CVPR'08 – IEEE INT'L CONF. ON COMPUTER VISION AND PATTERN RECOGNITION

Coarse-to-Fine Low-Rank Structure-from-Motion

A. Bartoli[1,2] V. Gay-Bellile[1,3] U. Castellani[4] J. Peyras[1] S. Olsen[2] P. Sayd[3]
[1] LASMEA, Clermont-Ferrand [2] DIKU, Copenhagen [3] CEA, LIST, Saclay [4] VIPS, Verona

Abstract

We address the problem of deformable shape and motion recovery from point correspondences in multiple perspective images. We use the low-rank shape model, i.e. the 3D shape is represented as a linear combination of unknown shape bases.

We propose a new way of looking at the low-rank shape model. Instead of considering it as a whole, we assume a coarse-to-fine ordering of the deformation modes, which can be seen as a model prior. This has several advantages. First, the high level of ambiguity of the original low-rank shape model is drastically reduced since the shape bases can not anymore be arbitrarily re-combined. Second, this allows us to propose a coarse-to-fine reconstruction algorithm which starts by computing the mean shape and iteratively adds deformation modes. It directly gives the sought after metric model, thereby avoiding the difficult upgrading step required by most of the other methods. Third, this makes it possible to automatically select the number of deformation modes as the reconstruction algorithm proceeds. We propose to incorporate two other priors, accounting for temporal and spatial smoothness, which are shown to improve the quality of the recovered model parameters.

The proposed model and reconstruction algorithm are successfully demonstrated on several videos and are shown to outperform the previously proposed algorithms.

1. Introduction

Recovering 3D shape and camera parameters from images is a major research topic in computer vision. The classical Structure-from-Motion paradigm assumes that the observed shape is rigid. It often uses image point tracks obtained by some means. The rigid shape assumption means that the viewing rays corresponding to the same physical point seen in different cameras intersect in space.

For the case of a deforming 3D shape, the assumption that the viewing rays intersect does not hold true. Model-free non-rigid Structure-from-Motion is tackled in *e.g.* [5, 4, 11, 13]. An approach that recently proved successful is the one using the low-rank shape model, which represents the 3D deforming shape by a linear combination of shape bases we call deformation modes or simply *modes*. The modes are point-dependent while the linear combination coefficients, called *configuration weights*, are view-dependent. The representative power of this model lies in its ability to capture, as Principal Component Analysis does, the structure underlying the actual deformations of the 3D shape. The main assumption on the 3D shape is that it consists of a single moving and deforming object, so that the deformation at each point has some sort of consistency with the other points, as is formally defined in [15]. The low-rank terminology stems from the fact that the number of modes is assumed much lower than the number of images and points.

The major difference of the proposed method with the previous ones lies in the coarse-to-fine definition of the low-rank shape model we use. Most of the previous methods treat modes equally, resulting in ambiguities as any mode can be replaced by a linear combination of the other ones. In contrast, we use the rule that a deformation mode encapsulates as much of the data variance left unexplained by the preceding modes as possible. This has important practical impacts, as the level of ambiguity is drastically reduced and makes a coarse-to-fine reconstruction algorithm possible. The idea is that the modes capture decreasingly important details in the deformation. Our model is based on composing this coarse-to-fine low-rank shape model with euclidean transformations accounting for the global displacement of the object of interest. The number of modes is automatically chosen based on Cross-Validation.

To summarize, this paper brings a novel low-rank Structure-from-Motion method which handles missing data, automatically selects the number of deformation modes and makes use of several different priors. We report experimental results on challenging datasets showing that the method gives sensible 3D shapes, allowing us to convincingly augment the video by adding a virtual 3D object on a deforming surface.

2. Previous Work and Contributions

Previous low-rank Structure-from-Motion methods differ by the optimization method and the priors they use,

and if they order the modes or not. Early methods such as [5] use no prior. They are based on computing an 'implicit model' for which the configuration weights and camera parameters are mixed up together through a mixing matrix. The implicit model is upgraded to the 'explicit model' (the model described so far). An efficient implicit model reconstruction method is described in [9]. Recent papers focus on how to compute the implicit to explicit upgrade [4, 13]. While most papers use an affine camera model, some recent papers consider the case of a perspective camera, *e.g.* [6, 12].

Aanaes and Kahl [1] take a different approach: they view the low-rank shape model as a mean shape, that they compute using rigid Structure-from-Motion, and modes that are found through Principal Component Analysis of the directional variance. The overall model parameters are refined together through bundle adjustment. In contrast, we compute the mean shape and iteratively add modes by minimizing the reprojection error. This has the advantage to result in a coarse-to-fine model, expressed in a metric framework, thus *avoiding the difficult implicit-to-explicit upgrading step*. The coarse-to-fine scheme ensures that the leading modes encapsulate coarsest deformations. We show that the deformation mode estimation problem can be splitted into several much smaller problems. The resulting algorithm is efficient and copes with missing data resulting from occlusions.

Finally, there are few papers on the crucial problem of selecting the number of modes. Existing approaches are based either on inspecting the eigenvalues of the data matrix [14] or on model selection criteria such as BIC [1] or GRIC [3]. We provide a solution based on Cross-Validation which, contrarily to previous approaches, does not assume a gaussian *iid* distribution with known variance on the residuals. We show that it gives very sensible results with respect to ground truth.

