Some Notes on Factorization Methods for Projective Structure and Motion

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

Abstract

We summarize the main known properties of factorization-based projective structure from motion, including the basic formulation, depth estimation and how it can sometimes be avoided, and some suggestions about statistical properties.
Keywords: structure from motion, projective reconstruction, factorization methods, affine approximation.

1 Multi-image Projection

We assume the **linear perspective** model of image formation, in which 3D points are imaged by finding where the straight ray joining them to a given 3D **viewing point** or **optical centre** cuts a given 3D **image plane**. **Homogeneous coordinates** allow this to be written very simply in terms of matrix multiplication [5]:

$$\lambda \boldsymbol{x} = \mathbf{P}\,\mathbf{x}$$

where $\mathbf{x} \simeq (x\ y\ z\ 1)^\top$ are the homogeneous coordinates of the 3D point $(x\ y\ z)$; $\boldsymbol{x} \simeq (u\ v\ 1)^\top$ are the homogeneous coordinates of the corresponding 2D image point $(u\ v)^\top$; $\mathbf{P}$ is a 3×4 rank 3 homogeneous **projection matrix** representing the camera; '$\simeq$' denotes equality of vectors or matrices modulo arbitrary positive rescaling; and the **projective depth** λ of $\boldsymbol{x}$ is its associated scale factor in the image projection equation. More generally, given n 3D points $\mathbf{x}_p, p = 1 \ldots n$ projected into m perspective images by projection matrices $\mathbf{P}_i, i = 1 \ldots m$, the individual projections $\lambda_{ip}\,\boldsymbol{x}_{ip} = \mathbf{P}_i\,\mathbf{x}_p$ can be assembled into a single big $3m \times n$ **joint projection** equation [15, 18]:

$$\begin{pmatrix} \lambda_{11}\,\boldsymbol{x}_{11} & \ldots & \lambda_{1n}\,\boldsymbol{x}_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1}\,\boldsymbol{x}_{m1} & \ldots & \lambda_{mn}\,\boldsymbol{x}_{mn} \end{pmatrix}_{3m\times n} = \begin{pmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_m \end{pmatrix}_{3m\times 4} \begin{pmatrix} \mathbf{x}_1 & \ldots & \mathbf{x}_n \end{pmatrix}_{4\times n}$$

We will write this compactly as $\mathbf{M} = \mathbf{P}\,\mathbf{X}$ where: the $3m\times n$ **depth-measurement** matrix $\mathbf{M}$ contains the rescaled image points $\lambda_{ip}\,\boldsymbol{x}_{ip}$; the $3m \times 4$ **joint projection matrix** $\mathbf{P} = (\mathbf{P}_1\ \ldots\ \mathbf{P}_m)^\top$

[1 Oct 98] This is a draft of my contribution to a joint paper on SFM methods, which is being written as a deliverable for the Esprit LTR project CUMULI (first author: A. Heyden). It is not intended to be an independent, finished paper in its own right. I've made it available because it summarizes the main properties of the projective factorization approach, which might be useful to someone.

characterizes the viewing geometry (camera positions and calibrations); and the $4 \times n$ **homogeneous shape matrix** $\mathbf{X} = (\mathbf{x}_1 \ \dots \ \mathbf{x}_n)$ characterizes the 3D point geometry. The joint projection equation implies that modulo uncertainty in the measurements $\boldsymbol{x}_{ip}$ and depths λ_{ip}, the depth-measurement matrix M has rank ≤ 4. This is immensely important because — given depth estimates and modulo problems with missing data and a slightly suboptimal error model — it allows reconstruction to be reduced to the well-studied classical problem of matrix factorization for which effective, reliable numerical methods exist [16, 15, 18, 14, 8]. The two main steps of 3D reconstruction thus become **depth recovery** and **limited rank factorization**. We will study each of these in turn, but first we turn to the properties of projective depths.

2 Projective Depth

The projective depths λ depend entirely on the normalization conventions adopted for their $\boldsymbol{x}$, $\mathbf{x}$ and $\mathbf{P}$. With **affine normalization** (last component scaled to 1) for image points, the depths becomes simply $\lambda = \mathbf{p}_{\text{opt}}\mathbf{x}$ where $\mathbf{p}_{\text{opt}}$ is the camera's **optical plane** — the third row of $\mathbf{P}$, denoting the plane parallel to the image one through the camera's optical centre. Assume that $\mathbf{x}$ is also normalized affinely. If $\mathbf{p}_{\text{opt}}$ is the **plane at infinity** $\mathbf{p}_\infty = (0\ 0\ 0\ 1)$, we say that the camera is **affine** and the depth of any finite point becomes simply 1. Hence, no depth recovery step is needed and factorization can start immediately [16]. Otherwise, $\mathbf{p}_{\text{opt}}$ is finite and (in a Euclidean frame) we can normalize so that its first three components form a unit vector. In this case $\lambda = \mathbf{p}_{\text{opt}}\mathbf{x}$ becomes the traditional **optical depth** — the orthogonal distance of $\mathbf{x}$ from the optical plane $\mathbf{p}_{\text{opt}}$.

