Efficient Three-view Triangulation Based on 3D Optimization

Klas Nordberg
Computer Vision Laboratory
Department of Electrical Engineering
Linköping University
klas@isy.liu.se

Abstract

Triangulation of a 3D point from two or more views can be solved in several ways depending on how perturbations in the image coordinates are dealt with. A common approach is *optimal triangulation* which minimizes the total L_2 reprojection error in the images, corresponding to finding a maximum likelihood estimate of the 3D point assuming independent Gaussian noise in the image spaces. Computational approaches for optimal triangulation have been published for the stereo case and, recently, also for the three-view case. In short, they solve an independent optimization problem for each 3D point, using relatively complex computations such as finding roots of high order polynomials or matrix decompositions. This paper discuss three-view triangulation and reports the following results: (1) the 3D point can be computed as multi-linear mapping (tensor) applied on the homogeneous image coordinates, (2) the set of triangulation tensors forms a 7-dimensional space determined by the camera matrices, (3) given a set of corresponding 3D/2D calibration points, the 3D residual L_1 errors can be optimized over the elements in the 7-dimensional space, (4) using the resulting tensor as initial value, the error can be further reduced by tuning the tensor in a two-step iterative process, (5) the 3D residual L_1 error for a set of evaluation points which lie close to the calibration set is comparable to the three-view optimal method. In summary, three-view triangulation can be done by first performing an optimization of the triangulation tensor and once this is done, triangulation can be made with 3D residual error at the same level as the optimal method, but at a much lower computational cost. This makes the proposed method attractive for real-time three-view triangulation of large data sets provided that the necessary calibration process can be performed.

1 Introduction

Triangulation (or reconstruction) of a 3D point from its projection $\mathbf{y}_1, \mathbf{y}_2$ in two images is a well-explored area in computer vision [3]. An common method is *optimal triangulation* which minimizes the total L_2 reprojection error in the image domain. This is

BMVC 2008 doi:10.5244/C.22.19

done by determining two image points $\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2$ that minimize the total L_2 reprojection error $d(\tilde{\mathbf{y}}_1, \mathbf{y}_1)^2 + d(\tilde{\mathbf{y}}_2, \mathbf{y}_2)^2$, where d is the Euclidean 2D distance measured in the image space, while at the same time they satisfy the epipolar constraint $\tilde{\mathbf{y}}_1^T \mathbf{F} \tilde{\mathbf{y}}_2 = 0$ where $\mathbf{F}$ is the fundamental matrix. A non-iterative computational approach for determining $\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2$ is given in [5], and a standard method can then be used to compute the 3D point from $\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2$.

Recently, the corresponding problem for three views has been treated in [9] which also presents a non-iterative method for finding the solution. This approach has a significant computational cost, e.g., eigenvalue analysis of a 47×47 matrix is performed and 128-bit floating point numerics is required to obtain useful robustness of the result. An alternative approach is described in [1], which significantly reduces the computational costs.

The methods mentioned above are optimal in the sense that they find a Maximum Likelihood estimate of the 3D point under the assumption that the image coordinates are affected by independent Gaussian noise. Although this is a reasonable assumption, they solve an independent optimization problem for each reconstructed point and no explicit optimization takes place in the 3D space. In fact, it can be shown that optimal triangulation can produce larger 3D errors than other methods in certain circumstances [5].

An alternative approach to stereo triangulation is presented in [6]. The mapping from image coordinates to the 3D point is represented as a multi-linear mapping (tensor) on the homogeneous image coordinates, except for points lying on a *blind plane* which must includes the two camera focal points. The triangulation tensor can be computed directly from the camera matrices and the blind plane. For applications where this plane can be placed outside the common field of view of the cameras the triangulation computation can be made very fast; a 4×9 matrix multiplied onto the outer product of the homogeneous image coordinates reshaped to a 9-dimensional vector. The triangulation tensor is not unique; for a given configuration of the two cameras, the set of possible triangulation tensors is at least 2-dimensional.

