Projective Reconstruction*

Richard Hartley,
Australian National University; National ICT Australia

Synonyms. Projective structure and motion.

Related Concepts. Structure and motion; Euclidean reconstruction; visual odometry, Simultaneous Localization and Mapping (SLAM); bundle-adjustment.

Description. From several images of a scene and the coordinates of corresponding points identified in the different images, it is possible to construct a 3-dimensional point-cloud model of the scene, and compute the camera locations. From uncalibrated images the model can be reconstructed up to an unknown projective transformation, which can be upgraded to a Euclidean model by adding or computing calibration information.

1 Introduction

Projective reconstruction refers to the computation of the structure of a scene from images taken with uncalibrated cameras, resulting in a scene structure, and camera motion that may differ from the true geometry by an unknown $3D$ projective transformation.

Suppose that a set of interest points are identified and matched (or tracked) in several images. The configuration of the corresponding $3D$ points and the locations of the cameras that took these images are supposed unknown. The task of reconstruction is to determine the values of these unknown quantities.

Formally, assume that a set of image points $\{\mathbf{x}_{ij}\}$ are known, where $\mathbf{x}_{ij}$ represents the image coordinates of the j-th point seen in the i-th image. It is generally not required that every point's location be known in every image, so only a subset of all possible $\mathbf{x}_{ij}$ are given. The Structure from Motion (SfM) problem is to determine the camera projection matrices $\mathtt{P}_i$ and the $3D$ point locations $\mathbf{X}_j$ such that the projection of the j-th point in the i-th image is the measured $\mathbf{x}_{ij}$. Assuming a pinhole (projective) camera model, this relationship is expressed as a linear relationship

$$\mathbf{x}_{ij} = \mathtt{P}_i \mathbf{X}_j \ , \tag{1}$$

where $\mathtt{P}_i$ is a 3×4 matrix of rank 3, $\mathbf{X}_j$ and $\mathbf{x}_{ij}$ are expressed in homogeneous coordinates, and the equality is intended to hold only up to an unknown scale factor λ_{ij}. More precisely, therefore, the projection equation is

$$\lambda_{ij}\, \mathbf{x}_{ij} = \mathtt{P}_i \mathbf{X}_j \ . \tag{2}$$

In the SfM problem, cameras $\mathtt{P}_i$ and points $\mathbf{X}_j$ are to be determined, given only the point correspondences.

1.1 Homogeneous coordinates

Both $2D$ (image) points and $3D$ (world) points are most conveniently expressed in *homogeneous coordinates*. Thus, an image point $\mathbf{x}$ is represented by a 3-vector $\mathbf{x} = (u, v, w)^\top$, known as its homogeneous representation. The relationship to the standard Euclidean (non-homogeneous) coordinates (x, y) of the point is given by $x = u/w$ and $y = v/w$. This process of division by the final coordinate of the homogeneous vector is known as *dehomogenization*. Note that two vectors $\mathbf{x} = (u, v, w)^\top$ and $\mathbf{x}' = (u', v', w')^\top$ represent the same point in Euclidean coordinates if and only if $\mathbf{x} = k\mathbf{x}'$ for some non-zero constant k. Thus a given point may be expressed in infinitely many different ways in homogeneous coordinates. This is analogous with the way a given rational number has many different representations, such as $1/2 = 2/4 = 3/6 = k/2k$ for any k. One particularly convenient homogeneous representation of a point is the 3-vector with unit final coordinate: $(x, y, 1)^\top$.

Homogeneous coordinates (3-vectors) with final coefficient zero do not coincide to any real point in non-homogenous coordinates, since the process of dehomogenization involves division by zero. Such points are commonly known as points at infinity.

*To appear in Encyclopedia of Computer Vision, Springer 2013

The vector $(0,0,0)^\top$ is not considered to be a valid set of homogeneous coordinates.

In a similar way, $3D$ points are represented by homogeneous 4-vectors $\mathbf{X} = (\mathtt{X}, \mathtt{Y}, \mathtt{Z}, \mathtt{T})^\top$. The main advantage of using homogeneous coordinates to represent world and image points is that equation (1) has a particularly simple form as a linear relationship between the homogeneous coordinates of the points.

Two homogeneous vectors differing by a constant multiplicative factor are considered to be *equivalent* representations of the same point. The set of all equivalence classes of (non-zero) homogeneous $(n+1)$-vectors forms the *projective n-space, $\mathcal{P}^n$*. In studying projective reconstruction, it is conventional to consider image points to lie in projective 2-space $\mathcal{P}^2$, whereas $3D$ points lie in projective 3-space $\mathcal{P}^3$. This identifies the projective space $\mathcal{P}^2$ as consisting of the (image) plane, augmented with points at infinity. Similarly, $\mathcal{P}^3$ consists of $\mathbb{R}^3$ along with a plane of points at infinity.

2 Ambiguity

Expressed in full generality, the solution to the reconstruction problem may only be determined up to an unknown projective transformation, applied both to points and cameras.

A projective transformation of $\mathcal{P}^3$, the model for 3-space containing world points, is a mapping

$$\mathbf{X} \mapsto \mathtt{H}\mathbf{X}$$

where $\mathtt{H}$ is a non-singular 4×4 matrix representing a mapping between homogeneous coordinates. Using this relationship, it is easily seen that the determination of camera matrices $\mathtt{P}_i$ and points $\mathbf{X}_j$ cannot be unique, given only corresponding image coordinates $\mathbf{x}_{ij}$. Consider

$$\begin{aligned} \mathbf{x}_{ij} &= \mathtt{P}_i \mathbf{X}_j \\ &= (\mathtt{P}_i \mathtt{H}^{-1})(\mathtt{H}\mathbf{X}_j) \\ &= \mathtt{P}'_i \mathbf{X}'_j \,. \end{aligned} \tag{3}$$

In this relationship, new points $\mathbf{X}'_j = \mathtt{H}\mathbf{X}_j$ are defined in terms of points $\mathbf{X}_j$, and similarly new camera matrices $\mathtt{P}'_i = \mathtt{P}_i\mathtt{H}^{-1}$ in terms of the camera matrices $\mathtt{P}_i$. Since both $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ and $(\{\mathtt{P}'_i\}, \{\mathbf{X}'_j\})$ give rise to the same projected image coordinates $\mathbf{x}_{ij}$, there is no way to choose between these two solutions to the reconstruction problem. In fact, there exists a complete family of solutions to the problem, corresponding to all possible choices of the matrix $\mathtt{H}$. All such solutions are related to each other by the application of a projective transformation, and are hence called *projectively equivalent*. A particular solution, consisting of camera matrices $\mathtt{P}_i$ and points $\mathbf{X}_j$ satisfying (1) is known as a *projective reconstruction* of the scene, computed from the given corresponding image points.

The effect of projective ambiguity is given shown in fig 1.

2.1 The projective reconstruction theorem

The above analysis does not rule out the possibility that other solutions to this reconstruction problem exist, not related to a particular obtained solution by any projective transformation.

