TR-H-284 025

Error Description of Projectively Reconstructed Point Sets.

Martin TONKO and Keisuke KINOSHITA

2000.1.26

ATR人間情報通信研究所
〒619-0288 京都府相楽郡精華町光台2-2 TEL: 0774-95-1011

ATR Human Information Processing Research Laboratories
2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan
Telephone: +81-774-95-1011
Fax : +81-774-95-1008

Error Description of Projectively Reconstructed Point Sets

M. Tonko, K. Kinoshita

ATR Human Information Processing Research Laboratories,
2-2 Hikari-dai Seika-cho Soraku-gun Kyoto-fu 619-0288, Japan

Tel: +81 (0)774-95-{1086, 1075}, Fax: +81 (0)774-95-1008,
Email: {mtonko, kino}@hip.atr.co.jp

Abstract

In this report, error propagation is derived for an existing sequence of equations, which allows to projectively reconstruct a 3D point set solely using given images of this point set. As input, image feature points, their position uncertainty, and the correspondence information must be given. All image feature points are assumed to be stochastically independent. The covariance tensor calculus is used to propagate the uncertainty of the input into an uncertainty of the output, which consists of projection matrices and 3D reconstructed points. In the course of error propagation *all* stochastic dependencies are modelled, resulting in an accurate description of the reconstruction error. This fact is experimentally validated. Furthermore, experiments with real image data are given.

1 Introduction

[Tonko & Kinoshita 99] discuss a 3D scanning system prototype which takes several images of a rigid textured object, establishes 2D point correspondences across the sequence of images and calculates a 3D Euclidian structure that is constrained by the 2D point correspondences. Principally the system is shown to work, but judging from the experimental results, one can see that there is room for improvements such as:

- Euclidan reconstruction is subject to reconstruction errors. The error distribution of each reconstructed 3D point should be estimated as precisely as possible.
- Error distributions should subsequently be used by an error reduction method.
- A reconstructed 3D point set describes the structure of an object, but is not visually appealling to the system operator. A method has to be applied which – based on the sequence of images and the relation of each 3D point with its 2D projections – renders image content on the 3D point set in order to achieve a realistic 3D impression of the object.

This report mainly deals with the first item, i.e. the unbiased estimation of the projective geometry and its uncertainty from point correspondences and point uncertainties. Readers interested in the second and third item are referred to [Kinoshita & Tonko 99] and [Doyon 99], respectively.

Once 2D point correspondences are established over the sequence of images, Euclidian reconstruction is accomplished by estimating

1. the two-dimensional epipolar,
2. the three-dimensional projective, and
3. the three-dimensional Euclidian geometries.

The analysis of Section 3 is used in Section 4 to Section 7 to quantify reconstruction error up to the projective level under the assumption of known error distributions for all 2D point features. Section 8 gives experimental results on the analysis.

2 Related Work

[Georgis et al. 98] derive uncertainty descriptions of projectively reconstructed points. They restrict their error analysis to a special scenario with two reference planes and four markers per plane that must be in the field of view of each camera. Their approach makes thus use of specific scene knowledge. [Csurka et al. 97] remark that – once the uncertainty of the fundamental matrix is estimated – uncertainty descriptions for projectively reconstructed points are obtainable, they show results, but do not give equations. [Kanatani 96] gives uncertainty descriptions for projectively reconstructed points. He bases his work on the knowledge of the essential matrix, not the fundamental matrix. [Collins 92] restricts his analysis of the projective reconstruction error to approximately planar scenes as valid for the special case of aerial imagery. [Grimson et al. 92] discuss an uncertainty analysis of object pose, where the position of image points is assumed to be uncertain and, more important, a model of the object is given. [Ha & Kweon 99] – without calculating the uncertainty of the reconstructed point cloud – devise a method, which reduces reconstruction error by enforcing angular constraints that depend on object model knowledge. As the approach discussed in this report, [Ha & Kweon 99]'s approach is based on [Bougnoux 98]. [Sun et al. 99] derive an error characterization of the factorization approach to shape and motion recovery from images, which are taken with an uncalibrated affine camera. In the past the factorization approach has been extended to deal with weak-perspective and para-perspective projection. However, both types of projection are only approximations to perspective projection. Arguing that error propagation is only a first order approximation of the truth, [Matei & Meer 99] use the bootstrap method to evaluate 3D rigid motion more accurately.

[Singh 90] details an interesting approach that allows to automatically extract the uncertainty of a 2D point feature correspondence by relating the Laplacian of intensity to the Fisher Information matrix. Therefore point uncertainties do not need to be predefined, but can be algorithmically found for each particular feature correspondence.

3 Error Propagation

If errors in the input data are unavoidable, the general way to improve the situation is to model errors of the input data statistically and propagate this uncertainty into the output and its uncertainty. Doing this the results do not get any better, but at least a quantiative assessment of error is possible (see Fig. 1). [Kanatani 96] gives a wealth of information on how to do that in a systematical and sound way.

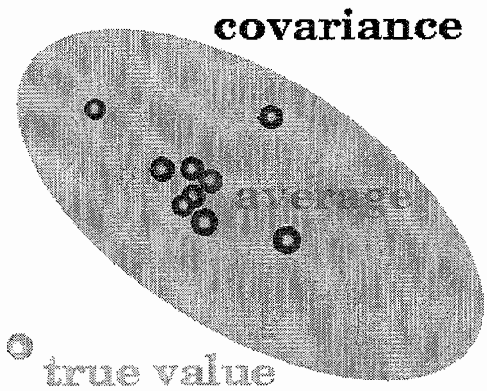

Figure 1: Error Description: Statistical modelling of data is to specify mean and covariance, i.e. error distribution. If mean and true value coincide, we say that the estimation is bias-free.

The notation of [Kanatani 96] is adopted, who uses bold lowercase letters $\boldsymbol{a} = (a_i)$, $i = 1, \ldots, n$ to describe n-vectors, bold uppercase letters $\boldsymbol{B} = (B_{ij})$, $i = 1, \ldots, m$, $j = 1, \ldots, n$ to describe mn-matrices, and calligraphic letters to denote tensors. For example, $\mathcal{C} = (C_{ijk})$, $i = 1, \ldots, m$, $j = 1, \ldots, n$, $k = 1, \ldots, p$ is an mnp-tensor.

In general, if the output can be calculated from the input using an analytic function, the following Theorem 1 can be applied to propagate the input error through the analytic function into an output error.

Theorem 1 (Error Propagation) *Given matrices $\boldsymbol{A_1}$, $\boldsymbol{A_2}$, ..., $\boldsymbol{A_n}$, and matrix $\boldsymbol{B}$ as a function of $\boldsymbol{A_1}$, ..., $\boldsymbol{A_n}$. Also the covariance tensors $\mathcal{V}[\boldsymbol{A}_i]$, $i = 1, \ldots, n$ are given. The covariance tensor $\mathcal{V}[\boldsymbol{B}]$ of $\boldsymbol{B}$ is approximated as*

$$\mathcal{V}[\boldsymbol{B}] = \sum_{i,j=1}^{n} \frac{\partial \boldsymbol{B}}{\partial \boldsymbol{A_i}} \, \mathcal{V}[\boldsymbol{A}_i, \boldsymbol{A}_j] \, \frac{\partial \boldsymbol{B}}{\partial \boldsymbol{A_j}}^T .$$

Unfortunately, not all functions are analytic and the question is how to proceed in that case, e.g. calculate the gradient of a non-analytic function. For the case of function *ker*, which computes the kernel of a matrix, a solution for the gradient can be found using the Perturbation Theorem 2 stated in [Kanatani 96].

Theorem 2 (Perturbation Theorem) *Let $\boldsymbol{A} = \boldsymbol{A}^T$ and $\boldsymbol{D}$ be square matrices. Further let $\{\lambda_i\}$ be the descendingly sorted eigenvalues and $\{\boldsymbol{u}_i\}$ the corresponding eigensystem of symmetric matrix $\boldsymbol{A}$:*

$$\boldsymbol{A}\ \boldsymbol{u}_i = \lambda_i\ \boldsymbol{u}_i, \quad \boldsymbol{u}_i^T\ \boldsymbol{u}_j = \delta_{ij} := \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} . \tag{1}$$

Consider a perturbed matrix $\boldsymbol{A}' = \boldsymbol{A} + \epsilon\boldsymbol{D}$ for a small ϵ. Let $\{\lambda_i'\}$ be the descendingly sorted Eigenvalues and $\{\boldsymbol{u}_i'\}$ the corresponding Eigensystem of $\boldsymbol{A}'$. Then the following approximations hold:

$$\lambda_i' \approx \lambda_i + \epsilon\ \boldsymbol{u}_i^T\ \boldsymbol{D}\boldsymbol{u}_i \tag{2}$$

$$\boldsymbol{u}_i' \approx \boldsymbol{u}_i + \epsilon \sum_{j \neq i} \frac{\boldsymbol{u}_j^T\ \boldsymbol{D}\boldsymbol{u}_i}{\lambda_i - \lambda_j}\ \boldsymbol{u}_j \tag{3}$$

Thus, it is possible to approximate noisy Eigenvalues and -vectors depending on the true Eigenvalues and -vectors plus the known error on the input matrix. Next, this is used to give an approximation of the gradient of function *ker*.

Theorem 3 *Let $\boldsymbol{A} = \boldsymbol{A}^T$ be a singular square matrix and $\boldsymbol{u}_n$ the eigenvector of $\boldsymbol{A}$, which corresponds to eigenvalue $\lambda_n = 0$. Then we can approximate:*

$$\frac{\partial \boldsymbol{u}_n}{\partial A_{ij}} \approx \sum_{k \neq n} \frac{(\boldsymbol{u}_k \boldsymbol{u}_n^T)_{ij}}{\lambda_k}\ \boldsymbol{u}_k \tag{4}$$

To proove this for element A_{ij} of $\boldsymbol{A}$, we define $\boldsymbol{A}' = \boldsymbol{A} + \epsilon\boldsymbol{D}$ with matrix $\boldsymbol{D}$ to be

$$D_{pq} = \begin{cases} 1, & p = i\ \&\ q = j \\ 0, & \text{otherwise} \end{cases} . \tag{5}$$

Since the Perturbation Theorem 2 is applicable, the following holds for small ϵ:

$$\boldsymbol{u}_n' \approx \boldsymbol{u}_n - \epsilon \sum_{k \neq n} \frac{\boldsymbol{u}_k^T\ \boldsymbol{D}\boldsymbol{u}_n}{\lambda_k}\ \boldsymbol{u}_k. \tag{6}$$

On the other hand $\frac{\partial \boldsymbol{u}_n}{\partial A_{ij}}$ is defined to be

$$\begin{aligned}
\frac{\partial \boldsymbol{u}_n}{\partial A_{ij}} &= \lim_{A'_{ij} \to A_{ij}} \frac{\boldsymbol{u}_n - \boldsymbol{u}'_n}{A_{ij} - A'_{ij}} \\
&\approx \lim_{\epsilon \to 0} \frac{\epsilon \sum_{k \neq n} \frac{\boldsymbol{u}_k^T \boldsymbol{D} \boldsymbol{u}_n}{\lambda_k} \boldsymbol{u}_k}{\epsilon} \\
&= \sum_{k \neq n} \frac{\boldsymbol{u}_k^T \boldsymbol{D} \boldsymbol{u}_n}{\lambda_k} \boldsymbol{u}_k = \sum_{k \neq n} \frac{(\boldsymbol{u}_k \boldsymbol{u}_n^T)_{ij}}{\lambda_k} \boldsymbol{u}_k . \qquad (7)
\end{aligned}$$

So far, these results are restricted to square matrices. Suppose that matrix $\boldsymbol{C}$ is not square. In this case we define matrix $\boldsymbol{A} = \boldsymbol{C}^T \cdot \boldsymbol{C}$ and to calculate $\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{C}}$, we multiply the result $\partial \boldsymbol{u}_n / \partial \boldsymbol{A}$ of Eq. (7) with

$$\begin{aligned}
\left(\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{C}}\right)_{ikmn} &= \frac{\partial A_{ik}}{\partial C_{mn}} = \frac{\partial \sum_j C_{ji} \cdot C_{jk}}{\partial C_{mn}} \\
&= \frac{\partial C_{mi} \cdot C_{mk}}{\partial C_{mn}} = \begin{cases} 0 , & i \neq n \;\&\; k \neq n \\ C_{mi} , & i \neq n \;\&\; k = n \\ C_{mk} , & i = n \;\&\; k \neq n \\ 2\, C_{mi}, & i = n \;\&\; k = n \end{cases} . \qquad (8)
\end{aligned}$$