The tensor based approach allows us to investigate which tensor minimizes 3D errors for a calibration set of corresponding 3D+2D+2D coordinates, without first determining the camera matrices. This is in contrast to the above mentioned optimal method which cannot take such *a priori* information into account and also requires accurately estimated camera matrices. Furthermore, this tensor calibration can be based on any reasonable type of 3D error that suits our purpose. Such a calibration for a stereo triangulation tensor is described in [8] which minimizes L_1 errors in the 3D space. This process starts with a standard estimation of the triangulation tensor from corresponding 3D+2D+2D coordinates and then refines the tensor by iteratively minimizing 3D and 2D errors. The resulting tensor provides 3D L_1 errors similar to the optimal method. In this comparison it is important to note that the computational cost of the optimal method is significantly higher than for the tensor based method.

1.1 This paper

The stereo triangulation method described in [8] is here generalized to three views. The same algebraic derivation used in [6] for the stereo case is here applied for three views with the main difference that the space of feasible tensors now is larger. Given a set of corresponding 3D+2D+2D+2D calibration points, an element of this space can be determined that minimizes the total 3D L_1 residual error. In comparison, this tensor does not produce as low errors as the optimal three-view method does. However, by applying a

second optimization stage which adjusts all elements of the tensor, the 3D residual error can be reduced to the same level as the optimal method. As a result, this one-shot optimization produces a tensor which both efficiently and accurately can reconstruct 3D points from three views, assuming that the calibration set is representative for the points used in later reconstructions.

2 A three-view triangulation tensor

In this section, the three-view triangulation tensor $\mathbf{K}$ is derived analogous to the approach presented in [6] for the two-view case. The presentation is based on the multi-view pinhole camera model in which the relation between 2D image coordinates and 3D world coordinates is described as

$$\mathbf{y}_i \sim \mathbf{C}_i \mathbf{x} \qquad [y_{i,\alpha} \sim C_{i,\alpha\beta}\, x_\beta] \tag{1}$$

where $\mathbf{x}$ and $\mathbf{y}_i$ are the homogeneous representation of corresponding 3D and 2D points, $\mathbf{C}_i$ is the camera matrix of view i, and $\sim$ denotes vector equality up to a scalar multiplication. We will sometime use the type of vector/matrix based notation shown to the left, sometimes the coordinate based notation shown to the right, and sometimes both. In the case of coordinate based notation, lower case Latin indices are used for enumeration of views, Greek indices enumerate elements of tensors, and an upper case Latin index represents a group of three Greek indices. Einstein's summation convention is assumed.

The outer product or tensor product of the three homogeneous image coordinates is

$$\mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{y}_3 \sim (\mathbf{C}_1 \mathbf{x}) \otimes (\mathbf{C}_2 \mathbf{x}) \otimes (\mathbf{C}_3 \mathbf{x}) \qquad [y_{1,\alpha} y_{2,\beta} y_{3,\gamma} \sim C_{1,\alpha\delta} x_\delta C_{2,\beta\varepsilon} x_\varepsilon C_{3,\gamma\phi} x_\phi] \tag{2}$$

Reorganizing the factors gives a more operational description of the mapping from $\mathbf{x}$ to $\mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{y}_3$

$$\mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{y}_3 \sim (\mathbf{C}_1 \otimes \mathbf{C}_2 \otimes \mathbf{C}_3)(\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}) \qquad [y_{1,\alpha} y_{2,\beta} y_{3,\gamma} \sim C_{1,\alpha\delta} C_{2,\beta\varepsilon} C_{3,\gamma\phi}\, x_\delta x_\varepsilon x_\phi] \tag{3}$$

$$\mathbf{Y} \sim \mathbf{C}\mathbf{X} \qquad [Y_{\alpha\beta\gamma} \sim C_{\alpha\beta\gamma\delta\varepsilon\phi}\, X_{\delta\varepsilon\phi}] \qquad [Y_I \sim C_{IJ} X_J] \tag{4}$$

In Equation (4) $\mathbf{C} = \mathbf{C}_1 \otimes \mathbf{C}_2 \otimes \mathbf{C}_3$ is seen as a 27×64 matrix which maps $\mathbf{X} = \mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}$ (reshaped to a 64-dimensional vector) to $\mathbf{Y} = \mathbf{y}_1 \otimes \mathbf{y}_2 \otimes \mathbf{y}_3$ (a 27-dimensional vector).

