Department of Electrical and Computer Systems Engineering

Technical Report MECSE-32-2005

Homography estimation and heteroscedastic noise - a first order perturbation analysis

P. Chen and D. Suter

MONASH UNIVERSITY

Homography estimation and heteroscedastic noise - a first order perturbation analysis

Pei Chen and David Suter
The Centre for Perceptive & Intelligent Machines in Complex Environments
Departement of Electrical & Computer Systems Engineering, Monash University

Abstract: It is well known that one can collect the coefficients of the homgraphies between two views into a large, rank deficient matrix. In principle, such an observation implies that one can refine the accuracy of the estimates of the homography coefficients by exploiting the rank constraint. However, the straightforward approach suggested by this observation is impractical because it requires many homographies and it also does not take into account correlations between the errors in the coefficients.

In a companion paper [4], we show how to jointly estimate multiple (but a realistic number of) homographies over 2 views. By studying the special structure of the homography, we show that it is possible to calculate the dimension 4 subspace of the homographies from ≥ 3 planes (and, in principle, with even two planes). This contradicts what seems to be the accepted situation regarding the exploitation of the rank-4 constraint amongst homographies: that more than 4 planes are needed to calculate and exploit the dimension 4 subspace. Practical issues arise because the homography coefficients, before rank-constrained refinement, are themselves estimates whose noise covariances need to be characterised and accounted for. In this paper, we develop a statistical analysis allowing for estimation of the covariance matrices required for the calculation of low-rank "denoised" homographies.

1 Introduction

The homography is a projective linear mapping with much utility in computer vision and photogrammetry. Much of this stems from the fact that the homography is the mapping between the images of corresponding points, when those are the image points of 3D planar "real world" points. The calculation of homographies has many applications: some stemming from its utility in the transfer of points from one image to another (registration, construction of panoramas etc.), and others from its utility in the extraction of camara and planar patch relative poses (robotics, photogrammetry etc.). Thus, advances in accuracy of homography registration has many applications. This paper is concerned with improving such accuracy. We do so by exploiting the well established fact that homographies are often related in particular ways related to subspaces and the ranks of associated matrices; in particular the rank 4 constraint [11] (see section 3).

One of the ways in which one can exploit the rank constraints is to project the noisy data (the measured homography coefficients which generally will not maintain the low rank predicted from the noise-free quantities) onto the low rank subspace as a form of "denoising". In [3], the "denoising" capacity of the projection of a large low-rank matrix has been analyzed. In the presence of i.i.d. Gaussian noise, the error, still residing in the low-rank approximation matrix, depends on the ratio between the degree of freedom of the subspace and that of the large matrix.

If we (temporarily, this issue is the main topic of this paper) suppose the errors in the homography matrix entries are i.i.d. Gaussian then, if there are enough planes observed ($>> 9$), the rank-4 projection will result in the error in the homographies being reduced by a factor of $\sqrt{\frac{4}{9}}$ [3]. That is, a one third (33%) reduction in the average error in the coefficents of the homographies can be gained. However, a large number of planes are required to gain such a ratio (we note that the gains reported in [3], using synthetic data, see figure 1 of that paper, *are much less*).

However, in practice, two factors prevent a useful direct application of the "denoising" capacity to the homography refinement to achieve even the modest gains reported in [3].

First, if we were to determine the dimension 4 subspace directly from the currently available approaches then at least 4 planes are required (and to actually gain any denoising from such a recovered subspace, *more than 4 planes are required.*) There are some image pairs where it would be difficult to find 4 planes (of any significant size).

Second, the refinement starts from the *estimated* homography parameters and, even if the image measurements that were used to estimate these parameters were corrupted with i.i.d. Gaussian noise, the same cannot be said about the parameters themselves. More precisely, the homography estimation suffers from a well-known heteroscedastic noise problem [8, 9]: The heteroscedastic noise arises due to the linearization in the direct linear transformation (DLT) algorithm [6].

Without tackling these issues, it isn't possible, in practice to achieve even the modest gains reported in [3] (i.e., using real data rather than synthetic data), let alone come close to the theoretical 33% gain for large numbers of homographies with i.i.d. noise in their coefficients.

In order to find a useful algorithm for so few planes, in a companion paper [4] we study the special structure of the homography. Starting from the fact that the homographies can be shown to lie in a rank 3 *affine* subspace, we show how 3 homographies suffice to obtain useful constraints as it turns out that only a subset of 3-dimensional affine subspaces are possible homographies. We show how to obtain the subspace with even one plane and the Fundamental matrix. Further, we show that there is even more exploitable structure in that the homography coefficients can be shown to lie in the "sum" of two rank one subspaces and thus useful constraints can be applied with even as few as two planes (without the fundamental matrix). The latter two special cases turn out to have some value in our overall algorithm, even though we target the situation with ≥ 3 planes.

A practical implmentation of the algorithm suggested by the abovementioned analysis, requires on to overcome the difficulty, associated with the heteroscedastic noise in the homography parameters. In this paper we study the covariance of the error in the parameters. To do this, we track the propagation of the noise in the image features, by employing first order approximation techniques.

In section 2, we set the background by reviewing the relationship between the homography and the associated projection matrices and the rank 4 constraint for the induced homography matrix. In section 3, we hint at how it is, in principle, possible to calculate a low-rank subspace with as few as two planes. The details, are in the companion paper [4] but we give enough details here to introduce a matrix of importance and whose noise characteristics we also analyse here. In section 4, we first review the DLT and the normalized DLT algorithms for the homography estimation. In section 5, we present how to analytically compute the statistical property of the error in the estimated homography parameters (and in an associated matrix required for our algorithm).

2 Rank-4 constraint

First, we cite the *Result 12.1* on page 312 of [6], which describes the relationship between a homography and the projection matrices. Given the projection matrices for 2 views

$$\mathbf{P} = [\mathbf{I}|\mathbf{0}] \qquad \mathbf{P}' = [\mathbf{R}|\mathbf{t}] \tag{1}$$

and the i^{th} plane defined by $\pi_i^T \mathbf{X} = 0$ with $\pi_i = \begin{bmatrix} \mathbf{v}_i^T & 1 \end{bmatrix}^T$, the homography induced by the plane is $\mathbf{x}' = \mathbf{H}_i \mathbf{x}$ with a matrix representation:

$$\mathbf{H}_i = \mathbf{R} - \mathbf{t}\mathbf{v}_i^T \tag{2}$$

Thus, with the knowledge of $\mathbf{R}$ and $\mathbf{t}$, the homography of the i^{th} plane is characterized by the vector $\mathbf{v}_i$. Note that this is a *particular* representation (we call it the *canonical representation*): all matrices related to this matrix by a scale are also representations of the same homography.

The matrix $\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \ldots & \mathbf{h}_n \end{bmatrix}_{9,n}$, whose columns are homographies in canonical form, can be expressed as the following, in terms of $\mathbf{R}$, $\mathbf{t}$, and $\{\mathbf{v}_i\}$:

$$\mathbf{H} = vec(\mathbf{R}^T) \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}_{1,n} - \mathbf{U}_{\mathbf{t}} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_n \end{bmatrix}_{3,n} \tag{3}$$

where

$$\mathbf{U}_{\mathbf{t}} = \begin{bmatrix} t_1 \mathbf{I}_3 \\ t_2 \mathbf{I}_3 \\ t_3 \mathbf{I}_3 \end{bmatrix}_{9,3} \tag{4}$$

If we know $\mathbf{R}$, $\mathbf{t}$ and $\{\mathbf{v}_i\}$, and we calculate the homographies according to (2), then homography matrix is embedded in a dimension 3 affine subspace. However, homographies are only defined up to a scale factor, and in practice we calculate the homography up to an unknown scale. We can choose to select a special matrix representation for the class of matrices representing the homography - for example, we can usually normalize the homography so that $||\mathbf{H}_i||_F = 1$. However, any such choice will lead to different (and unknown) scale factors between the chosen representative and the homography matrix defined by (2). In effect, this means that the left hand vector of ones in (3) are now factors generally different from unity. As a consequence, we can only calculate the dimension 4 subspace, rather than the dimension 3 affine subspace. However, the homography coefficients do lie in a particular restricted set of 3-dimensional affine manifolds and this fact can be exploited to find the embedding 4-dimensional subspace with less than 4 homographies.

3 Calculation of the 4-dimensional subspace

Calculating the dimension 4 subspace would seem to require at least 4 planes (and then perform an SVD to obtain a basis for the column space). This observation was made by Zelnik-Manor and Irani [14, 7] who used it to criticise the utility of the homography rank constraint and to contrast it with their homology based constraints which could be reduced to rank-3 form. However, due to the special structure of the expressions for the homographies, one can in fact adopt a less obvious approach that does not require 4 planes.

