Linear Projective Reconstruction from Matching Tensors

Bill Triggs
INRIA Rhône-Alpes,
655 avenue de l'Europe, 38330 Montbonnot St. Martin, France.
Bill.Triggs@inrialpes.fr ◇ *http://www.inrialpes.fr/movi/Triggs*

Abstract

This paper describes initial work on a family of projective reconstruction techniques that compute projection matrices directly and linearly from matching tensors estimated from the image data. The approach is based on 'joint image closure relations' — bilinear constraints between matching tensors and projection matrices, that express the fact that the former derive from the latter. The simplest methods use fundamental matrices and epipoles, alternative ones use trilinear tensors. It is possible to treat all of the image data uniformly, without reliance on 'privileged' images or tokens. The underlying theory is discussed, and the performance of the new methods is quantified and compared with that of several existing ones.

Keywords: Multi-image structure, projective reconstruction, matching tensors.

1 Introduction

Traditional stereo vision systems use carefully calibrated cameras to provide metric reconstruction from a single pair of static images. It has long been clear that the redundancy offered by further images can significantly increase the quality and stability of visual reconstructions, as well as extending their coverage to previously hidden parts of the scene. Furthermore, much of the 3D structure can be recovered without *any* prior camera calibration. Even in the extreme case of several distinct unknown projective cameras viewing the scene from unknown positions, the entire metric scene geometry can be recovered up to just 9 global parameters — 3 scale factors, 3 skews and 3 projective distortions[1] [4, 7, 13]. Various common scene or camera constraints can be used to further reduce this ambiguity, *e.g.* known vanishing points or length ratios, known skew or aspect ratio, motion-constancy of intrinsic parameters, ... [6]. This is especially relevant to applications such as scene modelling for virtual reality or robot navigation, where many images are needed to cover the scene and precise calibration is difficult owing to uncertain camera motions, changes in internal parameters (focus, zooming) or the use of several cameras.

There is a need for visual reconstruction methods with the following characteristics:

1) **Multi-image/multi-point/missing data:** It is hard to match features reliably across many images, especially under large changes of viewpoint. Reconstruction methods requiring long sequences of matches tend to run into missing data problems. For example, factorization methods [26, 25, 30, 24] are very stable and treat all images and points equally, but require completely filled 'blocks' of points *vs.* images. Traditional methods further limit these blocks to small fixed

This work was supported by INRIA Rhône-Alpes, the Esprit HCM network and Esprit LTR grant CUMULI. Submitted to Image & Vision Computing. An earlier version appeared in BMVC'96.

[1] If there is lens distortion, this can also (in theory) be recovered up to an unknown image homography.

numbers of images or points. The stability of such methods is critically dependent on the images chosen, and since these must usually be closely-spaced to allow reliable matching, overall accuracy suffers. It is possible to work around gaps in the data by 'patching together' several partial reconstructions, but it would be useful to have methods that handled missing data naturally, without relying on *ad hoc* patching, key points, or key images.

2) **Flexible calibration:** Calibration constraints come in many forms: prior knowledge, calibration images, scene or motion constraints, ... It is not always obvious how to incorporate them into the multi-image reconstruction process. Often it is simpler to ignore them at first, working projectively and only later going back and using them to 'straighten' the recovered projective structure. This 'stratification' school [6] has its critics [32, 20]. In particular, it is felt that stability may be compromised by failing to enforce reasonable camera and motion models at the outset. However as far as I know it is the only approach that has yet produced true multi-image reconstruction algorithms for general cameras and motions [25, 30, 29, 24].

3) **Precision/robustness/stability:** *Precision* means that the method gives accurate results when it works; *robustness* that it works reliably (*e.g.* in the face of mismatches or initialization errors); *stability* that the results are not overly sensitive to perturbations in the input data. Stability is a precondition for precision and robustness, but is easily compromised by degeneracies in either the viewing geometry or the algorithmic formulation used.

For the best precision there is no substitute for rigorous statistical parameter estimation, *e.g.* maximum likelihood. For this, a nonlinear cost reflecting a statistical error model of the image observations must be globally optimized over all unknown 3D structure and calibration parameters. With Gaussian errors, this reduces to covariance-weighted nonlinear least squares. Such statistical 'bundle adjustment' is a truism for photogrammetrists but seems to be tacitly discouraged in computer vision, where the traditional emphasis is on A.I. image understanding rather than precision (however *cf.* [17, 10, 19, 14, 9]). Efficient numerical methods exist for handling large problems, both off-line and in a linearized recursive framework [1, 18].

Rigorous, statistically weighted least squares should not be confused with 'unweighted' or 'linear least squares' minimization of *ad hoc* 'algebraic distances' — sums of squared algebraic constraint violations with no direct relation to measured image residuals. For example the 'linear' method for the fundamental matrix [12], reconstruction by affine and projective factorization [26, 25, 30, 24], and the new 'closure based' methods presented here, all linearize the problem and minimize algebraic distances using linear algebra techniques (*e.g.* SVD). Common characteristics of such methods are: (*i*) they are linear and much simpler to implement than the corresponding statistical methods; (*ii*) no prior initialization is needed; (*iii*) somewhat more than the minimal amount of data is required, to allow nonlinearities to be "linearized away"; (*iv*) they are sensitive to the relative weighting of different components of the error function (but the choice is not too critical once you realize it has to be made); (*v*) with suitable weighting, they give results not too far from (but still worse than) the statistical optimum. Criticisms include: (*i*) ignoring constraints may reduce stability and make the results difficult to interpret; (*ii*) general linear methods are often slower than dedicated nonlinear ones, as large matrices tend to be involved; (*iii*) it is difficult to detect outliers without a clear error model.

Bundle adjustment routines provide all of the desirable features listed above, except robustness against initialization. As they are only iterative improvement techniques, they require initial estimates for all unknown parameters. In practice they are seldom robust against gross errors in these, or even against re-parametrization (*e.g.* convergence tests are notoriously sensitive to this).