3. Background

3.1. Notation and Camera Model

Everything is in homogeneous coordinates. A 3D point $\mathbf{Q}$ projects to a 2D point $\hat{\mathbf{q}} \overset{\text{def}}{\sim} \mathsf{P}\mathbf{Q}$ through camera P, where P is a (3×4) perspective projection matrix. The *reprojection error* for this image point is the euclidean distance $d(\mathbf{q}, \hat{\mathbf{q}})$ between the model-predicted point $\hat{\mathbf{q}}$ and the corresponding data point $\mathbf{q}$. The corresponding *algebraic reprojection error* is given by using the following algebraic distance:

$$\mu^2(\mathbf{q}, \hat{\mathbf{q}}) \overset{\text{def}}{=} \|\mathsf{S}(\mathbf{q} \times \hat{\mathbf{q}})\|^2 \qquad \text{with} \qquad \mathsf{S} \overset{\text{def}}{=} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad (1)$$

where $\|\cdot\|$ is the two-norm for vectors and Frobenius norm for matrices. The point-to-line orthogonal distance between $\mathbf{q}$ and $\mathbf{l}$ is written $d_\perp(\mathbf{q}, \mathbf{l})$. The following is an algebraic approximation:

$$\mu^2_\perp(\mathbf{q}, \mathbf{l}) \overset{\text{def}}{=} \left(\mathbf{q}^\mathsf{T}\mathbf{l}\right)^2. \quad (2)$$

We use 'normalized' image coordinates which are known to improve the performance of algebraic approximations [7]. The data points lying on the deforming object in the image are written $\mathbf{q}_{i,j}$ where $i = 1, \ldots, n$ is the image index and $j = 1, \ldots, m$ the point index. The binary entries $v_{i,j}$ of the $(n \times m)$ visibility matrix V indicate missing data.

We write $SE(3)$ the group of euclidean transformations in 3-space; $\mathsf{E} \in SE(3)$ is a (4×4) matrix. We define $\mathcal{R}(\mathsf{E}) \overset{\text{def}}{=} \mathsf{R}$ and $\mathcal{T}(\mathsf{E}) \overset{\text{def}}{=} \mathbf{t}$ as the (3×3) rotation matrix and (3×1) translation vector in E respectively.

3.2. The Low-Rank Non-Rigid Shape Model

The deforming 3D points $\mathbf{S}_{i,j}$ are modeled by combining l modes and a mean shape $\bar{\mathbf{M}}_j^\mathsf{T} = (\bar{\mathbf{M}}_j^\mathsf{T} \;\; 1)$. Mode k is defined point-wise by $b_{k,j}\mathbf{C}_{k,j}$ with $\|\mathbf{C}_{k,j}\| = 1$ with $\mathbf{C}_{k,j}^\mathsf{T} = (\bar{\mathbf{C}}_{k,j}^\mathsf{T} \;\; 0)$ a direction vector and $b_{k,j}$ a deformation magnitude. Camera-wise configuration weights are written $a_{i,k}$. The l-mode shape is:

$$\mathbf{S}_{i,j}^l \overset{\text{def}}{\sim} \mathsf{D}_i \left(\mathbf{M}_j + \sum_{k=1}^{l} a_{i,k} b_{k,j} \mathbf{C}_{k,j} \right), \quad (3)$$

where the $\mathsf{D}_i \in SE(3)$ are aligning transformations, so that the mean shape and its deformations are expressed in an object-centred coordinate frame. Each mode allows a 3D point to move in some direction by a point-dependent and a view-dependent magnitude. The aligning transformations D_i are important since we want the deformation modes to express intrinsic object deformations as opposed to object displacements. The prediction of an image point, *i.e.* the reprojection of a 3D point under this model, writes:

$$\mathbf{s}_{i,j}^l \overset{\text{def}}{\sim} \mathsf{P}_i \mathbf{S}_{i,j}^l \sim \mathsf{P}_i \mathsf{D}_i \mathbf{M}_j + \bar{\mathsf{P}}_i \mathcal{R}(\mathsf{D}_i) \sum_{k=1}^{l} a_{i,k} b_{k,j} \bar{\mathbf{C}}_{k,j}, \quad (4)$$

with $\mathsf{P}_i = \mathsf{K}_i(\mathrm{I} \;\; \mathbf{0})\mathsf{E}_i$. We define the n-vector $\mathbf{a}_l \overset{\text{def}}{=} (a_{1,l} \;\; \cdots \;\; a_{n,l})$, the m-vector $\mathbf{b}_l \overset{\text{def}}{=} (b_{l,1} \;\; \cdots \;\; b_{l,m})$, the $3m$-vector $\bar{\mathbf{C}}_l^\mathsf{T} \overset{\text{def}}{=} (\bar{\mathbf{C}}_{l,1}^\mathsf{T} \;\; \cdots \;\; \bar{\mathbf{C}}_{l,m}^\mathsf{T})$ and $\bar{\mathbf{B}}_l$ similarly.

This model has ambiguities caused by internal 'gauge freedoms'. There is obviously an undetermined euclidean transformation between the mean shape and modes, and the aligning transformations. For globally estimated modes, as in standard approaches, there is an l^2 representational ambiguity since any mode can also be replaced by any linear combination of all modes. In our method, each mode is estimated conditioned on the coarser ones, yielding only a single degree of ambiguity for each mode. Indeed, equation (4) shows that mode l contributes through the exterior product $\mathbf{a}_l\mathbf{b}_l^\mathsf{T}$ which factors can be rescaled since $\forall \nu \in \mathbb{R}^\star$, $\mathbf{a}_l\mathbf{b}_l^\mathsf{T} = (\nu\mathbf{a}_l)\left(\frac{1}{\nu}\mathbf{b}_l^\mathsf{T}\right)$.

3.3. More Priors

The low-rank shape model is very sensitive to the number of modes. Since this is a empirical model, there might not be an ideal such number. Bad results are reported in [11] when the basic low-rank shape model is used to find the 3D shape. Priors are needed in order to better constrain the model. We review some generic priors, where generic means not specific to some object or object-class.

A simple prior is the one of assuming a part of the scene to be rigid [6]. [1] uses as prior the fact that the shape should be close in neighbouring frames. [11] uses a gaussian distribution prior on the configuration weights. This allows to marginalize the configuration weights out of the estimation, which can then be performed very efficiently. They also propose to model temporal camera smoothness through a Linear Dynamics model. The transition matrix is estimated along with the other unknown parameters.

