An Iterative Factorization Method for Projective Structure and Motion from Image Sequences.

Anders Heyden, Rikard Berthilsson, Gunnar Sparr
Dept of Mathematics, Lund University
Box 118, S-221 00 Lund, Sweden

Abstract

In this paper a novel recursive method for estimating structure and motion from image sequences is presented. The novelty lies in the fact that the output of the algorithm is independent of the chosen coordinate systems in the images as well as the ordering of the points. It relies on subspace and factorization methods and is derived from both ordinary coordinate representations and camera matrices and from a so called depth and shape analysis. Furthermore, no initial phase is needed to start the algorithm. It starts directly with the first two images and incorporates new images as soon as new corresponding points are obtained. The performance of the algorithm is shown on both simulated and real data.

Moreover, the two different approaches, one using camera matrices and the other using the concepts of affine shape and depth, are unified into a general theory of structure and motion from image sequences.

Key words: Projective reconstruction, recursive structure from motion, factorization methods, coordinate invariancy

1 Introduction

The problem of reconstruction of an unknown scene from a number of its projective images, obtained from uncalibrated cameras, has been treated by many researchers during the last years. This problem has sometimes been referred to as the *structure from motion problem* (SFM). However, since also the ego-motion of the camera is obtained, *structure and motion from images sequences* would be a better description. The first result obtained was that it is only possible to reconstruct the scene up to an unknown projective transformation, see for example [11] and [1].

Some present reconstruction algorithms rely on particular choices of coordinates in the images and the (unknown) object, e.g. projective or affine coordinates, where some points are sorted out in order to build up a basis, see [1], [8], [9] and [3]. The drawback of selecting some points to build up a basis is, firstly, that all points are not treated uniformly and, secondly, that the measurement errors may propagate uncontrollably.

Other algorithms rely on the so called *multilinear constraints*. These are obtained from 2, 3 or 4 images and are called *bilinear*, *trilinear* and *quadrilinear constraints* respectively. They express the fact that the image coordinates have to fulfil a constraint that is linear in the coordinates of each image separately. For 2 images the bilinear constraint is expressed by the fundamental matrix and is sometimes called the *epipolar constraint*. For 3 and 4 images the trilinear and quadrilinear constraints are expressed by the trifocal and quadrifocal tensor respectively. Then there are no multilinear constraints and tensors involving more than four images. The drawback of the reconstruction algorithms relying on these constraints is that they can only deal with 2, 3 or 4 images and there is no generic way to extend

these algorithms to more images. These types of algorithms can be found in [2], [4] and [6] among others.

Recently, some work has been done on finding more generic algorithms, that can deal with any number of images and points in a unified manner. One such attempt has been made in [15], where an extension of the well-known Tomasi-Kanade factorization algorithm to the projective case has been proposed. A drawback of this algorithm is that it is not independent of the chosen coordinate systems in the images and that the relative depths of the points are needed in order to carry out the factorization. To obtain these relative depths, the epipolar constraints have been used and thus the same problems as described above appear.

There are also several attempts towards recursive algorithms, i.e. algorithms where more and more images are used as they become available. One such algorithm has been presented in [10], using tools from automated control. Another has been presented in [7], using a statistical framework. The drawback of these algorithms, when used in the projective case, is that the result is dependent on the chosen coordinate systems in the images and in the last case, also on the used initial values obtained from three different images.

Another attempt towards more generic algorithms has been made in [14] and [5]. There the reconstruction problem is solved by minimizing a variational formula that is independent of the chosen coordinate system in the images and the ordering of the points. In this paper we will extend these algorithms to the recursive case, where more and more images are incorporated as new corresponding points become available.

The algorithm derived in [14] relies on the concepts of affine shape and depth, developed in a series of papers, see for example [11], [12] and [13]. On the other hand the algorithm used in [5] relies on the standard formulation with camera matrices. However, the final algorithms shows a lot of similarities and one of the goals of this paper will be to investigate and clarify these similarities.

This paper is organized as follows. In Section 2 we give a brief formulation of the projective structure from motion problem, first using coordinates and later subspaces. In Section 3 we present two dual methods to solve this problem and in Section 4 these methods are extended to the recursive case. These methods give reconstructions that are independent of the coordinate systems in the images as well as the ordering of the points. Moreover, the similarities between these methods are exploited and they are shown to be dual to each other. In Section 5 three different experiments are given, two using simulated data and one using real image data. The first simulated experiment is with randomly chosen camera positions and the second is with slowly varying camera movement. It is shown that the resulting back-projected errors in the images are only 50% larger than the standard deviation of the added noise. The experiment on real image data is done on a sequence of images of a toy-block scene. Finally, in Section 6, some conclusions are given.

2 Problem Formulation

The image formation system (the camera) is modeled by the equation

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \gamma f & sf & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix} [R|-Rt] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \Leftrightarrow \tag{1}$$

$$\lambda \mathbf{u} = K[R|-Rt]\mathbf{x} \ .$$

Here $\mathbf{x} = [xyz1]^T$ denotes object coordinates in extended form and $\mathbf{u} = [uv1]^T$ denotes extended image coordinates. The scale factor λ, called the **depth**, accounts for perspective effects and (R,t) represents a rigid transformation of the object, i.e. R denotes a 3×3 rotation matrix and t a 3×1 translation vector.

Finally, the parameters in K represent intrinsic properties of the image formation system: f represents the focal length, γ the aspect ratio and s the skew, modeling the geometry of the light sensitive elements and (x_0, y_0) is called the principal point, interpreted as the orthogonal projection of the focal point onto the image plane. The parameters in R and t are called **extrinsic parameters** and the parameters in K are called the **intrinsic parameters**. Observe that there are 6 extrinsic and 5 intrinsic parameters, in total 11, the same number as in an arbitrary 3×4 matrix defined up to a scale factor. If the extrinsic as well as the intrinsic parameters are unknown, (1) can compactly be written

$$\lambda \mathbf{u} = P\mathbf{x} \quad . \tag{2}$$

In the following we will assume that we have n points (with known correspondences) in m different images and that the intrinsic, as well as the extrinsic parameters, are allowed to vary between the different imaging instants. Let $\mathbf{u}_{i,j} = [u_{i,j}\, v_{i,j}\, 1]^T$ denote the extended coordinates of point number j in image number i and let $\mathbf{x}_j = [x_j\, y_j\, z_j\, 1]^T$ denote the extended coordinates of point number j in the object. Then (2) takes the form

$$\lambda_{i,j}\mathbf{u}_{i,j} = P_i\mathbf{x}_j \quad , \tag{3}$$

where P_i denotes the i:th camera matrix and $\lambda_{i,j}$ the depth of point j in image i.

