Appeared in Proceedings of EMMCVPR'05

Handling Missing Data in the Computation of 3D Affine Transformations

H. Martinsson[1], A. Bartoli[2], F. Gaspard[1], and J-M. Lavest[2]

[1] CEA LIST – LIST/DTSI/SARC/LCEI Bât 528, 91 191 Gif sur Yvette, France
tel: +33(0)1 69 08 82 98, fax: +33(0)1 69 08 83 95
hanna.martinsson@cea.fr

[2] LASMEA (CNRS / UBP) – 24 avenue des Landais, 63 177 Aubière, France
tel: +33(0)4 73 40 76 61, fax: +33(0)4 73 40 72 62
adrien.bartoli@univ-bpclermont.fr

Abstract. The reconstruction of rigid scenes from multiple images is a central topic in computer vision. Approaches merging partial 3D models in a hierarchical manner have proven the most effective to deal with large image sequences. One of the key building blocks of these hierarchical approaches is the alignment of two partial 3D models, which requires to express them in the same 3D coordinate frame by computing a 3D transformation. This problem has been well-studied for the cases of 3D models obtained with calibrated or uncalibrated pinhole cameras.

We tackle the problem of aligning 3D models – sets of 3D points – obtained using uncalibrated affine cameras. This requires to estimate 3D affine transformations between the models. We propose a factorization-based algorithm estimating simultaneously the aligning transformations and corrected points, exactly matching the estimated transformations, such that the reprojection error over all cameras is minimized. In the case of incomplete image data our algorithm uses an Expectation Maximization (EM) based scheme that alternates prediction of the missing data and estimation of the affine transformation.

We experimentally compare our algorithm to other methods using simulated and real data.

1 Introduction

Threedimensional reconstruction from multiple images of a rigid scene, often dubbed Structure-From-Motion (SFM), is one of the most studied problems in computer vision. The difficulties come from the fact that, using only feature correspondences, both the 3D structure of the scene and the cameras have to be computed. Most approaches rely on an initialisation phase optionally followed by self-calibration and bundle adjustment. Existing initialisation algorithms can be divided into three families, namely *batch*, *sequential* and *hierarchical* processes. Hierarchical processes [1] have proven the most successful for large image sequences. Indeed, batch processes such as the factorization algorithms [2] which reconstruct all features and cameras in a single computation step, do not easily handle occlusions, while sequential processes such as [3] which reconstruct each view on turn, may typically suffer from accumulation of the errors. Hierarchical processes merge partial 3D models obtained from sub-sequences, which allows to distribute the error over the sequence, and efficiently handle open and closed sequences. A key step of hierarchical processes is the fusion or the *alignment* of partial 3D models, which is done by *computing 3D motion from 3D feature correspondences*. This problem has been extensively studied in the projective [4,1] and the metric and Euclidean [5] cases.

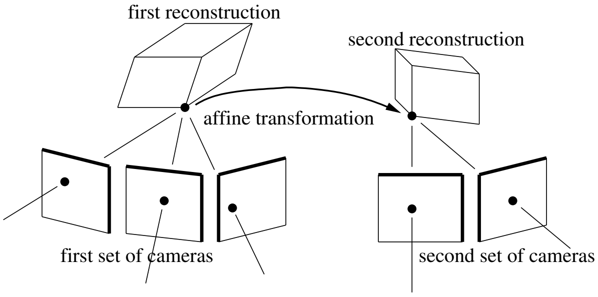


Fig. 1. The problem tackled in this paper is the Maximum Likelihood Estimation of 3D affine transformations between two affine reconstructions obtained from uncalibrated affine cameras.

We focus on the affine camera model [6], which is a reasonable approximation to the perspective camera model when the depth of the observed scene is small compared to the viewing distance. In this case, the partial 3D models obtained from sub-sequences, *i.e.* multiple subsets of cameras, are related by 3D affine transformations. We deal with the computation of such transformations from point correspondences, as illustrated on Fig. 1. We propose a Maximum Likelihood Estimator based on factorizing modified image point coordinates. We compute a 3D affine transformation and a set of 3D point correspondences which perfectly match, such that *the reprojection error in all sets of cameras is minimized*. It is intended to fit in hierarchical affine SFM processes of which the basic reconstruction block is, *e.g.* the affine factorization [2]. Our method does not make any assumption about the cameras, besides the fact that a reconstruction of each camera set using an affine camera model has been performed. The method relies on the important new concept of *orthonormal bases*. In the occlusion-free case, our algorithm needs one Singular Value Decomposition (SVD). However, in the case of incomplete measurement data, *i.e.* when some of the 3D points used for the alignment are not visible in all views, the factorization algorithm must be extended. We propose an Expectation-Maximization (EM) based scheme. The Expectation step predicts the missing data while the Maximization step maximizes the log likelihood.

We proposed the Maximum Likelihood Estimator in the case of complete data in [7]. The contribution of this paper with respect to the former one resides in the handling of missing data. We have also completed the experiments.

This paper is organized as follows. We give our notation and preliminaries in Sect. 2. In Sect. 3, we review the factorization approach to uncalibrated affine Structure-From-Motion. Our alignment method is described in Sect. 4, while other methods are summarized in Sect. 5. Experimental results are reported in Sect. 6. Our conclusions are given in Sect. 7.

2 Notation and Preliminaries

Vectors are typeset using bold fonts, *e.g.* $\mathbf{x}$, and matrices using sans-serif, calligraphic and greek fonts, *e.g.* A, $\mathcal{Q}$ and Λ. We do not use homogeneous coordinates, *i.e.* image point coordinates are 2-vectors: $\mathbf{x}^\mathsf{T} = (x \ y)$, where $^\mathsf{T}$ is transposition. The different sets of cameras are indicated with primes, *e.g.* P_1 and P'_1 are the first cameras of the camera sets. Index $i = 1 \ldots n$ is used for the cameras of a camera set and index $j = 1 \ldots m$ is used for the 3D points. The mean vector of a set of vectors, say $\{\mathbf{Q}_j\}$, is denoted $\bar{\mathbf{Q}}$. The Moore-Penrose pseudoinverse of matrix A is denoted $\mathsf{A}^\dagger$.

Let $\mathbf{Q}_j$ be a 3-vector and $\mathbf{x}_{ij}$ a 2-vector representing respectively a 3D and an image point. The uncalibrated affine camera is modeled by a (2×3) matrix P_i and a (2×1) translation vector $\mathbf{t}_i$, giving the projection equation

$$\mathbf{x}_{ij} = \mathsf{P}_i \mathbf{Q}_j + \mathbf{t}_i \ . \tag{1}$$

Calligraphic fonts are used for the measurement matrices, *e.g.*

$$\mathcal{X}_{(2n \times m)} = \begin{pmatrix} \mathcal{Y}_1 & \cdots & \mathcal{Y}_m \end{pmatrix} \qquad \text{and} \qquad \mathcal{Y}_j = \begin{pmatrix} \mathbf{x}_{1j}{}^\mathsf{T} & \cdots & \mathbf{x}_{nj}{}^\mathsf{T} \end{pmatrix}^\mathsf{T} \ ,$$

where $\mathcal{Y}_j$ contains all the measured image coordinates for the j-th point. The $(2n \times 3)$ 'joint projection' and $(3 \times m)$ 'joint structure' matrices are defined by

$$\mathcal{P} = \begin{pmatrix} \mathsf{P}_1{}^\mathsf{T} & \cdots & \mathsf{P}_n{}^\mathsf{T} \end{pmatrix}^\mathsf{T} \qquad \text{and} \qquad \mathcal{Q} = \begin{pmatrix} \mathbf{Q}_1 & \cdots & \mathbf{Q}_m \end{pmatrix} \ .$$

We assume that the noise on the image point positions has a Gaussian centered distribution and is i.i.d. Under these hypotheses, minimizing the reprojection error yields Maximum Likelihood Estimates.