However, this possibility is excluded by the projective reconstruction theorem, which essentially says that if the set of corresponding points $\mathbf{x}_{ij}$ are sufficiently numerous (at least 8 in number), and do not lie in some degenerate configuration, then the solution to the reconstruction problem is unique up to a projective transformation.

The exact statement of the theorem requires the definition of the *fundamental matrix* which will be considered next.

3 Two view reconstruction

Consider the reconstruction problem for only two images. Rather than using a subscript, entities belonging to the second camera are distinguished by a prime. Thus, the given input to this problem consists of corresponding points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i;\ i = 1, \ldots, n$, where the points $\mathbf{x}_i$ come from one image and the $\mathbf{x}'_i$ are the corresponding points in the other.

Let the camera matrices (unknown) be $\mathtt{P}$ and $\mathtt{P}'$, and let $\mathbf{X}_i$ be the $3D$ point corresponding to the image points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. The projection equations are

$$\begin{aligned} \lambda_i \mathbf{x}_i &= \mathtt{P}\mathbf{X}_i \\ \lambda'_i \mathbf{x}'_i &= \mathtt{P}'\mathbf{X}_i \end{aligned}$$

where the scale factors λ_i and λ'_i are explicitly written (but are unknown). These equations may be written in a single system

$$\begin{bmatrix} \mathtt{P} & \mathbf{x}_i & \\ \mathtt{P}' & & \mathbf{x}'_i \end{bmatrix} \begin{pmatrix} \mathbf{X}_i \\ -\lambda_i \\ -\lambda'_i \end{pmatrix} = \mathbf{0} \,. \tag{4}$$

Since this equation must have a non-zero solution $({\mathbf{X}_i}^\top, -\lambda_i, -{\lambda'_i}^\top)^\top$, the determinant of the matrix on the left (which shall be denoted as $\mathtt{A}$) must be zero. Since the point

coordinates $\mathbf{x}_i$ and $\mathbf{x}'_i$ each appear in a single column, it follows that the determinant is a bilinear expression in $(\mathbf{x}_i, \mathbf{x}'_i)$, and hence the equation $\det(\mathtt{A}) = 0$ can be written in the form

$$\mathbf{x}_i'^{\top} \mathtt{F} \mathbf{x}_i = 0 \,, \tag{5}$$

where $\mathtt{F}$ is a 3×3 matrix depending only on the two camera matrices $\mathtt{P}$ and $\mathtt{P}'$. Consequently, this equation will hold for any pair of corresponding points $(\mathbf{x}_i, \mathbf{x}'_i)$. The matrix $\mathtt{F}$ is called the *fundamental matrix* corresponding to the camera pair $(\mathtt{P}, \mathtt{P}')$.

Closer examination of the matrix $\mathtt{A}$ appearing in (4) reveals the exact form of the matrix $\mathtt{F}$. Expanding $\det(\mathtt{A})$ by cofactors down the last two columns yields the following formula:

$$\mathtt{F}_{jk} = (-1)^{j+k} \det \begin{bmatrix} \mathtt{P}^{(\sim j)} \\ \mathtt{P}'^{(\sim k)} \end{bmatrix} \,, \tag{6}$$

where $\mathtt{P}^{(\sim j)}$ is the 2×4 matrix obtained by omitting the j-th row of $\mathtt{P}$, and $\mathtt{P}'^{(\sim k)}$ is similarly defined.

Another way of writing the equation (5) is

$$(\mathbf{x}_i \otimes \mathbf{x}'_i)^{\top} \mathbf{f} = 0 \,, \tag{7}$$

where $(\mathbf{x}_i \otimes \mathbf{x}'_i)^{\top}$ is the vector

$$(u'_i u_i,\ u'_i v_i,\ u'_i w_i,\ v'_i u_i,\ v'_i v_i,\ v'_i w_i,\ w'_i u_i,\ w'_i v_i,\ w'_i w_i) \tag{8}$$

expressed in terms of coordinates $\mathbf{x}_i = (u_i, v_i, w_i)^{\top}$ and $\mathbf{x}'_i = (u'_i, v'_i, w'_i)^{\top}$. Further, $\mathbf{f}$ is the vector $(\mathtt{F}_{11}, \mathtt{F}_{12}, \ldots, \mathtt{F}_{33})^{\top}$ made up of the entries of the fundamental matrix $\mathtt{F}$.

3.1 Computing the fundamental matrix

Note that the equation (7) is a linear equation with unknowns equal to the entries of the fundamental matrix. The explicit form of the equation is given by (8). Given $n \geq 8$ point correspondences, one has a set of linear equations

$$\mathtt{A}\mathbf{f} = \mathbf{0}$$

where $\mathtt{A}$ is an $n \times 9$ matrix, with entries determined by the coordinates of the matched image points. This set of equations is solved to find $\mathbf{f}$.

Since this is a set of homogeneous equations, there is a solution $\mathbf{f} = \mathbf{0}$, which is not interesting; a non-zero solution is required. With exactly 8 point correspondences, there is an exact solution to this problem. With more points, a least-squares solution is computed. This is most conveniently done by solving the problem

$$\begin{array}{ll} \textbf{Minimize} & \|\mathtt{A}\mathbf{f}\| \\ \textbf{subject to} & \|\mathbf{f}\| = 1 \,, \end{array}$$

where the condition $\|\mathbf{f}\| = 1$ is imposed in order to obtain a unique solution (apart from sign). The solution is the eigenvector of $\mathtt{A}^{\top}\mathtt{A}$ corresponding to the smallest eigenvalue. Alternatively, if $\mathtt{A}$ has singular value decomposition

$$\mathtt{A} = \mathtt{U}\mathtt{D}\mathtt{V}^{\top}$$

then the required $\mathbf{f}$ is the last column of $\mathtt{V}$ (assuming that the singular values of $\mathtt{D}$ are in descending order). Once the solution $\mathbf{f}$ is found, the fundamental matrix $\mathtt{F}$ is reconstituted from the entries of $\mathbf{f}$.

The algorithm just described is the so-called 8-point algorithm for computing the fundamental matrix [20]. In order to get good results, it is necessary to preprocess the input image coordinates, using the so-called *normalized* 8-point algorithm, which will be described later.

Projective Reconstruction Theorem. This discussion leads to the basic theorem of projective reconstruction, which states that under appropriate conditions, the reconstruction of a scene from sufficiently many point correspondences in two views is unique up to projective transformation.

Theorem 3.1. *Let $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$; $i = 1, \ldots, n$ be point correspondences in two views and let $(\mathtt{P}, \mathtt{P}', \{\mathbf{X}_i\})$ be a pair of camera matrices, and some $3D$ points forming a $3D$ reconstruction; specifically stated:*

$$\begin{array}{rcl} \lambda_i \mathbf{x}_i & = & \mathtt{P}\mathbf{X}_i \\ \lambda'_i \mathbf{x}'_i & = & \mathtt{P}'\mathbf{X}_i \end{array} \tag{9}$$

for some unknown $\lambda_i, \lambda'_i \neq 0$. Let $\mathtt{H}$ be an invertible 4×4 matrix $\mathtt{H}$, and define

$$\begin{array}{rcl} \tilde{\mathtt{P}} & = & \mathtt{P}\mathtt{H}^{-1} \\ \tilde{\mathtt{P}}' & = & \mathtt{P}'\mathtt{H}^{-1} \\ \tilde{\mathbf{X}}_i & = & \mathtt{H}\mathbf{X}_i \,. \end{array} \tag{10}$$

Then the triple $(\tilde{\mathtt{P}}, \tilde{\mathtt{P}}', \{\tilde{\mathbf{X}}_i\})$ is also a reconstruction satisfying the equations (9).

