Provably-Convergent Iterative Methods For Projective Structure From Motion

Shyjan Mahamud (mahamud@cs.cmu.edu)
Martial Hebert (hebert@ri.cmu.edu)
Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA 15213, USA

Yasuhiro Omori* (yasuhiro@inf.ka.cit.nihon-u.ac.jp)
Dept. of Industrial Engineering and Management, Nihon University, Narashino, Chiba, Japan

Kenton McHenry (mchenry@students.uiuc.edu)
Jean Ponce (ponce@cs.uiuc.edu)
Dept. of Computer Science and Beckman Institute, University of Illinois, Urbana, IL 61801, USA

Abstract. The estimation of the projective structure of a scene from point correspondences established across multiple images can be formulated as the minimization of the mean-squared distance between predicted and observed image points with respect to the unknown projection matrices, the positions of the scene points, and their depths relative to the cameras observing them. Since these unknowns are not independent, however, constraints must be chosen to ensure that the optimization process is well posed. This article examines three plausible choices, and shows that the first one leads to the Sturm-Triggs projective factorization algorithm, while the other two lead to provably-convergent iterative approaches to projective structure from motion. These algorithms have been implemented, and experiments with both synthetic and real image sequences are used to compare them to the Sturm-Triggs algorithm and to bundle adjustment.

Keywords: Image sequence analysis, projective structure from motion, projective factorization, bundle adjustment.

1. Introduction

Let us consider n fixed points P_j $(j = 1, \ldots, n)$ observed by m perspective cameras. Given some fixed world coordinate system, we write

$$\boldsymbol{p}_{ij} = \frac{1}{z_{ij}} \mathcal{M}_i \boldsymbol{P}_j \tag{1}$$

for $i = 1, \ldots, m$ and $j = 1, \ldots, n$. Here, $\boldsymbol{p}_{ij} = (u_{ij}, v_{ij}, 1)^T$ and z_{ij} denote respectively the (homogeneous) coordinate vector of the projection of P_j into image number i expressed in the corresponding camera's coordinate frame and the depth of P_j relative to that frame; $\mathcal{M}_i$ is the 3×4 projection matrix associated

* This work was done while Y. Omori was visiting the Beckman Institute at the University of Illinois at Urbana-Champaign.

with this camera in the world coordinate frame; and $\boldsymbol{P}_j$ is the homogeneous coordinate vector of the point P_j in that frame.

We address the problem of reconstructing both the matrices $\mathcal{M}_i$ $(i = 1, \ldots, m)$ and the vectors $\boldsymbol{P}_j$ $(j = 1, \ldots, n)$ from the image correspondences $\boldsymbol{p}_{ij}$. Of course, the parameters z_{ij} are also unknown, but their values are not independent of the values of $\mathcal{M}_i$ and $\boldsymbol{P}_j$: Indeed $z_{ij} = \boldsymbol{m}_{i3} \cdot \boldsymbol{P}_j$, where $\boldsymbol{m}_{i3}^T$ denotes the third row of the matrix $\mathcal{M}_i$. We will come back to this (important) point shortly.

Faugeras (1992) and Hartley *et al.* (1992) have shown that when no assumption is made about the internal parameters of the cameras, the projection matrices and the points' positions can only be reconstructed up to an arbitrary projective transformation, i.e., if $\mathcal{M}_i$ and $\boldsymbol{P}_j$ are solutions of (1), so are $\mathcal{M}_i\mathcal{Q}$ and $\mathcal{Q}^{-1}\boldsymbol{P}_j$ for any nonsingular 4×4 matrix $\mathcal{Q}$. The parameters z_{ij}, on the other hand, are independent of the choice of $\mathcal{Q}$, hence the name of *projective depths* often given to them.

Several effective techniques for computing a projective scene representation from multiple images have been proposed, e.g., (Faugeras, 1992; Hartley et al., 1992; Mohr et al., 1992; Zhang et al., 1995; Shashua, 1995; Hartley and Zisserman, 2000; Faugeras et al., 2001), but, with a few exceptions, e.g., (Christy and Horaud, 1996; Sturm and Triggs, 1996; Heyden, 1997; Morris and Kanade, 1998; Chen and Medioni, 1999; Han and Kanade, 2000), current approaches to projective motion analysis do not handle multiple images in a uniform manner. Instead, they use the algebraic relations associated with small sets of pictures (Faugeras and Papadopoulo, 1997; Heyden, 1998) to stitch together the corresponding reconstructions into a common framework (Laveau and Faugeras, 1994). This may involve a pair of images and the associated fundamental matrix (Faugeras, 1992; Hartley et al., 1992; Luong et al., 1993; Zhang et al., 1995), three pictures and the corresponding trifocal tensor (Shashua, 1995; Hartley, 1997), and even four views and their quadrifocal tensor (Hartley, 1998; Heyden, 1998). Once initial estimates of the scene structure and camera parameters have been obtained, they can be refined using all images of all visible points and non-linear least-squares techniques, an approach known as *bundle adjustment* in photogrammetry (Thompson et al., 1966; Slama et al., 1980) (see also (Triggs et al., 2000; Hartley and Zisserman, 2000; Faugeras et al., 2001) for surveys in the computer vision context).

A general approach to projective structure from motion that handles multiple images in a uniform manner is to formulate this problem as the minimization of some mean-squared discrepancy between predicted and observed image points with respect to the unknown projection matrices, the positions of the scene points, and their projective depths. Different measures of discrepancy can be derived from (1), including the *geometric distance* typically minimized in bundle-adjustment methods, and the linearized *algebraic distance* proposed in the next section. Since, as noted before, the unknown parameters are not independent, it

is in general necessary to impose appropriate constraints on these parameters to ensure that the corresponding optimization process is well posed.

We examine in this paper three plausible constraints: The first one leads to the Sturm-Triggs (1996) projective factorization algorithm, and the other two lead to simple iterative algorithms that are guaranteed to converge to a local minimum of the error function they try to minimize (see the Appendix for convergence proofs based on the *global convergence theorem* (Zangwill, 1969) from numerical analysis). This is in stark contrast with related approaches (Sturm and Triggs, 1996; Heyden, 1997; Morris and Kanade, 1998; Chen and Medioni, 1999; Han and Kanade, 2000), whose convergence properties have not been elucidated yet. It is worth noting that the variant of the Sturm-Triggs algorithm proposed in (Oliensis, 1999) has been shown to have a monotonically decreasing (and thus convergent since it is bounded below by zero) error, which does *not* guarantee (by itself) convergence to a local minimum.[1]

The proposed algorithms have been implemented, and experiments with both synthetic and real image sequences are used to compare them to the projective factorization method introduced by Sturm and Triggs (1996) and to the non-linear bundle adjustment method of Morris, Kanatani and Kanade (2000). The computational efficiency of the proposed algorithms is discussed as well.

Preliminary versions of portions of this paper have appeared in (Mahamud and Hebert, 2000; Mahamud et al., 2001).

2. Background

Ideally, we would like to minimize the reprojection error, that is, the mean-squared distance between the observed image points and the point positions predicted from the parameters z_{ij}, $\mathcal{M}_i$ and $\boldsymbol{P}_j$, or equivalently,

$$\sum_{i,j} |\boldsymbol{p}_{ij} - \frac{1}{z_{ij}} \mathcal{M}_i \boldsymbol{P}_j|^2.$$

Unfortunately the corresponding optimization problem is difficult since the error we are trying to minimize is non-linear in the unknowns z_{ij}, $\mathcal{M}_i$ and $\boldsymbol{P}_j$. Instead, we will minimize the algebraic error

$$E \stackrel{\text{def}}{=} \sum_{i,j} |z_{ij} \boldsymbol{p}_j - \mathcal{M}_i \boldsymbol{P}_j|^2$$

with respect to the unknowns $\mathcal{M}_i$, $\boldsymbol{P}_j$ and z_{ij}. This error measure is not as intuitively satisfying as the previous one, but the rest of this article will show

[1] This can be illustrated by a simple example: Suppose we want to minimize the function $f(x) = x^2$ using the iterations $x_n = 1 + 1/n$. The error $f(x_n)$ monotonically decreases and converges to 1, which is of course not a local minimum of f.

that its minimization under various classes of constraints can serve as a unifying framework for a wide class of projective structure-from-motion techniques.

Note that the minimization of E is ill posed unless some constraints are imposed on the parameters $\mathcal{M}_i$, $\boldsymbol{P}_j$ and z_{ij}. Indeed, as mentioned earlier, these unknowns are not independent: The matrices $\mathcal{M}_i$ and the vectors $\boldsymbol{P}_j$ are only defined up to scale. Furthermore, if $\mathcal{M}_i$, $\boldsymbol{P}_j$ and z_{ij} are solutions of (1), so are $\alpha_i\mathcal{M}_i$, $\beta_j\boldsymbol{P}_j$ and $\alpha_i\beta_j z_{ij}$ for arbitrary values of the scalars α_i and β_j. In addition, the conversion of the geometric reprojection error into an algebraic one introduces a large class of trivial zeros of the error function E corresponding to zero projective depths z_{ij}. The most obvious example is to pick $\mathcal{M}_i = 0$ and $\boldsymbol{P}_j = 0$, but many other trivial zeros exist as well, for example $\mathcal{M}_i = \mathcal{M}_0$, and $\boldsymbol{P}_j = \boldsymbol{P}_0$, where $\mathcal{M}_0$ is an arbitrary rank-3 3×4 matrix and $\boldsymbol{P}_0$ is a nonzero vector in its kernel. This is true *independently* of the method used to minimize E. We will come back to this crucial point later.

We can obtain a more compact expression for E by introducing the data matrix $\mathcal{I}$ (Sturm and Triggs, 1996) defined by

$$\mathcal{I} \stackrel{\text{def}}{=} \begin{bmatrix} z_{11}\boldsymbol{p}_{11} & z_{12}\boldsymbol{p}_{12} & \cdots & z_{1n}\boldsymbol{p}_{1n} \\ z_{21}\boldsymbol{p}_{21} & z_{22}\boldsymbol{p}_{22} & \cdots & z_{2n}\boldsymbol{p}_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ z_{m1}\boldsymbol{p}_{m1} & z_{m2}\boldsymbol{p}_{m2} & \cdots & z_{mn}\boldsymbol{p}_{mn} \end{bmatrix},$$

and observing that, given m images of n points, (1) can be rewritten as

$$\mathcal{I} = \mathcal{M}\mathcal{P},$$

where

$$\mathcal{M} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{M}_1 \\ \mathcal{M}_2 \\ \cdots \\ \mathcal{M}_m \end{bmatrix} \text{ and } \mathcal{P} \stackrel{\text{def}}{=} [\, \boldsymbol{P}_1 \; \boldsymbol{P}_2 \; \ldots \; \boldsymbol{P}_n \,].$$

It follows immediately that

$$E = |\mathcal{I} - \mathcal{M}\mathcal{P}|^2,$$

where, this time, $|\mathcal{A}|$ denotes the Frobenius norm of the matrix $\mathcal{A}$, i.e., the square root of the sum of the squares of its entries.

Minimizing E is thus equivalent to finding the parameters z_{ij}, $\mathcal{M}$ and $\mathcal{P}$ that minimize the Frobenius norm of the difference between $\mathcal{I}$ and $\mathcal{M}\mathcal{P}$. As noted before, appropriate constraints on these unknowns must be chosen to ensure that the optimization process is well posed. We will examine in the rest of this presentation three plausible choices, and show that the first one leads to the Sturm-Triggs projective factorization algorithm, while the other two lead to new, provably-convergent approaches to projective structure from motion.

We discuss in this section the structure-from-motion algorithm proposed by Sturm and Triggs (1996). This algorithm relies on singular value decomposition to factorize the matrix $\mathcal{I}$ and minimize E (see also the articles of Christy and Horaud (1996), Heyden *et al.* (1997; 1999) and Han and Kanade (2000) for related approaches). It generalizes to the projective case the factorization approach to structure from motion originally proposed by Tomasi and Kanade (1992) in the affine setting. The Sturm-Triggs algorithm uses epipolar constraints between pairs of successive images to compute initial values for the projective depths z_{ij} (see (Sturm and Triggs, 1996) for details). The trivial solutions mentioned in the previous section are eliminated by introducing appropriate constraints on the entries of $\mathcal{I}$ (hence on the parameters z_{ij}), namely scaling the rows of this matrix so they have unit norms, then scaling its columns so they have unit norm.

At this point, the values of the projective depths are held constant, and singular value decomposition (SVD) is used to estimate the matrices $\mathcal{M}$ and $\mathcal{P}$ minimizing E. As shown in (Golub and Van Loan, 1996) for example, the $3m \times 4$ and $4 \times n$ matrices $\mathcal{M}$ and $\mathcal{P}$ minimizing $E = |\mathcal{I} - \mathcal{M}\mathcal{P}|^2$ can be chosen equal to $\mathcal{U}_4\sqrt{\mathcal{W}_4}$ and $\sqrt{\mathcal{W}_4}\mathcal{V}_4^T$, where $\mathcal{U}\mathcal{W}\mathcal{V}^T$ denotes the singular value decomposition of the matrix $\mathcal{I}$, and $\mathcal{U}_4$, $\mathcal{W}_4$ and $\mathcal{V}_4$ denote the $3m \times 4$, 4×4 and $4 \times n$ matrices formed by the four leftmost columns of $\mathcal{U}$, $\mathcal{W}$ and $\mathcal{V}$. Note that $\mathcal{M}$ and $\mathcal{P}$ are only defined up to an arbitrary projective transformation, so we could have taken just as well $\mathcal{M} = \mathcal{U}_4$ and $\mathcal{P} = \mathcal{W}_4\mathcal{V}_4^T$ for example.

