COMPLETE 3-D MODELS FROM VIDEO: A GLOBAL APPROACH

Bruno B. Gonçalves and Pedro M. Q. Aguiar

Institute for Systems and Robotics, Instituto Superior Técnico, Lisboa, Portugal
E-mail: bbgo@mega.ist.utl.pt, aguiar@isr.ist.utl.pt

ABSTRACT

We address the automatic recovery of complete 3-D object models from video streams. Usually, complete 3-D models are built by fusing several depth maps, each computed from a small set of consecutive video frames, using structure from motion techniques. However, since from a small number of similar views, it is very difficult to obtain accurate depth maps, their fusion becomes non-trivial and human interaction is in general required to assemble the complete 3-D model. Instead of using intermediate depth maps, we propose a method to recover complete 3-D models *directly* from the 2-D motions in the *entire set* of available video frames. The difficulty that arises when processing long videos is that different regions of the object are seen at different time instants. Our method decides wether a region that has become visible is a region that was seen before, or a previously unseen region, by seeking the *simplest rigid object* that describes well the observed 2-D motions. This global approach increases significantly the accuracy of the estimates of the 3-D shape of the object and the 3-D motion of the camera. Experiments with artificial data and real video demonstrate the good performance of our method.

1. INTRODUCTION

In areas ranging from virtual reality and digital video to robotics, an increasing number of applications need three-dimensional (3-D) models of real-world objects. Although expensive active sensors, *e.g.*, laser rangefinders, have been frequently used to acquire 3-D data, in many relevant situations only two-dimensional (2-D) video data is available and the 3-D object models have to be recovered from their 2-D projections. In this paper, we address the automatic recovery of complete 3-D models from video sequences.

Related work Since the strongest cue to infer 3-D shape from video is the 2-D motion of the image brightness pattern, our problem has been often referred to as structure from motion (SFM). Since using a large number of views, rather than simply two consecutive frames, leads to more constrained problems, thus to more accurate 3-D models, current research has been focused on multi-frame SFM. The factorization method of Tomasi and Kanade [1] overcomes the difficulties of multi-frame SFM—nonlinearity and large number of unknowns—by using matrix subspace projections. In [1], the trajectories of feature points are collected into an observation matrix that, due to the rigidity of the 3-D object, is highly rank deficient in a noiseless situation. The 3-D shape of the object and the 3-D motion of the camera are recovered from the rank deficient matrix that best matches the observation matrix. The work of [1] was extended in several ways, *e.g.*, geometric projection models [2], parametric surface models [3].

Work partially supported by FCT grant POSI/SRI/41561/2001.

References [1, 2, 3] assume that any object region that is being modelled is visible during the entire video sequence, thus leading to a complete observation matrix. Clearly, this is not the case when processing videos that show views all around (non-transparent) 3-D objects. Object self-occlusion, as well as limited field of view and tracking failures, motivated researchers to extend the factorization method to cope with incomplete observation matrices [4, 5, 6]. However, none of the methods above deal with "self-inclusion", *i.e.*, with the fact that a region that disappears due to self-occlusion may appear again later. When this happens, the re-appearing region is usually treated as a new region, *i.e.*, as a region that was never seen before. This procedure has two drawbacks: i) the problem becomes less constrained than it should, thus leading to less accurate estimates of 3-D structure; and ii) further 3-D processing is needed to fuse the recovered multiple versions of the same real-world regions.

Proposed approach We propose a global approach to recover complete 3-D models from video. Global in the sense that our method computes the *simplest* 3-D rigid object that best matches the *entire* set of 2-D image projections. This way we avoid having to post-process several partial 3-D models, each obtained from a smaller set of frames, or an inaccurate 3-D model obtained from the entire set of frames without detecting re-appearing regions. We develop a global cost function that balances two terms—model fidelity and complexity penalization. The model fidelity term measures the error between the model—a 3-D shape and a set of re-appearing regions—and the observations, as in Maximum Likelihood estimation. This error is simply given by the distance of a re-arranged observation matrix to the appropriate space of rank deficient matrices. The penalty term measures the complexity of the 3-D model, which is easily coded by the number of feature points used to describe the observations. By minimizing this global cost, we get what statisticians usually call a Penalized Likelihood (PL) [7] estimate of the 3-D structure. Through PL estimation, re-appearing regions are then detected when the increase of the complexity of the 3-D model does not compensate a slightly better fit to the observations, meaning that a more complex 3-D model would fit the observation noise rather than the 3-D real-world object.