Hence, there is still a need for stable and relatively tractable suboptimal reconstruction methods that require no prior initialization, take into account as many as possible of the above properties, and can be used as input to nonlinear methods if more precision is required. Partly in response to this, there has recently been a significant amount of work on the theoretical foundations of multi-image projection and reconstruction [11, 10, 19, 18, 23, 2, 22, 8, 15, 16, 31, 27, 28, 3]. The problem turns out to have a surprisingly rich mathematical structure and several complementary

approaches exist. The field is developing rapidly and there is no space for a survey here, so I will only mention a few isolated results. The epipolar constraint (the geometry of stereo pairs) is now well understood (*e.g.* [5]). Shashua [22] and Hartley [11] developed the theory of the trivalent tensor (three view constraint). Faugeras and Mourrain [8] and I [28] systematically studied the complete family of multi-image constraints (only one was unknown: a quadrilinear one).

As a means to this, I developed a tensorial approach to multi-image vision [28], which nicely unifies the geometric and algebraic aspects of the subject. This lead to the **joint image** picture, in which the combined homogeneous coordinates of all the images of a 3D point are stacked into a single big 'joint image' vector. The geometry of this space can be related to that of the original 3D points via the stacked projection matrices. All of the familiar image entities — points, lines, homographies, matching tensors, *etc* — fall naturally out of this picture as the joint image representatives of the corresponding 3D objects. The approach is also 'dual' (in the sense of Carlsson [3]) to Sparr's 'affine shape' formalism [23, 15, 24], where coordinates are stacked by point rather than by image.

In the MOVI group, we have recently developed several families of projective reconstruction methods based on the joint image approach. The factorization-based 'projective depth recovery' methods [25, 30] use matching tensors to recover a coherent set of projective scale factors for the image points. This gives an implicit reconstruction, which can be concretized by factorizing the matrix of rescaled image points into projection and structure matrices by a process analogous to the Tomasi-Kanade-Poelman method for affine structure [26, 21]. Factorization-based methods give an implicit linear least squares fit to all of the image data. They are simple and extremely stable, but have the serious practical disadvantage that each point must be visible in every image (modulo 'hallucination' [26]). This is unrealistic when there are many images covering a wide range of viewing positions.

The current paper represents a first attempt to overcome this problem. It describes a new family of reconstruction methods that extract projection matrices directly and linearly from estimated matching tensors, after which the scene structure can be recovered linearly by back-projecting the image measurements. The projections are estimated using 'joint image closure relations' — bilinear constraints between projections and their matching tensors, analogous to the depth recovery relations used for projective factorization, but with projection matrices replacing image points.

In principle, the closure based reconstruction methods treat all of the images uniformly, so they have the potential to be significantly more stable than the commonly used approach of initially reconstructing from two key images, then reprojecting into the other ones to estimate the remaining projection matrices. On the other hand, because they only use the image data indirectly via the matching tensors, they are not as stable as factorization based methods. The suggestion is that they will prove good replacements for the 'stereo + reprojection' methods (whose main application is probably to initialize more refined nonlinear least squares iterations), but that when tokens are visible in every image factorization will still be the best linear method.

The rest of the paper outlines the theory of the closure relations, describes the resulting reconstruction algorithms and their implementation, reports on an initial experimental study of their performance, and ends with a short discussion.

2 Theory

This section sketches the theoretical background of multi-image reconstruction, and discusses the 'joint image closure relations' on which the new reconstruction methods are based. The theory is not difficult, but when more than two images are involved the equations are hard to express without using tensorial notation. We will use ordinary matrix-vector notation except for a few trivalent tensor equations, so you should be able to follow most of the paper without a knowledge of tensors. An *extremely* brief introduction to them follows — see [28, 27] for more details. All quantities are assumed to be projective, expressed in homogeneous coordinates.

Tensors are just multidimensional arrays of components. Vectors (1-index arrays) and matrices (2-index arrays) are examples. Each index is associated with a specific space (the 3D world, image i, ...), and inherits the corresponding change-of-basis law. Many common vector and matrix operations generalize directly to tensors, provided we specify which of the many indices the operation applies to. (For matrices, the index is implicit in the 'juxtaposition = multiplication' rule). To keep track of the indices, we write them out explicitly: $a, b, c \ldots$ for world-space indices and $A_i, B_i, C_i \ldots$ for image i ones. The most common operation is **contraction** — summing a corresponding pair of indices over the range of their values, as in vector dot-product, matrix product or trace. The summation signs are elided: any index that appears twice in a term is implicitly summed over.

A further complication is that in projective geometry each space has a corresponding **dual**, *e.g.* in each image, the space of points is dual to the space of lines (hyperplanes). This means that every index actually comes in two varieties: point-like or **contravariant** and hyperplane-like or **covariant**. These have *different* (complementary) transformation laws under changes of basis, so they must be carefully distinguished: point indices are written as superscripts, hyperplane ones as subscripts. Contractions are only meaningful between covariant-contravariant pairs of indices from the same space, *e.g.* there is *no* meaningful 'dot product' between pairs of projective points — the result would be completely dependent on the basis chosen.

World points $\mathbf{X}^a$ project to image ones $\mathbf{x}^{A_i}$ by contraction with 3×4 projection matrices $\mathbf{P}_a^{A_i}$: $\mathbf{x}^{A_i} \sim \mathbf{P}_a^{A_i} \mathbf{X}^a$ (implicit summation over a). $\mathbf{e}_1^{A_2}$ denotes the epipole of camera 1 in image 2; $\mathbf{F}_{A_1 B_2}$ the fundamental matrix between images 1 and 2; and $\mathbf{G}_{A_1}{}^{B_2 C_3}$ the trivalent tensor between images 2 and 3 based in image 1. (There are also corresponding trivalent tensors based in images 2 and 3). In ordinary matrix-vector notation, $\mathbf{X}$ stands for $\mathbf{X}^a$, $\mathbf{x}_i$ for $\mathbf{x}^{A_i}$, $\mathbf{P}_i$ for $\mathbf{P}_a^{A_i}$, $\mathbf{e}_{ij}$ for $\mathbf{e}_i^{A_j}$, and $\mathbf{F}_{ij}$ for $\mathbf{F}_{A_i B_j}$.

