Provably-Convergent Iterative Methods for Projective Structure from Motion

Shyjan Mahamud and Martial Hebert
Robotics Institute
Carnegie-Mellon University
Pittsburgh, PA 15213, USA

Yasuhiro Omori* and Jean Ponce
Beckman Institute
University of Illinois
Urbana, IL 61801, USA

Astract: The estimation of the projective structure of a scene from image correspondences can be formulated as the minimization of the mean-squared distance between predicted and observed image points with respect to the projection matrices, the scene point positions, and their depths. Since these unknowns are not independent, constraints must be chosen to ensure that the optimization process is well posed. This paper examines three plausible choices, and shows that the first one leads to the Sturm-Triggs projective factorization algorithm, while the other two lead to new provably-convergent approaches. Experiments with synthetic and real data are used to compare the proposed techniques to the Sturm-Triggs algorithm and bundle adjustment.

1 Introduction

Let us consider n fixed points $P_1, .., P_n$ observed by m perspective cameras. Given some fixed world coordinate system, we can write

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

for $i = 1, .., m$ and $j = 1, .., n$, where $\boldsymbol{p}_{ij} = (u_{ij}, v_{ij}, 1)^T$ 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 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 $\mathcal{M}_i$ $(i = 1, .., m)$ and the vectors $\boldsymbol{P}_j$ $(j = 1, .., n)$ from the image correspondences $\boldsymbol{p}_{ij}$. Of course, z_{ij} is also unknown, but its value is 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 $\mathcal{M}_i$. Faugeras [3] and Hartley *et al.* [7] have shown that when the internal parameters of the cameras are unknown, the camera motion and the scene structure can only be reconstructed up to an arbitrary projective transformation $\mathcal{Q}$. The parameters z_{ij} are independent of the choice of $\mathcal{Q}$, hence the name of *projective depths* often used for these parameters.

Several effective techniques for computing a projective scene representation from multiple images have been proposed (e.g., [3, 7, 18]), but, with a few exceptions (e.g., [1, 2, 5, 12, 14]), current approaches to projective motion analysis do not handle multiple images in a uniform manner. Instead, they use the now classical algebraic relations associated with small sets of pictures (e.g., [6, 18]) to stitch together the corresponding reconstructions into a common framework. Once initial estimates of the scene structure and camera motion 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. This article introduces two iterative approaches to projective structure and motion estimation that, unlike related approaches [1, 5, 12, 14], are guaranteed to converge to a local minimum of the error function. These algorithms are simple but the convergence proofs, based on the Global Convergence Theorem from [10], are a bit involved and they are relegated to an Appendix. Experiments are used to compare them to the projective factorization method of Sturm and Triggs [14] and to the bundle adjustment method of Morris, Kanatani and Kanade [13].

2 Background

Ideally, we would like to minimize the mean-squared geometric distance $E_0 = \sum_{i,j} |\boldsymbol{p}_{ij} - \frac{1}{z_{ij}} \mathcal{M}_i \boldsymbol{P}_j|^2$ between the observed image points and the point positions predicted from the parameters z_{ij}, $\mathcal{M}_i$ and $\boldsymbol{P}_j$. Unfortunately, the corresponding optimization problem is highly non-linear. Instead, we will minimize $E \stackrel{\text{def}}{=} \sum_{i,j} |z_{ij} \boldsymbol{p}_{ij} - \mathcal{M}_i \boldsymbol{P}_j|^2$. This error measure is not as geometrically satisfying as the previous one, but the rest of this article will show that its minimization under various classes of constraints can serve as a unifying

*This work was done while Y. Omori was visiting the Beckman Institute. He is now with the Dept. of Industrial Eng. and Management, Nihon University, Narashino, Chiba, Japan.