Introduce the following notation for the extended coordinates of the points in image i

$$\mathbf{U}_i = \begin{bmatrix} u_{i,1} & u_{i,2} & u_{i,3} & \dots & u_{i,n} \\ v_{i,1} & v_{i,2} & v_{i,3} & \dots & v_{i,n} \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix}, \quad i = 1, \dots, m \quad . \tag{4}$$

The extended object coordinates will be described by

$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_n \\ y_1 & y_2 & y_3 & \dots & y_n \\ z_1 & z_2 & z_3 & \dots & z_n \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \quad . \tag{5}$$

Finally, it is convenient to describe the depth, $\lambda_{i,j}$ in (1), of point j in image i by the diagonal matrices

$$\Lambda_i = \begin{bmatrix} \lambda_{i,1} & 0 & \dots & 0 \\ 0 & \lambda_{i,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_{i,n} \end{bmatrix}, \quad i = 1, \dots, m \quad . \tag{6}$$

Now, (3) can be written as follows,

$$\mathbf{U}_i\Lambda_i = P_i\mathbf{X}, \quad i = 1, \dots, m \quad . \tag{7}$$

Denote the linear subspace in $\mathbb{R}^n$ spanned by the rows in $\mathbf{U}_i$ by $\mathcal{D}_i$, i.e.

$$\mathcal{D}_i = \text{linhull}\{\bar{u_i}, \bar{v_i}, \bar{1}\} := \{\mu_1\bar{x_i} + \mu_2\bar{y_i} + \mu_3\bar{1} \,|\, \mu_i \in \mathbb{R}\} \quad . \tag{8}$$

Here linhull denotes the linear hull, $\bar{u_i}$ and $\bar{v_i}$ denote the vector consisting of the x- and y-coordinates of the points in image i respectively and $\bar{1}$ denotes an n-vector consisting of 1:s, i.e.

$$\bar{u_i} = (u_{i,1}, u_{i,2}, \dots, u_{i,n}), \tag{9}$$

$$\bar{v_i} = (v_{i,1}, v_{i,2}, \dots, v_{i,n}), \tag{10}$$

$$\bar{1} = (1, 1, \dots, 1) \quad . \tag{11}$$

Similarly, let $\mathcal{D}$ denote the linear subspace in $\mathbb{R}^n$ spanned by the rows in $\mathbf{X}$, i.e.

$$\mathcal{D} = \text{linhull}\{\bar{x}, \bar{y}, \bar{z}, \bar{1}\} \ , \tag{12}$$

where $\bar{x}$, $\bar{y}$ and $\bar{z}$ denote the vector consisting of the x-, y- and z-coordinates of the object points respectively, i.e.

$$\bar{x} = (x_1, x_2, \ldots, x_n), \tag{13}$$
$$\bar{y} = (y_1, y_2, \ldots, y_n), \tag{14}$$
$$\bar{z} = (z_1, z_2, \ldots, z_n) \ . \tag{15}$$

Let $\Lambda_i \mathcal{D}_i$ denote the linear space

$$\Lambda_i \mathcal{D}_i = \{\lambda_{i,1} v_1 + \ldots \lambda_{i,n} v_n \,|\, (v_1, \ldots, v_n) \in \mathcal{D}_i\} \ , \tag{16}$$

i.e. the diagonal matrix Λ_i is interpreted as an operator that acts on a linear subspace by multiplication of the components by the diagonal elements. The subspace $\Lambda_i \mathcal{D}_i$ can also be written as

$$\Lambda_i \mathcal{D}_i = \text{linhull}\{\Lambda_i \bar{u}_i, \Lambda_i \bar{v}_i, \Lambda_i \bar{1}\} \ . \tag{17}$$

Proposition 2.1. *The camera matrix equations (7) are equivalent to the subspace equations*

$$\Lambda_i \mathcal{D}_i \subset \mathcal{D} \ . \tag{18}$$

Proof. Given $\bar{w} \in \Lambda_i \mathcal{D}_i$ we have from (16) and (8)

$$\bar{w} = t_1 \Lambda_i \bar{u}_i + t_2 \Lambda_i \bar{v}_i + t_3 \Lambda_i \bar{1}, \quad t_i \in \mathbb{R} \ .$$

Let $\mathbf{w}$, $\mathbf{u}_i$, $\mathbf{v}_i$ and $\mathbf{1}$ denote row-vectors corresponding to $\bar{w}$, $\bar{u}_i$, $\bar{v}_i$ and $\bar{1}$ respectively we obtain

$$\begin{aligned} \mathbf{w} =& t_1 \mathbf{u}_i \Lambda_i + t_2 \mathbf{v}_i \Lambda_i + t_3 \mathbf{1} \Lambda_i = \\ =& t_1 P_i^1 \mathbf{X} + t_2 P_i^2 \mathbf{X} + t_3 P_i^3 \mathbf{X} \in \mathcal{D} \ , \end{aligned}$$

where the second equality is obtained from (7) and P_i^j denotes the j:th row of P_i.

On the other hand, assume that (18) is valid. Then $\Lambda_i \bar{u}_i \in \Lambda_i \mathcal{D}_i \subset \mathcal{D}$ can be written as

$$\Lambda_i \bar{u}_i = p_1^1 \bar{x} + p_2^1 \bar{y} + p_3^1 \bar{z} + p_4^1 \bar{1}, \quad p_j^1 \in \mathbb{R} \ .$$

In the same way $\Lambda_i \bar{v}_i$ and $\Lambda_i \bar{1}$ can be written as linear combinations of $\bar{x}$, $\bar{y}$, $\bar{z}$ and $\bar{1}$ with coefficients p_j^2 and p_j^3 respectively. Then (7) is valid with $(P_i)_{j,k} = (p_j^k)$, which concludes the proof. ∎

The subspace equation in (18) can intuitively be understood from (7) interpreted as an equation that projects the 4-dimensional space spanned by the rows of $\mathbf{X}$ to the 3-dimensional space spanned by the rows of $\mathbf{U}_i \Lambda_i$.

One advantage of the formulation (18) is that it is independent of the chosen coordinate systems in the images, since $\mathcal{D}_i$ is the same subspace for every choice of affine coordinates in the images, i.e. $\mathcal{D}_i$ is an affine invariant (even a complete affine invariant). This follows from the fact that an affine change of coordinates in image i can be written as

$$\hat{\mathbf{U}}_i = H_i \mathbf{U}_\mathbf{i} \ , \tag{19}$$

where $\hat{\mathbf{U}}_i$ denotes the new coordinates and H_i a non-singular 3×3 matrix of the form

$$H_i = \begin{bmatrix} a_{1,1} & a_{1,2} & b_1 \\ a_{2,1} & a_{2,2} & b_2 \\ 0 & 0 & 1 \end{bmatrix} \quad , \tag{20}$$

The coordinate change in (19) changes $\bar{u}_i$ and $\bar{v}_i$ to

$$a_{1,1}\bar{u}_i + a_{1,2}\bar{v}_i + b_1\bar{1} \quad \text{and} \quad a_{2,1}\bar{u}_i + a_{2,2}\bar{v}_i + b_2\bar{1} \ ,$$

which obviously spans the same space together with $\bar{1}$ as $\bar{u}_i$ and $\bar{v}_i$ do.