3 Structure-From-Motion Using Factorization

Given a set of point matches $\{\mathbf{x}_{ij}\}$, the factorization algorithm is employed to recover all cameras $\{\hat{\mathsf{P}}_i, \hat{\mathbf{t}}_i\}$ and 3D points $\{\hat{\mathbf{Q}}_j\}$ at once [2]. Under the aforementioned hypotheses on the noise distribution, this algorithm computes Maximum Likelihood Estimates [8] by minimizing the reprojection error

$$\min_{\hat{\mathcal{P}}, \hat{\mathcal{Q}}, \{\hat{\mathbf{t}}_i\}} \mathcal{R}^2(\hat{\mathcal{P}}, \hat{\mathcal{Q}}, \{\hat{\mathbf{t}}_i\}) \quad \text{with} \quad \mathcal{R}^2(\mathcal{P}, \mathcal{Q}, \{\mathbf{t}_i\}) = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} d^2(\mathbf{x}_{ij}, \mathsf{P}_i \mathbf{Q}_j + \mathbf{t}_i) \ , \tag{2}$$

where $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$.

Step 1: Computing the translation. Given the uncalibrated affine projection (1), the first step of the algorithm is to compute the translation $\hat{\mathbf{t}}_i$ of each camera in order to cancel it out from the projection equation. This is achieved by nullifying the partial derivatives of the reprojection error (2) with respect to $\hat{\mathbf{t}}_i$: $\frac{\partial \mathcal{R}^2}{\partial \hat{\mathbf{t}}_i} = 0$. A short calculation shows that if we fix the arbitrary centroid of the 3D points to the origin, then $\hat{\mathbf{t}}_i = \bar{\mathbf{x}}_i$. Each set of image points is therefore centered on its centroid, *i.e.* $\mathbf{x}_{ij} \leftarrow \mathbf{x}_{ij} - \bar{\mathbf{x}}_i$, to obtain *centered coordinates*: $\mathbf{x}_{ij} = \mathsf{P}_i \mathbf{Q}_j$.

Step 2: Factorizing. The problem is reformulated as

$$\min_{\hat{\mathcal{P}}, \hat{\mathcal{Q}}} \mathcal{R}^2(\hat{\mathcal{P}}, \hat{\mathcal{Q}}) \qquad \text{with} \qquad \mathcal{R}^2(\mathcal{P}, \mathcal{Q}) = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} d^2(\mathbf{x}_{ij}, \mathsf{P}_i \mathbf{Q}_j) \ .$$

The reprojection error can be rewritten by gathering the terms using the measurement, the 'joint projection' and the 'joint structure' matrices as

$$\mathcal{R}^2(\mathcal{P}, \mathcal{Q}) \propto \|\mathcal{X} - \mathcal{P}\mathcal{Q}\|^2 \ ,$$

and the problem is solved by computing the Singular Value Decomposition [9] of matrix $\mathcal{X}$, $\mathcal{X}_{2n \times m} = \mathsf{U}_{2n \times m} \Sigma_{m \times m} \mathsf{V}_{m \times m}^{\mathsf{T}}$. Let $\Sigma = \Sigma_u \Sigma_v$ be any decomposition of matrix Σ. The motion and structure are obtained by 'truncating' the decomposition or nullifying all but the 3 first singular values, which leads to

$$\mathcal{P} = \psi(\mathsf{U}\Sigma_u) \qquad \text{and} \qquad \mathcal{Q} = \psi^{\mathsf{T}}(\mathsf{V}\Sigma_v^{\mathsf{T}}) \ ,$$

where $\psi(\mathsf{W})$ returns the matrix formed with the 3 leading columns of matrix W. Note that the solution $\mathcal{P} = \psi(\mathsf{U})$ and $\mathcal{Q} = \psi^{\mathsf{T}}(\mathsf{V}\Sigma)$ has the property $\mathcal{P}^{\mathsf{T}}\mathcal{P} = \mathrm{I}$, which is useful for our alignment method, see Sect. 4.

The 3D model is obtained up to a global affine transformation. Indeed, for any (3×3) invertible matrix B,

$$\tilde{\mathcal{P}} = \hat{\mathcal{P}}\mathsf{B} \qquad \text{and} \qquad \tilde{\mathcal{Q}} = \mathsf{B}^{-1}\hat{\mathcal{Q}} \tag{3}$$

give the same reprojection error that $\mathcal{P}$ and $\mathcal{Q}$ since $\mathcal{R}^2(\tilde{\mathcal{P}}, \tilde{\mathcal{Q}}) = \|\mathcal{X} - \tilde{\mathcal{P}}\tilde{\mathcal{Q}}\| = \|\mathcal{X} - \hat{\mathcal{P}}\mathsf{B}\mathsf{B}^{-1}\hat{\mathcal{Q}}\|^2 = \|\mathcal{X} - \mathcal{P}\mathcal{Q}\|^2 = \mathcal{R}^2(\mathcal{P}, \mathcal{Q})$.

As presented above, the factorization algorithm do not handle occlusions. Though some algorithms have been proposed, see *e.g.* [10], they are not appropriate for Structure-From-Motion from large image sequences.

4 Alignment of 3D Affine Reconstructions

We formally state the alignment problem in the two camera set case and present our algorithm, dubbed 'FactMLE-EM'.

4.1 Problem Statement

Consider two sets of cameras $\{(\mathsf{P}_i, \mathbf{t}_i)\}_{i=1}^{n}$ and $\{(\mathsf{P}'_i, \mathbf{t}'_i)\}_{i=1}^{n'}$ and associated structures $\{\mathbf{Q}_j \leftrightarrow \mathbf{Q}'_j\}_{j=1}^{m}$ obtained by reconstructing a rigid scene using *e.g.* the above-described factorization algorithm. Without loss of generality, we take $n = n'$ and the reprojection error over two sets is given by

$$\mathcal{C}^2(\mathcal{Q}, \mathcal{Q}') = \frac{1}{2nm} \left(\mathcal{R}^2(\mathcal{P}, \mathcal{Q}, \{\mathbf{t}_i\}) + \mathcal{R}'^2(\mathcal{P}', \mathcal{Q}', \{\mathbf{t}'_i\}) \right) \ . \tag{4}$$

Letting $(\hat{\mathsf{A}}, \hat{\mathbf{t}})$ represent the aligning (3×3) affine transformation, the Maximum Likelihood Estimator is formulated by

$$\min_{\hat{\mathcal{Q}}, \hat{\mathcal{Q}}'} \mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \qquad \text{s.t.} \qquad \hat{\mathbf{Q}}'_j = \hat{\mathsf{A}} \hat{\mathbf{Q}}_j + \hat{\mathbf{t}} \ . \tag{5}$$

4.2 A Factorization-Based Algorithm

Our method to solve problem (5) uses a three-step factorization strategy. We first describe it in the occlusion-free case and then propose an iterative extension for the missing data case.