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 unknowns $\mathcal{M}_i$, $\boldsymbol{P}_j$ and z_{ij}. Indeed, E admits trivial zero minima corresponding to some of all of the z_{ij} being zero: examples include choosing zero values for all parameters z_{ij}, $\mathcal{M}_i$ and $\boldsymbol{P}_j$ (we will call this the *all-zero solution* in the sequel), or zero values for all z_{ij}, with $\mathcal{M}_i = \mathcal{M}_0$, 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 unit vector in its null space. These trivial minima arise because the transition from E_0 to E introduces non-physical solutions of (1) with at least some of the projective depths equal to zero. They will always occur, irrespective of the algorithm used for the minimization. We will revisit this important issue in Section 6.

Let us introduce the image data matrix $\mathcal{I}$ [14]

$$\mathcal{I} \stackrel{\text{def}}{=} \begin{pmatrix} z_{11}\boldsymbol{p}_{11} & \dots & z_{1n}\boldsymbol{p}_{1n} \\ \dots & \dots & \dots \\ z_{m1}\boldsymbol{p}_{m1} & \dots & z_{mn}\boldsymbol{p}_{mn} \end{pmatrix},$$

and observe that, given m images of n points, (1) can be rewritten as $\mathcal{I} = \mathcal{M}\mathcal{P}$, where $\mathcal{M}^T = (\mathcal{M}_1^T, .., \mathcal{M}_m^T)$ and $\mathcal{P} = (\boldsymbol{P}_1, .., \boldsymbol{P}_n)$. It follows immediately that minimizing E is equivalent to finding the parameters z_{ij}, $\mathcal{M}$ and $\mathcal{P}$ that minimize the Frobenius norm of the difference between $\mathcal{I}$ and $\mathcal{M}\mathcal{P}$ under the appropriate constraints.

2.1 The Sturm-Triggs Algorithm

The Sturm-Triggs factorization algorithm [14] generalizes to the projective case the affine factorization algorithm proposed by Tomasi and Kanade [15] (see [2, 5, 8] for related approaches), using epipolar constraints between pairs of successive images to compute initial values for the projective depths z_{ij} [14]. The all-zero solution mentioned in the previous section is avoided by scaling the rows of $\mathcal{I}$ so they have unit norms, then scaling the columns of this matrix so they have unit norm. At this point, the values of the projective depths are held constant, and the matrices $\mathcal{M}$ and $\mathcal{P}$ minimizing E are found using singular value decomposition (SVD): as show in [4], these matrices can be taken equal to $\mathcal{M} = \mathcal{U}_4\sqrt{\mathcal{W}_4}$ and $\mathcal{P} = \sqrt{\mathcal{W}_4}\mathcal{V}_4^T$, where $\mathcal{U}\mathcal{W}\mathcal{V}^T$ denotes the SVD 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 $\mathcal{V}$. The original algorithm stops here, but Triggs [16] proposed later to make this scheme iterative by refining the estimate of the projective depths at each iteration (Table 1).

There is no guarantee that the iterative process will converge (even to a local minimum) because of the renormalization step at each iteration. Unfortunately, this step is necessary to avoid the all-zero solution.

Compute an initial estimate of the projective depths z_{ij}, with $i = 1, .., m$ and $j = 1, .., n$. Repeat: (1) normalize each row of the data matrix $\mathcal{I}$, then normalize each one of its columns; (2) use singular value decomposition to compute the matrices $\mathcal{M}$ and $\mathcal{P}$ minimizing $\lvert\mathcal{I} - \mathcal{M}\mathcal{P}\rvert^2$; (3) for $i = 1, .., m$ and $j = 1, .., n$, find the value of z_{ij} minimizing $\lvert z_{ij}\boldsymbol{p}_{ij} - \mathcal{M}_i\boldsymbol{P}_j\rvert^2$ using linear least squares; until convergence.

Table 1: The Sturm-Triggs factorization algorithm.