In this section, we hint at how to calculate the basis vectors of the embedding 4-dimensional subspace, by studying the special structure of the homography from (2, 3).

In fact, one should not be surprised that 4 planes are not required to calculate the subspace since the basis is clearly dependent only on $\mathbf{t}$ and $\mathbf{R}$. Let us consider this problem from a purely algebraic point of view (and ignoring the internal structure of $\mathbf{R}$ etc.). For n planes, from (2), we need $9+3+3n$ parameters: 9 for $\mathbf{R}$, 3 for $\mathbf{t}$ and 3 for each $\mathbf{v}_i$. While for n homographies, there are $9n$ parameters (not all independent of course). If

$$9n \geq 9+3+3n \tag{5}$$

there exists possibility of determining $\mathbf{R}$, $\mathbf{t}$ and $\{\mathbf{v}_i\}$. The solution for (5) is $n \geq 2$. Two planes suffice to determine the dimension 4 subspace. However, it is also possible to calculate the subspace from the fundamental matrix and only one plane.

3.1 Calculation from ≥ 4 planes

This is the well established approach an is included here for completeness and to introduce some notation.

Suppose that n ($n \geq 4$) planes are observed over 2 views. For the i^{th} plane, there exists a homography with matrix representation [1]
$\mathbf{H}_i$: $\mathbf{H}_i = \begin{bmatrix} h_{1,i} & h_{2,i} & h_{3,i} \\ h_{4,i} & h_{5,i} & h_{6,i} \\ h_{7,i} & h_{8,i} & h_{9,i} \end{bmatrix}$, which projects the points of the first view upon the second view. The homography $\mathbf{H}_i$ can be estimated from ≥ 4 points [6]. If we arrange the vector $\mathbf{h}_i = \begin{bmatrix} h_{1,i} & h_{2,i} & \dots & h_{9,i} \end{bmatrix}^T$ as the i^{th} column of a matrix $\mathbf{H}$, i.e, $\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \dots & \mathbf{h}_n \end{bmatrix}$, then $rank(\mathbf{H}) = 4$ [11], as shown above. In other words, all the vector homographies (columns of $\mathbf{h}_i$) are restricted to a dimension 4 subspace. All we need do, in principle, is to use the Singular Valued Decomposition to project the columns of $\mathbf{H}_i$ onto the basis formed by the singular vectors associated with the four largest singular values. However, as mentioned above, this will not be optimal if the noise in the homography coefficients is heteroscedastic. Moreover, to achieve any reasonable "denoising" effect, much greater than 4 planes are needed in practice.

[1] In this paper, a homography will be represented either as a 3×3 matrix or as a 9×1 vetor. We will use $\mathbf{H}_i$ or $\mathbf{h}_i$ for a matrix homography or its vectorised form, respectively. When its form can be determined from the context, we only use the term of "homography", which might be a matrix or a vector. $\mathbf{H}$ will denote the matrix that have several vector homographies as its columns, as defined above. $\mathbf{H}$ is refered to as "homography matrix".

3.2 Calculation from ≥ 3 planes

The crux is to note that the special structure of the Homography matrix implies that we need only know the camera pose translation $\mathbf{t}$, or, more accurately, only the direction of that vector. Thus, we decompose the problem of finding the subspace basis into first finding this direction, then solving for the remaining basis vector.

3.2.1 Calculation of the direction of translation t

From equation (3), we can see that second part of the column span is determined by $\mathbf{U_t}$, as defined in (4), which has only 3 parameters.

From the structure of $\mathbf{U_t}$, one can see that only a restricted set of affine spaces are possible - regardless of the values of $\mathbf{t}$ there are directions in R^9 that are not in the span of the columns of $\mathbf{U_t}$.

Moreover, we can determine $\mathbf{U_t}$ up to a scale by determing the direction of $\mathbf{t}$. In the following, we show that the direction of $\mathbf{t}$ can be calculated from only 3 homographies.

Proof: Suppose 2 homographies with matrix representations $\mathbf{H}_i$ and $\mathbf{H}_j$ have been calculated, up to different unknown scales (all that can be done in practice): $\mathbf{H}_i = \lambda_i \left(\mathbf{R} - \mathbf{t}\mathbf{v}_i^T\right)$ and $\mathbf{H}_j = \lambda_j \left(\mathbf{R} - \mathbf{t}\mathbf{v}_j^T\right)$.

Note: there exist 2 independent 3-vectors $\mathbf{l}_i$ for $i = 1, 2$ that span $\mathbf{t}^\perp$ (i.e., $span\{\mathbf{l}_1, \mathbf{l}_2\} = \mathbf{t}^\perp$) and further

$$\mathbf{l}^T\mathbf{H}_i = \lambda \mathbf{l}^T\mathbf{H}_j \tag{6}$$

where $\mathbf{l} = \begin{bmatrix} l_x & l_y & l_z \end{bmatrix}^T = c_1\mathbf{l}_1 + c_2\mathbf{l}_2 \in span\{\mathbf{l}_1, \mathbf{l}_2\}$ and $\lambda = \frac{\lambda_j}{\lambda_i}$.

From (6), their cross product is a zero vector, i.e., $\left[\mathbf{H}_i^T\mathbf{l}\right]_\times \mathbf{H}_j^T\mathbf{l} = \mathbf{0}$, for any c_1 and c_2. In column vector notation, we write the cross product as:

$$\mathbf{M}_{i,j}\mathbf{L} = \mathbf{0} \tag{7}$$

where $\mathbf{L} = \begin{bmatrix} l_x^2 & l_y^2 & l_z^2 & l_xl_y & l_xl_z & l_yl_z \end{bmatrix}^T$ and $\mathbf{M}_{i,j}$ is a 3×6 matrix, whose coefficients are defined in the appendix.

For n ($n \geq 3$) homographies, we stack all the pairwise matrices $\mathbf{M}_{i,j}$ for $i < j$, obtaining a $3C_n^2 \times 6$ matrix $\mathbf{M}$:

$$[\mathbf{M}]_{3C_2^n,6} = \begin{bmatrix} \mathbf{M}_{1,2}^T & \dots & \mathbf{M}_{1,n}^T & \mathbf{M}_{2,3}^T & \dots & \mathbf{M}_{2,n}^T & \dots & \mathbf{M}_{n-1,n}^T \end{bmatrix}_{3C_n^2,6}^T \tag{8}$$

First, we prove that $rank(\mathbf{M}) = 3$ and $\mathbf{M}$ has a dimension-3 null space, except when all the planes are parallel.

Define $\mathbf{L}_1 = \begin{bmatrix} l_{1,x}^2 & l_{1,y}^2 & l_{1,z}^2 & l_{1,x}l_{1,y} & l_{1,x}l_{1,z} & l_{1,y}l_{1,z} \end{bmatrix}^T$,
$\mathbf{L}_2 = \begin{bmatrix} l_{2,x}^2 & l_{2,y}^2 & l_{2,z}^2 & l_{2,x}l_{2,y} & l_{2,x}l_{2,z} & l_{2,y}l_{2,z} \end{bmatrix}^T$ and
$\mathbf{L}_{1,2} = \begin{bmatrix} 2l_{1,x}l_{2,x} & 2l_{1,y}l_{2,y} & 2l_{1,z}l_{2,z} & l_{1,x}l_{2,y} + l_{2,x}l_{1,y} & l_{1,x}l_{2,z} + l_{2,x}l_{1,z} & l_{1,y}l_{2,z} + l_{2,y}l_{1,z} \end{bmatrix}^T$. From the analysis above, it can be easily obtained that $\mathbf{M}_{i,j}\mathbf{L}_1 = \mathbf{0}$ and $\mathbf{M}_{i,j}\mathbf{L}_2 = \mathbf{0}$, as special

cases of (7). Consequently, from $\mathbf{L} = c_1^2\mathbf{L}_1 + c_2^2\mathbf{L}_2 + c_1c_2\mathbf{L}_{1,2}$ and the fact that $\mathbf{M}_{i,j}\mathbf{L} = \mathbf{0}$ holds for any c_1 and c_2, $\mathbf{M}_{i,j}\mathbf{L}_{1,2} = \mathbf{0}$ also holds. Thus, there exists a dimension-3 subspace: $span\{\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_{1,2}\}$, which is contained in the null space of $\mathbf{M}_{i,j}$ for $i < j$. Consequently, $span\{\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_{1,2}\}$ is contained in the null space of $\mathbf{M}$, because the subspace of $span\{\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_{1,2}\}$ is independent of c_1 and c_2.