Step 1: Orthonormalizing. We propose the important concept of *orthonormal bases.* We define a reconstruction to be in an orthonormal basis if the joint projection matrix is column-orthonormal. Given a joint projection matrix $\mathcal{P}$, one can find a 3D affine tranformation represented by the (3×3) matrix N, which applies as B in (3), such that $\mathcal{P}\mathsf{N}$ is column-orthonormal, *i.e.* such that $\mathsf{N}^\mathsf{T} \mathcal{P}^\mathsf{T} \mathcal{P} \mathsf{N} = \mathrm{I}_{(3\times 3)}$. We call the transformation N an *orthonormalizing transformation.* The set of orthonormalizing tranformations is 3-dimensional since for any 3D rotation matrix U, NU still is an orthonormalizing transformation for $\mathcal{P}$. We use the QR decomposition $\mathcal{P} = \mathsf{QR}$, see *e.g.* [9], giving an upper triangular orthonormalizing transformation $\mathsf{N} = \mathsf{R}^{-1}$. Other choices are possible for computing an N, *e.g.* if $\mathcal{P} = \mathsf{U}\Sigma\mathsf{V}^\mathsf{T}$ is an SVD of $\mathcal{P}$, then $\mathsf{N} = \mathsf{V}\Sigma^{-1}$ has the required property. Henceforth, we assume that all 3D models are expressed in orthonormal bases

$$\begin{cases} \mathcal{P} \leftarrow \mathcal{P}\mathsf{N} \\ \mathcal{P}' \leftarrow \mathcal{P}'\mathsf{N}' \end{cases} \quad \text{and} \quad \begin{cases} \mathcal{Q} \leftarrow \mathsf{N}^{-1}\mathcal{Q} \\ \mathcal{Q}' \leftarrow \mathsf{N}'^{-1}\mathcal{Q}' \end{cases} .$$

An interesting property of orthonormal bases is that $\mathcal{P}^\dagger = \mathcal{P}^\mathsf{T}$. Hence, triangulating points in these bases is simply done by $\mathcal{Q} = \mathcal{P}^\mathsf{T}\mathcal{X}$.

Note that the matrix $\mathcal{P}$ computed by factorization, see Sect. 3, may already satisfy $\mathcal{P}^\mathsf{T}\mathcal{P} = \mathrm{I}$. However, if at least one of the cameras is not used for the alignment, *e.g.* if none of the 3D point correspondences project in this camera, or if the cameras come as the result of the alignment of partial 3D models, then $\mathcal{P}$ will *not* satisfy $\mathcal{P}^\mathsf{T}\mathcal{P} = \mathrm{I}$, thus requiring the orthonormalization step.

Step 2: Eliminating the translation. The translation part of the sought-after transformation can not be computed directly, but can be eliminated from the equations. First, center the image points to eliminate the translation part of the cameras: $\mathbf{x}_{ij} \leftarrow \mathbf{x}_{ij} - \mathbf{t}_i$ and $\mathbf{x}'_{ij} \leftarrow \mathbf{x}'_{ij} - \mathbf{t}'_i$. Second, consider that the partial derivatives of the reprojection error (4) with respect to $\hat{\mathbf{t}}$ must vanish: $\frac{\partial \mathcal{C}^2}{\partial \hat{\mathbf{t}}} = 0$. By using the constraint $\hat{\mathbf{Q}}'_j = \hat{\mathsf{A}}\hat{\mathbf{Q}}_j + \hat{\mathbf{t}}$ from (4) and expanding using (4), we get

$$\begin{aligned}
\sum_{i=1}^{n'}\sum_{j=1}^{m}\left(\mathsf{P}'^{\mathsf{T}}_i\mathsf{P}'_i\hat{\mathbf{t}} - \mathsf{P}'^{\mathsf{T}}_i\mathbf{x}'_{ij} + \mathsf{P}'^{\mathsf{T}}_i\mathsf{P}'_i\hat{\mathsf{A}}\hat{\mathbf{Q}}_j\right) &= 0\\
\sum_{j=1}^{m}\left(\mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathbf{t}} - \mathcal{P}'^{\mathsf{T}}\mathcal{Y}'_j + \mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathsf{A}}\hat{\mathbf{Q}}_j\right) &= 0\\
m\mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathbf{t}} - m\mathcal{P}'^{\mathsf{T}}\bar{\mathcal{Y}}' + m\mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathsf{A}}\bar{\hat{\mathcal{Q}}} &= 0 \ ,
\end{aligned}$$

which leaves us with $\hat{\mathbf{t}} = (\mathcal{P}'^{\mathsf{T}}\mathcal{P}')^{-1}(\mathcal{P}'^{\mathsf{T}}\bar{\mathcal{Y}}' - \mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathsf{A}}\bar{\hat{\mathcal{Q}}})$ that, thanks to the orthonormal basis property $\mathcal{P}'^{\dagger} = \mathcal{P}'^{\mathsf{T}}$, further simplifies to

$$\hat{\mathbf{t}} = \mathcal{P}'^{\mathsf{T}}\bar{\mathcal{Y}}' - \hat{\mathsf{A}}\bar{\hat{\mathcal{Q}}} \ . \tag{6}$$

Note that if the same entire sets of reconstructed points are used for the alignment, then we directly obtain $\hat{\mathbf{t}} = \mathbf{0}$ since $\bar{\mathcal{Y}}' = \mathbf{0}$ and $\bar{\hat{\mathcal{Q}}} = \mathbf{0}$. This is rarely the case in practice, especially if the alignment is used to merge partial 3D models.

Third, consider that the m partial derivatives of the reprojection error (4) with respect to each $\hat{\mathbf{Q}}_j$ must vanish as well: $\frac{\partial \mathcal{C}^2}{\partial \hat{\mathbf{Q}}_j} = 0$, and expand as above

$$\begin{aligned}
\sum_{i=1}^{n}\left(\mathsf{P}^{\mathsf{T}}_i\mathsf{P}_i\hat{\mathbf{Q}}_j - \mathsf{P}^{\mathsf{T}}_i\mathbf{x}_{ij}\right) + \sum_{i=1}^{n'}\left(\hat{\mathsf{A}}^{\mathsf{T}}\mathsf{P}'^{\mathsf{T}}_i\mathsf{P}'_i\hat{\mathsf{A}}\hat{\mathbf{Q}}_j - \hat{\mathsf{A}}^{\mathsf{T}}\mathsf{P}'^{\mathsf{T}}_i\mathbf{x}'_{ij} + \hat{\mathsf{A}}^{\mathsf{T}}\mathsf{P}'^{\mathsf{T}}_i\mathsf{P}'_i\hat{\mathbf{t}}\right) &= 0\\
\mathcal{P}^{\mathsf{T}}\mathcal{P}\hat{\mathbf{Q}}_j - \mathcal{P}^{\mathsf{T}}\mathcal{Y}_j + \hat{\mathsf{A}}^{\mathsf{T}}\mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathsf{A}}\hat{\mathbf{Q}}_j - \hat{\mathsf{A}}^{\mathsf{T}}\mathcal{P}'^{\mathsf{T}}\mathcal{Y}'_j + \hat{\mathsf{A}}^{\mathsf{T}}\mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathbf{t}} &= 0 \ .
\end{aligned}$$

The sum over j of all these derivatives also vanishes, giving

$$\mathcal{P}^{\mathsf{T}}\mathcal{P}\bar{\hat{\mathcal{Q}}} - \mathcal{P}^{\mathsf{T}}\bar{\mathcal{Y}} + \hat{\mathsf{A}}^{\mathsf{T}}\mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathsf{A}}\bar{\hat{\mathcal{Q}}} - \hat{\mathsf{A}}^{\mathsf{T}}\mathcal{P}'^{\mathsf{T}}\bar{\mathcal{Y}}' + \hat{\mathsf{A}}^{\mathsf{T}}\mathcal{P}'^{\mathsf{T}}\mathcal{P}'\hat{\mathbf{t}} = 0 \ .$$

By replacing $\hat{\mathbf{t}}$ by its expression (6), and after some minor algebraic manipulations, we obtain

$$\mathcal{P}^{\mathsf{T}}\mathcal{P}\bar{\hat{\mathcal{Q}}} - \mathcal{P}^{\mathsf{T}}\bar{\mathcal{Y}} = 0 \quad \Longrightarrow \quad \bar{\hat{\mathcal{Q}}} = \mathcal{P}^{\dagger}\bar{\mathcal{Y}} \tag{7}$$

and by substituting in (6) and using the orthonormal basis property, we get

$$\hat{\mathbf{t}} = \mathcal{P}'^{\mathsf{T}}\bar{\mathcal{Y}}' - \hat{\mathsf{A}}\mathcal{P}^{\mathsf{T}}\bar{\mathcal{Y}} \ . \tag{8}$$

It is common in factorization methods to center the data with respect to their centroid to cancel the translation part of the transformation. Equation (8) means that the data must be centered with respect to the *reconstructed centroid* of the image points, not with respect to the actual 3D centroid.