2.2 Bundle Adjustment

Bundle-adjustment methods originate from photogrammetry [17]. In these methods, the projective depths do not appear as independent variables, and Cartesian image coordinates are used instead of projective ones to rewrite (1) as

$$u_{ij} = \frac{\boldsymbol{m}_{i1} \cdot \boldsymbol{P}_j}{\boldsymbol{m}_{i3} \cdot \boldsymbol{P}_j}, \quad v_{ij} = \frac{\boldsymbol{m}_{i2} \cdot \boldsymbol{P}_j}{\boldsymbol{m}_{i3} \cdot \boldsymbol{P}_j}.$$

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, the parameters of the matrices $\mathcal{M}_i$ and $\boldsymbol{P}_j$ minimizing the mean-squared distance between predicted and observed image points are found using non-linear least-squares techniques.

3 A Convergent 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 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-Trigss method is that the values of the projective depths are estimated by minimizing E under the constraint that the columns $\boldsymbol{d}_j$ $(j = 1, .., n)$ of the matrix $\mathcal{I}$ have unit norm. Concretely, let us assume that we are at some stage of the minimization process, fix the value of $\mathcal{M}$ to its current estimate and compute, for $j = 1, .., n$, the values of $\boldsymbol{z}_j = (z_{1j}, .., z_{mj})^T$ and $\boldsymbol{P}_j$ that minimize $E_j = \sum_i \lvert z_{ij}\boldsymbol{p}_j - \mathcal{M}_i\boldsymbol{P}_j\rvert^2$. 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 yields $\boldsymbol{P}_j = \mathcal{M}^\dagger\boldsymbol{d}_j$, where $\mathcal{M}^\dagger = (\mathcal{M}^T\mathcal{M})^{-1}\mathcal{M}^T$ is the pseudoinverse of $\mathcal{M}$. In turn, substituting this value in the definition of E_j yields $E_j = \lvert(\text{Id} - \mathcal{M}\mathcal{M}^\dagger)\boldsymbol{d}_j\rvert^2$. Now, $\mathcal{M}$ is a $3m \times 4$ matrix of rank 4, whose SVD $\mathcal{U}\mathcal{W}\mathcal{V}^T$ 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 $|\boldsymbol{d}_j|^2 = 1$. Observing that

$$\boldsymbol{d}_j = \mathcal{Q}_j \boldsymbol{z}_j, \quad \text{where} \quad \mathcal{Q}_j = \begin{pmatrix} \boldsymbol{p}_{1j} & .. & \boldsymbol{0} \\ .. & .. & .. \\ \boldsymbol{0} & .. & \boldsymbol{p}_{mj} \end{pmatrix},$$

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 $\mathcal{R}_j = \mathcal{U}^T \mathcal{Q}_j$. This is a generalized eigenvalue problem, whose solution is the eigenvector associated with the largest eigenvalue. 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 initial projective depth values are set to 1.

Compute an initial estimate of the projective depths z_{ij}, with $i = 1, .., m$ and $j = 1, .., n$ and normalize each column of the data matrix $\mathcal{I}$.
Repeat:
(1) use singular value decomposition to compute the matrices $\mathcal{M}$ and $\mathcal{P}$ minimizing $\lvert\mathcal{I} - \mathcal{M}\mathcal{P}\rvert^2$;
(2) for $j = 1$ to n, compute the matrices $\mathcal{R}_j$ and $\mathcal{Q}_j$ and find the value of $\boldsymbol{z}_j$ maximizing $\lvert\mathcal{R}_j \boldsymbol{z}_j\rvert^2$ under the constraint $\lvert\mathcal{Q}_j \boldsymbol{z}_j\rvert^2 = 1$;
(3) update the value of $\mathcal{I}$ accordingly;
until convergence.

Table 2: The iterative factorization algorithm.

3.2 Convergence

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 $E^{(1)}$ is the value of the error at the end of step 1, we have therefore $E^{(1)} \leq E^{(0)}$. Now step 2 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 $E^{(2)}$ at the end of this step is smaller than or equal to $E^{(1)}$. This shows that the error decreases monotonically, and since it is bounded below by zero, it also converges 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 full convergence proof is outlined in the appendix (see [11] for details).

4 A Convergent 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 $\boldsymbol{P}_j$. As shown in the appendix, the corresponding minimization algorithm is guaranteed to converge to a local minimum of E.