Thus, $rank(\mathbf{M}) \leq 6-3$, because $\mathbf{M}$ has a width of 6. Another fact is $rank(\mathbf{M}) \geq 3$ holds if not all the planes are parallel. (Here, we do not present a proof for this.) Thus, $rank(\mathbf{M}) = 3$ holds under this assumption.

Then, we show how to calculate the vector $\mathbf{t}$ from the knowledge of the dimension-3 null space of $\mathbf{M}$. In practice, we calculate this dimension-3 null space of $\mathbf{M}$ up to a non-singular 3×3 transform $\mathbf{A}$. That is, the bais vectors spanning the null-space $[\mathbf{N}]_{6,3} = \begin{bmatrix} \mathbf{n}_1 & \mathbf{n}_2 & \mathbf{n}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{L}_1 & \mathbf{L}_2 & \mathbf{L}_{1,2} \end{bmatrix} \mathbf{A}$. Each column of $\mathbf{N}$ can be arranged as a 3×3 matrix: $\mathbf{N}_i = \begin{bmatrix} n_{1,i} & n_{4,i} & n_{5,i} \\ n_{4,i} & n_{2,i} & n_{6,i} \\ n_{5,i} & n_{6,i} & n_{3,i} \end{bmatrix}$.

Now, we will prove that $\mathbf{t}$ is the left null vector of $\mathbf{N}_i$, i.e., $\mathbf{t}^T\mathbf{N}_i = \mathbf{0}$.

If $\mathbf{A}$ is an identity matrix, i.e., $\mathbf{A} = \mathbf{I}_3$, $\mathbf{N}'_i = \begin{bmatrix} l_{i,x} & l_{i,y} & l_{i,z} \end{bmatrix}^T \begin{bmatrix} l_{i,x} & l_{i,y} & l_{i,z} \end{bmatrix}$ for $(i = 1,2)$, and $\mathbf{N}'_3 = \begin{bmatrix} l_{1,x} & l_{1,y} & l_{1,z} \end{bmatrix}^T \begin{bmatrix} l_{2,x} & l_{2,y} & l_{2,z} \end{bmatrix} + \begin{bmatrix} l_{2,x} & l_{2,y} & l_{2,z} \end{bmatrix}^T \begin{bmatrix} l_{1,x} & l_{1,y} & l_{1,z} \end{bmatrix}$. Then $\mathbf{t}^T\mathbf{N}'_i = \mathbf{0}$ holds for $i = 1,2,3$, because of $\mathbf{l}_i \in \mathbf{t}^\perp$ for $i = 1,2$.

For any nonsingular $\mathbf{A}$, $\mathbf{N}_i = \sum_{k=1}^{3} a_{k,i}\mathbf{N}'_k$, because the column of $\mathbf{N}$ is a linear combination of $\mathbf{L}_1$, $\mathbf{L}_1$ and $\mathbf{L}_{1,2}$. Thus $\mathbf{t}^T\mathbf{N}_i = \mathbf{0}$ also holds. In order to make full use of the available data, we juxtapose $\mathbf{N}_i$ as:

$$\mathbf{S} = \begin{bmatrix} \mathbf{N}_1 & \mathbf{N}_2 & \mathbf{N}_3 \end{bmatrix} \tag{9}$$

The solution of $\mathbf{t}$, up to an unknown scale, is the left null vector of $\mathbf{S}$. In presence of noise, the left singular vector of $\mathbf{S}$, associated with the least singular value, is taken as the solution of $\mathbf{t}$. **End of the proof**

In the analysis above, we need to calculate the dimension 3 null subspace of $\mathbf{M}$. If the homographies are noise free, there exists such a null subspace. However, error is inevitably introduced in the homographies, due to the presence of noise in feature points. In practice, we can use the SVD to calculate this null subspace: taking the 3 right singular vectors that are associated with the 3 smallest singular values.

However, in doing so, the approach severely suffers from the noise in the images, especially for the case of 3 planes. First, the homography estimation suffers from the heteroscedastic noise. The error in the homography parameters cannot be modeled as i.i.d. Gaussian again. Second, the process of calculating the matrix $\mathbf{M}$ also introduces the heteroscedastic noise even if the error in the homographies could be modeled as i.i.d. Gaussian. Because of these considerations, the direct SVD solution is very sensitive to the noise in the images.

In order to overcome this difficulty with the heteroscedastic noise, it is necessary to model the error in the matrix of $\mathbf{M}$ and then to employ the bilinear approach [1, 2] or the AP approach [10] to calculate weighted rank-3 approximation matrix: $\mathbf{M}^3$. Then, a reasonable solution of $\mathbf{N}$ is the null subspace of this $\mathbf{M}^3$. We present the statistical analysis of the error in the homography parameters and in $\mathbf{M}$ in section 5. The definition of the weighted rank-r approximation matrix is provided in the appendix.

4 Normalized DLT algorithm for homography estimation embedded in a dimension 4 subspace

4.1 Overview of the DLT and the normalized DLT algorithm for homography estimation

Here, we first review the normalized direct linear transform (DLT) algorithm [6] for homography estimation. In this section, we suppose the third coordinate of the homogeneous representation of a point is 1.

For a homography $\mathbf{H}_k = \begin{bmatrix} h_{1,k} & h_{2,k} & h_{3,k} \\ h_{4,k} & h_{5,k} & h_{6,k} \\ h_{7,k} & h_{8,k} & h_{9,k} \end{bmatrix}$, which maps the $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & 1 \end{bmatrix}^T$ of the k^{th} plane in the first view upon $\mathbf{x}' = \begin{bmatrix} x_1' & x_2' & 1 \end{bmatrix}^T$ on the second view: $\mathbf{x}' = \lambda \mathbf{H}_k \mathbf{x}$.

From $\mathbf{x}' \times \mathbf{H}_k \mathbf{x} = \mathbf{0}$, each pair of the matches, $\{\mathbf{x}_i, \mathbf{x}_i'\}$, produces a 3×9 matrix:

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{0} & -\mathbf{x}_i^T & x_{2,i}'\mathbf{x}_i^T \\ \mathbf{x}_i^T & \mathbf{0} & -x_{1,i}'\mathbf{x}_i^T \\ -x_{2,i}'\mathbf{x}_i^T & x_{1,i}'\mathbf{x}_i^T & \mathbf{0} \end{bmatrix} \tag{10}$$

which satisfies $\mathbf{A}_i \mathbf{h}_k = \mathbf{0}$. Stack ≥ 4 $\mathbf{A}_i$ as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1^T & \dots & \mathbf{A}_n^T \end{bmatrix}^T \tag{11}$$

$\mathbf{A}\mathbf{h}_k = \mathbf{0}$ holds. $\mathbf{h}_k$ is taken as the right singular vector of $\mathbf{A}$, associated with the least singular value. This is the DLT algorithm [6] for homography estimation.

In [6], a normalization step has been recommended. It consists of a translation and a scaling, so that the centroid of the transformed points is the origin $(0, 0)$ and their average distance from the origin is $\sqrt{2}$. Suppose the centroid of the original points is (c_1, c_2) and their average distance to this centroid is l. The normalization transform $\mathbf{T}$ is

$$\begin{bmatrix} \frac{1}{l} & 0 & -\frac{c_1}{l} \\ 0 & \frac{1}{l} & -\frac{c_2}{l} \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

Similarly, a normalization transform for the second view, $\mathbf{T}'$, can be calculated.

The normalized DLT algorithm takes the DLT algorithm as the core algorithm. First, calculate the transformed points for each view and their associated normalization transforms $\mathbf{T}$ and $\mathbf{T}'$. Second, using DLT, calculate the homography $\widetilde{\mathbf{H}}_k$ from the normalized matches. Last, in the denormalization step, set

$$\mathbf{H}_k = \mathbf{T}'^{-1} \widetilde{\mathbf{H}}_k \mathbf{T} \tag{13}$$

as the homography for the plane in the original views.

4.2 Homography estimation embedded in a dimension 4 subspace

In [3], it has been shown that, for the case of > 4 planes over 2 views, the accuracy of the homography can be improved by projecting the homographies upon to the dimension 4 subspace, which is calculated from the homographies. However, although the new homography claculated this way is optimal, measured by its distance to the dimension 4 subspace, it cannot be ensured that this new homography is optimal in terms of the mapping accuracy.

Here, we first show how to calculate the homography embedded in a dimension 4 subspace: We do not directly project one homography upon the dimension 4 subspace, instead we try to find a homography, embedded in the dimension 4 subspace, which is optimal in terms of the minimization of $||\mathbf{Ah}||_F$.