Furthermore, if the set of vectors $\{\mathbf{x}_i \otimes \mathbf{x}'_i\}$ has rank 8 (spans a linear subspace of dimension 8 in $\mathcal{R}^9$), then any reconstruction $(\tilde{\mathtt{P}}, \tilde{\mathtt{P}}', \{\tilde{\mathbf{X}}_i\})$ satisfying (9) is related to the original reconstruction $(\mathtt{P}, \mathtt{P}', \{\mathbf{X}_i\})$ by (10) for some non-invertible matrix $\mathtt{H}$.

This theorem was proved in [7, 13].

Note the condition that the set of vectors $\{\mathbf{x}_i \otimes \mathbf{x}'_i\}$ has rank 8 is exactly the condition that the set of equations of the form

(7) has a unique solution. If the rank of the vectors $\{\mathbf{x}_i \otimes \mathbf{x}'_i\}$ is equal to 9, then there is no solution to the equations (7) and the point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ can not be a valid set of points corresponding to projections of a set of $3D$ points in two images.

If the vectors $\{\mathbf{x}_i \otimes \mathbf{x}'_i\}$ span a space of dimension less than 8 (for instance if there are fewer than 8 point correspondences), then there is not a unique matrix $\mathtt{F}$ satisfying the condition (5), and the reconstruction may not be unique up to projectivity.

3.2 Extraction of Camera Matrices

Once the fundamental matrix has been computed, it is possible to extract a pair of camera matrices directly from $\mathtt{F}$. The decomposition is not unique, since according to Theorem 3.1 there are many pairs of camera matrices $(\mathtt{P}, \mathtt{P}')$ that correspond to the same fundamental matrix $\mathtt{F}$. It is always possible to assume that one of the camera matrices is of the form $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$, so the problem is simply to compute the other camera matrix $\mathtt{P}'$.

An algorithm to do this is as follows.

1. Compute the singular value decomposition

   $$\mathtt{F} = \mathtt{U}\mathtt{D}\mathtt{V}^\top$$

   where $\mathtt{D} \approx \mathrm{diag}(p, q, 0)$. Note that since $\mathtt{F}$ should have rank 2, the last singular value should be zero, but with noise this will not exactly hold.

2. Define matrices

   $$\mathtt{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad ; \quad \mathtt{Z} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

   Define $\widehat{\mathtt{D}} = \mathrm{diag}(p, q, r)$ for some value of r, and observe that

   $$\mathtt{F} = \mathtt{U}\mathtt{D}\mathtt{U}^\top = (\mathtt{U}\mathtt{Z}\mathtt{U}^\top)\,(\mathtt{U}\mathtt{W}^\top\widehat{\mathtt{D}}\mathtt{V}^\top) = \mathtt{S}\,\mathtt{M}\,. \qquad (11)$$

   where $\mathtt{S}$ is a skew-symmetric matrix, and $\mathtt{M}$ is defined by this equation. The value of r may be arbitrarily chosen.

3. A pair of camera matrices corresponding to the fundamental matrix $\mathtt{F}$ are now

   $$\mathtt{P} = [\mathtt{I} \mid \mathbf{0}] \quad ; \quad \mathtt{P}' = [\mathtt{M} \mid \mathbf{u}_3] \qquad (12)$$

   where $\mathbf{u}_3$ is the third column of $\mathtt{U}$.

Notes.

1. The vector $\mathbf{u}_3$ satisfies ${\mathbf{u}_3}^\top\mathtt{F} = {\mathbf{u}_3}^\top\mathtt{S} = \mathbf{0}$; it is the generator of the left null-space of $\mathtt{F}$.

2. The value of r, the last diagonal entry of $\widehat{\mathtt{D}}$, may be chosen arbitrarily, but a good choice is to set $r = (p+q)/2$ so that $\mathtt{M}$ is far from singular.

3. If $r = 0$, the matrix $\mathtt{M}$ is singular, but has a particularly simple form; namely $\mathtt{M} = \mathtt{S}\mathtt{F}$. The corresponding camera $\mathtt{P}' = [\mathtt{S}\mathtt{F} \mid \mathbf{u}_3]$ is sometimes used, but it has the property that the left-hand 3×3 block is singular, so the camera centre lies at a non-finite point.

3.3 Complete projective reconstruction algorithm

It is now possible to state a complete algorithm for projective reconstruction of a scene from two images. Suppose a set of image correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i;\ i = 1, \ldots, n$ are given.

1. From the image correspondences, compute the fundamental matrix $\mathtt{F}$ linearly from equations (7), as described in section 3.1.

2. From $\mathtt{F}$ find the two camera projection matrices $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$ and $\mathtt{P}' = [\mathtt{M} \mid \mathbf{t}]$, as in section 3.2.

3. The corresponding $3D$ points $\mathbf{X}_i$ may be computed linearly as the least-squares solution to equations (4). This process is called *triangulation*.

The linear triangulation method via equations (4) does not give optimal results. A method optimal in the presence of noise is given in [14, 15].

3.4 The normalized eight-point algorithm

It was pointed out in [11] that the simple version of the 8-point algorithm given above can lead to very poor results in some circumstances, but this problem is largely alleviated by simple normalization of the image coordinates.

The issue with the 8-point algorithm for computing $\mathtt{F}$ is that the vector (8) expressed in terms of image point coordinates can contain entries of widely different magnitude. This leads to poor conditioning of the linear equations used to solve for $\mathtt{F}$. In addition, the results are dependent on the particular coordinate system (origin and scale) used to express image points.

Given corresponding image points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ one may define normalized coordinates $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}'_i$ obtained from the original coordinates by the following operations.

1. Each $\mathbf{x}_i$ is replaced by $\mathbf{x}_i - \bar{\mathbf{x}}$, where $\bar{\mathbf{x}}$ is the mean (barycentre) of all the coordinates $\mathbf{x}_i$. This corresponds to a shift of the coordinate origin so that the mean of the $\mathbf{x}_i$ is at the origin.

2. The points are scaled so that their average (alternatively, their root-mean-squared) Euclidean distance from the origin is equal to $\sqrt{2}$. This is done by applying a common scaling to all the points $\mathbf{x}_i - \bar{\mathbf{x}}$. The resulting point is $\hat{\mathbf{x}}_i$.

The reason for choosing an average distance of $\sqrt{2}$ is so that the *average* point has homogeneous coordinates $(1, 1, 1)^\top$.