The original Sturm-Triggs algorithm stops at this point, but Triggs (1996) later proposed to make this scheme iterative by refining the estimate of the projective depths at each iteration (this is easy if $\mathcal{M}$ and $\mathcal{P}$ are assumed to be known), renormalizing $\mathcal{I}$, switching back to re-estimating $\mathcal{M}$ and $\mathcal{P}$, etc. The iterative algorithm is summarized in Table I.

Although this method gives good results on both synthetic and real data, there is no guarantee that the iterative process will converge (or even that the error will decrease with each iteration) because of the renormalization of the matrix $\mathcal{I}$ in step 2(a). Unfortunately, this step is necessary to avoid a trivial solution to the minimization process. The iterative factorization algorithms proposed by Christy and Horaud (1996), Heyden *et al.* (1997; 1999) and Han and Kanade (2000) have not formally been shown to be convergent either, although they also give good results in practice. The variant of the Sturm-Triggs algorithm proposed by Oliensis (1999) uses a different normalization strategy—namely, holding the norm of the data matrix $\mathcal{I}$ constant. Its error can be shown to monotonically decrease with iterations, thus converging to some value that, unfortunately, is not guaranteed to be a local minimum. Section 3 will discuss an alternative that is guaranteed to converge to a local minimum.

Table I. The Sturm-Triggs factorization algorithm for projective shape from motion.

1. Compute an initial estimate of the projective depths z_{ij}, with $i = 1, \ldots, m$ and $j = 1, \ldots, n$.
2. Repeat:
   (a) normalize each row of the data matrix $\mathcal{I}$, then normalize each one of its columns;
   (b) use singular value decomposition to compute the matrices $\mathcal{M}$ and $\mathcal{P}$ minimizing $|\mathcal{I} - \mathcal{M}\mathcal{P}|^2$;
   (c) use linear least squares to compute, for $i = 1, \ldots, m$ and $j = 1, \ldots, n$, the value of z_{ij} minimizing $|z_{ij}\boldsymbol{p}_{ij} - \mathcal{M}_i \boldsymbol{P}_j|^2$;

   until convergence.

2.2. Bundle Adjustment

Bundle-adjustment methods originate from the field of photogrammetry (Thompson et al., 1966; Slama et al., 1980). They seek to minimize the mean-squared reprojection error between observed and predicted image features via (usually non-linear) least-squares optimization techniques such as the Gauss-Newton or Levenberg-Marquardt algorithms.

In these techniques, the projective depths do not appear as independent variables, and Cartesian image coordinates are used instead of projective ones to rewrite the perspective projection equation (1) as

$$\begin{cases} u_{ij} = \dfrac{\boldsymbol{m}_{i1} \cdot \boldsymbol{P}_j}{\boldsymbol{m}_{i3} \cdot \boldsymbol{P}_j}, \\ v_{ij} = \dfrac{\boldsymbol{m}_{i2} \cdot \boldsymbol{P}_j}{\boldsymbol{m}_{i3} \cdot \boldsymbol{P}_j}. \end{cases}$$

where the vectors $\boldsymbol{m}_{ij}^T$ $(j = 1, 2, 3)$ denote the three rows of the projection matrix $\mathcal{M}_i$. In this setting, minimizing the mean-squared distance between predicted and observed image points reduces to a non-linear optimization problem in the parameters of the matrices $\mathcal{M}_i$ and vectors $\boldsymbol{P}_j$, as outlined in Table II.

3. A Provably-Convergent Iterative Factorization Method

3.1. Principle of the Approach

The approach proposed in this section solves the convergence problem of the Sturm-Triggs algorithm in a simple manner. As before, the minimization of

Table II. Bundle-adjustment algorithm. Here the vectors $\boldsymbol{m}_{i1}^T$, $\boldsymbol{m}_{i2}^T$ and $\boldsymbol{m}_{i3}^T$ denote the rows of the projection matrix $\mathcal{M}_i$ $(i = 1, \ldots, m)$. Since the error function being minimized is not homogeneous in the unknown parameters, the last coordinate of the vectors $\boldsymbol{m}_{i3}$ $(i = 1, \ldots, m)$ and $\boldsymbol{P}_j$ $(j = 1, \ldots, n)$ is normally held constant and equal to 1.

1. Compute an initial estimate of the projection matrices $\mathcal{M}_i$ and of the vectors $\boldsymbol{P}_j$, with $i = 1, \ldots, m$ and $j = 1, \ldots, n$.
2. Minimize
$$\sum_{ij} \left[\left(u_{ij} - \frac{\boldsymbol{m}_{i1} \cdot \boldsymbol{P}_j}{\boldsymbol{m}_{i3} \cdot \boldsymbol{P}_j} \right)^2 + \left(v_{ij} - \frac{\boldsymbol{m}_{i2} \cdot \boldsymbol{P}_j}{\boldsymbol{m}_{i3} \cdot \boldsymbol{P}_j} \right)^2 \right]$$
with respect to the matrices $\mathcal{M}_i$ and the vectors $\boldsymbol{P}_j$.

E alternates steps where the motion and structure parameters are estimated from the data matrix with steps where the projective depths are computed from the motion and structure estimates. The key difference with the Sturm-Triggs method is that the values of the projective depths are estimated by minimizing E under the constraint that the columns $\boldsymbol{d}_j = (z_{1j}\boldsymbol{p}_{1j}^T, \ldots, z_{mj}\boldsymbol{p}_{mj}^T)^T$ $(j = 1, \ldots, n)$ of the matrix $\mathcal{I}$ have unit norm. This constraint avoids the renormalization step and, as will be shown below, it guarantees convergence.

Concretely, let us assume that, at some stage of the minimization process, we fix the value of $\mathcal{M}$ to its current estimate and compute, for $j = 1, \ldots, n$, the values of $\boldsymbol{z}_j \stackrel{\text{def}}{=} (z_{1j}, \ldots, z_{mj})^T$ and $\boldsymbol{P}_j$ that minimize

$$E_j \stackrel{\text{def}}{=} \sum_{i=1}^{m} |z_{ij}\boldsymbol{p}_j - \mathcal{M}_i \boldsymbol{P}_j|^2.$$

These values will of course minimize E as well. Writing that the gradient of E_j with respect to the vector $\boldsymbol{P}_j$ is zero at a minimum yields:

$$0 = \frac{\partial E_j}{\partial \boldsymbol{P}_j} = 2 \sum_{i=1}^{m} \mathcal{M}_i^T (z_{ij}\boldsymbol{p}_{ij} - \mathcal{M}_i \boldsymbol{P}_j),$$

or

$$\mathcal{M}^T \boldsymbol{d}_j = \mathcal{M}^T \mathcal{M} \boldsymbol{P}_j \Longleftrightarrow \boldsymbol{P}_j = \mathcal{M}^\dagger \boldsymbol{d}_j,$$

where $\mathcal{M}^\dagger \stackrel{\text{def}}{=} (\mathcal{M}^T\mathcal{M})^{-1}\mathcal{M}^T$ is the pseudoinverse of $\mathcal{M}$. In turn, substituting this value in the definition of E_j yields

$$E_j = |(\text{Id} - \mathcal{M}\mathcal{M}^\dagger)\boldsymbol{d}_j|^2.$$

Now, $\mathcal{M}$ is a $3m \times 4$ matrix of rank 4, whose singular value decomposition $\mathcal{U}\mathcal{W}\mathcal{V}^T$ is formed by the product of a column-orthogonal $3m \times 4$ matrix $\mathcal{U}$, a

4×4 non-singular diagonal matrix $\mathcal{W}$ and a 4×4 orthogonal matrix $\mathcal{V}^T$. The pseudoinverse of $\mathcal{M}$ is $\mathcal{M}^\dagger = \mathcal{V}\mathcal{W}^{-1}\mathcal{U}^T$; substituting this value in the expression of E_j and taking into account the fact that $|\boldsymbol{d}_j|^2 = 1$ immediately yields

$$E_j = |(\mathrm{Id} - \mathcal{U}\mathcal{U}^T)\boldsymbol{d}_j|^2 = 1 - |\mathcal{U}\boldsymbol{d}_j|^2.$$

In turn, this means that minimizing E_j with respect to $\boldsymbol{z}_j$ and $\boldsymbol{P}_j$ is equivalent to maximizing $|\mathcal{U}\boldsymbol{d}_j|^2$ under the constraint $|\boldsymbol{d}_j|^2 = 1$. Finally, observing that

$$\boldsymbol{d}_j = \mathcal{Q}_j \boldsymbol{z}_j, \quad \text{where} \quad \mathcal{Q}_j \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{p}_{1j} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{p}_{2j} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \dots & \boldsymbol{p}_{mj} \end{bmatrix},$$

shows that minimizing E_j is equivalent to maximizing $|\mathcal{R}_j \boldsymbol{z}_j|^2$ with respect to $\boldsymbol{z}_j$ under the constraint $|\mathcal{Q}_j \boldsymbol{z}_j|^2 = 1$, where $\mathcal{R}_j \stackrel{\text{def}}{=} \mathcal{U}^T \mathcal{Q}_j$. Note that the matrices $\mathcal{Q}_j$ depend only on the input feature points and thus remains constant throughout the iterations. This is a generalized eigenvalue problem, whose solution is the unit vector $\boldsymbol{z}_j$ corresponding to the largest scalar λ such that:

$$\mathcal{R}_j^T \mathcal{R}_j \boldsymbol{z}_j = \lambda \mathcal{Q}_j^T \mathcal{Q}_j \boldsymbol{z}_j$$

The minimization step where the projective depths are held constant and $\mathcal{M}$ and $\mathcal{P}$ are updated is the same as in the Sturm-Triggs approach. The overall algorithm is summarized in Table III. The initial projective depth values are set to 1 or they can be computed as before from estimates of the epipolar geometry.

3.2. Convergence

It is easy to show that the error E converges monotonically. Indeed, let $E^{(0)}$ be the current error value at the beginning of each iteration; the first step of the algorithm does not change the vectors $\boldsymbol{z}_j$ but minimizes E with respect to the unknowns $\mathcal{M}$ and $\boldsymbol{P}_j$. If $E^{(a)}$ is the value of the error at the end of step 3(a), we have therefore $E^{(a)} \leq E^{(0)}$. Now step 3(b) does not change the matrices $\mathcal{M}$ and $\mathcal{P}$ but minimizes each error term E_j with respect to the vectors $\boldsymbol{z}_j$. Therefore the error $E^{(b)}$ at the end of this step is smaller than or equal to $E^{(a)}$. This shows that the error decreases in a monotone manner at each iteration, and since it is bounded below by zero, it also converges monotonically to some value E^*. Monotonic convergence of the error E to E^* is not sufficient for our purpose, however, since it does not guarantee the convergence of the parameters $\mathcal{M}_i$ and $\boldsymbol{P}_j$, and does not imply that E^* is a local minimum of E. The proof of these two properties is given in the Appendix using the global convergence theorem from (Zangwill, 1969; Luenberger, 1984).

Table III. A provably-convergent iterative factorization algorithm for projective shape from motion. Note that the matrix $\mathcal{I}$ is only normalized once during initialization (step 2 of the algorithm).

1. Compute an initial estimate of the projective depths z_{ij}, with $i = 1, \ldots, m$ and $j = 1, \ldots, n$.
2. Normalize each column of the data matrix $\mathcal{I}$.
3. Repeat:
   (a) use singular value decomposition to compute the matrices $\mathcal{M}$ and $\mathcal{P}$ minimizing $|\mathcal{I} - \mathcal{M}\mathcal{P}|^2$;
   (b) for $j = 1$ to n, compute the matrices $\mathcal{R}_j$ and find the value of $\boldsymbol{z}_j$ maximizing $|\mathcal{R}_j \boldsymbol{z}_j|^2$ under the constraint $|\mathcal{Q}_j \boldsymbol{z}_j|^2 = 1$ as the solution of a generalized eigenvalue problem;
   (c) update the value of $\mathcal{I}$ accordingly;

   until convergence.

4. A Provably-Convergent Iterative Bilinear Algorithm

We now present a simple alternative to the factorization-based algorithms discussed in the previous sections. Unlike those, the proposed method does not attempt to estimate the projective depths. Instead, these are shown to be redundant, and their elimination leads to a new expression for E as the squared norm of a vector that is a bilinear function of the matrices $\mathcal{M}_i$ and vectors $\boldsymbol{P}_j$. In turn, an appropriate choice of constraints reduces the minimization of E to the solution of a series of eigenvalue problems. Unlike related bilinear algorithms for affine and projective structure from motion (Morris and Kanade, 1998; Chen and Medioni, 1999), the proposed method can be shown to converge to a local minimum of E. A major advantage of this approach is that it does not require all points to be visible in all the images. This is a crucial consideration in building practical modeling systems since, for most motion sequences, it is unlikely that all points will be visible in all images. In contrast, factorization techniques are unable to deal with such cases.

To fix the notation, we will assume that, for $j = 1, \ldots, n$, the point $\boldsymbol{P}_j$ is visible in $m_j \leq m$ images with indices in the range I_j, and that $n_i \leq n$ points are visible in image number i ($i = 1, \ldots, m$), with indices in the range J_i. Note that if p is the total number of image points observed, we have $p = \sum_{i=1}^{m} n_i = \sum_{j=1}^{n} m_j \leq mn$, and of course $|I_j| = m_j$ and $|J_i| = n_i$, where $|S|$ denotes the number of elements in a finite set S.