Obviously, if the 3D models have been obtained by the factorization method of Sect. 3, then the centroid of the 3D points corresponds to the reconstructed centroid, *i.e.* $\bar{\mathbf{Q}} = \mathcal{P}^{\mathsf{T}}\bar{\mathcal{Y}}$ and $\bar{\mathbf{Q}}' = \mathcal{P}'^{\mathsf{T}}\bar{\mathcal{Y}}'$, provided that the same sets of views are used for reconstruction and alignment.

To summarize, we cancel the translation part out of the sought-after transformation by translating the reconstructions and the image points as shown below

$$\begin{cases} \mathbf{Q}_j \leftarrow \mathbf{Q}_j - \mathcal{P}^{\mathsf{T}}\bar{\mathcal{Y}} \\ \mathbf{Q}'_j \leftarrow \mathbf{Q}'_j - \mathcal{P}'^{\mathsf{T}}\bar{\mathcal{Y}}' \end{cases} \quad \text{and} \quad \begin{cases} \mathbf{x}_{ij} \leftarrow \mathbf{x}_{ij} - \mathsf{P}_i\mathcal{P}^{\mathsf{T}}\bar{\mathcal{Y}} \\ \mathbf{x}'_{ij} \leftarrow \mathbf{x}'_{ij} - \mathsf{P}'_i\mathcal{P}'^{\mathsf{T}}\bar{\mathcal{Y}}' \end{cases} .$$

The reprojection error (4) is rewritten

$$\mathcal{C}^2(\mathcal{Q}, \mathcal{Q}') = \frac{1}{2nm}\left(\|\mathcal{X} - \mathcal{P}\mathcal{Q}\|^2 + \|\mathcal{X}' - \mathcal{P}'\mathcal{Q}'\|^2\right) \tag{9}$$

and problem (5) is reformulated as

$$\min_{\hat{\mathcal{Q}}, \hat{\mathcal{Q}}'} \mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \qquad \text{s.t.} \qquad \hat{\mathbf{Q}}'_j = \hat{\mathsf{A}}\hat{\mathbf{Q}}_j \ . \tag{10}$$

Step 3: Factorizing. Thanks to the orthonormal basis property $\mathcal{P}^{\mathsf{T}}\mathcal{P} = \mathrm{I}$, and since for any column-orthonormal matrix $\mathcal{A}$, $\|\mathcal{A}\mathbf{x}\| = \|\mathbf{x}\|$, we can rewrite the reprojection error for a single set of cameras as

$$\mathcal{R}^2(\mathcal{P}, \mathcal{Q}) \ \propto \ \|\mathcal{X} - \mathcal{P}\mathcal{Q}\|^2 \ = \ \|\mathcal{P}^{\mathsf{T}}\mathcal{X} - \mathcal{Q}\|^2 \ .$$

This allows to rewrite the reprojection error (9) as

$$\mathcal{C}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \ \propto \ \|\mathcal{P}^{\mathsf{T}}\mathcal{X} - \hat{\mathcal{Q}}\|^2 + \|\mathcal{P}'^{\mathsf{T}}\mathcal{X}' - \hat{\mathcal{Q}}'\|^2 \ = \ \| \underbrace{\begin{pmatrix} \mathcal{P}^{\mathsf{T}}\mathcal{X} \\ \mathcal{P}'^{\mathsf{T}}\mathcal{X}' \end{pmatrix}}_{\mathsf{\Lambda}} - \underbrace{\begin{pmatrix} \hat{\mathcal{Q}} \\ \hat{\mathcal{Q}}' \end{pmatrix}}_{\mathsf{\Delta}} \|^2 \ .$$

By introducing the constraint $\hat{\mathcal{Q}}' = \hat{\mathsf{A}}\hat{\mathcal{Q}}$ from (10) and, as in Sect. 3, an unknown global affine transformation B we can write

$$\mathsf{\Delta} \ = \ \begin{pmatrix} \mathrm{I} \\ \hat{\mathsf{A}} \end{pmatrix} \mathsf{B}\mathsf{B}^{-1}\hat{\mathcal{Q}} \ = \ \underbrace{\begin{pmatrix} \mathsf{B} \\ \hat{\mathsf{A}}\mathsf{B} \end{pmatrix}}_{\tilde{\mathcal{M}}} \underbrace{\mathsf{B}^{-1}\mathcal{Q}}_{\tilde{\mathcal{Q}}} \ .$$

The problem is reformulated as

$$\min_{\tilde{\mathcal{M}}, \tilde{\mathcal{Q}}} \|\mathsf{\Lambda} - \tilde{\mathcal{M}}\tilde{\mathcal{Q}}\|^2 \ .$$

A solution is given by the SVD of matrix $\mathsf{\Lambda}$

$$\mathsf{\Lambda}_{(6\times m)} = \mathsf{U}_{(6\times 6)}\mathsf{\Sigma}_{(6\times 6)}\mathsf{V}^{\mathsf{T}}_{(6\times m)} \ .$$

As in Sect. 3, let $\Sigma = \Sigma_u \Sigma_v$ be any decomposition of matrix Σ. We obtain $\tilde{\mathcal{M}} = \psi(\mathsf{U}\Sigma_u)$ and $\tilde{\mathcal{Q}} = \psi^{\mathsf{T}}(\mathsf{V}\Sigma_v^{\mathsf{T}})$. Using the partitioning $\tilde{\mathcal{M}} = \begin{pmatrix} \tilde{\mathsf{M}} \\ \tilde{\mathsf{M}}' \end{pmatrix}$, we get

$$\begin{cases} \mathsf{B} = \tilde{\mathsf{M}} \\ \hat{\mathsf{A}} = \tilde{\mathsf{M}}'\mathsf{B}^{-1} \\ \hat{\mathcal{Q}} = \mathsf{B}\tilde{\mathcal{Q}} \end{cases} .$$

Obviously, one needs to undo the effect of the orthonormalizing transformations, as follows

$$\begin{cases} \hat{\mathsf{A}} \leftarrow \mathsf{N}'\hat{\mathsf{A}}\mathsf{N}^{-1} \\ \hat{\mathcal{Q}} \leftarrow \mathsf{N}\hat{\mathcal{Q}} \end{cases} .$$

This algorithm runs with $m \geq 4$ point correspondences.

Note that it is possible to solve the problem without using the orthonormalizing transformations. This solution requires however to compute the SVD of a $(2(n + n') \times m)$ matrix, made by stacking the measurement matrices $\mathcal{X}$ and $\mathcal{X}'$, and is therefore much more computationally expensive than the algorithm above, and may be intractable for large sets of cameras and points.

4.3 Dealing with missing data.

The missing data case arises when some of the 3D points used for the alignment are not visible in all views. We propose an Expectation Maximization based extension of the algorithm to handle this case.

The EM algorithm is an iterative method which estimates the model parameters, given an incomplete set of measurement data. The main idea is to alternate between predicting the missing data and estimating the model. Since the log likelihood cannot be maximized using factorization, due to the missing data, it is replaced by its conditional expectation given the observed data, using the current estimate of the parameters. In the case where the log likelihood is a linear function of the missing data, this simply consists in replacing the missing data by their conditional expectations given the observed data at current parameter values. This approximated log likelihood is then maximized so as to yield a new estimate of the parameters. Monotone convergence to a local minimum of the Maximum Likelihood residual error (4) is shown *e.g.* in [11].