One applies these operations to the points $\mathbf{x}_i$ and $\mathbf{x}'_i$ independently, Note that both normalization steps are simple affine transformations of the points. These transformations may be written as

$$\hat{\mathbf{x}}_i = \mathtt{T}\mathbf{x}_i \quad ; \quad \hat{\mathbf{x}}'_i = \mathtt{T}'\mathbf{x}'_i \tag{13}$$

where $\mathtt{T}$ and $\mathtt{T}'$ are 3×3 matrices acting on the homogeneous representations of the points.

Once this normalization has taken place, the computation of the fundamental matrix, and the complete projective reconstruction may be carried out using the normalized coordinates. The result is a fundamental matrix $\widehat{\mathtt{F}}$ satisfying the condition

$$\hat{\mathbf{x}}_i'^\top \widehat{\mathtt{F}} \hat{\mathbf{x}}_i = 0 \tag{14}$$

from which by substitution using (13) one has

$$(\mathbf{x}_i'^\top \mathtt{T}'^\top)\widehat{\mathtt{F}}(\mathtt{T}\mathbf{x}_i) = 0 = \mathbf{x}_i'^\top \mathtt{F}\mathbf{x}_i \ .$$

From this it follows that $\mathtt{F} = \mathtt{T}'^\top \widehat{\mathtt{F}} \mathtt{T}$ is the fundamental matrix corresponding to the original points.

Similarly, if $\widehat{\mathtt{P}}$ and $\widehat{\mathtt{P}'}$ are camera matrices belonging to a reconstruction from the normalized image coordinates, then

$$\hat{\mathbf{x}}_i = \widehat{\mathtt{P}}\widehat{\mathbf{X}}_i \ ; \ \hat{\mathbf{x}}'_i = \widehat{\mathtt{P}}'\widehat{\mathbf{X}}_i \ .$$

Once more, substituting for $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}'_i$, it follows that

$$\mathbf{x}_i = \mathtt{T}^{-1}\widehat{\mathtt{P}}\widehat{\mathbf{X}}_i \ ; \ \mathbf{x}'_i = \mathtt{T}'^{-1}\widehat{\mathtt{P}}'\widehat{\mathbf{X}}_i$$

which implies that the reconstruction $(\mathtt{P}, \mathtt{P}', \{\mathbf{X}_i\})$ for the original points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ is given by

$$\mathtt{P} = \mathtt{T}^{-1}\widehat{\mathtt{P}} \ ; \ \mathtt{P}' = \mathtt{T}'^{-1}\widehat{\mathtt{P}}' \ ; \ \mathbf{X}_i = \widehat{\mathbf{X}}_i \ .$$

This normalized 8-point algorithm gives markedly superior results to the unnormalized algorithm, which should never be used directly. For more details and analysis, see [11].

4 Three view reconstruction

The 8-point algorithm and other methods involving the fundamental matrix are useful for reconstruction from two views.

If three images of a scene are available, and point correspondences are known across all three views, then such linear methods can be extended to three-image reconstruction, using the *trifocal tensor*. This is an extension of the fundamental matrix to three views.

In this analysis of three-view reconstruction, it is convenient from a notational point of view to denote the three camera matrices as $\mathtt{A}$, $\mathtt{B}$ and $\mathtt{C}$, instead of $\mathtt{P}_1$, $\mathtt{P}_2$ and $\mathtt{P}_3$.

Given a three-way image-point correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i$, the goal is to find camera matrices $\mathtt{A}$, $\mathtt{B}$ and $\mathtt{C}$ and points $\mathbf{X}_i$ such that

$$\mathbf{x}_i = \mathtt{A}\mathbf{X}_i \ ; \ \mathbf{x}'_i = \mathtt{B}\mathbf{X}_i \ ; \ \mathbf{x}''_i = \mathtt{C}\mathbf{X}_i \ . \tag{15}$$

This may be written in a form similar to (4), as follows:

$$\begin{bmatrix} \mathtt{A} & \mathbf{x}_i & & \\ \mathtt{B} & & \mathbf{x}'_i & \\ \mathtt{C} & & & \mathbf{x}''_i \end{bmatrix} \begin{pmatrix} \mathbf{X}_i \\ -\lambda_i \\ -\lambda'_i \\ -\lambda''_i \end{pmatrix} = \mathbf{0} \ . \tag{16}$$

In this case, the 9×7 matrix on the left is not square. Nevertheless, since there is a solution $(\mathbf{X}_i, -\lambda_i, -\lambda'_i, -\lambda''_i)^\top$, the matrix must be rank-deficient. Consequently, any 7×7 submatrix must have vanishing determinant. Each such determinant implies a trilinear relationship between the coefficients of the matching points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i$.

It is not necessary to consider all possible 7×7 submatrices to obtain useful relationships. Given three camera matrices $\mathtt{A}$, $\mathtt{B}$ and $\mathtt{C}$ one can define a triply-indexed entity $\mathcal{T}_i^{qr}$

$$\mathcal{T}_i^{qr} = (-1)^{i+1} \det \begin{bmatrix} \mathtt{A}^{(\sim i)} \\ \mathtt{B}^{(q)} \\ \mathtt{C}^{(r)} \end{bmatrix} . \tag{17}$$

Here, all indices range from 1 to 3. Further, $\mathtt{B}^{(q)}$ and $\mathtt{C}^{(r)}$ represent rows q and r of the matrices $\mathtt{A}$ and $\mathtt{B}$, whereas $\mathtt{A}^{(\sim i)}$ means the matrix $\mathtt{A}$ with row i omitted. This results in a 4×4 matrix, whose determinant with the indicated sign is the chosen value $\mathcal{T}_i^{qr}$. This triply-indexed set of 27 values is known as the *trifocal tensor* corresponding to the three cameras. Note that this tensor depends only on the camera matrices, and not any image points.

Now, it may be shown [16, 12] that the coordinates of any matching triple $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i$ satisfy trilinear relations

$$\sum_{i,j,k,q,r=1}^{3} x^i x'^j x''^k \epsilon_{jqu} \epsilon_{krv} \mathcal{T}_i^{qr} = 0_{uv}. \tag{18}$$

This relation may is to be interpreted as follows:

1. The indices on the point coordinates, such as x^i, denote the i-th coordinate of the homogeneous vector representing the point $\mathbf{x} = (x^1, x^2, x^3)^\top$.

2. The symbol ϵ_{jqu} (and similarly ϵ_{krv}) has value 0 unless j, q and u are all distinct; otherwise it is $+1$ if jqu is an even permutation of the three indices 1,2 and 3, and -1 if it is an odd permutation.

3. The indices u and v are free indices, and each choice of u and v leads to a different trilinear relation, for a total of 9 distinct relations. However, only 4 of these relations are linearly independent.

In the case where the first camera matrix $\mathtt{A}$ has the canonical form $[\mathtt{I} \mid \mathbf{0}]$, the expression (17) for the trifocal tensor may be written simply as

$$\mathcal{T}_i^{qr} = b_i^q c_4^r - b_4^q c_i^r, \tag{19}$$

where b_i^q is the element in row q, column i of $\mathtt{B}$ and c_i^r is defined analogously.