Finally, we are able to express the gradient of function ker.

$$\begin{aligned}
\left(\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{C}}\right)_{ijk} &= \sum_{l,m} \left(\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{A}}\right)_{ilm} \left(\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{C}}\right)_{lmjk} \\
&= \sum_{m \neq k} \left(\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{A}}\right)_{ikm} \left(\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{C}}\right)_{kmjk} \\
&\quad + \sum_{l \neq k} \left(\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{A}}\right)_{ilk} \left(\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{C}}\right)_{lkjk} + \left(\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{A}}\right)_{ikk} \left(\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{C}}\right)_{kkjk} \\
&= \sum_{m \neq k} \left(\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{A}}\right)_{ikm} C_{jm} + \sum_{l \neq k} \left(\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{A}}\right)_{ilk} C_{jl} + \left(\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{A}}\right)_{ikk} C_{jk} \\
&= \sum_{m} \left(\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{A}}\right)_{ikm} C_{jm} + \sum_{l} \left(\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{A}}\right)_{ilk} C_{jl} \\
&= \sum_{m} \sum_{p \neq n} \frac{(\boldsymbol{u}_p \boldsymbol{u}_n^T)_{km}}{\lambda_p} (\boldsymbol{u}_p)_i \, C_{jm}
\end{aligned}$$

$$+\sum_{l}\sum_{q\neq n}\frac{(\boldsymbol{u}_q\boldsymbol{u}_n^T)_{lk}}{\lambda_q}\ (\boldsymbol{u}_q)_i\ C_{jl}. \tag{9}$$

Given the covariance tensor $\mathcal{V}[\boldsymbol{C}]$, the uncertainty $\boldsymbol{V}[\boldsymbol{u}_n]$ is calculated using Theorem 1. In the case of matrix $\boldsymbol{A} = \boldsymbol{C} \cdot \boldsymbol{C}^T$, a similar description of $\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{C}}$ can be obtained.

Once we are able to state the Euclidian reconstruction problem in terms of functions that allow to apply Theorems 1 to 3, an error description for the reconstructed 3D point cloud can be found. However, since the error is propagated under the assumption of negligible second order derivatives, the error description is bound to be an approximation of the true error.

4 Uncertainty of the Canonical Projective Geometry

In the sequel, it is assumed that the fundamental matrix $\boldsymbol{F}$ and its uncertainty tensor $\mathcal{V}[\boldsymbol{F}]$ has already been estimated from point correspondences $\{\boldsymbol{x}_\alpha, \boldsymbol{x}'_\alpha\}$ and their uncertainties $\boldsymbol{V}[\boldsymbol{x}_\alpha]$ and $\boldsymbol{V}[\boldsymbol{x}'_\alpha]$. $\boldsymbol{F}$ and $\mathcal{V}[\boldsymbol{F}]$ can be estimated with the integrated method of [Kanatani & Mishima 98] or by running uncertainty estimation á la [Csurka et al. 97] on top of fundamentalmatrix estimation á la [Hartley 95] or [Zhang 96]. In any case, the rank constraint of matrix $\boldsymbol{F}$ should be exactly observed. It is assumed that there is no stochastic dependency between arbitrary point correspondence pairs $\{\boldsymbol{x}_\alpha, \boldsymbol{x}'_\alpha\}$ and $\{\boldsymbol{x}_\beta, \boldsymbol{x}'_\beta\}$, where $\alpha \neq \beta$.

Given a fundamental matrix $\boldsymbol{F}$, the canonical projective geometry is determined by first estimating one epipol $\boldsymbol{u}'$ as the normalized eigenvector of matrix $\boldsymbol{F}\boldsymbol{F}^T$ corresponding to the smallest eigenvalue. Then, according to [Luong & Viéville 94], the canonic pair $(\boldsymbol{P}, \boldsymbol{P}')$ of projective projection matrices is defined as

$$\boldsymbol{P} = (\boldsymbol{I} \mid 0) \quad \text{and} \quad \boldsymbol{P}' = (\boldsymbol{M} \mid \boldsymbol{u}')\,, \tag{10}$$

where $\boldsymbol{I}$ is the 3×3 identity matrix, $\boldsymbol{0}$ is the 3-dimensional nullvector, and equation $\boldsymbol{M} = -\lfloor \boldsymbol{u}' \rfloor_\times \boldsymbol{F}$ holds. For matrix $\lfloor \boldsymbol{a} \rfloor_\times$ equation $\lfloor \boldsymbol{a} \rfloor_\times \boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$ is valid.

Given the covariance tensor $\mathcal{V}[\boldsymbol{F}]$ of fundamental matrix $\boldsymbol{F}$, we want to determine the covariance tensors $\mathcal{V}[\boldsymbol{P}]$ and $\mathcal{V}[\boldsymbol{P}']$ of matrices $\boldsymbol{P}$ and $\boldsymbol{P}'$, respectively. Since $\boldsymbol{P}$ is a constant 34-matrix, its covariance tensor $\mathcal{V}[\boldsymbol{P}]$ is the 3434-nulltensor. In order to find $\mathcal{V}[\boldsymbol{P}']$, let $\{\lambda_i\}$ be the descendingly sorted eigenvalues and $\{\boldsymbol{u}_i\}$ the corresponding eigensystem of symmetric 3×3-matrix $(\boldsymbol{F}\boldsymbol{F}^T)$. As a first step, we calculate the uncertainty $\boldsymbol{V}[\boldsymbol{u}']$ of the epipole $\boldsymbol{u}'$. There are two ways to yield the covariance matrix $\boldsymbol{V}[\boldsymbol{u}']$.

- The first way proceeds using the theory of [Kanatani & Mishima 98], some intermediate steps are discussed in Section 10. According to Theorem 4, we have

$$\boldsymbol{V}[\boldsymbol{u}'] = (\boldsymbol{F}\boldsymbol{F}^T)_r^- \ \boldsymbol{G} \ (\boldsymbol{F}\boldsymbol{F}^T)_r^-, \tag{11}$$

  where

$$(\boldsymbol{F}\boldsymbol{F}^T)_r^- = \sum_{j=1}^{2} \frac{1}{\lambda_j} \boldsymbol{u}_j \boldsymbol{u}_j^T \quad \text{and} \quad \boldsymbol{G} = (G_{ij}) \tag{12}$$

$$\text{with} \quad G_{ij} = \sum_{k,l=1}^{3} \mathcal{V}[\boldsymbol{F}\boldsymbol{F}^T]_{iklj} \ \left(\boldsymbol{u}_n \boldsymbol{u}_n^T\right)_{kl}.$$

  The covariance tensor $\mathcal{V}[\boldsymbol{F}\boldsymbol{F}^T]$ is found to be

$$\begin{aligned} \mathcal{V}[\boldsymbol{F}\boldsymbol{F}^T]_{iklj} \ &= \ \sum_{n,q=1}^{3} \{F_{kn}\left(F_{jq}\mathcal{V}[\boldsymbol{F}]_{inlq} + F_{lq}\mathcal{V}[\boldsymbol{F}]_{injq}\right) \\ &\quad + F_{in}\left(F_{jq}\mathcal{V}[\boldsymbol{F}]_{knlq} + F_{lq}\mathcal{V}[\boldsymbol{F}]_{knjq}\right)\}. \end{aligned} \tag{13}$$

- The other way is to simply use the covariance tensor calculus as detailed in Subsection 10.2, particularly Eq. (64), in combination with Eq. (9). We prefer to go this way, since our implementation reveals that it is more numercially stable compared to the first way.

Next, we note that matrix $\boldsymbol{M}' = -\lfloor \boldsymbol{u}' \rfloor_\times \boldsymbol{F} = \mathfrak{F}(\boldsymbol{F}, \lfloor \boldsymbol{u}' \rfloor_\times)$ is a function $\mathfrak{F}$ of matrices $\boldsymbol{F}$ and $\lfloor \boldsymbol{u}' \rfloor_\times$. Furthermore, matrix $\lfloor \boldsymbol{u}' \rfloor_\times = \mathfrak{G}_1(\boldsymbol{u}')$ can be obtained via linear transformation $\lfloor \boldsymbol{u}' \rfloor_\times = \mathcal{H}\boldsymbol{u}'$ with 333-tensor $\mathcal{H} = (H_{ijk})$ from vector $\boldsymbol{u}'$, where

$$H_{ijk} = \begin{cases} 1, & (i,j,k) \in \{(1,2,3),(2,3,1),(3,1,2)\} \\ -1, & (i,j,k) \in \{(1,3,2),(2,1,3),(3,2,1)\} \\ 0, & \text{otherwise} \end{cases} . \tag{14}$$

Since $\mathcal{V}[\boldsymbol{F}]$ and $\boldsymbol{V}[\boldsymbol{u}']$ are known, we can use the covariance tensor calculus of Subsection 10.2, particularly Eq. (64), to find

$$\begin{aligned}\mathcal{V}[\boldsymbol{M}'] &= \frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{F}}\,\mathcal{V}[\boldsymbol{F}]\,\frac{\partial \boldsymbol{M}'^T}{\partial \boldsymbol{F}} + \frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{u}'}\,\boldsymbol{V}[\boldsymbol{u}']\,\frac{\partial \boldsymbol{M}'^T}{\partial \boldsymbol{u}'} + \\ &\quad \frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{F}}\,\mathcal{V}[\boldsymbol{F},\boldsymbol{u}']\,\frac{\partial \boldsymbol{M}'^T}{\partial \boldsymbol{u}'} + \frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{u}'}\,\mathcal{V}[\boldsymbol{u}',\boldsymbol{F}]\,\frac{\partial \boldsymbol{M}'^T}{\partial \boldsymbol{F}}\,. \end{aligned} \tag{15}$$

We can calculate the gradient tensors

$$\begin{aligned}\left(\frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{F}}\right)_{ijmn} &= -\frac{\partial \sum_k \left(\lfloor \boldsymbol{u} \rfloor_\times\right)_{ik} F_{kj}}{\partial F_{mn}} \\ &= -\sum_{k,l} H_{ikl}\,\left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{F}}\right)_{lmn} F_{kj} - \begin{cases} (\lfloor \boldsymbol{u} \rfloor_\times)_{im}, & n = j \\ 0 \quad , & \text{else} \end{cases}\end{aligned} \tag{16}$$

as well as

$$\begin{aligned}\left(\frac{\partial \boldsymbol{M}}{\partial \boldsymbol{u}'}\right)_{ijl} &= -\frac{\partial \sum_k \left(\lfloor \boldsymbol{u} \rfloor_\times\right)_{ik} F_{kj}}{\partial (\boldsymbol{u}')_l} \\ &= -\sum_{k=1}^{3} \frac{\partial (\lfloor \boldsymbol{u} \rfloor_\times)_{ik}}{\partial (\boldsymbol{u}')_l}\, F_{kj} + (\lfloor \boldsymbol{u} \rfloor_\times)_{ik}\, \underbrace{\frac{\partial F_{kj}}{\partial (\boldsymbol{u}')_l}}_{=O} \\ &= -\sum_{k=1}^{3} H_{ikl}\, F_{kj}.\end{aligned} \tag{17}$$

We further have

$$\left(\frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{F}}\,\mathcal{V}[\boldsymbol{F}]\,\frac{\partial \boldsymbol{M}'^T}{\partial \boldsymbol{F}}\right)_{ijkl} = \sum_{m,n,p,q} \frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{F}}_{ijmn} \cdot \frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{F}}_{klpq} \cdot \mathcal{V}[\boldsymbol{F}]_{mnpq} \tag{18}$$

and

$$\left(\frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{u}'}\,\boldsymbol{V}[\boldsymbol{u}']\,\frac{\partial \boldsymbol{M}'^T}{\partial \boldsymbol{u}'}\right)_{ijkl} = \sum_{m,n} \frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{u}'}_{ijm} \cdot \frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{u}'}_{kln} \cdot \boldsymbol{V}[\boldsymbol{u}']_{mn} \tag{19}$$

as well as

$$\left(\frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{F}}\,\mathcal{V}[\boldsymbol{F},\boldsymbol{u}']\,\frac{\partial \boldsymbol{M}'^T}{\partial \boldsymbol{u}'}\right)_{ijkl} = \sum_{m,n,p} \frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{F}}_{ijmn} \cdot \frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{u}'}_{klp} \cdot \mathcal{V}[\boldsymbol{F},\boldsymbol{u}']_{mnp} \tag{20}$$

and furtheron

$$\left(\frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{u}'}\,\mathcal{V}[\boldsymbol{u}',\boldsymbol{F}]\,\frac{\partial \boldsymbol{M}'^T}{\partial \boldsymbol{F}}\right)_{ijkl} = \sum_{m,n,p} \frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{u}'}_{ijm} \cdot \frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{F}}_{klnp} \cdot \mathcal{V}[\boldsymbol{u}',\boldsymbol{F}]_{mnp}. \quad (21)$$