The next step is to make explicit the fact that $\mathbf{X} = \mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}$ always is a completely symmetric third order tensor on $\mathbb{R}^4$, hence an element of a 20-dimensional subspace denoted X. Let $\mathbf{P}$ denote the projection operator of X, i.e, $\mathbf{P}\mathbf{X} = \mathbf{X}$ for $\mathbf{X} \in X$. This $\mathbf{P}$ can then be seen as a 64×64 matrix, which allows us to write

$$\mathbf{Y} \sim \mathbf{C}\mathbf{P}\mathbf{X} = \mathbf{M}\mathbf{X} \qquad [Y_I \sim C_{IJ} P_{JL} X_L = M_{IL} X_L] \tag{5}$$

In this notation, $\mathbf{M} = \mathbf{C}\mathbf{P}$ corresponds to a 27×64 matrix which acts on $\mathbf{X}$. The rank of $\mathbf{M}$ is 17, which can be motivated as follows. In the case that $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3$ represent corresponding image coordinates, i.e., they are the images of the same 3D point $\mathbf{x}$, they must satisfy an incident relation which takes the algebraic form

$$\mathbf{Y} \cdot \mathbf{T} = 0 \qquad [Y_I T_I = 0] \tag{6}$$

where both $\mathbf{Y}$ and $\mathbf{T}$ represent 27-dimensional vectors. This relation is similar to the epipolar constraint defined by the fundamental matrix for two views. In [7] it is shown that

the set of point matching constraint tensors $\mathbf{T}$ forms a 10-dimensional vector space. Consequently, the codomain of $\mathbf{M}$ is a 27-dimensional space but it includes a 10-dimensional subspace which always is perpendicular to any image of an element in X, hence the range of $\mathbf{M}$ is of dimension $27-10=17$.

We now consider $\mathbf{M}^+$, the 64×27 pseudo-inverse matrix of $\mathbf{M}$. The codomain of $\mathbf{M}$ is X (20-dimensional) but the range is 17-dimensional (same dimension as the rank of $\mathbf{M}$) which implies there is a 3-dimensional space of X which cannot be reconstructed by $\mathbf{M}^+$. A basis for this "space of missing dimensions" can be found in a straight-forward way[1], it must be spanned by the three tensors $\mathbf{N}_k=\mathbf{n}_k\otimes\mathbf{n}_k\otimes\mathbf{n}_k, k=1,2,3$ where $\mathbf{n}_k$ is the homogeneous coordinates of the focal point of camera k: $\mathbf{C}_k\,\mathbf{n}_k=\mathbf{0}$. This can be summarized as: assuming that $\mathbf{Y}=\mathbf{y}_1\otimes\mathbf{y}_2\otimes\mathbf{y}_3$ is given by Equation (2) it follows that

$$\mathbf{M}^+\,\mathbf{Y}\sim\mathbf{X}+c_1\mathbf{N}_1+c_2\mathbf{N}_2+c_3\mathbf{N}_3 \qquad [M^+_{IJ}Y_J\sim X_I+c_kN_{k,I}] \tag{7}$$

where c_k are three scalars which depend on $\mathbf{x}$. Both sides of Equation (7) are a third order tensor on $\mathbb{R}^3$: $[M^+_{\alpha\beta\gamma J}Y_J\sim X_{\alpha\beta\gamma}+c_kN_{k,\alpha\beta\gamma}]$. It can be linearly combined with a second order tensor to produce a first order tensor; an element of $\mathbb{R}^4$. The inner product between the second order *auxiliary tensor* $\mathbf{A}$ and a third order tensor is denoted "$\mathbf{A}\odot$", as in

$$\mathbf{A}\odot\mathbf{M}^+\,\mathbf{Y}\sim\mathbf{A}\odot\mathbf{X}+c_k\,\mathbf{A}\odot\mathbf{N}_k \qquad [A_{\beta\gamma}M^+_{\alpha\beta\gamma J}Y_J\sim A_{\beta\gamma}X_{\alpha\beta\gamma}+c_kA_{\beta\gamma}N_{k,\alpha\beta\gamma}] \tag{8}$$