Since the reconstruction of both camera sets using factorization needs a complete data set, we are limited to the points visible in all views for the initial reconstruction. This allows to reconstruct all cameras, but only part of the 3D points. We then triangulate the missing points in order to complete the 3D point cloud. This preliminary expectation step yields a completed set of 3D data, that can be used in the alignement algorithm.

However, the reprojection error, *i.e.* the negative log likelihood, still cannot be minimized because of the incomplete measurement matrix $\mathcal{X}$. The expectation step predicts the missing image points by reprojecting them from the completed 3D points, namely for the missing point $\mathbf{x}_{ij}$, we set $\mathbf{x}_{ij} \leftarrow \mathsf{P}_i\hat{\mathbf{Q}}_j + \mathbf{t}_i$.

Table 1. The proposed Maximum Likelihood alignment algorithm.

OBJECTIVE

Given $m \geq 4$ 3D point correspondences $\{\mathbf{Q}_j \leftrightarrow \mathbf{Q}'_j\}$ obtained by affine reconstruction and triangulation of the missing data from two sets of images, with respectively n cameras $\{(\mathsf{P}_i, \mathbf{t}_i)\}$ and n' cameras $\{(\mathsf{P}'_i, \mathbf{t}'_i)\}$, as well as measured image points $\{\mathbf{x}_{ij}\}$ and $\{\mathbf{x}'_{ij}\}$ forming an incomplete data set, compute the affine transformation $(\hat{\mathsf{A}}, \hat{\mathbf{t}})$ and corrected point positions $\{\hat{\mathbf{Q}}_j \leftrightarrow \hat{\mathbf{Q}}'_j\}$ such that the reprojection error e is minimized.

ALGORITHM

1. **Compute the orthonormalizing transformations**:

$$\left(\cdots \ \mathsf{P}_i{}^\mathsf{T} \ \cdots\right)^\mathsf{T} \overset{\text{QR}}{=} \mathcal{P}\mathsf{N}^{-1} \qquad \text{and} \qquad \left(\cdots \ \mathsf{P}'^{\mathsf{T}}_i \ \cdots\right)^\mathsf{T} \overset{\text{QR}}{=} \mathcal{P}'\mathsf{N}'^{-1} \ .$$

2. **Form the 'joint projection' and the measurement matrices**:

$$\mathcal{X} = \begin{pmatrix} & \vdots & \\ \cdots & (\mathbf{x}_{ij} - \mathbf{t}_i) & \cdots \\ & \vdots & \end{pmatrix} \qquad \text{and} \qquad \mathcal{X}' = \begin{pmatrix} & \vdots & \\ \cdots & (\mathbf{x}'_{ij} - \mathbf{t}'_i) & \cdots \\ & \vdots & \end{pmatrix} \ .$$

3. **Expectation-Maximization**:
   (a) **Expectation**. Predict the missing point $\mathbf{x}_{ij}$ by setting $\mathbf{x}_{ij} \leftarrow \mathsf{P}_i\hat{\mathbf{Q}}_j$. Compute the reconstructed centroids:

$$\mathbf{C} = \frac{\mathcal{P}^\mathsf{T}}{m}\sum_{j=1}^{m}\begin{pmatrix} \vdots \\ \mathbf{x}_{ij} \\ \vdots \end{pmatrix} \qquad \text{and} \qquad \mathbf{C}' = \frac{\mathcal{P}'^\mathsf{T}}{m}\sum_{j=1}^{m}\begin{pmatrix} \vdots \\ \mathbf{x}'_{ij} \\ \vdots \end{pmatrix} \ .$$

   Cancel the translations:

$$\mathcal{X} = \begin{pmatrix} & \vdots & \\ \cdots & (\mathbf{x}_{ij} - \mathsf{P}_i\mathbf{C}) & \cdots \\ & \vdots & \end{pmatrix} \qquad \text{and} \qquad \mathcal{X}' = \begin{pmatrix} & \vdots & \\ \cdots & (\mathbf{x}'_{ij} - \mathsf{P}'_i\mathbf{C}') & \cdots \\ & \vdots & \end{pmatrix} \ .$$

   (b) **Maximization**. Factorize:

$$\begin{pmatrix} \mathcal{P}^\mathsf{T}\mathcal{X} \\ \mathcal{P}'^\mathsf{T}\mathcal{X}' \end{pmatrix} \overset{\text{SVD}}{=} \mathsf{U}\Sigma\mathsf{V}^\mathsf{T} \ \text{ and set } \ \begin{pmatrix} \tilde{\mathsf{M}} \\ \tilde{\mathsf{M}}' \end{pmatrix} = \psi(\mathsf{U}\sqrt{\Sigma}) \ \text{ and } \ \tilde{\mathcal{Q}} = \psi^\mathsf{T}(\mathsf{V}\sqrt{\Sigma}) \ .$$

   (c) **Recover the corrected points**. Set $\hat{\mathcal{Q}} = \mathsf{N}\tilde{\mathsf{M}}\tilde{\mathcal{Q}}$ and $\hat{\mathcal{Q}}' = \mathsf{N}'\tilde{\mathsf{M}}'\tilde{\mathcal{Q}}$.
   (d) **Transfer the points to the original coordinate frames**. Extract the corrected points $\hat{\mathbf{Q}}_j$ from $\hat{\mathcal{Q}}$. Translate them as $\hat{\mathbf{Q}}_j \leftarrow \hat{\mathbf{Q}}_j + \mathbf{C}$.
   (e) **Compute the reprojection error**:
   Set $e^2 = \frac{1}{2nm}\left(\sum_{j=1}^{m}\left(\sum_{i=1}^{n} d^2(\mathbf{x}_{ij} - \mathsf{P}_i\hat{\mathbf{Q}}_j) + \sum_{i=1}^{n'} d^2(\mathbf{x}'_{ij} - \mathsf{P}'_i\hat{\mathbf{Q}}'_j)\right)\right)$.
   (f) **Loop**. If convergence is not reached (see Sect. 4.3), loop on step (a).
4. **Recover the transformation**: Set $\hat{\mathsf{A}} = \mathsf{N}'\tilde{\mathsf{M}}'\tilde{\mathsf{M}}^{-1}\mathsf{N}^{-1}$ and $\mathbf{t} = \mathbf{C}' - \hat{\mathsf{A}}\mathbf{C}$.

The maximization step consists in applying the algorithm described in the complete data case. This yields an estimate of the sought-after transformation $(\hat{\mathsf{A}}, \hat{\mathbf{t}})$ as well as corrected point positions $\{\hat{\mathbf{Q}}_j \leftrightarrow \hat{\mathbf{Q}}'_j\}$.

These two steps are alternated, thus forming an iterative procedure where the corrected points are used in the expectation at the next iteration. In order to decide whether convergence is reached, the change in reprojection error between two iterations is measured. When the reprojection error stabilizes, the final result is returned.

Table 1 gives a summary of the algorithm with its EM extension.

5 Other Algorithms

We briefly describe two other alignment algorithms. They do not yield Maximum Likelihood Estimates under the previously-mentioned hypotheses on the noise distribution. They rely on 3D measurements and therefore naturally handle missing image data.