In order to find $\mathcal{V}[\boldsymbol{P}']$ for $\boldsymbol{P}'$ as it is specified in Eq. (10), we derive

$$\begin{aligned} \left(\frac{\partial \boldsymbol{P}'}{\partial \boldsymbol{M}'}\right)_{ijkl} &= \begin{cases} 1, & i=k \;\&\; j=l \leq 3 \\ 0, & \text{else} \end{cases} \\ &\text{and} \\ \left(\frac{\partial \boldsymbol{P}'}{\partial \boldsymbol{u}'}\right)_{ijk} &= \begin{cases} 1, & i=k \;\&\; j=4 \\ 0, & \text{else} \end{cases} . \end{aligned} \quad (22)$$

Finally, we yield

$$\mathcal{V}[\boldsymbol{P}']_{ijkl} = \begin{cases} \mathcal{V}[\boldsymbol{M}']_{ijkl} &, \quad i,j,k,l \in \{1,2,3\} \\ \boldsymbol{V}[\boldsymbol{u}']_{ik} &, \quad i,k \in \{1,2,3\} \quad \text{and} \quad j=l=4 \\ \mathcal{V}[\boldsymbol{M}',\boldsymbol{u}']_{ijk}, & \quad i,j,k \in \{1,2,3\} \quad \text{and} \quad l=4 \\ \mathcal{V}[\boldsymbol{M}',\boldsymbol{u}']_{jli}, & \quad i,k,l \in \{1,2,3\} \quad \text{and} \quad j=4 \end{cases} . \quad (23)$$

5 Reconstruction in the Canonical Projective Geometry

Given the canonic pair $(\boldsymbol{P},\boldsymbol{P}') = ((\boldsymbol{I} \mid \boldsymbol{0}),(\boldsymbol{M}' \mid \boldsymbol{u}'))$ of projective projection matrices, their respective covariance tensors $\mathcal{V}[\boldsymbol{P}]$ and $\mathcal{V}[\boldsymbol{P}']$ as well as a pair $(\boldsymbol{x},\boldsymbol{x}')$ of image points and their respective covariance matrices $\boldsymbol{V}[\boldsymbol{x}]$ and $\boldsymbol{V}[\boldsymbol{x}']$. We reconstruct the scene point $\boldsymbol{z}$ in the canonic frame using a method published by [Zhang 96], which combines the two standard projection equations for the two image points in a homogeneous linear system. Since its kernel contains scene point $\boldsymbol{z}$, we can apply the theory of Subsection 10.1 to get an uncertainty description for vector $\boldsymbol{z}$.

For projective reconstruction of $\boldsymbol{z}$ from $\boldsymbol{x} = (x_1, x_2, 1)$, $\boldsymbol{x}' = (x_1', x_2', 1)$, and the projection matrices $\boldsymbol{P}$, and $\boldsymbol{P}'$, we use the method given in [Zhang 96]. Let $\boldsymbol{p}_i$ and $\boldsymbol{p}_i'$ be the i-th row vectors of $\boldsymbol{P}$, and $\boldsymbol{P}'$, respectively. Then the method relies on singular value decomposition to find the nullspace of the linear system

$$A \cdot \boldsymbol{z} = \begin{pmatrix} \boldsymbol{A}_0 \\ \boldsymbol{A}'_0 \end{pmatrix} \cdot \boldsymbol{z} = \begin{pmatrix} \boldsymbol{p}_1 - x_1 \cdot \boldsymbol{p}_3 \\ \boldsymbol{p}_2 - x_2 \cdot \boldsymbol{p}_3 \\ \boldsymbol{p}'_1 - x'_1 \cdot \boldsymbol{p}'_3 \\ \boldsymbol{p}'_2 - x'_2 \cdot \boldsymbol{p}'_3 \end{pmatrix} \cdot \boldsymbol{z} = \boldsymbol{0}, \tag{24}$$

i.e. the eigenvector corresponding to the smallest eigenvalue of $A^T \cdot A$. In a comparison of projective reconstruction methods (cf. [Rothwell et al. 97]), this method was found to be superior to four other common approaches. In order to calculate $\mathcal{V}[\boldsymbol{A}]$, we state

$$\begin{aligned} \left(\frac{\partial \boldsymbol{A}_0}{\partial \boldsymbol{P}}\right)_{ijkl} &= \frac{\partial(\boldsymbol{P}_{ij} - x_i \boldsymbol{P}_{3j})}{\partial \boldsymbol{P}_{kl}} = \left\{ \begin{array}{ll} \frac{\partial(\boldsymbol{P}_{il} - x_i \boldsymbol{P}_{3l})}{\partial \boldsymbol{P}_{kl}}, & j = l \\ 0 \quad , & \text{else} \end{array} \right\} \\ &= \left\{ \begin{array}{rl} 1, & j = l,\ i = k \leq 2 \\ -x_i, & j = l,\ k = 3,\ i \leq 2 \\ 0, & \text{else} \end{array} \right. . \end{aligned} \tag{25}$$

Furthermore, we find

$$\left(\frac{\partial \boldsymbol{A}_0}{\partial \boldsymbol{x}}\right)_{ijk} = \frac{\partial(\boldsymbol{P}_{ij} - x_i \boldsymbol{P}_{3j})}{\partial \boldsymbol{x}_k} = \left\{ \begin{array}{rl} -\boldsymbol{P}_{3j}, & i = k \leq 2 \\ 0, & \text{else} \end{array} \right. . \tag{26}$$

Analog statements can be given for the gradients $\frac{\partial \boldsymbol{A}'_0}{\partial \boldsymbol{P}'}$ and $\frac{\partial \boldsymbol{A}'_0}{\partial \boldsymbol{x}'}$. Using the equations

$$\begin{aligned} \left(\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{A}_0}\right)_{ijmn} &= \left\{ \begin{array}{ll} 1, & i = m \leq 2, j = n \\ 0, & \text{else} \end{array} \right. \\ &\text{and} \\ \left(\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{A}'_0}\right)_{ijmn} &= \left\{ \begin{array}{ll} 1, & i - 2 = m \leq 2, j = n \\ 0, & \text{else} \end{array} \right. , \end{aligned} \tag{27}$$

we arrive at the following four gradient descriptions:

$$\begin{aligned} \left(\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{P}}\right)_{ijkl} &= \sum_{m,n} \left(\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{A}_0}\right)_{ijmn} \left(\frac{\partial \boldsymbol{A}_0}{\partial \boldsymbol{P}}\right)_{mnkl} = \left\{ \begin{array}{ll} \left(\frac{\partial \boldsymbol{A}_0}{\partial \boldsymbol{P}}\right)_{ijkl}, & i \leq 2 \\ 0 \quad , & \text{else} \end{array} \right\} \\ &= \left\{ \begin{array}{rl} 1, & 1 \leq i = k \leq 2,\ j = l \\ -x_i, & 1 \leq i \leq 2,\ k = 3,\ j = l \\ 0, & \text{else} \end{array} \right. , \end{aligned} \tag{28}$$

$$\left(\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{x}}\right)_{ijk} = \begin{cases} -\boldsymbol{P}_{3j}, & 1 \leq i = k \leq 2 \\ 0, & \text{else} \end{cases}, \tag{29}$$

$$\left(\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{P}'}\right)_{ijkl} = \begin{cases} 1, & 3 \leq i = k+2 \leq 4,\ j = l \\ -x_{i-2}, & 3 \leq i \leq 4,\ k = 3,\ j = l \\ 0, & \text{else} \end{cases}, \tag{30}$$

$$\left(\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{x}'}\right)_{ijk} = \begin{cases} -\boldsymbol{P}'_{3j}, & 3 \leq i = k+2 \leq 4 \\ 0, & \text{else} \end{cases}. \tag{31}$$

If also the 16 covariance tensors and matrices, which describe all stochastic relations between $\boldsymbol{P}$, $\boldsymbol{x}$, $\boldsymbol{P}'$, $\boldsymbol{x}'$, are given, then an approximation for the error distribution of $\boldsymbol{A}$ can be found. Subsequently, these 16 covariance tensors and matrices are discussed.

- The covariance tensors $\mathcal{V}[\boldsymbol{P}]$, $\mathcal{V}[\boldsymbol{P}']$ and matrices $\boldsymbol{V}[\boldsymbol{x}]$, $\boldsymbol{V}[\boldsymbol{x}']$ as well as $\boldsymbol{V}[\boldsymbol{x}, \boldsymbol{x}']$, $\boldsymbol{V}[\boldsymbol{x}', \boldsymbol{x}]$ are known.

- Since $\mathcal{V}[\boldsymbol{P}]$ is 3434-nulltensor, the covariance tensors $\mathcal{V}[\boldsymbol{P}, \boldsymbol{x}]$, $\mathcal{V}[\boldsymbol{P}, \boldsymbol{P}']$, $\mathcal{V}[\boldsymbol{P}, \boldsymbol{x}']$, $\mathcal{V}[\boldsymbol{x}, \boldsymbol{P}]$, $\mathcal{V}[\boldsymbol{P}', \boldsymbol{P}]$, and $\mathcal{V}[\boldsymbol{x}', \boldsymbol{P}]$ are also nulltensors.

- The remaining covariance tensors $\mathcal{V}[\boldsymbol{x}, \boldsymbol{P}']$, $\mathcal{V}[\boldsymbol{P}', \boldsymbol{x}]$, $\mathcal{V}[\boldsymbol{P}', \boldsymbol{x}']$, $\mathcal{V}[\boldsymbol{x}', \boldsymbol{P}']$ all stochastically relate an image point to projection matrix $\boldsymbol{P}'$. Gradient tensors $\frac{\partial \boldsymbol{M}'}{\partial \boldsymbol{F}}$ and $\frac{\partial \boldsymbol{P}'}{\partial \boldsymbol{M}'}$ are known from Eqs. (16) and (22), respectively. Therefore, gradient tensor $\frac{\partial \boldsymbol{P}'}{\partial \boldsymbol{F}}$ is computable. Also, $\mathcal{V}[\boldsymbol{x}, \boldsymbol{F}]$, $\mathcal{V}[\boldsymbol{F}, \boldsymbol{x}]$, $\mathcal{V}[\boldsymbol{F}, \boldsymbol{x}']$, $\mathcal{V}[\boldsymbol{x}', \boldsymbol{F}]$ can be derived from the epipolar error equation (cf. [Kanatani 96])
$$\hat{e} = \boldsymbol{x}' \cdot \boldsymbol{F} \cdot \boldsymbol{x}. \tag{32}$$
Thus, two of the four remaining covariance tensors are defined to be
$$\mathcal{V}[\boldsymbol{x}, \boldsymbol{P}'] = \mathcal{V}[\boldsymbol{x}, \boldsymbol{F}] \left(\frac{\partial \boldsymbol{P}'}{\partial \boldsymbol{F}}\right)^T \quad \text{and} \quad \mathcal{V}[\boldsymbol{P}', \boldsymbol{x}] = \left(\frac{\partial \boldsymbol{P}'}{\partial \boldsymbol{F}}\right) \mathcal{V}[\boldsymbol{F}, \boldsymbol{x}]. \tag{33}$$
The covariance tensors $\mathcal{V}[\boldsymbol{P}', \boldsymbol{x}']$ and $\mathcal{V}[\boldsymbol{x}', \boldsymbol{P}']$ are defined analogously.

Again, the theory of Section 3, particulary Eq. (9), tells us how to find a covariance description $\boldsymbol{V}[\boldsymbol{z}]$ for reconstructed 3D point $\boldsymbol{z}$. Eq. (72) of Subsection 10.3 is useful for the calculation of the normalization of 3D point $\boldsymbol{z}$ and its covariance description $\boldsymbol{V}[\boldsymbol{z}]$.

