Structure Estimation and Surface Triangulation of Deformable Objects

Charlotte Svensson[1], Henrik Aanæs[2], and Fredrik Kahl[1]

[1] Centre for Mathematical Sciences, Lund University,
Box 118, SE-221 00 Lund, Sweden
{lotta, fredrik}@maths.lth.se
[2] Informatics and Mathematical Modelling, Technical University of Denmark,
DK-2800 Kongens Lyngby, Denmark
haa@imm.dtu.dk

Abstract. A system is developed that from an image sequence of a deformable object automatically extracts features and tracks them through the sequence, estimates the non-rigid 3D structure and finally computes a surface triangulation. Also the camera motion is acquired. The object is supposed to deform according to a linear model, while the motion of the camera can be arbitrary. No domain specific prior of the object is required.
For the structure estimation a two-step approach is used, where we first obtain an initial estimate of the structure and motion, and then obtain an optimal solution via a non-linear optimization scheme. The triangulation is optimized to yield a non-rigid faceted surface that well approximates the true 3D surface.

1 Introduction

The estimation of structure and motion from image sequences is one of the most studied problems within computer vision. However, almost all the efforts in this area have dealt with rigid objects. Since the world is not a rigid place, it is important to have a system working for deforming objects as well. A common approach to the non-rigid problem is to use a prior model of the object, for example when human body or facial motion is studied [14, 12, 9].

We do not use a prior model, but employ the Principal Component Analysis (PCA) framework, whereby the object is supposed to deform according to a linear model. This type of model is fairly general and have proven to be very effective for expressing many types of deforming objects, e.g. [5]. In the works by [4, 16] such a linear model was used and the structure was estimated via a factorization algorithm. We extend this approach by applying a modified bundle adjustment algorithm to minimize the ML-error.

However, the main novelty compared to previos work is the improved surface modeling. We use the optimized structure to compute a non-rigid surface triangulation, using an approach similar to that of Morris & Kanade [11].

J. Bigun and T. Gustavsson (Eds.): SCIA 2003, LNCS 2749, pp. 709–716, 2003.

2 Tracking

The image sequence is supposed to be taken by a video camera. The feature points are tracked through the sequence using a standard low-level tracking technique, where the correlation of a small window around the feature point between two consecutive frames is used to get the best whole-pixel position. Then we optimize on sub-pixel level, allowing a small affine transformation of the patch.

Without the use of a prior model, tracking can also be facilitated using optical flow, as was introduced by Lucas & Kanade [10]. Applying rank constraints to the flow field helps to overcome the aperture problem, and has been used for both rigid [8] and non-rigid scenes [16].

3 Approximate Solution

3.1 Model Description

The structure of frame i is denoted by $\mathbf{S}_i = \begin{bmatrix} Q_{i1} \cdots Q_{in} \end{bmatrix}$, where Q_{ij} denotes the 3D coordinates of point j in frame i. The Principal Component Analysis (PCA) framework is employed, whereby the object is supposed to deform according to a linear model, i.e.

$$\mathbf{S}_i = \mathbf{S}_\mu + \sum_{k=1}^{r} \beta_{ik} \mathbf{S}_k \ , \tag{1}$$

where β_{ik} is a scalar, $\mathbf{S}_k$ is a 3D mode of variation and $\mathbf{S}_\mu$ is the mean shape. However, we only have the 2D coordinates $w_{ij} = [\, x_{ij}\, y_{ij} \,]^T$, which are the 2D projections of the features Q_{ij}:

$$\lambda_{ij} \begin{bmatrix} x_{ij} \\ y_{ij} \\ 1 \end{bmatrix} = \mathbf{P}_i \begin{bmatrix} Q_{ij} \\ 1 \end{bmatrix} \ , \tag{2}$$

where $\mathbf{P}_i$ is a 3×4 projection matrix. Hence, the problem is to estimate the camera motion $\mathbf{P}_i$ and the structure $\mathbf{S}_i$, i.e. both the mean shape, its modes of variation and the scalars β_{ik}, from the given image data w_{ij}. Also the number of modes of variation, r, needs to be selected. If too few modes are used, the model cannot fully express the non–rigid structure, whereas excess modes would lead to modeling noise. How this model selection can be done automatically was described in [2].

3.2 Motion Estimation

An initial estimate of the camera motion is obtained assuming a rigid structure and solving for structure and motion. This can be done applying the fast factorization technique by Tomasi and Kanade [15], which assumes a linear approximation of the perspective camera model, or with some other standard structure and motion estimation technique, see e.g. [7].