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} is zero at an extremum of this function yields $0 = \partial E / \partial z_{ij} = 2\boldsymbol{p}_{ij}^T (z_{ij}\boldsymbol{p}_{ij} - \mathcal{M}_i \boldsymbol{P}_j)$, or $z_{ij} = (\boldsymbol{p}_{ij}^T \mathcal{M}_i \boldsymbol{P}_j)/|\boldsymbol{p}_{ij}|^2$. Substituting in the definition of E and assuming that the homogeneous image coordinate vectors $\boldsymbol{p}_{ij}$ have been scaled to unit norm during preprocessing, shows, after some algebraic manipulation that, at an extremum of this function, we must have $E = \sum_{ij} |\boldsymbol{p}_{ij} \times (\mathcal{M}_i \boldsymbol{P}_j)|^2$. Our task is thus reduced to minimizing this expression under appropriate constraints on the matrices $\mathcal{M}_i$ and the vectors $\boldsymbol{P}_j$. Note that this minimization process *never* involves the explicit estimation of the projective depths. The next question is how to choose the right constraints for the minimization. We have chosen to use the simplest constraints, that is, to constraint both the projection matrices and the points to be of unit norm: $|\mathcal{M}_i|^2 = 1$ and $|\boldsymbol{P}_j|^2 = 1$ for $i = 1, .., m$ and $j = 1, .., n$.

4.1 Algorithm

We propose to start with 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 [17]. Variants of this approach include the bilinear methods of Morris and Kanade [12] and Chen and Medioni [1], and the photogrammetric method of *block successive over relaxation* (see [17] for a discussion).

Let us rewrite our error function as

$$E = \sum_{j=1}^{n} E_j^{(\mathcal{M})} = \sum_{i=1}^{m} E_i^{(\mathcal{P})} \quad \text{with} \quad \begin{cases} E_j^{(\mathcal{M})} = \sum_{i=1}^{m} |\boldsymbol{p}_{ij} \times (\mathcal{M}_i \boldsymbol{P}_j)|^2, \\ E_i^{(\mathcal{P})} = \sum_{j=1}^{n} |\boldsymbol{p}_{ij} \times (\mathcal{M}_i \boldsymbol{P}_j)|^2. \end{cases}$$

The algorithm presented in this section alternates (1) steps where the point positions $\boldsymbol{P}_j$ are held constant

while, for $i = 1, .., n$, the error $E_i^{(\mathcal{P})}$ 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 $E_j^{(\mathcal{M})}$ is minimized under the constraint $|P_j|^2 = 1$. 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 $E_i^{(\mathcal{P})} = |C_i \boldsymbol{m}_i|^2$, 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 $3n \times 12$ matrix

$$\begin{pmatrix} -w_{i1}\boldsymbol{P}_1^T & \boldsymbol{0}^T & u_{i1}\boldsymbol{P}_1^T \\ \boldsymbol{0}^T & -w_{i1}\boldsymbol{P}_1^T & v_{i1}\boldsymbol{P}_1^T \\ -v_{i1}\boldsymbol{P}_1^T & u_{i1}\boldsymbol{P}_1^T & \boldsymbol{0}^T \\ \cdots & \cdots & \cdots \\ -w_{in}\boldsymbol{P}_n^T & \boldsymbol{0}^T & u_{in}\boldsymbol{P}_n^T \\ \boldsymbol{0}^T & -w_{in}\boldsymbol{P}_n^T & v_{in}\boldsymbol{P}_n^T \\ -v_{in}\boldsymbol{P}_n^T & u_{in}\boldsymbol{P}_n^T & \boldsymbol{0}^T \end{pmatrix}.$$

In particular, finding the matrix $\mathcal{M}_i$ with unit Frobenius norm that minimizes $E_i^{(\mathcal{P})}$ 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 $E_j^{(\mathcal{M})} = |\mathcal{D}_j \boldsymbol{P}_j|^2$, where $\mathcal{D}_j$ is the $3m \times 4$ matrix

$$\begin{pmatrix} [\boldsymbol{p}_{1j}\times]\mathcal{M}_1 \\ \cdots \\ [\boldsymbol{p}_{mj}\times]\mathcal{M}_m \end{pmatrix}$$

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 $E_j^{(\mathcal{M})}$ 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. The initial $\boldsymbol{P}_j$ values are obtained using the Tomasi-Kanade [15] affine factorization method, which is equivalent to setting the initial projective depths to 1.