Consider the projections $\lambda_{ip}\mathbf{x}_{ip} = \mathbf{P}_i \mathbf{X}_p$ of n homogeneous world points $\mathbf{X}_p$, $p = 1, \ldots, n$, into m images via 3×4 perspective projection matrices $\mathbf{P}_i$, $i = 1, \ldots, m$. The resulting mn homogeneous image points $\mathbf{x}_{ip}$ are only defined up to unknown scale factors λ_{ip}, called **projective depths**. As each $\mathbf{P}_i$ and $\mathbf{X}_p$ can be arbitrarily rescaled, there is some superficial freedom in the choice of these scales. However there is a strong underlying coherence that embodies the projective structure of the scene: the depths λ_{ip} really do capture the projective part of visual depth. An algebraic result of the coherence is the low rank (four) of the rescaled data matrix:

$$\begin{pmatrix} \lambda_{11}\mathbf{x}_{11} & \cdots & \lambda_{1n}\mathbf{x}_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1}\mathbf{x}_{m1} & \cdots & \lambda_{mn}\mathbf{x}_{mn} \end{pmatrix} = \begin{pmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_m \end{pmatrix} \begin{pmatrix} \mathbf{X}_1 & \cdots & \mathbf{X}_n \end{pmatrix}$$

It is useful to view this column-by-column, as the projection of world points $\mathbf{X}_p$ to $3m$-component **joint image space** vectors via the stacked $3m \times 4$ **joint projection** matrix $\mathbf{P}$:

$$\begin{pmatrix} \lambda_{1p}\mathbf{x}_{1p} \\ \vdots \\ \lambda_{mp}\mathbf{x}_{mp} \end{pmatrix} = \mathbf{P}\,\mathbf{X}_p \qquad \text{where} \qquad \mathbf{P} \equiv \begin{pmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_m \end{pmatrix}$$

The joint projection can be viewed as a projective injection mapping the 3D projective world bijectively to the **joint image** — a 3D projective subspace of $(3m-1)$-D projective joint image space [28, 27]. This is a faithful projective copy of the world expressed entirely in image coordinates. Projection from it to the individual images is a trivial forgetting of coordinates and scale factors. Projective reconstruction of the joint image amounts to recovering the missing depths λ_{ip}. This is a canonical process[2] up to a once-and-for-all choice of scales for the projections $\mathbf{P}_i$. The four

[2] 'Canonical' means that it characterizes the imaging geometry and is characterized uniquely (up to the scales) by it; it does not depend on the world or image coordinate systems used; and it is in some sense the 'natural' arena of action for *any* reconstruction method.

columns of the joint projection matrix form a spanning basis for the joint image. The coordinates of a rescaled joint image point with respect to this basis are exactly the corresponding 3D point's homogeneous world coordinates. But neither the basis nor the world coordinates are canonical: only the geometric position of the point in the joint image is recoverable from the image data.

The above geometry can be converted directly to algebra. The 4×4 minors (submatrix determinants) of the joint projection encode the location of the joint image (and hence the projective camera geometry) in a well-defined algebraic sense: they are its 'Grassmann-Plücker coordinates'. Moreover, the minors turn out to be just the components of the **matching tensors** between the images. These generate the multilinear constraints that tokens in different images must satisfy if they are to be the projections of a single world token. They can also be used for projective depth recovery, and to transfer tokens between images. There are four basic types of matching tensors: **epipoles** $\mathbf{e}_{ij}$ (tensorially: $\mathbf{e}_i^{A_j}$), **fundamental matrices** $\mathbf{F}_{ij}$ ($\mathbf{F}_{A_iB_j}$), **trivalent tensors** $\mathbf{G}_{A_i}{}^{B_jC_k}$ and **quadrivalent tensors** $\mathbf{H}^{A_iB_jC_kD_l}$. These are formed from minors with respectively 3+1, 2+2, 2+1+1, and 1+1+1+1 rows from 2, 2, 3 and 4 images i, j, k, l [22, 8, 28].

The 'joint image closure relations' that underlie the new reconstruction methods are bilinear constraints between projection matrices and the corresponding matching tensors. They guarantee that the projections are coherent with the joint image subspace defined by the tensors. Algebraically, they express the four-dimensionality ("closure") of the joint image. The simplest way to derive them is to append any column of the $3m \times 4$ joint projection matrix to the existing matrix, to form a rank deficient $3m \times 5$ matrix. The 5×5 minors of this matrix vanish. Expand by cofactors in the appended column. The coefficients are matching tensor components (4×4 minors of the original joint projection matrix). Closer examination reveals five basic types of relation. We use only the simplest two here[3]:

$$\mathbf{F}_{ji}\,\mathbf{P}_i + [\mathbf{e}_{ij}]_\times \mathbf{P}_j \;=\; \mathbf{0} \qquad \text{F-e closure} \tag{1}$$

$$\mathbf{G}_{B_j}{}^{A_iC_k}\,\mathbf{P}_a^{B_j} + \mathbf{e}_j^{A_i}\,\mathbf{P}_a^{C_k} - \mathbf{P}_a^{A_i}\,\mathbf{e}_j^{C_k} \;=\; \mathbf{0} \qquad \text{e-G-e closure} \tag{2}$$

These relations provide constraints between matching tensors (which can be estimated from the image data) and columns of the joint projection matrix. For each column, (1) contains 3 constraints of which 2 are linearly independent, while (2) contains $3 \times 3 = 9$ constraints of which 5 are linearly independent. By accumulating enough of these constraints, we can solve linearly for the four $3m$-component joint projection columns, up to an overall 4×4 linear transformation that amounts to a homography of the reconstructed world space. Geometrically, the joint image (the 4D subspace spanned by the columns of the joint projection) is the null space of the constraints. Given the projections, the scene reconstruction can be completed by linearly back-projecting image structure into the world space, which amounts to solving redundant linear equations

$$\mathbf{x}_{ip} \wedge (\mathbf{P}_i \mathbf{X}_p) = \mathbf{0} \tag{3}$$

for the world points $\mathbf{X}_p$ in terms of their images $\mathbf{x}_{ip}$ and the projection matrices $\mathbf{P}_i$.