3.3 Varying Structure Estimation

With the approximate motion estimate, the remaining task in getting an approximate solution is estimating the structure, i.e. $\mathbf{S}_\mu$, $\mathbf{S_k}$ and β_{ik}.

We note that good estimates of mean and variance of Gaussian distributed variables are obtained by computing the mean of the observations and the squared residuals with regards to this mean. However, full information of the $\mathbf{S}_i$ is unavailable, since the images are only 2D projections hereof. Thus, an image can be viewed as having a 3D observation with high uncertainty along the viewing direction. Hence, we form a weighted mean, where the weights V_i, of size 3×3, capture the direction where there is no information. With $\mathbf{S}_i^{dir}$ denoting the direct estimate of the structure, the weighted mean becomes

$$\mathbf{S}_\mu = \left(\sum_{i=1}^{m} V_i \right)^{-1} \sum_{i=1}^{m} V_i \mathbf{S}_i^{dir} \quad , \tag{3}$$

and the variance

$$\mathbf{S}_{\mathbf{\Sigma}} = \mathcal{V}^{-1} \sum_{i=1}^{m} \bar{V}_i \left(\bar{\mathbf{S}_i^{dir}} - \bar{\mathbf{S}_\mu} \right) \left(\bar{\mathbf{S}_i^{dir}} - \bar{\mathbf{S}_\mu} \right)^T \bar{V}_i{}^T \quad , \tag{4}$$

where $\bar{V}_i$ and $\mathcal{V}$ are $3n \times 3n$ matrices given by

$$\bar{V}_i = \begin{bmatrix} V_i & & 0 \\ & \ddots & \\ 0 & & V_i \end{bmatrix} \quad \text{and} \quad \mathcal{V} = \sum_{i=1}^{m} \bar{V}_i \begin{bmatrix} 1 \cdots 1 \\ \vdots \ddots \vdots \\ 1 \cdots 1 \end{bmatrix} \bar{V}_i{}^T \quad .$$

The formulas for V_i and $\mathbf{S}_i^{dir}$ are given in [2].

After the model selection, whereby $\mathbf{S}_k$, $k = 1, ..., r$ are deducted from $\mathbf{S}_{\mathbf{\Sigma}}$, the β_{ik} can be found by linear least squares minimization between the model and the image data.

4 Perspective Solution

4.1 Optimal Solution

Similar to traditional bundle adjustment [13], we propose to use a non–linear optimization algorithm on the observation model (2) to get a "gold standard solution"[7]. The collection of object points are parameterized by (1), and a Levenberg–Marquardt approach is applied in order to minimize the reprojection errors in the images.

We assume that the cameras are calibrated, but the same framework and approach would work in the uncalibrated case as well. Each camera is parameterized with a rotation matrix and the coordinates of the camera centre.

4.2 Ambiguities

In the rigid case, there is an ambiguity concerning the world coordinate system and global scale, i.e. the structure and motion can only be determined up to an unknown Euclidean transformation [7]. This ambiguity naturally extends to the non-rigid case. In addition, each mode in the linear model (1) introduces four extra degrees of freedom in the reconstruction. In [1] it was shown that this results in an ambiguity concerning relative translation and scale between the camera centres and the deforming modes of the object. This ambiguity is restricted by imposing a cost for two consecutive instances to differ, as a regularizing prior.

Also, there is an ambiguity concerning the parameterization itself, i.e. between (i) the mean $\mathbf{S}_\mu$ and the modes $\mathbf{S}_k$ and (ii) the weights β_{ik}. This introduces $r(r+1)$ extra degrees of freedom for r modes. They will not change the solution, but may slow down the convergence. More details are given in [2].

5 Surface Triangulation

5.1 Surface Model

It is a standard technique in computer graphics to represent a surface with a triangulation, giving a faceted surface, see e.g. [6]. Our surface model is described by the 3D points, $\mathbf{S}_i$, from Section 4 together with a triangulation, T, and a texture map, A. The triangulation specifies a set of edges and faces connecting all the 3D points in such a way that one faceted surface is created. Since we are dealing with deformable objects, a specific triangle, or facet, in the model has different shape and position for each frame. The texture map for the triangle is however constant through the sequence, since we assume constant lighting and a lambertian reflectance model.

For a given set of points on a surface, the triangulation is not unique, and our goal is to find the triangulation for which the faceted surface best matches the true object surface. For this optimal triangulation, a corresponding texture map is computed.