[9] uses a temporal smoothness prior penalizing variations in the implicit camera matrices, embedding both the camera parameters and configuration weights:

$$\sum_{k=1}^{l} \|\Delta \mathbf{a}_k\|^2 = \left\|\Delta \begin{pmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_l \end{pmatrix}\right\|^2 \tag{5}$$

where Δ is some finite difference operator. They also propose a surface-shape prior. It is based on the fact that points close in the images are close in space, provided they lie on a continuous surface.

We use those two priors. We measure the closeness of points on the mean shape: $\varphi_{j,g} \stackrel{\text{def}}{=} \rho(d^2(\mathbf{M}_j, \mathbf{M}_g))$, with ρ some localized kernel (we use a truncated gaussian) and write the surface-shape penalty as:

$$\sum_{k=1}^{l}\sum_{j=1}^{m}\sum_{g=1}^{m} \varphi_{j,g}^2 \|\bar{\mathbf{B}}_{k,j} - \bar{\mathbf{B}}_{k,g}\|^2 = \sum_{k=1}^{l} \left\|\Omega \bar{\mathbf{B}}_k\right\|^2, \tag{6}$$

where Ω is a highly sparse matrix depending on the $\varphi_{j,g}$ with three times as many rows as non-zero $\varphi_{j,g}$ and $3m$ columns.

We consider another class of priors that has not been used so far in the literature, on the ordering of the deformation modes. We require mode $l+1$ to express as much of the variance remaining unexplained by the l-mode shape as possible. This naturally leads early modes to explain coarse deformations. This kind of priors is difficult to express in the classical framework where all modes are estimated at once. It however fits very well into the framework of iteratively adding modes of variations, as shown below.

4. Coarse-to-Fine Low-Rank Shape

4.1. Overview

The algorithm we propose is based on recovering the mean shape points $\mathbf{M}_j$, giving a coarse approximation to the true shape, in accordance with the mean shape definition in [15]. Modes are added until some criterion is met.

Most of the other methods estimate all the modes and configuration weights at once. In contrast, our solution tries to embed as much of the variance of the data as possible in the current mode to be estimated, which naturally complies with the mode ordering prior described in the previous section. More precisely, the $l+1$ mode is selected so that the shape minimizes the cost. We thus end up with a series of nested minimization problems. This way of solving the problem has several computational advantages, as is shown later in the paper.

Our algorithm is based on the following relationships steming from the shape model (3):

$$\mathbf{S}^0_{i,j} = \mathsf{D}_i \mathbf{M}_j \tag{7}$$

$$\mathbf{S}^{l+1}_{i,j} = \mathbf{S}^l_{i,j} + a_{i,l+1} b_{l+1,j} \mathsf{D}_i \mathbf{C}_{l+1,j}. \tag{8}$$

We proceed as follow. First, we compute the mean shape points $\mathbf{M}_j$ and the aligning displacements D_i through the 0-mode shape (7). Second, we iteratively triangulate the modes[1], *i.e.* the shapes bases $b_{k,j}\mathbf{C}_{k,j}$ and configuration weights $a_{i,k}$ from (8). A cost function using the reprojection error as data term and the above-mentioned priors is minimized at each step. We stop adding modes when some model complexity selection criterion is met, see §4.4.

4.2. Mean Shape and Aligning Displacements

In order to find the displacements D_i that globally align the deforming object to the world coordinate frame and the mean shape points $\mathbf{M}_j$, we minimize the reprojection error[2] for the 0-mode shape:

$$\min_{\mathbf{M}_1,\ldots,\mathbf{M}_m,\mathsf{D}_1,\ldots,\mathsf{D}_n} \sum_{i=1}^{n}\sum_{j=1}^{m} v_{i,j}\, d^2(\mathbf{q}_{ij}, \mathsf{P}_i \mathsf{D}_i \mathbf{M}_j),$$

which is a calibrated camera instance of the Structure-from-Motion problem, that we solve using standard techniques including bundle adjustment, see *e.g.* [7]. The cameras P_i can either be estimated based on some rigid part in the scene such as the background or be set to some canonical position. We stress that it does not change the result of our algorithm, *i.e.* the estimated deforming surface will be the same whatever the P_i thanks to the D_i.

4.3. Mode Triangulation

The mode triangulation problem is stated as:

$$\min_{\mathbf{a}_{l+1},\bar{\mathbf{B}}_{l+1}} \sum_{i,j} v_{i,j}\, d^2(\mathbf{q}_{i,j}, \mathbf{s}^{l+1}_{i,j}) + \lambda \|\Delta \mathbf{a}_{l+1}\|^2 + \kappa \|\Omega \bar{\mathbf{B}}_{l+1}\|^2. \tag{9}$$

[1] Since the global motion of the object is known at this step, we call 'mode triangulation' the estimation of a mode.

[2] Using a temporal or spatial prior at this stage is not very important since rigid Structure-from-Motion is usually well-posed.

This is a nonlinear least squares optimization problem since *(i)* there is a product between the configuration weights and *(ii)* the modes, and the euclidean distance is used to compare the image points. As in the rigid triangulation case, the euclidean distance can be dealt with an algebraic approximation. The problem however remains nonlinear and difficult to handle in this form since the different views and points are all linked.

First, we drop the priors and compute an initial solution. Second, we refine the complete cost function (9) through nonlinear minimization.

We show that the optimal, *i.e.* reprojection error minimizing, directions in $\bar{\mathbf{C}}_{l+1}$ of the modes can be computed independently from each other and from the other unknowns. We thus split the computation into two main steps. First, we compute the optimal directions in $\bar{\mathbf{C}}_{l+1}$. Second, we compute the optimal configuration weights in a_{l+1} and magnitudes of the modes in b_{l+1}. Each step finds a suboptimal initial solution using linear least squares approximations and refines it by minimizing the reprojection error in a nonlinear manner.