Another useful normalization introduces a 3D reference plane $\mathbf{p}_{\text{ref}}$ and point $\mathbf{x}_{\text{ref}}$ and demands that points $\mathbf{x}$ and cameras $\mathbf{P}$ be normalized so that $\frac{(\mathbf{p}_{\text{ref}}\mathbf{x})(\mathbf{p}_{\text{opt}}\mathbf{x}_{\text{ref}})}{\mathbf{p}_{\text{ref}}\mathbf{x}_{\text{ref}}} = 1$ (*e.g.*, choose $\mathbf{p}_{\text{ref}}\mathbf{x}_{\text{ref}} = 1$ and normalize cameras by $\mathbf{p}_{\text{opt}}\mathbf{x}_{\text{ref}} = 1$ and points by $\mathbf{p}_{\text{ref}}\mathbf{x} = 1$). Then the depth $\lambda = \mathbf{p}_{\text{opt}}\mathbf{x}$ can be identified with the cross ratio $(\mathbf{p}_{\text{opt}}, \mathbf{p}_{\text{ref}}; \mathbf{x}, \mathbf{x}_{\text{ref}}) \equiv \frac{(\mathbf{p}_{\text{opt}}\mathbf{x})(\mathbf{p}_{\text{ref}}\mathbf{x}_{\text{ref}})}{(\mathbf{p}_{\text{opt}}\mathbf{x}_{\text{ref}})(\mathbf{p}_{\text{ref}}\mathbf{x})}$. If $\mathbf{p}_{\text{opt}} \to \mathbf{p}_{\text{ref}}$ or $\mathbf{x} \to \mathbf{x}_{\text{ref}}$ the depth reduces to 1. For $\mathbf{p}_{\text{ref}} \to \mathbf{p}_\infty$ the usual affine 3D point normalization arises, and $\mathbf{p}_{\text{opt}} \to \mathbf{p}_{\text{ref}} = \mathbf{p}_\infty$ is the case of the affine camera. The camera normalization $\mathbf{p}_{\text{opt}}\mathbf{x}_{\text{ref}} = 1$ is well behaved for distant cameras, unlike the usual Euclidean unit vector one. Usually, $\mathbf{x}_{\text{ref}}$ would be taken near the centre of the working volume and $\mathbf{p}_{\text{ref}}$ near infinity.

For factorization based reconstruction it is convenient to leave the 3D point and camera normalizations floating until after the points and projection matrices have been recovered. The scale freedoms $\mathbf{P}_i \to \mu_i\, \mathbf{P}_i$, $\mathbf{x}_p \to \nu_p\, \mathbf{x}_p$ allow any estimated set of depths to be freely rescaled according to $\lambda_{ip} \to \mu_i\, \nu_p\, \lambda_{ip}$, *i.e.* each row and each column of the the $m \times n$ depth matrix (λ_{ip}) can be freely rescaled without damaging the rank 4 factorization property. For example, we could set any chosen row and any chosen column arbitrarily, say to ones. In practice, we use this rescaling freedom to make the error model implicit in the factorization mimic to the natural image error — see 'balancing' below. Although the mn depths are only defined modulo these $m+n-1$ arbitrary scale factors, their remaining $mn - (m + n - 1) = (m - 1)(n - 1)$ d.o.f. do still carry the projective 'signature' of the point-camera distances, and hence give the information required for reconstruction.

3 Avoiding Depth Estimation

Above we saw that for affine cameras (*i.e.* ones with optical planes at infinity, $\mathbf{p}_{\text{opt}} = \mathbf{p}_\infty$), the normalizations could be chosen to reduce all of the projective depth scale factors to unity, $\lambda_{ip} = 1$. Since this eliminates the need for a depth recovery step, it is worth considering when it happens — or can be made to. Consider two points in two cameras. With affine image point normalization, the cross ratio $(\mathbf{p}_{\mathrm{opt}_1}, \mathbf{p}_{\mathrm{opt}_2}; \mathbf{x}_1, \mathbf{x}_2) = \frac{(\mathbf{p}_{\mathrm{opt}_1}\mathbf{x}_1)(\mathbf{p}_{\mathrm{opt}_2}\mathbf{x}_2)}{(\mathbf{p}_{\mathrm{opt}_1}\mathbf{x}_2)(\mathbf{p}_{\mathrm{opt}_2}\mathbf{x}_1)}$ becomes $\frac{\lambda_{11}\lambda_{22}}{\lambda_{12}\lambda_{21}}$. The 3D point and camera normalizations can be chosen so that all four depths are one if and only if this cross ratio is also unity, which happens only when: (*i*) the points are coincident; (*ii*) the planes are coincident; or (*iii*) the line through the points intersects the line of intersection of the planes. If there are several non-colinear points and several cameras, their pairwise cross ratios can all be one only if: (*i*) all of the points lie on a plane through the common line of intersection of the optical planes; or (*ii*) all of the optical planes are coincident. Hence, for non-coplanar points and affine image normalization, the depths can be reduced to unity only when the geometry is *projectively equivalent* to the affine camera case, *i.e.* only for *arbitrary sets of projective cameras, all of whose optical planes coincide*.

We will call such camera geometries and the associated scene reconstructions **pseudo-affine**. They are not necessarily close to affine — the optical plane may be far from infinity — but they have many affine-like properties and ambiguities. Under any fixed but unknown 3D projectivity which moves the optical plane to infinity, a pseudo-affine geometry *becomes* affine, *i.e.* affine cameras viewing the projectively deformed scene produce exactly the same images as the original perspective cameras. Hence, affine matching, transfer and image synthesis methods all give *exact* results, and affine reconstruction methods give exact *projective* structure (*i.e.* structure modulo and affinity and the unknown 3D projectivity that pulls the reconstructed optical plane back in from infinity).

Pseudo-affine geometries have *all of their epipoles at infinity*, since the camera centres all lie on the common optical plane, which projects to the image line at infinity. Examples include side-by-side parallel axis stereo rigs, and sideways looking cameras in moving cars. Conversely, any camera geometry with at least one non-colinear camera centre and all of its epipoles at infinity is pseudo-affine. More generally, any set of cameras with coplanar centres — including *any* 2 or 3 camera geometry — can be reduced to pseudo-affinity by warping the images to move the line of epipoles to infinity, or more simply by using the line of epipoles[1] $\boldsymbol{l}$ *as* the line at infinity, *i.e.* normalizing image points to $\boldsymbol{l}\boldsymbol{x} = 1$. Clearly, the epipoles (or at least their common line) are needed, and the results are ill-conditioned for 3D points near the plane of centres.