2. 3-D STRUCTURE FROM VIDEO WITH OCCLUSION

The majority of the approaches to the recovery of SFM start by estimating the 2-D motion in the image plane. This is usually done by tracking the image projections of distinctive feature points of the 3-D object. When the 3-D object is rigid and not too close to the camera, the observations, *i.e.*, the trajectories of N feature projections along F frames, are modelled by

$$\mathbf{W}_{2F\times N} = \mathbf{M}_{2F\times 4}\mathbf{S}_{4\times N} + \text{noise}, \qquad (1)$$

where $\mathbf{W}$ collects the coordinates of the feature projections, $\mathbf{M}$ depends on the camera-object 3-D motion, and $\mathbf{S}$ describes the object shape, *i.e.*, it contains the 3-D coordinates of the feature points, see [1]. The problem of recovering SFM amounts then to estimating matrices $\mathbf{M}$ and $\mathbf{S}$ from the observation matrix $\mathbf{W}$.

Matrix factorization method In the early nineties, Tomasi and Kanade introduced model (1) and proposed the now widely known factorization method [1], a computationally simple approach to the recovery of rigid SFM. They noted that although the observation matrix $\mathbf{W}$ in (1) may be huge—dimension $2F \times N$ ($x-$ and $y-$ coordinates of each feature projection)—, it is well approximated by a rank 4 matrix because it is a noisy version of the product of a $2F \times 4$ motion matrix $\mathbf{M}$ by a $4 \times N$ shape matrix $\mathbf{S}$, see (1). The factorization method of [1] exploits this by computing the motion and shape matrices $\mathbf{M}$ and $\mathbf{S}$ from the factors of the rank 4 matrix $\widehat{\mathbf{W}}$ that best matches the observation matrix $\mathbf{W}$. The rank 4 matrix $\widehat{\mathbf{W}}$ is easily obtained from the Singular Value Decomposition (SVD) of $\mathbf{W}$, $\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$, after selecting the 4 largest eigenvalues and the associated eigenvectors,

$$\widehat{\mathbf{W}} = \arg \min_{\widetilde{\mathbf{W}} \in \mathcal{S}_4} \|\mathbf{W} - \widetilde{\mathbf{W}}\|_F \quad \Rightarrow \quad \widehat{\mathbf{W}} = \mathbf{U}_{2F\times 4}\mathbf{\Sigma}_{4\times 4}\mathbf{V}_{4\times N}. \tag{2}$$

Here, $\|.\|_F$ represents the Frobenius norm and $\mathcal{S}_4$ denotes the space of the $2F \times N$ rank 4 matrices.

Factorization with missing data It is often the case in practice that, due to occlusion and tracking failures, the trajectories of the projections of the feature points in the observation matrix $\mathbf{W}$ are incomplete. In this case, the SVD of $\mathbf{W}$ can not be computed and, unlike (2), there is not known closed-form solution to the problem of finding the rank 4 matrix $\widehat{\mathbf{W}}$ that best matches $\mathbf{W}$. Naturally, the cost function in (2) is generalized to this missing data case by summing only the errors of the known entries of $\mathbf{W}$, *i.e.*,

$$\widehat{\mathbf{W}} = \arg \min_{\widetilde{\mathbf{W}} \in \mathcal{S}_4} \|(\mathbf{W} - \widetilde{\mathbf{W}}) \odot \mathbf{M}\|_F, \tag{3}$$

where $\odot$ represents the elementwise product and the binary matrix $\mathbf{M}$ is such that $m_{ij} = 1$ if w_{ij} is known and $m_{ij} = 0$ otherwise.