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.

4.2 Convergence

As before, let $E^{(0)}$ be the current error value at the beginning of each iteration. During the first step of the iteration, we minimize, for $i = 1, \ldots, n$ the error $E_i^{(\mathcal{P})}$ with respect to the matrix $\mathcal{M}_i$ and thus also minimize the total error $E = \sum_{i=1}^m E_i^{(\mathcal{P})}$. It follows that the new value $E^{(1)}$ of the error must be smaller than $E^{(0)}$. During the second step, the error $E_j^{(\mathcal{M})}$ is minimized with respect to $\boldsymbol{P}_j$ for $j = 1, \ldots, n$. It follows that the total error $E = \sum_{j=1}^n E_j^{(\mathcal{M})}$ 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, monotonic convergence of the error E to E^* does not immediately imply 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.

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.

Table 3: An iterative bilinear algorithm for projective shape from motion.

5 Implementation and Results

This section compares the two proposed methods with our implementation of the Sturm-Triggs iterative factorization technique and the bundle-adjustment implementation described in [13]. 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 [15] 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 algorithms, by the root mean-squared distance between the predicted and observed image points. Figure 1 plots, for each of the data sets used in our experiments, the mean and maximum reprojection errors (in pixels) for the various methods as a function of the number of iterations.

5.1 Synthetic Images

Our first experiment compares 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 are taken by a camera looking directly at the sphere center,