A dual formulation, used in [14] can be obtained by introducing the **affine shape**, $\mathcal{S}_i$, of $\mathbf{U}_i$ as the null-space to the matrix $\mathbf{U}_i$ in (4) and analogously for $\mathbf{X}$, denoted $\mathcal{S}$. Also $\mathcal{S}_i$ is a complete affine invariant of the point configuration described by $\mathbf{U}_i$. Furthermore, $\mathcal{S}_i$ is a perspective image of $\mathcal{S}$, with depths Λ_i, if and only if

$$\Lambda_i \mathcal{S} \subset \mathcal{S}_i \ . \tag{21}$$

The subspaces $\mathcal{S}_i$ and $\mathcal{S}$ are orthogonal complements to the subspaces $\mathcal{D}_i$ and $\mathcal{D}$, i.e.

$$\mathcal{S} = \mathcal{D}^{\perp} = \{ u \in I\!R^n \,|\, v \cdot u = 0, \, \forall v \in \mathcal{D} \} \ .$$

For further details, see [11], [12] and [13]. The equivalence between (21) and (18) follows from the fact that (21) is the orthogonal complement of (18) in $I\!R^n$, making the approach of [14] dual to the one in [5].

The advantage of the introduced notations in (4), (5) and (6) is that (1) can be written, for m images of n points,

$$\begin{bmatrix} \mathbf{u}_1\Lambda_1 \\ \mathbf{u}_2\Lambda_2 \\ \mathbf{u}_3\Lambda_3 \\ \vdots \\ \mathbf{u}_m\Lambda_m \end{bmatrix} = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \vdots \\ P_m \end{bmatrix} \mathbf{X} = \mathbf{P}\mathbf{X} \ . \tag{22}$$

Note that the sum of two subspaces s_1 and s_2 is defined by

$$s_1 + s_2 = \{ x + y \,|\, x \in s_1, \, y \in s_2 \} \ .$$

Using the subspace analogy in (18), we have the following theorem.

Theorem 2.1. *The camera equations in (7) can be written as the single subspace equation*

$$\Lambda_1\mathcal{D}_1 + \Lambda_2\mathcal{D}_2 + \Lambda_3\mathcal{D}_3 + \ldots + \Lambda_m\mathcal{D}_m \subseteq \mathcal{D} \ . \tag{23}$$

Proof. The theorem follows directly from the discussion above. ∎

Note that (23) also follows directly from (18). Observe that since $\dim \Lambda_i\mathcal{D}_i = 3$ and $\dim \mathcal{D} = 4$, either is $\dim(\Lambda_1\mathcal{D}_1 + \ldots + \Lambda_m\mathcal{D}_m) = 3$ or 4. In the first case, all spaces $\Lambda_i\mathcal{D}_i$ coincide, which is equivalent to that all images are projectively equivalent. Disregarding this degenerate situation, we find that equality holds in (23).

Our goal now is to use (23) to design an algorithm for calculating $\mathbf{X}$, Λ_i and P_i from the image data $\mathbf{U}_i$. Then $\mathbf{X}$ gives us the reconstruction (structure) and P_i give us the camera matrices (motion; obtained as the null-spaces of P_i).

The same subspace formula can be obtained in the dual approach, since (21) implies that

$$\mathcal{S} \subseteq \bar{\Lambda}_1 \mathcal{S}_1 \cap \bar{\Lambda}_2 \mathcal{S}_2 \cap \bar{\Lambda}_2 \mathcal{S}_3 \cap \ldots \cap \bar{\Lambda}_m \mathcal{S}_m \ , \tag{24}$$

where $\bar{\Lambda}_i = \Lambda_i^{-1}$, which again is dual to (23). By the same argument as above, equality holds in non-degenerate situations.

Observe the well-known fact that it is only possible to reconstruct the object and the ego-motion up to an arbitrary projective transformation, since given P_i and $\mathbf{X}$, that fulfils (22), we can replace them by $P_i A^{-1}$ and $A\mathbf{X}\Lambda$, where A is a nonsingular 4×4 matrix and Λ is a diagonal matrix, which makes the last coordinate equal to one[1], and (22) will still be fulfilled. Furthermore, the depths $\lambda_{i,j}$ can not be determined uniquely. The obtained depths, for a particular reconstruction, are called the **relative depths**, since they can only be determined up to an unknown scale factor even when the reconstruction is fixed.

3 A Proximity Measure for Reconstruction

Before presenting a criterion for solving the structure and motion problem, we will need some fundamental results on projection matrices, representing orthogonal projections.

The orthogonal projection $\mathbf{T}v$ of $v \in \mathbb{R}^n$ onto the subspace $\mathcal{T}$ is defined according to

1. $\mathbf{T}v \in \mathcal{T}$,
2. $v - \mathbf{T}v \in \mathcal{T}^\perp$.

The easiest way to calculate the orthogonal projection is given in the following lemma.

Lemma 3.1. *The orthogonal projection* $\mathbf{T}v$ *of* v *onto the subspace spanned by the columns of* M *is given by*

$$\mathbf{T}v = M(M^T M)^{-1} M^T v \ ,$$

i.e. the projection matrix is $M(M^T M)^{-1} M^T$.

Proof. It follows from property (1) above that $\mathbf{T}v$ can be written as $M\mu$ for some $\mu \in \mathbb{R}^p$, where p denotes the number of columns in M, i.e. $\mathbf{T}v = M\mu$. From property (2) it follows that

$$\begin{aligned} M^T(v - M\mu) = 0 \quad &\Rightarrow \quad M^T v = M^T M\mu \quad \Rightarrow \\ \mu = (M^T M)^{-1} M^T v \quad &\Rightarrow \quad \mathbf{T}v = M\mu = M(M^T M)^{-1} M^T v \ . \end{aligned}$$

■

The matrix $M^+ = M(M^T M)^{-1} M^T$ is known as the Moore-Penrose pseudo-inverse and is also characterized as $x = M^+ y$ being the least squares solution to $y = Mx$.

Note that if the subspace is given by the rows of M (instead of the columns), the projection matrix becomes

$$M^T (MM^T)^{-1} M$$

instead and the projection equation becomes

$$\mathbf{T}(v) = vM^T (MM^T)^{-1} M \ ,$$

for a row-vector v.

[1]The diagonal matrix is only a rescaling of the point coordinates, needed to interpret the elements in the matrix $\mathbf{X}$ as coordinates and not only as a representative of the subspace $\mathcal{D}$.

For convenience, we set $\mathbf{V}_i = \mathbf{U}_i\Lambda_i$. Now, the matrix

$$\mathbf{T}_i = \mathbf{V}_i^T(\mathbf{V}_i\mathbf{V}_i^T)^{-1}\mathbf{V}_i \ , \tag{25}$$

defines the orthogonal projection from $\mathbb{R}^n$ onto $\Lambda_i\mathcal{D}_i$. Introduce the normalized sum of projections

$$\mathbf{T} = \frac{1}{m}\sum_1^m \mathbf{T}_i \ . \tag{26}$$

The purpose of the factor $1/m$ in (26) is to make the size of the entries in $\mathbf{T}$ more independent of the number of images, in the sense that the trace of $\mathbf{T}$ is independent of m. (Since $\mathbf{T}_i$ are orthogonal projection matrices to subspaces of rank 3, $\text{trace}\,\mathbf{T}_i = 3$, and consequently $\text{trace}\,\mathbf{T} = 3$. Note that from (23), we immediately get for any vector $v \in \mathbb{R}^n$, that

$$\mathbf{T}v = \frac{1}{m}\sum_1^m \mathbf{T}_i v \in \mathcal{D} \ . \tag{27}$$

Since $\dim(\mathcal{D}) = 4$ in non-degenerate situations, we consequently obtain that

$$\text{rank}(\mathbf{T}) = 4 \ . \tag{28}$$

Furthermore, the matrix $\mathbf{T}$ becomes independent of the choice of basis in the different $\mathcal{D}_i$ according to the following Lemma.