The **depth recovery relations** used for projective factorization [25, 30, 27] follow directly from the above closure constraints. Attaching a world point $\mathbf{X}_p$ to each projection gives bilinear constraints between the matching tensors and the *correctly rescaled* image points $\lambda_{ip}\mathbf{x}_{ip} \equiv \mathbf{P}_i\mathbf{X}_p$:

$$\mathbf{F}_{ji}\,(\lambda_{ip}\mathbf{x}_{ip}) + \mathbf{e}_{ij} \wedge (\lambda_{jp}\mathbf{x}_{jp}) \;=\; \mathbf{0} \tag{4}$$

$$\mathbf{G}_{B_j}{}^{A_iC_k}\,(\lambda_j \mathbf{x}^{B_j}) - (\lambda_i \mathbf{x}^{A_i})\,\mathbf{e}_j^{C_k} + \mathbf{e}_j^{A_i}\,(\lambda_k \mathbf{x}^{C_k}) \;=\; \mathbf{0} \tag{5}$$

Given the matching tensors, a coherent set of projective depths for the images of each world point can be recovered linearly using these relations. These already contain a virtual projective reconstruction, implicit in the fact that the rescaled data matrix (2) has rank 4. The reconstruction can be consolidated and 'read off' by any convenient matrix factorization algorithm [25, 30].

[3] $[\mathbf{x}]_\times$ denotes the skew 3×3 matrix giving the vector cross product: $[\mathbf{x}]_\times \mathbf{y} = \mathbf{x} \wedge \mathbf{y}$.

Another way to express (1) is to note that $\mathbf{F}_{ji}$ has rank 2 and hence can be decomposed (non-uniquely) as $\mathbf{F}_{ji} = \mathbf{u}_j\mathbf{v}_i^\top - \mathbf{v}_j\mathbf{u}_i^\top$. Here, $\mathbf{u}_i \leftrightarrow \mathbf{u}_j$ and $\mathbf{v}_i \leftrightarrow \mathbf{v}_j$ turn out to be corresponding pairs of epipolar line-vectors (with appropriate relative scaling), and hence $\mathbf{e}_{ij} = \mathbf{u}_j \wedge \mathbf{v}_j$, $\mathbf{e}_{ji} = \mathbf{v}_i \wedge \mathbf{u}_i$. Suitable $\mathbf{u}$'s and $\mathbf{v}$'s are easily obtained by rescaling the SVD basis of $\mathbf{F}_{ji}$. Since $[\mathbf{e}_{ij}]_\times = \mathbf{u}_j\mathbf{v}_j^\top - \mathbf{v}_j\mathbf{u}_j^\top$, the combined $\mathbf{F}$-$\mathbf{e}$ closure constraints from images i-j and j-i have rank just 2 and are spanned by the rows of a 2×6 matrix $\mathbf{U}_{ij}$:

$$\begin{pmatrix} \mathbf{F}_{ji} & [\mathbf{e}_{ij}]_\times \\ [\mathbf{e}_{ji}]_\times & \mathbf{F}_{ij} \end{pmatrix} = \begin{pmatrix} -\mathbf{v}_j & \mathbf{u}_j \\ \mathbf{v}_i & -\mathbf{u}_i \end{pmatrix} \mathbf{U}_{ij} \qquad \text{where} \qquad \mathbf{U}_{ij} = \begin{pmatrix} \mathbf{u}_i^\top & \mathbf{u}_j^\top \\ \mathbf{v}_i^\top & \mathbf{v}_j^\top \end{pmatrix}$$

In fact, the $\mathbf{u}$'s and $\mathbf{v}$'s extracted from the SVD of $\mathbf{F}_{ji}$ combine to form a basis of the 2D orthogonal complement of the i-j joint image. (The space spanned by the 4 columns of the i-j joint projection matrix $\binom{\mathbf{P}_i}{\mathbf{P}_j}$, or equivalently by those of the i-j rescaled data matrix $\binom{\lambda_{i1}\mathbf{x}_{i1} \;\cdots\; \lambda_{in}\mathbf{x}_{in}}{\lambda_{j1}\mathbf{x}_{j1} \;\cdots\; \lambda_{jn}\mathbf{x}_{jn}}$). Hence, another way to obtain the constraint matrix $\mathbf{U}_{ij}$ is to use any two image reconstruction method (*e.g.* factorization) and extract the left null space of the resulting i-j joint projection or rescaled data matrix, *e.g.* by QR or SVD.

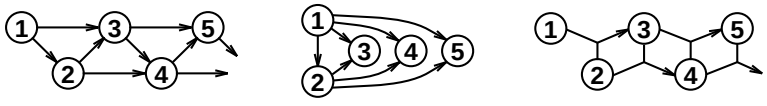
Similarly, the $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ closure constraint (2) can be written (in 3×3 blocks) as a 9×9 rank 5 matrix

$$\left(\begin{array}{c|c|ccc} -\mathbf{e}_j^{x_k}\,\mathbf{I}_{3\times3} & \mathbf{G}_\bullet{}^{x_k} & \mathbf{e}_{ji} & \mathbf{0} & \mathbf{0} \\ -\mathbf{e}_j^{y_k}\,\mathbf{I}_{3\times3} & \mathbf{G}_\bullet{}^{y_k} & \mathbf{0} & \mathbf{e}_{ji} & \mathbf{0} \\ -\mathbf{e}_j^{z_k}\,\mathbf{I}_{3\times3} & \mathbf{G}_\bullet{}^{z_k} & \mathbf{0} & \mathbf{0} & \mathbf{e}_{ji} \end{array}\right) \begin{pmatrix} \mathbf{P}_i \\ \mathbf{P}_j \\ \mathbf{P}_k \end{pmatrix} = \mathbf{0}$$