4.3.1 Initializing the Mode Directions in $\bar{\mathbf{C}}_{l+1}$

Splitting the problem. We show how problem (9) can be reformulated on a point-wise basis by estimating independently the direction $\mathbf{C}_{l+1,j}$ of each mode. This is based on casting the reprojection error as a sum of squared point-to-line distances. Substituting equation (3) into (4):

$$\mathbf{s}_{i,j}^{l+1} \sim \underbrace{\mathsf{P}_i \mathsf{D}_i \mathbf{S}_{i,j}^{l}}_{\sim \mathbf{s}_{i,j}^{l}} + a_{i,l+1} b_{l+1,j} \bar{\mathsf{P}}_i \mathcal{R}(\mathsf{D}_i) \bar{\mathbf{C}}_{l+1,j}. \quad (10)$$

This represents an image point parameterized by its position $a_{i,l+1}b_{l+1,j}$ on an image line parameterized by its base point $\mathbf{s}_{i,j}^{l}$ and direction $\bar{\mathsf{P}}_i \mathcal{R}(\mathsf{D}_i)\bar{\mathbf{C}}_{l+1,j}$. By replacing the reprojected points $\mathbf{s}_{i,j}^{l+1}$ from (10) into each reprojection error term in (9), we get:

$$\min_{\mathbf{a}_{l+1}, \mathbf{B}_{l+1}} \sum_{i,j} v_{i,j}\, d^2(\mathbf{q}_{i,j}, \mathbf{s}_{i,j}^{l} + a_{i,l+1} b_{l+1,j} \bar{\mathsf{P}}_i \mathcal{R}(\mathsf{D}_i) \bar{\mathbf{C}}_{l+1,j}).$$

Each term is the squared euclidean distance between an image point $\mathbf{q}_{i,j}$ and the above described point on line. In order to get rid of the offset which depends on the unknown configuration weight $a_{i,l+1}$ and mode magnitude $b_{l+1,j}$, we replace the point-to-point distance d by the point-to-line distance $d_\perp$. This is done by introducing the line coordinates $\mathbf{s}_{i,j}^{l} \times (\bar{\mathsf{P}}_i \mathcal{R}(\mathsf{D}_i)\bar{\mathbf{C}}_{l+1,j})$, giving:

$$\min_{\bar{\mathbf{C}}_{l+1}} \sum_{i=1}^{n} \sum_{j=1}^{m} v_{i,j}\, d_\perp^2(\mathbf{q}_{i,j}, \mathbf{s}_{i,j}^{l} \times (\bar{\mathsf{P}}_i \mathcal{R}(\mathsf{D}_i) \bar{\mathbf{C}}_{l+1,j})).$$

In this reformulated minimization problem, each mode direction $\bar{\mathbf{C}}_{l+1,j}$ in $\bar{\mathbf{C}}_{l+1}$ is independent. It can thus be split as m independent smaller problems:

$$\min_{\bar{\mathbf{C}}_{l+1,j}} \sum_{i=1}^{n} v_{i,j}\, d_\perp^2(\mathbf{q}_{i,j}, \mathbf{s}_{i,j}^{l} \times (\bar{\mathsf{P}}_i \mathcal{R}(\mathsf{D}_i) \bar{\mathbf{C}}_{l+1,j})). \quad (11)$$

Linear estimation. The first step to compute each mode direction $\bar{\mathbf{C}}_{l+1,j}$ is to make a linear least squares approximation to the above stated optimization problem. We approximate the euclidean point-to-line distance by the algebraic one in (2):

$$d_\perp^2(\mathbf{q}_{i,j}, \mathbf{s}_{i,j}^{l} \times (\bar{\mathsf{P}}_i \mathcal{R}(\mathsf{D}_i) \bar{\mathbf{C}}_{l+1,j})) \approx \left(\mathbf{q}_{i,j}^{\mathsf{T}} [\mathbf{s}_{i,j}^{l}]_\times \bar{\mathsf{P}}_i \mathcal{R}(\mathsf{D}_i) \bar{\mathbf{C}}_{l+1,j}\right)^2.$$

The sum over i is minimized to get the initial estimate of $\bar{\mathbf{C}}_{l+1,j}$ with $\|\bar{\mathbf{C}}_{l+1,j}\| = 1$ as required, by the right singular vector corresponding to the smallest singular value of:

$$\mathsf{A} = \begin{pmatrix} v_{1,j}\ \mathbf{q}_{1,j}^{\mathsf{T}} [\mathbf{s}_{1,j}^{l}]_\times \bar{\mathsf{P}}_1 \mathcal{R}(\mathsf{D}_1) \\ \vdots \\ v_{n,j}\ \mathbf{q}_{n,j}^{\mathsf{T}} [\mathbf{s}_{n,j}^{l}]_\times \bar{\mathsf{P}}_n \mathcal{R}(\mathsf{D}_n) \end{pmatrix},$$

where the rows vanishing due to a missing image point (*i.e.* for which $v_{i,j} = 0$) are obviously dropped. The minimum number of image points is $n \geq 2$.

Nonlinear refinement. The second step is to nonlinearly refine the initial estimate of each $\bar{\mathbf{C}}_{l+1,j}$. We minimize the reprojection error using Levenberg-Marquardt. This is very computationally efficient since each of the directions has only 3 parameters and is processed independently. Among the 3 parameters, only 2 are independent, which makes rank-deficient the Jacobian matrix J in the normal equations. This can be dealt with by adding a penalty $(\|\bar{\mathbf{C}}_{l+1,j}\|^2 - 1)^2$ to the error function.

4.3.2 Initializing the Configuration Weights in $\mathbf{a}_{l+1}$ and the Mode Magnitudes in $\mathbf{b}_{l+1}$

Principle. The optimal estimate depends on all the unknown parameters since the image points $\mathbf{s}_{i,j}^{l+1}$ for all views and points depend on $\mathbf{a}_{l+1}\mathbf{b}_{l+1}^{\mathsf{T}}$. We exploit the 1D model ambiguity: we normalize by each of the unknown parameters in $\mathbf{a}_{l+1}$ on turn, making linear the product with the other factor. The results are then combined together.