In [5], the authors used the Expectation-Maximization (EM) approach to missing data problems [8] to minimize the missing data cost function (3). They also derived an extension of the power method [9] that estimates in alternate steps the column and row spaces of the solution of (3). Both algorithms are computationally simple and have good convergent behavior when adequately initialized. In [5], the initialization is computed by composing the column and row spaces of known sub-matrices of $\mathbf{W}$.

3. COMPLETE MODELS — RE-APPEARING FEATURES

To build complete a complete model of a 3-D object, we must use a video stream containing views that completely "cover" the object, typically, a video obtained by rotating the camera around the object. Obviously, as the camera moves, some feature points disappear, due to object self-occlusion, remain invisible during certain period and then re-appear. In general, each feature point has thus several tracking periods. Although this is always the case when constructing complete 3-D models, it also happens very frequently when processing real-life videos in general.

Current tracking algorithms, as well as the factorization methods that deal with occlusion, *e.g.*, [4, 6], or the method [5], outlined in section 2, do not consider a re-appearing feature as another observation of a previously seen point. They rather consider as many feature points as tracking periods. To illustrate this point, we represent on the left image of Fig. 1 the typical shape of the known entries of the observation matrix $\mathbf{W}$. Each feature trajectory is represented by a column of $\mathbf{W}$. The three last columns (in gray) correspond to re-appearing features, *i.e.*, they are second tracking periods of the features that were first tracked and collected in the three first columns. Our goal in this paper is to re-arrange the observation matrix $\mathbf{W}$ into a smaller matrix $\mathbf{W}_R$ that merges all the tracking periods of the same feature in the same column. For $\mathbf{W}$ shown on the left side of Fig. 1, the re-arranged matrix $\mathbf{W}_R$ would be as shown on its right. Finding matrix $\mathbf{W}_R$ is equivalent to detecting the re-appearing features. Note that the statements of section 2 remain valid for the re-arranged matrix $\mathbf{W}_R$; in particular, $\mathbf{W}_R$ is rank 4 in a noiseless situation, just like the original observation matrix $\mathbf{W}$. As pointed out before, the advantage of using $\mathbf{W}_R$ rather than $\mathbf{W}$ is that the SFM problem becomes more constrained, thus leading to more accurate estimates of the 3-D structure.

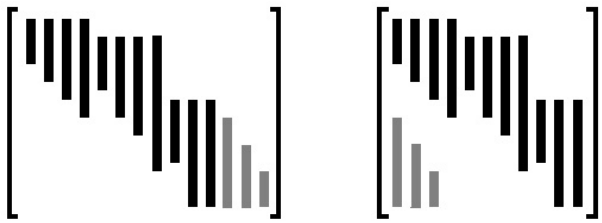

Fig. 1. Left: Original observation matrix $\mathbf{W}$. Right: Re-arranged observation matrix $\mathbf{W}_R$, after detecting re-appearing features.

Local approach When a given feature has two tracking periods, current factorization methods [4, 5, 6], return a 3-D shape containing two 3-D feature points that correspond to the same 3-D point of the real-world object. Although these two 3-D feature points would coincide in a noiseless situation, in practice they are just close to each other. To demonstrate this, we represent on the left plot of Fig. 2, the 3-D shape recovered from a set of synthesized trajectories of 40 features, by using the factorization of [5]. To simulate occlusion, three of the trajectories were artificially "interrupted", leading to three pairs of recovered 3-D points, marked with small circles in the left plot of Fig. 2.

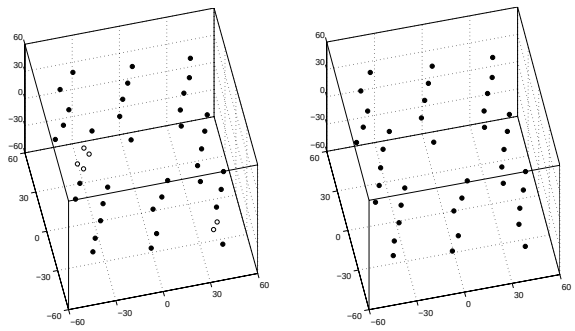

Fig. 2. 3-D shape recovered from the original matrix $\mathbf{W}$, *i.e.*, without detecting re-appearing features (left) and from the re-arranged matrix $\mathbf{W}_R$, *i.e.*, after detecting re-appearing features (right).