Here, the 27 components of $\mathbf{G}_{A_j}{}^{B_iC_k}$ are viewed as three 3×3 matrices, for $C_k = x, y, z$. As before, the rank remains 5 even if further bilinear or trilinear closure constraints are added for the same images taken in a different order (but *cf.* the discussion on scaling below). Any rank 5 decomposition $\mathbf{U}_{ijk}$ of this constraint matrix (*e.g.* by SVD) gives a trivalent equivalent of the above $\mathbf{U}_{ij}$ matrix. For any such $\mathbf{U}_{ijk}$, each of its 5 rows contains three 3-component row vectors which define a matching triplet of image lines, and hence a corresponding 3D line. (If $\{\mathbf{u}_i, \mathbf{u}_j, \mathbf{u}_k\}$ is such a triplet, the closure constraint says that the pulled-back visual planes meet in a common 3D line: $(\mathbf{u}_i\mathbf{P}_i) + (\mathbf{u}_j\mathbf{P}_j) + (\mathbf{u}_k\mathbf{P}_k) = \mathbf{0}$). The 4D projective space of linear combinations of these 5 line-triplet vectors bijectively spans the entire 4D space (Plücker quadric) of lines in 3D, *except* that the correspondence is singular for lines in the trifocal plane.

The complete closure-based reconstruction process runs roughly as follows. A very large number of closure constraints is available, relating the projections of any selection of 2, 3, or even (for higher closure constraints) 4 or 5 images. It would be impractical to enforce all of these, but in any case they are highly redundant and only a small subset of them need be used in practice. The choice must depend on the correspondences and matching tensors available, convenience, and a run time *vs.* redundancy trade-off. To fully constrain the projections, each image (except the first pair) must be related to *at least* two others. This can be done with one $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ constraint or two $\mathbf{F}$-$\mathbf{e}$ ones, in either their full or reduced ($\mathbf{U}$-matrix) versions. (The experiments below use the full versions).

This paper considers only the simplest possible choices, based on minimal sets of constraints for the first two types of closure relation. Each image is connected to exactly two previous ones in a chain. The following types of chain have been considered



F-e serial **F-e parallel** **e-G-e serial**

Serial chains connect each image to the two immediately preceding ones, while parallel ones connect each image to two 'key frames'. For the $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ chains, the trivalent tensor based in (with

covariant index in) the middle image of the triplet is used, *e.g.* , $\mathbf{e}_2^{A_1} - \mathbf{G}_{B_2}{}^{A_1 C_3} - \mathbf{e}_2^{C_3}$ for images 1-2-3. Note that the basic formulation is symmetric in that it allows any pair or triplet of images to be incorporated. Choosing a particular constraint topology breaks this symmetry, but the choice is at least under user control (modulo suitable estimates of the matching tensors).

Each constraint contributes several rows to a big $3m$-column, m image constraint matrix (unused elements are zero). It is essential to choose consistent relative scalings (see below), but once this is done the constraint matrix generically has rank $3m-4$. Its null space is exactly the joint image (the 4D space spanned by the joint projection columns). Any basis for the null space provides four $3m$-component column vectors that can be regarded as the columns of a valid reconstructed joint projection. The freedom of choice in the basis corresponds to a 4×4 nonsingular mixing of the columns, which amounts to a projective deformation of the reconstructed world coordinates.

The above process enforces a particular relative scaling for the projection matrices, so it is necessary to choose coherent scalings for the overlapping constraint equations. In fact, matching tensors inherit 'natural' scalings from their definitions as minors of projection matrices, but these are lost when they are estimated from image data. The closure relations depend critically on these scalings, so the relevant part of them must be recovered.

It turns out that the scales can be chosen arbitrarily modulo one constraint for each closed loop in the above chains. The same constraints guarantee the existence of consistent choices of depths in the depth recovery equations (4) or (5), and it turns out to be easiest to recover the scalings using this. For each closed loop, scalings are chosen arbitrarily and the depths of (a selection of) measured image points are propagated around the loop by a chain of depth recovery steps (*cf.* [25]). Then, one of the tensor scales is modified to make the average 'closed-loop gain' unity, as it must be for consistency. For the $\mathbf{F}$-$\mathbf{e}$ constraint this involves 3-image loops (*e.g.* $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$), while for the $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ one we multiply (5) by $[\mathbf{e}_{21}]_\times$ so that only two terms survive, and then propagate through just two images (*e.g.* $2 \rightarrow 3 \rightarrow 2$). The required epipoles are also estimated from $\mathbf{G}$ and (5), by multiplying by $[\mathbf{x}_1]_\times$ or $[\mathbf{x}_3]_\times$ and solving. The epipoles and scalings could also be found bilinearly from $\mathbf{G}$ alone, but for maximum stability I prefer to use linear methods based on the image data.

Numerically, once the combined constraint matrix has been assembled there are several ways to calculate its null space. The experiments reported here use the four smallest singular vectors of the SVD, but eigendecomposition of the normal matrix gives similar results. These methods are numerically stable and easily handle redundant constraints, but all of them are rather slow when there are many images, as large matrices with many zeros are involved. With sparse sets of constraints (as here), the null-space could also be estimated using various sparse or recursive methods. These should be much faster than the full SVD, although some stability may be lost — more investigation is needed here.

In fact, it is clear (in retrospect) from the above discussion that one can also view closure-based reconstruction as a means of 'gluing together' many overlapping virtual 2 or 3 image reconstructions into a coherent multi-image whole. Each reconstruction implicitly provides a 6×4 or 9×4 joint projection matrix in some arbitrary world frame. The closure-based framework characterizes these by their 2 or 5 dimensional left null spaces. These have the advantage of being independent of the world frames chosen, and directly extractable from the matching tensors without passing through an explicit intermediate reconstruction. Finally, the accumulated null space constraints are re-inverted to give the combined joint projection matrix. In retrospect, it is unclear whether passing through a large $(3m-4)$-D null space computation is an effective means of patching together several (implicit) 4D partial reconstructions. This must rest as a subject for future work.