5.1 Minimizing the Non-Symmetric Transfer Error

This algorithm, dubbed 'TrError', is specific to the two camera set case. It is based on minimizing a non-symmetric 3D transfer error $\mathcal{E}(\hat{\mathsf{A}})$ as follows

$$\min_{\hat{\mathsf{A}},\hat{\mathbf{t}}} \mathcal{E}^2(\hat{\mathsf{A}}, \hat{\mathbf{t}}) \qquad \text{with} \qquad \mathcal{E}^2(\hat{\mathsf{A}}) = \frac{1}{m} \sum_{j=1}^{m} \|\mathbf{Q}'_j - \hat{\mathsf{A}}\mathbf{Q}_j - \hat{\mathbf{t}}\|^2 \ .$$

Differentiating $\mathcal{E}^2$ with respect to $\hat{\mathbf{t}}$ and nullifying the result allows to eliminate the translation by centering each 3D point set on its centroid. By rewriting the error function and applying standard linear least-squares, one obtains

$$\hat{\mathsf{A}} \ = \ \mathcal{Q}'\mathcal{Q}^{\dagger} \qquad \text{and} \qquad \hat{\mathbf{t}} \ = \ \bar{\mathbf{Q}}' - \hat{\mathsf{A}}\bar{\mathbf{Q}} \ .$$

5.2 Direct 3D Factorization

This algorithm, dubbed 'Fact3D', is based on directly factorizing the 3D reconstructed points. It is not restricted to the two camera set case, but for simplicity, we only describe this case. Generalization to multiple camera sets is trivial. The algorithm computes the aligning transformation $(\hat{\mathsf{A}}, \hat{\mathbf{t}})$ and perfectly corresponding points $\{\hat{\mathbf{Q}}_j \leftrightarrow \hat{\mathbf{Q}}'_j\}$. The reconstructed cameras are not taken into account by this algorithm, which entirely relies on 3D measures on the reconstructed points. Under certain conditions, this algorithm is equivalent to the proposed FactMLE-EM.

The problem is stated as

$$\min_{\hat{\mathcal{Q}},\hat{\mathcal{Q}}'} \mathcal{D}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \qquad \text{s.t.} \qquad \hat{\mathbf{Q}}'_j = \hat{\mathsf{A}}\hat{\mathbf{Q}}_j + \hat{\mathbf{t}} \ ,$$

where the 3D error function employed is defined by

$$\mathcal{D}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') = \frac{1}{2m} \left(\|\mathcal{Q} - \hat{\mathcal{Q}}\|^2 + \|\mathcal{Q}' - \hat{\mathcal{Q}}'\|^2 \right) \ .$$

Minimizing this error function means that if the noise *on the 3D point coordinates* were Gaussian, centered and i.i.d., which is *not* the case with our actual hypotheses, then this algorithm would yield the Maximum Likelihood Estimate.

Step 1: Eliminating the translation. By using the technique from Sect. 4.2, we obtain $\hat{\mathbf{t}} = \bar{\mathbf{Q}}' - \hat{\mathsf{A}}\bar{\mathbf{Q}}$. As in most factorization methods, cancelling the translation part out according to the error function $\mathcal{D}$ is done by centering each set of 3D points on its actual centroid: $\hat{\mathbf{Q}}_j \leftarrow \hat{\mathbf{Q}}_j - \bar{\mathbf{Q}}$ and $\hat{\mathbf{Q}}'_j \leftarrow \hat{\mathbf{Q}}'_j - \bar{\mathbf{Q}}'$. Henceforth, we assume to work in centered coordinates. The problem is rewritten as

$$\min_{\hat{\mathcal{Q}}, \hat{\mathcal{Q}}'} \mathcal{D}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \qquad \text{s.t.} \qquad \hat{\mathbf{Q}}'_j = \hat{\mathsf{A}}\hat{\mathbf{Q}}_j \ .$$

Step 2: Factorizing. Following the approach in Sect. 4.2, we rewrite $\mathcal{D}$ as

$$\mathcal{D}^2(\hat{\mathcal{Q}}, \hat{\mathcal{Q}}') \propto \| \begin{pmatrix} \mathcal{Q} \\ \mathcal{Q}' \end{pmatrix} - \begin{pmatrix} \hat{\mathcal{Q}} \\ \hat{\mathcal{Q}}' \end{pmatrix} \|^2 = \| \underbrace{\begin{pmatrix} \mathcal{Q} \\ \mathcal{Q}' \end{pmatrix}}_{\Lambda} - \underbrace{\begin{pmatrix} \mathsf{B} \\ \mathsf{AB} \end{pmatrix}}_{\tilde{\mathcal{M}}} \underbrace{\mathsf{B}^{-1}\hat{\mathcal{Q}}}_{\tilde{\mathcal{Q}}} \|^2 \ .$$

Using SVD of matrix $\Lambda = \mathsf{U}\Sigma\mathsf{V}^{\mathsf{T}}$, we obtain $\tilde{\mathcal{M}} = \psi(\mathsf{U}\Sigma_u)$ and $\tilde{\mathcal{Q}} = \psi^{\mathsf{T}}(\mathsf{V}\Sigma_v^{\mathsf{T}})$. By using the partitioning $\tilde{\mathcal{M}} = \left(\tilde{\mathsf{M}}\tilde{\mathsf{M}}'\right)^{\mathsf{T}}$, we get

$$\hat{\mathsf{A}} = \tilde{\mathsf{M}}'\tilde{\mathsf{M}}^{-1} \qquad \text{and} \qquad \hat{\mathcal{Q}} = \tilde{\mathsf{M}}\tilde{\mathcal{Q}} \ .$$

6 Experimental Evaluation

We evaluate our algorithm using simulated and real data. The implementation of all three compared algorithms, *i.e.* FactMLE-EM, TrError and Fact3D, as well as the generation of simulated data, have been done in C++.

6.1 Simulated Data

We generate m 3D points and two sets of n weak perspective cameras each. The pose of a camera is defined by its three dimensional location, viewing direction and roll angle (rotation angle around the optical axis). The corresponding affine projection matrix is given by a (2×3), truncated, rotation matrix $\bar{\mathsf{R}}_i$ together with a two-dimensional translation vector $\mathbf{t}_i$, both of which premultiplied by an internal calibration matrix. More precisely, we use weak perspective cameras $\mathsf{P}_i = \mathsf{A}_i\bar{\mathsf{R}}_i$ and $\mathbf{t}_i = \mathsf{A}_i\bar{\mathbf{T}}_i$, where A_i is the internal calibration matrix

$$\mathsf{A}_i = k_i \begin{pmatrix} \tau_i & 0 \\ 0 & 1 \end{pmatrix} \ .$$

The scale factor k_i models the average depth of the object and the focal length of the camera, and τ models the aspect ratio that we choose very close to 1. The 3D points are chosen from a uniform distribution inside a thin rectangular parallelepiped with dimensions $1 \times 1 \times (1-d)$, and the scale factors k_i are chosen so that the points are uniformly spread in 400×400 pixel images.

We generate three point sets containing the point visibles (i) in the first camera set, (ii) in the second one and (iii) in both camera sets. The third subset contains m_c points, whereas the two first subsets both contains $m - m_c$ points. Hence, m points are used to perform SFM on each camera set, while m_c points are used for the alignment. The points are projected onto the images where they are visible and gaussian noise with zero mean and standard deviation σ is added.

In order to assess the behaviour of the algorithms in the presence of non-perfectly affine cameras, we introduce the factor $0 \leq a \leq 1$. Let Z_{ij} be the depth of the j-th 3D point with respect to camera i, we scale the projected points $\mathbf{x}_{ij}$ by $\mathbf{x}_{ij} \leftarrow \frac{1}{\nu}\mathbf{x}_{ij}$ with $\nu = a + (1-a)Z_{ij}$, meaning that for $a = 1$, the points does not change and the projection is perfectly affine, and when a tends towards 0, the points undergo stronger and stronger perspective effects. The points are further scaled so that their standard deviation remains invariant, in order to keep them wellspread in the images.

So as to simulate the problem of incomplete data, *e.g.* due to occlusions, we generate a list of missing image points. We introduce the probability p_{point} that any given 3D point is occluded in some images and the probability p_{image} that it is occluded in one particular image. For simplicity, we take $p_{point} = p_{image} = p$, which gives a rate of missing data of p^2.