The constraints. Assume $a_{\zeta,l+1} \neq 0$ for some $\zeta \in 1, \ldots, n$, and define $\mathbf{a}_{l+1}^{\zeta} \stackrel{\text{def}}{=} \frac{\mathbf{a}_{l+1}}{a_{\zeta,l+1}}$ and $\mathbf{b}_{l+1}^{\zeta} \stackrel{\text{def}}{=} a_{\zeta,l+1}\mathbf{b}_{l+1}$. Keeping only the terms related to view ζ in the cost function (9) gives:

$$\min_{\mathbf{b}_{l+1}^{\zeta}} \sum_{j=1}^{m} v_{\zeta,j}\, d^2(\mathbf{q}_{\zeta,j}, \mathbf{s}_{\zeta,j}^{l} + b_{l+1,j}^{\zeta} \bar{\mathsf{P}}_\zeta \mathcal{R}(\mathsf{D}_\zeta) \bar{\mathbf{C}}_{l+1,j}).$$

This minimization problem can be split on a point-wise basis, and is equivalent to solving m 1D problems:

$$\min_{b^{\zeta}_{l+1,j}} v_{\zeta,j}\, d^2(\mathbf{q}_{\zeta,j}, \mathbf{s}^l_{\zeta,j} + b^{\zeta}_{l+1,j}\bar{\mathsf{P}}_{\zeta}\mathcal{R}(\mathsf{D}_{\zeta})\bar{\mathbf{C}}_{l+1,j}).$$

This is a single-view point-on-line triangulation problem, solved by orthogonally projecting $\mathbf{q}_{\zeta,j}$ onto the image line $\mathbf{l}^{l+1}_{\zeta,j} \sim \mathbf{s}^l_{\zeta,j} \times (\bar{\mathsf{P}}_{\zeta}\mathcal{R}(\mathsf{D}_{\zeta})\bar{\mathbf{C}}_{l+1,j})$ to give $b^{\zeta}_{l+1,j}$. The problem can not be solved, however, if $v_{\zeta,j} = 0$, *i.e.* if the point j is not seen in view ζ, and also if the line $\mathbf{l}^{l+1}_{\zeta,j}$ is not well-defined, *i.e.* if $d(\mathbf{s}^l_{\zeta,j}, \bar{\mathsf{P}}_{\zeta}\mathcal{R}(\mathsf{D}_{\zeta})\bar{\mathbf{C}}_{l+1,j}) < \epsilon$, where ϵ is some threshold that we typically choose as few pixels. This problem happens if $\bar{\mathbf{C}}_{l+1,j}$ deforms the point along the viewing ray with respect to camera i.

At this stage, we end up with several, scaled versions $\mathbf{b}^{\zeta}_{l+1}$, $\zeta = 1, \ldots, n$ of $\mathbf{b}_{l+1}$, with missing data, related by $\mathbf{b}^{\zeta}_{l+1} = a_{\zeta,l+1}\mathbf{b}_{l+1}$.

Finding the factors. The $\mathbf{b}^{\zeta}_{l+1}$ vectors must be registered together in order to get the overall sought-after vector $\mathbf{b}_{l+1}$ without holes. This is done by computing the other factor $\mathbf{a}_{l+1}$. The $\mathbf{b}^{\zeta}_{l+1}$ are defined in such a way that $\mathbf{b}^{\zeta}_{l+1}a_{\eta,l+1} - \mathbf{b}^{\eta}_{l+1}a_{\zeta,l+1} = 0$. We solve for $\mathbf{a}_{l+1}$ through:

$$\min_{\mathbf{a}_{l+1}} \sum_{\zeta=1}^{n}\sum_{\eta=1}^{n} \|\mathbf{b}^{\zeta}_{l+1}a_{\eta,l+1} - \mathbf{b}^{\eta}_{l+1}a_{\zeta,l+1}\|^2,$$

which is a linear least squares problem, under the constraint $\|\mathbf{a}_{l+1}\| = 1$. Thanks to $\mathbf{a}_{l+1}$, the $\mathbf{b}^{\zeta}_{l+1}$ are rescaled and averaged to get $\mathbf{b}_{l+1}$.

Another possible way to solve the problem is to consider equation $\mathbf{b}^{\zeta}_{l+1} = a_{\zeta,l+1}\mathbf{b}_{l+1}$. This actually shows that we can formulate the problem as rank-1 matrix factorization with missing data, $\begin{pmatrix}\mathbf{b}^1_{l+1} & \cdots & \mathbf{b}^n_{l+1}\end{pmatrix} \rightarrow \mathbf{b}_{l+1}\mathbf{a}^{\mathsf{T}}_{l+1}$.

4.3.3 Nonlinear Refinement

We have to solve the minimization problem (9). Optimizing over the $\bar{\mathbf{B}}_{l+1,j} = b_{l+1,j}\bar{\mathbf{C}}_{l+1,j}$ directly allows to get rid of the constraints $\|\bar{\mathbf{C}}_{l+1,j}\| = 1$. The issue is that $3m + n$ unknowns must be tuned jointly. Carefully examining the pattern of the Jacobian matrix is thus very important for efficient nonlinear least squares minimization. Indeed, it defines the pattern of the Gauss-Newton approximation to the Hessian matrix, the design matrix in the normal equations to be solved at each iteration of the minimization. The Jacobian has three parts, illustrated for a toy example on figure 1. The first part, related to the data term looks like the one obtained in classical bundle adjustment with well-organized blocks. The second part is related to the temporal prior. Choosing for instance a first order derivative prior gives an $((n-1)\times n)$ Jacobian matrix Δ with ones on the main diagonal and minus ones on the first upper diagonal. The third part depends on the amount of interaction between the points, contained in the $\varphi_{j,g}$ parameters. It typically is very sparse since the localized kernel ρ allows a point to interact with its nearest neightbours only.