Suppose the dimension 4 subspace basis $\mathbf{U} \in R^{9,4}$ is known and the linearization matrix is $\mathbf{A}$ as in (11). The subspace constrained DLT solution is as follows: First, calculate the solution of $\mathbf{AUx} = \mathbf{0}$ as $\hat{\mathbf{x}} = \mathbf{b}$ (standard smallest singular value way). Second, take the $\mathbf{Ub}$ as the solution of the homography, which is obviously embedded in the subspace $\mathbf{U}$.

As in the normalized DLT, we also use the normalization step in this dimension-4 constrained homography estimation. Suppose n planes are available. First, taking all the feature points in the n planes as a whole set, calculate the normalization transforms $\mathbf{T}$ and $\mathbf{T}'$, for the first view and the second view respectively. Second, for each normalized plane, calculate its homography, and calculate the dimension 4 subspace $\mathbf{U}$ of these homographies. Third, for each normalized plane, calculate its subspace-$\mathbf{U}$ constrained homography. Finally, calculate the denormalized homographies for all the planes, as in the denormalization step of the normalized DLT.

This is the basic idea. However, we incorporate several refinements as presented in [4].

5 Statistical analysis of the error in the homography parameters and in the matrix M in (8)

In section 4 we sketeched some ideas behind [4]. One issue we address in detail in this paper is: how to obtain the statistical properties of the error in the homography parameters and in the matrix $\mathbf{M}$ in (8). In this section, we present a statistical analysis of the error in the estimated homography parameters. The covariance matrix of the error in 9 parameters is analytically computed. First, we show how to calculate the covariance matrix of the errors for the DLT algorithm. Then, we extend this to the normalized DLT algorithm. We also show how to compute the covariance matrix of the error in the matrix $\mathbf{M}$ in (8). We assume a small noise level so that the first order expansions/approximation can be used.

5.1 The case of the DLT algorithm

Suppose the matrix $\mathbf{A}$ in (11) is obtained from n noise free feature matches and that the i^{th} noise free feature matches $\mathbf{x}_i$ and $\mathbf{x}'_i$ are corrupted with the noise of $(\varepsilon_{i,1}, \varepsilon_{i,2})$, and $\left(\varepsilon'_{i,1}, \varepsilon'_{i,2}\right)$, respectively. [2]

The essence of the analysis below is to represent the error in the homography parameters in terms of the random variables $\{\varepsilon_{i,1}, \varepsilon_{i,2}, \varepsilon'_{i,1}, \varepsilon'_{i,2}\}$ for $(1 \le i, j \le n)$. Here, we use the second subscript in $\varepsilon_{i,\bullet}$ to denote the x or y coordinates in the 2d images.

Using the SVD [5], $\mathbf{A}$ can be decomposed as: $\mathbf{A} = \mathbf{USV}^T$, where $\mathbf{U} \in \mathbf{R}^{3n,3n}$, $\mathbf{V} \in \mathbf{R}^{9,9}$, $\mathbf{UU}^T = \mathbf{I}_{3n}$, $\mathbf{VV}^T = \mathbf{I}_9$, and $\mathbf{S} = diag\{s_1, s_2, \ldots, s_8, 0\} \in R^{3n,9}$. The noise-free homography vector is the 9^{th} column of $\mathbf{V}$: $\mathbf{v}_9$. Due to the noise of $\{\varepsilon_{i,1}, \varepsilon_{i,2}\}$ and $\{\varepsilon'_{i,1}, \varepsilon'_{i,2}\}$, in $\mathbf{x}_i$ and $\mathbf{x}'_i$ respectively, the error $\mathbf{E}_i$ in the i^{th} block of $\mathbf{A}$, $\mathbf{A}_i$, is:

$$\mathbf{E}_i = \begin{bmatrix} 0 & 0 & 0 & -\varepsilon_{i,1} & -\varepsilon_{i,2} & 0 & E_{i,\{1,7\}} & E_{i,\{1,8\}} & \varepsilon'_{i,2} \\ \varepsilon_{i,1} & \varepsilon_{i,2} & 0 & 0 & 0 & 0 & E_{i,\{2,7\}} & E_{i,\{2,8\}} & -\varepsilon'_{i,1} \\ E_{i,\{3,1\}} & E_{i,\{3,2\}} & -\varepsilon'_{i,2} & E_{i,\{3,4\}} & E_{i,\{3,5\}} & \varepsilon'_{i,1} & 0 & 0 & 0 \end{bmatrix} \quad (14)$$

where $E_{i,\{1,7\}} = x'_{i,2}\varepsilon_{i,1} + x_{i,1}\varepsilon'_{i,2} = -E_{i,\{3,1\}}$, $E_{i,\{1,8\}} = x'_{i,2}\varepsilon_{i,2} + x_{i,2}\varepsilon'_{i,2} = -E_{i,\{3,2\}}$, $E_{i,\{2,7\}} = -x'_{i,1}\varepsilon_{i,1} - x_{i,1}\varepsilon'_{i,1} = -E_{i,\{3,4\}}$, and $E_{i,\{2,8\}} = -x'_{i,1}\varepsilon_{i,2} - x_{i,2}\varepsilon'_{i,1} = -E_{i,\{3,5\}}$. Quadratic terms have been dropped.

Define $\mathbf{C}$ as the tranformed error matrix:

$$\mathbf{C} = \mathbf{U}^T\mathbf{EV} \quad (15)$$

From the matrix perturbation theory [13, 12], the first order perturbed solution for the DLT algorithm is

$$\mathbf{v}'_9 = \mathbf{v}_9 - \sum_{i=1}^{8} \frac{c_{i,9}\mathbf{v}_i}{s_i} \quad (16)$$

The second term in (16) is the error in the estimated parameters.

The entries in $\mathbf{E}_i$ are random variables, as are $c_{i,j}$ also. Each $c_{i,j}$ is a linear combination of the $4n$ random variables: $\{\varepsilon_{i,1}, \varepsilon_{i,2}, \varepsilon'_{i,1}, \varepsilon'_{i,2}\}$ for $1 \le i, j \le n$. Consequently, the second term in (16) is also a random vector: each entry of which is a linear combination of the $4n$ random variables. Thus, we can express the error in the parameters as a $9 \times 4n$ matrix: $\Xi_{\mathbf{h}} \in R_{9\times 4n}$.

In order to do this, we represent the errors of $\mathbf{E}$ as a linear combination of the $4n$ matrices: a stack of $4n$ $3n \times 9$ matrices, each of which represents the error component in one of $4n$ "directions": $\{\varepsilon_{i,1}, \varepsilon_{i,2}, \varepsilon'_{i,1}, \varepsilon'_{i,2}\}$ for $1 \le i \le n$.

$$\mathbf{E} = \sum_{i=1}^{n} \left(\varepsilon_{i,1}\mathbf{E}^{4i-3} + \varepsilon_{i,2}\mathbf{E}^{4i-2} + \varepsilon'_{i,1}\mathbf{E}^{4i-1} + \varepsilon'_{i,2}\mathbf{E}^{4i}\right) \quad (17)$$

For example, the $(4(i-1)+j)^{th}$ matrix $\mathbf{E}^{4(i-1)+j}$, for $1 \le i \le n$ and $1 \le j \le 4$, is

$$\begin{bmatrix} \mathbf{0}^T & \ldots & \mathbf{0}^T & \left[\overline{\mathbf{E}}^{4(i-1)+j}\right]^T & \mathbf{0}^T & \ldots & \mathbf{0}^T \end{bmatrix}^T \quad (18)$$

[2]Note that $\mathbf{x}$ in subsection 4.1 is used for the homogeneous representation of a feature point. By a slight abuse of notation, we will also use $\mathbf{x}$ to represent the feature points in nonhomogeneous form: with x and y coordinates as its two entries.

where $\mathbf{0}$ is a 3×9 zero matrix and only the i^{th} block $\overline{\mathbf{E}}^{4(i-1)+j}$ is nonzero. Using (14), the 3×9 matrix $\overline{\mathbf{E}}^{4(i-1)+j}$ can be calculated: see appendix C.

Then, for each $3n \times 9$ matrix $\mathbf{E}^i$, calculate $\mathbf{C}^i$ as

$$\mathbf{C}^i = \mathbf{U}^T \mathbf{E}^i \mathbf{V} \tag{19}$$

Substituting (19) and (17) into (15), the transformed error matrix $\mathbf{C}$ is represented as a $3n \times 9$ random matrix:

$$\mathbf{C} = \sum_{i=1}^{n} \left(\varepsilon_{i,1} \mathbf{C}^{4i-3} + \varepsilon_{i,2} \mathbf{C}^{4i-2} + \varepsilon'_{i,1} \mathbf{C}^{4i-1} + \varepsilon'_{i,2} \mathbf{C}^{4i} \right) \tag{20}$$

Consequently, the second term in (16) can be computed. Specifically, take the 9×1 vector ξ_j

$$\xi_j = -\sum_{i=1}^{8} \frac{c^j_{i,9} \mathbf{v}_i}{s_i} \tag{21}$$

as the j^{th} column of $\Xi_{\mathbf{h}_k} = \begin{bmatrix} \xi_1 & \xi_2 & \dots & \xi_{4n} \end{bmatrix}$.