A 3D model is reconstructed from each of the two camera sets using the factorization algorithm described in Sect. 3. Once the camera matrices and 3D points are estimated, only the m_c points common to the two camera sets are considered for the alignment. We define the overlap ratio of the two camera sets to be $\theta = m_c/m$, *i.e.* for $\theta = 1$ all points are seen in all views, while for $\theta = 0$, the two sets of cameras do not share corresponding points.

Each of the three alignment algorithms yields estimates for the 3D affine transformation and corrected point clouds, except TrError which only gives the transformation. The comparison of the algorithms being based on the re-projection error, the point clouds used to compute it need to be re-estimated so that this error is minimized, given an estimated transformation. This must be done for TrError and Fact3D, but is useless for FactMLE-EM.

We use the following default setting for the simulations: $n = 5$ views, $m = 250$ points, $\theta = 0.2$ (*i.e.* a 20% overlap and $m_c = 50$ points common to the two 3D models), $\sigma = 3.0$ pixels, $d = 0.95$ (flat 3D scene), $a = 1$ (perfectly affine projections) and $p = 0.3$ (rate of missing data $p^2 = 0.09$). We vary each parameter at a time. Figures 2, 3 and 4 show the reprojection error averaged over 500 simulations for the three algorithms for different parameter values.

In Fig. 2, we vary the number of common points m_c (coupled with the total number of points m, so as to keep the overlap constant) and the number of cameras n, the former from 4 to 60, corresponding respectively to $m = 20$ and

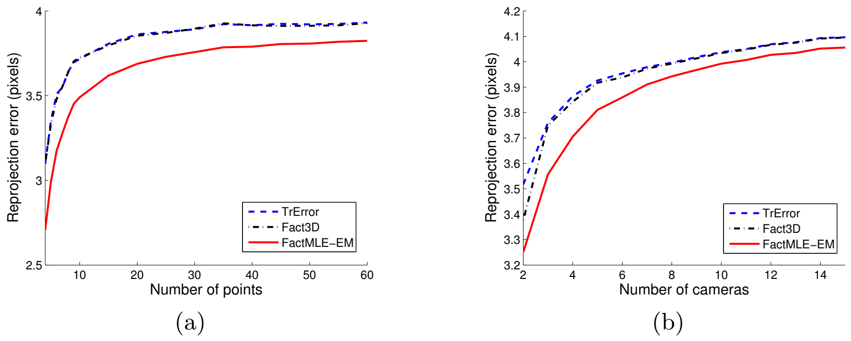


Fig. 2. Reprojection error against (a) the number of points m_c and (b) the number of cameras n.

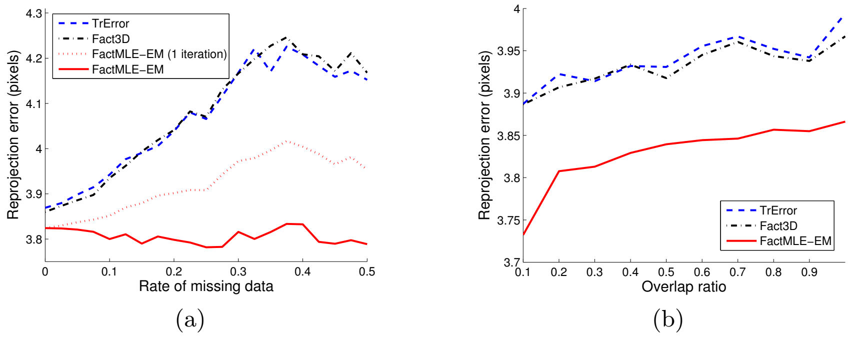


Fig. 3. Reprojection error against (a) the rate of missing data and (b) the extent of overlap θ between the two sets of cameras. For $\theta = 1$, all points are seen in all views.

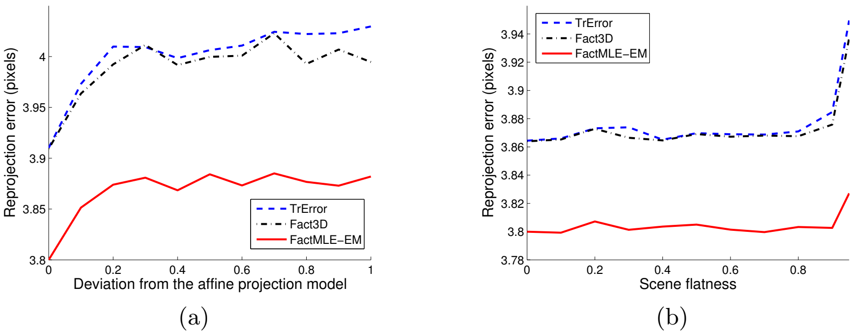


Fig. 4. Reprojection error against (a) the deviation a from the affine projection model and (b) the scene flatness d. For $a = 1$, the projection is perfectly affine. For $d = 1$, all 3D points lie on a plane.

$m = 300$, and the latter from 2 to 15. We see that for $m_c > 20$, the number of points has a much smaller influence on the errors. Whereas FACT3D and TRERROR show similar behaviour, FACTMLE-EM is distinguished by its lower reprojection error. The difference between our method and the other two seems to be more important in the cases where we have few points or few cameras.

In Fig. 3, the rate of missing data and the overlap ratio (coupled with the number of common points m_c, so as to keep the total number of points m constant) are varied, the former from 0 to 0.5 and the latter from 0.1 to 1.0. In order to emphasize the contribution of the EM scheme, in Fig. 3(a) we also display the reprojection error of FACTMLE-EM after the first iteration. When the rate of missing data grows, the three methods show different tendencies. Whereas FACTMLE-EM handles missing data well, the other methods prove to be unstable. However, considering only one iteration of FACTMLE-EM, the reprojection error increases just as for the other methods. The difference in performance is thus provided by the EM iterations.

In Fig. 4 the deviation from the affine model a varies from 0 to 1, from a perfectly affine projection, and the flatness of the simulated data d varies from 0 to 1, *i.e.* from a cube to a plane. Despite the fact that the alignement is affine, even completely projective cameras seem to be well modeled by the three methods. In fact, the error induced by the affine approximation is small compared to the added noise. The flatness of the scene does not change the result, except for very flat scenes making the algorithms unstable, FACT3D and TRERROR somewhat more than FACTMLE-EM. This result was expected since planar scenes are singular for the computation of a 3D affine transformation.

Simulations with varying σ reveal a quasi linear relationship between the the noise level and the reprojection error. The slope is somewhat less steep in the case of FACTMLE-EM than for the other two methods, indicating that our method is less sensitive to noise.

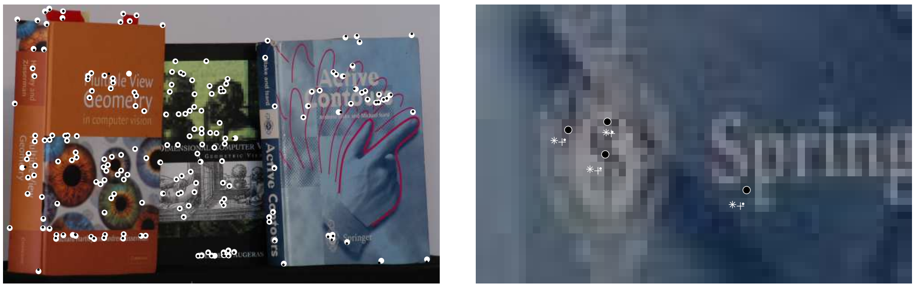
Although the three algorihtms have similar behaviour throughout the sequence of tests, except when varying the rate of missing data, FACTMLE-EM consistently outperforms the other ones.