Moreover, since the gradient tensors $\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{A}}$ as well as $\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{P}}$, $\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{P}'}$ and $\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{x}}$, $\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{x}'}$ are known, gradient expressions $\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}}$, $\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}'}$, $\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{P}}$, and $\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{P}'}$ can be calculated. Finally, we yield the covariance tensors $\mathcal{V}[\boldsymbol{z}, \boldsymbol{P}]$, $\mathcal{V}[\boldsymbol{P}, \boldsymbol{z}]$, $\mathcal{V}[\boldsymbol{P}', \boldsymbol{z}]$, $\mathcal{V}[\boldsymbol{z}, \boldsymbol{P}']$ as well as the covariance matrices $\boldsymbol{V}[\boldsymbol{z}, \boldsymbol{x}]$, $\boldsymbol{V}[\boldsymbol{z}, \boldsymbol{x}']$, $\boldsymbol{V}[\boldsymbol{x}, \boldsymbol{z}]$, $\boldsymbol{V}[\boldsymbol{x}', \boldsymbol{z}]$.

It is important to realize that reconstruction introduces through projection matrices $\boldsymbol{P}$ and $\boldsymbol{P}'$ a stochastical dependency $\boldsymbol{V}[\boldsymbol{z}_\alpha, \boldsymbol{z}_\beta]$ between arbitrary reconstructed 3D points $\boldsymbol{z}_\alpha$ and $\boldsymbol{z}_\beta$:

$$\begin{aligned} \boldsymbol{V}[\boldsymbol{z}_\alpha, \boldsymbol{z}_\beta] \;=\; & \frac{\partial \boldsymbol{z}_\alpha}{\partial \boldsymbol{P}}\, \mathcal{V}[\boldsymbol{P}]\, \frac{\partial \boldsymbol{z}_\beta}{\partial \boldsymbol{P}}^T + \frac{\partial \boldsymbol{z}_\alpha}{\partial \boldsymbol{P}}\, \mathcal{V}[\boldsymbol{P}, \boldsymbol{P}']\, \frac{\partial \boldsymbol{z}_\beta}{\partial \boldsymbol{P}'}^T \\ & + \frac{\partial \boldsymbol{z}_\alpha}{\partial \boldsymbol{P}'}\, \mathcal{V}[\boldsymbol{P}', \boldsymbol{P}]\, \frac{\partial \boldsymbol{z}_\beta}{\partial \boldsymbol{P}}^T + \frac{\partial \boldsymbol{z}_\alpha}{\partial \boldsymbol{P}'}\, \mathcal{V}[\boldsymbol{P}']\, \frac{\partial \boldsymbol{z}_\beta}{\partial \boldsymbol{P}'}^T . \end{aligned} \tag{34}$$

In the case of reconstruction in the canonical projective geometry the first three summands of Eq. (34) are nulltensors.

6 Uncertainty of the General Projective Geometry

Given the canonic pair $(\boldsymbol{P}, \boldsymbol{P}') = ((\boldsymbol{I} \mid \boldsymbol{0}), (\boldsymbol{M}' \mid \boldsymbol{u}'))$ of projective projection matrices, their respective covariance tensors $\mathcal{V}[\boldsymbol{P}]$ and $\mathcal{V}[\boldsymbol{P}']$ as well as five reconstructed scene points $\boldsymbol{z}_\alpha$ and their respective covariance matrices $\boldsymbol{V}[\boldsymbol{z}_\alpha]$, we can specify a homography $\boldsymbol{H}$, which transforms the descriptions relative to the canonical projective frame into an arbitrary projective frame.

Define $\boldsymbol{Z} = (\boldsymbol{z}_1 \;\ldots\; \boldsymbol{z}_5)$ with $\boldsymbol{z}_\alpha$, $\alpha = 1, \ldots, 5$. Matrix $\boldsymbol{Z}$ defines a projective basis in projective space $\mathcal{P}^3$. Necessarily, the $\boldsymbol{z}_\alpha$ are linearly independent and

$$\exists \boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_5) \neq \boldsymbol{0} : \boldsymbol{Z} \cdot \boldsymbol{\lambda} = \boldsymbol{0}. \tag{35}$$

With the projective equations $\boldsymbol{z}_\alpha = \lambda_\alpha \cdot \boldsymbol{z}_\alpha$ and $\boldsymbol{z}_5 = \sum_{\alpha=1}^{4} \lambda_\alpha \cdot \boldsymbol{z}_\alpha$, we have

$$\boldsymbol{Z} = \underbrace{(\lambda_1 \boldsymbol{z}_1 \;\ldots\; \lambda_4 \boldsymbol{z}_4)}_{=:\boldsymbol{H}} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \tag{36}$$

which defines homography $\boldsymbol{H}$. Eq. (35) is a homogeneous linear equation and the theory of Section 3, particulary Eq. (9), tells us how to find a covariance description $\boldsymbol{V}[\boldsymbol{\lambda}]$ for solution vector $\boldsymbol{\lambda}$. Prior to that, covariance tensor $\mathcal{V}[\boldsymbol{Z}]$ is to be defined using Eq. (34) and

$$\left(\frac{\partial \boldsymbol{Z}}{\partial \boldsymbol{z}_l}\right)_{ijk} = \left\{ \begin{array}{ll} 1, & i=k,\ j=l \\ 0, & \text{else} \end{array} \right. , \tag{37}$$

where k indexes the k-th component of vector $\boldsymbol{z}_l$. A calculation yields

$$\begin{aligned} \mathcal{V}[\boldsymbol{Z}]_{ijkl} &= \left(\sum_{m,n=1}^{5} \frac{\partial \boldsymbol{Z}}{\partial \boldsymbol{z}_m} \boldsymbol{V}[\boldsymbol{z}_m, \boldsymbol{z}_n] \frac{\partial \boldsymbol{Z}}{\partial \boldsymbol{z}_n}^T\right)_{ijkl} && (38) \\ &= \sum_{m,n=1}^{5} \left(\sum_{p,q} \left(\frac{\partial \boldsymbol{Z}}{\partial \boldsymbol{z}_m}\right)_{ijp} \boldsymbol{V}[\boldsymbol{z}_m, \boldsymbol{z}_n]_{pq} \left(\frac{\partial \boldsymbol{Z}}{\partial \boldsymbol{z}_n}\right)_{klq}\right) && (39) \\ &= \boldsymbol{V}[\boldsymbol{z}_j, \boldsymbol{z}_l]_{ik}. && (40) \end{aligned}$$

An approximation to the covariance tensor $\mathcal{V}[\boldsymbol{H}]$ of homography $\boldsymbol{H}$ is defined as

$$\mathcal{V}[\boldsymbol{H}] = \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{Z}}\, \mathcal{V}[\boldsymbol{Z}]\, \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{Z}}^T + \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{\lambda}}\, \boldsymbol{V}[\boldsymbol{\lambda}]\, \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{\lambda}}^T + \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{Z}}\, \mathcal{V}[\boldsymbol{Z}, \boldsymbol{\lambda}]\, \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{\lambda}}^T + \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{\lambda}}\, \mathcal{V}[\boldsymbol{\lambda}, \boldsymbol{Z}]\, \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{Z}}^T. \tag{41}$$

Next, we note $\boldsymbol{H}_{ij} = \lambda'_j \cdot \boldsymbol{Z}_{ij}$, $j \le 4$ as well as $\lambda'_j = \lambda_j / \lambda_5$ and derive

$$\left(\frac{\partial \lambda'_j}{\partial \boldsymbol{Z}}\right)_{kl} = \frac{1}{\lambda_5^2} \left(-\lambda_j \cdot \frac{\partial \lambda_5}{\partial \boldsymbol{Z}_{kl}} + \lambda_5 \cdot \frac{\partial \lambda_j}{\partial \boldsymbol{Z}_{kl}}\right). \tag{42}$$

Then the gradients $\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{\lambda}}$ and $\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{Z}}$ are specified as follows:

$$\frac{\partial \boldsymbol{H}_{ij}}{\partial \boldsymbol{Z}_{kl}} = \frac{\partial(\lambda'_j \cdot \boldsymbol{Z}_{ij})}{\partial \boldsymbol{Z}_{kl}} = \frac{\partial \lambda'_j}{\partial \boldsymbol{Z}_{kl}} \cdot \boldsymbol{Z}_{ij} + \lambda'_j \cdot \frac{\partial \boldsymbol{Z}_{ij}}{\partial \boldsymbol{Z}_{kl}} = \frac{\partial \lambda'_j}{\partial \boldsymbol{Z}_{kl}} \cdot \boldsymbol{Z}_{ij} + \left\{ \begin{array}{ll} \lambda'_j, & i=k, j=l \le 4 \\ 0, & \text{else} \end{array} \right. \tag{43}$$

and

$$\frac{\partial \boldsymbol{H}_{ij}}{\partial \boldsymbol{\lambda}_k} = \frac{\partial(\lambda'_j \cdot \boldsymbol{Z}_{ij})}{\partial \boldsymbol{\lambda}_k} = \frac{\partial \lambda'_j}{\partial \boldsymbol{\lambda}_k} \cdot \boldsymbol{Z}_{ij} + \lambda'_j \cdot \frac{\partial \boldsymbol{Z}_{ij}}{\partial \boldsymbol{\lambda}_k} = \left\{ \begin{array}{ll} \frac{1}{\lambda_5} \cdot \boldsymbol{Z}_{ij}, & k=j \le 4 \\ -\frac{\lambda_j}{\lambda_5^2} \cdot \boldsymbol{Z}_{ij}, & k=5 \\ 0, & \text{else} \end{array} \right\} + \lambda'_j \cdot \frac{\partial \boldsymbol{Z}_{ij}}{\partial \boldsymbol{\lambda}_k}. \tag{44}$$

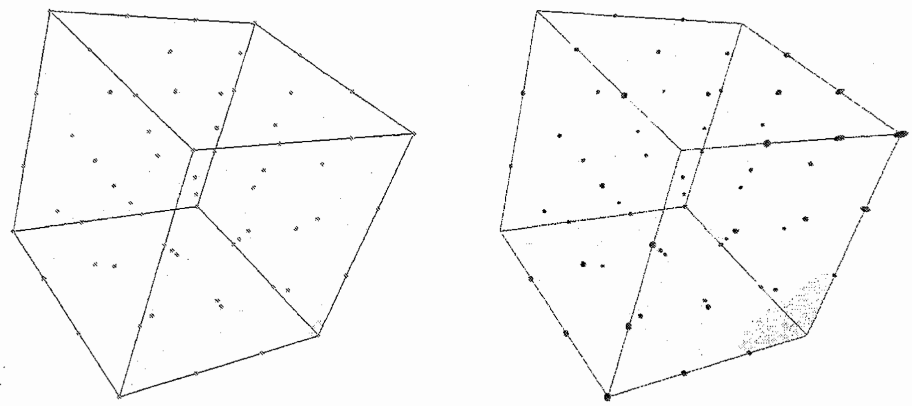

Figure 2: Left Image: A synthetic 3D grid of $4 \times 4 \times 4$ equally spaced 3D points is defined and used to assess the quality of error description that is possible with the approach detailed in this report. Right Image: The results of empiric error estimation based on 100 generated image data sets is shown.

For covariance tensor $\mathcal{V}[\boldsymbol{Z}, \boldsymbol{\lambda}]$ equation $\mathcal{V}[\boldsymbol{Z}, \boldsymbol{\lambda}] = \mathcal{V}[\boldsymbol{Z}] \cdot \frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{Z}}^T(\boldsymbol{E}[\boldsymbol{Z}])$ holds and Eq. (42) defines each component of $\frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{Z}}$. A similar equation can be set up for covariance tensor $\mathcal{V}[\boldsymbol{\lambda}, \boldsymbol{Z}]$.