Although the analysis above looks complicated, the computation can be greatly reduced by taking into consideration of these two facts: From (21), only the first 8 entries of the 9^{th} column of $\mathbf{C}^j$ are needed, and each $\mathbf{E}^j$ has only one nonzero 3×9 block in (18).

As said above, the aim of this statistical analysis is to represent the error in the homography parameters in terms of the random variables of $\{\varepsilon_{i,1}, \varepsilon_{i,2}, \varepsilon'_{i,1}, \varepsilon'_{i,2}\}$ for $(1 \leq i, j \leq n)$:

$$\Delta(\mathbf{h}_k) = \Xi_{\mathbf{h}_k} \mathbf{e} \tag{22}$$

where $\mathbf{e} = \begin{bmatrix} \varepsilon_{1,1} & \varepsilon_{1,2} & \varepsilon'_{1,1} & \varepsilon'_{1,2} & \dots & \varepsilon_{n,1} & \varepsilon_{n,2} & \varepsilon'_{n,1} & \varepsilon'_{n,2} \end{bmatrix}^T_{4n \times 1}$.

of the matrix $\Xi_{\mathbf{h}_k}$. Thus, The error covariance matrix is

$$\mathbf{C}_{\mathbf{h}_k} = \Xi_{\mathbf{h}_k} \Pi \Xi^T_{\mathbf{h}_k} \tag{23}$$

where Π is the $4n \times 4n$ covariance matrix for the noise $\mathbf{e}$ in the image points. This fact will be employed in the statistical analysis of the normalized DLT algorithm. In the special case, where i.i.d. 0-mean-σ^2-variance Gaussian (feature point) noise is assumed, the error covariance matrix in the homography is

$$\mathbf{C}_{\mathbf{h}_k} = \sigma^2 \Xi_{\mathbf{h}_k} \Xi^T_{\mathbf{h}_k} \tag{24}$$

Note that the calculation above of $\mathbf{C}^i = \mathbf{U}^T \mathbf{E}^i \mathbf{V}$ has to be computed $4n$ times, for $(1 \leq i \leq 4n)$. By a considerable abuse of notation that simplifies the exposition, such an operation will be denoted as $\mathbf{C} = \mathbf{U}^T \mathbf{E} \mathbf{V}$, even though $\mathbf{U}^T$ and $\mathbf{V}$ are respectively $3n \times 3n$ and 9×9 matrices while $\mathbf{E}$ is a stack of $4n$ $3n \times 9$ matrices.

5.1.1 The effect of the normalization step

In this subsection, we analyze the effect of the normalization step on the calculated homography. Here, different from the standard DLT definition, the normalization step is to scale the homography coefficients so that its Frobenius norm is 1,

$$\Delta(\frac{\mathbf{h}_k}{||\mathbf{h}_k||_F}) = \frac{\Delta(\mathbf{h}_k)}{||\mathbf{h}_k||_F} + \mathbf{h}_k \Delta(\frac{1}{||\mathbf{h}_k||_F}) \tag{25}$$

where $\Delta(\mathbf{h}_k)$ is defined in (22), $\Delta(\frac{1}{||\mathbf{h}_k||_F}) = -\frac{1}{||\mathbf{h}_k||_F^3} \sum_{i=1}^{9} h_i \Delta(h_i)$, h_i is the i^{th} component of $\mathbf{h}_k$ and $\Delta(h_i)$ is the i^{th} row of $\Delta \mathbf{h}_k$.

5.1.2 Replacing noise free data

When presenting a statistical analysis of the error in the homography above, we assumed that the noise free feature points are available. We now examine this assumption. From (16) and (21), each column ξ_k is a linear combination of $\{\mathbf{v}_i | i < 9\}$. This means that the matrix $\Xi_{\mathbf{h}_k}$ lies in the subspace spanned by these 8 vectors. Consequently, the covariance matrix $\mathbf{C}_{\mathbf{h}_k}$ in (24) and (23) has a zero singlar value and the associated singlar vector is the ground truth homography.

In practice, we do not have this knowledge of the ground truth data. An obvious solution, as adopted in this paper, is instead to use the noisy data (actually observed) instead. In assessing the impact of this approximation, we use the following measures to describe the differences:

$$\frac{\left| \left(\tilde{\mathbf{h}}_k - \mathbf{h}_k\right)^T \mathbf{u}_i - \left(\tilde{\mathbf{h}}_k - \mathbf{h}_k\right)^T \tilde{\mathbf{u}}_i \right|}{\left| \left(\tilde{\mathbf{h}}_k - \mathbf{h}_k\right)^T \mathbf{u}_i \right|} \tag{26}$$

where $\tilde{\mathbf{h}}_k$ and $\mathbf{h}_k$ denote the homographies, calculated from noisy data and noise free data, respectively, and $\tilde{\mathbf{u}}_i$ and $\mathbf{u}_i$ are the singular vectors of the covariance matrices $\mathbf{C}_{\tilde{\mathbf{h}}_k}$ and $\mathbf{C}_{\mathbf{h}_k}$ also from noisy data and noise free data, respectively. (26) measures the differences of the error's projections upon the directions $\tilde{\mathbf{u}}_i$ and $\mathbf{u}_i$, i.e., the effect of the replacement of noisy data for noise free data. Experiments show that, for $i < 9$, the above measure is less than 0.01. This means that the difference introduced by this replacement of noisy data for noise free data can be overlooked.

It is quite another matter when one considers the 9^{th} direction $\tilde{\mathbf{u}}_9$ of $\mathbf{C}_{\tilde{\mathbf{h}}_k}$. From the above calculations, it can be seen that, even with the noisy data, $\mathbf{C}_{\tilde{\mathbf{h}}_k}$ still has a rank of 8, and its null vector is the calculated homographyy. Were the ground truth feature points available, $\tilde{\mathbf{u}}_9$ can be expressed as

$$\tilde{\mathbf{u}}_9 = \mathbf{u}_9 + \sum_{j=1}^{8} \lambda_j \mathbf{u}_j \tag{27}$$

In practice, because of $||\tilde{\mathbf{u}}_9||_F = 1$,

$$\tilde{\mathbf{u}}_9 = \frac{\mathbf{u}_9 + \sum_{j=1}^{8} \lambda_j \mathbf{u}_j}{||\mathbf{u}_9 + \sum_{j=1}^{8} \lambda_j \mathbf{u}_j||_F} = \frac{\mathbf{u}_9 + \sum_{j=1}^{8} \lambda_j \mathbf{u}_j}{\sqrt{1 + \sum_{j=1}^{8} \lambda_j^2}} \approx \left(1 - \frac{1}{2}\sum_{j=1}^{8} \lambda_j^2\right)\left(\mathbf{u}_9 + \sum_{j=1}^{8} \lambda_j \mathbf{u}_j\right)$$

So the projecion of the error upon the $\tilde{\mathbf{u}}_9$ direction is

$$(\tilde{\mathbf{u}}_9 - \mathbf{u}_9)^T \tilde{\mathbf{u}}_9 = c - 3c^2 + 2c^3 \tag{28}$$

where

$$c = \frac{1}{2}\sum_{j=1}^{8} \lambda_j^2 \tag{29}$$

From (27), $\lambda_j = \tilde{\mathbf{u}}_9^T \mathbf{u}_j = (\tilde{\mathbf{u}}_9 - \mathbf{u}_9)^T \mathbf{u}_j$ for $j < 9$. It is clear that λ_j is the error's projection upon $\mathbf{u}_j$, i.e., approximately the error's projection upon $\tilde{\mathbf{u}}_j$, because the difference between these two directions, measured by (26), can be ingored. (Note that $\tilde{\mathbf{h}}_k$ and $\mathbf{h}_k$ are $\tilde{\mathbf{u}}_9$ and $\mathbf{u}_9$, respectively). Since $\lambda_j << 1$, the second order and third order terms in (28) can also be ignored.

To first-order perturbation, λ_j is indeed a 0-mean Gaussian random variable, with its variance as the j^{th} largest singlar value of $\mathbf{C}_{\mathbf{h}_k}$: σ_j^2. Thus, c in (29) is a chi-square-like random variable: its expectation is $E(c) = \frac{1}{2}\sum_{j=1}^{8} \sigma_j^2$ and its variance is $var(c) = \frac{1}{2}\sum_{j=1}^{8} \sigma_j^4$.