A complete three-view reconstruction algorithm can then be outlined as follows:

1. From point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i$ for $i = 1, \ldots, n$ each relation of the form (18) gives 4 linearly independent linear constraints on the entries of the trifocal tensor. From 7 point correspondences there are sufficiently many equations to compute $\mathcal{T}_i^{qr}$ linearly.

2. As with with two-view reconstruction, it is possible to determine the form of the two other camera matrix $\mathtt{B}$ and $\mathtt{C}$ from the entries of the trifocal tensor using the formula (19).

3. Finally, by triangulation from three views based on the equation (16), one can find the world points $\mathbf{X}_i$, completing the reconstruction from three views.

A few more comments.

1. In the definition (18), the first camera matrix $\mathtt{A}$ is treated differently from the two others (in that two rows of $\mathtt{A}$ appear in the determinant, but only one from $\mathtt{B}$ and $\mathtt{C}$). There are two other similarly defined trifocal tensors in which matrices $\mathtt{B}$ or $\mathtt{C}$ are distinguished in this way.

2. Unlike with the fundamental matrix, there are relations similar to (18) that hold for line correspondences, or mixed line and point correspondences. Thus, computation of the trifocal tensor, and hence projective reconstruction is possible not only from point correspondences, but from mixed correspondences of this type.

Minimal configurations. The reconstruction algorithms from two or three views described in section 3 and section 4 require 8 or 7 points respectively. However, it is possible to carry out reconstruction using only 7 points from 2 views, or as few as 6 points from three views.

From two views, the algorithm is easily explained. Given only 7 points correspondences, the set of equations $\mathbf{x}_i'^\top \mathtt{F} \mathbf{x}_i = 0$ represents a set of 7 homogeneous equations in the 9 entries of $\mathtt{F}$. The solution to this equation set is a two-parameter family $\mathtt{F} = \lambda \mathtt{F}_1 + \mu \mathtt{F}_2$ where $\mathtt{F}_1$ and $\mathtt{F}_2$ are determined by solving this system.

The condition that the fundamental matrix $\mathtt{F}$ must be a singular matrix gives a further equation $\det \mathtt{F} = 0$. Since $\mathtt{F}$ is a 3×3 matrix, this leads to a cubic homogeneous equation in λ and μ. Solving this cubic equation gives either one or three real solutions for the ratio $\lambda : \mu$, and hence one or three solutions (determined as ever up to scale) for the fundamental matrix $\mathtt{F}$. In short, from 7 point correspondences one or three possible fundamental matrices may be computed. From these possible values of $\mathtt{F}$ the rest of the method described previously will lead to a projective reconstruction, in fact either a unique or three possible reconstructions.

A method for computing the projective reconstruction from three views of 6 points is described in [27].

5 Factorization algorithms

The algorithms described previously for projective reconstruction work on two or three images. In many cases, one has many more images of a scene to use for reconstruction. To handle this situation, a variant of the Tomasi-Kanade factorization algorithm [30] may be used to do reconstruction from many views at once. This is the algorithm of Sturm and Triggs [29] for projective reconstruction.

As input, consider a set of image points $\mathbf{x}_{ij}$ for $i = 1, \ldots, m$ and $j = 1, \ldots n$, where $\mathbf{x}_{ij}$ represents the image of the j-th point in the i-th image. It is assumed (and required) that every point should be visible in every image, so $\mathbf{x}_{ij}$ is defined for all (i, j).

The projection equations are of the form

$$\lambda_{ij}\, \mathbf{x}_{ij} = \mathtt{P}_i \mathbf{X}_j \ , \tag{20}$$

where the constants λ_{ij} are required scale factors, the so-called *projective depths* of the points. This set of equations may be put together in one matrix equation as follows.

$$\begin{bmatrix} \lambda_{11}\mathbf{x}_{11} & \lambda_{12}\mathbf{x}_{12} & \ldots & \lambda_{1n}\mathbf{x}_{1n} \\ \lambda_{21}\mathbf{x}_{21} & \lambda_{22}\mathbf{x}_{22} & \ldots & \lambda_{2n}\mathbf{x}_{2n} \\ \vdots & & \ddots & \vdots \\ \lambda_{m1}\mathbf{x}_{m1} & \lambda_{m2}\mathbf{x}_{m2} & \ldots & \lambda_{mn}\mathbf{x}_{mn} \end{bmatrix} \tag{21}$$

$$= \begin{bmatrix} \mathtt{P}_1 \\ \mathtt{P}_2 \\ \vdots \\ \mathtt{P}_m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \ldots & \mathbf{X}_n \end{bmatrix} . \tag{22}$$

In this equation the matrix on the left has dimension $3m \times n$, since each $\lambda_{ij}\mathbf{x}_{ij}$ is a 3-vector. This set of equations has the form

$$\Lambda \odot \mathtt{W} = \mathtt{PX} \tag{23}$$

where

$$\Lambda = \begin{bmatrix} \lambda_{11} & \ldots & \lambda_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1} & \ldots & \lambda_{mn} \end{bmatrix} \ ; \ \mathtt{W} = \begin{bmatrix} \mathbf{x}_{11} & \ldots & \mathbf{x}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{x}_{m1} & \ldots & \mathbf{x}_{mn} \end{bmatrix} \tag{24}$$

and $\odot$ is to be interpreted as an elementwise product, so that $\Lambda \odot \mathtt{W}$ is the matrix on the left of (21).

From the form of (21) it is evident that the matrix on the right has rank 4, since it is the product $\mathtt{PX}$ of matrices of dimension $3m \times 4$ and $4 \times n$. This is a low-rank constraint on the matrix $\Lambda \odot \mathtt{W}$ of depth-weighted image coordinates.

Unfortunately, although the matrix $\mathtt{W}$ of image coordinates is known, the matrix Λ of projective-depths is not known. With an incorrect set of projective depths, the matrix $\Lambda \odot \mathtt{W}$ will not have the expected rank 4. This suggests an algorithm in which the factorization and the projective depths are estimated alternately as follows.

1. Form the matrix $\mathtt{W}$ of homogeneous image coordinates as given in (24), and define Λ^0 in which all $\lambda_{ij}^0 = 1$. Then carry out the following steps iteratively for $k = 0, \ldots, N$
   (a) Find the closest rank-4 matrix $\mathcal{W}^k$ to $\Lambda^k \odot \mathtt{W}$.
   (b) Define Λ^{k+1} to be the matrix of weights λ_{ij}^{k+1} so that $\Lambda^{k+1} \odot \mathtt{W}$ is as close as possible to $\mathcal{W}^k$ under Frobenius norm.
2. Compute a final factorization $\mathcal{W}^N = \mathtt{PX}$, to obtain $\mathtt{P}$ and $\mathtt{X}$ providing the camera matrices and point locations respectively.

In step 1(a), the low-rank factorization is carried out by Singular Value Decomposition. Suppose $\Lambda^k \odot \mathtt{W} = \mathtt{UDV}^\top$. Let $\widehat{\mathtt{D}}$ be the matrix obtained by setting all but the four first (largest) diagonal entries of $\mathtt{D}$ to zero. Then set $\mathcal{W}^k = \mathtt{U}\widehat{\mathtt{D}}\mathtt{V}^\top$. The number of iterations N is vaguely defined in this algorithm. The intention is to continue until "convergence" but as will be remarked below, continuing to convergence is problematic.