In practice, the $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ method turns out to be quite a lot slower than the $\mathbf{F}$-$\mathbf{e}$ one, mainly because larger matrices are involved at each step. However it is also significantly more stable. In particular, for a camera moving in a straight line, the fundamental matrices and epipoles of different images coincide. This is a well-known singular case for epipolar-line-based token transfer, and $\mathbf{F}$-$\mathbf{e}$ closure based reconstruction fails here too. The failure is intrinsic to any method based solely

on epipolar geometry (rather than image measurements). Camera zooms centred on the unique epipole leave the epipolar geometry unchanged and hence can not be recovered. (The problem still exists for two images, but there it can be absorbed by a 3D homography). In contrast, trivalent transfer and $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ reconstruction are well behaved for aligned centres, as is reconstruction by $\mathbf{F}$-$\mathbf{e}$ depth recovery and factorization. Basically, some information about positions along epipolar lines is needed to stabilize things. This can be provided by trivalent transfer, or even better by anchoring onto explicit image correspondences.

3 Implementation

Now we summarize the reconstruction algorithms, and discuss a few important implementation details. The $\mathbf{F}$-$\mathbf{e}$ closure algorithm has the following steps:

0) Extract and match features between images.
1) Standardize the image coordinates (see below).
2) Estimate fundamental matrices and epipoles connecting each image to at least two others.
3) Correct the scales of the fundamental matrices and epipoles using (4) (*cf.* section 2).
4) Build the constraint matrix of equations (1) and use SVD to find its 4D null space.
5) Extract the projection matrices from the null space column vectors.
6) Back-project and solve for 3D structure using (3).
7) De-standardize the projection matrices (see below).

The $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ closure based method follows the same pattern, except that: (*i*) both point and line features can be used to estimate the trivalent tensors; (*ii*) equation 5 is used to correct the trivalent scaling, and equation (2) to build the constraint matrix.

The current implementations use linear methods to estimate fundamental matrices and trivalent tensors. With properly standardized coordinates these turn out to be very stable and surprisingly accurate [12]. Using a nonlinear least squares iteration to refine the estimates marginally improves the stability of (for example) the long serial chains of the $\mathbf{F}$-$\mathbf{e}$ method, but not enough to change the basic conclusions. The linear method for $\mathbf{F}$ includes a final 3×3 SVD to enforce $\det \mathbf{F} = 0$ and calculate the epipoles. The epipoles for the $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ method are found linearly from $\mathbf{G}$ and the image data using (5).

For accurate results it is *essential* to work in a well-adapted coordinate system. This is standard numerical practice, but it is particularly important when there are implicit least-squares trade-offs between redundant constraints, as here. If some components of the input vectors are typically much larger than others — for example when homogeneous pixel coordinates $(x, y, z) \sim (256, 256, 1)$ are used — some constraints have a much higher implicit weight than others and this significantly distorts the estimated solution. Hartley has underlined the importance of this for fundamental matrix estimation [12], and it is equally true for reconstruction. In practice it makes little difference which of the many possible standardization schemes is used. Here, the pixel coordinates are scaled uniformly into the unit square $[-1, 1] \times [-1, 1]$, homogenized, and normalized as 3-vectors to norm 1. This is easy, fast, independent of the image, and works equally well for visible and off-image virtual points (*e.g.* distant vanishing points or epipoles). Figure 1 shows the effect of standardization: pixel coordinates (scale ~ 256) give reconstructions hundreds of times worse than well standardized ones (scale ~ 1). The error rises rapidly at scales below 10^{-1} owing to (32 bit) floating point truncation error.

4 Experiments

To help quantify the performance of the algorithms, I have run a series of simulations using synthetic data. The algorithms have also been tested on hand-matched points extracted from real

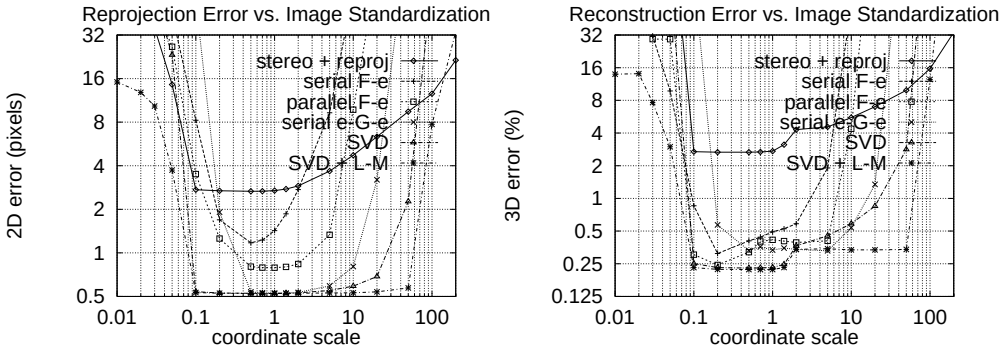


Figure 1: Mean reprojection and reconstruction error *vs.* image coordinate standardization.

images, and an implementation on 'live' images is in progress. The simulations are based on trial scenes consisting of random 3D points in the unit cube. These are viewed by identical perspective cameras spaced evenly along a 90° arc of radius 2, looking directly at the centre of the scene. These are ideal conditions for accurate reconstruction, but many other configurations have also been tested, including infinitesimal viewing angles and distant scenes with negligible perspective. When cameras are added, their spacing is decreased so that the total range of viewing angles remains the same. The positions of the projected image points are perturbed by uniform random noise. Mean-square (and median and maximum) 2D reprojection and 3D reconstruction errors are accumulated over 50 trials. The 3D error is the residual after projective alignment of the reconstruction with the scene. Unless otherwise stated, default values of 10 views, 50 points and ± 1 pixel noise are used.

Figure 2 summarizes the results, giving image reprojection and 3D reconstruction errors *vs.* image noise, number of points and number of views. The new techniques under test are serial and parallel chain $\mathbf{F}$-$\mathbf{e}$ closure, and serial chain $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ closure. For comparison, several existing techniques are also shown.