In order to account for error c in the $\tilde{\mathbf{u}}_9$ direction, we scale the normalized homography upto a factor of $1 - c$ and set the 9^{th} singular value of $\mathbf{C}_{\tilde{\mathbf{h}}_k}$ as $var(c)$.

5.2 Extension to the normalized DLT algorithm

In the normalized DLT algorithm (13), two factors have to be considered : First, $\mathbf{T}$ and $\mathbf{T}'$ depend upon the measurements and so are random matrices; Second, the noise in the normalized matches will not be i.i.d.

Note that, as in section 1, we have use the symbols $\mathbf{h}_k$ and $\mathbf{H}_k$, for the matrix and vector notations of a homography.

From (13),

$$\Delta(\mathbf{H}_k) = \Delta(\mathbf{T}'^{-1})\widetilde{\mathbf{H}}_k\mathbf{T} + \mathbf{T}'^{-1}\Delta(\widetilde{\mathbf{H}}_k)\mathbf{T} + \mathbf{T}'^{-1}\widetilde{\mathbf{H}}_k\Delta(\mathbf{T}) = \Xi_{\mathbf{H}_k}\mathbf{e} \tag{30}$$

In (30), all $\Delta(\bullet)$ are 3×3 random matrices, which can be represented by a stack of $4n$ 3×3 matrices.

5.2.1 Calculation of $\Delta(\widetilde{\mathbf{H}}_k)$

The critical point to calculate $\Delta(\widetilde{\mathbf{H}}_k)$ in (30) is to analyze the random variable of the inverse of the scale, as $\frac{1}{l}$ in (12); where $l = \sqrt{\frac{\sum_{i=1}^{n}(x_{i,1}-\overline{x}_1)^2+(x_{i,2}-\overline{x}_2)^2}{2n}}$, $\overline{x}_1 = \frac{\sum_{i=1}^{n} x_{i,1}}{n}$ and $\overline{x}_2 = \frac{\sum_{i=1}^{n} x_{i,2}}{n}$. Define $\underline{x}_{i,\bullet}$ as the centered coordinates: $\underline{x}_{i,1} = x_{i,1} - \overline{x}_1$ and $\underline{x}_{i,2} = x_{i,2} - \overline{x}_2$. Due to the noise in the feature points, the error in the centered coordinates is

$$\Delta\left(\underline{x}_{i,1}\right) = \begin{bmatrix} -1/n & \ldots & -1/n & (n-1)/n & -1/n & \ldots & -1/n \end{bmatrix} \begin{bmatrix} \varepsilon_{1,1} & \ldots & \varepsilon_{n,1} \end{bmatrix}^T$$

$$\Delta\left(\underline{x}_{i,2}\right) = \left[\begin{array}{ccccccc} -1/n & \ldots & -1/n & (n-1)/n & -1/n & \ldots & -1/n \end{array}\right] \left[\begin{array}{ccc} \varepsilon_{1,2} & \ldots & \varepsilon_{n,2} \end{array}\right]^T$$

where the $(n-1)/n$ are the i^{th} components in the vectors. Thus, the error in the inverse of l is:

$$\Delta\left(\frac{1}{l}\right) = -\frac{1}{2nl^3}\sum_{i=1}^{n}\left[\underline{x}_{i,1}\Delta\left(\underline{x}_{i,1}\right) + \underline{x}_{i,2}\Delta\left(\underline{x}_{i,2}\right)\right] \tag{31}$$

The normalized image feature is $\frac{\underline{x}_{i,\bullet}}{l}$. The error in it is $\Delta\left(\frac{\underline{x}_{i,\bullet}}{l}\right) = \underline{x}_{i,\bullet}\Delta\left(\frac{1}{l}\right) + \frac{\Delta\left(\underline{x}_{i,\bullet}\right)}{l}$, which can be expressed in $\mathbf{p}_{i,\bullet}^T\mathbf{e}$. Similarly, the error in the second normalized view is $\mathbf{p}_{i,\bullet}'^T\mathbf{e}$. We stack the vectors $\mathbf{p}^T$ and $\mathbf{p}'^T$ as

$$\mathbf{P} = \left[\begin{array}{ccccccccc} \mathbf{p}_{i,1} & \mathbf{p}_{i,2} & \mathbf{p}'_{1,1} & \mathbf{p}'_{1,2} & \cdots & \mathbf{p}_{n,1} & \mathbf{p}_{n,2} & \mathbf{p}'_{n,1} & \mathbf{p}'_{n,1} \end{array}\right]^T$$

$\mathbf{Pe}$ is the error in the normalized coordinates. According to (22),

$$\Delta(\widetilde{\mathbf{H}}_k) = \Xi\mathbf{Pe} = \Xi_{\widetilde{\mathbf{H}}_k}\mathbf{e} \tag{32}$$

where Ξ is calculated as in $\Xi_{\mathbf{h}}$ in (22), however, Ξ is arranged as a stack of $4n$ 3×3 matrices.

5.2.2 Calculation of $\Delta(\mathbf{T})$

Another quantity will be used in calculating $\Delta(\mathbf{T})$ and $\Delta(\mathbf{T}'^{-1})$ is $\Delta(c_i)$, the error in the centroid of the original feature points.

$$\Delta c_\bullet = \frac{1}{n}\sum_{i=1}^{n}\varepsilon_{i,\bullet} \tag{33}$$

From (12),

$$\Delta(\mathbf{T}) = \left[\begin{array}{ccc} \Delta(\frac{1}{l}) & 0 & -c_1\Delta(\frac{1}{l}) - \frac{\Delta(c_1)}{l} \\ 0 & \Delta(\frac{1}{l}) & -c_2\Delta(\frac{1}{l}) - \frac{\Delta(c_2)}{l} \\ 0 & 0 & 0 \end{array}\right] \tag{34}$$

Substituting (31) and (33) into (34), we can calculate $\Delta\mathbf{T}$.

From the first order approximation, $(\mathbf{T}+\Delta\mathbf{T})^{-1} = \mathbf{T}^{-1} - \mathbf{T}^{-1}\Delta\mathbf{T}\mathbf{T}^{-1}$

$$\Delta(\mathbf{T}'^{-1}) = -\mathbf{T}'^{-1}\Delta(\mathbf{T}')\mathbf{T}'^{-1} \tag{35}$$

where $\Delta(\mathbf{T}')$ can be calculated as in (34).

Substituting (32), (34) and (35) into (30), we obtain $\Delta(\mathbf{H}_k)$ and the covariance matrix of $\mathbf{h}_k$: $\Xi_{\mathbf{h}_k}$.

5.3 Noise level estimation

Now, we have represented the errors in the homography parameters in terms of random variables, i.e. the noise in the image feature points. For i.i.d. Gaussian noise or general noise in the feature points, the covariance matrix for the error in the parameters, can be obtained by (24) or (23). To do so, we need to know some statistical properties of the noise in the image points. Here, we consider the simplest case, where the noise in the image points is i.i.d. 0-mean-σ^2-variance Gaussian noise. In this case, the covariance matrix in (24) is $\mathbf{C} = \sigma^2 \Xi\Xi^T$. In some cases, the covariance matrix, up to a scale, suffices, i.e., we can take $\mathbf{C} = \Xi\Xi^T$ as the covariance matrix. Although the noise level σ is not needed in ***Algorithm*** and ***Core algorithm*** of [4], we still present how to estimate the noise level, here. The estimate is very accurate in our experiment, although we do not provide experimental results about this in this paper. The major tool, as in the sections above, is also the first order approximation.

Due to the noise in the image points, there exists difference between the projection $\mathbf{H}_k\mathbf{x}$ and $\mathbf{x}'$. In this part, we will show that the projection error can also be represented as a random variable, which depends on the noise in the images: $\{\varepsilon_{i,1}, \varepsilon_{i,2}, \varepsilon'_{i,1}, \varepsilon'_{i,2}\}$ for $(1 \le i, j \le n)$.