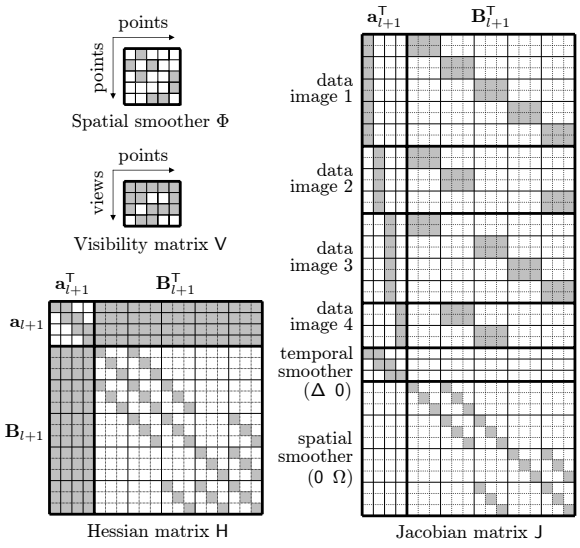

Figure 1. Structure of the Jacobian and Hessian matrices on a toy example with $n = 4$ views and $m = 5$ points.

4.4. A Stopping Criterion

The algorithm we describe in the previous sections is based on iteratively adding modes to the low-rank shape model. A criterion for stopping adding new modes is thus necessary. Each time a mode is added, the number of degrees of freedom of the model grows, making the cost decrease, as is shown in our experimental results. This makes one naturally thinks of using a model selection approach as a stopping criterion. However, the problem at hand does not fulfill the usual model selection assumptions. The first reason is that the number of modes is virtually unlimited: as many modes as desired can be added to the shape model, whereas classical model selection usually operates onto a limited number of models. The second reason is that model selection criteria such as AIC, BIC or GRIC are based on a particular distribution of the residuals, namely a possibly robustified gaussian distribution, see [8, 10]. For the low-rank shape model, the residuals should be interpreted differently. Their dependency on the noise on image point position is very weak. They are mostly due to the deviance of the empirical low-rank shape model from the physics of the actual images. It is difficult to assume any prior distribution for this deviance.

We propose to use Cross-Validation as a criterion for selecting the number of modes. The idea is to partition the

data in a training and a test set, and average the test error over several such partitions. This approach, which has rarely been used for geometric model selection in computer vision, does not require a specific known distribution of the residuals, and directly reflects the ability of the model to extrapolate to new data. More precisely, we use u-fold Cross-Validation, which splits the data into u subsets or 'folds'. Typical values for u range from 3 to 10. We use $u = 4$ in our experiments, and split the data image-point-wise: each fold is a subset of image points, and must allow the algorithm to reconstruct all views and points (for instance, we do not remove all image points in a single view). The test error is obtained by comparing the test dataset with its prediction.

The typical behaviour of the Cross-Validation score is to decrease until the optimal number of modes is reached, and then to increase. It first decreases since the model with not enough modes is too restrictive to explain well the data and thus can not make good predictions. It then increases since with more modes than enough, the model fits unwanted effects in the data, *i.e.* it is too flexible to predict new data. This typical behaviour is however not what we observe when the priors are used. In this case, the Cross-Validation score decreases rapidly until the optimal number of modes is reached, and then remains steady. This is explained by the fact that the priors inhibitate the degrees of freedom of the extra modes, as also reported in [11]. Our stopping criterion has two parts: we stop adding modes when either the Cross-Validation score increases or when its decrease is below some threshold, that we choose as $\varepsilon = 10^{-4}$ in our experiments.

Computing the Cross-Validation score requires to fit the new mode to each of the u training sets. For that purpose, and for computational efficiency, we keep $u + 1$ models: the u models which use the folds as training set, and the one which uses all the data.

OBJECTIVE

Given a set of corresponding image points $\mathbf{q}_{i,j}$ on a deforming object and cameras $\mathsf{P}_i \sim \mathsf{K}_i(\mathrm{I}\ \mathbf{0})\mathsf{E}_i$ obtained by some means, compute globally aligning displacements $\mathsf{D}_i \in SE(3)$ for each frame i and a set of frame-varying, low-rank 3D shapes $\mathbf{S}^l_{i,j}$ in a coarse-to-fine manner, *i.e.* the cost for $\mathbf{S}^{l+1}_{i,j}$ is lower than for $\mathbf{S}^l_{i,j}$. The number of modes l is estimated using Cross-Validation (CV): each computation is carried out over u randomly selected folds to compute the CV score $\mathcal{G}^l$.

ALGORITHM

Mean Shape and Aligning Displacement Computation

1. (§4.2) Run calibrated camera Structure-from-Motion with the image points $\mathbf{q}_{i,j}$ as inputs and intrinsic parameters K_i giving new cameras $\mathsf{K}_i(\mathrm{I}\ \mathbf{0})\mathsf{A}_i$ and mean shape points $\mathbf{M}_j$
2. Set the aligning displacements $\mathsf{D}_i \leftarrow \mathsf{E}_i^{-1}\mathsf{A}_i$
3. (§4.4) Compute the CV score $\mathcal{G}^0$, and set $l \leftarrow 0$
4. Initialize the shape estimate with the mean shape for every frame: $\mathbf{S}_{i,j} \leftarrow \mathbf{M}_j$

Iterative Mode Triangulation

1. (§4.3.1) Initialize the mode directions $\bar{\mathbf{C}}_{l+1,j}$
2. (§4.3.2) Compute the configuration weights $a_{i,l+1}$ and mode magnitudes $b_{i,l+1}$
3. (§4.3.3) Nonlinear refinement: minimize the reprojection error over the modes and configuration weights
4. (§4.4) Compute the CV score $\mathcal{G}^{l+1}$
5. (§4.4) Stop if $\mathcal{G}^{l+1} \geq \mathcal{G}^l$ or $\mathcal{G}^l - \mathcal{G}^{l+1} \leq \varepsilon$
6. Update the 3D shape: $\mathbf{S}^{l+1}_{i,j} \leftarrow \mathbf{S}^l_{i,j} + a_{i,l+1}b_{l+1,j}\mathbf{C}_{l+1,j}$
7. Set $l \leftarrow l + 1$ and loop to step 1

Table 1. Overview of our coarse-to-fine (C2F) low-rank Structure-from-Motion algorithm. The priors are taken into account at step 3 of mode triangulation.