Variants of the method. It has been observed [24] that the bare projective algorithm as given above will converge to a *trivial* limit in which all the values of λ_{ij} will be zero, except for those values in 4 columns of Λ. This solution is spurious, since zero-values of the projective depths are not possible for a geometrically valid reconstruction. In addition, convergence is very slow. Therefore, different variants on the algorithm have been proposed, as follows.

1. In the original paper of Sturm and Triggs [29] an initialization of the projective depths is proposed, in which projective depths are derived from two-view reconstructions.
2. A viable strategy is to carry out only a fixed small number of alternation steps, since this significantly improves the solution without encountering a trivial solution.
3. A further step of normalization of the projective depths λ_{ij} may be used [16]. Observe that if $\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j$, for all (i, j), then for any constants c_i and d_j,

$$c_i d_j \lambda_{ij} \mathbf{x}_{ij} = (c_i \mathtt{P}_i)\,(d_j \mathbf{X}_j) \ . \tag{25}$$

   Thus, each λ_{ij} may be replaced by $c_i d_j \lambda_{ij}$ without materially changing the factorization. Thus, one may at will multiply each i-th row of Λ by c_i and the j-th column by a constant d_j. In [16] it is suggested that constants c_i and d_j may be chosen so that first the rows, then the columns of Λ sum to unity. However, no analysis of this normalization procedure is given there.

4. More complex schemes for normalization schemes are given in [24] and [21, 22], for which convergence to a meaningful (local) minimum of some cost function is demonstrated.

5. Methods to accommodate missing data or outliers in projective factorization algorithms have been proposed. Though many algorithms have addressed missing data in matrix factorization (for instance [18, 28, 2, 3, 4], a notable paper addressing projective factorization specifically is [5].

6. L_1-factorization has been recognized as more robust alternative to matrix factorization; an effective method is given in [6].

6 Bundle adjustment

Given measured image points $\mathbf{x}_{ij}$ in several images, the projection equations $\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j$ can not be satisfied exactly if there is any inaccuracy, or noise, in the measurements. Therefore, in finding the projection matrices $\mathtt{P}_i$ and $3D$ points $\mathbf{X}_j$ to satisfy these equations, it is appropriate to find an approximate solution. Typically, this solution will be one that minimizes some appropriate cost function representing a residual error in the solution.

Since errors arise in the measurement of the coordinates of image points, it is appropriate to seek a solution that minimizes the error with respect to the measured image coordinates. This corresponds to choosing a cost function of the form

$$C(\{\mathbf{X}_j\}, \{\mathtt{P}_i\}) = \sum_{i,j\in\mathcal{N}} d(\mathbf{x}_{ij}, \mathtt{P}_i\mathbf{X}_j)^2 \ . \qquad (26)$$

where $\mathcal{N}$ is a set of pairs (i, j) for which $\mathbf{x}_{ij}$ is measured. Further, $d(\mathbf{x}_{ij}, \mathtt{P}_i\mathbf{X}_j)$ represents the Euclidean distance in the 2-dimensional image plane between the measured point $\mathbf{x}_{ij}$ and the projected point $\mathtt{P}_i\mathbf{X}_j$, This is commonly referred to as the *reprojection error*. The cost is to be minimized over all choices of $\mathtt{P}_i$ and $\mathbf{X}_j$. This is a non-linear function. The choice of the squared distance means that a non-linear least-squares cost function is to be minimized. The motivation for this choice is the observation that the solution to this least-squares problem represents the Maximum Likelihood (ML) solution, under the assumption that each image measurement error conforms to an isotropic Gaussian distribution, each point measurement being independent of the others.

Minimizing the cost function (26) over all choices of the variables $\mathtt{P}_i$ and $\mathbf{X}_j$ is known as *bundle-adjustment*. Since this is a non-linear optimization problem, an iterative algorithm is required. The most common algorithm used to minimize this cost function is the Levenberg-Marquardt algorithm [19, 23, 16, 31]. In order to converge to the globally optimal solution a good initial solution is necessary. Such an initial solution is found by applying any of the algorithms previously described in this article.

Robust cost functions. The cost function (26) is suitable, and represents the ML solution if the measured point coordinates conform to a Gaussian distribution, and may be used if there are no gross errors (outliers) among the measured points. In most cases, this is unlikely, and a more robust cost function is to be preferred. In this case, the squared Euclidean distance function $d(\cdot,\cdot)^2$ is replaced by some other function $f(\cdot,\cdot)$ that is more tolerant of outliers, meaning that $f(\mathbf{x}, \mathbf{y})$ grows less rapidly than $d(\mathbf{x}, \mathbf{y})^2$ as the distance between the two arguments $\mathbf{x}$ and $\mathbf{y}$ increases. A good choice of robust cost function is the Huber cost function [17, 16]

$$C(\{\mathbf{X}_j\}, \{\mathtt{P}_i\}) = \sum_{i,j\in\mathcal{N}} H\left(d(\mathbf{x}_{ij}, \mathtt{P}_j\mathbf{X}_i)\right)^2 \ . \qquad (27)$$

where $H(x)^2$ is quadratic for $|x| < \delta$ and linear for $|x| \geq \delta$, and δ is some threshold approximately equal to the standard deviation of the measurements.

Sparse methods. A reasonable sized reconstruction problem may involve 1000 camera matrices $\mathtt{P}_i$ and $100,000$ points $\mathbf{X}_j$. Consequently, the cost function (26) depends on a large number of variables ($311,000$ parameters if the cameras are parametrized by 11 parameters). Since the central step in the Levenberg-Marquardt optimization process involves the solution of equations to compute the update of the parameters, this would involve solving a very large set of equations in all the variables. For a dense set of equations in $300,000$ parameters, this would be almost impossible.

Fortunately, the set of equations involved in this update process is quite sparse, so the problem is tractable. To see this, note that if a single point $\mathbf{X}_j$ is moved, then only the image points $\mathbf{x}_{ij}$ involving this point are affected. Similarly, if some camera matrix $\mathtt{P}_i$ is altered, then only image points $\mathbf{x}_{ij}$ are changed. This means that each image measurement depends only on the parameters of one $3D$ point and one camera. This sparse dependence structure for the cost function results in a special sort of sparse structure for the Jacobian matrix. Sparse solution methods may then be used to accelerate the update step, and allow it to be run in reasonable time. Methods that are used for this numerical problem include

the Schurr complement method [16], in which the sparseness of the Jacobian is used to allow the camera updates to be computed first, followed by the point updates. The exact form of the equations is given in [16]. Alternatively, conjugate gradient methods [1] may be used; in such methods the sparseness of the equation set lends itself naturally to sparse methods.

