithm was first presented in [2]. The improvement of the algorithm's performance w.r.t. the proposed user-choices is illustrated by means of experimental results.

1. INTRODUCTION

The estimation of *3D* motion and structure of objects from monocular or stereo image sequences has been for years a problem for computer vision researchers [4]. This task is recently given considerable attention, especially after the guidelines of the Moving Picture Experts Group regarding MPEG-4 and MPEG-7 standards.

Regarding the employed projection model, the two cases mainly considered in literature are the perspective and the orthographic ones, whereas approximations on these models have also been treated. Exact mathematical solutions have been proposed for both methods including for example [6] for the perspective and [3,7] for the orthographic case. Considerable attention has been given lately to the improvement of the solution's stability in the presence of noisy input estimates (optical flow, feature correspondences). Such work can be found for example in [8] for the perspective and [1,2,5] for the orthographic case.

Although, in general, the obtained results are far from the desired, it can be seen that, for most algorithms, certain choices of the user improve dramatically the quality of the estimates. Present work focuses on making these points clear, providing guidelines for the interested user in order to improve structure and motion computation procedures. At the same time, the theoretical interpretation for the proposed user choices is discussed. The introduced guidelines are directly applicable to the reconstruction algorithm of [2], while they can be extended to other algorithms including the previously referenced ones.

2. BACKGROUND

2.1 Theoretical Analysis

The movement of a point $\mathbf{p}(x,y,z)$ on the object to $\mathbf{p}'(x',y',z')$, can be represented by,

$$\mathbf{p}' = \mathbf{R}\,\mathbf{p} + \mathbf{T}, \qquad (1)$$

where $\mathbf{R}$, $\mathbf{T}$ are the rotation matrix and translation vector respectively. Assume we are given three frames of a video sequence containing a rigid object. Let $\mathbf{r}_p^0$, $\mathbf{r}_p^1$, $\mathbf{r}_p^2$ be the *2D* reference vectors for each point p (i.e. the position of the point on the image plane) in frames 0,1,2 respectively. Let also $v_p^{0\to1}, v_p^{0\to2}$ be the corresponding motion vectors using frame 0 as reference (e.g. $v_p^{0\to1} = \mathbf{r}_p^1 - \mathbf{r}_p^0$). On the basis of three point correspondences (p=1,2,3) for the transition between frames 0 and 1, the 2x2 motion matrices $\boldsymbol{K}$ and $\widetilde{\boldsymbol{K}}$ are defined through [2]:

$$\Delta\mathbf{r}^1 = \boldsymbol{K}^{0\to1}\Delta\mathbf{r}^0,\ \Delta\mathbf{v}^{0\to1} = \widetilde{\boldsymbol{K}}^{0\to1}\Delta\mathbf{r}^0, \qquad (2)$$

where $\Delta\mathbf{v}^{0\to1} = [\mathbf{v}_3^{0\to1} - \mathbf{v}_1^{0\to1} \quad \mathbf{v}_2^{0\to1} - \mathbf{v}_1^{0\to1}]$, while $\Delta\mathbf{r}^0 = [\mathbf{r}_3^0 - \mathbf{r}_1^0 \quad \mathbf{r}_2^0 - \mathbf{r}_1^0]$ and $\Delta\mathbf{r}^1 = [\mathbf{r}_3^1 - \mathbf{r}_1^1 \quad \mathbf{r}_2^1 - \mathbf{r}_1^1]$.