6.2 Real Data

We applied the algorithms to real image sequences as follows. A number of images of a scene were taken from different angles and grouped into two sets. A certain number of point correspondences were defined within each one of the image sets, as well as for all the images, thus forming the measurement matrices $\mathcal{X}$ and $\mathcal{X}'$.

The camera used is an uncalibrated digital Nikon D100 with a lens of focal length $80 - 200$ mm, giving an image size of 2240×1488 pixels.

The 'books' sequence. We used a series of images of a rather flat scene, together with a large set of point correspondences, given by a tracking algorithm, shown in Fig. 5(a). So as to keep the experimental conditions close to the hypothesis of affine cameras, the photos are taken far away from the object using a large zoom.



(a) One image from the ‘books’ sequence overlaid with the $m_c = 196$ point correspondences in white and reprojected points in black.

(b) A detail from the image in (a) showing the original points in black and reprojected points in white, from FactMLE-EM (points), Fact3D (stars) and TrError (crosses).

Fig. 5. Results from the ‘books’ sequence.

This group of images consists of two sets of respectively $n = 2$ and $n' = 3$ images, together with the $m_c = 196$ common point correspondences, and respectively $m = 628$ and $m' = 634$ correspondences for the two sets, giving an approximate overlap of 80%. The reprojection errors we obtained are:

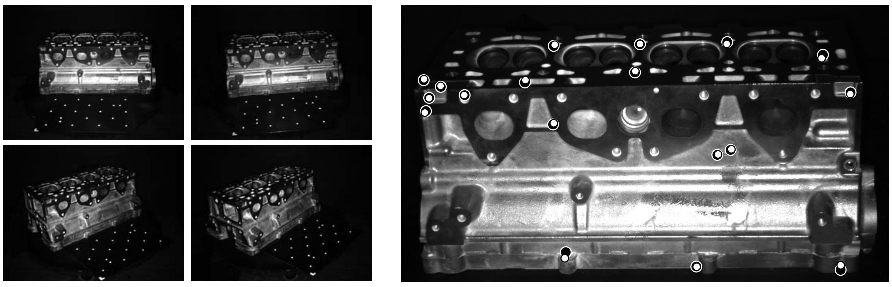
FactMLE-EM	1.90 pixels
Fact3D	1.97 pixels
TrError	2.17 pixels

A detail of an image with the reprojected points due to all three methods is shown in Fig. 5(b). As predicted by the tests on simulated data, FactMLE-EM performs better than Fact3D and TrError.

The ‘cylinder head’ sequence. This sequence was acquired under different conditions than the previous one. The photos were taken with the same camera, using a lens with a focal length of 12 mm, at a distance of 60 cm of the object, which is 40 cm long. The points, shown in Fig. 6(b), were manually entered. Using these settings, the affine camera model does not apply and the reconstruction performed prior to the alignment is therefore less reliable. Nevertheless, the result of the alignment is rather good. This group of images consists of two sets of respectively $n = n' = 2$ images, together with the $m_c = 18$ common point correspondences, and respectively $m = 22$ and $m' = 23$ correspondences for the two sets, giving an approximate overlap of 31%. The reprojection errors are:

FactMLE-EM	3.03 pixels
Fact3D	3.04 pixels
TrError	3.05 pixels

The two sets of images are displayed in Fig. 6(a) and the given point matches together with the FactMLE-EM reprojections are displayed in Fig. 6(b).



(a) The two sets of images of the ‘cylinder head’ sequence.

(b) The original points in black together with their FactMLE-EM reprojections in white.

Fig. 6. Results from the ‘cylinder head’ sequence.

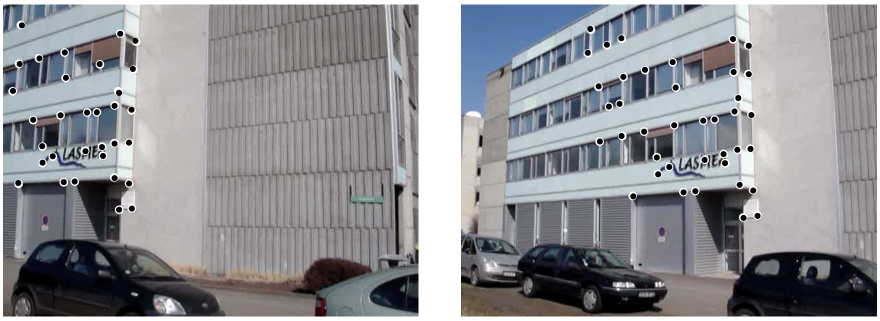
The ‘building’ sequence. The point correspondences are once again given by a tracking algorithm, but this time the data set is incomplete. We need at least two views of a 3D point in order to use it for the reconstruction, so we keep only those points that are present in two or more images. We then define a point correspondence to be common to the two sets and thus used for the alignment of the two reconstructions, as soon as it is present in (at least two images in each one of) the two sets. This group of images consists of two sets of respectively $n = n' = 5$ images, together with the $m_c = 40$ common point correspondences, and respectively $m = 94$ and $m' = 133$ correspondences for the two sets, giving an overlap of 43% and 30% respectively. The rates of missing data are for the first camera set 31% (13% for the common points) and for the second camera set 22% (11% for the common points). We note that the missing points are essentially not due to occlusions but to failure in the tracking algorithm or to the points being out of range in the images. The reprojection errors we obtained are:

FactMLE-EM	0.78 pixels
Fact3D	0.84 pixels
TrError	0.85 pixels

As predicted by the simulations with varying rate of missing data, the difference between the methods is more important when processing incomplete data. Whereas Fact3D and TrError yield similar errors, FactMLE-EM distinguishes itself with a significantly lower error. The results are displayed in Fig. 7.

7 Conclusions

We presented a method to compute the Maximum Likelihood Estimate of 3D affine transformations, under standard hypotheses on the noise distribution, aligning sets of 3D points obtained from uncalibrated affine cameras. The method



(a) (b)

Fig. 7. The original common points in white together with their FactMLE-EM reprojections in black. The two images are the first ones in the respective camera sets.

computes all aligning transformations in a single computation step in the occlusion-free case, by minimizing the reprojection error over all points and all images. An iterative extension is presented for the missing data case. Experimental results on simulated and real data show that the proposed method consistently performs better than other methods based on 3D measurements.

Future work could be devoted to the incorporation of other types of features.

References

1. Fitzgibbon, A., Zisserman, A.: Automatic camera recovery for closed or open image sequences. In: ECCV. (1998) 311–326
2. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: A factorization method. IJCV **9** (1992) 137–154
3. Beardsley, P., Zisserman, A., Murray, D.: Sequential updating of projective and affine structure from motion. IJCV **23** (1997) 235–259
4. Csurka, G., Demirdjian, D., Horaud, R.: Finding the collineation between two projective reconstructions. Comp. Vision and Image Underst. **75** (1999) 260–268
5. Walker, M., Shao, L., Volz, R.: Estimating 3D location parameters using dual number quaternions. Computer Vision, Graphics and Image Processing: Image Understanding **54** (1991) 358–367
6. Mundy, J., Zisserman, A., eds.: Geometric Invariance in Computer Vision. The MIT Press, Cambridge, MA, USA (1992)
7. Bartoli, A., Martinsson, H., Gaspard, F., Lavest, J.M.: On aligning sets of points reconstructed from uncalibrated affine cameras. In: SCIA. (2005) 531–540
8. Reid, I., Murray, D.: Active tracking of foveated feature clusters using affine structure. IJCV **18** (1996) 41–60
9. Golub, G., van Loan, C.: Matrix Computation. The Johns Hopkins University Press, Baltimore (1989)
10. Jacobs, D.: Linear fitting with missing data: Applications to structure-from-motion and to characterizing intensity images. In: CVPR. (1997) 206–212
11. McLachlan, G., Krishnan, T.: The EM algorithm and extensions. John Wiley & Sons, Inc. (1997)