7 Euclidean update

A projective reconstruction may be used as an initial step towards a geometrically correct (Euclidean) reconstruction. There are various ways in which this can be done:

1. By determining or knowing the calibration of the cameras. The camera calibration may be known a-priori, or determined through the process of auto-calibration [9]. Constraints on the camera parameters, such as known focal length, or an assumption that some cameras have the same shared internal parameters, may be enforced easily during bundle-adjustment. Automatic methods for auto-calibration often compute an affine reconstruction first, followed by an update to a Euclidean reconstruction and full determination of the camera calibration parameters [10, 8, 26]. This process is known as stratification.

2. By the knowledge of the $3D$ Euclidean coordinates of some number of *ground-control points*; at least 5 such points are required [13].

3. If partial camera calibration is known, the full calibration and Euclidean reconstruction may be computed more simply than if no calibration information is given. A notable paper demonstrating this is [25] and more details on self-calibration given different types of partial camera calibration are given in [16].

Figure 2 illustrates the steps from projective to Euclidean reconstruction via stratification.

A large scale reconstruction, computed from thousands of images is shown in fig 3.

References

[1] Sameer Agarwal, Noah Snavely, Steven M. Seitz, and Richard Szeliski. Bundle adjustment in the large. In *Proceedings of the 11th European conference on Computer vision: Part II*, ECCV'10, pages 29–42. Springer-Verlag, 2010.

[2] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *Proc. 7th European Conference on Computer Vision, Part I, LNCS 2350, Copenhagen, Denmark*, pages 707–720. Springer-Verlag, 2002.

[3] S. Brandt. Closed-form solutions for affine reconstruction under missing data. In *Proc. 7th European Conference on Computer Vision, Part I, LNCS 2350, Copenhagen, Denmark*, pages 109–114. Springer-Verlag, 2002.

[4] A.M. Buchanan and Fitzgibbon A. W. Damped Newton algorithms for matrix factorization with missing data. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 316–322, 2005.

[5] Yuchao Dai, Hongdong Li, and Mingyi He. Element-wise factorization for N-view projective reconstruction. In *Proc. European Conference on Computer Vision*, 2010.

[6] Anders Eriksson and Anton van den Hengel. Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l1 norm. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 771–778, 2010.

[7] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Proc. 2nd European Conference on Computer Vision, Santa Margharita Ligure, Italy*, pages 563–578. Springer-Verlag, 1992.

[8] O. D. Faugeras. Stratification of three-dimensional vision: projective, affine, and metric representation. *Journal of the Optical Society of America*, A12:465–484, 1995.

[9] O. D. Faugeras, Q. Luong, and S. Maybank. Camera self-calibration: Theory and experiments. In *Proc. 2nd European Conference on Computer Vision, Santa Margharita Ligure, Italy*, pages 321–334. Springer-Verlag, 1992.

[10] R.I. Hartley. Euclidean reconstruction from uncalibrated views. In J. Mundy, A. Zisserman, and D. Forsyth, editors, *Applications of Invariance in Computer Vision*, LNCS 825, pages 237–256. Springer-Verlag, 1994.

[11] R.I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580 – 593, October 1997.

[12] R.I. Hartley. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, 22(2):125–140, March 1997.

[13] R.I. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 761 – 764, 1992.

[14] R.I. Hartley and P. Sturm. Triangulation. In *ARPA Image Understanding Workshop*, pages 957–966, 1994.

[15] R.I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, November 1997.

[16] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision – 2nd Edition*. Cambridge University Press, 2004.

[17] P. J. Huber. *Robust Statistics*. John Wiley and Sons, 1981.

[18] D. W. Jacobs. Linear fitting with missing data for structure-from-motion. *Computer Vision and Image Understanding*, 82:57–81, 2001.

[19] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.

[20] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, September 1981.

[21] S. Mahamud and M. Hebert. Iterative projective reconstruction from multiple views. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages II–430–437, 2000.

[22] S. Mahamud, M. Hebert, Y. Omori, and J. Ponce. Provably-convergent iterative methods for projective structure from motion. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1018–1025, 2001.

[23] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Indust. Appl. Math.*, 11:431–441, 1963.

[24] J. Oliensis and R. Hartley. Iterative extensions of the Sturm/Triggs algorithm: convergence and nonconvergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2217 – 2233, December 2007.

[25] M. Pollefeys, R. Koch, and L. Van Gool. Self calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proc. 6th International Conference on Computer Vision, Bombay, India*, pages 90–96, 1998.

[26] M. Pollefeys and L. Van Gool. Stratified self-calibration with the modulus constraint. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):707 – 724, August 1999.

[27] Long Quan. Invariants of six points and projective reconstruction from three uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:34–46, 1995.

[28] H. Y. Shum, K. Ikeuchi, and R. Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(9):854–867, 1995.

[29] P. Sturm and W. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Proc. 4th European Conference on Computer Vision, Cambridge*, pages 709–720, 1996.

[30] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *International Journal of Computer Vision*, 9(2):137–154, November 1992.

[31] W. Triggs, P.F. McLauchlan, R.I. Hartley, and A. Fitzgibbon. Bundle adjustment for structure from motion. In *Vision Algorithms: Theory and Practice*, pages 298–372. Springer-Verlag, 2000.

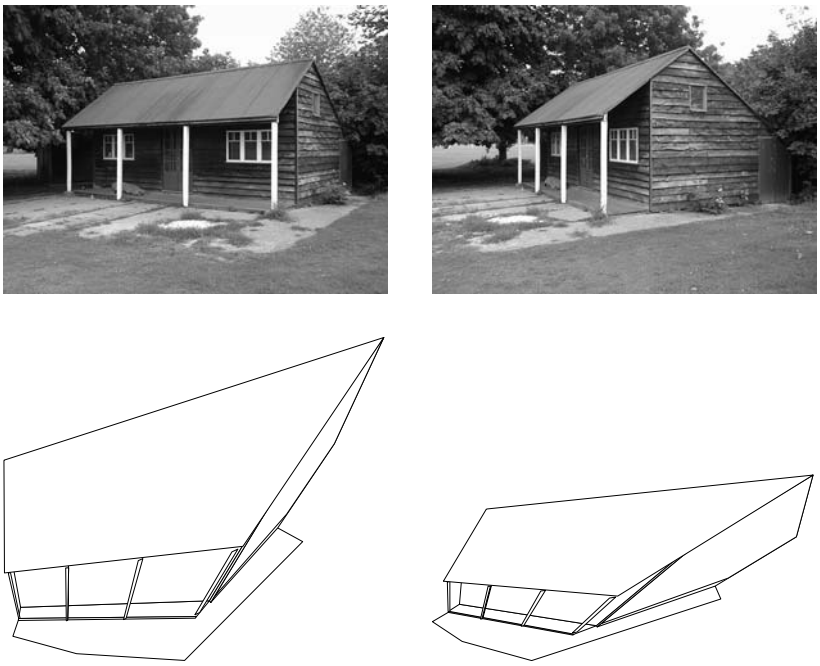

Figure 1: **Projective reconstruction.** *(Top) Original image pair. (Bottom) Two views of a 3D projective reconstruction of the scene. The lines of the wireframe link the computed 3D points. The reconstruction requires no information about the camera matrices, or information about the scene geometry. In a projective reconstruction, the resulting model is distorted by an arbitrary projective transformation from the true geometrically correct model. (Figures derived from [16].)*