Even when the optical planes are not coincident, (pseudo-)affine geometry may still be a good approximation. If all of the cross ratios $(\mathbf{p}_{\mathrm{opt}_i}, \mathbf{p}_{\mathrm{opt}_j}; \mathbf{x}_p, \mathbf{x}_q)$ are within ϵ of one, there is a choice of scales with all $\lambda_{ip} = 1 + \mathcal{O}(\epsilon)$. Under the pseudo-affine approximation the depths λ_{ip} are set to 1. The depth-measurement matrix $\mathbf{M}$ becomes simply the measurement matrix $(\boldsymbol{x}_{ip})$, but deviates systematically from the true, rank 4 $\mathbf{M}$ by $\mathcal{O}(\epsilon)$. This implies (see below) that the resulting reconstruction has a reprojected image error and (at best, if the camera-scene geometry is strong) a relative 3D reconstruction error of roughly the same order. By expanding $(\mathbf{p}_{\mathrm{opt}_i}, \mathbf{p}_{\mathrm{opt}_j}; \mathbf{x}_p, \mathbf{x}_q) - 1$ in an affinizing frame, one can show that for any set of distant cameras viewing a compact object, the bias of the pseudo-affine approximation is $\epsilon \approx \mathcal{O}(\frac{\text{object diameter}}{\text{object/optical-plane distance}})$. Geometrically this means that *perspective effects must be small*[2], *i.e.* there must be a projective frame in which all camera planes are near the plane at infinity, in comarison to the size of the object. Similarly, for cameras whose optical planes are nearly aligned we can measure their separation by placing an arbitrary origin among them, to get a bound $\epsilon = \mathcal{O}(\frac{\text{max. origin/optical-plane distance}}{\text{min. scene/optical-plane distance}} + (\text{angle between optical planes}))$. So even for non-compact scenes, the pseudo-affine approximation is good for any set of cameras whose optical planes are nearly parallel, and separated by much less than the scene-camera distance. Rectification to parallel geometry — or more simply normalizing image points against a nominal, roughly vertical

[1]For coincident epipoles (2 images or colinear centres) any epipolar line can be used as $\boldsymbol{l}$, so long as the corresponding line is used in all the other images.

[2]**Perspective** denotes the slight convergence (by $\mathcal{O}(\frac{\text{object size}}{\text{camera distance}})$) of the optical rays joining 3D points to the camera centre, as they cross the object. For an affine camera, the centre is infinitely distant and the rays become parallel.

epipolar line — may significantly improve the pseudo-affine approximation by removing the angular term. Note that all of these bounds depend only on the optical planes, not on the positions of the optical centres on them.

For image synthesis and low precision 3D applications, the small systematic errors introduced by the (pseudo-)affine camera approximation are often quite acceptable. If not, it can often be used to initialize some iterative method that takes better account of perspective effects [18, 12]. However it is essential to use 3D normalizations and parametrizations that are continuous near infinity, *e.g.* the affine camera should transition smoothly to a perspective one. Also beware that relief-reversed reconstructions with reversed camera displacements often give a second local minimum which should be checked for[3].

4 Depth Estimation

This section develops closed form estimators for the projective depths λ_{ip}, for cases where they are not constants. Any projective reconstruction method can be used for depth estimation, simply by reconstructing and then reprojecting to find the relevant depths. Conversely, the depths carry a latent reconstruction so any depth estimator can be viewed as an implicit reconstructor. Most methods take a small fixed number of images or points, and implicitly reconstruct (or at least find depths for) all points in those images, or all images having those points. Hence, projective factorization amounts to using matrix factorization to consolidate a 'patchwork' of implicit partial projective reconstructions into a coherent whole, averaging errors across many points and images and 'reading off' the structure implicit in the depth-measurement matrix in a convenient form. This is entirely analogous to 'hallucination'-like methods for dealing with missing data by patching together several partial reconstructions [16, 19, 11]. It is at best an effective but slightly clumsy heuristic, 'not truly multi-image', with few if any provable optimality properties, and which can only be safely extended over a limited number of 'patches'. Fortunately, such 'coherency extension' heuristics usually allow overlapping, redundant constraints to be incorporated so that — at the cost of implementation complexity and extra computation — a more rigidly cross-linked structure can be created wherever there is data to support it. For example, for two image partial reconstruction on chains of more than about 30 images, merging the partial reconstructions from adjacent images gives somewhat 'floppy' results, while adding even a few selected, more distant pairs greatly enhances rigidity.

Consider the joint projection matrix $\mathbf{P} = (\mathbf{P}_1 \ \ldots \ \mathbf{P}_m)^\top$. This characterizes the camera geometry, but it is only defined up to a 4×4 projectivity representing the arbitrary choice of 3D coordinates. For image-based calculations a more intrinsic 3D-coordinate-free representation is useful. This can be had by forming the 4×4 determinants of all possible choices of four rows of $\mathbf{P}$ — the **Grassmann coordinates** of $\mathbf{P}$. By the multiplication rule for determinants, the results are independent of the choice of 3D frame modulo an overall rescaling. And they do still characterize the camera geometry, as can be seen by choosing a frame in which (say) the first four rows of $\mathbf{P}$ take some canonical form like $\mathbf{I}_{4\times4}$, when determinants of other rows against three of these four just reproduce components of $\mathbf{P}$. However the resulting representation is massively redundant with $\binom{3m}{4}$ components rather than $3m \times 4$, and in fact must satisfy a set of very complicated bilinear **Grassmann identities**. Nevertheless, the results are useful because they

[3] *C.f.* the relief-reversed 'far side' of the bas relief ambiguity. If detected, this can be corrected by relief-reversing the reconstruction, reversing the estimated camera displacements, and refitting. It is actually an *exact* ambiguity for any number of affine or projective cameras, related to the projective and affine coordinate inversion symmetry $(x\; y\; z\; 1)^\top \to (-x \ -y \ -z\; 1)^\top$. But parametrizations that keep the camera centres 'on this side of (spherical projective) infinity' destroy the symmetry and leave a displaced, inequivalent local minimum.

can be estimated directly from image measurements — in fact they are the components of the well known **matching tensors**: **epipoles** $\boldsymbol{e}$ are formed from determinants with 3 rows in one image and 1 in another, **fundamental matrices** $\boldsymbol{F}$ from $2+2$ rows in two images, **trifocal tensors** $\boldsymbol{G}$ from $2+1+1$ rows in three images, and **quadrifocal tensors** $\boldsymbol{H}$ from $1+1+1+1$ rows in four images. Note that — analogous to the projective depths of image points — the matching tensors inherit a coherent relative scaling from their definition in terms of projections. Each image projection has an arbitrary scale factor, but for m images there are many more than m matching tensors so their scales can not all be set independently. For many applications (in particular redundant projective depth estimation) it is essential to estimate coherent scale factors for the matching tensors involved. For the moment, suppose this to have been done.