A simple way to detect re-appearing features is based on a local analysis of the distance between recovered 3-D points. However, this procedure fails in practice due to the sensibility to the threshold below which the features would be considered to correspond to the same 3-D point. In fact, the distance between the features that correspond to the same 3-D point, depends not only on the noise level but also on the camera-object distance, which is very difficult to estimate accurately enough. The detection of re-appearing features must then be based on a global approach.

4. GLOBAL APPROACH

We formulate the detection of re-appearing features as a model selection problem, where a model is represented by a re-arranged observation matrix $\mathbf{W}_R$. Matrix $\mathbf{W}_R$ codes the number of feature points P_R and the correspondences between columns of the original observation matrix $\mathbf{W}$ and points of the 3-D real-world object. To select the best model, we develop a global PL cost function.

PL cost PL estimation balances the accuracy of the model with its complexity. The PL estimate $\mathbf{W}_R$ leads then to the minimization

$$\mathbf{W}_R = \arg\min_{\mathbf{W}_r} \left\{ E_{\mathcal{S}_4}\left(\mathbf{W}_r\right) + \alpha P_r \right\}, \tag{4}$$

where $E_{\mathcal{S}_4}\left(\mathbf{W}_r\right)$ measures the error of the model $\mathbf{W}_r$ as its distance to the space of rank 4 matrices, *i.e.*, it is a likelihood term, and P_r is the number of feature points of the model, *i.e.*, it codes the model complexity. The parameter α balances the two terms.

We evaluate the likelihood term

$$E_{\mathcal{S}_4}(\mathbf{W}_r) = \min_{\widetilde{\mathbf{W}} \in \mathcal{S}_4} \|(\mathbf{W}_r - \widetilde{\mathbf{W}}) \odot \mathbf{M}_r\|_F, \tag{5}$$

where the binary matrix $\mathbf{M}_r$ accounts for the known entries of the model $\mathbf{W}_r$, by using the algorithm outlined in section 2, see (3).

We performed several experiences in order to find a valid range for the weight parameter α. By testing pertinent values for the noise level, % of missing data, number of features, number of images, and number of re-appearing features, we concluded that

$$\alpha \in \left[1 \times 10^{-4}, 4 \times 10^{-4}\right] \tag{6}$$

leads to a good balance between maximizing the number of correct detections of re-appearing features (probability of detection) and minimizing the number of incorrect detections (probability of false alarm). Obviously, the parameter α could also be chosen by using principles as the Minimum Description Length (MDL) or Akaike's information criteria (AIC).

Minimization algorithm To minimize the cost (4), our algorithm starts by selecting from the original observation matrix $\mathbf{W}$, pairs of columns that can be merged with each other. Naturally, these pairs correspond to disjoint tracking periods. For example, for the matrix $\mathbf{W}$ in the right side of Fig. 1, each one of the six last columns could be merged with the first column, therefore, in what respects to the feature corresponding to the first column, there are seven possible situations: it could either have re-appeared, generating one of the trajectories of the six last columns, or remain occluded for the remaining of the video.

To obtain a computationally feasible algorithm, we prune the search—our algorithm decides by comparing the costs of merging each selected pair of columns with the one of considering that there are not re-appearing features, *i.e.*, of model $\mathbf{W}_R = \mathbf{W}$. The process is then repeated until the cost (4) does not decrease by merging any selected pair of columns. The search could be further pruned by using the local approach outline in the previous section to guide the process, thus testing only pairs of columns corresponding to feature points that are close in the 3-D space.

The iterative algorithm of [5], used to compute the likelihood term (5), converges in very few iterations when adequately initialized. Although the initialization procedure described in [5] is computationally expensive due to several SVD's, we reduce the computational complexity of our method by performing this initialization step only once, when testing the model that corresponds to the original observation matrix, $\mathbf{W}_R = \mathbf{W}$, and using the resulting initial guess also when testing the other models.