Since we are also interested in the statistic relation $\mathcal{V}[\boldsymbol{z}_\alpha, \boldsymbol{H}]$ of reconstructed 3D point $\boldsymbol{z}_\alpha$ and homography $\boldsymbol{H}$, gradient $\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{z}_\alpha}$ is derived:

$$\left(\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{z}_\alpha}\right)_{ijk} = \frac{\partial \boldsymbol{H}}{\partial(\boldsymbol{\lambda}, \boldsymbol{Z})} \cdot \frac{\partial(\boldsymbol{\lambda}, \boldsymbol{Z})}{\partial \boldsymbol{z}_\alpha} = \sum_n \frac{\partial \boldsymbol{H}_{ij}}{\partial \boldsymbol{\lambda}_n} \left(\frac{\partial \boldsymbol{\lambda}}{\partial \boldsymbol{Z}} \cdot \frac{\partial \boldsymbol{Z}}{\partial \boldsymbol{z}_\alpha}\right)_{nk} + \left(\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{Z}} \cdot \frac{\partial \boldsymbol{Z}}{\partial \boldsymbol{z}_\alpha}\right)_{ijk} \tag{45}$$

$$= \sum_n \frac{\partial \boldsymbol{H}_{ij}}{\partial \boldsymbol{\lambda}_n} \left(\sum_{p,q} \frac{\partial \boldsymbol{\lambda}_n}{\partial \boldsymbol{Z}_{pq}} \left(\frac{\partial \boldsymbol{Z}}{\partial \boldsymbol{z}_\alpha}\right)_{pqk}\right) + \left(\sum_{m,n} \frac{\partial \boldsymbol{H}_{ij}}{\partial \boldsymbol{Z}_{mn}} \left(\frac{\partial \boldsymbol{Z}}{\partial \boldsymbol{z}_\alpha}\right)_{mnk}\right) \tag{46}$$

$$= \sum_n \frac{\partial \boldsymbol{H}_{ij}}{\partial \boldsymbol{\lambda}_n} \frac{\partial \boldsymbol{\lambda}_n}{\partial \boldsymbol{Z}_{km}} + \frac{\partial \boldsymbol{H}_{ij}}{\partial \boldsymbol{Z}_{km}} \tag{47}$$

Other statistic relations we are interested in are $\mathcal{V}[\boldsymbol{P}, \boldsymbol{H}]$ and $\mathcal{V}[\boldsymbol{P}', \boldsymbol{H}]$. These covariance tensors are determined by equations

$$(\mathcal{V}[\boldsymbol{P}, \boldsymbol{H}])_{ijkl} = \sum_{m,n} \left(\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{P}}\right)_{klmn} \cdot (\mathcal{V}[\boldsymbol{P}])_{ijmn}$$

and

$$(\mathcal{V}[\boldsymbol{P}', \boldsymbol{H}])_{ijkl} = \sum_{m,n} \left(\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{P}'}\right)_{klmn} \cdot (\mathcal{V}[\boldsymbol{P}'])_{ijmn}, \tag{48}$$

where gradients $\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{P}}$ and $\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{P}'}$ are products of gradients $\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{z}_\alpha}$ and $\frac{\partial \boldsymbol{z}_\alpha}{\partial \boldsymbol{P}}$, $\alpha = 1, \ldots, 5$, as well as $\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{z}_\alpha}$ and $\frac{\partial \boldsymbol{z}_\alpha}{\partial \boldsymbol{P}'}$, respectively.

The actual transform of the projective basis is done by $\boldsymbol{Q} = \boldsymbol{P} \cdot \boldsymbol{H}$ and $\boldsymbol{Q}' = \boldsymbol{P}' \cdot \boldsymbol{H}$. Using the covariance tensor calculus on the input covariances $\mathcal{V}[\boldsymbol{H}]$, $\mathcal{V}[\boldsymbol{H}, \boldsymbol{P}]$, $\mathcal{V}[\boldsymbol{H}, \boldsymbol{P}']$, we can come up with uncertainty descriptions $\mathcal{V}[\boldsymbol{Q}]$, $\mathcal{V}[\boldsymbol{Q}']$, but also with $\mathcal{V}[\boldsymbol{Q}, \boldsymbol{Q}'] = \frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{H}} \cdot \mathcal{V}[\boldsymbol{H}] \cdot \frac{\partial \boldsymbol{Q}'}{\partial \boldsymbol{H}}^T$.

7 Reconstruction in the General Projective Geometry

The theory of Section 5 can be used for the task of reconstructing a 3d point in the general projective geometry that is specified by matrices $\boldsymbol{Q}$ and $\boldsymbol{Q}'$ of Section 6. Apart from the uncertainty descriptions $\mathcal{V}[\boldsymbol{x}, \boldsymbol{Q}]$, $\mathcal{V}[\boldsymbol{x}, \boldsymbol{Q}']$, $\mathcal{V}[\boldsymbol{x}, \boldsymbol{Q}']$, and $\mathcal{V}[\boldsymbol{x}', \boldsymbol{Q}']$ all other uncertainty descriptions such as $\mathcal{V}[\boldsymbol{Q}]$ and $\mathcal{V}[\boldsymbol{Q}']$ are known. We can approximate

$$\mathcal{V}[\boldsymbol{Q}, \boldsymbol{x}] = \frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{P}} \cdot \mathcal{V}[\boldsymbol{P}, \boldsymbol{x}] \quad \text{where} \quad \left(\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{P}}\right)_{ijkl} = \sum_{mn} \left(\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{H}}\right)_{ijmn} \cdot \left(\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{P}}\right)_{mnkl}. \tag{49}$$

The covariance tensors $\mathcal{V}[\boldsymbol{x}, \boldsymbol{Q}]$, $\mathcal{V}[\boldsymbol{x}, \boldsymbol{Q}']$, $\mathcal{V}[\boldsymbol{x}, \boldsymbol{Q}']$, and $\mathcal{V}[\boldsymbol{x}', \boldsymbol{Q}']$ can be derived analogously.

8 Experiment

8.1 Synthetic Image Data

This experiment with synthetic data serves to assess the error estimation capabilities of the approach. Comparison to the truth as well as to empirically estimated error descriptions is possible.

A 3D grid of $4 \times 4 \times 4$ equally spaced 3D points and 3 artificial cameras are defined. The 3D grid is projected to their respective image planes (see Fig. 2). Image noise of $\sigma = 0.6$[pixel] is added to the components of the projected points. In order to find the empirical error distribution, 100 data sets are generated this way. Per data set, a 3D point set is projectively reconstructed using the approach given in [Tonko & Kinoshita 99]. The transformation into metric space is facilitated with the knowledge of the true homography, that transforms the true projective reconstruction into the true metric reconstruction.

Also image data set 6 is used to projectively reconstruct a 3D point cloud. Its error distribution is estimated with the approach detailed in this paper. The transfer into Euclidian space is done with an estimated homography, which can be calculated from the three images. Fig. 3 shows a close-up of 16 reconstructed 3D points. Precisely, the true 3D points, the empirically estimated 3D points including error distribution (light colors) as well as those estimated from data set 6 are shown (dark colors). The true points coincide with the empirical mean. Comparing the ellipsoids, you can judge that the estimation of ellipsoid orientation is quite good. Certainly, there is an over- or underestimation of the ellipsoid volume. All ellipsoids correspond to a 75%-confidence. Despite this low confidence, the true point is in all cases already within the dark colored ellipsoids, which correspond to the single shot estimation based on image data set 6.

Next, the question "Is modelling of stochastic dependencies necessary?" is of interest. Fig. 4 shows four true 3D points and two error distributions per true point. The dark ellipsoids are calculated assuming that there is no stochastic dependency between the projection matrices and that those matrices are exact. It is further assumed that all reconstructed 3D points are stochastically independent from each other. The light ellipsoids are calculated under the assumption that all stochastic dependencies have to be modelled, i.e. the error propagation approach of this report is used. Note that each ellipsoid volume corresponds to a 99%-confidence. One can see that all four dark ellipsoids do not contain the true value whereas the light ellipsoids do. Thus, the light ellipsoids describe the error better than the dark ellipsoids and it is strongly recommended to model the entire set of stochastic dependencies as done in this paper.

Fig. 5 shows that there is an unwanted error amplification outside of the region that is surrounded by the four reference points of the projective basis. This fact is also observed by [Georgis et al. 98], who state "...unless the points are projected within a region more or less surrounded by the reference points, we are bound to have amplification of the [3D position] error ...".

8.2 Real Image Data

In this experiment, real image data is used for 3D reconstruction (see Fig. 6). Three images of a polyhedral structure consisting of three planes, which are orthogonal to each other, are acquired. Subsequently, the projective reconstruction approach as detailed in this report and [Tonko & Kinoshita 99] is applied to find a projective as well as 3D Euclidian reconstruction.

Fig. 7 shows the Euclidian reconstruction of the polyhedral structure from different views. One can see that angular information as well as length ratios are picked up with some error that has to be described. Fig. 8 shows the results of single-shot error distribution estimation as presented in this report. Note that the error estimation is done for a projective reconstruction. Using the estimated homography, which transforms the projective into an Euclidan reconstruction, as is, error ellipsoids can be transformed into and displayed in Euclidian space.

For each reconstructed point, its 99%-confidence region is given as light gray rendered ellipsoid. Apart from the lower right image, all other images show two planar faces that are tilted towards a third planar face. Most of the error ellipsoids point into the direction of the tilt, i.e. they specify high error in the direction of the tilt. The lower right image shows not so much error when judging only from its appearance. Naturally, the error ellipsoids are quite small, i.e. the amount of light gray covering the dark gray rendered structure is small.

9 Conclusion

Based on an existing sequence of equations that allows to reconstruct 3D information from images taken with uncalibrated cameras (see [Tonko & Kinoshita 99]), this report explains how uncertainty in the input data can be propagated through this sequence of equations into output data. The covariance tensor calculus is used to find approximate error descriptions for each equation of the sequence. The use of perturbation theory enables us to propagate uncertainty also in case of nullspace problems.

In the course of error propagation, we take *all* stochastic dependencies into account, which exist because of the nature of the problem. In particular, stochastic dependencies between two distinct reconstructed 3D points, but also between 3D points and projective projection matrices. Doing this, the propagation of uncertainty gets much more complicated. However, a synthetic experiment validates that for the case of modelling all stochastic dependencies, on average a 99%-confidence error distribution really contains the true value. Also it is shown that errors are damped or amplified in certain 3D point positions, which is due to the fact that the error propagation detailed in this report is a first order approximation of the truth.

The resulting error information can be used by error reduction approaches like the one detailed in [Kinoshita & Tonko 99] which reduces the position uncertainty of each 3D point of a reconstructed point set based on a new image of this point set.

10 Theorems and Equations

10.1 Uncertainty of the Nullspace

Given a perturbed mn-matrix $\boldsymbol{C}'$ of rank $r = \min\{\mathsf{m}, \mathsf{n}\} - 1$ and its covariance tensor $\mathcal{V}[\boldsymbol{C}']$, we want to find vector $\boldsymbol{u}'$ of the nullspace of $\boldsymbol{C}'$ and its covariance matrix $\boldsymbol{V}[\boldsymbol{u}']$.

Since [Kanatani & Mishima 98] give results for the special case of the fundamental matrix and one epipol, we can easily generalize their result. First, we note that vector $\boldsymbol{u}'$ coincides with the eigenvector $\boldsymbol{u}'_n$ of the symmetric matrix $\boldsymbol{A}' = \boldsymbol{C}'^T\boldsymbol{C}'$, where $\boldsymbol{u}'_n$ corresponds to eigenvalue $\lambda'_n = 0$. $\boldsymbol{u}'$ can thus be derived via eigenanalysis of $\boldsymbol{A}'$.

Matrix $\boldsymbol{V}[\boldsymbol{u}']$ can be derived with the *perturbation theorem* given in [Kanatani 96], which states the relations

$$\lambda'_i \approx \lambda_i + \epsilon\, \boldsymbol{u}_i^T\, \boldsymbol{D}\boldsymbol{u}_i \tag{50}$$

$$\boldsymbol{u}'_i \approx \boldsymbol{u}_i + \epsilon \sum_{j \neq i} \frac{\boldsymbol{u}_j^T\, \boldsymbol{D}\boldsymbol{u}_i}{\lambda_i - \lambda_j}\, \boldsymbol{u}_j \tag{51}$$

between the Eigenvalues λ_i, λ'_i and respective Eigenvectors $\boldsymbol{u}_i$, $\boldsymbol{u}'_i$ of square matrices $\boldsymbol{A}$ and $\boldsymbol{A}' = \boldsymbol{A} + \epsilon\boldsymbol{D}$ (also see Theorem 2).