4.1. Principle of the Approach

Before introducing the new constraints that will be used in the minimization of E, let us show that, in general, the corresponding optimization process does *not*, in fact, require the estimation of the projective depths. Writing that the derivative of E with respect to z_{ij} should be zero at an extremum of this function yields:

$$0 = \frac{\partial E}{\partial z_{ij}} = \frac{\partial E_{ij}}{\partial z_{ij}} = 2\boldsymbol{p}_{ij}^T(z_{ij}\boldsymbol{p}_{ij} - \mathcal{M}_i\boldsymbol{P}_j),$$

or

$$z_{ij} = \frac{1}{|\boldsymbol{p}_{ij}|^2}(\boldsymbol{p}_{ij}^T\mathcal{M}_i\boldsymbol{P}_j).$$

Substituting in the definition of E shows that, at an extremum of this function, we must have

$$E = \sum_{ij} \frac{1}{|\boldsymbol{p}_{ij}|^4}|(\boldsymbol{p}_{ij}\boldsymbol{p}_{ij}^T - |\boldsymbol{p}_{ij}|^2\mathrm{Id})\mathcal{M}_i\boldsymbol{P}_j|^2.$$

Given two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ in $\mathbb{R}^3$, the well-known identity holds:

$$|(\boldsymbol{a}\boldsymbol{a}^T - |\boldsymbol{a}|^2\mathrm{Id})\boldsymbol{b}|^2 = |\boldsymbol{a}|^2|\boldsymbol{a}\times\boldsymbol{b}|^2,$$

where "$\times$" denotes the operator that associates with two vectors their cross product. In particular, at one of its extrema, E has the form

$$E = \sum_{ij} \frac{1}{|\boldsymbol{p}_{ij}|^2}|\boldsymbol{p}_{ij}\times(\mathcal{M}_i\boldsymbol{P}_j)|^2.$$

The homogeneous image coordinate vectors $\boldsymbol{p}_{ij}$ can be scaled to unit norm during preprocessing, and our task is thus finally reduced to minimizing

$$E = \sum_{ij} |\boldsymbol{p}_{ij}\times(\mathcal{M}_i\boldsymbol{P}_j)|^2$$

under appropriate constraints on the matrices $\mathcal{M}_i$ and the vectors $\boldsymbol{P}_j$. Note that this minimization process *never* involves the estimation of the projective depths.

The next question is how to choose the right constraints for the minimization. We have chosen to constrain both the projection matrices and the points to have unit norm, i.e., $|\mathcal{M}_i|^2 = 1$ and $|\boldsymbol{P}_j|^2 = 1$ for $i = 1,..,m$ and $j = 1,..,n$ (see (Morris and Kanade, 1998) and (Chen and Medioni, 1999) for related approaches in the affine and projective cases respectively). As mentioned earlier, E admits trivial zeros satisfying these constraints, corresponding to picking $\mathcal{M}_i = \mathcal{M}_0$ $(i = 1,\ldots,m)$ and $\boldsymbol{P}_j = \boldsymbol{P}_0$ $(j = 1,\ldots,n)$, where $\mathcal{M}_0$ is an arbitrary rank-3 3×4 matrix with unit Frobenius form and $\boldsymbol{P}_0$ is a unit vector in its kernel (there are other trivial zeros corresponding to lower-rank values of $\mathcal{M}_0$ and

families of vectors in their kernels). On the other hand, the iterative bilinear algorithm presented in the next section can be proven to always converge to a local minimum of E (see Appendix), and it has never converged to a trivial solution in any of our experiments (see Section 5 for details).

4.2. Algorithm

We propose to start with some initial estimates of the vectors $\boldsymbol{P}_j$ and alternate steps where these vectors are kept constant (resp. estimated) while the matrices $\mathcal{M}_i$ are estimated (resp. kept constant). This is an instance of a class of techniques for structure from motion called *resection-intersection* methods in photogrammetry (Triggs et al., 2000), that interleave steps where the camera parameters are estimated while the observed point positions are kept fixed (*resection*) with steps where the point positions are estimated while the camera parameters are kept constant (*intersection*). Variants of this approach include the bilinear methods of Morris and Kanade (1998) and Chen and Medioni (1999) for affine and structure from motion, and the photogrammetric method of *block successive over relaxation* (Brown, 1976).

Let us rewrite our error function as

$$E = \sum_{i=1}^{m} A_i = \sum_{j=1}^{n} B_j,$$

where

$$\begin{cases} A_i \stackrel{\text{def}}{=} \sum_{j \in J_i} |\boldsymbol{p}_{ij} \times (\mathcal{M}_i \boldsymbol{P}_j)|^2 & \text{for} \quad i = 1, \ldots, m, \\ B_j \stackrel{\text{def}}{=} \sum_{i \in I_j} |\boldsymbol{p}_{ij} \times (\mathcal{M}_i \boldsymbol{P}_j)|^2 & \text{for} \quad j = 1, \ldots, n. \end{cases}$$

The algorithm presented in this section alternates (1) steps where the point positions $\boldsymbol{P}_j$ are held constant while, for $i = 1, .., m$, the error A_i is minimized under the constraint $|\mathcal{M}_i|^2 = 1$ with (2) steps where the matrices $\mathcal{M}_i$ are held constant while, for $j = 1, .., n$, the error B_j is minimized under the constraint $|\boldsymbol{P}_j|^2 = 1$. It is clear that the constraints $|\mathcal{M}_i|^2 = 1$ and $|\boldsymbol{P}_j|^2 = 1$ for $i = 1, .., m$ and $j = 1, .., n$ will remain satisfied throughout the process.

Let us first fix the vectors $\boldsymbol{P}_j$. The error term associated with the projection matrix $|\mathcal{M}_i|^2 = 1$ $(i = 1, \ldots, m)$ can be expressed as

$$A_i = |C_i \boldsymbol{m}_i|^2,$$

where, $\boldsymbol{m}_i$ denotes the vector of $\mathbb{R}^{12}$ defined by $\boldsymbol{m}_i^T = (\boldsymbol{m}_{i1}^T, \boldsymbol{m}_{i2}^T, \boldsymbol{m}_{i3}^T)$, where $\boldsymbol{m}_{i1}^T$, $\boldsymbol{m}_{i2}^T$ and $\boldsymbol{m}_{i3}^T$ are the rows of $\mathcal{M}_i$, and $\mathcal{C}_i$ is the the $3n_i \times 12$ matrix obtained by stacking the n_i instances of the matrix

$$\begin{bmatrix} -w_{ij}\boldsymbol{P}_j^T & \boldsymbol{0}^T & u_{ij}\boldsymbol{P}_j^T \\ \boldsymbol{0}^T & -w_{ij}\boldsymbol{P}_j^T & v_{ij}\boldsymbol{P}_j^T \\ -v_{ij}\boldsymbol{P}_j^T & u_{ij}\boldsymbol{P}_j^T & \boldsymbol{0}^T \end{bmatrix}$$

corresponding to all values of j in J_i. In particular, finding the matrix $\mathcal{M}_i$ with unit Frobenius norm that minimizes A_i is equivalent to finding the unit vector $\boldsymbol{m}_i$ minimizing $|\mathcal{C}_i \boldsymbol{m}_i|^2$. This is an eigenvalue problem, whose solution is the unit eigenvector of the matrix $\mathcal{C}_i^T \mathcal{C}_i$ associated with the smallest eigenvalue of this matrix.

Let us now fix the matrices $\mathcal{M}_i$ and rewrite the error term associated with the vector $\boldsymbol{P}_j$ $(j = 1, \ldots, n)$ as

$$B_j = |\mathcal{D}_j \boldsymbol{P}_j|^2,$$

where $\mathcal{D}_j$ is the $3m_j \times 4$ matrix obtained by stacking the m_j instances of the matrix $[\boldsymbol{p}_{ij}\times]\mathcal{M}_i$ for all values of i in I_j, and $[\boldsymbol{a}\times]$ denotes the 3×3 skew-symmetric matrix such that $[\boldsymbol{a}\times]\boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$. Thus the unit vector $\boldsymbol{P}_j$ minimizing B_j can be found as the unit eigenvector of the matrix $\mathcal{D}_j^T \mathcal{D}_j$ associated with its smallest eigenvalue. Thus both steps of the alternating algorithm can be reduced to eigenvalue problems.

Table IV. An iterative bilinear algorithm for projective shape from motion.

Compute an initial estimate of the vectors $\boldsymbol{P}_1 \ldots, \boldsymbol{P}_n$ and normalize these vectors. Repeat: (1) for $i = 1$ to m, compute the unit vector $\boldsymbol{m}_i$ that minimizes $\lvert\mathcal{C}_i \boldsymbol{m}_i\rvert^2$; (2) for $j = 1$ to n, compute the unit vector $\boldsymbol{P}_j$ that minimizes $\lvert\mathcal{D}_j \boldsymbol{P}_j\rvert^2$; until convergence.

As mentioned earlier, the proposed algorithm *does not* require all points to be visible in all images: At each iteration, each point can be estimated independently, using all the frames it is observed in. Likewise, each projection matrix can be estimated independently, using all the points visible in the corresponding frame. The initial estimates of the vectors $\boldsymbol{P}_j$ are obtained using the Tomasi-Kanade (1992) affine factorization method, which is equivalent to setting the initial projective depths to 1. In the case of missing data, it is not possible to initialize the algorithm directly through factorization over all the frames. Instead, it is sufficient to compute an approximate reconstruction by applying affine factorization to subsets of the input sequence and registering these subsets with each other. This is described in detail in Section 5.3.

4.3. Convergence

Let us now show that the error function E is guaranteed to converge. As before, let $E^{(0)}$ be the current error value at the beginning of each iteration. During step (1) of the algorithm, we minimize, for $i = 1, \ldots, m$ the error A_i with respect to the vector $\boldsymbol{m}_i$ (or equivalently, the matrices $\mathcal{M}_i$) for $i = 1, \ldots, m$) and thus also minimize the total error $E = \sum_{i=1}^{m} A_i$. It follows that the new value $E^{(1)}$ of the error must be smaller than $E^{(0)}$. During step 2, the error B_j is minimized with

respect to the vectors $\boldsymbol{P}_j$. It follows that the total error $E = \sum_{j=1}^{n} B_j$ is also minimized, and the corresponding error $E^{(2)}$ must be smaller than the current error $E^{(1)}$. This process will eventually converge since the error is bounded below by zero. As in Section 3, the monotonic convergence of the error E to some value E^* does not imply the convergence of the arguments $\mathcal{M}_i$ and $\boldsymbol{P}_j$, and it does not imply that E^* is a local minimum of the error function. The proof of these two properties is given in the Appendix.

5. Implementation and Results

The proposed algorithms have been implemented in MATLAB and tested on synthetic and real image sequences. For robustness, we follow Hartley's suggestion (Hartley, 1995) and preprocess the image data for each frame by centering and scaling the point positions so the average distance from the origin is $\sqrt{2}$ pixel. This transformation is of course undone before measuring the reprojection error.

The next sections compare the two methods proposed in this paper with our MATLAB implementation of the Sturm-Triggs iterative factorization technique and the bundle-adjustment implementation described in (Morris et al., 2000). The abbreviations shown in Table V are used in all plots.

Table V. Abbreviations used for the four algorithms tested in this section.

Symbol	Method
F	Proposed iterative **F**actorization method
B	Proposed iterative **B**ilinear method
ST	**S**turm-**T**riggs iterative factorization method
BA	**B**undle-**A**djustment method

In our tests, the initial guesses for all methods are the same: They correspond to choosing unit projective depths for the two iterative factorization techniques, or equivalently, using a rank-4 version of the Tomasi-Kanade affine factorization algorithm (Tomasi and Kanade, 1992) to estimate the positions of the scene points for the bilinear and bundle-adjustment methods. Accordingly, the initial errors are of course also the same for all approaches.

These errors are measured, for all of the tested algorithms, by the average reprojection error, i.e.,

$$\frac{1}{p}\sum_{ij}\sqrt{\left(u_{ij}-\frac{\boldsymbol{m}_{i1}\cdot\boldsymbol{P}_j}{\boldsymbol{m}_{i3}\cdot\boldsymbol{P}_j}\right)^2+\left(v_{ij}-\frac{\boldsymbol{m}_{i2}\cdot\boldsymbol{P}_j}{\boldsymbol{m}_{i3}\cdot\boldsymbol{P}_j}\right)^2},$$

where the vectors $\boldsymbol{m}_{i1}^T$, $\boldsymbol{m}_{i2}^T$ and $\boldsymbol{m}_{i3}^T$ denote as before the rows of the projection matrix $\mathcal{M}_i$ $(i = 1, \ldots, m)$. In every case, the position of the scene points computed by the tested method is used to estimate the projection matrices and compute the corresponding error.[2]

The next three sections compare the four algorithms based on the average and maximum errors at convergence as well as the number of iterations required to get there. It is important to note that the convergence plots (error vs. iteration number) record the convergence rate of the various algorithms, but they do not (directly) reflect their overall computational cost. The computational complexity of the four algorithms will be discussed in detail in Section 5.5.

5.1. Synthetic Images

In this section, we compare the performance of the various methods on synthetic data. In each trial, thirty points are selected at random within a sphere of radius 100 units; 10 training views of these points were taken by a camera looking directly at the sphere center, with an optical center located at a random point on a surface patch located at a distance of 150 units from the origin and sustending an angle of 30° from the origin. The relatively small size of this patch is chosen to make the structure from motion estimation task challenging since most of the information about the scene structure is embedded in the camera translations. Also, the relatively large range of depths spanned by the observed points in each view produces large perspective distortions. The image size is 512×512 pixels, with a focal length of 256. Zero-mean Gaussian noise with a standard deviation of 1 pixel is added to all input data.

The accuracy of the 3D reconstruction is evaluated in two ways:

- *Using the 2D reprojection error on test images.* In this method, another 10 test views are taken from random points on a sphere of radius 150 units with the same amount of noise as for the training images. These test views are not confined to the small patch from which the training views were taken. Figure 1 shows the results of our experiments with this synthetic data, averaged over 20 trials.