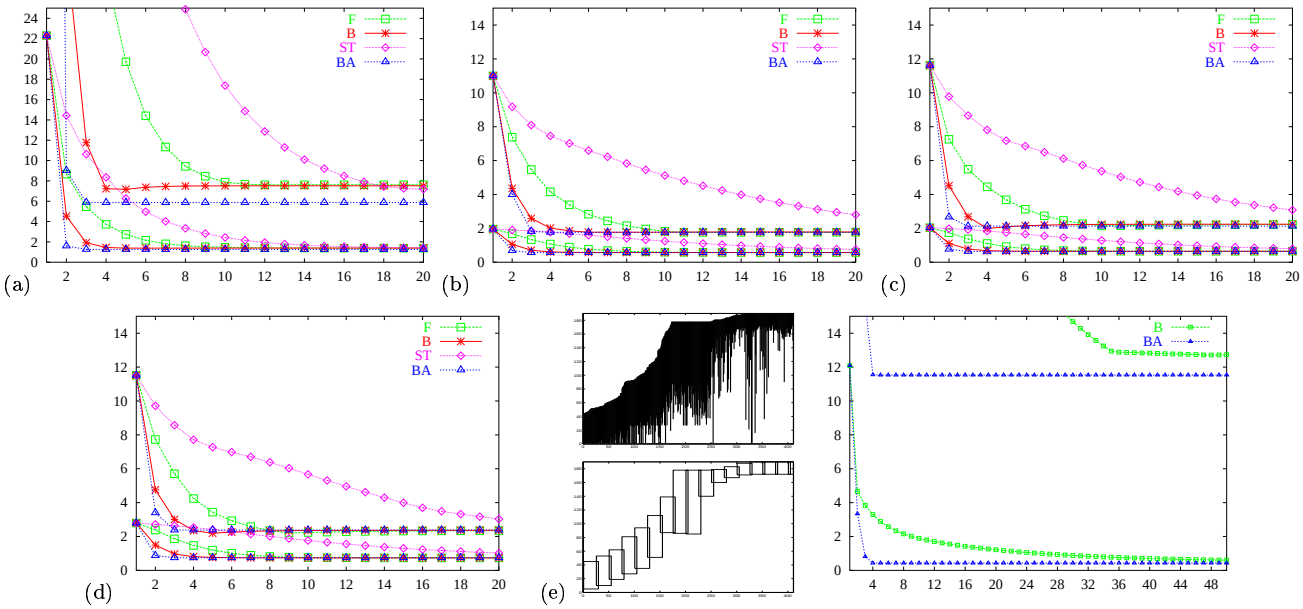

Figure 1: Experimental results: (a) synthetic data; (b) full Castle data; (c) alternate Castle data; (d) inner Castle data; (e) Wilshire data. The following symbols are used in all plots: F and B stand for the proposed **F**actorization and **B**ilinear algorithms, and ST and BA respectively stand for the **S**turm-**T**riggs and **B**undle **A**djustment techniques.

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. An additional 10 views are taken from random point on a sphere of radius 150 units to serve as test images. These test views are not confined to the small patch from which the training views have been taken. The image size is 512×512 pixels, with a focal length of 256. Gaussian noise of one pixel has been added to all input data. Figure 1(a) shows the results of our experiments with these synthetic data, averaged over 20 trials. As shown by this figure, all methods converge in less than 20 iterations and yield comparable errors. The main difference between the four algorithms is their convergence rate: bundle adjustment is the technique that requires the fewest iterations to converge, closely followed by the bilinear method of Section 4, then by the iterative factorization of Section 3 and finally the Sturm-Triggs iterative factorization method. As shown in the rest of this section, the same pattern emerges from the rest of our experiments.

5.2 Real Images

Figure 1(b) 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 at convergence, the final average error being below one pixel in all cases. 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 1(c)), while the middle 10 frames are used for training and the outside 10 images for testing in the second experiment (Figure 1(d)). 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 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. Factorization algorithms, on the other hand, normally require all points to be visible in all images (see [15] for modifications of factorization to handle missing data.) We have compared the bilinear and bundle-adjustment algorithms on the Wilshire data kindly provided by Andrew Fitzgibbon and Andrew Zisserman (Figure 1(e)). This data set consists of 411 points and 190 frames. As shown by Figure 1(e,top-left), not all points are visible in all images. We have split the data into consecutive blocks of 30 frames with a 5-frame overlap, and used the Tomasi-Kanade method in the maximal full rectangle of each

block (Figure 1(e,bottom-left)) to compute an affine reconstruction of the corresponding points. The successive reconstructions have then been registered. As shown by Figure 1(e,right), the initial errors are much larger in this case, and it takes the algorithm about 20 iterations to reach sub-pixel mean error, as opposed to only about 4 iterations for bundle adjustment. Although it is not meaningful to compare directly the running times of the MATLAB implementation of the algorithms (the bundle adjustement implementation uses compiled C code as well), it is worth noting that the bilinear algorithm takes 4 minutes on the Wilshire data, while bundle adjustment takes three hours. This large difference suggests that the low cost of the bilinear iterations greatly outweighs the fast convergence of the bundle adjustment.

5.4 Cost Comparison

The singular value decomposition of a $k \times l$ matrix can be computed in time $O(kl \min(k,l))$ [4]. 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, is $O(mn)$, since the computation of the matrices $\mathcal{A}_j$ and $\mathcal{C}_j$ takes $O(m)$ time, the computation of the matrices $\mathcal{B}_i$ and $\mathcal{D}_i$ takes $O(n)$ time, and the computation of the corresponding generalized eigenvectors and eigenvalues takes constant time. In contrast, each iteration of a Gauss-Newton or Levenberg-Marquart solution to bundle adjustment costs $O((m+n)^3)$. The performance of the four algorithms can be improved: fixed-rank approximations of the singular value decomposition $\mathcal{U}\mathcal{W}\mathcal{V}^T$ can be used to compute only the portions of $\mathcal{U}$ and $\mathcal{V}$ associated with a fixed number of singular values [16]. 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 appears are both bounded by k, with a complexity reduced to $O((m+n)^2 k)$. The bilinear algorithm also benefits from sparsity, with a cost of $O((m+n)k)$ under the same assumptions.