Evidently, the most stable techniques are 'SVD' and 'SVD+L-M': SVD-based projective factorization [25, 30], and a Levenberg-Marquardt-like nonlinear least squares algorithm initialized from this. However, remember that these are only applicable when points can be matched across all images, while the other techniques require matches across only 2-3 images[4].

The '2 image' methods simply reconstruct the scene from two images, and then reproject to estimate the projection matrices for the remaining ones. The 'serial 2 image' method uses only the first two images, and hence involves a considerable amount of extrapolation. This can be very inaccurate, but it is realistic in the sense that practical two image methods are often restricted to nearby images when tracking is difficult. The serial $\mathbf{F}$-$\mathbf{e}$ and $\mathbf{e}$-$\mathbf{G}$-$\mathbf{e}$ closure methods fuse a series of small, inaccurate steps of this sort and still manage to produce significantly better results, despite the potential for accumulation of errors.

In contrast, the 'parallel 2 image' method uses the first and last images of the sequence, and hence maintains a constant baseline. The same applies to the 'parallel $\mathbf{F}$-$\mathbf{e}$' closure method, which links each image to the two end ones. These results require unrealistically wide matching windows, but they provide a clear indication of the "integrating power" of the closure formalism. In particular, adding more images does continue to improve the 'parallel $\mathbf{F}$-$\mathbf{e}$' closure results, while the 'parallel 2 image' results stay roughly constant (as expected). However, the closure method seems to need about 10 images just to overcome the extreme stability of the 2 image factorization method.

All of the methods scale linearly with noise and initially improve as more points are added, but

[4]To allow fair comparison, the point reconstruction step for each method has been allowed to combine data from all the images using the recovered projections.

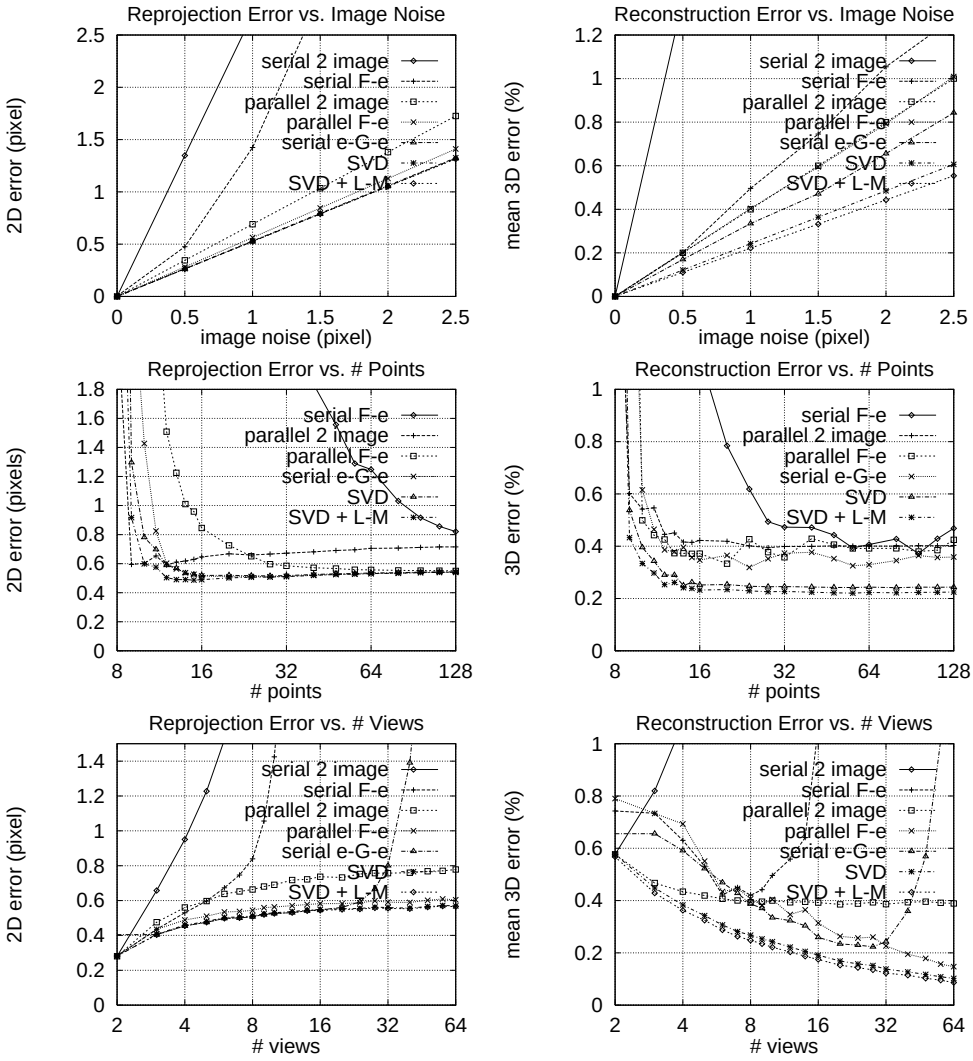


Figure 2: Mean reprojection and reconstruction error *vs.* noise, number of points and number of views.

level off after about 20 points. The serial methods eventually worsen as more images are added and their baseline decreases: the '2 image' one immediately (as expected); the **F-e** one after about 10 images; and the **e-G-e** one after about 30. In general, the trivalent methods are significantly more stable than the fundamental matrix ones. It definitely pays to select images as widely separated as possible for the closure constraints, even if this means having to use several 'key' images. The instabilities arising from long chains seem to be far greater than any biases introduced by working from 'key' images. However, tracking reliability puts strong practical limitations on the separations that can be attained.

All of the methods are stable for both close and distant scenes (modulo straight line motion for **F-e** closure), but all of them (especially the fundamental matrix ones) give very poor results for points near the axis of fronto-parallel motion, as there is no stereo baseline there for point reconstruction. (Surface continuity constraints are essential in this case).