Appropriate transition indices have been added to $\boldsymbol{K}, \widetilde{\boldsymbol{K}}$. It turns out that use of $\mathbf{L} = \mathrm{adj}(\boldsymbol{K}) = \mathrm{adj}(\widetilde{\boldsymbol{K}} + \mathbf{I})$ may simplify the subsequent mathematical notation. Matrix $\mathbf{L}$ (or $\boldsymbol{K}, \widetilde{\boldsymbol{K}}$) characterizes the structure and motion of a plane (defined by a triplet of points) in the *3D* space. Considering the i-th triplet of points, along with their correspondences between frames I and J, $\mathbf{L}_i^{I\to J}$ denotes the corresponding $\mathbf{L}$ matrix. In this sense, given four point correspondences over three frames, the unique solution is provided by the algorithm using as input four $\mathbf{L}$-matrices - namely $\mathbf{L}_1^{0\to1}, \mathbf{L}_2^{0\to1}, \mathbf{L}_1^{0\to2}$ and $\mathbf{L}_2^{0\to2}$. The estimation of motion and structure in this case involves estimation of matrices $\mathbf{R}^{0\to1}, \mathbf{T}^{0\to1}$, $\mathbf{R}^{0\to2}, \mathbf{T}^{0\to2}$ and depth z_p for each employed point $\mathbf{r}_p$ ($\mathbf{r}_p^0$ for example in frame 0).

For two triplets of points i, j the following 2x2 matrix is defined

$$\Delta\mathbf{L}_k^{I\to J} = \mathbf{L}_i^{I\to J} - \mathbf{L}_j^{I\to J}. \qquad (3)$$

For each transition, the 2x2 matrix $\mathbf{Y}^{I\to J}$ is computed as

$$\mathbf{Y}^{I\to J} = \frac{1}{M}\sum_{k=1}^{M}(\Delta\mathbf{L}_k^{I\to J})^T \Delta\mathbf{L}_k^{I\to J}, \qquad (4)$$

with eigenvalues $\lambda_{\min}^{I\to J} < \lambda_{\max}^{I\to J}$ corresponding to the unit-norm eigenvectors $\mathbf{c}^{I\to J}, \mathbf{J}\mathbf{c}^{I\to J}$, for $\mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$.

The quantities defined up to this point are those expressions of the available motion vectors that are sufficient for the computation of both **R** and **T**. In fact, writing $\mathbf{R} = \begin{bmatrix} \mathbf{R}_{2x2} & \mathbf{r}_1 \\ \mathbf{r}_2^T & r_{33} \end{bmatrix}$ as a composite matrix, where $\mathbf{R}_{2x2} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{33} \end{bmatrix}$, $\mathbf{r}_1 = [\, r_{13} \;\; r_{23} \,]^T$ and $\mathbf{r}_2 = [\, r_{31} \;\; r_{32} \,]^T$, one may observe that the entire rotation matrix is determined solely by $\mathbf{r}_1$, $\mathbf{r}_2$ and r_{33}.

As a first step of the algorithm, $\mathbf{r}_1^{I\to J}, \mathbf{r}_1^{I\to K}$ and $\mathbf{r}_2^{I\to J}, \mathbf{r}_2^{I\to K}$ are computed, where as usual the superscripts denote the successive transitions between frames. It has been proved that $\mathbf{r}_1^{I\to J}$ is equal to the eigenvector $\mathbf{c}^{I\to J}$ within a scalar ambiguity, i.e. $\mathbf{r}_1^{I\to J} = \rho^{I\to J}\mathbf{c}^{I\to J}$. For any two transitions in the form of $I \to J$ and $I \to K$, it has been also proved that the ambiguity ratio w can be computed as

$$\left(w^{I\to J, I\to K}\right)^2 = \left(\frac{\rho^{I\to J}}{\rho^{I\to K}}\right)^2 = \frac{\lambda_{\max}^{I\to J} - \lambda_{\min}^{I\to J}}{\lambda_{\max}^{I\to K} - \lambda_{\min}^{I\to K}} \qquad (5)$$

As a next step, the least-squares estimates of unknowns $r_{33}^{I\to J}$ and $r_{33}^{I\to K}$ are proved to be given by

$$\begin{bmatrix} r_{33}^{I\to J} \\ r_{33}^{I\to K} \end{bmatrix} = (\Theta^{I\to J, I\to K})^{-1}\, \Psi^{I\to J, I\to K} \qquad (6)$$

where the 2x2 matrix $\Theta^{I\to J, I\to K}$ and the 2x1 vector $\Psi^{I\to J, I\to K}$ contain summations of **L** matrix products, and expressions of $\mathbf{c}^{I\to J}, \mathbf{c}^{I\to K}$, $w^{I\to J, I\to K}$ and the minimum eigenvalues of $\mathbf{Y}^{I\to J}$ and $\mathbf{Y}^{I\to K}$.