6 Discussion

The bilinear algorithm proposed in this paper can be thought of as an *alternation* technique [17], 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 [17] 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 outweighs the loss in convergence rate. The iterative factorization algorithms of Sturm and Triggs [14], Heyden *et al.* [8], and the factorization method of Section 3, are also alternation techniques since they interleave steps where $\mathcal{M}$ and $\mathcal{P}$ are estimated with steps where the projective depths are estimated.

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 [10, 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 perpsective 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 (at this point at least) be guaranteed.

Even if, for a given algorithm, the solutions computed at each iteration are all non-trivial, it is not possible to prevent the *limit* from being a trivial solution. More precisely, it is possible to have $z_{ij} \neq 0$ at every iteration but still have $z_{ij} = 0$ some i, j in the limit. Fundamentally, this is a consequence of the non-compactness of the space formed by excluding the trivial solutions. This

is also the reason why, in the Sturm-Triggs approach, the constraints cannot avoid all the trivial solutions and achieve provable convergence.

Experimentally, the algorithms proposed in this paper have never converged to a trivial solution in our experiments. A likely 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 $E' = \sum_{i,j} |\boldsymbol{p}_{ij} - \mathcal{M}_i \boldsymbol{P}_j|^2$. More investigation would be needed to verify that it is indeed the reason for the avoidance of trivial solutions observed in practice.

Appendix

We now show that, for the bilinear and factorization algorithms, the projection matrices and the points converge, that the error converges to a local minimum, and that convergence to a solution is guaranteed irrespective of the starting points. The proof relies on a general result from optimization theory, the *global convergence theorem*, or *GCT* [10, Chapter 6]. See, for example, [9] for its application to the study of the convergence of pose-estimation algorithms in computer vision.

The GCT represents algorithms as point-to-set mappings, that map each input value to a set of compatible outputs. Although it is often necessary to view an algorithm as a point-to-set mapping for convergence analysis, in actual implementation of the algorithm, a single output is chosen out of the set of outputs generated by the algorithm. This is relevant to the two algorithms proposed in this paper because they both rely on solving eigenvalue problems, and the matrices involved may have eigenvalues of multiplicity greater than one, forcing arbitrary choices among the eigenvectors.

In this setting, a convergent algorithm is a *closed* mapping $A : X \to Y$ from a topological space X of input parameters to a topological space Y of output values.

Definition 1 *A mapping $A : X \to Y$ 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 such that $y_k \in A(x_k)$ to some point y in Y imply that $y \in A(x)$. 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 $A : X \to 2^X$, and a sequence of points x_k in X such that for $k \geq 0$, $x_{k+1} \in A(x_k)$. When*

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

II. there exists a continuous 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. A is closed at every point outside Γ;

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

Strictly speaking, the GCT guarantees only convergence to the solution set of *subsequences*. This allows for possible alternation of x_k between multiple solutions. In practice, an arbitrary choice among such solutions is used as final output of the algorithm since they all have the same error. The existence of subsequences is a consequence of the compactness assumption since a convergent subsequence can always be extracted from a sequence in a compact space.

Because of space limitations we only give the outline of the proof for the bilinear algorithm. The convergence proof for the factorization algorithm is similar, except that it involves a lemma showing that the mapping that associates to a matrix its rank-4 SVD-based factorization is closed [11].

Let us denote by $\mathcal{M}$ the collection of all the matrices $\mathcal{M}_i$ and by $\mathcal{P}$ the collection of all the vectors $\boldsymbol{P}_j$. We 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 $(\mathcal{M}^0, \mathcal{P}^0)$. We will need the following lemma, which states that the mapping that associates the generalized eigenvectors to a pair of matrices is closed. Note that it is necessary to use point-to-set mappings because, in general, the multiplicity of the eigenvalue may be greater than one.