Now consider the $3m \times 4$ joint projection $\mathbf{P}$, and append to it another copy of any column of $\mathbf{P}$. The resulting $3m \times 5$ matrix obviously has rank 4, so all of its 5×5 minors vanish. Expand these minors by their last columns, giving sums $\sum(\text{entry of } \mathbf{P}) \cdot (4 \times 4 \text{ minor of } \mathbf{P}) = 0$. Rewriting the 4×4 minors as matching tensors gives bilinear **closure relations** between the matching tensors and projections $\mathbf{P}$. As the last column was chosen arbitrarily, these are tensorial in both the matching tensors and projections. There are five types of relation in all [17], the simplest two being[4]

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

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

Contraction with some 3D point $\mathbf{x}$ (or equivalently, using $\mathbf{P}\,\mathbf{X}$ as the fifth column above) gives five types of **depth recovery relation**, bilinear between *correctly rescaled* image points and matching tensors [17, 15, 18]. The simplest two are

$$\boldsymbol{F}_{ji}\, (\lambda_i \boldsymbol{x}_i) - \boldsymbol{e}_{ij} \wedge (\lambda_j \boldsymbol{x}_j) \;=\; 0 \qquad \boldsymbol{F}\text{-}\boldsymbol{e} \text{ depth recovery} \tag{3}$$

$$\boldsymbol{G}_{B_j}{}^{A_i C_k}\, (\lambda_j \boldsymbol{x}^{B_j}) - (\lambda_i \boldsymbol{x}^{A_i})\, \boldsymbol{e}_j^{C_k} + \boldsymbol{e}_j^{A_i}\, (\lambda_k \boldsymbol{x}^{C_k}) \;=\; 0 \qquad \boldsymbol{e}\text{-}\boldsymbol{G}\text{-}\boldsymbol{e} \text{ depth recovery} \tag{4}$$

These relations allow projective depths to be recovered linearly from image points and matching tensor estimates. For example, the F-e relation gives a least squares solution [15]

$$\frac{\lambda_j}{\lambda_i} \;\approx\; \frac{(\boldsymbol{e}_{ij} \wedge \boldsymbol{x}_j)^\top \boldsymbol{F}_{ji}\, \boldsymbol{x}_i}{\|\boldsymbol{e}_{ij} \wedge \boldsymbol{x}_j\|^2}$$

Note that since $\boldsymbol{F}_{ji}\, \boldsymbol{x}_i$ and $\boldsymbol{e}_{ij} \wedge \boldsymbol{x}_j$ represent the same epipolar line, only their relative scales are constrained. As $\|\boldsymbol{F}_{ji} \boldsymbol{x}_i\|$ is also proportional to the distance of $\boldsymbol{x}_i$ from the epipole $\boldsymbol{e}_{ji}$, this says roughly that the relative depth is proportional to the relative distance of the points from the two epipoles along their respective epipolar lines.

Similarly, the three image relations (4) provide $3 \times 3 = 9$ linear constraints on their 3 depths (only 2 of which are linearly independent for exact $\boldsymbol{G}$, $\boldsymbol{e}$ and $\boldsymbol{x}$). These are easily solved in linear least squares for any two of the depths in terms of the remaining one, or for the single unknown depth if two of them have previously been estimated. Alternatively, pre- and/or post-multiplication by appropriate $[\boldsymbol{e}]_\times$ and $[\boldsymbol{x}]_\times$ matrices eliminates one of the terms, reducing the problem to a straightforward 2 depth one.

If the images are closely spaced, it may be preferable to use the infinitesimal form of the epipolar constraint [21, 22, 9] to estimate depth. This applies to points $\boldsymbol{x} = (x\ y\ 1)^\top$ and their measured image velocities $\dot{\boldsymbol{x}} = (\dot{x}\ \dot{y}\ 0)^\top$, rather than to pairs of corresponding image points $\boldsymbol{x}, \boldsymbol{x}'$. The fundamental matrix $\boldsymbol{F}$ is replaced by $(\boldsymbol{e}, \boldsymbol{S})$ where $\boldsymbol{e}$ is the infinitesimal epipole (focus

[4] $[\boldsymbol{x}]_\times$ denotes the skew 3×3 matrix giving the vector cross product: $[\boldsymbol{x}]_\times \boldsymbol{y} = \boldsymbol{x} \wedge \boldsymbol{y}$. We use tensor notation for the e-G-e equation — don't worry about this if you can't follow it.

of expansion) and $\boldsymbol{S}$ is a symmetric 3×3 matrix satisfying $\boldsymbol{e}^\top \boldsymbol{S}\, \boldsymbol{e} = 0$, which compensates for the infinitessimal camera rotation. $(\boldsymbol{e}, \boldsymbol{S})$ are defined only up to a common rescaling. It is convenient to normalize them so that $\boldsymbol{e}$ is a unit vector $\|\boldsymbol{e}\|^2 = 1$. The **infinitesimal depth recovery relation** then becomes (*c.f.* (3) and [22] eqns. (19), (33), (36) and (45)):

$$(\frac{\dot{\lambda}}{\lambda} + \mu)\, \boldsymbol{e} \wedge \boldsymbol{x} \;=\; (\boldsymbol{I} - \boldsymbol{e}\, \boldsymbol{e}^\top)\, \boldsymbol{S}\, (\boldsymbol{I} + \boldsymbol{e}\, \boldsymbol{e}^\top)\, \boldsymbol{x} - \boldsymbol{e} \wedge \dot{\boldsymbol{x}}$$