As a final step to reconstructing $\mathbf{x}$, we will choose the $\mathbf{A}$ such that $\mathbf{A}\odot\mathbf{N}_k=\mathbf{0}$ $[A_{\beta\gamma}N_{k,\alpha\beta\gamma}=0_\alpha]$ for $k=1,2,3$. Assuming that this can be done, it follows that

$$(\mathbf{A}\odot\mathbf{M}^+)\,\mathbf{Y}\sim\mathbf{A}\odot\mathbf{X} \qquad [A_{\beta\gamma}M^+_{\alpha\beta\gamma J}Y_J\sim A_{\beta\gamma}X_{\alpha\beta\gamma}=x_\alpha(x_\beta A_{\beta\gamma}x_\gamma)] \tag{9}$$

To see what this means, recall that $\mathbf{M}^+$ represents a 64×27 matrix. The 64-dimensional codomain of this mapping is the direct product $\mathbb{R}^4\otimes\mathbb{R}^4\otimes\mathbb{R}^4$ which, however, is reduced to only $\mathbb{R}^4$ after the inner product with $\mathbf{A}$. Consequently, $\mathbf{A}\odot\mathbf{M}^+$ can be represented by 4×27 matrix $\mathbf{K}$. Furthermore, $\mathbf{A}\odot\mathbf{X}$ can be rewritten as $\mathbf{x}[(\mathbf{x}\otimes\mathbf{x})\cdot\mathbf{A}]=\mathbf{x}(x_\beta A_{\beta\gamma}x_\gamma)$ which leads to

$$\mathbf{K}\,\mathbf{Y}\sim\mathbf{x} \qquad [K_{\alpha J}Y_J\sim x_\alpha] \tag{10}$$

if it can also be assumed that $(\mathbf{x}\otimes\mathbf{x})\cdot\mathbf{A}=x_\beta A_{\beta\gamma}x_\gamma\neq 0$. To summarize, assuming that an auxiliary tensor $\mathbf{A}$ can be found such that $\mathbf{A}\odot\mathbf{N}_k=\mathbf{0}$ for $k=1,2,3$, *and* $(\mathbf{x}\otimes\mathbf{x})\cdot\mathbf{A}\neq 0$ then Equation (10) implies that $\mathbf{x}$ is reconstructed from the linear mapping $\mathbf{K}=\mathbf{A}\odot\mathbf{M}^+$ acting on $\mathbf{Y}$, the tensor product of all three homogeneous image coordinates.

2.1 The auxiliary tensor A

It remains to show that a suitable $\mathbf{A}$ can be found and to do so it is here assumed that the cameras focal points are not co-linear. It should be noted that since $\mathbf{M}^+$ has its range in X, the space of completely symmetric third order tensor on $\mathbb{R}^4$, it follows that $\mathbf{A}$ can be restricted to symmetric second order tensors since any anti-symmetric part of $\mathbf{A}$ vanishes in the linear combination $\odot$. Let $\mathbf{p}_{00}$ be the dual homogeneous coordinates of the plane which includes all three focal points. In addition to $\mathbf{p}_{00}$ three other planes which are distinct from $\mathbf{p}_{00}$ can be selected such that each plane includes two focal points:

$$\mathbf{p}_{ij}\text{ includes focal points }\mathbf{n}_i\text{ and }\mathbf{n}_j\text{ for }ij=12,\ 23,\text{ and }31 \tag{11}$$

[1] This space is the intersection of the null space of $\mathbf{M}$ and X.