Lemma 3.2. *The matrix* $\mathbf{T}$ *defined in (26) is independent of the chosen basis for* $\mathcal{D}_i$.

Proof. Let the rows of $\mathbf{V}_i$ be an arbitrary basis for $\Lambda_i\mathcal{D}_i$. Then any other representation of $\Lambda_i\mathcal{D}_i$ as the row space of a matrix is obtained by multiplying $\mathbf{V}_i$ by an arbitrary non-singular matrix A_i from the left, i.e. as $A_i\mathbf{V}_i$. Consider the definition of $\mathbf{T}_i$ in (25) with $A_i\mathbf{V}_i$ instead of $\mathbf{V}_i$ as a representative of $\Lambda_i\mathcal{D}_i$,

$$\begin{aligned}(A_i\mathbf{V}_i)^T(A_i\mathbf{V}_i(A_i\mathbf{V}_i)^T)^{-1}A_i\mathbf{V}_i &= \mathbf{V}_i^TA_i^T(A_i\mathbf{V}_i\mathbf{V}_i^TA_i^T)^{-1}A_i\mathbf{V}_i = \\ &= \mathbf{V}_i^TA_i^TA_i^{-T}(\mathbf{V}_i\mathbf{V}_i^T)^{-1}A_i^{-1}A_i\mathbf{V}_i = \\ &= \mathbf{V}_i^T(\mathbf{V}_i\mathbf{V}_i^T)^{-1}\mathbf{V}_i \ .\end{aligned}$$

This shows that $\mathbf{T}$ is independent of the chosen basis. ∎

Corollary 3.1. *The matrix* $\mathbf{T}$ *is independent of affine changes of coordinates in the images.*

Observe that the only undetermined parameters in $\mathbf{T}$ are the relative depths in Λ_i, everything else can be measured from the images.

Looking at (23) the condition (28) can be interpreted as follows. The individual projections, $\mathbf{T}_i$, are the orthogonal projections onto the subspaces $\Lambda_i\mathcal{D}_i$. (23) says that the sum of these three-dimensional subspaces are contained in a four-dimensional subspace, $\mathcal{D}$. Thus the sum of these projection matrices has rank less than or equal to 4.

Definition 3.1. Let σ_i, $i = 1,\ldots,n$ be the singular values of $\mathbf{T}$. The **proximity measure** of $\mathbf{T}$ is defined as

$$\mathcal{P} = \sigma_5 \ . \tag{29}$$

∎

The proximity measure measures the degree of 4-dimensionality of the sum of the subspaces $\mathcal{D}_i$. This choice of proximity measure is somewhat ad hoc. Another choice could be to minimize the ratio σ_5/σ_4 instead. However, for computational purposes the definition above is more convenient.

Now the reconstruction problem can be formulated by a variational formula

$$\min_{\{\lambda_{i,j}\}} \mathcal{P} \quad . \tag{30}$$

In the noise free case the minimum value is equal to 0. One of the advantages with this formulation of the reconstruction problem is

Theorem 3.1. *When noise is present in the measurements, minimizing the variational formula in (30) gives a projective reconstruction that is independent of the chosen coordinate systems.*

Proof. Follows directly from the previous lemma and the construction of the proximity measure. ∎

The actual reconstruction can be obtained from the singular value decomposition of $\mathbf{T}$. Observe first that $\mathbf{T}$ is symmetric, since it is a sum of symmetric matrices, $\mathbf{T}_i$ according to (25) and (26). Let $\mathbf{T} = U^T \Sigma U$, where U is an orthogonal matrix and Σ is a diagonal matrix containing the singular values, σ_i, of $\mathbf{T}$. Let $\tilde{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \sigma_4, 0, \ldots, 0)$ and let $\tilde{U}$ denote the first three columns of U. Then

$$\mathbf{P} = \tilde{U}\tilde{\Sigma} \tag{31}$$

gives the camera matrices and

$$\mathbf{X} = \tilde{U}^T \tag{32}$$

gives a projective reconstruction fulfilling (22). From the reconstruction in (32) a projective basis consisting of 5 points in general position can be chosen. Then projective coordinates with respect to these 5 basis points can easily be calculated by multiplying $\mathbf{X}$ with a diagonal matrix from the right and a non-singular 4×4 matrix from the left (corresponds to a projective transformation) such that the first 5 points have coordinates $(1,0,0,0)$, $(0,1,0,0)$, $(0,0,1,0)$, $(0,0,0,1)$ and $(1,1,1,1)$ respectively.

Using the dual approach we obtain a similar formulation as follows. Denote by

$$\mathbf{S}_i = I - \mathbf{T}_i$$

the orthogonal projection onto the subspace $\bar{\Lambda}_i \mathcal{S}_i$ and observe that $x \in \bar{\Lambda}_i \mathcal{S}_i \Leftrightarrow \mathbf{S}_i x = x$ implies that

$$\bigcap_{i=1}^{m} \bar{\Lambda}_i \mathcal{S}_i = \left\{ x \,|\, \frac{1}{m} \sum_{i=1}^{m} \|\mathbf{S}_i x\|^2 = \|x\|^2 \right\} \quad . \tag{33}$$

By introducing

$$\mathbf{M} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{S}_i \quad , \tag{34}$$

it follows that

$$\bigcap_{i=1}^{m} \bar{\Lambda}_i \mathcal{S}_i = \left\{ x \,|\, x^T \mathbf{M} x = \|x\|^2 \right\} \quad .$$

According to (24), it follows that

$$\dim \bigcap_{i=1}^{m} \bar{\Lambda}_i \mathcal{S}_i = n - 4 \quad .$$

From properties of quadratic forms it follows that this is the case if and only if $\mathbf{M}$ has a singular value 1 of multiplicity $n-4$. Observing that $\mathbf{M} = I - \mathbf{T}$, this is in perfect agreement with the properties of $\mathbf{T}$. Let τ_i denote the singular values of M, as proximity measure we may chose

$$\pi = 1 - \tau_{n-4} = \sigma_4 \tag{35}$$

and the same variational formula as above

$$\min_{\lambda_{i,j}} \pi \ .$$

Then the reconstruction can be obtained in a similar way as described above by using the singular value decomposition of $\mathbf{M}$. In detail, let $\mathbf{M} = V^T D V$, then the range of first $n-4$ rows of V can serve as an estimate of $\mathcal{S}$.

Observe that the singular values of $\mathbf{M}$ and $\mathbf{T}$ are related by $\tau_i = 1 - \sigma_j$ for some i, j. Observe, furthermore, that $\mathbf{M}$ and $\mathbf{T}$ are not, in general, projection matrices, but they are symmetric, positive definite with norm ≤ 1.

4 A Recursive Algorithm

One problem with using (30) is that the solution, Λ_i, $i = 1, \ldots, m$, is not unique, since the scale factors Λ_i in (1) for the first image can not be determined uniquely. In fact, they can be chosen arbitrarily in the linear space $\mathcal{D}$, also called the **depth space** of X, see [14]. This can be seen from (7), since the last row in $P_i\mathbf{X}$ is a linear combination of the rows in $\mathbf{X}$. One way to circumvent this problem is to fix all scale factors in the first image to 1, by introducing **the kinetic depths**

$$q_{i,j} = \frac{\lambda_{i,j}}{\lambda_{1,j}} \tag{36}$$

and replace all $\lambda_{i,j}$ in the previous equations with $q_{i,j}$. These kinetic depths are unique modulo constant factors and independent of the individual coordinate representations.