- *Using the 3D error against ground truth.* In this method, we compute a global metric "upgrade" of the projective reconstruction using the method described in (Ponce, 2000).[3] Since the metric reconstruction is recovered up to a global similarity transformation, we align it with the ground-truth

[2] Of course, the mean-squared reprojection error is *not* the error measure E minimized by the reconstruction algorithms tested in this section, except for bundle adjustment. This may slightly bias the results in favor of the latter method.

[3] This method makes minimal assumptions that hold for most real cameras—that is, rectangular pixels and/or known aspect ratio (Heyden and Åström, 1998; Pollefeys, 1999). We refer the interested reader to (Ponce, 2000) for details.

data via linear least squares, then measure the average 3D error as plotted in Figure 2.

As shown by Figure 1, all methods converge in less than 20 iterations and yield comparable average and maximum errors at convergence. The main difference between the four algorithms is their convergence rate: Bundle adjustment requires the fewest iterations to converge, closely followed by the bilinear method proposed in Section 4, then by the iterative factorization introduced in Section 3, and finally the Sturm-Triggs iterative factorization method. As shown in the next section, the same pattern emerges for the rest of our experiments. To be fair, one should observe that the Sturm-Triggs algorithm may converge faster when initialized with projective depths estimated from pairwise epipolar constraints as originally proposed in (Sturm and Triggs, 1996; Triggs, 1996), although the same would apply to the other algorithms.

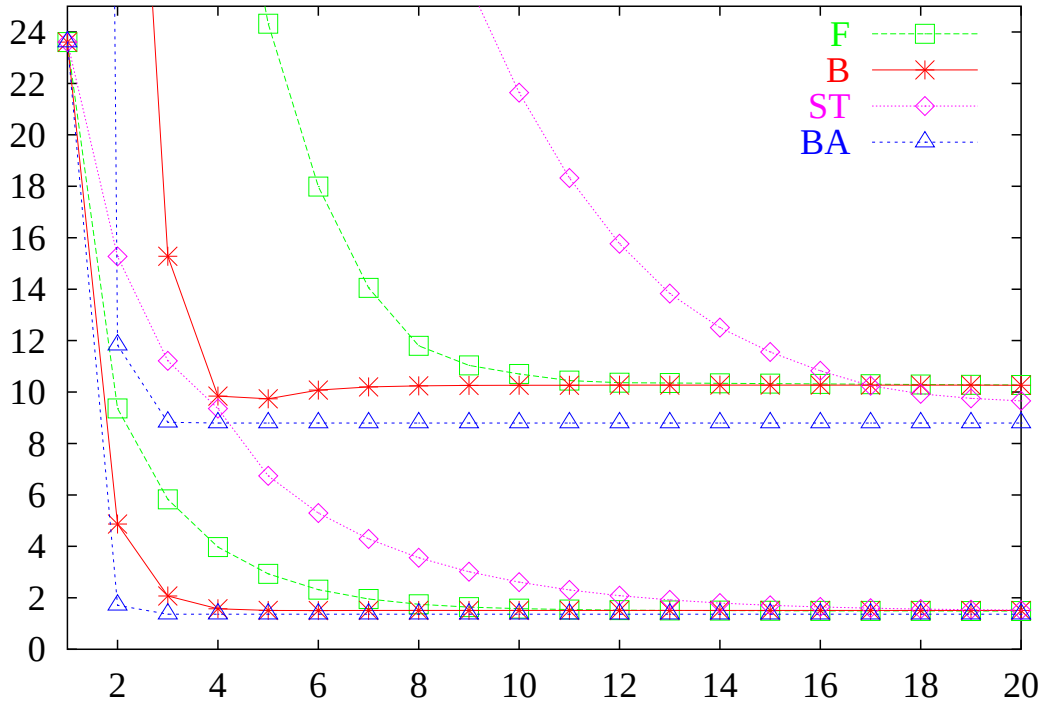


Figure 1. Experiment I: Convergence of the 2D reprojection error on test images for the four algorithms on synthetic data perturbed by additive Gaussian noise with zero mean and a standard deviation of 1 pixel. The mean and maximum errors (in pixels) associated with each algorithm are plotted as a function of the iteration number. See Table V for the abbreviations in the legend.

As shown by Figure 2, the initial 3D errors for all the methods are very poor (corresponding to about a quarter of the size of the sphere from which the 3D points were sampled). This corroborates our claim that our synthetic data is challenging due to large perspective distortions. As was the case for the

2D reprojection error, however, all methods converge in less than 20 iterations and yield comparable 3D error of about 4 units compared with a radius of 100 units for the sphere. Since the errors measured in this experiment are associated with *metric* reconstructions, we also include in Figure 2 (under the label "BA-M") a comparison with the bundle adjustment method of (Morris et al., 2000) set to compute the metric reconstruction directly. As shown by the figure, the performance of the direct method is poor in this case, probably due to local minima. This should not be seen as a failure of the particular bundle-adjustment method used in our experiments (Morris et al., 2000), but rather as further evidence in favor of a "stratified" approach to structure from motion (Faugeras, 1995).

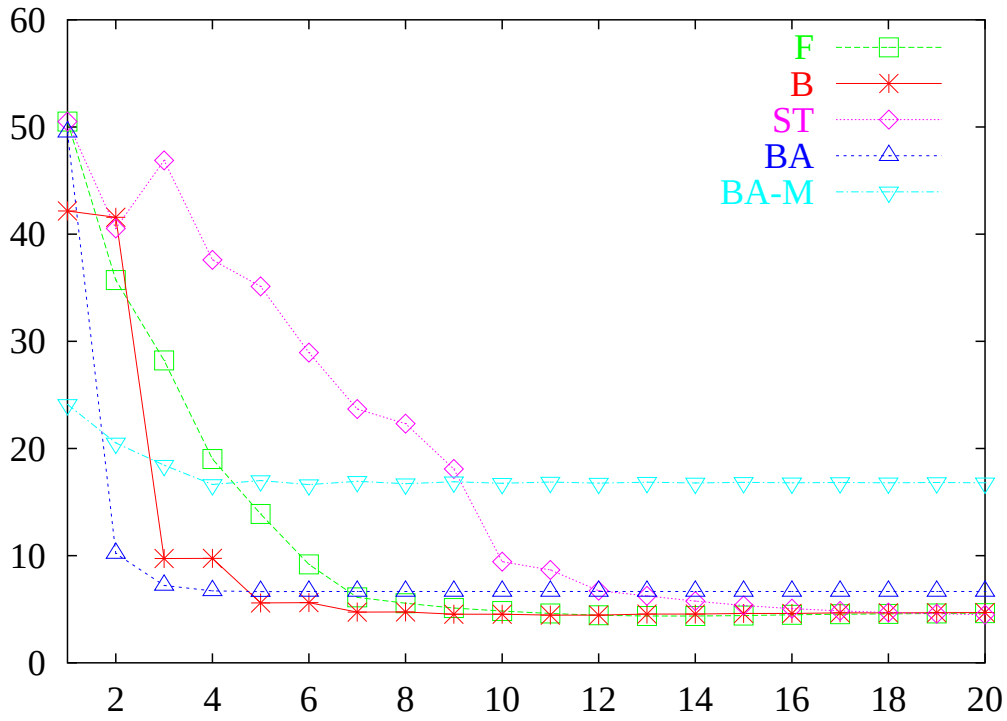


Figure 2. Experiment I (continued): Convergence of the average 3D error for the four algorithms on synthetic data perturbed by additive Gaussian noise with zero mean and a standard deviation of 1 pixel.

5.2. Real Images

Figure 3 shows the average and maximum reprojection errors obtained on the CASTLE data kindly provided by Marc Pollefeys. This data set consists of 20 images of 30 points. Again, all algorithms yield comparable errors upon convergence, the final average error being below one pixel in all cases. The convergence pattern observed before is repeated here.

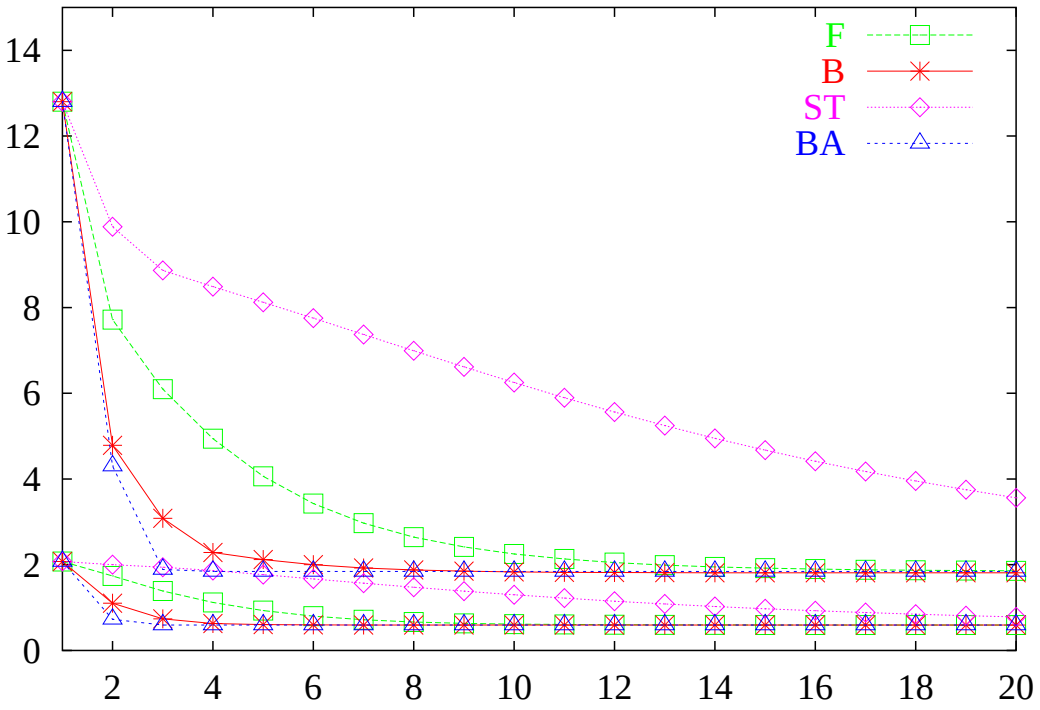


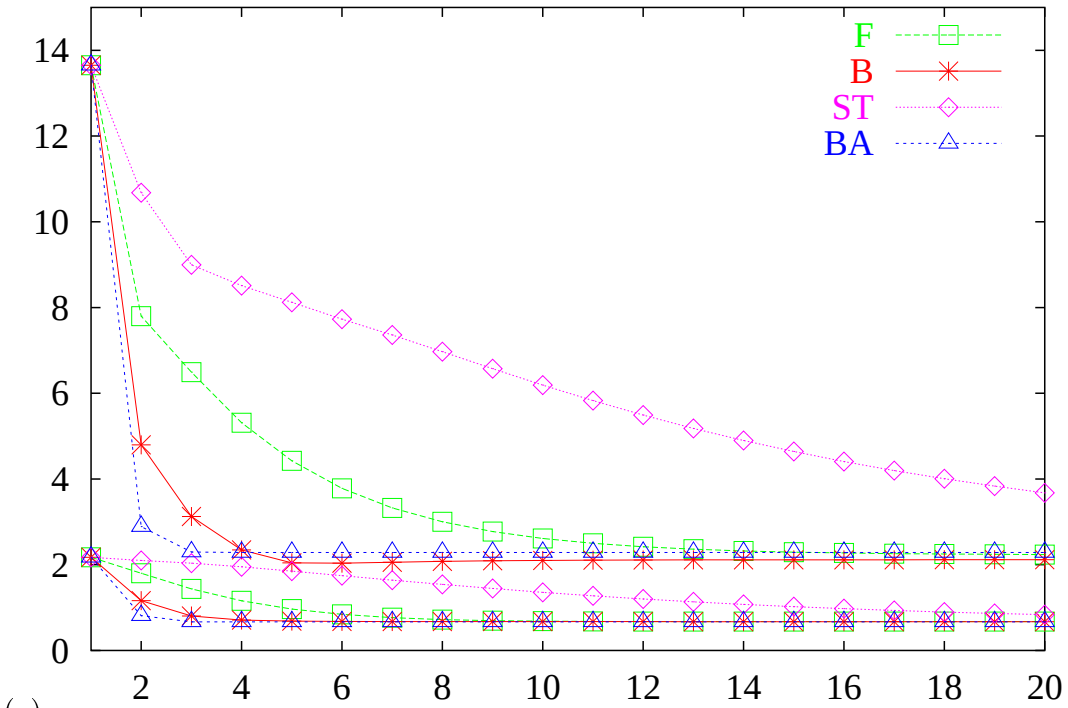
Figure 3. Experiment II: CASTLE data.

We have conducted two experiments to evaluate the extrapolation power of the four methods: In the first one, alternate frames are used for training and testing (Figure 4(a)), while the middle 10 frames are used for training and the outside 10 images for testing in the second experiment (Figure 4(b)). In these two examples, the position of the scene points estimated from the training data is used to estimate the camera positions for each test image (this is easily done using linear least squares, similar to the projection matrix estimation step in the bilinear algorithm of Section 4), from which the image errors can immediately be computed.