The rest elements of the rotation matrices can be straightforwardly computed. Rotation is estimated irrespective of translation or depth. Although the absolute depth cannot be determined, relative depth may be estimated by fixing an arbitrary point of the rigid object onto the image plane.

2.2 The Algorithm in Steps

Based on the theoretical analysis of [2] described in short in the previous subsection, the algorithm can now be decomposed in few simple steps. Initially,

A-1 Compute the motion field for each transition using a motion-estimation scheme.

A-2 Based on some confidence measures of the motion estimates, decide on the subset of motion estimates to be employed in the computations or employ all motion estimates with appropriate weights.

For each transition,

B-1 Divide the available points (point correspondences) in triplets and compute the **L**-matrices using eq. (2).

B-2 Group triplets in triplet-pairs and compute Δ**L**-matrices from eq. (3).

B-3 Compute **Y** and then $\lambda_{\max}$, $\lambda_{\max}$, $\mathbf{c}_1$.

For certain pairs of transitions,

C-1 Compute w from eq. (5)

C-2 Compute rotation matrices' elements r_{33} using eq. (6), and then rotation matrices.

C-3 Compute translations and depth for all employed points using eq. (1).

Step A-2 is skipped once noise-free (exact) point correspondences are available. At this point, a number of questions arise, as far as user-choices are concerned, on the steps A-2, B-1 and B-2 of the algorithm. Namely: (i) how can one measure efficiently the relative confidence of the motion estimates, (ii) how many point correspondences should be employed, (iii) what is the most robust-to-noise method for the division of available points into triplets, (iv) what are the best pairs of **L**-matrices in order to derive Δ**L**s.

3. USER CHOICES

The investigation of the questions posed in Section 2.2 is carried out in the subsequent subsections

3.1 Using More Point Correspondences

Since generally motion vector estimates contain errors (modeled as additive noise), one should use as many point correspondences as possible. In particular, it can be proved that $\mathbf{Y}^{I\to J}$ defined by eq. (4) is a strongly consistent estimator of its noise-free counterpart and it can be used in place of the corresponding noise-free quantity in the steps B-3 and C-1 of the algorithm. Strong convergence of the obtained estimates is achieved as $M \to \infty$, i.e., as the number of the employed triplet pairs increases provided that the following rules are obeyed in the choice of the employed points, point-triplets and point-triplet pairs: (R1) for each frame, each point triplet (i) is formed on the basis of distinct reference points (p) and (R2) for each frame, each point-triplet pair (k) is formed on the basis of distinct triplets (i). Similar convergence results and rules hold true for the matrices $\Theta^{I\to J, I\to K}$ and $\Psi^{I\to J, I\to K}$, which are used in step C-2 of the algorithm.

Consequently, for a given set of p=1...P points and associated motion vectors, the user should (i) employ the highest possible number P of available point correspondences, (ii) partition the available P points into $N = \lfloor P/3 \rfloor$ distinct point triplets, (iii) form $M = \lfloor N/3 \rfloor$ distinct triplet pairs. As $P \to \infty$ both M, N tend also to ∞.