5. EXPERIMENTS

Illustrative behavior To illustrate our method, we processed the observation matrix that lead to the left plot of Fig. 2. The 3-D shape recovered by our global method is shown in the right plot, where each of the pairs of re-appearing features have been correctly detected as representing the same 3-D point.

Error analysis We quantified the gain of using our method to recover SFM by measuring the 3-D reconstruction error. We synthesized noisy trajectories of 40 features on the surface of a cube, 15 of them being "interrupted" to simulate object self-occlusion, leading to a 60×55 observation matrix $\mathbf{W}$ with 53.9% known entries ($x-$ and $y-$ coordinates in $[0, 120] \times [0, 160]$, noise standard deviation $\sigma = 3$). Our method generated a 60×40 re-arranged matrix $\mathbf{W}_R$ with 74.2% known entries. By using $\mathbf{W}_R$ rather then $\mathbf{W}$ to recover SFM, we reduced the 3-D shape estimation error by approximately 50% and the 3-D motion error by approximately 70%.

Sensitivity to the noise In order evaluate the sensitivity of our method to the observation noise, we plotted in Fig. 3 the probabilities of correctly detecting re-appearing features and the probability of incorrect detections (false alarms) as functions of the noise level, for the experimental setup described above, now with 5 re-appearing features. These probabilities were estimated from 100 runs for each level of noise. Although re-appearing features become harder to detect as the noise level increases, our method correctly detects more than 90% of them when the noise level is below $\sigma = 5$. The plot of Fig. 3 also shows that our method has approximately zero false alarms for noise levels below $\sigma = 7$.

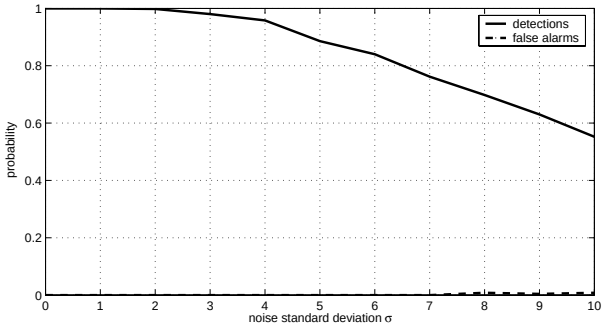

Fig. 3. Probability of detection and probability of false alarm.

Synthetic video We used an artificially generated video with 37 frames of a cube. We tracked 17 features points in the cube surface. Due to the camera-object rotation, several features become occluded along the video sequence and one of them re-appears in

the last frames. The resulting 74×19 observation matrix $\mathbf{W}$ has 77.5% known entries. Fig. 4 shows representative frames of the synthetic video. The top-left and bottom right images also represent, superimposed with the video frame, the re-appearing feature.

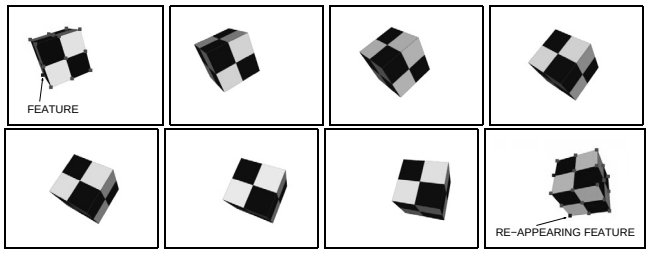

Fig. 4. Synthetic video. Top-left: frame 1 with superimposed features. Bottom-right: frame 26 and one re-appearing feature.

On the left side of Fig. 5, we show the 3-D shape recovered by processing matrix $\mathbf{W}$, *i.e.*, without detecting the re-appearing feature. This shape is particularly inaccurate (note the rightmost feature point) because the video of Fig. 4 does not provide views that cover the entire object. On the right, we show the result of using our global method to detect re-appearing features, *i.e.*, the result of processing the re-arranged matrix $\mathbf{W}_R$. The 3-D shapes in Fig. 5 clearly show that our method leads to a more accurate estimate of the 3-D shape of the object.