We propose a recursive algorithm consisting of the following steps:

1. Start by putting $q_{i,j} = 1$ for all available images.
2. Calculate $\mathbf{T}$ from (27) (or $\mathbf{M}$ from (34)) using all at this time available images.
3. Calculate the singular value decomposition of $\mathbf{T}$ (or $\mathbf{M}$), i.e. $\mathbf{T} = U^T \Sigma U$ (or $\mathbf{M} = V^T D V$) and the proximity measure $\mathcal{P}$ (or π).
4. Let $\mathbf{X}$ denote the first four rows of U (or let $\mathbf{S}$ denote the first $n-4$ rows of V), which will be used as an approximation of the object, as done in (32).
5. Add one or several new images.
6. Use (18) to estimate Λ_i from $\mathbf{x}_i$ and $\mathbf{X}$ (or use (21) to estimate Λ_i from $\mathbf{S}_i$ and $\mathbf{S}$), for all images.
7. Calculate $q_{i,j}$ from (36) and goto 2.

We have in this algorithm fixed the kinetic depths for the first image to $q_{1,j} = 1$ according to (36) and from that assumption the reconstruction is unique up to an unknown affine transformation. Other reconstructions are obtained by multiplying the depths with a vector in the depth space of $\mathbf{X}$, see [13].

Observe that new images only enter in the algorithm at step 5. This is motivated by the fact that when an approximation to the reconstructed object is available, we get a good approximation of the relative depths for the new image. Then this image can be used to make a better approximation of the

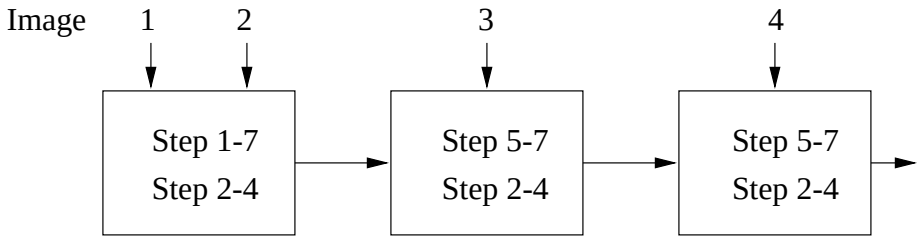

Figure 1: Illustration of the recursive algorithm.

reconstructed object. If the new image is used directly in step 2, we have to guess the relative depths and using them in step 3 and 4 probably do not make the approximation of the reconstructed object better.

The proposed scheme is illustrated in Figure 1, in the case where one new image enters in each loop of the algorithm.

We start with the first two images and carry out step 1 to 7, which gives an approximation to the reconstruction and the relative depths. Then carry out step 2 to 4 again in order to update the reconstruction using the obtained relative depths. In step 5 the third image enters and the relative depths of all images are calculated in 6 and 7. Then the reconstruction is updated again in step 2 to 4, before the fourth image enters in step 5 and so on. When a lot of images has become available it is tedious to calculate relative depths to all of them in every loop. A way to circumvent this is to limit the number of images, where the relative depths are updated to say 10 but use all images in step 2 to 4, since this does not increase the complexity of the algorithm after the 10 first images.

When the initial step has been done the algorithm works by switching between updating the reconstruction and updating the relative depths:

- Update the reconstruction, using the present images and relative depths.
- Update the relative depths of the present images and calculate an estimate of the relative depths of the new image, using the reconstruction.

Step 6 above can be carried out as follows. Note that, when the object, in this case $\mathbf{X}$, is known, then (7) is linear in the unknown parameters Λ_i and P_i. However, in order to maintain our subspace approach, we will use (18) and determine Λ_i such that the subspace $\Lambda_i \mathcal{D}_i$ is contained, as well as possible, in $\mathcal{D}$. Let

$$T_{\mathcal{D}} = \mathbf{X}^T (\mathbf{X}\mathbf{X}^T)^{-1} \mathbf{X}$$

denote the projection matrix onto $\mathcal{D}$ and

$$T_{\mathcal{S}} = I - T_{\mathcal{D}}$$

the projection matrix onto the orthogonal complement of $\mathcal{D}$. Then (7) can be written

$$T_{\mathcal{S}} (\mathbf{U}_i \Lambda_i)^T = 0 \quad , \tag{37}$$

which are $3n$ linear equations in the n unknown relative depths, for each $i = 1, \ldots m$.

Proposition 4.1. *Let $\mathbf{u}_i^j$, $j = 1,2,3$ denote the three rows of $\mathbf{U}_i$, now considered as columns instead. Then step 6 above can be formulated as the optimization problem*

$$\min_{\|\Lambda_i\|=1} \sum_{j=1}^{3} \|T_{\mathcal{S}} \Lambda_i \mathbf{u}_i^j\|^2 \quad . \tag{38}$$

Furthermore, if an orthonormal representation of $\mathbf{U}_i$ is used, i.e. $\mathbf{u}_i^j$ are orthonormal, the result is independent of affine coordinate changes in the images.

Proof. The first statement, that (38) solves step (6) is obvious from the discussion above.

The independence of the solution to the optimization problem in (38) on the chosen basis can be seen as follows. The minimization problem in (38) can be rewritten as

$$\min_{\|\Lambda_i\|=1} \|T_{\mathcal{S}}\Lambda_i\mathbf{U}_i^T\|_F^2 = \min_{\|\Lambda_i\|=1} \operatorname{trace}\left(T_{\mathcal{S}}\Lambda_i\mathbf{U}_i^T(T_{\mathcal{S}}\Lambda_i\mathbf{U}_i^T)^T\right) \ , \tag{39}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\operatorname{trace} A$ denotes the trace of A. Assume that the rows in $\mathbf{U}_i$ are orthonormal. Any other choice of an orthonormal basis for the row span of $\mathbf{U}_i$ can be written as $R\mathbf{U}_i$, where R denotes an orthogonal matrix. It can easily be seen from (39) that $\|Q_{\mathcal{D}}\Lambda_i\mathbf{U}_i^TR^T\|_F$ is independent of R, since

$$\begin{aligned} T_{\mathcal{S}}\Lambda_i\mathbf{U}_i^TR^T(T_{\mathcal{S}}\Lambda_i\mathbf{U}_i^TR^T)^T &= T_{\mathcal{S}}\Lambda_i\mathbf{U}_i^TR^TR\mathbf{U}_i\Lambda_iT_{\mathcal{S}}^T = \\ &= T_{\mathcal{S}}\Lambda_i\mathbf{U}_i^T\mathbf{U}_i\Lambda_iT_{\mathcal{S}}^T = \\ &= T_{\mathcal{S}}\Lambda_i\mathbf{U}_i^T(T_{\mathcal{S}}\Lambda_i\mathbf{U}_i^T)^T \end{aligned}$$

and thus the solution to the optimization problem in (38) is independent of the chosen orthonormal basis. ■