Lemma 1 *The mapping that associates with the matrices $\mathcal{U}$ and $\mathcal{V}$ and the scalar $\gamma > 0$ the vectors $\boldsymbol{x}^*$ minimizing $|\mathcal{U}\boldsymbol{x}|^2$ under the constraint $|\mathcal{V}\boldsymbol{x}|^2 = \gamma^2$ is closed.*

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 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 3(a) and 3(b) of the iteration. 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 since it can be shown that they correspond to critical minimization paths that will never be followed in practice.

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 point-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 3(a) of the algorithm. This step minimizes, for $i = 1, .., m$, the error $E_i^{(\mathcal{P})}$ with respect to $\mathcal{M}_i$ under the constraint $|\mathcal{M}_i|^2 = 1$. Since all other matrices $\mathcal{M}_k$ and all the points $\boldsymbol{P}_j$ are 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 3(a), 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 decrease after step 3(a). 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 3(a), a similar line of reasoning applied to step 3(b) 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 [10]. The fact that A_2 and A_4 are closed mappings follows directly from Lemma 1.

Finally, we show that A is closed by using two lemmas from [10]. The first lemma states that the composition $B \circ A$ of a point-to-point mapping A continuous at x and a point-to-set mapping B closed on $A(x)$ is closed at x. This lemma implies that both $A_2 \circ A_1$ and $A_4 \circ A_3$ are closed. The second lemma states that the combination of two closed mappings is itself closed if the range of the first mapping is compact. This lemma implies that $A = (A_4 \circ A_3) \circ (A_2 \circ A_1)$ is closed because the range of $A_2 \circ A_1$ is a compact set, the set of matrices $\mathcal{M}_i, i = 1, .., m$; $|\mathcal{M}_i|^2 = 1$.

The bilinear algorithm, formalized as above to a point-to-set mapping, satisfies the conditions of the GCT and is therefore 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 by the National Science Foundation under grant IRI-990709.

References

[1] Q. Chen and G. Medioni. Efficient iterative solution to m-view projective reconstruction problem. In *CVPR*, vol. II, pp. 55–61, 1999.

[2] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *PAMI*, 18(11):1098–1104, 1996.

[3] O.D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *ECCV*, 1992.

[4] G.H. Golub and C.F. Van Loan. Matrix Computations. John Hopkins Univ. Press, 1996.

[5] M. Han and T. Kanade. Creating 3d models with uncalibrated cameras. In *WACV*, 2000.

[6] R. Hartley. Lines and points in three views and the trifocal tensor. *IJCV*, 22(2):125–140, 1997.

[7] R.I. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *CVPR*, pp. 761–764, 1992.

[8] A. Heyden, R. Berthilsson, and G. Sparr. An iterative factorization method for projective structure and motion from image sequences. *Im. Vis. Comput.*, 17, 1999.

[9] C.P. Lu and G. Hager. Fast and globally convergent pose estimation from video images. *PAMI*, 22(2), 2000.

[10] D.G. Luenberger. *Linear and nonlinear programming.* Addison-Wesley, 1984.

[11] S. Mahamud, M. Hebert, Y. Omori and J. Ponce. Provably-convergent iterative methods for projective structure from motion. CMU Tech. Rep., 2001.

[12] D.D. Morris and T. Kanade. A unified factorization algorithm for points, line segments and planes with uncertainty models. In *ICCV*, pp. 696–702, 1998.

[13] D.D. Morris, K. Kanatani, and T. Kanade. Uncertainty modeling for optimal structure from motion. In B. Triggs, A. Zisserman, and R. Szeliski, *Vision Algorithms: Theory and Practice.* Springer-Verlag, 2000.

[14] P. Sturm and B. Triggs. A factorization-based algorithm for multi-image projective structure and motion. In *ECCV*, pp. 709–720, 1996.

[15] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 9(2):137–154, 1992.

[16] B. Triggs. Factorization methods for projective structure from motion. In *CVPR*, pp. 845–851, 1996.

[17] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A.W. Fitzgibbon. Bundle adjustment - a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski (eds.), *Vision Algorithms: Theory and Practice*, pp. 298–472. Springer-Verlag, 2000.

[18] Z. Zhang, R. Deriche, O.D. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artif. Intell.*, 78:87–119, 1995.