This amounts to four distinct planes represented by four linearly independent vectors $\mathbf{p}_{00}$, $\mathbf{p}_{12}$, $\mathbf{p}_{23}$, and $\mathbf{p}_{31}$ such that the following relations are valid

$$\mathbf{n}_k \cdot \mathbf{p}_{00} = 0, \quad k = 1,2,3 \qquad \mathbf{n}_i \cdot \mathbf{p}_{ij} = \mathbf{n}_j \cdot \mathbf{p}_{ij} = 0, \quad ij = 12,23,31 \tag{12}$$

From this construction of the four planes it follows directly that $\mathbf{A} = \mathbf{p}_{00} \otimes \mathbf{p}_{00}$ satisfies

$$\mathbf{A} \odot \mathbf{N}_k = (\mathbf{p}_{00} \otimes \mathbf{p}_{00}) \odot (\mathbf{n}_k \otimes \mathbf{n}_k \otimes \mathbf{n}_k) = \mathbf{n}_k (\mathbf{n}_k \otimes \mathbf{n}_k) \cdot (\mathbf{p}_{00} \otimes \mathbf{p}_{00}) = \mathbf{n}_k (\mathbf{n}_k \cdot \mathbf{p}_{00})^2 = 0 \tag{13}$$

for $k = 1,2,3$ and therefore is a feasible choice for the auxiliary tensor. Using the same arguments in combinations with Equation (12), the following six tensors are also feasible

$$\begin{aligned} &\mathbf{p}_{00} \otimes \mathbf{p}_{12} + \mathbf{p}_{12} \otimes \mathbf{p}_{00} \qquad &&\mathbf{p}_{00} \otimes \mathbf{p}_{23} + \mathbf{p}_{23} \otimes \mathbf{p}_{00} \\ &\mathbf{p}_{00} \otimes \mathbf{p}_{31} + \mathbf{p}_{31} \otimes \mathbf{p}_{00} \qquad &&\mathbf{p}_{12} \otimes \mathbf{p}_{23} + \mathbf{p}_{23} \otimes \mathbf{p}_{12} \\ &\mathbf{p}_{23} \otimes \mathbf{p}_{31} + \mathbf{p}_{31} \otimes \mathbf{p}_{23} \qquad &&\mathbf{p}_{31} \otimes \mathbf{p}_{12} + \mathbf{p}_{12} \otimes \mathbf{p}_{31} \end{aligned} \tag{14}$$

This leaves us with a in total seven linearly independent feasible choices of $\mathbf{A}$ and any linear combination of these is also a feasible choice. In summary, there is a 7-dimensional space of feasible auxiliary tensors $\mathbf{A}$ which implies that there is a 7-dimensional space of 4×27 matrices $\mathbf{K} = \mathbf{A} \odot \mathbf{M}^+$ which can reconstruct $\mathbf{x}$ from the image coordinates.

2.2 The matching condition

Before we move on to make practical use of the triangulation tensor $\mathbf{K}$ it should be noted that its construction implies that

$$\mathbf{K}\,\mathbf{T} = \mathbf{0} \qquad [K_{\alpha J} T_J = 0_\alpha] \tag{15}$$

for any three-view point matching constraint tensor $\mathbf{T}$. This follows from the fact that any $\mathbf{T}$ lies in the orthogonal complement of the range of $\mathbf{M}$ and consequently lie in the null space of $\mathbf{M}^+$. This *matching condition* on $\mathbf{K}$ will play an important role later on when the elements of $\mathbf{K}$ are adjusted to optimize reconstruction errors. This adjustment will in general imply that the resulting tensor $\mathbf{K}$ fails to meet the matching condition. This can be dealt with by readjusting $\mathbf{K}$ relative the matching condition:

$$\mathbf{K}' = \mathbf{K}\,(\mathbf{I} - \mathbf{P}') \qquad [K'_{\alpha J} = K_{\alpha L}(\delta_{LJ} - P'_{LJ})] \tag{16}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{P}'$ is the projection operator for the 10-dimensional space of three-view point matching constraint tensors, both 27×27 matrices. This construction of $\mathbf{K}'$ assures that $\mathbf{K}'\,\mathbf{T} = \mathbf{0}$.