One reason for the early failure of **F-e** closure is the fact that it is singular whenever three adjacent camera centres are aligned. This happens to an increasing extent as the spacing along the circular baseline decreases, adding to the natural uncertainty associated with the short baseline

itself. For this reason, it is advisable to use the **e**-**G**-**e** method (or an equivalent **U** matrix derived from reconstruction of at least 3 images) whenever straight line motions are involved.

The factorization method is notable for being linear yet close to optimal. It is based on **F**-**e** depth recovery (4) — essentially the same equations as the **F**-**e** closure based method, but applied directly to the image points rather than to the projections. Clearly, the direct use of image data gives a significant improvement in accuracy. Unfortunately, factorization is practically limited as it requires every token to be visible in every image: this is why the closure-based methods were developed.

5 Summary

The closure relation based projective reconstruction techniques work reasonably well in practice, except that the **F**-**e** method fails for aligned camera centres. If there are many images, closure is more accurate than the common 'reconstruct from 2 images and reproject for the other projections' paradigm, but it can not compete with projective factorization when features can be tracked through all the images. In principle there is no need to single out 'privileged' features or images. But short chains of closure relations turn out to be significantly more stable than long ones, so in practice it is probably best to relate all of the images to a few 'key' ones (or perhaps hierarchically). The trivalent techniques are slower, but significantly more stable than the fundamental matrix based ones.

Future work will implement the methods on real images, investigate fast recursive solutions of the reconstruction equations, study the stabilizing effects of incorporating redundant constraints, and compare the closure-based methods with direct techniques for merging several partial reconstructions.

References

[1] K. B. Atkinson. *Close Range Photogrammetry and Machine Vision*. Whittles Publishing, Roseleigh House, Latheronwheel, Caithness, Scotland, 1996.

[2] S. Carlsson. Multiple image invariance using the double algebra. In J. Mundy, A. Zissermann, and D. Forsyth, editors, *Applications of Invariance in Computer Vision*, volume 825 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[3] S. Carlsson. Duality of reconstruction and positioning from projective views. In P. Anandan, editor, *IEEE Workshop on Representation of Visual Scenes*. IEEE Press, 1995.

[4] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[5] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, 1993.

[6] O. Faugeras. Stratification of 3-d vision: Projective, affine, and metric representations. *J. Optical Society of America*, A 12(3):465–84, March 1995.

[7] O. Faugeras, Q.-T. Luong, and S. J. Maybank. Camera self calibration: Theory and experiments. In G. Sandini, editor, *European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[8] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 951–6, Cambridge, MA, June 1995.

[9] W. Förstner. 10 pros and cons of performance characterization in computer vision. In *Workshop on Performance Characterization of Vision Algorithms*, Cambridge, U.K., 1996.

[10] R. Hartley. Euclidean reconstruction from multiple views. In 2^{nd} *Europe-U.S. Workshop on Invariance*, pages 237–56, Ponta Delgada, Azores, October 1993.

[11] R. Hartley. Lines and points in three views – an integrated approach. In *Image Understanding Workshop*, Monterey, California, November 1994.

[12] R. Hartley. In defence of the 8-point algorithm. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 1064–70, Cambridge, MA, June 1995.

[13] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 761–4, Urbana-Champaign, Illinois, 1992.

[14] R. Hartley and P. Sturm. Triangulation. In *ARPA Image Understanding Workshop*, pages 957–66, Monterey, November 1994.

[15] A. Heyden. Reconstruction from image sequences by means of relative depths. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 1058–63, Cambridge, MA, June 1995.

[16] A. Heyden and K. Åström. A canonical framework for sequences of images. In *IEEE Workshop on Representations of Visual Scenes*, Cambridge, MA, June 1995.

[17] K. Kanatani. *Geometric Computation for Machine Vision*. Oxford University Press, 1993.

[18] P. F. McLauchlan and D. W. Murray. A unifying framework for structure and motion recovery from image sequences. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 314–20, Cambridge, MA, June 1995.

[19] R. Mohr, B. Boufama, and P. Brand. Accurate projective reconstruction. In 2^{nd} *Europe-U.S. Workshop on Invariance*, page 257, Ponta Delgada, Azores, October 1993.

[20] J. Oliensis and V. Govindu. Experimental evaluation of projective reconstruction in structure from motion. Technical report, NEC Research Institute, 4 Independence Way, Princeton N.J., 1995.

[21] C. J. Poelman and T. Kanade. A parapersective factorization method for shape and motion recovery. In J-O. Eklundh, editor, *European Conf. Computer Vision*, pages 97–108, Stockholm, 1994. Springer-Verlag.

[22] A. Shashua. Algebraic functions for recognition. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 1995.

[23] G. Sparr. A common framework for kinetic depth, reconstruction and motion for deformable objects. In J-O. Eklundh, editor, *European Conf. Computer Vision*, pages 471–82, Stockholm, 1994. Springer-Verlag.

[24] G. Sparr. Simultaneous reconstruction of scene structure and camera locations from uncalibrated image sequences. In *Int. Conf. Pattern Recognition*, Vienna, 1996.

[25] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conf. Computer Vision*, pages 709–20, Cambridge, England, 1996. Springer-Verlag.

[26] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Computer Vision*, 9(2):137–54, 1992.

[27] B. Triggs. The geometry of projective reconstruction I: Matching constraints and the joint image. Submitted to *Int. J. Computer Vision*.

[28] B. Triggs. Matching constraints and the joint image. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 338–43, Cambridge, MA, June 1995.

[29] B. Triggs. Autocalibration and the absolute quadric. Submitted to CVPR'97, Puerto Rico, 1996.

[30] B. Triggs. Factorization methods for projective structure and motion. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 845–51, San Francisco, CA, 1996.

[31] M. Werman and A. Shashua. The study of 3D-from-2D using elimination. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 473–9, Cambridge, MA, June 1995.

[32] A. Zisserman and S.J. Maybank. A case against epipolar geometry. In 2^{nd} *Europe-U.S. Workshop on Invariance*, pages 69–88, Ponta Delgada, Azores, October 1993.