Remark. In order to achieve an orthonormal basis, spanning $\mathcal{D}_i$, $(\mathbf{U}_i\mathbf{U}_i^T)^{-1/2}\mathbf{U}_i$, is used instead of $\mathbf{U}_i$. This can be seen from the fact that the rows of the matrix $(\mathbf{U}_i\mathbf{U}_i^T)^{-1/2}\mathbf{U}_i$ obviously span the same space as the rows of the matrix $\mathbf{U}_i$ and they are orthogonal since

$$(\mathbf{U}_i\mathbf{U}_i^T)^{-1/2}\mathbf{U}_i((\mathbf{U}_i\mathbf{U}_i^T)^{-1/2}\mathbf{U}_i)^T = I \ .$$

This procedure of selecting an orthonormal basis is necessary in order to achieve a result that is independent on the chosen coordinate systems in the images. It is also advantageous for numerical reasons. ■

Theorem 4.1. *The output of the recursive algorithm is independent of the chosen coordinate systems in the images as well as of the ordering of the points.*

Proof. The statement follows from the previous proposition and lemma and the construction of the proximity measure. ■

In the dual approach step 6 can be solved similarly using (21) in the following way. Let $T_{\mathcal{S}_i}$ denote the projection matrix onto the subspace $\mathcal{S}_i$ and $T_{\mathcal{D}_i} = I - T_{\mathcal{S}_i}$ the corresponding projection onto the orthogonal complement. Then (21) can be written

$$T_{\mathcal{D}_i}\Lambda\mathbf{S} = 0 \ . \tag{40}$$

The left hand sides in (37) and (40) are transposes and can be solved in the same way.

It turns out that the use of the kinetic depths, $q_{i,j}$, above instead of the relative depths, $\lambda_{i,j}$, is not necessary for this algorithm to work. Instead it is possible to work directly with the relative depths and thus omit step 7 in the algorithm. Moreover, the convergence properties of the algorithm are better when using relative depths instead of kinetic depths, according to experimental studies. This is the case, despite the fact that there are more free variables than needed, i.e. a lot of Gauge freedom. This fact might be explained as the extra degrees of freedom makes it possible to go a straighter and shorter way to the minima. However, the limit of $\lambda_{i,j}$ is not unique. Using other initial values might result in a different (but projectively equivalent) limit.

The advantage of using the relative depths, $\lambda_{i,j}$, is that we obtain an algorithm that is also independent on the ordering of the images, when the number of used images are constant. However, in the recursive

case this fact is not important. The relative depths have been used in the experiments, for simplicity and for better convergence properties.

Observe that it is not necessary to calculate the whole singular value decomposition of $\mathbf{T}$ in step 3. It is sufficient to calculate the 5 largest singular values and the vectors corresponding to the 4 largest singular values, which is the same as calculating the 5 largest eigenvalues and the eigenvectors corresponding to the 4 largest eigenvalues.

Observe also that there are not at this moment any theoretical proof that the algorithm will converge. However, experimental evidence as well as recently started theoretical studies, indicates that the algorithm has very good convergence properties, both globally and locally. Even the rate of convergence is very good, based on experimental observations. After using this algorithm for some time on a number of different experiments, we have not found any single case where the algorithm does not converge to the correct solution. In all these experiments the initial relative depths have been chosen $\lambda_{i,j} = 1$, corresponding to an affine approximation. This indicates that there is no need to first estimate fundamental matrices and relative depths and then use these as initial values. However, if these relative depths obtained from fundamental matrices where available (as in [15]) the convergence properties would of course be as good as with the other initialization.

The proposed algorithm assumes a sufficiently general motion and structure, in order to make sure that the rank of $\mathbf{T}$ is at least 4 for every choice of relative depths. Otherwise, the motion, or the scene, is degenerate (e.g. planar scene, only zooming or pure rotations). As all reconstruction algorithms the performance degrades in non-general or nearly non-general situations.

5 Experiments

Experiments have been carried out on synthetic data with added noise as well as on a real image sequence. Since the two methods described above are dual to each other and give equivalent algorithms, only the first one has been used in the experiments. The proximity measure, $\mathcal{P}$, are recorded for each iteration as well as the estimated standard deviation of the errors in the images obtained from back projection of the reconstruction to the images, described in more detail below. In both experiments, we have a time window of ten consecutive images, where the relative depths are computed.

5.1 Simulated Data

The first experiment has been carried out as follows. We have tried to make the simulation setup close to a real life situation. The object consists of 15 points, which have been chosen randomly within a box with side equal to 400 units. Then the camera has been positioned at a distance of 2000 units away from the center of the object. The focal lengths have been chosen to 1000 plus a random number with standard deviation 100. The skew, aspect ratio and principal point have also been chosen randomly with mean values and standard deviations $(0, 0.2)$, $(1, 0.2)$ and $(0, 10)$ units respectively. The object has been rotated by a random rotation matrix between the different imaging instants. The obtained pixel values of the points in the images vary between -400 and 400 pixels. Different levels of noise have been added to the images. The results are shown in Table 1, where σ denotes the standard deviation of normal independent noise added to the points in the images and $\hat{\sigma}$ denotes the estimated standard deviation of the errors in the back-projected images. This estimate has been obtained as follows. When the estimated reconstruction, $\hat{\mathbf{X}}$, has been obtained, using step 4 in the algorithm, the depths are estimated using step 6 and finally the camera matrices are estimated in least squares sense from (7), i.e.

$$\hat{P}_i = \mathbf{U}_i \hat{\Lambda}_i \hat{\mathbf{X}}^+ \ ,$$

where $\hat{\mathbf{X}}^+$ denotes the Moore-Penrose pseudo-inverse of $\hat{\mathbf{X}}$ and $\hat{\ }$ denotes estimated quantities. From these camera matrices, P_i, and the reconstruction, $\hat{\mathbf{X}}$, the back-projected image coordinates, $\hat{\mathbf{U}}_i$, can

σ	0	0.1	0.2	0.5	1	2	5
$\hat{\sigma}$	0.01	0.2	0.3	0.9	1.7	3.4	8.6

Table 1: Errors in the images and estimated errors in the reconstructions in the first experiment.

easily be calculated using (7), i.e.

$$\hat{\mathbf{U}}_i\hat{\Lambda}_i = \hat{P}_i\hat{\mathbf{X}}, \quad i = 1,\ldots,m \ .$$

Then the estimated standard deviation in the images are obtained from

$$\hat{\sigma} = \Big(\frac{1}{d}\sum_{i,j=1}^{m,n}(u_{i,j}-\hat{u}_{i,j})^2 + (v_{i,j}-\hat{v}_{i,j})^2\Big)^{1/2} \ , \tag{41}$$

where the denominator, $d = 2mn - 3n - 11m + 15$, is the number of free parameters, coming from $2mn$ measured coordinates in the images, $3n$ coordinates in the reconstruction, $11m$ parameters in the camera matrices and 15 degrees of freedom in a three-dimensional projective transformation.

As Table 1 indicates, the estimated standard deviation $\hat{\sigma}$ is only slightly larger than the actual uncertainty σ in the images. For $\sigma = 0.3$,

Figure 2 illustrate the development of the proximity measure $\mathcal{P}$ ($\log_{10}(\mathcal{P})$ is plotted) and the estimated standard deviation $\hat{\sigma}$, versus the number of images taken into account.