Here, everything is a function of time t. $\mu(t)$ is arbitrary but the same for all points in image t — it is related to the arbitrary choice of scale for projection matrix $\mathbf{P}(t)$. Multiplying by $\boldsymbol{x}^\top$ and using symmetry and $\boldsymbol{e}^\top \boldsymbol{S}\, \boldsymbol{e} = 0$ gives the **infinitesimal epipolar constraint** [21]

$$\boldsymbol{x}^\top \boldsymbol{S}\, \boldsymbol{x} \;=\; \boldsymbol{x}^\top (\boldsymbol{e} \wedge \dot{\boldsymbol{x}})$$

Solving the depth recovery equation in least squares gives

$$\lambda(t) = \lambda(0) \exp\left(-\int_0^t \left(\mu + \frac{(\boldsymbol{e} \wedge \boldsymbol{x})^\top \left(\boldsymbol{e} \wedge \dot{\boldsymbol{x}} - \boldsymbol{S}\, (\boldsymbol{I} + \boldsymbol{e}\, \boldsymbol{e}^\top) \boldsymbol{x} \right)}{\|\boldsymbol{e} \wedge \boldsymbol{x}\|^2} \right) dt \right)$$

Now consider how to apply the above depth recovery equations in practice. The simplest approach is to arrange the images in a tree structure, assign arbitrary (say, unit) depths in the root image, and propagate these recursively to each image in turn using estimated matching tensors and the depth recovery relations at each node. The simplest trees are a serial chain linking each image to the previous one, and a parallel fan linking all subsequent images to the first one. These are the two extremes of tree-like connectivity, the one treating each image uniformly but having to propagate depths through many images, the other relying on a single 'hub' image, but having maximally short propagation chains. As might be expected from its wider baselines and shorter chains, the parallel structure turns out to be geometrically the stronger and is recommended for normal use. However serial chains also perform surprisingly well and seem to be usable for up to about 20–40 images (depending on geometry, noise levels, number of matches, *etc.*). One common advantage of working serially — the fact that correspondence is easier so that more feature matches are available — seems to be of relatively little benefit for pure factorization methods, as these in any case require feature matches across *all* images. However, serial tracking is probably the most effective means of establishing the required correspondences in the first place [16].

More generally, one could include additional, redundant depth estimation equations to strengthen the geometry. In general, a (sparse) homogeneous least squares calculation would then be needed to solve for the depth vector of each point. The only difficulty is that the matching tensors must be assigned consistent scales to make the depths found by different paths in the interconnectivity graph agree. Heuristically, this can be arranged by choosing an independent set of loops spanning those of the graph, and rescaling one tensor in each loop so that depths propagated via the left side of the loop agree (on average over all features) with those propagated via the right side. In the tree case there are no loops to cause a conflict, so the matching tensor scales are arbitrary. However it should be emphasized that in practice, the parallel tree structure usually produces results so close to optimal (as measured by comparing residual errors with a projective bundle adjustment), that there seems to be little need for the additional redundancy.

Another way to do reconstruction [19, 20] is to assemble a sufficient set of the closure constraints (1), (2) into a large linear equation $\mathbf{A}\mathbf{P} = 0$, find the joint projection $\mathbf{P}$ as the 4D null space of this matrix, then estimate the structure $\mathbf{X}$ by back projection given $\mathbf{P}$. This turns out to be less stable than factorization as the image measurements are only used indirectly, but it easily handles missing data as only matching tensor estimates are required. Each image must be linked to *two* previous ones (whereas one suffices for factorization), so coherent scales must be estimated for the matching tensors. Also, if the optical centres are aligned the trifocal method must be used as the epipolar one is singular. See [20] for details. Jacobs has developed a somewhat similar method that uses (rescaled) image measurements instead of matching tensors [11].

Another, more trivial case in which the depths are easily found is *coplanar points*. There is no need for reconstruction here as the projective 3D structure is already visible in each image. However consistent depths allow a rank 3 factorization to consolidate the image measurements into a unified planar structure estimate, with associated plane-to-image homography estimates. Appropriate depths can be derived from (the last rows of) estimated inter-image homographies $\boldsymbol{H}$. Since $\boldsymbol{x}' \simeq \boldsymbol{H}\boldsymbol{x}$, a consistent depth assignment is $\lambda'\boldsymbol{x}' = \boldsymbol{H}(\lambda\boldsymbol{x})$, *i.e.* $\frac{\lambda'}{\lambda} = \frac{\boldsymbol{x}'^\top \boldsymbol{H}\boldsymbol{x}}{\|\boldsymbol{x}'\|^2}$ or $\frac{\lambda'}{\lambda} = \frac{\boldsymbol{l}_\infty{}' \boldsymbol{H}\boldsymbol{x}}{\boldsymbol{l}_\infty{}'\boldsymbol{x}'}$.

5 Factorization

Now consider the factorization stage of reconstruction for a complete, known depth-measurement matrix $\mathbf{M}_{3m\times n} = (\lambda_{ip}\,\boldsymbol{x}_{ip})_{i=1\ldots m,p=1\ldots n}$ (*i.e.* the depths λ_{ip} have already been recovered). $\mathbf{M}$ theoretically has rank 4 but in practice noise in the image measurements $\boldsymbol{x}_{ip}$ and recovered depths λ_{ip} destroys this property. Also, suppose that we manage to decompose $\mathbf{M}$ approximately into a bilinear rank 4 product $\mathbf{M}_{3m\times n} \approx \mathbf{P}_{3m\times 4}\mathbf{X}_{4\times n}$, where the $\mathbf{P}$ and $\mathbf{X}$ factors play the roles respectively of the joint projection matrix and the homogeneous shape matrix. Without further constraints on $\mathbf{P}$ and $\mathbf{X}$, such a factorization can at best be unique up to a 4×4 linear redefinition $\mathbf{P} \to \mathbf{P}\,\mathbf{H}_{4\times 4}^{-1}$, $\mathbf{X} \to \mathbf{H}_{4\times 4}\mathbf{X}$ which leaves $\mathbf{P}\,\mathbf{X}$ unchanged, and hence is 'as good' as the original factorization. However, to the extent that we can identify the $\mathbf{P}$ and $\mathbf{X}$ factors with valid joint projection and shape estimates, $\mathbf{H}$ corresponds to a projective change of 3D coordinates, so this ambiguity would occur in any projective reconstruction method.