In particular, because $\lambda_n = 0$ and in general $(\boldsymbol{a}^T\boldsymbol{b})\,\boldsymbol{c} = (\boldsymbol{c}\boldsymbol{a}^T)\,\boldsymbol{b}$, we have

$$\boldsymbol{u}'_n = \boldsymbol{u}_n + \epsilon \sum_{j=1}^{r} \frac{\boldsymbol{u}_j^T\, \boldsymbol{D}\boldsymbol{u}_n}{\lambda_n - \lambda_j}\boldsymbol{u}_j = \boldsymbol{u}_n - \epsilon \sum_{j=1}^{r} \frac{\boldsymbol{u}_j\, \boldsymbol{u}_j^T}{\lambda_j}\, \boldsymbol{D}\, \boldsymbol{u}_n = \boldsymbol{u}_n - \epsilon(\boldsymbol{A})_r^-\, \boldsymbol{D}\, \boldsymbol{u}_n, \tag{52}$$

where the rank-constrained generalized inverse $(\boldsymbol{A})_r^-$ of $\boldsymbol{A}$ is defined as $(\boldsymbol{A})_r^- = \sum_{j=1}^{r} \frac{1}{\lambda_j}\ \boldsymbol{u}_j \boldsymbol{u}_j^T$ (see [Kanatani 96]). Since $\boldsymbol{V}[\boldsymbol{u}_n'] = E[\boldsymbol{\Delta u}_n\ \boldsymbol{\Delta u}_n^T]$ for $\boldsymbol{u}_n' = \boldsymbol{u}_n + \boldsymbol{\Delta u}_n$ and $(\boldsymbol{A})_r^- = ((\boldsymbol{A})_r^-)^T$, we get

$$\begin{aligned}
\boldsymbol{V}[\boldsymbol{u}_n'] &= E[\left(\epsilon(\boldsymbol{A})_r^-\ \boldsymbol{D}\ \boldsymbol{u}_n\right)\ \left(\epsilon(\boldsymbol{A})_r^-\ \boldsymbol{D}\ \boldsymbol{u}_n\right)^T] \\
&= \epsilon^2\ E[(\boldsymbol{A})_r^-\ \boldsymbol{D}\boldsymbol{u}_n\boldsymbol{u}_n^T\boldsymbol{D}^T\ (\boldsymbol{A})_r^-] && (53) \\
&= \epsilon^2\ (\boldsymbol{A})_r^-\ E[\boldsymbol{D}\boldsymbol{u}_n\boldsymbol{u}_n^T\boldsymbol{D}^T]\ (\boldsymbol{A})_r^- && (54) \\
&= (\boldsymbol{A})_r^-\ \boldsymbol{G}\ (\boldsymbol{A})_r^-. && (55)
\end{aligned}$$

Here, nn-matrix $\boldsymbol{G}$ is elementwise defined as

$$G_{ij} = E\left[\sum_{k,l=1}^{\mathsf{n}} \epsilon^2\ D_{ik}\ D_{lj}\ \left(\boldsymbol{u}_n\boldsymbol{u}_n^T\right)_{kl}\right] = \sum_{k,l=1}^{\mathsf{n}} \mathcal{V}[\boldsymbol{A}']_{iklj}\ \left(\boldsymbol{u}_n\boldsymbol{u}_n^T\right)_{kl} \qquad (56)$$

and $E[\cdot]$ denotes expectation for scalars.

In the following, we determine the covariance tensor $\mathcal{V}[\boldsymbol{A}']$ of matrix $\boldsymbol{A}'$. In general, if covariance tensor $\mathcal{V}[\boldsymbol{C}']$ of mn-matrix $\boldsymbol{C}'$ is known, the covariance tensor $\mathcal{V}[\boldsymbol{A}']$ of nn-matrix $\boldsymbol{A}'$ is calculated using

$$\mathcal{V}[\boldsymbol{A}']_{iklj} = \sum_{m,p=1}^{\mathsf{m}} \sum_{n,q=1}^{\mathsf{n}} \left(\frac{\partial \boldsymbol{A}'}{\partial \boldsymbol{C}'}\right)_{ikmn} \left(\frac{\partial \boldsymbol{A}'}{\partial \boldsymbol{C}'}\right)_{ljpq} \mathcal{V}[\boldsymbol{C}']_{mnpq}. \qquad (57)$$

Given matrix $\boldsymbol{C}'$ and matrix $\boldsymbol{A}' = \boldsymbol{C}'^T \cdot \boldsymbol{C}'$, we get

$$\begin{aligned}
\left(\frac{\partial \boldsymbol{A}'}{\partial \boldsymbol{C}'}\right)_{ikmn} &= \frac{\partial A_{ik}'}{\partial C_{mn}'} = \frac{\partial \sum_j C_{ji}' \cdot C_{jk}'}{\partial C_{mn}'} \\
&= \frac{\partial C_{mi}' \cdot C_{mk}'}{\partial C_{mn}'} = \left\{\begin{array}{cl} 0 & ,\quad i \neq n\ \&\ k \neq n \\ C_{mi}' & ,\quad i \neq n\ \&\ k = n \\ C_{mk}' & ,\quad i = n\ \&\ k \neq n \\ 2\ C_{mi}' & ,\quad i = n\ \&\ k = n \end{array}\right. . && (58)
\end{aligned}$$

Furthermore, using Eq. (58) we have

$$\begin{aligned}
\mathcal{V}[\boldsymbol{A}']_{iklj} &= \sum_{m,p=1}^{\mathsf{m}} \sum_{n\in\{i,k\}} \sum_{q\in\{l,j\}} \left(\frac{\partial \boldsymbol{A}'}{\partial \boldsymbol{C}'}\right)_{ikmn} \left(\frac{\partial \boldsymbol{A}'}{\partial \boldsymbol{C}'}\right)_{ljpq} \mathcal{V}[\boldsymbol{C}']_{mnpq} && (59) \\
&= \sum_{m,p=1}^{\mathsf{m}} \{C_{mk}' \left(C_{pj}'\mathcal{V}[\boldsymbol{C}']_{mipl} + C_{pl}'\mathcal{V}[\boldsymbol{C}']_{mipj}\right) \\
&\quad + C_{mi}' \left(C_{pj}'\mathcal{V}[\boldsymbol{C}']_{mkpl} + C_{pl}'\mathcal{V}[\boldsymbol{C}']_{mkpj}\right)\}. && (60)
\end{aligned}$$

In an analog calculation, given matrix $\boldsymbol{C}'$ and matrix $\boldsymbol{A}' = \boldsymbol{C}' \cdot \boldsymbol{C}'^T$, we get

$$\begin{aligned}\left(\frac{\partial \boldsymbol{A}'}{\partial \boldsymbol{C}'}\right)_{ikmn} &= \frac{\partial A'_{ik}}{\partial C'_{mn}} = \frac{\partial \sum_j C'_{ij} \cdot C'_{kj}}{\partial C'_{mn}} \\ &= \frac{\partial C'_{in} \cdot C'_{kn}}{\partial C'_{mn}} = \begin{cases} 0 \;, & i \neq m \;\&\; k \neq m \\ C'_{in} \;, & i \neq m \;\&\; k = m \\ C'_{kn} \;, & i = m \;\&\; k \neq m \\ 2\, C'_{in}, & i = m \;\&\; k = m \end{cases} .\end{aligned} \tag{61}$$

Furthermore, using Eq. (61) we have

$$\mathcal{V}[\boldsymbol{A}']_{iklj} = \sum_{m \in \{i,k\}} \sum_{p \in \{l,j\}} \sum_{n,q=1}^{\mathsf{n}} \left(\frac{\partial \boldsymbol{A}'}{\partial \boldsymbol{C}'}\right)_{ikmn} \left(\frac{\partial \boldsymbol{A}'}{\partial \boldsymbol{C}'}\right)_{ljpq} \mathcal{V}[\boldsymbol{C}']_{mnpq} \tag{62}$$

$$\begin{aligned} &= \sum_{n,q=1}^{\mathsf{n}} \{C'_{kn} \left(C'_{jq} \mathcal{V}[\boldsymbol{C}']_{inlq} + C'_{lq} \mathcal{V}[\boldsymbol{C}']_{injq}\right) \\ &\quad + C'_{in} \left(C'_{jq} \mathcal{V}[\boldsymbol{C}']_{knlq} + C'_{lq} \mathcal{V}[\boldsymbol{C}']_{knjq}\right)\}. \end{aligned} \tag{63}$$

Therefore, we are able to state

Theorem 4 *Let $\boldsymbol{C}'$ be a perturbed* mn*-matrix of rank $r = \min\{\mathsf{m}, \mathsf{n}\} - 1$ and let $\mathcal{V}[\boldsymbol{C}']$ be its covariance tensor $\mathcal{V}[\boldsymbol{C}']$. Then the vector $\boldsymbol{u}'$ of the nullspace of $\boldsymbol{C}'$ is found to be the eigenvector $\boldsymbol{u}'_n$ corresponding to the smallest eigenvalue $\lambda'_n = 0$ of matrix $\boldsymbol{A}' = \boldsymbol{C}'^T \boldsymbol{C}'$. Further, Eqs. (55), (56), and (60) describe, how $\boldsymbol{V}[\boldsymbol{u}']$ is calculated from $\boldsymbol{C}'$, $\mathcal{V}[\boldsymbol{C}']$, $\boldsymbol{u}_n$, and $\boldsymbol{A}$. If $\boldsymbol{A}' = \boldsymbol{C}'\boldsymbol{C}'^T$, then Eqs. (55), (56), and (63) are used for calculation. Since we do not know the true value of $\boldsymbol{u}_n$ and $\boldsymbol{A}$, we use $\boldsymbol{u}'_n$ and $\boldsymbol{A}'$, respectively, in the implementation.*

10.2 Covariance Tensor Calculus

Given matrices $\boldsymbol{A_1}$, $\boldsymbol{A_2}$, ..., $\boldsymbol{A_n}$, and matrix $\boldsymbol{B}$ as a function of matrices $\boldsymbol{A_1}, \ldots, \boldsymbol{A_n}$. Assume that also the covariance tensors $\mathcal{V}[\boldsymbol{A}_i]$, $i = 1, \ldots, n$ are given. The covariance tensor $\mathcal{V}[\boldsymbol{B}]$ of $\boldsymbol{B}$ is then approximated as

$$\mathcal{V}[\boldsymbol{B}] = \sum_{i,j=1}^{n} \frac{\partial \boldsymbol{B}}{\partial \boldsymbol{A}_i} \, \mathcal{V}[\boldsymbol{A}_i, \boldsymbol{A}_j] \, \frac{\partial \boldsymbol{B}}{\partial \boldsymbol{A}_j}^T . \tag{64}$$

For tensors $\mathcal{A} = (A_{ijkl})$ and $\mathcal{B} = B_{ijkl}$ the product $\mathcal{A}\mathcal{B}\mathcal{A}^T$ is a tensor whose $(ijkl)$ element is $\sum_{m,n,p,q} A_{ijmn}A_{klpq}B_{mnpq}$. The product $\mathcal{B}\mathcal{A}^T$ is a tensor whose $(mnkl)$ element is $\sum_{p,q} A_{klpq}B_{mnpq}$.

If $\boldsymbol{B} = \mathfrak{G}(\boldsymbol{A})$ depends via function $\mathfrak{G}$ on $\boldsymbol{A}$ and $\mathcal{V}[\boldsymbol{A}, \boldsymbol{B}]$ needs to be calculated, we use the Taylor expansion $\mathfrak{G}(\boldsymbol{A}) = \mathfrak{G}(\boldsymbol{E}[\boldsymbol{A}]) + \frac{\partial \mathfrak{G}}{\partial \boldsymbol{A}}(\boldsymbol{E}[\boldsymbol{A}])\,(\boldsymbol{A} - \boldsymbol{E}[\boldsymbol{A}])$ and get

$$\begin{aligned}
\mathcal{V}[\boldsymbol{A}, \boldsymbol{B}] &= \mathcal{E}\left[(\boldsymbol{A} - \boldsymbol{E}[\boldsymbol{A}]) \otimes (\mathfrak{G}(\boldsymbol{A}) - \boldsymbol{E}[\mathfrak{G}(\boldsymbol{A})])\right] && (65)\\
&= \mathcal{E}\left[(\boldsymbol{A} - \boldsymbol{E}[\boldsymbol{A}]) \otimes \left\{\frac{\partial \mathfrak{G}}{\partial \boldsymbol{A}}(\boldsymbol{E}[\boldsymbol{A}]) \cdot (\boldsymbol{A} - \boldsymbol{E}[\boldsymbol{A}])\right\}\right] && (66)\\
&= \mathcal{E}\left[(\boldsymbol{A} - \boldsymbol{E}[\boldsymbol{A}]) \otimes (\boldsymbol{A} - \boldsymbol{E}[\boldsymbol{A}]) \cdot \frac{\partial \mathfrak{G}}{\partial \boldsymbol{A}}^T(\boldsymbol{E}[\boldsymbol{A}])\right] && (67)\\
&= \mathcal{E}\left[(\boldsymbol{A} - \boldsymbol{E}[\boldsymbol{A}]) \otimes (\boldsymbol{A} - \boldsymbol{E}[\boldsymbol{A}])\right] \cdot \frac{\partial \mathfrak{G}}{\partial \boldsymbol{A}}^T(\boldsymbol{E}[\boldsymbol{A}]) && (68)\\
&= \mathcal{V}[\boldsymbol{A}] \cdot \frac{\partial \mathfrak{G}}{\partial \boldsymbol{A}}^T(\boldsymbol{E}[\boldsymbol{A}]). && (69)
\end{aligned}$$

Analog to that, the following can be proven:

$$\mathcal{V}[\boldsymbol{B}, \boldsymbol{A}] = \frac{\partial \mathfrak{G}}{\partial \boldsymbol{A}}(\boldsymbol{E}[\boldsymbol{A}]) \cdot \mathcal{V}[\boldsymbol{A}]. \tag{70}$$

For matrices $\boldsymbol{A} = (A_{ij})$ and $\boldsymbol{B} = (B_{ij})$, the tensor product $\boldsymbol{A} \otimes \boldsymbol{B}$ is a tensor whose $(ijkl)$ element is $A_{ij}B_{kl}$. For tensor $\boldsymbol{A} = (A_{ijkl})$ and matrix $\boldsymbol{B} = (B_{ij})$, the product $\mathcal{A}\boldsymbol{B}$ is a matrix whose (ij) element is $\sum_{k,l} A_{ijkl}B_{kl}$. In fact, Eq. (64) can be derived using the calculations of Eqs. (65) to (69).