3.2 Weighted Estimators

Since, in general, most of the motion vector estimates do not coincide with the true ones, we should define a number of criteria to decide on the relative confidence of the motion estimates. Then, a weighted least-squares solution should be employed by involving in the formulas of the previous section appropriate weights w.r.t. the confidence of each estimate. As an example, matrix $\mathbf{Y}^{I\to J}$, defined in (4), is substituted by $\mathbf{Y}^{I\to J} = \frac{1}{M}\sum_{k=1}^{M}\theta_k(\Delta\mathbf{L}_k^{I\to J})^T\Delta\mathbf{L}_k^{I\to J}$, where θ_k denote appropriate weights imposed to triplet-pair k. Weight θ_k relies on the confidence of the triplets i and j forming the triplet-pair k, which in turn are expressions of confidence values associated to the involved motion vectors.

In order to assign a confidence value w_p to a particular motion vector $\mathbf{v}_p$ we take into account (a) the local intensity smoothness, (b) the local motion vectors variance and (c) the homogeneity of a motion vector in its neighborhood, i.e., confidence for $\mathbf{v}_p$ is computed as the linear combination,

$$w_p = \alpha\, w_p^S + \beta\, w_p^V + \gamma\, w_p^D, \qquad (7)$$

where w_p^S = {*sum of the first derivatives in a block centered at p*}, w_p^V = {*the inverse of the motion vectors variance in a block centered at p*}, w_p^D = {*the inverse of the bias of* $\mathbf{v}_p$ *w.r.t. the mean motion vector computed in a block centered at p*}.

3.3 Choosing Point Triplets

Apart from the desired accuracy of the motion vectors we employ, as input in the algorithm, there are some other motion vector properties that we may exploit in order to improve the algorithm's performance. As a matter of fact, these properties concern more the 'grouping' of motion vectors rather than the motion vectors themselves. For example, intuitively, we would prefer a triplet of points defining a plane undergoing a large displacement in orientation. Moreover triplets of points relatively far from each other tend to be more insensitive to noise, as it will be seen in the sequel.

In [2], it was proved that if ξ represents the unit vector perpendicular to a triangle defined by a point triplet in *3D* space and $\mathbf{R}_{2x3}$ is the 2x3 matrix that contains the first 2 rows of the 3x3 rotation matrix of the triangle w.r.t. any arbitrary axis, then for this triangle, motion matrix $\boldsymbol{K}$ can be expressed as:

$$\boldsymbol{K} = \boldsymbol{R}_{2x3}\begin{bmatrix}1 & 0\\ 0 & 1\\ p & q\end{bmatrix} \qquad (8)$$

where the scalars $p = -\xi_x/\xi_z$, $q = -\xi_y/\xi_z$ contain the orientation information of the triangle.

From eq. (8) we conclude that $\Delta\boldsymbol{K} = \boldsymbol{K}_2 - \boldsymbol{K}_1$, and equivalently matrix $\Delta\mathbf{L}_k^{I\to J}$ in equation (3), tends to be more 'informative' when matrices $\mathbf{L}_i^{I\to J}, \mathbf{L}_j^{I\to J}$ correspond to triplets defining triangles with large difference in orientation. The latter is encapsulated in scalar factors $\Delta p = p_2 - p_1$ and $\Delta q = q_2 - q_1$. Thus, one should try to maximize factors Δp, Δq by choosing triplet-pairs that seem to define pairs of triangles in *3D* space with their surface normals as close to perpendicular as possible.

Having decided on the choice of triplet-pairs, one faces the problem of how to choose the point triplets themselves (algorithm's step B-1). In equation (8), matrix $\boldsymbol{K}$ (equivalently $\mathbf{L}$) appears to be more 'informative' for a given rotation when scalars p, q are larger in absolute value. From their definition, $p = q = 0$ when the triangle is parallel to the image plane, whereas $p,\ q = \infty$, when it is perpendicular to it. The latter extreme case cannot happen, since then the triangle projects onto a straight line and thus cannot be observed. However, the valuable information which intuitively rises, is the fact that we should look for projections of triplets of points forming triangles as close as possible to the perpendicular to the image plane *3D* triangle. One easily-observed case where the latter is likely to happen is around abrupt edges.