The basic numerical tool for matrix factorization is the **singular value decomposition (SVD)**. Given m, n and an arbitrary input matrix $\mathbf{A}_{m\times n}$, SVD decomposes it into $\mathbf{U}_{m\times n}\mathbf{S}_{n\times n}\mathbf{V}_{n\times n}^\top$ where $\mathbf{U}$ and $\mathbf{V}$ have orthonormal columns and $\mathbf{S} = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ is a diagonal, positive semi-definite matrix containing the **singular values** of $\mathbf{A}$. If the singular values σ_i are distinct and sorted in decreasing order, the decomposition is unique up to the mutual signs of corresponding columns of $\mathbf{U}$ and $\mathbf{V}$. The SVD's importance for matrix factorization stems from a backwards error theorem [6] saying that if $\mathrm{rank}(\mathbf{A}) \geq k$, the rank k matrix closest to $\mathbf{A}$ in either the L_2 or Frobenius norms[5] is that produced by zeroing all but the k largest singular values in $\mathbf{S}$, and then recalculating $\mathbf{USV}^\top$. If there are repeated singular values the solution may not be unique, but zeroing still produces a possible one. As the columns of $\mathbf{U}$ and $\mathbf{V}$ corresponding to zero singular values can be discarded, the rank k SVD becomes $\mathbf{U}_{m\times k}\,\mathbf{S}_{k\times k}\,\mathbf{V}_{n\times k}^\top$ where the columns of $\mathbf{U}_{m\times k}$ and $\mathbf{V}_{n\times k}$ are still orthonormal and $\mathbf{S}_{k\times k}$ is still diagonal. Given this, it is easy to absorb $\mathbf{S}$ into either $\mathbf{U}$ or $\mathbf{V}$ to make a bilinear rank k decomposition of the form $\mathbf{A}_{m\times n} \approx \mathbf{B}_{m\times k}\mathbf{C}_{k\times n}$, which can be applied directly to the visual reconstruction problem.

The above theorem says that a rank 4 SVD finds $\mathbf{P}$, $\mathbf{X}$ for which the reconstituted 'reprojection' $\mathbf{P}\,\mathbf{X}$ is 'as close as possible' to the noisy input matrix $\mathbf{M}$. However the error is measured in the abstract L_2 or Frobenius norms on $\mathbf{M}$, not in the natural image error metric. Moreover, the theorem only considers *reprojection* error — it says nothing at all about the accuracy of the individual factors $\mathbf{P}$, $\mathbf{X}$ *as a 3D reconstruction*. We will consider the second point first.

[5]The L_2 matrix norm is the maximum factor by which the matrix can inflate a vector, with vector size measured by the usual L_2 vector norm $\|\mathbf{v}\|_2^2 = \sum_i \mathbf{v}_i^2$: $\|\mathbf{A}\|_2 \equiv \max_{\mathbf{v}\neq 0} \frac{\|\mathbf{A}\mathbf{v}\|_2}{\|\mathbf{v}\|_2} = \sigma_1(\mathbf{A})$. The Frobenius matrix norm is simply $\|\mathbf{A}\|_F^2 = \sum_{ij} \mathbf{A}_{ij}^2 = \sum_i \sigma_i^2(\mathbf{A})$.

If we measure reconstruction errors relatively in terms of $\|\Delta\mathbf{P}\|_2/\|\mathbf{P}\|_2$ and $\|\Delta\mathbf{X}\|_2/\|\mathbf{X}\|_2$, it can be shown that relative noise of order $\epsilon = \|\Delta\mathbf{M}\|_2/\|\mathbf{M}\|_2$ changes the recovered projective structure by at most $\mathcal{O}(\|\Delta\mathbf{M}\|_2/\sigma_4) = \mathcal{O}(\epsilon\,\sigma_1/\sigma_4)$. Here, $\sigma_1 = \|\mathbf{M}\|_2$ and σ_4 are the largest and smallest nonzero singular values of the true $\mathbf{M}$. (The measured $\mathbf{M}$ can also be used provided $\epsilon \ll \frac{\sigma_4}{\sigma_1}$). The condition number or noise amplification factor σ_1/σ_4 measures the weakness of the camera-scene geometry. For example, to measure structure to relative precision p given relative image noise n ($= \frac{\text{pixels noise}}{\text{image width}} \approx 10^{-5}$ to 10^{-2} for typical applications), we need geometric strength $\sigma_4/sigma_1 \geq \mathcal{O}(n/p)$. The L_2 norms $\|\Delta\mathbf{P}\|_2$ and $\|\Delta\mathbf{X}\|_2$ are not physically very meaningful error metrics as they depend (among other things) on the recovered homogeneous scale factors, but we use them for want of something better in projective space.

The above result follows from the following SVD forwards error theorem [6]: A perturbation $\Delta\mathbf{M}$ causes singular value perturbations $|\Delta\sigma_i| \leq \|\Delta\mathbf{M}\|_2$ and $\sum_i |\Delta\sigma_i|^2 \leq \|\Delta\mathbf{M}\|_F^2$, and perturbs a singular subspace $\mathbf{W}$ for which $\|\Delta\mathbf{M}\|_2 < \gamma(\mathbf{W})$ by at most an angle θ where $|\sin\theta| \leq \frac{\|\Delta\mathbf{M}\|_2}{\gamma(\mathbf{W})-\|\Delta\mathbf{M}\|_2}$. Here $\mathbf{W}$ can be any subspace spanned by one or more columns of $\mathbf{U}$ or one or more columns of $\mathbf{V}$, and $\gamma(\mathbf{W})$ is the **absolute gap** between $\mathbf{W}$ and its complement, *i.e.* $\min_{ij} |\sigma_i - \sigma_j|$ where i ranges over the columns of $\mathbf{W}$ and j over the remaining columns.