Suppose that the noise free homography is $\mathbf{H}_k = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_4 & h_5 \\ h_7 & h_8 & h_9 \end{pmatrix}$. $\mathbf{H}$ projects each point $\{x_{i,1}, x_{i,2}\}$ in the first view upon the second view as, by taking $x'_{i,1}$ as an example:

$$x'_{i,1} = \frac{h_1 * x_{i,1} + h_2 * x_{i,2} + h_3}{h_7 * x_{i,1} + h_8 * x_{i,2} + h_9} \tag{36}$$

Due to the noise, the projection upon the second view is

$$\frac{[h_1 + \Delta(h_1)] * (x_{i,1} + \varepsilon_{i,1}) + [h_2 + \Delta(h_2)] * (x_{i,2} + \varepsilon_{i,2}) + h_3 + \Delta(h_3)}{[h_7 + \Delta(h_7)] * (x_{i,1} + \varepsilon_{i,1}) + [h_8 + \Delta(h_8)] * (x_{i,2} + \varepsilon_{i,2}) + h_9 + \Delta(h_9)} \tag{37}$$

According to the first order approximation, $\frac{a+\Delta a}{b+\Delta b} = \frac{a}{b} + \frac{\Delta a}{b} - \frac{a\Delta b}{b^2}$ approximately holds. From this, (37) equals to

$$x'_{i,1} + \frac{A}{E} - \frac{BD}{E^2} \tag{38}$$

where $A = h_1\varepsilon_{i,1} + h_2\varepsilon_{i,2} + x_{i,1}\Delta(h_1) + x_{i,2}\Delta(h_2) + \Delta(h_3)$, $D = h_1 * x_{i,1} + h_2 * x_{i,2} + h_3$, $B = h_7\varepsilon_{i,1} + h_8\varepsilon_{i,2} + x_{i,1}\Delta(h_7) + x_{i,2}\Delta(h_8) + \Delta(h_9)$, and $E = h_7 * x_{i,1} + h_8 * x_{i,2} + h_9$. Note that the second order terms, like $\Delta(h_\bullet)\varepsilon_{i,\circ}$, have been dropped. Including the noise in the observed $x'_{i,1}$, the projection error is actually $\frac{A}{E} - \frac{BD}{E^2} - \varepsilon'_{i,1}$. It can be represented as $\mathbf{q}_{i,1}^T\mathbf{e}$. Similarly, from the projection of the second coordinate, $\mathbf{q}_{i,2}^T\mathbf{e}$ can be obtained. Stack $\mathbf{q}_{i,\bullet}^T$ as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q}_{1,1} & \mathbf{q}_{1,2} & \cdots & \mathbf{q}_{n,1} & \mathbf{q}_{n,2} \end{bmatrix}^T \tag{39}$$

In practice, the projection error is actually available, as ς. Then, $\mathbf{Q}\mathbf{e} = \varsigma$ approximately holds. Because the i.i.d. Gaussian noise is assumed here, $||\mathbf{q}_{i,\bullet}||_F^2\sigma^2 = \varsigma^2_{2(i-1)+\bullet}$. Then, the noise level can be estimated as:

$$\hat{\sigma} = \frac{||\varsigma||_F}{||\mathbf{Q}||_F} \tag{40}$$

Note, the noise levels in two views can be assumed as different, up to a scale. Suppose σ_1 and σ_2 are the noise level in the 1^{st} and 2^{nd} views, respectively. Suppose, further, $\sigma_1 = \lambda\sigma_2$. Then, by multiplying the $(4 \bullet +1)^{th}$ and $(4 \bullet +2)^{th}$ columns of $\mathbf{Q}$ by a factor of λ, we can calculate $\hat{\sigma}_2$, according to (40). Consequently, $\hat{\sigma}_1 = \lambda\hat{\sigma}_2$.

5.4 Error in the matrix M in (8)

As said in the above, the error in the matrix $\mathbf{M}$ in (8) cannot be considered as i.i.d. Gaussian, due to 2 reasons: the heteroscedastic error, introduced in the calculation of $\mathbf{M}$; and the heteroscedastic error in the homography parameters.

As in the analysis above, we will track how the errors in the homography parameters propagate in the matrix $\mathbf{M}$. First, assume the error in each homography $\mathbf{h}_i$ is $\eta_i = \begin{bmatrix} \eta_{i,1} & \dots & \eta_{i,9} \end{bmatrix}^T$. Define the errors in the n homographies as η: $\eta = \begin{bmatrix} \eta_1^T & \dots & \eta_n^T \end{bmatrix}^T$. Only one assumption for these errors is made: the errors in each homography are of zero mean and independent from those in another homography, as is reasonable in the problem in this paper.

Obviously, the errors in the matrix $\mathbf{M}$ cannot be considered as row independent or column independent. Thus, we have to arrange the $3C_2^n \times 6$ matrix $\mathbf{M}$ as a $18C_2^n \times 1$ vector: $vec(\mathbf{M}^T)$. Correspondingly, by tracking the error in the homographies, the error in this vector, denoted by ζ, is:

$$\zeta = \Phi\eta \tag{41}$$

where the quadratic order errors has been dropped.

Φ is an $18C_n^2 \times 9n$ matrix, with $\Phi_{i,j}$ as an $18 \times 9n$ matrix:

$$\Phi = \begin{bmatrix} \Phi_{1,2}^T & \dots & \Phi_{1,n}^T & \Phi_{2,3}^T & \dots & \Phi_{2,n}^T & \dots & \Phi_{n-1,n}^T \end{bmatrix}^T \tag{42}$$

The definition of $\Phi_{i,j}$ in (42) will be given in the appendix.

From (41), the covariance matrix of ζ is

$$\mathbf{C}_\zeta = \Phi E(\eta\eta^T)\Phi^T \tag{43}$$

The covariance matrix $E(\eta_i\eta_i^T)$ have been studied in sections 5.1 and 5.2: It can be calculated from the feature points while calculating the homography. $E(\eta\eta^T)$ is a block diagonal matrix: $E(\eta\eta^T) = diag\left\{E(\eta_i\eta_i^T)\right\}$ because η_i is independent of η_j for $i \neq j$.

6 Conclusion

In this paper, we show how to analytically compute the statistical property of the error in the homography parameters and an associated matrix required for the algorithms in [4]. To the best of our knowledge, no similar work has been done to analyze the statical property of the error in the estimated parameters. This techniques of our work is potentially useful in many problems, where the estimated parameters will be used as the input for further analysis. A direct application is to employ the same techniques in the calculation of the induced dimension-4 homography subspace in the cases of 2-plane-over-multiple-plane or multiple-plane-over-multiple-plane [14, 7].

A Definition of weighted rank-r approximation matrix

We suppose 0-mean noise in the entries of the matrix $\mathbf{M} \in R^{m,n}$ but we do not assume row or column independence. In order to characterize the noise in $\mathbf{M}$, we first rearrange $\mathbf{M}$ as a vector $vec(\mathbf{M}) \in R^{mn,1}$. Suppose the covariance matrix for the noise in $vec(\mathbf{M})$ is $\mathbf{C}$. The weighted rank-r approximation matrix of $\mathbf{M}$ is defined to be $\mathbf{M}^r$ that has properties: $rank\,(\mathbf{M}^r) = r$ and $\mathbf{M}^r$ minimizes the objective function of $(vec\,(\mathbf{M}-\mathbf{X}))^T\,\mathbf{C}^{-}vec\,(\mathbf{M}-\mathbf{X})$. Methods for finding rank r approximation matrix can be found, as the Bilinear approach in [1, 2] and the AP approach in [10].

B Definition of the matrix M in (7) in section 3.2.1

$$\mathbf{M}_{i,j} = \begin{bmatrix} h_{i,1}h_{j,2}-h_{j,1}h_{i,2} & h_{i,4}h_{j,5}-h_{j,4}h_{i,5} & h_{i,7}h_{j,8}-h_{j,7}h_{i,8} & M_{i,j}^{1,4} & M_{i,j}^{1,5} & M_{i,j}^{1,6} \\ h_{i,1}h_{j,3}-h_{j,1}h_{i,3} & h_{i,4}h_{j,6}-h_{j,4}h_{i,6} & h_{i,7}h_{j,9}-h_{j,7}h_{i,9} & M_{i,j}^{2,4} & M_{i,j}^{2,5} & M_{i,j}^{2,6} \\ h_{i,2}h_{j,3}-h_{j,2}h_{i,3} & h_{i,5}h_{j,6}-h_{j,5}h_{i,6} & h_{i,8}h_{j,9}-h_{j,8}h_{i,9} & M_{i,j}^{3,4} & M_{i,j}^{3,5} & M_{i,j}^{3,6} \end{bmatrix}$$

with $M_{i,j}^{1,4} = h_{i,1}h_{j,5}-h_{j,1}h_{i,5}+h_{i,4}h_{j,2}-h_{j,4}h_{i,2}$, $M_{i,j}^{1,5} = h_{i,1}h_{j,8}-h_{j,1}h_{i,8}+h_{i,7}h_{j,2}-h_{j,7}h_{i,2}$, $M_{i,j}^{1,6} = h_{i,4}h_{j,8}-h_{j,4}h_{i,8}+h_{i,7}h_{j,5}-h_{j,7}h_{i,5}$, $M_{i,j}^{2,4} = h_{i,1}h_{j,6}-h_{j,1}h_{i,6}+h_{i,4}h_{j,3}-h_{j,4}h_{i,3}$, $M_{i,j}^{2,5} = h_{i,1}h_{j,9}-h_{j,1}h_{i,9}+h_{i,7}h_{j,3}-h_{j,7}h_{i,3}$, $M_{i,j}^{2,6} = h_{i,4}h_{j,9}-h_{j,4}h_{i,9}+h_{i,7}h_{j,6}-h_{j,7}h_{i,6}$, $M_{i,j}^{3,4} = h_{i,2}h_{j,6}-h_{j,2}h_{i,6}+h_{i,5}h_{j,3}-h_{j,5}h_{i,3}$, $M_{i,j}^{3,5} = h_{i,2}h_{j,9}-h_{j,2}h_{i,9}+h_{i,8}h_{j,3}-h_{j,8}h_{i,3}$, and $M_{i,j}^{3,6} = h_{5,i}h_{9,j}-h_{5,j}h_{9,i}+h_{8,i}h_{6,j}-h_{8,j}h_{6,i}$