In addition, as mentioned in Section 3.1, it is essential that $\mathbf{L}_i^{I\to J}$ matrices are formed on the basis of distinct triplets of points. In this case, the effect of noise in motion vectors for each triplet is proved to be encapsulated in a scalar parameter,

$$\pi_j^2 = \sigma_\varepsilon^2\,\frac{\left\|\mathbf{r}_3^j-\mathbf{r}_1^j\right\|^2+\left\|\mathbf{r}_2^j-\mathbf{r}_1^j\right\|^2+\left\|\mathbf{r}_3^j-\mathbf{r}_2^j\right\|^2}{\left|\det\left[\mathbf{r}_3^j-\mathbf{r}_1^j \quad \mathbf{r}_2^j-\mathbf{r}_1^j\right]\right|^2} \qquad (9)$$

Equation (9) indicates that for given SNR levels the influence of noise is proportional to the ratio on the RHS. Any triplet of non-collinear *2D* reference vectors $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$, defines a rectangle (or triangle) whose sides are given as vectors by $\mathbf{r}_2 - \mathbf{r}_1$, $\mathbf{r}_3 - \mathbf{r}_1$. Scalar $det([\mathbf{r}_3-\mathbf{r}_1\ \mathbf{r}_2-\mathbf{r}_1])$ is then proved to yield the area of the respective rectangle. It can be easily proved that for any kind of triangle, this factor tends to be smaller as its side lengths go larger. In this way, by choosing `large' triangles, i.e. by keeping the employed differential reference vectors reasonably large, the noisy terms are minimized.

4. SIMULATIONS

The proposed guidelines were tested on three different types of objects: (i) the computer generated scene of two *3D* planar surfaces (ii) the *3D* model of a teapot and (iii) three frames of a natural sequence containing a coffee can. The 3 scenes for the teapot and the planar surfaces were produced using $\phi^{0\to1}$=9°, $\phi^{0\to2}$=15° and $\phi^{0\to1}$=20°, $\phi^{0\to2}$=40° respectively.

In the examples of the planar surfaces and the teapot model, the motion fields were artificially disturbed by uniformly distributed random noise, in order to illustrate how the proposed guidelines improve the algorithm's performance in the presence of noise. In this sense, Table 1 depicts the improvement in the estimation of rotation parameters for the planar surfaces for increasing number of employed point correspondences (SNR=-5db in differential motion fields) on the basis of 50 Monte Carlo runs. Table 2 depicts the improvement in the estimation of rotation parameters for the planar surfaces for increasing differential reference vectors in the formulation of point triplets (SNR=-5db in differential motion fields). All triplets were chosen on orthogonal triangles of equal side lengths x. Each time side length x was increased, 50 Monte Carlo tests were performed, employing 4500 point correspondences. The teapot model was next used to illustrate that following the guidelines of Section 3, we obtain accurate results even in the case of extremely low SNR levels (Figure 1). For each SNR level 50 Monte Carlo experiments were run (see Table 3). Using all the aforementioned guidelines, the algorithm's performance was tested for the natural sequence as well. In this case, the algorithm's performance was particularly enhanced by the use of appropriate weights, as analyzed in Section 3.2. The estimated motion vector confidence using the three proposed criteria and the reconstructed *3D* model are depicted in Figure 2. The reconstruction scheme is considered to have been particularly successful taking into account the quality of the motion estimates provided by the motion estimation scheme.

Regarding the execution times, it has to be mentioned that the proposed approach is of considerably low computational cost. The most time-consuming task of the system is the motion estimation scheme, in comparison to which the execution time of the reconstruction algorithm is negligible.