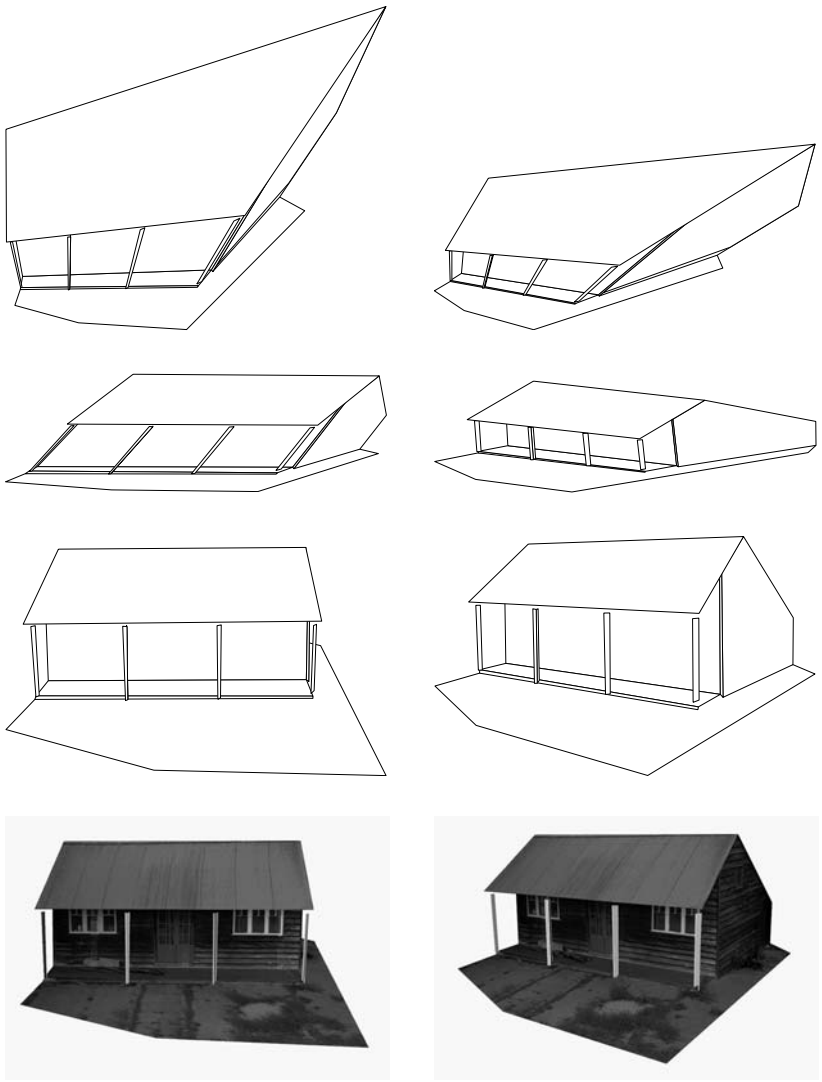

Figure 2: **Stratification.** *The projective reconstruction (top row) obtained by uncalibrated reconstruction techniques is first upgraded to an affine reconstruction (second row). In the affine reconstruction, parallel lines in the image are parallel in the reconstruction, but geometric structures are still skewed. In the final stage of the reconstruction, the true Euclidean model (third row) is computed, in which angles and dimensions are correct up to an indeterminate scale. The fourth row shows two views of the texture-mapped model. (Figures derived from [16].)*

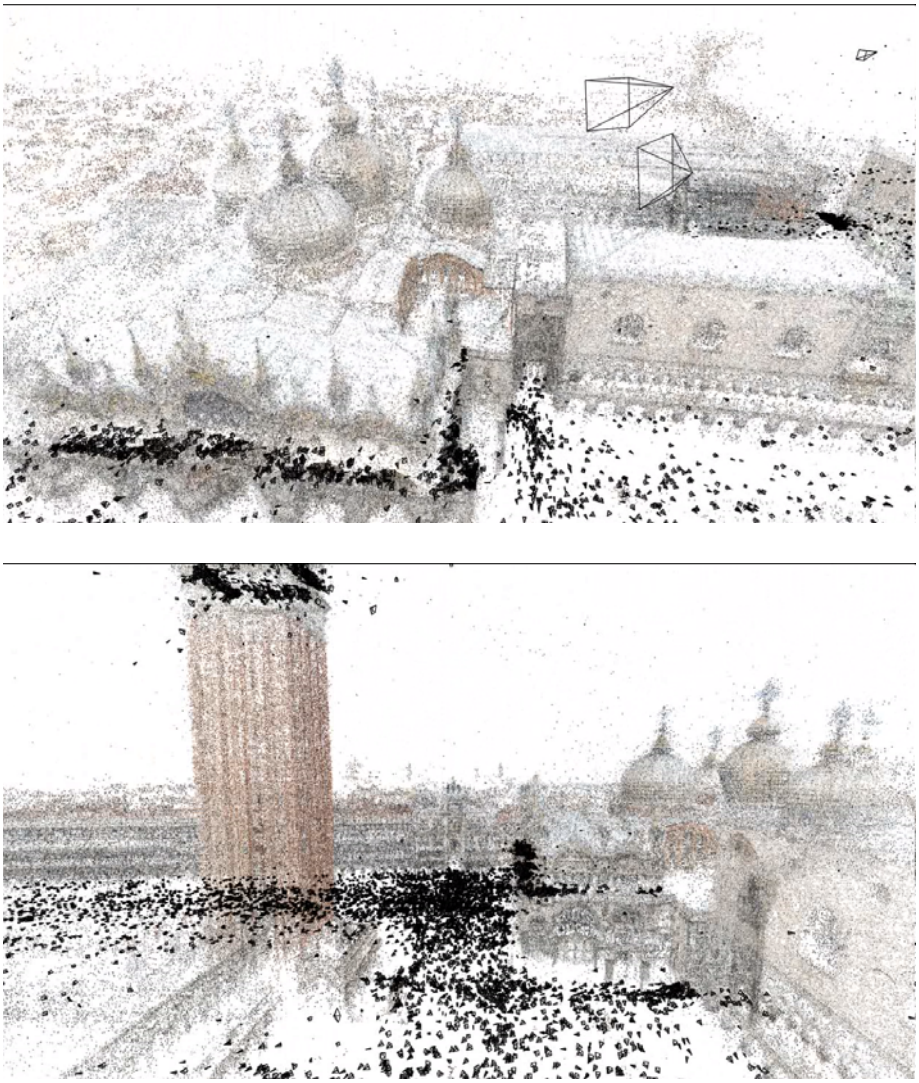

Figure 3: *Views of reconstruction of San Marco, Venice from Flickr images. The top image shows San Marco Cathedral and the doge's palace. Below is shown the campanile at left, and the palace on the right. Black pyramids show the position and orientation of the cameras. Figures are reproduced with thanks to Noah Snavely.*