10.3 Miscellaneous

- Since equation

  $$\boldsymbol{A}' = \sum_{k \neq n} \lambda'_k\, \boldsymbol{u}'_k \cdot \boldsymbol{u}'^T_k \tag{71}$$

  holds for matrix $\boldsymbol{A}'$ and its eigensystem excluding eigenvector $\boldsymbol{u}'_n$ for eigenvalue $\lambda'_n = 0$, we can compute the derivative $\frac{\partial \boldsymbol{A}'}{\partial \boldsymbol{u}'_n}$ to be the null-tensor. Thus, $\frac{\partial \boldsymbol{C}'}{\partial \boldsymbol{u}'_n} = \frac{\partial \boldsymbol{C}'}{\partial \boldsymbol{A}'}\frac{\partial \boldsymbol{A}'}{\partial \boldsymbol{u}'_n}$ also equals the nulltensor.

- Suppose n-vector $\boldsymbol{z} = (z_i)$, $z_n \neq 0$, and its covariance matrix $\boldsymbol{V}[\boldsymbol{z}]$ are given and $\boldsymbol{z}' = 1/z_n \cdot \boldsymbol{z}$ as well as $\boldsymbol{V}[\boldsymbol{z}']$ is needed. Then, according to [Kanatani 96],

$$\boldsymbol{V}[\boldsymbol{z}'] = \frac{\boldsymbol{Q}_0\, \boldsymbol{V}[\boldsymbol{z}]\, \boldsymbol{Q}_0^T}{z_n^2} \quad \text{with} \quad \boldsymbol{Q}_0 = \boldsymbol{I} - \boldsymbol{z}' \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}^T \tag{72}$$

can be used to obtain the uncertainty description $\boldsymbol{V}[\boldsymbol{z}']$ of vector $\boldsymbol{z}'$. Note that $\boldsymbol{I}$ is the $n \times n$ identity matrix.

References

[Bougnoux 98] S. Bougnoux: *From Projective to Euclidian Space under any Practical Situation, a Criticism of Self-Calibration.* In N. Ahuja, U. Desai (Eds.): Proc. Int. Conf. on Computer Vision ICCV'98. Bombay, India, January 1998. Narosa, New Delhi, Madras, Bombay, et al., pp. 790–796.

[Csurka et al. 97] G. Csurka, C. Zeller, Z. Zhang, O. Faugeras: *Characterizing the Uncertainty of the Fundamental Matrix*. Computer Vision and Image Understanding **68**:1 (1997) 18–36.

[Collins 92] R. T. Collins: *Projective Reconstruction of Approximately Planar Scenes.* In J. Harmon (Ed.): Proc. 21. AIPR Workshop **SPIE-1839**, pp. 174–185.

[Doyon 99] J.-P. Doyon: *Point Cloud Mesh Generation and Rendering*. ATR Technical Report TR-H-280, December 1999, (in press).

[Faugeras 92] O. Faugeras: *What can be seen in Three Dimensions with an Uncalibrated Stereo Rig?*. In G. Sandini (Ed.): Proc. European Conference on Computer Vision ECCV'92. St. Margherita Ligure, Italy, May 1992. Springer-Verlag, Berlin, Heidelberg, New York/NY, Lecture Notes in Computer Science **LNCS-588**, pp. 563–578.

[Georgis et al. 98] N. Georgis, M. Petrou, J. Kittler: *Error Guided Design of a 3D Vision System.* IEEE Trans. on Pattern Analysis and Machine Intelligence **PAMI 20**:4 (1998) 366–379.

[Grimson et al. 92] W. E. L. Grimson, D. P. Huttenlocher, T. D. Alter: *Recognizing 3D Objects from 2D Images: An Error Analysis*. Proc. IEEE Conf. on Comp. Vision and Pattern Recognition, 1992.

[Ha & Kweon 99] J. E. Ha, I. S. Kweon: *Calibration and 3D Structure Recovery under Varying Cameras using Known Angles*. Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. Kyongju, Korea, October 1999. IEEE Computer Society Press. pp. 746–751.

[Hartley 95] R. Hartley: *In Defence of the Eight-Point Algorithm*. Proc. Int. Conf. on Computer Vision. Boston/MA, July 1995. IEEE Computer Society Press. pp. 1064–1070.

[Kinoshita & Tonko 99] K. Kinoshita, M. Tonko: *Improving the Accuracy of 3D Information from Uncalibrated Images*. Computer Vision and Image Media Meeting **CVIM 118-9**, Gunma, Japan, September 1999, pp. 59–66. (In Japanese).

[Kanatani 96] K. Kanatani: *Statistical Optimization for Geometric Computation: Theory and Practise*. Elsevier, Amsterdam, et al., 1996. Series on Machine Intelligence and Pattern Recognition, Vol. 18.

[Kanatani & Mishima 98] K. Kanatani, H. Mishima: *Strictly Optimal Fundamental Matrix: Reliability Analysis and Computation*. In Proc. Int. Conf. on Face and Object Recognition. Kyoto, Japan, July 1998. *(to appear)*.

[Luong & Viéville 94] Q.-T. Luong, T. Vieville: *Canonical Representations for the Geometries of Multiple Projective Views*. Tech. Report UCB/CSD 93-772, Computer Science Division, University of California, Berkeley/CA, July 1994.

[Matei & Meer 99] B. Matei, P. Meer: *Bootstrapping a Heteroscedastic Regression Model with Application to 3D Rigid Motion Evaluation*. In B. Triggs et. al. (Eds.): Pre-Proc. ICCV'99 Workshop on Vision Algorithms – Theory & Practise, Corfu, Greece, 21–22 September 1999, pp. 43–50.

[Rothwell et al. 97] C. Rothwell, O. Faugeras, G. Csurka: *A Comparison of Projective Reconstruction Methods for Pairs of Views*. Computer Vision and Image Understanding **68**:1 (1997) 37–58.

[Singh 90] A. Singh: *An Estimation-Theoretic Framework for Image-Flow Computation*. In Proc. Int. Conf. on Computer Vision. Osaka, Japan, December 1990. IEEE Computer Society Press. pp. 168–177.

[Sun et al. 99] Z. Sun, V. Ramesh, A. M. Tekalp: *Error Characterization of the Factorization Approach to Shape and Motion Recovery*. In B. Triggs et. al. (Eds.): Pre-Proc. ICCV'99 Workshop on Vision Algorithms – Theory & Practise, Corfu, Greece, 21–22 September 1999, pp. 25–31.

[Tonko & Kinoshita 99] M. Tonko, K.Kinoshita: *On the Integration of Point Feature Acquisition, Tracking, and Uncalibrated Metric Reconstruction*. International Society for Optical Engineering, Meeting on Intelligent Robots and Computer Vision XVIII, Boston/MA, 20-21 September 1999, pp. 299–310.

[Zhang 96] Z. Zhang: *Determining the Epipolar Geometry and its Uncertainty*. INRIA Research Report No. 2927, July 1996.

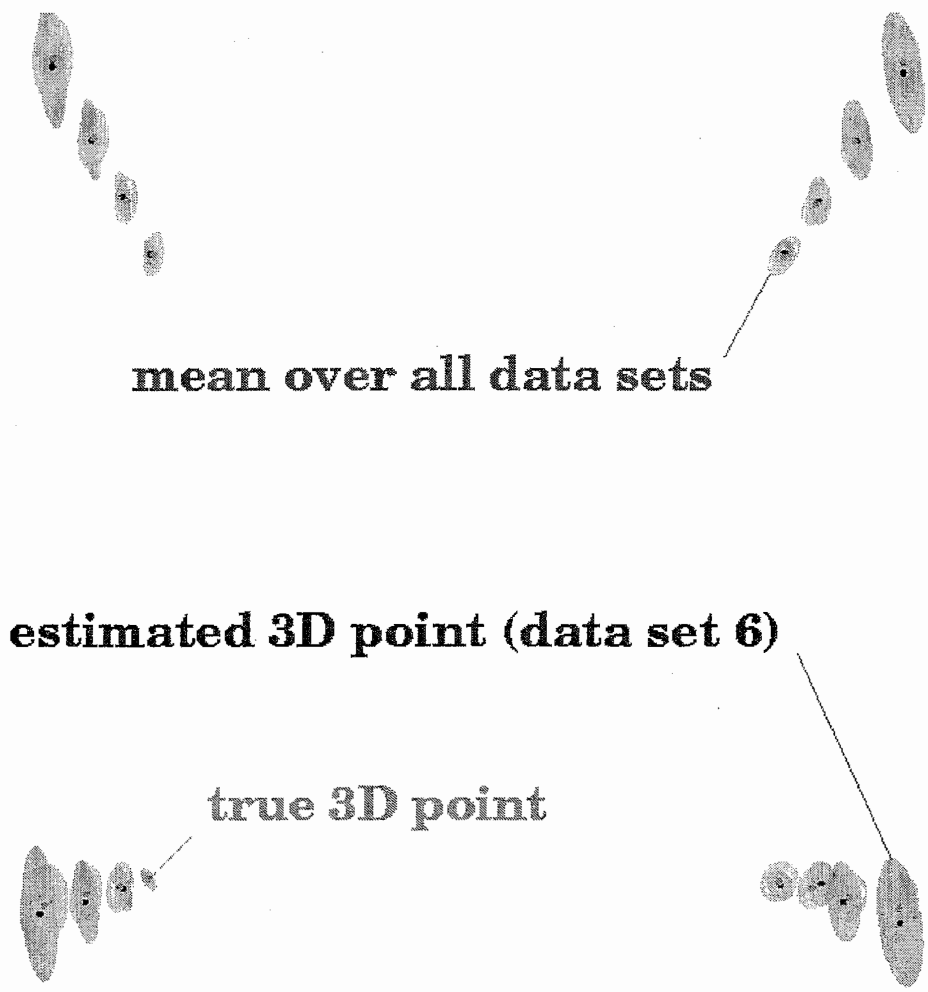

Figure 3: Close-up of 16 reconstructed 3D points: The true 3D points, the empirically estimated 3D points including error distribution (light colors) as well as those estimated from data set 6 are shown (dark colors). The true points coincide with the empirical mean.