5.2 Surface Estimation

For a given triangulation, the texture map is easily found from the image sequence by mapping the images onto the triangulation. One particular triangle corresponds to a 3D facet and, if not occluded, an image triangle for each frame, cf. Figure 1. Now consider one such triangle. For each frame, the texture of the image triangle is mapped onto the mean triangle. For a good triangulation, the facets lies close to (the same part of) the true surface for all frames. This means that the mapped textures will be more or less the same. A facet not coinciding with the surface will look very different in different frames due to rotation and deformation of the model. Hence, optimizing the triangulation corresponds to

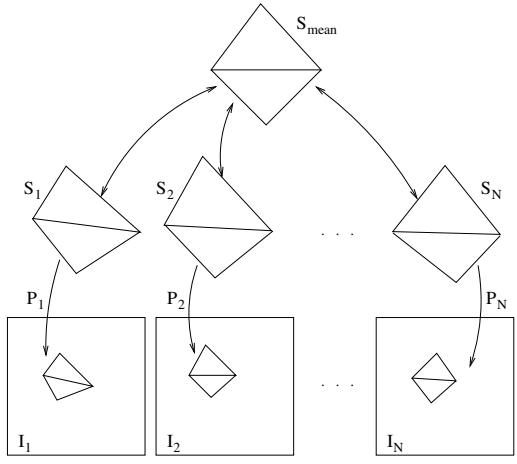

Fig. 1. The surface model is illustrated for two triangles. Here, S means the 3D points plus the triangulation.

minimizing the variance of the mapped texture triangles,

$$\Sigma = \sum_{i=1}^{N} (A_i - A_\mu)^2 \ , \tag{5}$$

where A_i denotes the texture map obtained from all triangles visible in frame i, A_μ is the mean texture across all frames and N is the number of frames. In the optimal case all A_i will be the same, i.e. $A_i = A$.

To optimize the triangulation we use the method described in [11]. A new triangulation is obtained from edge swapping. Two adjacent triangles share an edge and two vertices, and two new triangles are found by deleting this common edge and making a new one between the two vertices of the triangles that were not in common. Which edges to swap is found by a greedy search algorithm, which at each iteration finds the edge swap that will reduce the cost (5) the most. Once the optimized triangulation, T, is found, the texture map, A, is given by the mean texture across all the frames, A_μ.

6 Experimental Results

6.1 Synthetic Data

The triangulation algorithm was first run on a synthetic data set consisting of a box with a checkerboard pattern. Three sides of a box is constructed by 13 nodes, i.e. 7 corner nodes plus two nodes on each side, and a triangulation is made in such a way that three planes are obtained. The box is deformed by moving only the common corner node along a straight line, i.e. we have a one

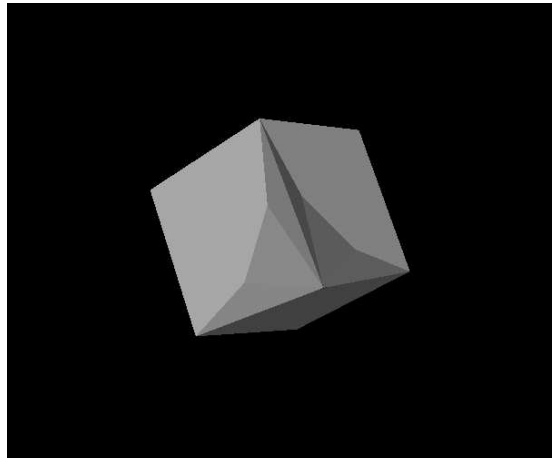

Fig. 2. The structure subject to some deformation. The box is shown without texture but with lighting for better visualization.

mode deformation where the rectangular box deforms to a structure consisting of several plane surfaces.

The same nodes that were made to build the box are used as nodes in the triangulation algorithm. The initialization of the triangulation gives a mesh not describing a rectangular box, but after optimization the triangulation is the same as the true one, cf. Figure 3.

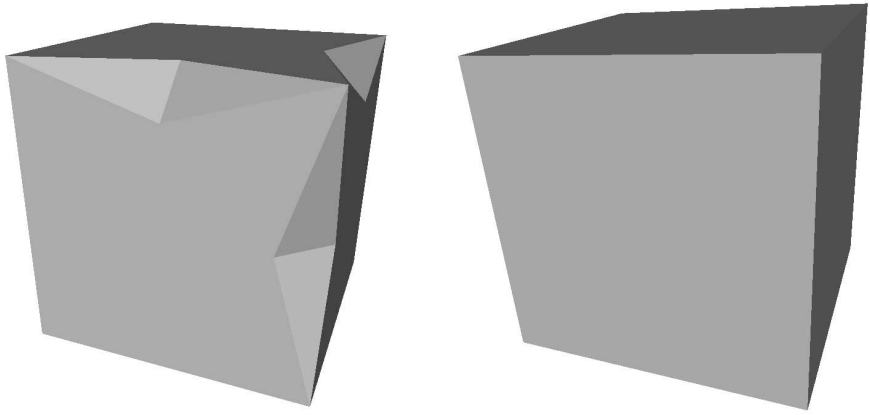

Fig. 3. The mean shape described by the initial (left) and final (right) triangulation.