Now consider how the SVD reprojection error $\Delta\mathbf{M} = \mathbf{P}\,\mathbf{X} - \mathbf{M}$ relates to the natural image error. SVD minimizes the Frobenius norm

$$\begin{aligned}
\|\Delta\mathbf{M}\|_F^2 &= \sum_{ip} \|\hat{\lambda}_{ip}\,\hat{\boldsymbol{x}}_{ip} - \lambda_{ip}\,\boldsymbol{x}_{ip}\|^2 \\
&= \sum_{ip} \lambda_{ip}^2\,\|\frac{\hat{\lambda}_{ip}}{\lambda_{ip}}\,\hat{\boldsymbol{x}}_{ip} - \boldsymbol{x}_{ip}\|^2 \\
&= \sum_{ip} \lambda_{ip}^2\,\|(\hat{\boldsymbol{x}}_{ip} - \boldsymbol{x}_{ip}) + (\frac{\hat{\lambda}_{ip}}{\lambda_{ip}} - 1)\,\hat{\boldsymbol{x}}_{ip}\|^2
\end{aligned}$$

where the hatted quantities are the 'reprojections' implied by $\mathbf{P}\,\mathbf{X}$. Hence, if we have estimated the depths accurately $\frac{\hat{\lambda}_{ip}}{\lambda_{ip}} \approx 1$, the SVD truncation error reduces to the sum of the natural uniform image errors $\|\hat{\boldsymbol{x}}_{ip} - \boldsymbol{x}_{ip}\|^2$, *weighted by projective depth*. So if we want the SVD to approximate the uniform image error metric, we should choose normalizations that make the depths *as close to unity as possible*. Above we saw that this happens *exactly* for (pseudo-)affine cameras, so SVD factorization for these produces results which are almost[6] optimal in the uniform image error metric. We also saw above that the normalization invariant cross ratios $(\mathbf{p}_{\mathrm{opt}_i}, \mathbf{p}_{\mathrm{opt}_j}; \mathbf{x}_p\mathbf{x}_q) = \frac{(\mathbf{p}_{\mathrm{opt}_i}\mathbf{x}_p)(\mathbf{p}_{\mathrm{opt}_j}\mathbf{x}_q)}{(\mathbf{p}_{\mathrm{opt}_i}\mathbf{x}_q)(\mathbf{p}_{\mathrm{opt}_j}\mathbf{x}_p)}$ are the only barrier to setting all the depths to one, so SVD can be made to roughly reproduce the uniform image error model only for geometries in which these are not too far from one.

Now consider the second term above, $(\hat{\lambda}_{ip}/\lambda_{ip} - 1)\,\hat{\boldsymbol{x}}_{ip}$. It might seem that since many depth recovery equations are chained together, the depth estimates λ_{ip} would soon deviate completely from their true values $\hat{\lambda}_{ip}$. Although this eventually happens, it takes a surprisingly long time — typically around 20–40 images. The main reason is that the depth error depends almost entirely on the component of $(\boldsymbol{x}_{ip} - \hat{\boldsymbol{x}}_{ip})$ *along* the epipolar line (the only other contribution is the point's small influence on the matching tensor estimates). For smoothly moving cameras, the epipolar geometry changes little between frames, so the depth error in one equation is almost entirely cancelled by an equal and opposite contribution of $(\boldsymbol{x}_{ip} - \hat{\boldsymbol{x}}_{ip})$ to the next — almost as if image i had been left out

[6]The only perturbations are the residual $(\hat{\lambda}_{ip}/1 - 1)$ terms, or in the Tomasi-Kanade approach which uses centred data and a rank 3 factorization, the reliance on the estimated centroid.

of the chain altogether. Hence, the error $(\hat{\lambda}_{ip}/\lambda_{ip} - 1)\,\hat{\boldsymbol{x}}_{ip}$ is smaller and significantly less biased than might be expected. It is also at least somewhat plausible that the correlation between the $\hat{\boldsymbol{x}}_{ip}$ and $(\hat{\boldsymbol{x}}_{ip} - \boldsymbol{x}_{ip})$ terms will often be rather small, as the random fluctuations in $(\hat{\boldsymbol{x}}_{ip} - \boldsymbol{x}_{ip})$ should not be strongly correlated with the actual image position $\boldsymbol{x}_{ip}$. With this no correlation hypothesis, the error sum becomes approximately $\sum_{ip} \lambda_{ip}^2 \,\|\hat{\boldsymbol{x}}_{ip} - \boldsymbol{x}_{ip}\|^2 + (\hat{\lambda}_{ip} - \lambda_{ip})^2\,\|\hat{\boldsymbol{x}}_{ip}\|^2$, so the total SVD error is the total weighted image error plus the summed depth estimation errors $(\hat{\lambda}_{ip} - \lambda_{ip})^2$, weighted by the roughly known and well behaved quantities $\|\hat{\boldsymbol{x}}_{ip}\|^2$.

Practically, we can arrange to make the depths as close as possible to one by a heuristic **balancing** procedure. First the depths are estimated as above using the depth recovery equations, starting from 1 at the root image. Often they will be very unbalanced, shrinking or growing exponentially along image chains at a rate depending on the chosen scales of $\boldsymbol{F}$ and $\boldsymbol{e}$. (Normalizing to $\|\boldsymbol{F}\|_F^2 = 2$, $\|\boldsymbol{e}\|^2 = 1$ minimizes this). Then we use the notional scale freedoms $\mathbf{P}_i \to \mu_i\,\mathbf{P}_i$, $\mathbf{x}_p \to \nu_p\,\mathbf{x}_p$, rescaling each row of the $m \times n$ depth matrix (λ_{ip}) to a vector of length n, and then rescaling each resulting column to a vector of length m. The column scaling to some extent undoes the row one, but repeating the cycle two or three times (convergence is very rapid) moves the depths about as near to one on average as they can be. The results are then applied to the measurements $\boldsymbol{x}_{ip}$ and used for factorization. Alternatively, the depth-measurement matrix itself can be balanced by rescaling triplets of rows and columns, with much the same effect.