5. Experimental Results

We provide experimental results on simulated and real data. For each dataset, we compare our algorithm with the one by Torresani *et al.* which is shown in [11] to give the best results compared to other methods in the literature. We name it TORRESANI. Our algorithm is summarized in table 1. We use two variants: C2F - NO PRIOR which does not use the two smoothness priors, and C2F - PRIORS which uses them.

We did not encounter any local minimum in the Cross-Validation score in our experiments.

5.1. Simulated Data

We have two data generation models. The first one is the Candide face model [2]. The second one is the shark sequence available from the authors of [11]. We found that the CMU mocap datasets were either close to rigid or not 'homogeneous' enough for the low-rank shape model. For each dataset, we measure the reprojection error, the Cross-Validation score and the 3D error as functions of the number of modes, the amount of missing data and the number of images. The graphs we show are for the Candide face model – similar results as obtained for the shark sequence. The default setup is $n = 10$ images and $m = 113$ points.

The first set of experiments is illustrated on figure 2 (left and middle). It is meant to assess if Cross-Validation effectively gives a sensible way of selecting the number of modes. We observe that our C2F - NO PRIOR is very sensitive to an overestimated number of modes: with more than 2 modes, the 3D error grows rapidly, while both C2F - PRIORS and TORRESANI remains stable. The Cross-Validation

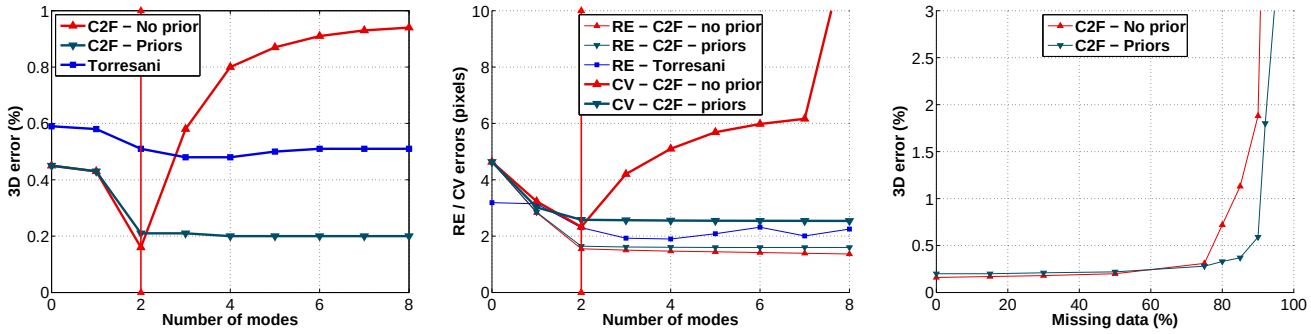

Figure 2. (left) The 3D error as a function of the number of modes. (middle) The reprojection error (RE) and Cross-Validation score (CV) as functions of the number of modes. (right) The 3D error as a function of the percentage of missing data. The vertical bars show minima of the CV score and 3D error curves.

score behaves similarly to the 3D error. In particular, it allows us selecting the optimal number of modes with respect to the 3D error for our C2F - NO PRIOR while for our C2F - PRIORS, the number of modes is slightly overestimated, which does not degrade the quality of the 3D shape, as already observed for TORRESANI in [11]. As expected, the reprojection error decreases as the number of modes increases.

The second set of experiments, shown on figure 2 (right) shows how the algorithms behave against the amount of missing data. Our C2F - PRIORS recovers the 3D shape with up to more than 92% missing data. Thanks to the good behavior of the Cross-Validation score, which allows our C2F - NO PRIOR selecting a sensible number of modes, even with no prior, it handles up to 90% missing data. As for TORRESANI it diverges in most cases.

The third set of experiments computes the success rate of the selected number of modes for C2F - NO PRIOR with the Cross-Validation score. The success rate is 94%, 89% and 88% for respectively no missing data, 25% and 50% missing data. This is very satisfying since in most failures, the number of modes is mis-estimated by only 1.

The fourth set of experiments compares the behaviour of the algorithms with respect to the number of points and views. The graphs are not shown here due to lack of space. As expected, the smaller the number of points or views, the smaller the reprojection error, and the larger the 3D error and Cross-Validation score.

5.2. Real Data

The paper dataset. This video has 203 images of size 720×576. We used a direct, *i.e.* intensity based, approach to recover the parameters of a Free-Form Deformation (FFD) that provided us with 140 point correspondences. Figure 3 shows the results we obtained. Our C2F - NO PRIOR and C2F - PRIORS selected 0 mode and 3 modes and reached 7.10 and 0.84 pixels of reprojection error respectively. C2F - NO PRIOR thus performs very badly for this sequence, giving a very distorted 3D shape. This shows that using the priors can not be avoided, since C2F - PRIORS gives good results, with 1.18 pixels for the Cross-Validation score, showing good predictivity.

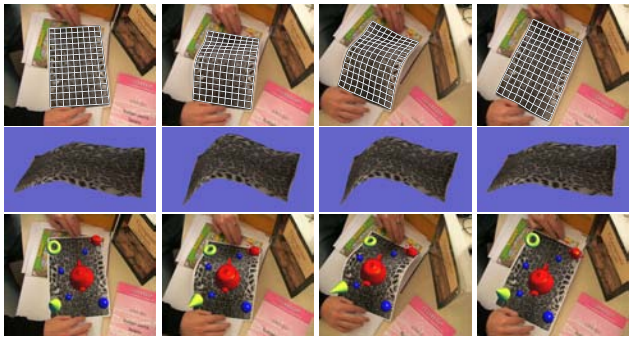

Figure 3. The paper dataset. (first row) Some of the images with the FFD mesh we track. (second row) New view synthesis with the reconstructed surface. (third row) The augmented images.