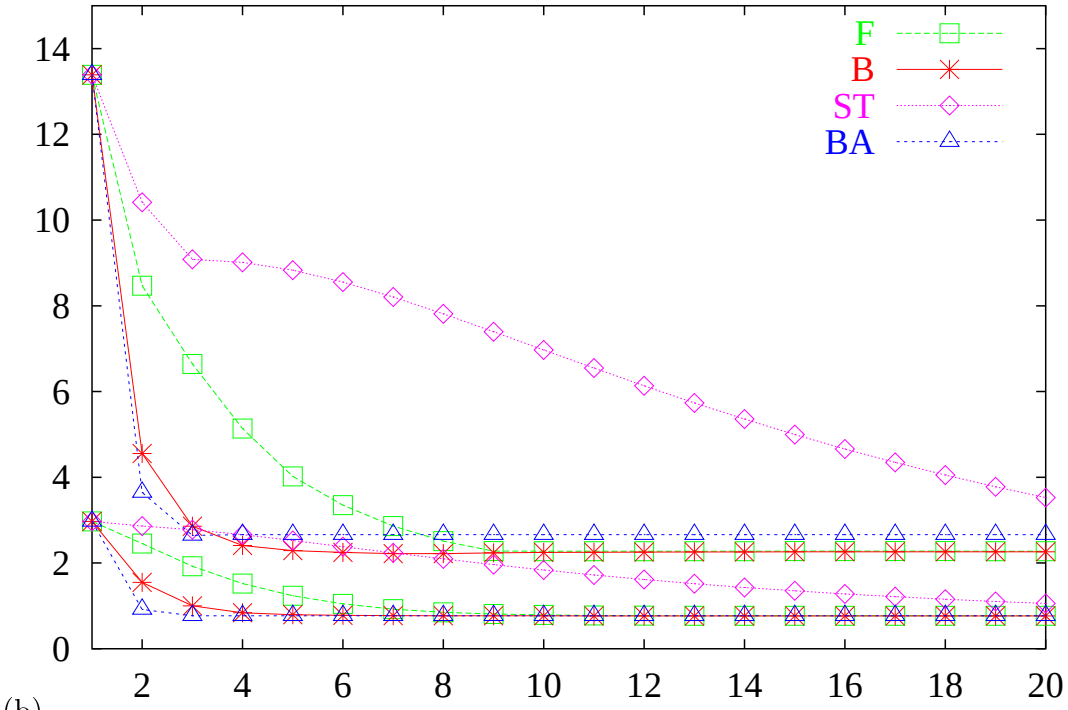
Qualitatively, the results are similar to those obtained using all of the data for training and testing, confirming that the four methods are capable of predicting with a good accuracy views that are not part of the training data.

5.3. Real Images with Missing Data

As mentioned earlier, the bilinear method proposed in Section 4 can handle missing data. This is of course also the case for bundle-adjustment techniques, but factorization algorithms, on the other hand, normally require all points to



(a)



(b)

Figure 4. Experiment III: Incomplete CASTLE data: (a) alternate frames are used for training and testing; (b) the middle ten frames are used for training and the remaining ones are used for testing.

be visible in all images.[4] We have compared the bilinear and bundle-adjustment algorithms on the WILSHIRE data kindly provided by Andrew Fitzgibbon and Andrew Zisserman (Figure 5).

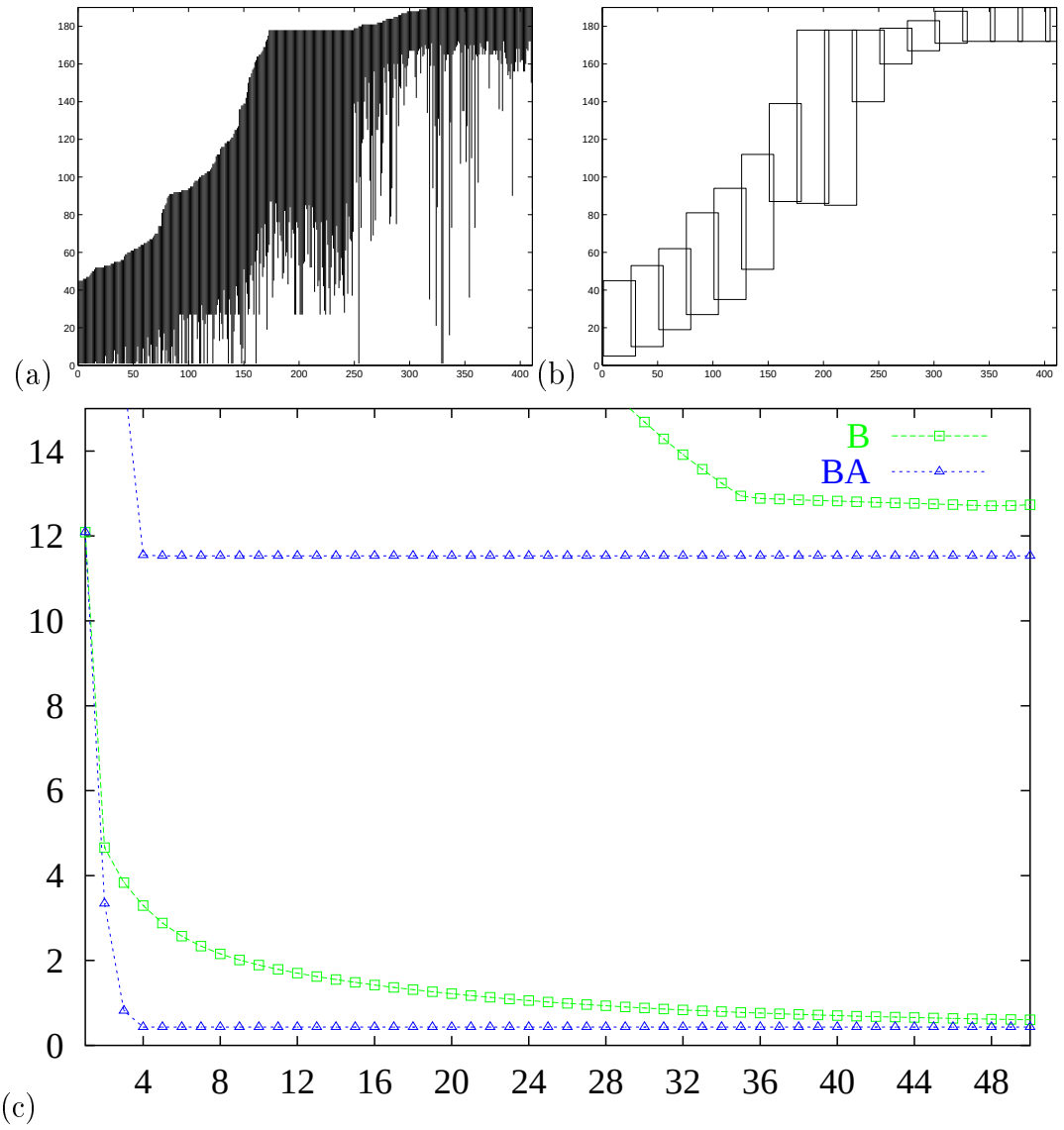


Figure 5. Experiment IV: WILSHIRE data: (a) the frames where each of the 411 scene points are visible are indicated by vertical line segments; (b) maximal 30-point rectangles where the same points are visible in all frames; (c) comparison of bundle adjustment with the bilinear algorithm.

This data set consists of 411 points and 190 frames. As shown by Figure 5(a), not all points are visible in all images. For initialization, we have divided the

[4] See, however (Tomasi and Kanade, 1992; Shum et al., 1995; Jacobs, 1997) for various extensions of factorization that are capable, at least to some extent, of handling missing data.

data into consecutive blocks of 30 points with a 5-point overlap, and used the Tomasi-Kanade method in the maximal full rectangle of each block (Figure 5(b)) to compute an affine reconstruction of the corresponding points. The successive reconstructions have been registered by estimating the affine transformations separating the positions of the points common to pairs of consecutive blocks. As shown by Figure 5(c), the initial errors are much larger in this case, and it takes the bilinear algorithm about 50 iterations to converge, as opposed to only 4 iterations for bundle adjustment.

A theoretical cost comparison of the bundle-adjustment and bilinear algorithms will be given in Section 5.5. In the mean time, it is worth giving some empirical timing results on this fairly large data set: The code for both algorithms is written in MATLAB (plus some compiled C code for bundle adjustment). On an 800MHZ Pentium III, running 50 iterations of the bilinear algorithm takes only 4mn on the WILSHIRE data, while 4 iterations of bundle adjustment take 3 hours.

5.4. A Large-Scale Modeling Experiment

Here, we apply the bilinear algorithm to the problem of constructing three-dimensional object models from image sequences. The test data set consists of 192 frames of a teddy bear acquired by a hand-held camera and (roughly) spanning a 360° range of viewpoints. A total of 2480 features are tracked automatically throughout the sequence using Birchfield's implementation (Birchfield, 1998) of the Kanade-Lucas-Tomasi (KLT) feature tracker (Tomasi and Kanade, 1992). As before, not all points are visible in all images, and the tracking program is set to keep about 1000 features visible at all times by acquiring new points when necessary. To improve the robustness of our algorithm, point tracks shorter than 10 frames are rejected, and the last 5 images of each track are rejected as well since the tracker is often unreliable just before a feature disappears. The structure of the data matrix is shown in Figure 6 using the same conventions as in Figure 5. The data is divided into consecutive blocks of 100 features with a minimum overlap of 50 points between consecutive blocks. To increase the overlap between consecutive blocks, the points are sorted so that those appearing early and disappearing quickly appear first in each block.

It was simply impossible to run the bundle-adjustment method in a reasonable time on such a large data set. The bilinear algorithm converges in about 50 iterations (Figure 6(c)) and 2mn 20s of CPU time on a 2.0 GHz Pentium IV. After convergence, the projective reconstruction is converted into a metric one using the technique described in (Ponce, 2000). Three views of the reconstructed teddy bear are shown in Figure 7. In each case, the reconstruction is displayed (from left to right) as a point cloud, a shaded triangulation obtained by using *alpha shapes* (Edelsbrunner and Mücke, 1994) to interpolate the estimated points, and a texture-mapped view of this triangulation. This example shows the feasibility

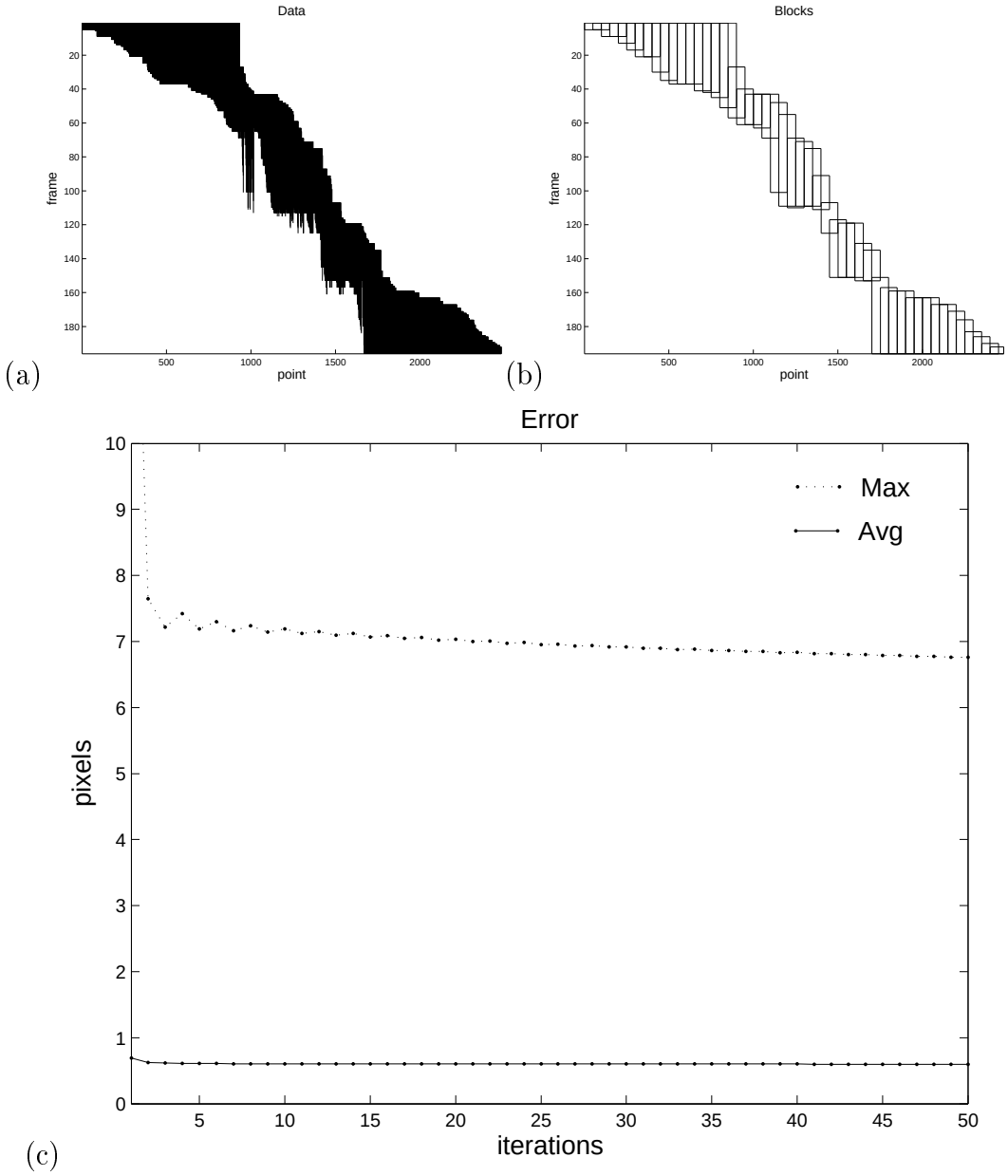


Figure 6. Experiment V: TEDDY BEAR data: (a) the frames where each of the 2480 scene point are visible are indicated by vertical line segments; (b) maximal 100-point rectangles where the same points are visible in all frames; (c) reprojection errors for the bilinear algorithm.

of using the bilinear algorithm in a realistic modeling setting that includes large numbers of features and frames as well as missing data.