Figure 3 illustrates the RMS (root mean square) Euclidean error in the reconstruction and a projective invariant of the scene versus the number of images. The Euclidean reconstruction error has been obtained by choosing a projective transformation of the reconstruction that fits as good as possible to the known Euclidean object. This has been done by solving the equations

$$M\hat{\mathbf{X}} = \mathbf{X}_E \,\mathrm{diag}(\gamma) \ ,$$

for M and γ in the least squares sense, where again $\hat{\mathbf{X}}$ denotes the estimated projective reconstruction, M denotes a 4×4 matrix corresponding to the projective transformation, $\mathbf{X}_E$ denotes the known Euclidean reconstruction and γ denote unknown scalar factors. The projective invariant has been calculated by firstly making a projective transformation of the estimated reconstruction, such that the first five points have homogeneous coordinates $(1,0,0,0)$, $(0,1,0,0)$, $(0,0,1,0)$, $(0,0,0,1)$ and $(1,1,1,1)$ respectively. Then the ratio x_6/z_6 (in the new coordinates) has been chosen.

Note that the RMS error of the Euclidean reconstruction is in the limit only about 1% of the size of the object, which is somehow unexpected. Note also that the projective invariant converges nicely.

In the second experiment we have a similar setup to the preceding experiment, but with the difference that both the intrinsic and the extrinsic parameters vary more slowly between the imaging instants. The variation is approximately a few pixels between the images. We achieve this by choosing a few numbers randomly and using them as Fourier coefficients for a function. The limits for the allowed variation of the obtained parameters, are about the same as before and also here we add different levels of noise to the images.

It is seen, from Table 2, that the results for this setup is about the same as in the preceding one and, for $\sigma = 0.3$, Figure 4 and Figure 5 also indicate this.

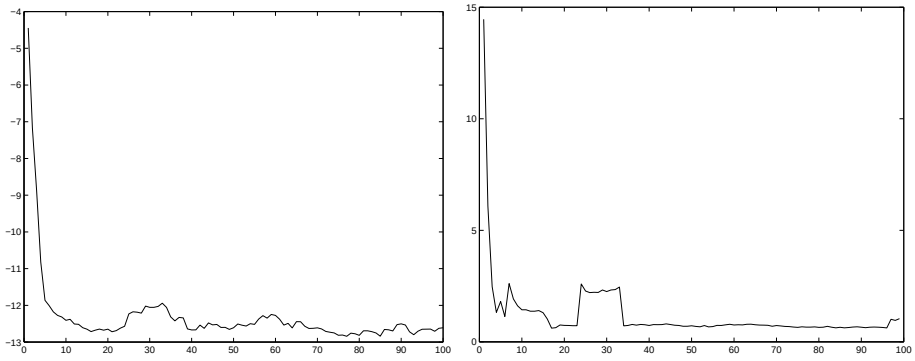

Figure 2: $\log_{10}(\mathcal{P})$ and estimated standard deviation, versus number of images in the first experiment.

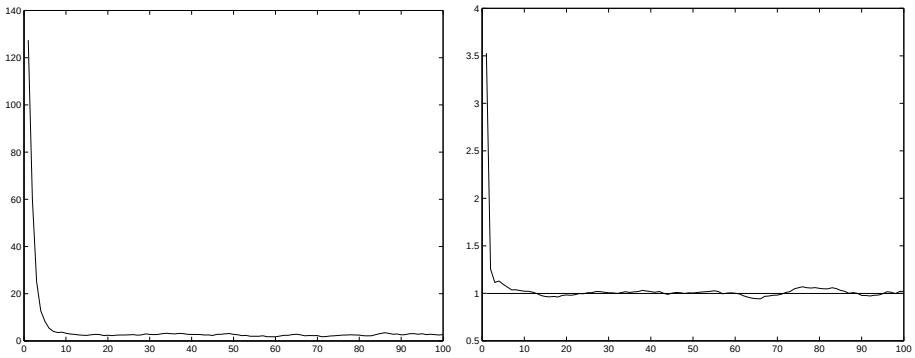

Figure 3: RMS error of reconstruction and the projective invariant, versus number of images in the first experiment. The correct value of the projective invariant is shown as a line.

σ	0	0.1	0.2	0.5	1	2	5
$\hat{\sigma}$	0.03	0.2	0.3	0.8	1.6	3.1	7.1

Table 2: Errors in the images and estimated errors in the reconstructions in the second experiment.

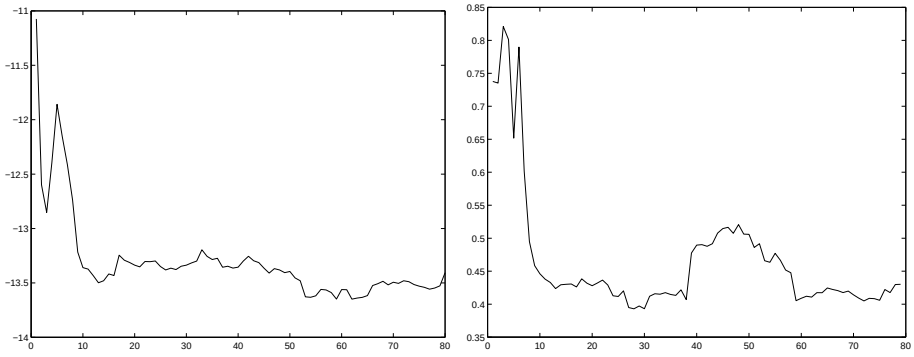

Figure 4: $\log_{10}(\mathcal{P})$ and estimated standard deviation, versus number of images in the second experiment.

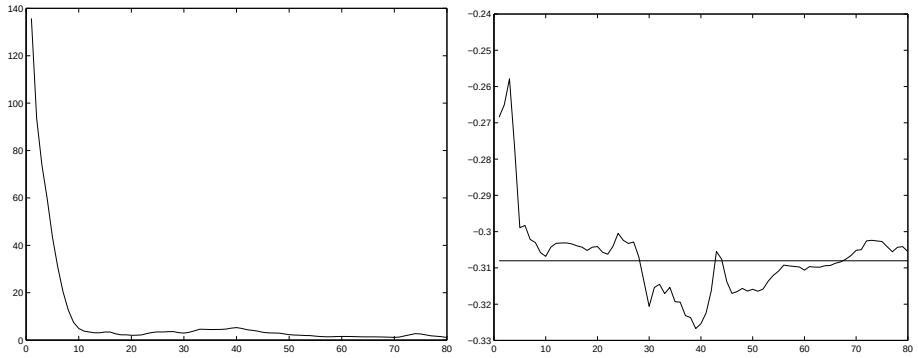

Figure 5: RMS error of reconstruction and the projective invariant, versus number of images in the second experiment. The correct value of the projective invariant is shown as a line.

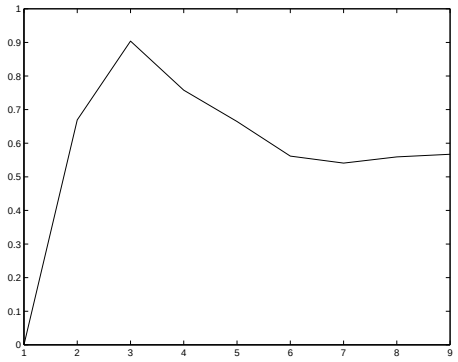

Figure 6: Estimated standard deviation in the images versus number of images in the third experiment.