6.2 Real Data

The second test sequence is a 135 frames video sequence of a talking person. Corner points from the first frame were extracted as features, and these are mainly located around the eyes, nose and mouth. In the structure estimation, every 5:th frame was used and some outliers had to be removed by hand. However, we are facing problems with the triangulation, possibly because the deformation is rather complex and we have only used two modes of deformation. To obtain a smoother, more appealing, triangulated surface, we also need to have more points at the cheeks and forehead, but such points are very hard to track. This work is still in progress.

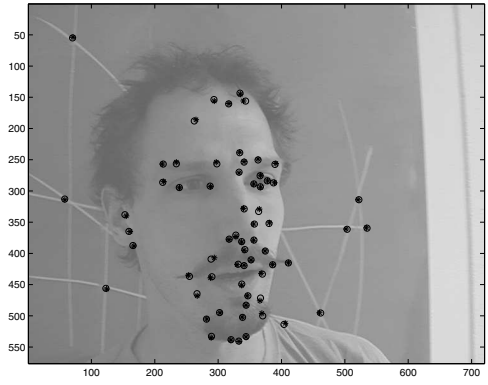

Fig. 4. Tracked (*) and reprojected (o) points after structure estimation.

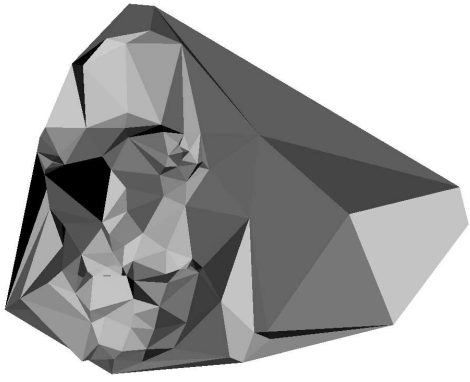

Fig. 5. A triangulation of the surface. Note that the background points are part of the model.

References

1. H. Aanæs and F. Kahl. Estimation of deformable structure and motion. Workshop on Vision and Modelling of Dynamic Scenes, ECCV'02, Copenhagen, Denmark, 2002.
2. H. Aanæs and F. Kahl. Estimation of deformable structure and motion. Technical report, Centre for Mathematical Sciences, Lund University, January 2002.
3. H. Akaike. A new look at the statistical model identification. *IEEE Trans. on Automatic Control*, 19(6):716–723, 1974.
4. C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, pages 690–6 vol.2, 2000.
5. T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models – their training and application. *Computer Vision, Graphics and Image Processing*, 61(1):38–59, January 1995.
6. J.D. Foley, A. van Dam, S.K. Feiner and J.F. Hughes Computer Graphics: Principles and Practice in C (2nd Edition). *Addison-Wesley Pub Co.*, 1995.
7. R. Hartley and A. Zisserman. *Multiple View Geometry.* Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK, 2000.
8. M. Irani. Multi-frame optical flow estimation using subspace constraints. *IEEE International Conference on Computer Vision (ICCV)*, Corfu, September 1999
9. A. Lanitis, C.J. Taylor and T.F. Cootes. Automatic interpretation of human faces and hand gestures using flexible models. In *International Workshop on Automatic Face- and Gesture-Recognition*, 1995
10. B. Lucas and T. Kanade An iterative image registration technique with an application to stereo vision. *Proceedings 7th International Joint Conference on Artificial Intelligence*, 1981
11. D.D. Morris and T. Kanade. Image-consistent surface triangulation. *IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000*, vol 1, pp 332-338, 2000.
12. H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European Conf, Computer Vision'2000*, volume 2, pages 702–718, 2000.
13. C.C. Slama, editor. *Manual of Photogrammetry.* American Society of Photogrammetry, Falls Church, VA, 4:th edition, 1984.
14. C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3d body tracking. In *IEEE Conf. Computer Vision and Pattern Recognition'2001*, volume I, pages 447–454, 2001.
15. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int'l J. Computer Vision'92*, 9(2):137–154, November 1992.
16. L. Torresani, D.B. Yang, E.J. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 1:493 –500, 2001.