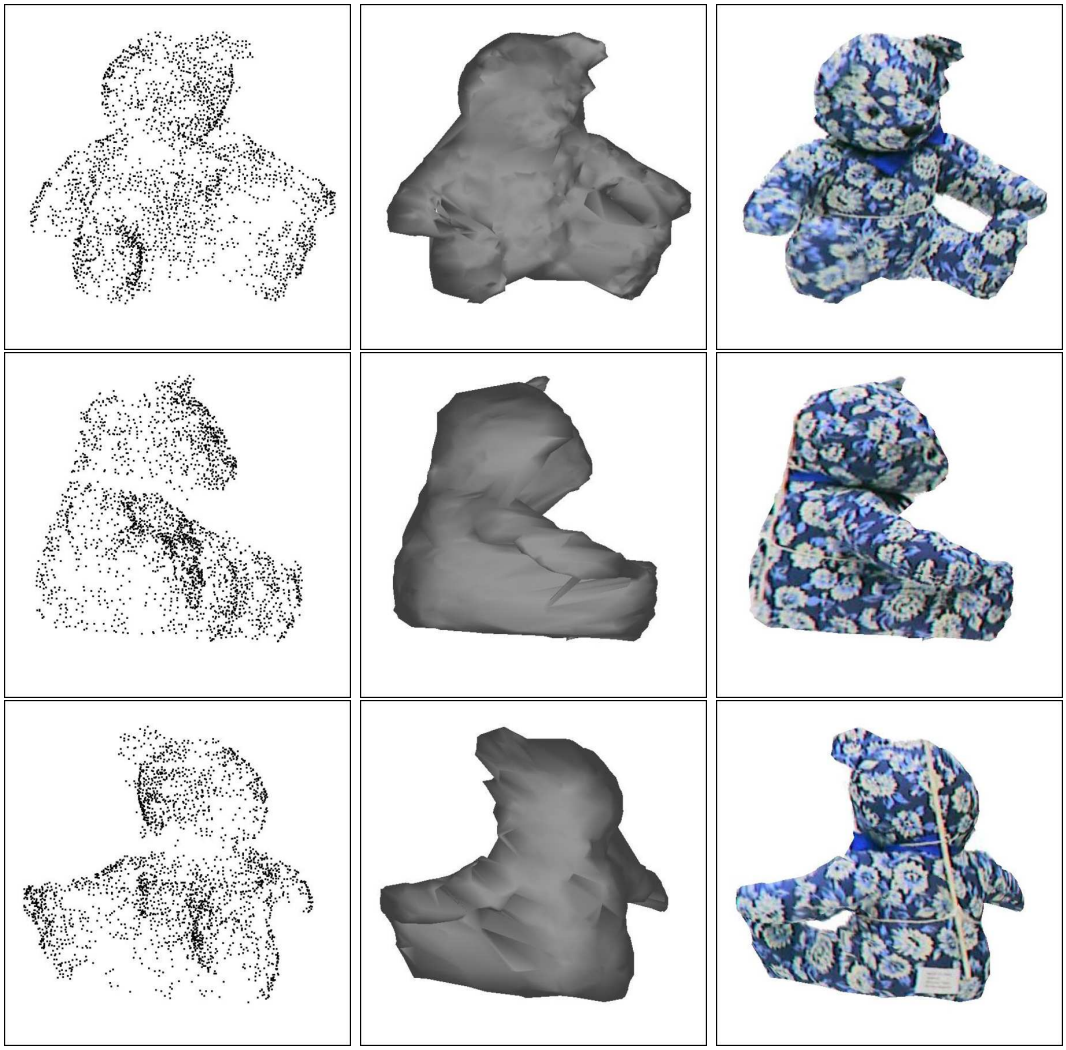


Figure 7. Three views of the metric reconstruction of the teddy bear: From left to right: point cloud; surface reconstruction; texture-mapped surface.

5.5. Theoretical Cost Comparison

This section compares the computational costs of the four algorithms. Let us start with the cost per iteration (Table VI): The singular value decomposition of a $k \times l$ matrix can be computed in time $O(kl \min(k, l))$ (Golub and Van Loan, 1996).[5] It follows that the cost of each iteration of either one of the factorization techniques discussed in this paper is dominated by the SVD step and costs $O(mn \min(3m, n))$. Each iteration of the proposed bilinear algorithm, on the other hand, takes $O(mn)$ time, since the computation of the matrices $\mathcal{C}_i^T \mathcal{C}_i$ takes $O(n)$ time, the computation of the matrices $\mathcal{D}_j^T \mathcal{D}_j$ takes $O(m)$ time, and the computation of the eigenvectors and eigenvalues of these 12×12 and 4×4

[5] The hidden constants depend on the actual implementation.

matrices takes constant time.[6] In contrast, each iteration of a Gauss-Newton or Levenberg-Marquart solution to bundle adjustment has cost $O((m+n)^3)$ (Hartley and Zisserman, 2000).

Table VI. Comparison of the computational costs per iteration of the various methods as a function of m (number images) and n (number of points). See text for details.

Method	Cost per Iteration (Flops)
Factorization Methods	$O(mn \min(3m, n))$
Bilinear Algorithm	$O(mn)$
Bundle Adjustment	$O((m+n)^3)$

The performance of the four algorithms can in principle be improved by careful implementation: Fixed-rank approximations of the singular value decomposition $\mathcal{U}\mathcal{W}\mathcal{V}^T$ of a matrix only compute the portions of $\mathcal{U}$ and $\mathcal{V}$ associated with a fixed number of singular values (Triggs, 1996), and they can used to speed up the factorization and bilinear algorithms. Likewise, sparse bundle-adjustment algorithms can be used when the number of points visible in each frame and the number of frames where each point is visible are both bounded by k, with a complexity reduced to $O((m+n)^2 k)$. The bilinear algorithm also benefits from sparsity of course, with a cost of $O((m+n)k)$ under the same assumptions.

The cost per iteration of an algorithm is of course only a partial measure of its performance. Its rate of convergence is just as important. The Gauss-Newton and Levenberg-Marquardt algorithms used in most implementations of bundle adjustment (the latter is used in the program from (Morris et al., 2000) used in our experiments) can be shown to have a quadratic convergence rate in favorable cases (Gill and Murray, 1974; Heath, 1997).[7] The convergence rate of the factorization and bilinear techniques has not been assessed yet, but as for other resection-intersection methods (Triggs et al., 2000), it is likely to be linear. Empirically, bundle adjustment converges faster than the factorization techniques in all of our experiments, and notably faster than the bilinear algorithm in the WILSHIRE experiment. However, it is worth noting that the empirical results presented in the previous sections suggest that the low cost of the bilinear iterations far outweigh the fast convergence of bundle adjustment in appropriate settings. In particular, for large problems, the bilinear algorithm

[6] Solving the corresponding homogeneous linear least-squares using singular value decomposition instead of considering them as eigenvalue problems has the same time complexity.

[7] This is due to the fact that they approximate Newton's method, whose convergence rate is quadratic. However, because both the Gauss-Newton algorithm and its Levenberg-Marquardt variant ignore the second-derivative terms involved in Newton's method, they may only converge linearly or even not converge at all when improperly initialized or when the residuals at convergence are large.

provides an efficient way to compute a reasonable projective reconstruction that can be passed as input to a couple of iterations of bundle adjustment.

6. Discussion

We have presented two simple, provably-convergent iterative algorithms for projective structure and motion estimation from multiple images. They have been implemented and experiments involving both synthetic and real image sequences have demonstrated that they compare favorably to the Sturm-Triggs iterative factorization algorithm and can both be used as an efficient and accurate initialization step before bundle adjustment.

A crucial advantage of the proposed iterative bilinear method, compared to its factorization-based counterparts, is that it can readily handle missing data since it does not require all the points to be visible in all frames. This provides a practical solution for large-scale problems in which the missing data issue cannot be sidestepped.

The bilinear algorithm proposed in this paper can be thought of as an *alternation* technique (Triggs et al., 2000), that interleaves structure and motion estimation steps until convergence. Alternation approaches to structure from motion are sometimes maligned for their reputed inefficiency and slow convergence near local minima (see the discussion in (Triggs et al., 2000) for example). However, the experiments presented in Section 5 seem to indicate an excellent rate of convergence for the bilinear algorithm on both synthetic and real images, except in the WILSHIRE experiment where bundle adjustment clearly converges faster. Even in that case, however, the gain in running time per iteration far outweighs the loss in convergence rate.

The iterative factorization algorithms of Sturm and Triggs (1996), Heyden *et al.* (1997; 1999), and indeed the convergent iterative factorization method proposed in this paper, are in fact also alternation techniques since they interleave steps where $\mathcal{M}$ and $\mathcal{P}$ are estimated with steps where the projective depths are estimated instead. The results presented in Section 5 also demontrate their good behavior on real and synthetic image sequences. In our mind, these algorithms are not intended to replace bundle adjustment, but they can be used to quickly find good initial estimates of the point positions and projection matrices before starting the bundle adjustment iterations.

Let us conclude by examining once again the issue of trivial solutions to the structure-from-motion problem, focussing on the bilinear algorithm. This algorihm minimizes the error E under the constraints $|\mathcal{M}_i|^2 = 1$ and $|\boldsymbol{P}_j|^2 = 1$. This choice obviously avoids the all-zero solution, but other trivial solutions may in principle be found (e.g., the solution mentioned in the introduction where all parameters z_{ij} are zero with $\mathcal{M}_i = \mathcal{M}_0$, $\boldsymbol{P}_j = \boldsymbol{P}_0$, and $\boldsymbol{P}_0$ in the kernel of the non-zero matrix $\mathcal{M}_0$). Other constraints on the matrices $\mathcal{M}_i$ and the vectors $\boldsymbol{P}_j$ could be used instead: For example, enforcing $\sum_{ij} |\mathcal{M}_i \boldsymbol{P}_j|^2 = 1$ avoids the all-

zero solution. On the other hand, this version of the algorithm is still susceptible to (different) trivial solutions. For example, we could take $\mathcal{M}_2 = \ldots = \mathcal{M}_m = 0$ (and thus $z_{ij} = 0$ for $i > 1$), pick an arbitrary non-zero matrix for $\mathcal{M}_1$, and choose the vectors $\boldsymbol{P}_j$ as solutions of $\boldsymbol{p}_{1j} = \mathcal{M}_1 \boldsymbol{P}_j$. Note that this is also a potential zero minimum for the proposed factorization algorithm.

In fact, it appears that trivial solutions will in principle be possible for *any* algorithm based on Equation (1) whose convergence can be proven using the methodology used in the appendix: The overall convergence proof scheme relies critically on the compactness of the constraint space (see (Luenberger, 1984, Chapter 6) for examples of non-convergence of seemingly trivial examples if the space is not compact). But it is only possible to guarantee that trivial solutions will not occur by imposing constraints of the form $z_{ij} \neq 0$ for all values of i and j, which amounts to defining non-compact regions associated with the corresponding open sets in parameter space. Thus, it appears that the transformation from the original perspective equation (1) to its linearized form $z_{ij}\boldsymbol{p}_{ij} = \mathcal{M}_i \boldsymbol{P}_j$ will either *always* introduce the possibility of non-physical solutions, or yield algorithms whose convergence cannot (yet) be guaranteed.

Empirically, the algorithms proposed in this paper have never converged to a trivial solution in our experiments. A plausible explanation is that we start the iterations at a solution that minimizes an error function that is closely related to E, namely the affine reconstruction error $E' = \sum_{i,j} |\boldsymbol{p}_{ij} - \mathcal{M}_i \boldsymbol{P}_j|^2$. It is worth noting that an iterative approach to affine structure from motion related to our bilinear algorithm has recently been proposed by Hartley and Schaffalitzky (2002) to handle missing data (see (Rother and Carlsson, 2002) and (Brand, 2002) for related work). This affine algorithm can be shown to converge to the *global* minimum of the corresponding error function when there is no missing data. It remains to be determined wether our projective formulation shares this desirable property.

Appendix

We showed in Sections 3 and 4 that the error function converges monotonically for both the bilinear and the factorization algorithms. We now show that the projection matrices and the points also converge, that the error converges to a local minimum, and that the two algorithms are globally convergent, that is, convergence to a solution is guaranteed irrespective of the starting point. The proof relies on a powerful general result from numerical analysis, the global convergence theorem (Zangwill, 1969) described in detail in (Luenberger, 1984, Chapter 6). See, for example, (Lu and Hager, 2000) for its application to the study of the convergence of pose-estimation algorithms in computer vision.

The Global Convergence Theorem

The global convergence theorem (or GCT) provides a general framework for studying the convergence properties of iterative optimization algorithms. The GCT represents algorithms as point-to-set mappings, associating with each input value a set of compatible outputs. In the actual implementation of an algorithm, a single output is of course chosen out of the set of possible outputs. This is relevant to the two algorithms proposed in this paper because they rely on either computing the singular value decomposition of a matrix or solving a generalized eigenvalue problem, and the matrices involved may have eigenvalues of multiplicity greater than one, forcing arbitrary choices among the corresponding eigenvectors.

In this setting, a convergent algorithm is a *closed* mapping $A : X \to \mathcal{P}(Y)$ from a topological space X of input parameters to the set $\mathcal{P}(Y)$ formed by all subsets of a topological space Y of output values.

DEFINITION 1. *A mapping $A : X \to \mathcal{P}(Y)$ is said to be closed at a point x in X when the convergence of a sequence x_k in X to x and the convergence of a sequence y_k in $A(x_k)$ to some point y in Y imply that y is an element of $A(x)$. A is said to be closed when it is closed at every point in X.*

THEOREM 1. (GCT). *Consider a topological space X, a solution set $\Gamma \subset X$, a mapping $A : X \to \mathcal{P}(X)$, and a sequence of points x_k in X such that for $k \geq 0$, x_{k+1} belongs to $A(x_k)$. When*

I. *the points x_k ($k \geq 0$) belong to a compact set S;*

II. *there exists a continuous* descent function *$Z : X \to \mathbb{R}$ such that*

(a) *if $x \notin \Gamma$, then $Z(y) < Z(x)$ for all y in $A(x)$,*
(b) *if $x \in \Gamma$, then $Z(y) \leq Z(x)$ for all y in $A(x)$;*

III. *the mapping A is closed at every point outside Γ;*

the limit of any convergent subsequence of x_k belongs to Γ.