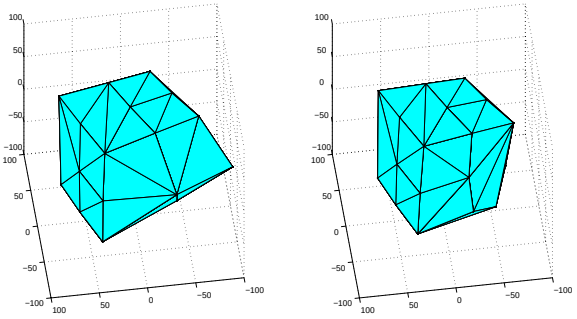

Fig. 5. 3-D shape recovered from the video in Fig. 4. Left: without detecting re-appearing features. Right: using our method.

Real video We used a video obtained by rotating a hand-held camera around a table with a pile of CD's. Fig. 6 shows several frames of the video sequence. We tracked 20 feature points, located on corners of the CD boxes. Along the video sequence, all features disappear and several of them re-appear because the camera performs a complete turn around the CD pile.

On the left image of Fig. 7, we show the 3-D shape recovered from the video of Fig. 6 without detecting re-appearing features—the circles indicate examples of pairs of features that corresponds to the same 3-D point. On the right, we show the result of applying our global method to detect re-appearing features. Note that each of the above mentioned pairs was correctly identified as representing the same point of the 3-D object.

6. CONCLUSION

We proposed a global approach to build complete 3-D models from video. The 3-D model is inferred as the *simplest* rigid object that

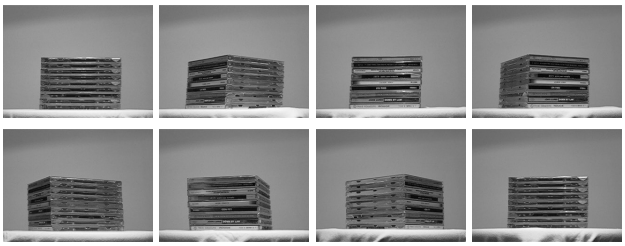

Fig. 6. Real video sequence.

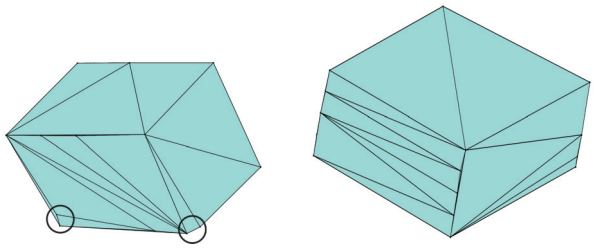

Fig. 7. 3-D shape recovered from the video in Fig. 6. Left: without detecting re-appearing features. Right: using our method.

agrees with all the observed data, *i.e.*, with the *entire* set of video frames. Our experiments show that this method leads to more accurate estimates of 3-D shape than those obtained by either processing several smaller subsets of views or processing the entire video without taking into account re-appearing regions.

7. REFERENCES

[1] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *Int. Journal of Computer Vision*, vol. 9, no. 2, 1992.

[2] C. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," *IEEE T-PAMI*, vol. 19, no. 3, 1997.

[3] P. Aguiar and J. Moura, "Three-dimensional modeling from two-dimensional video," *IEEE T-IP*, vol. 10, no. 10, 2001.

[4] D. Jacobs, "Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images," in *IEEE Computer Vision Pattern Recognoition*, 1997.

[5] R. Guerreiro and P. Aguiar, "3D structure from video streams with partially overlapping images," in *IEEE ICIP*, New York, USA, 2002.

[6] P. Chen and D. Suter, "Recovering the missing components in a large noisy low-rank matrix: Application to SFM," *IEEE T-PAMI*, 2004.

[7] P. Green, "Penalized likelihood," in *Encyclopedia of Statistical Sciences*. John Wiley & Sons, New York, 1998.

[8] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, John Wiley & Sons, New York, 1997.

[9] G. Golub and C. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 1996.