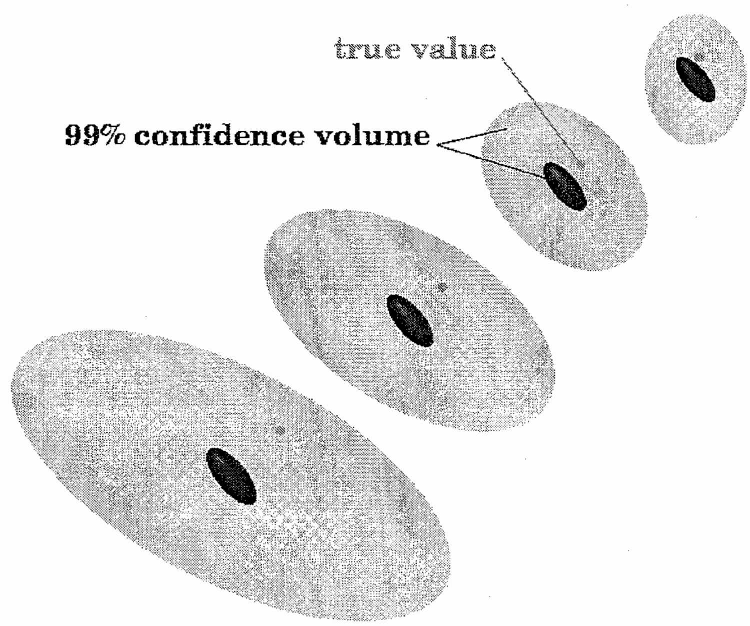

Figure 4: Is modelling of stochastic dependencies necessary? One can see that all four dark ellipsoids – calculated under the assumption that some stochastic dependencies can be neglected – do not contain the true value, whereas the light ellipsoids – calculated under the assumption that all stochastic dependencies have to be modelled – do.

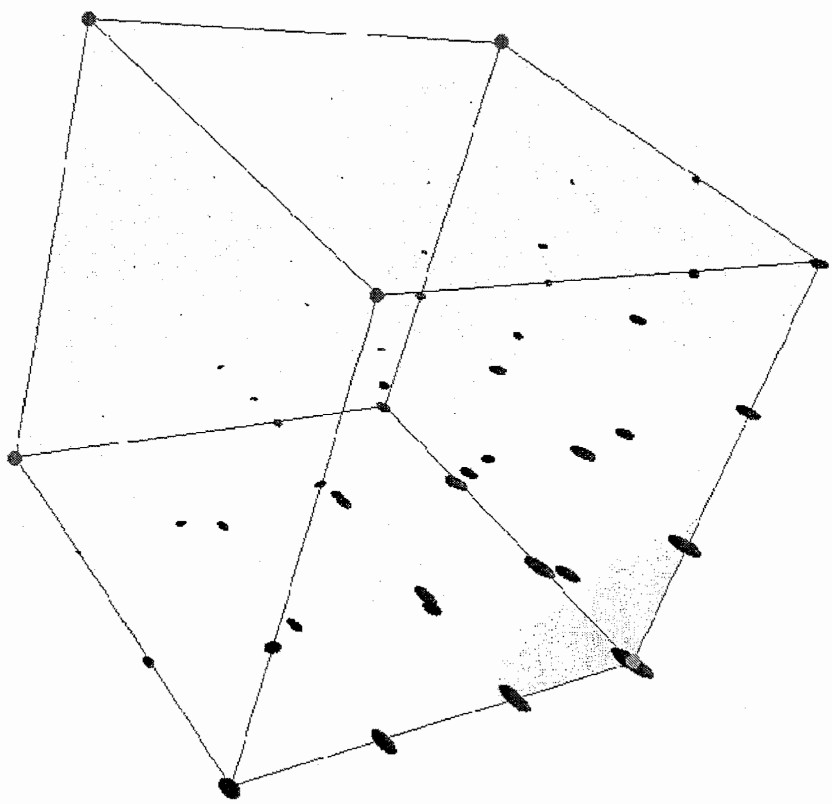

Figure 5: Error Amplification Region: There is an unwanted error amplification outside of the region that is surrounded by the four reference points of the projective basis (see the four light grey points in the upper left region of the image).

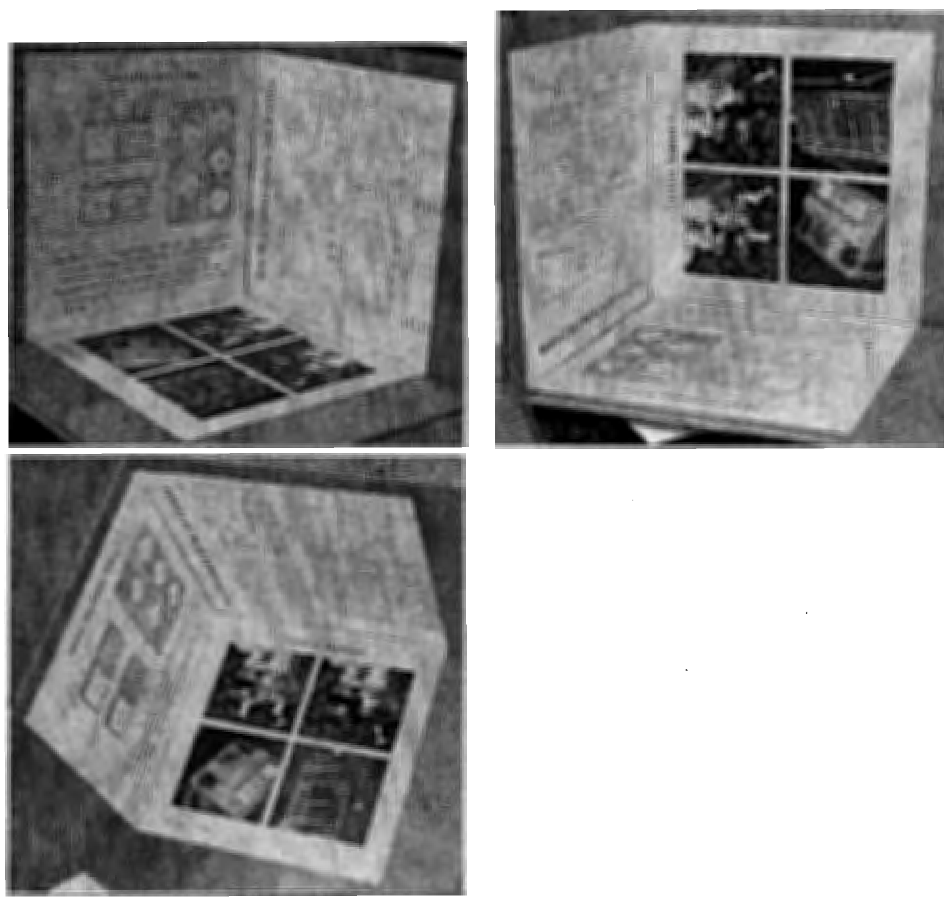

Figure 6: Real Image Data: Three images of a polyhedral structure consisting of three planes, which are orthogonal to each other, are acquired.

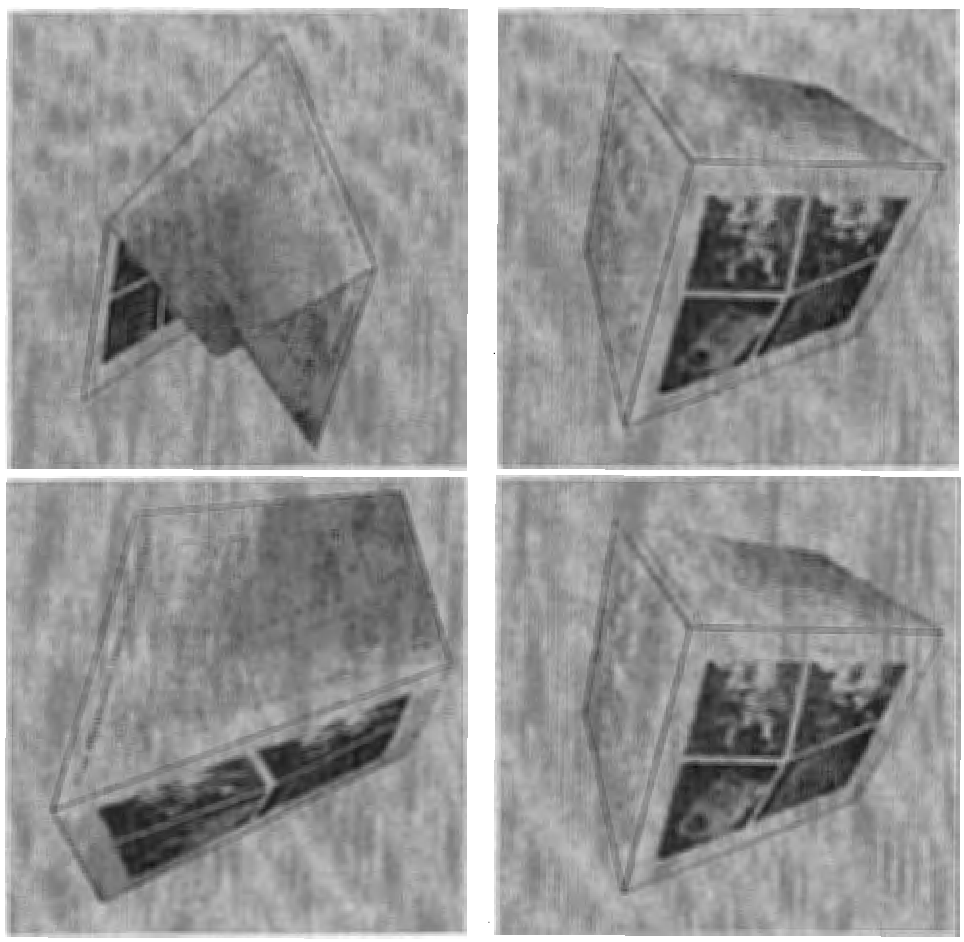

Figure 7: Reconstruction of the polyhedral structure from different views. One can see that angular information as well as length ratios are picked up with some error.

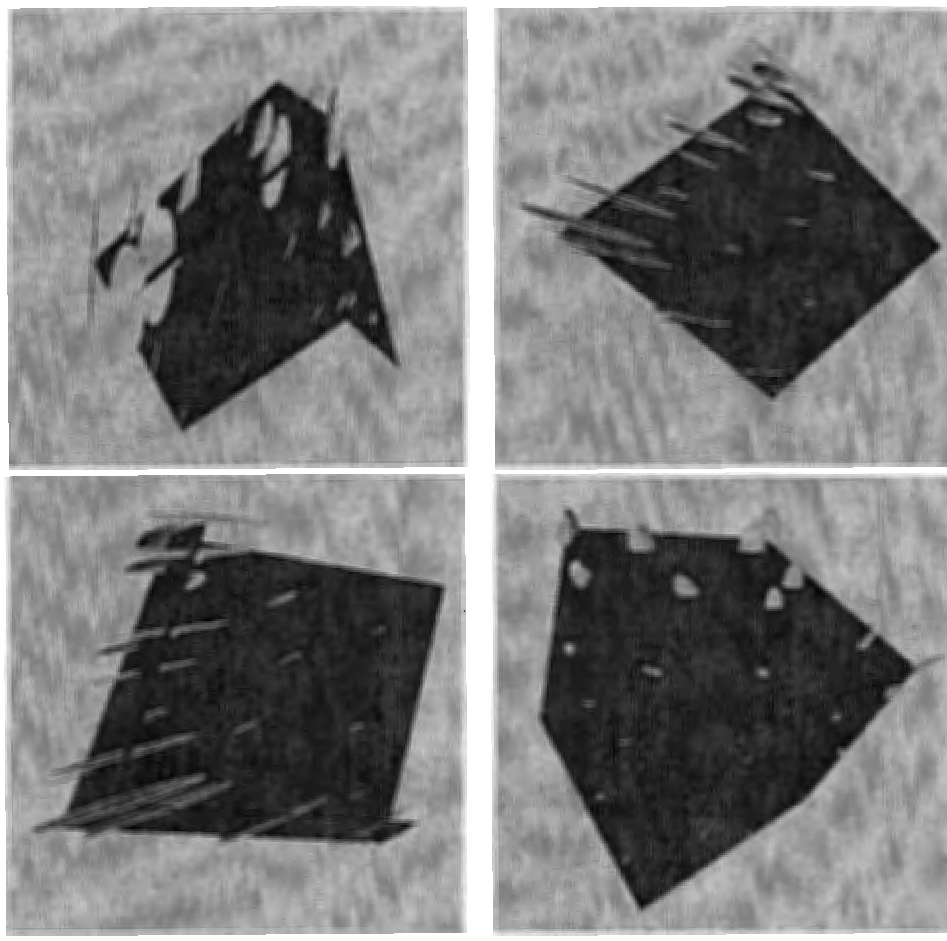

Figure 8: Results of single-shot error distribution estimation as presented in this report. For each reconstructed point, its 99%-confidence region is given as light gray rendered ellipsoid.