Strictly speaking, the GCT only guarantees the convergence of subsequences to the solution set, which allows for x_k to alternate between multiple solutions. In practice, an arbitrary choice among these solutions is used as the final output of the algorithm since they all yield the same error. The existence of convergent subsequences follows directly from the compactness assumption.

We will need three lemmas in order to apply the GCT to our problem. The first one shows that the mapping that associates with a matrix its closest rank-4 factorization in the sense of the Frobenius form is closed. To be correct, this statement needs to be rephrased in the terminology of point-to-set mappings because an entire family of such factorizations may exist (in addition to the inherent projective ambiguity of the factorization), corresponding to singular

values with multiplicities greater than one and the different choices of orthonormal bases for the eigenspaces of $\mathcal{I}\mathcal{I}^T$ and $\mathcal{I}^T\mathcal{I}$ associated with these singular values.

LEMMA 1. *The mapping A that associates with a $3m \times n$ matrix $\mathcal{I}$ the set $A(\mathcal{I})$ of all matrix pairs $(\mathcal{M}^*, \mathcal{P}^*)$ in $\mathbb{R}^{3m\times 4} \times \mathbb{R}^{4\times n}$ such that*

$$\forall \mathcal{M} \in \mathbb{R}^{3m\times 4},\ \forall \mathcal{P} \in \mathbb{R}^{4\times n} \quad |\mathcal{I} - \mathcal{M}^*\mathcal{P}^*|^2 \leq |\mathcal{I} - \mathcal{M}\mathcal{P}|^2$$

is closed.

Proof. Let us consider a sequence $\mathcal{I}_k \to \mathcal{I}$ of $3m \times n$ matrices and a sequence of matrix pairs $(\mathcal{M}_k^*, \mathcal{P}_k^*)$ associated with $\mathcal{I}_k$ by the mapping A. Let us assume further that the sequences $\mathcal{M}_k^*$ and $\mathcal{P}_k^*$ converge toward two matrices $\mathcal{M}^*$ and $\mathcal{P}^*$. We can write

$$\forall \mathcal{M} \in \mathbb{R}^{3m\times 4},\ \forall \mathcal{P} \in \mathbb{R}^{4\times n} \quad |\mathcal{I}_k - \mathcal{M}_k^*\mathcal{P}_k^*|^2 \leq |\mathcal{I}_k - \mathcal{M}\mathcal{P}|^2.$$

Since the norm is a continuous function, we obtain by passing to the limit on both sides:

$$\forall \mathcal{M} \in \mathbb{R}^{3m\times 4},\ \forall \mathcal{P} \in \mathbb{R}^{4\times n} \quad |\mathcal{I} - \mathcal{M}^*\mathcal{P}^*|^2 \leq |\mathcal{I} - \mathcal{M}\mathcal{P}|^2.$$

Therefore, $(\mathcal{M}^*, \mathcal{P}^*)$ belongs to $A(\mathcal{I})$. ∎

LEMMA 2. *The mapping A that associates with two matrices $\mathcal{U}$ and $\mathcal{V}$ and the scalar $\gamma > 0$ the set $A(\mathcal{U}, \mathcal{V}, \gamma)$ of all vectors $\boldsymbol{x}^*$ such that $|\mathcal{V}\boldsymbol{x}^*|^2 = \gamma^2$ and*

$$\forall \boldsymbol{x}, \quad |\mathcal{V}\boldsymbol{x}|^2 = \gamma^2 \Longrightarrow |\mathcal{U}\boldsymbol{x}^*|^2 \leq |\mathcal{U}\boldsymbol{x}|^2$$

is closed.

Proof. Let us consider the sequences $\mathcal{U}_k \to \mathcal{U}$, $\mathcal{V}_k \to \mathcal{V}$, $\gamma_k \to \gamma$, with $\gamma_k > 0$ and $\gamma > 0$, and an additional sequence $\boldsymbol{x}_k^*$ of vectors that minimize $|\mathcal{U}_k\boldsymbol{x}|^2$ under the constraint $|\mathcal{V}_k\boldsymbol{x}|^2 = \gamma_k^2$, *i.e.*, $\boldsymbol{x}_k^*$ belongs to $A(\mathcal{U}_k, \mathcal{V}_k, \gamma_k)$ (there may of course be several such vectors $\boldsymbol{x}_k^*$ when the corresponding generalized eigenvalue has multiplicity greater than one). Let us assume further that the sequence $\boldsymbol{x}_k^*$ converges to some limit $\boldsymbol{x}^*$. Since the norm is a continuous function, we have $|\mathcal{V}\boldsymbol{x}^*|^2 = \gamma^2$. By continuity, if $\boldsymbol{x}$ is a vector such that $\mathcal{V}\boldsymbol{x} \neq 0$, there exists some index $k_{\boldsymbol{x}}$ such that for any $k > k_{\boldsymbol{x}}$, $\mathcal{V}_k\boldsymbol{x} \neq 0$. In particular, we can define, for $k > k_{\boldsymbol{x}}$, the vector $\hat{\boldsymbol{x}}_k = (\gamma/|\mathcal{V}_k\boldsymbol{x}|)\boldsymbol{x}$, so $|\mathcal{V}_k\hat{\boldsymbol{x}}_k|^2 = \gamma^2$. In particular,

$$\forall \boldsymbol{x}, \quad \mathcal{V}\boldsymbol{x} \neq 0 \Longrightarrow \exists k_{\boldsymbol{x}}, \forall k > k_{\boldsymbol{x}}, \quad |\mathcal{U}_k\boldsymbol{x}_k^*|^2 \leq |\mathcal{U}_k\hat{\boldsymbol{x}}_k|^2 = \gamma^2 \frac{|\mathcal{U}_k\boldsymbol{x}|^2}{|\mathcal{V}_k\boldsymbol{x}|^2}.$$

Passing to the limit on both sides, we obtain

$$\forall \boldsymbol{x}, \quad \mathcal{V}\boldsymbol{x} \neq 0 \Longrightarrow |\mathcal{U}\boldsymbol{x}^*|^2 \leq \gamma^2 \frac{|\mathcal{U}\boldsymbol{x}|^2}{|\mathcal{V}\boldsymbol{x}|^2};$$

and restricting this inequality to the vectors $\boldsymbol{x}$ such that $|\mathcal{V}\boldsymbol{x}|^2 = \gamma^2$ shows that $\boldsymbol{x}^*$ minimizes $|\mathcal{U}\boldsymbol{x}|^2$ under the constraint $|\mathcal{V}\boldsymbol{x}|^2 = \gamma^2$. Therefore, $\boldsymbol{x}^*$ belongs to $A(\mathcal{U}, \mathcal{V}, \gamma)$. ∎

Finally, we state without proof one lemma from (Luenberger, 1984) that spells out some of the conditions under which mappings are closed:

LEMMA 3.
(a) A mapping A is closed if it is a continuous function of its input.
(b) The composition $B \circ A$ of a point-to-point mapping A continuous at x with a point-to-set mapping B closed on $A(x)$ is closed at x.
(c) The composition $B \circ A$ of two closed mappings A and B is closed if the range of A is compact.

Convergence of the Proposed Factorization Algorithm

Let us denote the $m \times n$ matrix formed by all projective depths z_{ij} by $\mathcal{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$. We show in this section that the iterative factorization algorithm proposed in Section 3 converges globally to some limit $(\mathcal{M}^*, \mathcal{P}^*, \mathcal{Z}^*)$, i.e., that it finds a local minimum E^* of the objective function E starting from any initial value $(\mathcal{M}^0, \mathcal{P}^0, \mathcal{Z}^0)$.

To apply the GCT, we need to define the appropriate parameter space X, the solution set Γ, the compact set S, the descent function $Z : X \to \mathbb{R}$ and the mapping A associated with our algorithm. Let us represent the set of all triples $(\mathcal{M}, \mathcal{P}, \mathcal{Z})$ by $\mathbb{R}^{12m+4n+mn} = \mathbb{R}^{12m} \times \mathbb{R}^{4n} \times \mathbb{R}^{m \times n}$, endowed with the Euclidean norm, and define $X = S$ to be the variety of $\mathbb{R}^{12m+4n+mn}$ formed by its elements that satisfy: (a) $|\mathcal{Q}_j \boldsymbol{z}_j|^2 = 1$ for $j = 1, \ldots, n$, and (b) $\mathcal{M}$ and $\mathcal{P}$ minimize $|I - MP|^2$, endowed with the induced topology.

The Z part of S is obviously compact owing to condition (a) (recall that $\mathcal{Q}_j$ is a constant matrix for all $j = 1, .., m$). The $(\mathcal{M}, \mathcal{P})$ part of S is obtained by taking the SVD $\mathcal{I} = \mathcal{U}\mathcal{W}\mathcal{V}^T$, where $\mathcal{U}$ and $\mathcal{V}$ are orthogonal matrices that live in a compact space. We always have $|\mathcal{W}| = |\mathcal{I}| = n$ before factorization due to the normalization of each column of $\mathcal{I}$. Therefore, the space of matrices $\mathcal{W}$ computed from the normalized data matrices is also compact. Since the matrices $\mathcal{M}$ and $\mathcal{P}$ are continuous functions of the matrices $\mathcal{U}$, $\mathcal{V}$, and $\mathcal{W}$ (for example, $\mathcal{M} = \mathcal{U}\mathcal{W}^{1/2}$ and $\mathcal{P} = \mathcal{W}^{1/2}\mathcal{V}^T$), they also form a compact space. Therefore the combined space S is compact.

The mapping A associated with each iteration of the algorithm maps the current estimate of $\mathcal{M}$, $\mathcal{P}$ and $\mathcal{Z}$ onto the next one computed by steps 3(a)

and 3(b) of the iteration. S is compact, and since the points $x_k = (\mathcal{M}^k, \mathcal{P}^k, \mathcal{Z}^k)$ generated by the iterations of the algorithm belong to S, assumption I of the GCT holds.

We define Z as the restriction of the error function $E : \mathbb{R}^{12m+4n+mn} \to \mathbb{R}$ to S, and take Γ to be the set of local minima of Z. We showed in Section 3 that the error E decreases at each step of the algorithm, i.e., $E(A(x)) \leq E(x)$ for any x in $X = S$, and in particular for any x in Γ, thus II(b) is satisfied.

To prove that II(a) holds as well, let us assume that $E(A(x)) = E(x)$, and show that x must lie in Γ. Let us first consider step 3(a) of the algorithm. This step minimizes the error B_j with respect to $\mathcal{M}$ and $\mathcal{P}$, thus the gradient of E with respect to the corresponding parameters must be zero. Now step 3(b) does not change $\mathcal{M}$ and $\mathcal{P}$ but minimizes E with respect to $\mathcal{Z}$. Thus the gradient of E with respect to these parameters is zero as well. It follows that the gradient of E in x is zero and that we are at a critical point of this function.

We have shown that assumptions I and II of the GCT hold. To finish our proof, let us finally decompose A in a set of elementary mappings and show that A is closed thus III holds as well. The algorithm can be decomposed into four mappings:

- A_1 maps the scaled measurement matrix $\mathcal{I}$ onto $\mathcal{M}, \mathcal{P}$ through a rank-four factorization.

- A_2 maps $\mathcal{M}$ and $\mathcal{P}$ onto the corresponding matrices $\mathcal{R}_j$.

- A_3 maps $\mathcal{R}_j$ onto the eigenvector $\boldsymbol{z}_j$ corresponding to the largest eigenvalue of the generalized eigenproblem $\mathcal{R}_j^T \mathcal{R}_j \boldsymbol{z}_j = \lambda \mathcal{Q}_j^T \mathcal{Q}_j \boldsymbol{z}_j$. Note that $\mathcal{Q}_j$ remains constant throughout the iterations and thus does not need to be included in the definition of the mappings.

- A_4 updates the value of $\mathcal{I}$.

The mappings A_1 and A_4 are continuous functions of their inputs and hence closed (Lemma 3(a)). The mappings A_2 and A_3 are closed according to Lemmas 1 and 2 respectively. Since A_1 is closed and A_2 is continuous, $A_2 \circ A_1$ is closed according to Lemma 3(b). Similarly, $A_4 \circ A_3$ is closed for the same reason.

Finally, the range of A_2 is compact because it is the set of matrices of the form $\mathcal{U}^T \mathcal{Q}_j$, where $\mathcal{U}$ is column-orthogonal and $\mathcal{Q}_j$ is a constant matrix. Therefore the range of $A_2 \circ A_1$ is also compact, and $A = (A_4 \circ A_3) \circ (A_2 \circ A_1)$ is closed according to Lemma3(c). Therefore, assumption III of the GCT holds and the proposed factorization algorithm satisfies the all three conditions of the GCT and is therefore globally convergent.

Convergence of the Proposed Bilinear Algorithm

Let us now show that the iterative bilinear algorithm proposed in Section 4 converges globally to some solution $(\mathcal{M}^*, \mathcal{P}^*)$, i.e., that it finds a local minimum E^* of the objective function E starting from any initial value $(\mathcal{M}^0, \mathcal{P}^0)$.