One further point is that when linearly estimating matching tensors, it is essential to work in well standardized image coordinates[7] [1, 7]. This guarantees 'good numerical conditioning' or more precisely that the various error contributions all have roughly their correct statistical weights. The same need occurs in depth recovery and in the factorization step, and it turns out that the same standardization is not too far from optimal in all three cases.

Iterative SVD: The depth-measurement matrices can become quite large. The SVD of an $m \times n$ matrix with full estimates of $\mathbf{U}$ and $\mathbf{V}$ takes between about 6 and 20 $mn\min(m,n)$ floating point operations, depending on the relative size of m and n (see [6] for detailed complexity estimates). For a $3m \times n$ rank 4 factorization, this is $\mathcal{O}(\min(3m,n))$ more work than we might hope for given the size of the input data, so it can be slow when there are both many points and many images. For large problems, it turns out that a limited rank factorization can be extracted fairly quickly and reliably by iterative SVD methods, provided that the above mentioned ratio $\|\Delta\mathbf{M}\|_2/\sigma_4 = \epsilon\sigma_1/\sigma_4$ (which determines whether stable reconstruction is possible under *any* SVD method) is not too close to 1.

The simplest algorithm of this type is called the **block power method**, **simultaneous** or **orthogonal iteration** (used, *e.g.* by [13]). Each iteration replaces a running estimate of $\mathbf{U}$ with the left factor of the QR or SVD of the $3m \times 4$ matrix $\mathbf{M}\,\mathbf{M}^\top\mathbf{U}$. Convergence is linear at rate $\|\Delta\mathbf{M}\|_2/\sigma_4$, but can be accelerated somewhat by appropriate shifting strategies (*i.e.* using instead $(\mathbf{M}\,\mathbf{M}^\top - s\mathbf{I})\,\mathbf{U}$ for shifts s designed to move unwanted singular values close to 0). Slightly more sophisticated is **Chebyshev iteration** which replaces the simple power iteration in $\mathbf{M}\,\mathbf{M}^\top$ with a two term (Chebyshev polynomial) recurrence. If good bounds are available for the unwanted singular values, this significantly enhances convergence 'because the Chebyshev is the smallest possible polynomial over the interval of unwanted values'. However the most efficient current iterative SVD methods use **Lanczos bidiagonalization** and its cousins. These are again two term recurrences, but with coefficients that adapt themselves automatically to the spectrum of $\mathbf{M}$, giving an optimal (in the Chebyshev sense) solution over the 'Krylov subspace' spanned by the generated powers $(\mathbf{M}\,\mathbf{M}^\top)^p\,\mathbf{U}_0$. All of these methods start from an initial vector or matrix $\mathbf{U}_0$. The choice

[7] I prefer to scale the image into the centred unit square $[-1,1] \times [-1,1]$, and then normalize homogeneous points according to $\|\boldsymbol{x}\|_2 = 1$ or $\|\boldsymbol{x}\|_\infty = 1$ (it makes little difference which).

is not critical but a good one helps, for example, the first $k = 4$ columns of a partial, pivoted Gram-Schmidt decomposition of $\mathbf{M}$. This is an active research field — see, *e.g.* [2, 3, 6, 10, 4] for details of these and other iterative methods. A good modern C code is M. Berry's SVDPACKC, available from NETLIB (`http://www.netlib.org`).

References

[1] P. Beardsley, A. Zisserman, and D. Murray. Navigation using affine structure from motion. In J-O. Eklundh, editor, *European Conf. Computer Vision*, volume II, pages 85–96, Stockholm, 1994. Springer-Verlag.

[2] M. Berry. Large scale singular value computations. *Int. J. Supercomputer Applications*, 6(1):13–49, 1992.

[3] M. Berry and R. Fierro. Low-rank orthogonal decompositions for information retrieval applications. Technical Report ut-cs-95-284, Computer Science Dept., U. of Tennessee, April 1995. To appear in *Numerical Linear Algebra with Applications*.

[4] Åke Björk. *Numerical Methods for Least Squares Problems*. SIAM Press, Philadelphia, PA, 1996.

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

[6] G. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1989.

[7] 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.

[8] A. Heyden. Projective structure and motion from image sequences using subspace methods. In *Scandinavian Conf. on Image Analysis*, pages 963–8, Lappeenranta, Finland, 1997.

[9] A. Heyden and K. Åström. Multilinear forms in the continuous case. In *IEEE Conf. Computer Vision & Pattern Recognition*, San Francisco, 1996.

[10] S. Van Huffel and J. Vandewalle. *The Total Least Squares Problem: Computational Aspects and Analysis*. SIAM Press, Philadelphia, PA, 1991.

[11] D. Jacobs. Linear fitting with missing data: Applications to structure from motion and to caharacterizing intensity images. In *IEEE Conf. Computer Vision & Pattern Recognition*, pages 206–212, Puerto Rico, 1997.

[12] F. Kahl and A. Heyden. Robust self-calibration and Euclidean reconstruction via affine approximation. In *Int. Conf. Pattern Recognition*, pages 47–55, Brisbane, Australia, 1998.

[13] T. Morita and T. Kanade. A sequential factorization method for recovering shape and motion from image streams. In *ARPA Image Understanding Workshop*, pages 1177–88, Monterey, November 1994.

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

[15] 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, U.K., 1996. Springer-Verlag.

[16] 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.

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

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

[19] B. Triggs. Linear projective reconstruction from matching tensors. In *British Machine Vision Conference*, pages 665–74, Edinburgh, September 1996.

[20] B. Triggs. Linear projective reconstruction from matching tensors. *Image & Vision Computing*, 15(8):617–26, August 1997.

[21] T. Viéville and O. Faugeras. Motion analysis with a camera with unknown and possibly varying intrinsic parameters. In E. Grimson, editor, *IEEE Int. Conf. Computer Vision*, pages 750–6, Cambridge, MA, June 1995.

[22] T. Viéville and O. Faugeras. The first order expansion of motion equations in the uncalibrated case. *Computer Vision and Image Understanding*, 64(1):128–46, 1996.