C Definition of $\overline{\mathbf{E}}^{4(i-1)+k}$ in (18) in section 5.1

$$\overline{\mathbf{E}}^{4(i-1)+1} = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & x'_{i,2} & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -x'_{i,1} & 0 & 0 \\ -x'_{i,2} & 0 & 0 & x'_{i,1} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\overline{\mathbf{E}}^{4(i-1)+2} = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 & 0 & x'_{i,2} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -x'_{i,1} & 0 \\ 0 & -x'_{i,2} & 0 & 0 & x'_{i,1} & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\overline{\mathbf{E}}^{4(i-1)+3} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -x_{i,1} & -x_{i,2} & -1 \\ 0 & 0 & 0 & 0 & x_{i,1} & x_{i,2} & 1 & 0 & 0 \end{bmatrix}$$

$$\overline{\mathbf{E}}^{4(i-1)+4} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & x_{i,1} & x_{i,2} & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -x_{i,1} & -x_{i,2} & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

D Definition for $\Phi_{i,j}$ in (42) in section 5.4

$$\Phi_{i,j} = \begin{bmatrix} \mathbf{0}_{6,9(i-1)} & \phi_{i,j,1,2} & \mathbf{0}_{6,9(j-i-1)} & \phi_{i,j,1,4} & \mathbf{0}_{6,9(n-j)} \\ \mathbf{0}_{6,9(i-1)} & \phi_{i,j,2,2} & \mathbf{0}_{6,9(j-i-1)} & \phi_{i,j,2,4} & \mathbf{0}_{6,9(n-j)} \\ \mathbf{0}_{6,9(i-1)} & \phi_{i,j,3,2} & \mathbf{0}_{6,9(j-i-1)} & \phi_{i,j,3,4} & \mathbf{0}_{6,9(n-j)} \end{bmatrix}$$

where

$$\phi_{i,j,1,2} = \begin{bmatrix} h_{j,2} & -h_{j,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_{j,5} & -h_{j,4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_{j,8} & -h_{j,7} & 0 \\ h_{j,5} & -h_{j,4} & 0 & h_{j,2} & -h_{j,1} & 0 & 0 & 0 & 0 \\ h_{j,8} & -h_{j,7} & 0 & 0 & 0 & 0 & h_{j,2} & -h_{j,1} & 0 \\ 0 & 0 & 0 & h_{j,8} & -h_{j,7} & 0 & h_{j,5} & -h_{j,4} & 0 \end{bmatrix}$$

$$\phi_{i,j,1,4} = \begin{bmatrix} -h_{i,2} & h_{i,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -h_{i,5} & h_{i,4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -h_{i,8} & h_{i,7} & 0 \\ -h_{i,5} & h_{i,4} & 0 & -h_{i,2} & h_{i,1} & 0 & 0 & 0 & 0 \\ -h_{i,8} & h_{i,7} & 0 & 0 & 0 & 0 & -h_{i,2} & h_{i,1} & 0 \\ 0 & 0 & 0 & -h_{i,8} & h_{i,7} & 0 & -h_{i,5} & h_{i,4} & 0 \end{bmatrix}$$

$$\phi_{i,j,2,2} = \begin{bmatrix} h_{j,3} & 0 & -h_{j,1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & h_{j,6} & 0 & -h_{j,4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & h_{j,9} & 0 & -h_{j,7} \\ h_{j,6} & 0 & -h_{j,4} & h_{j,3} & 0 & -h_{j,1} & 0 & 0 & 0 \\ h_{j,9} & 0 & -h_{j,7} & 0 & 0 & 0 & h_{j,3} & 0 & -h_{j,1} \\ 0 & 0 & 0 & h_{j,9} & 0 & -h_{j,7} & h_{j,6} & 0 & -h_{j,4} \end{bmatrix}$$

$$\phi_{i,j,2,4} = \begin{bmatrix} -h_{i,3} & 0 & h_{i,1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -h_{i,6} & 0 & h_{i,4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -h_{i,9} & 0 & h_{i,7} \\ -h_{i,6} & 0 & h_{i,4} & -h_{i,3} & 0 & h_{i,1} & 0 & 0 & 0 \\ -h_{i,9} & 0 & h_{i,7} & 0 & 0 & 0 & -h_{i,3} & 0 & h_{i,1} \\ 0 & 0 & 0 & -h_{i,9} & 0 & h_{i,7} & -h_{i,6} & 0 & h_{i,4} \end{bmatrix}$$

$$\phi_{i,j,3,2} = \begin{bmatrix} 0 & h_{j,3} & -h_{j,2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_{j,6} & -h_{j,5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_{j,9} & -h_{j,8} \\ 0 & h_{j,6} & -h_{j,5} & 0 & h_{j,3} & -h_{j,2} & 0 & 0 & 0 \\ 0 & h_{j,9} & -h_{j,8} & 0 & 0 & 0 & 0 & h_{j,3} & -h_{j,2} \\ 0 & 0 & 0 & 0 & h_{j,9} & -h_{j,8} & 0 & h_{j,6} & -h_{j,5} \end{bmatrix}$$

$$\phi_{i,j,3,4} = \begin{bmatrix} 0 & -h_{i,3} & h_{i,2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -h_{i,6} & h_{i,5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -h_{i,9} & h_{i,8} \\ 0 & -h_{i,6} & h_{i,5} & 0 & -h_{i,3} & h_{i,2} & 0 & 0 & 0 \\ 0 & -h_{i,9} & h_{i,8} & 0 & 0 & 0 & 0 & -h_{i,3} & h_{i,2} \\ 0 & 0 & 0 & 0 & -h_{i,9} & h_{i,8} & 0 & -h_{i,6} & h_{i,5} \end{bmatrix}$$

References

[1] P. Chen. *An investigation of statistical aspects of linear subspace analysis for computer vision applications*. PhD thesis, Monash University, 2004.

[2] P. Chen and D. Suter. A bilinear approach to the parameter estimation of a general heteroscedastic linear system, with application to conic fitting. *Submitted to J. Math. Img. Vis.*, 2004.

[3] P. Chen and D. Suter. An analysis of linear subspace approaches for computer vision and pattern recognition. *Accepted by Int'l J. Computer Vision*, 2005.

[4] P. Chen and D. Suter. Jointly estimating multiple homographies over 2 views: Revisiting the rank 4 constraint in homographies over 2 views. *in preparation*, 2005.

[5] G. H. Golub and C. F. V. Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 3nd edition, 1996.

[6] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2000.

[7] L. Zelnik-Manorand M. Irani. Multi-view subspace constraints on homographies. *IEEE Trans Pattern Analysis and Machine Intelligence*, 24(2):214–223, 2002.

[8] Y. Leedan and P. Meer. Estimation with bilinear constraints in computer vision. In *Proc. Int'l Conf. Computer Vision*, pages 733–738, 1999.

[9] Y. Leedan and P. Meer. Heteroscedastic regression in computer vision: Problems with bilinear constraint. *Int'l J. Computer vision*, 37(2):127–150, 2000.

[10] J. H. Manton, R. Mahony, and Y. Hua. The geometry of weighted low-rank approximations. *IEEE Trans Signal Processing*, 51(2):500–514, 2003.

[11] A. Shashua and S. Avidan. The rank 4 constraint in multiple (> 2) view geometry. In *Proc. European Conf. Computer Vision*, pages 196–206, 1996.

[12] G. W. Stewart and J. G. Sun. *Matrix perturbation theory*. Academic press, 1990.

[13] J. H. Wilkinson. *The algebraic eigenvalue problem*. Clarendon Press, Oxford, 1965.

[14] L. Zelnik-Manor and M. Irani. Multi-view subspace constraints on homographies. In *Proc. Int'l Conf. Computer Vision*, pages 710–715, 1999.