To apply the GCT, we need again to define the appropriate parameter space X, the solution set Γ, the compact set S, the descent function $Z : X \to \mathbb{R}$ and the mapping A associated with our algorithm. Let us represent the set of all pairs $(\mathcal{M}, \mathcal{P})$ by $\mathbb{R}^{12m+4n} = \mathbb{R}^{12m} \times \mathbb{R}^{4n}$, endowed with the Euclidean norm, and define $X = S$ to be the variety of $\mathbb{R}^{12m+4n}$ formed by the matrices $\mathcal{M}_i$ and vectors $\boldsymbol{P}_j$ such that $|\mathcal{M}_i|^2 = 1$ and $|\boldsymbol{P}_j|^2 = 1$, endowed with the induced topology. The mapping A associated with each iteration of the algorithm maps the current estimate of $\mathcal{M}$ and $\mathcal{P}$ onto the next one computed by steps 1 and 2 of one iteration (Table IV). S is compact, and since the points $x_k = (\mathcal{M}^k, \mathcal{P}^k)$ generated by the iterations of the algorithm belong to S, assumption I of the GCT holds.

We define Z as the restriction of the error function $E : \mathbb{R}^{12m+4n} \to \mathbb{R}$ to S, and take Γ to be the set of critical points of Z, or equivalently

$$\Gamma = \{x \in X \mid \forall u \in T(x),\ \nabla E(x) \cdot u = 0\},$$

where $T(x)$ denotes the tangent space to S at x, and $\nabla E(x)$ denotes the gradient of E with respect to the $12m + 4n$ coordinates of x. Note that Γ includes the local minima as well as local maxima and saddle points. The former will never be found by our algorithm since the error decreases at each step, and the latter can be ignored in practice since unless we are already exactly on top of a saddle point, reaching a saddle point is prone to small perturbations (Mahamud et al., 2001).

Let us now show that the hypotheses II(a) and II(b) of the GCT are satisfied, i.e., that Z is indeed a descent function. For clarity, the proof is outlined considering A as a point-to-point mapping, i.e., choosing one solution at each step. The proof for the general pont-to-set-case is a direct extension requiring more cumbersome notations. We showed in Section 4 that the error E decreases at each step of the algorithm, i.e., $E(A(x)) \leq E(x)$ for any x in $X = S$, and in particular for any x in Γ, thus II(b) is satisfied.

To prove that II(a) holds as well, let us assume that $E(A(x)) = E(x)$, and show that x must lie in Γ. Let us first consider step 1 of the algorithm. This step minimizes, for $j = 1, .., n$, the error A_i with respect to $\mathcal{M}_i$ under the constraint that $|\mathcal{M}_i|^2 = 1$. Since all the other matrices $\mathcal{M}_k$ and all other points $\boldsymbol{P}_j$ are also held constant, this means that E is also minimized with respect to $\mathcal{M}_i$ under the same constraint. In other words, if $x = (\mathcal{M}, \mathcal{P})$ and $\mathcal{M}^o$ is the solution computed after step 1, we must have $E(\mathcal{M}^o, \mathcal{P}) \leq E(\mathcal{M}', \mathcal{P})$ for any $\mathcal{M}'$ satisfying the constraints. Since $E(A(x)) = E(x)$, $E(x)$ and $E(\mathcal{M}^o, \mathcal{P})$ must also be equal because, otherwise, E would strictly decrease after step 1. Therefore, $E(x) \leq E(\mathcal{M}, \mathcal{P}')$ for all $\mathcal{M}'$, that is, x is a local minimum with respect to $\mathcal{M}$ under the

constraint. In particular, we obtain, for $i = 1, .., m$, that $\nabla E(x) \cdot u = 0$ for the portion of $T(x)$ spanned by the $\mathcal{M}_i$ coordinates of x. Using the result of step 1, a similar line of reasoning applied to step 2 shows that, for $j = 1, .., n$, we have $\nabla E(x) \cdot u = 0$ for the portion of $T(x)$ spanned by the $\boldsymbol{P}_j$ coordinates of x. Combining the two results, we obtain that $\nabla E(x) \cdot u = 0$ for all x in $T(x)$, thus x is an element of Γ and II(a) is satisfied.

To show that III holds as well, we first decompose A into four elementary mappings:

- A_1 associates with $\mathcal{P}$ the matrices $\mathcal{C}_i$ $(i = 1, .., m)$;
- A_2 associates with $\mathcal{C}_i$, the matrix $\mathcal{M}_i$ constructed from the eigenvector $\boldsymbol{m}_i$ associated with the minimum eigenvalue of $\mathcal{C}_i^T \mathcal{C}_i$;
- A_3 associates with $\mathcal{M}$ the matrices $\mathcal{D}_j$ and $(j = 1, .., n)$;
- A_4 associates with $\mathcal{D}_j$, the eigenvector $\boldsymbol{P}_j$ associated with the minimum eigenvalue of $\mathcal{D}_j^T \mathcal{D}_j$.

A_1 and A_3 are continuous functions of their inputs and hence closed (Lemma 3(a)). The fact that A_2 and A_4 are closed mappings follows directly from Lemma 2. Finally, we can show that $A = (A_4 \circ A_3) \circ (A_2 \circ A_1)$ is closed by applying Lemma 3 as we did in the case of the factorization algorithm. Therefore, the bilinear algorithm satisfies all three conditions of the GCT and is globally convergent.

Acknowledgments We wish to thank Marc Pollefeys for kindly providing the CASTLE data, Andrew Fitzgibbon and Andrew Zisserman for kindly providing the WILSHIRE data, and Daniel Morris for providing his implementation of the bundle adjustment algorithm. We also wish to thank Eric de Sturler and Mike Heath for useful discussions. This work was supported in part by the Beckman Institute and in part by the National Science Foundation under grants IRI-990709 and IIS-0312438.

References

Birchfield, S.: 1998, 'KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker'.

Brand, M.: 2002, 'Incremental singular value decomposition of uncertain data with missing values'. In: *European Conference on Computer Vision (ECCV)*.

Brown, D.: 1976, 'The bundle adjustment – Progress and prospects'. *Int. Archives Photogrammetry* **21**(3).

Chen, Q. and G. Medioni: 1999, 'Efficient iterative solution to m-view projective reconstruction problem'. In: *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, Vol. II. Fort Collins, Colorado, pp. 55–61.

Christy, S. and R. Horaud: 1996, 'Euclidean shape and motion from multiple perspective views by affine iterations'. *IEEE Trans. Patt. Anal. Mach. Intell.* **18**(11), 1098–1104.

Edelsbrunner, H. and E. P. Mücke: 1994, 'Three-Dimensional Alpha Shapes'. *ACM Transactions on Graphics* **13**(1), 43–72.

Faugeras, O.: 1992, 'What can be seen in three dimensions with an uncalibrated stereo rig?'. In: G. Sandini (ed.): *Proc. European Conf. Comp. Vision*, Vol. 588 of *Lecture Notes in Computer Science*. Santa Margherita, Italy, pp. 563–578.

Faugeras, O.: 1995, 'Stratification of 3D vision: projective, affine and metric representations'. *J. Opt. Soc. Am. A* **12**(3), 465–484.

Faugeras, O., Q.-T. Luong, and T. Papadopoulo: 2001, *The Geometry of Multiple Images*. MIT Press.

Faugeras, O. and T. Papadopoulo: 1997, 'Gaussman-Caylay algebra for modeling systems of cameras and the algebraic equations of the manifold of trifocal tensors'. Technical Report 3225, INRIA Sophia-Antipolis.

Gill, P. and W. Murray: 1974, 'Newton-type methods for unconstrained and linearly constrained optimization'. *Math. Programming* **28**, 311–350.

Golub, G. and C. Van Loan: 1996, *Matrix computations*. John Hopkins University Press. Third edition.

Han, M. and T. Kanade: 2000, 'Creating 3D Models with Uncalibrated Cameras'. In: *Proceedings WACV*.

Hartley, R.: 1995, 'In defence of the 8-point algorithm'. In: *Proc. Int. Conf. Comp. Vision*. Boston, MA, pp. 1064–1070.

Hartley, R.: 1997, 'Lines and points in three views and the trifocal tensor'. *Int. J. of Comp. Vision* **22**(2), 125–140.

Hartley, R.: 1998, 'Computation of the quadrifocal tensor'. In: *Proc. European Conf. Comp. Vision*. pp. 20–35.

Hartley, R., R. Gupta, and T. Chang: 1992, 'Stereo from uncalibrated cameras'. In: *Proc. IEEE Conf. Comp. Vision Patt. Recog.* Champaign, IL, pp. 761–764.

Hartley, R. and F. Schaffalitzky: 2002, 'PowerFactorization: A method for 3D reconstruction with missing'. Australian National University.

Hartley, R. and A. Zisserman: 2000, *Multiple view geometry in computer vision*. Cambridge University Press.

Heath, M.: 1997, *Scientific Computing: An Introductory Survey*. McGraw-Hill.

Heyden, A.: 1997, 'Projective structure and motion from image sequences using subspace methods'. In: *Scandinavian Conference on Image Analysis*. pp. 963–968.

Heyden, A.: 1998, 'A common framework for multiple view tensors'. In: *Proc. European Conf. Comp. Vision*. pp. 3–19.

Heyden, A. and K. Åström: 1998, 'Minimal conditions on intrinsic parameters for Euclidean reconstruction'. In: *Asian Conference on Computer Vision*. Hong Kong.

Heyden, A., R. Berthilsson, and G. Sparr: 1999, 'An iterative factorization method for projective structure and motion from image sequences'. *Image and Vision Computing* **17**, 981–991.

Jacobs, D.: 1997, 'Linear fitting with missing data'. In: *Proc. IEEE Conf. Comp. Vision Patt. Recog.* San Juan, Puerto Rico, pp. 206–212.

Laveau, S. and O. Faugeras: 1994, '3D scene representation as a collection of images and fundamental matrices'. Technical Report 2205, INRIA Sophia-Antipolis.

Lu, C. and G. Hager: 2000, 'Fast and Globally Convergent Pose Estimation From Video Images'. *PAMI* **22**(2).

Luenberger, D.: 1984, *Linear and nonlinear programming*. Addison-Wesley. Second edition.

Luong, Q.-T., R. Deriche, O. Faugeras, and T. Papadopoulo: 1993, 'On determining the fundamental matrix: analysis of different methods and experimental results'. Technical Report 1894, INRIA Sophia-Antipolis.

Mahamud, S. and M. Hebert: 2000, 'Iterative projective reconstruction from multiple views'. In: *Proc. IEEE Conf. Comp. Vision Patt. Recog.* Hilton Head, SC, pp. II–430–437.

Mahamud, S., M. Hebert, Y. Omori, and J. Ponce: 2001, 'Provably-Convergent Iterative Methods for Projective Structure from Motion'. In: *Proc. IEEE Conf. Comp. Vision Patt. Recog.* pp. 1018–1025.

Mohr, R., L. Quan, F. Veillon, and B. Boufama: 1992, 'Relative 3D Reconstruction Using Multiple Uncalibrated Images'. Technical Report RT 84-IMAG 12-LIFIA, LIFIA-IRIMAG.

Morris, D. and T. Kanade: 1998, 'A unified factorization algorithm for points, line segments and planes with uncertainty models'. In: *Proc. Int. Conf. Comp. Vision.* Bombay, India, pp. 696–702.

Morris, D., K. Kanatani, and T. Kanade: 2000, 'Uncertainty Modeling for Optimal Structure from Motion'. In: B. Triggs, A. Zisserman, and R. Szeliski (eds.): *Vision Algorithms: Theory and Practice.* Springer-Verlag. Lecture Notes in Computer Science 1883.

Oliensis, J.: 1999, 'Fast and Accurate Self–Calibration'. In: *Proc. Int. Conf. Comp. Vision.* Corfu, Greece, pp. 745–752.

Pollefeys, M.: 1999, 'Self-calibration and metric 3D reconstruction from uncalibrated image sequences'. Ph.D. thesis, Katholieke Universiteit Leuven.

Ponce, J.: 2000, 'Metric Upgrade of a Projective Reconstruction Under the Rectangular Pixel Assumption'. In: *Second Workshop on Structure from Multiple Images of Large Scale Environments.* Dublin, Ireland, pp. 18–27. Preprints.

Rother, C. and S. Carlsson: 2002, 'Linear multi view reconstruction with missing data'. In: *European Conference on Computer Vision (ECCV).*

Shashua, A.: 1995, 'Algebraic functions for recognition'. *IEEE Trans. Patt. Anal. Mach. Intell.* **17**(8), 779–789.

Shum, H., K. Ikeuchi, and R. Reddy: 1995, 'Principal Component Analysis with Missing Data and its Application to Polyhedral Object Modeling'. *IEEE Trans. Patt. Anal. Mach. Intell.* **17**(9), 854–867.

Slama, C., C. Theurer, and S. Henriksen (eds.): 1980, *Manual of photogrammetry.* American Society of Photogrammetry. Fourth edition.

Sturm, P. and B. Triggs: 1996, 'A factorization-based algorithm for multi-image projective structure and motion'. In: *Proc. European Conf. Comp. Vision.* pp. 709–720.

Thompson, M., R. Eller, W. Radlinski, and J. Speert (eds.): 1966, *Manual of Photogrammetry.* American Society of Photogrammetry. Third Edition.

Tomasi, C. and T. Kanade: 1992, 'Shape and Motion from Image Streams under Orthography: a Factorization Method'. *Int. J. of Comp. Vision* **9**(2), 137–154.

Triggs, B.: 1996, 'Factorization methods for projective structure from motion'. In: *Proc. IEEE Conf. Comp. Vision Patt. Recog.* pp. 845–851.

Triggs, B., P. McLauchlan, R. Hartley, and A. Fitzgibbon: 2000, 'Bundle adjustment - A modern synthesis'. In: B. Triggs, A. Zisserman, and R. Szeliski (eds.): *Vision Algorithms: Theory and Practice.* Springer-Verlag, pp. 298–372. Lecture Notes in Computer Science 1883.

Zangwill, W.: 1969, *Nonlinear programming: A unified approach.* Prentice-Hall.

Zhang, Z., R. Deriche, O. Faugeras, and Q.-T. Luong: 1995, 'A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry'. *Artificial Intelligence Journal* **78**, 87–119.