M	$\phi^{0\to1}$		$\phi^{0\to2}$	
	mean	std	mean	std
600	20.42	11.84	37.41	11.10
3000	19.07	5.97	39.36	4.57
5400	19.59	3.23	40.05	2.75
7800	19.63	2.66	39.96	2.33
10200	19.96	2.08	40.08	1.98

Table 1: Improvement of the estimated rotation parameters for planar surfaces for increasing number of correspondences.

x	$\phi^{0\to1}$		$\phi^{0\to2}$	
	mean	std	mean	std
6	36.05	58.99	59.85	76.33
26	19.49	4.10	39.08	4.09
46	19.91	1.60	39.80	1.57
66	19.59	1.25	39.77	1.35
71	19.90	0.96	39.85	1.05

Table 2: Improvement of the estimated rotation parameters for planar surfaces for increasing triangle area.

SNR	$\phi^{0\to1}$		$\phi^{0\to2}$	
	mean	std	mean	std
-15	9.87	3.52	18.02	3.84
-10	7.80	2.13	13.87	2.02
-5	8.76	0.91	14.80	0.92
-3	8.89	0.67	14.93	0.73

Table 3: Estimated rotation parameters for the teapot model for increasing SNR.

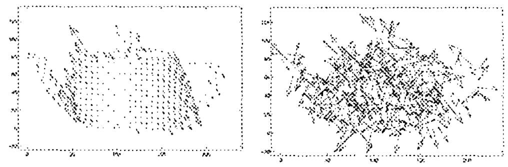

Fig. 1: Noise-free and noise-contaminated differential motion fields for the teapot model.

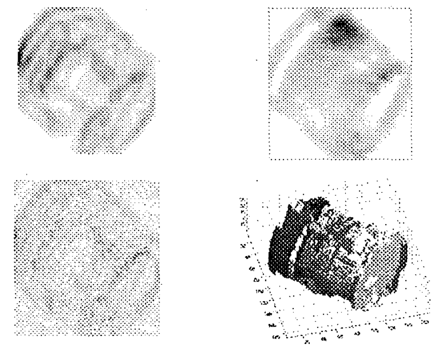

Fig. 2: Motion vector confidence criteria and 3D reconstruction for a natural sequence.

5. CONCLUSIONS

The adopted *3D* reconstruction algorithm [2] was examined from an `experiment design' point of view. All user choices were systematically elaborated targeting to optimal handling of the available point correspondence information. The introduced guidelines were validated by means of simulated experiments. The methodology introduced by this work can possibly be extended to other 3D motion extraction algorithms handling orthographic or even perspective projection.

6. REFERENCES

[1] G. Bozdagi, A.M. Tekalp and L. Onural, "An improvement to MBASIC algorithm for 3-D motion and depth estimation," *IEEE Trans. on Image Processing (special issue)*, vol. 3, pp. 711-716, June 1994.

[2] A. Delopoulos and Y. Xirouhakis, "Robust Estimation of Motion and Shape based on Orthographic Projections of Rigid Objects," *IMDSP98 (IEEE)*, Alpbach Austria, July 1998.

[3] T. S. Huang and C. H. Lee, "Motion and structure from orthographic projections," *IEEE Trans. PAMI*, vol. 11, pp. 536-540, May 1989.

[4] T. S. Huang and A. N. Netravali, "Motion and structure from feature correspondences: A review," *Proc.IEEE*, vol. 82, pp. 252-269, Feb. 1994.

[5] C.Tomasi and T. Kanade, "Shape and Motion from Image Streams under Orthography: a Factorization Method," *Intl. J. of Computer Vision*, vol.9, no.2, pp.137-154, 1992.

[6] R. Tsai and T.S. Huang, "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces," *IEEE Trans. PAMI*, vol.6, pp.13-27, Jan. 1984.

[7] S. Ullman, "The Interpretation of Visual Motion," Cambridge, MA, MIT Press, 1979.

[8] J. Weng, N. Ahuja and T. S. Huang, "Optimal Motion and Structure Estimation," *IEEE Trans. PAMI*, vol. 15, no. 9, pp. 864-884, Sept. 1993.