5.2 Real Data

A closely space image sequence consisting of 10 images, where taken using a CCD-camera mounted on an ABB-robot. The scene consists of a number of toy blocks and some other things. We used the Harris corner detector together with a standard correlation tracker and obtained 14 tracked corners through the whole sequence. The typical number of corners obtained per image is about $100 - 200$, but some points are lost from one image to the next and thus omitted even if they appear again later. Using a more sophisticated tracker it would be possible to obtain more than 14 tracked corners. However, the objective is to show how the recursive algorithm works on real data and not to implement a sophisticated tracker.

In Figure 6, the estimated standard deviation in the images versus the number of images are shown. It can be seen that the standard deviation is very low for a small number of images. This is due to the fact that since the image are so similar it is easy to obtain a reconstruction that projects very closely to the detected corners. Then the standard deviation gets larger when more image are used, since they differ more and more from each other. Finally, the standard deviation goes down and stabilizes for a large number of images, due to the convergence of the algorithm. Finally, in Figure 7, the first and the seventh images of the sequence are shown, together with detected corners (asterisks) and re-projected corners (circles).

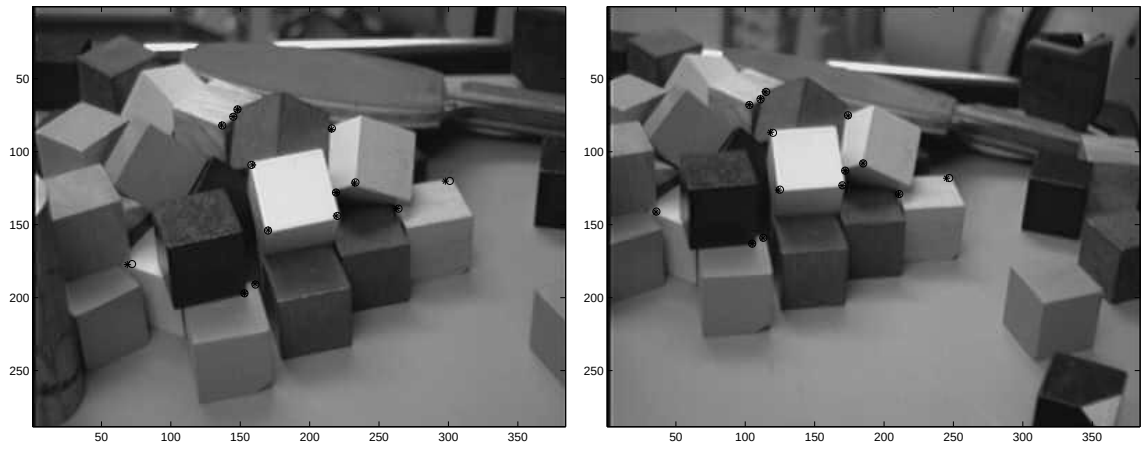

Figure 7: The first and the seventh image together with detected corners (*) and re-projected corners (o).

6 Conclusions

In this paper we have presented a recursive algorithm that solves the structure and motion from image sequences problem in a generic way. The algorithm relies on subspace methods and has the following advantages:

- The result is independent of the chosen coordinate systems in the different images.
- The result is independent of the ordering of the points.
- The convergence is good (less than 10 steps).

The performance of the algorithm has been shown on simulated data with added noise. We have shown that the resulting errors in the images are only about 50% higher than the standard deviation of the added noise.

Moreover, two different approaches to structure and motion from image sequences have been unified. Both are based on coordinate independent representations; one using the concept of affine shape and depth and the other using depth spaces. We have shown that they are dual to each other and result in equivalent algorithms.

References

[1] Faugeras, O., D., What can be seen in three dimensions with an uncalibrated stereo rig?, *ECCV'92, Lecture notes in Computer Science, Vol 588. Ed. G. Sandini, Springer-Verlag,* 1992, pp. 563-578.

[2] Hartley, R., I., Projective Reconstruction and Invariants from Multiple Images, *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 16, no. 10, pp. 1036-1041, 1994.

[3] Heyden, A., Reconstruction from Image Sequences by means of Relative Depths, *Proc. ICCV'95, IEEE Computer Society Press,* 1995, pp. 1058-1063, Also to appear in *IJCV, International Journal of Computer Vision.*

[4] Heyden, A., Åström, K., A Canonical Framework for Sequences of Images, *Proc. IEEE Workshop on Representation of Visual Scenes,* 1995.

[5] Heyden, A., Projective Structure and Motion from Image Sequences using Subspace Methods, *submitted to SCIA'97, Scandinavian Conference on Image Analysis, Lappenranta, Finland.*

[6] Luong, Q.-T., Vieville, T., Canonic Representations for the Geometries of Multiple Projective Views, *ECCV'94, Lecture notes in Computer Science, Vol 800. Ed. Jan-Olof Eklund, Springer-Verlag,* 1994, pp. 589-599.

[7] McLauchlan, P., F., Murray, D., W., A unifying framework for structure and motion recovery from image sequences, *Proc. ICCV'95, IEEE Computer Society Press,* 1995, pp. 314-320.

[8] Quan, L., Invariants of 6 Points from 3 Uncalibrated Images, *ECCV'94, Lecture notes in Computer Science, Vol 801. Ed. J-O. Eklund, Springer-Verlag* 1994, pp. 459-470.

[9] Shashua, A., Navab, N., Relative Affine Structure: Theory and Application to 3D Reconstruction from Perspective Views, *Proc. Conf. Computer Vision and Pattern Recognition*, 1994, pp. 483-489.

[10] Soatto, S., Perona, P., Dynamic Rigid Motion Estimation From Weak Perspective, *Proc. ICCV'95, IEEE Computer Society Press,* 1995, pp. 321-328.

[11] Sparr, G., An algebraic-analytic method for affine shapes of point configurations, *proceedings 7th Scandinavian Conference on Image Analysis,* 1991, pp. 274-281.

[12] Sparr, G., Depth-Computations from Polyhedral Images, *ECCV'92, Lecture notes in Computer Science, Vol 588. Ed. G. Sandini, Springer-Verlag,* 1992, pp. 378-386. Also in *Image and Vision Computing, Vol 10.*, 1992, pp. 683-688.

[13] Sparr, G., A Common Framework for Kinetic Depth, Reconstruction and Motion for Deformable Objects, *ECCV'94, Lecture notes in Computer Science, Vol 801. Ed. J-O. Eklund, Springer-Verlag* 1994, pp. 471-482.

[14] Sparr, G., Simultaneous Reconstruction of Scene Structure and Camera Locations from Uncalibrated Image Sequences, *proceedings 13th International Conference on Pattern Recognition,* 1996, pp. 328-333.

[15] Sturm, P., Triggs, B., A Factorization Based Algorithm for Multi-Image Projective Structure and Motion, *ECCV'96, Lecture notes in Computer Science, Vol 1065. Ed. B. Buxton and R. Cipolla, Springer-Verlag* 1996, pp. 709-720.