We then simulated an occlusion by removing 24 points on 120 images, *i.e.* slightly more than 10% of the data. C2F - PRIORS selected 3 modes, and reached 1.44 pixels of reprojection error and 1.82 pixels for the Cross-Validation score, which, although slightly higher than in the full data case, is reasonable.

The face dataset. We extracted a 100 image, 624×352, video of Gabrielle Solis from the series "Desperate Housewives", and ran a 2D Active Appearance Model (AAM) to track her face. We then reconstructed the camera and the 68 vertices of the AAM with our algorithm. Figure 4 shows the result. Both C2F - PRIORS and C2F - NO PRIOR found that 4 modes are required. They respectively obtained 0.91 and 0.82 pixels for the reprojection error, and 1.15 and 1.22 pixels for the Cross-Validation score. These values show that the reconstructed model has a good predictivity. We stress

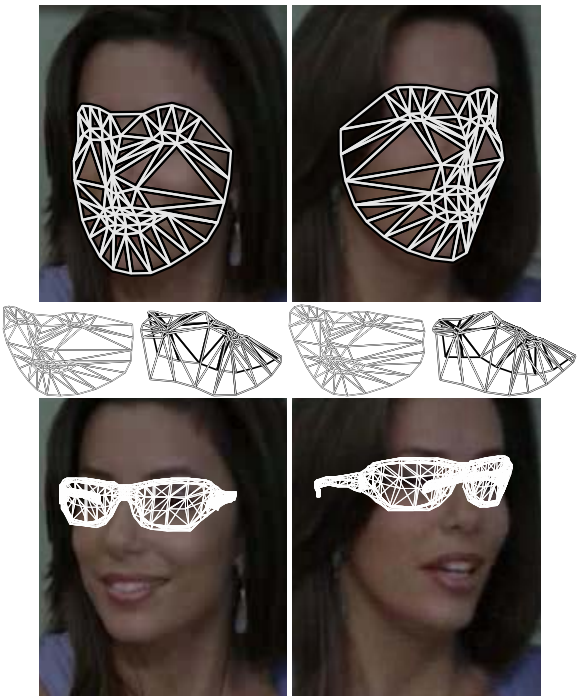

Figure 4. The face dataset. (top) Two out of the 100 images overlaid with the face AAM used for tracking. (middle) The reconstructed AAM vertices. (bottom) The augmented images.

that the a priori knowledge that a face is in the images is used only at the tracking step: our method reconstructs the deforming structure in a generic manner.

6. Conclusion

We proposed a method that allows reconstructing a new coarse-to-fine low-rank shape model of a deforming object from a single video. Our method handles missing data, uses the full perspective camera model and automatically selects the optimal number of deformation modes by Cross-Validating the model. Experimental results on simulated data show that the automatically selected number of modes corresponds to the minimal 3D error. We use two smoothness priors which are shown to improve the quality of the reconstruction. Our method outperforms previous ones in terms of accuracy. The main statement we make is that Cross-Validation is a sensible way of assessing the number of modes in the model in that it looks similar to the 3D error.

An open research topic is the one of automatically selecting the weighting parameters for the priors. Most of the authors reports heuristic means or uses trial and error, as we did in our experiments. A possible solution is to minimize the Cross-Validation score over the weighting parameters. It is not clear if it can be done in a reasonable amount of time, though.

Acknowledgments. We would like to thank Mathieu Perriollat for his help on new view synthesis.

References

[1] H. Aanæs and F. Kahl. Estimation of deformable structure and motion. In *Proceedings of the Vision and Modelling of Dynamic Scenes Workshop*, 2002. 2, 3

[2] J. Ahlberg. CANDIDE-3 – an updated parameterized face. Technical report, Dept. of Electrical Engineering, Linkping University, Sweden, 2001. 6

[3] A. Bartoli and S. Olsen. A batch algorithm for implicit non-rigid shape and motion recovery. In *Proceedings of the Workshop on Dynamical Vision at ICCV*, 2005. 2

[4] M. Brand. A direct method for 3D factorization of non-rigid motion observed in 2D. In *International Conference on Computer Vision and Pattern Recognition*, 2005. 1, 2

[5] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *International Conference on Computer Vision and Pattern Recognition*, 2000. 1, 2

[6] A. D. Bue, X. Lladó, and L. Agapito. Non-rigid metric shape and motion recovery from uncalibrated images using priors. In *International Conference on Computer Vision and Pattern Recognition*, 2006. 2, 3

[7] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. Second Edition. 2, 3

[8] K. Kanatani. Geometric information criterion for model selection. *International Journal of Computer Vision*, 26(3):171–189, 1998. 5

[9] S. Olsen and A. Bartoli. Using priors for improving generalization in non-rigid structure-from-motion. In *British Machine Vision Conference*, 2007. 2, 3

[10] P. H. S. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision*, 50(1):27–45, 2002. 5

[11] L. Torresani, A. Hertzmann, and C. Bregler. Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 2007. To appear. 1, 3, 6, 7

[12] R. Vidal and D. Abretske. Nonrigid shape and motion from multiple perspective views. In *European Conference on Computer Vision*, 2006. 2

[13] J. Xiao and T. Kanade. A linear closed-form solution to non-rigid shape and motion recovery. *International Journal of Computer Vision*, 67(2):233–246, March 2006. 1, 2

[14] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *European Conference on Computer Vision*, 2006. 2

[15] A. J. Yezzi and S. Soatto. Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images. *International Journal of Computer Vision*, 53(2):153–167, March 2003. 1, 3