The matching condition is not invariant with respect to coordinate transformations in the three image spaces. Transform each of the three image coordinates in accordance to $\bar{\mathbf{y}}_k = \mathbf{H}_k\,\mathbf{y}_k, k = 1,2,3$ where $\mathbf{H}_k$ is an arbitrary homography. This gives

$$\bar{\mathbf{Y}} = (\mathbf{H}_1 \otimes \mathbf{H}_2 \otimes \mathbf{H}_3)\,\mathbf{Y} = \mathbf{H}\,\mathbf{Y} \qquad \text{where} \qquad \mathbf{H} = \mathbf{H}_1 \otimes \mathbf{H}_2 \otimes \mathbf{H}_3 \tag{17}$$

Assuming that tensor $\mathbf{K}$ triangulates from $\mathbf{Y}$ to $\mathbf{x}$: $\mathbf{K}\,\mathbf{Y} \sim \mathbf{x}$, we require that a transformed $\bar{\mathbf{K}}$ instead can triangulate from $\bar{\mathbf{Y}}$ which is satisfied by $\bar{\mathbf{K}} = \mathbf{K}\,\mathbf{H}^{-1}$. Similarly, a three-view point matching constraint tensor $\mathbf{T}$ must transforms as $\bar{\mathbf{T}} = (\mathbf{H}^T)^{-1}\,\mathbf{T}$ to assure that Equation (6) is satisfied for $\bar{\mathbf{Y}}$ and $\bar{\mathbf{T}}$. This means that Equation (15) can be written

$$\mathbf{0} = \mathbf{K}\,\mathbf{T} = \bar{\mathbf{K}}\,\mathbf{H}\,\mathbf{H}^T\,\bar{\mathbf{T}} \tag{18}$$

Figure 1: The three camera views. The camera centers are approximately 1.3 m from the corner and approximately 0.5 m apart. Each plane is covered with a calibration patter of 10×15 small dots, approximately 18 mm between the rows and columns.

Consequently, Equation (15) can only be valid for certain choices of image coordinate systems in the three views. Choosing suitable coordinate systems in which the matching condition is satisfied for $\mathbf{K}$ is one of the key elements of the optimization method presented in the following section.

2.3 Short summary of how to construct K

Here we summarize the computations needed to compute the triangulation tensor $\mathbf{K}$. As prerequisites we need the three camera matrices $\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3$ and a feasible auxiliary tensor $\mathbf{A}$. The latter can be found by first determining the three camera focal points, find four planes $\mathbf{p}_{00}, \mathbf{p}_{12}, \mathbf{p}_{23}, \mathbf{p}_{31}$ and construct the seven linearly independent second order symmetric tensors as described in Section 2.1. A feasible $\mathbf{A}$ is given as any linear combination of these seven tensors. In practice, however, we should choose $\mathbf{A}$ such that $\mathbf{A} \cdot (\mathbf{x} \otimes \mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$ is non-zero for all $\mathbf{x}$ which are visible in the scene since this factor is multiplied onto the reconstructed 3D homogeneous coordinates of $\mathbf{x}$. Finally, we also need $\mathbf{P}$, the 64×64 matrix which is the projection operator for X, the space of completely symmetric third order tensors on $\mathbb{R}^4$.

To construct $\mathbf{K}$, we first form $\mathbf{C}_1 \otimes \mathbf{C}_2 \otimes \mathbf{C}_3$ as the Kronecker product of the three matrices, represented by a 27×64 matrix and compute $\mathbf{M} = \mathbf{C}\,\mathbf{P}$. Next, the Moore-Penrose pseudo-inverse of $\mathbf{M}$ is formed by computing its singular value decomposition: $\mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ where we assume a *truncated* form of $\mathbf{S}$, i.e., it is a 17×17 full rank diagonal matrix (remember that 10 singular values are zero, see Section 2). This gives the pseudo-inverse as $\mathbf{M}^+ = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T$, which can be represented as a 64×27 matrix. Finally, we perform the inner product with $\mathbf{A}$ to get $\mathbf{A} \odot \mathbf{M}^+$. This can be done by reshaping $\mathbf{A}$ to a 16-dimensional vector and $\mathbf{M}^+$ to a 16×108 matrix and multiply $\mathbf{A}$ from left. The resulting 108-dimensional vector is then reshaped back to the 4×27 matrix $\mathbf{K}$.

3 Euclidean 3D optimization: experiments and refinement of K

In this section we will apply the triangulation tensor $\mathbf{K}$ described in the last section on a set of real 3D/2D data and evaluate its performance. As will be noticed it matters which

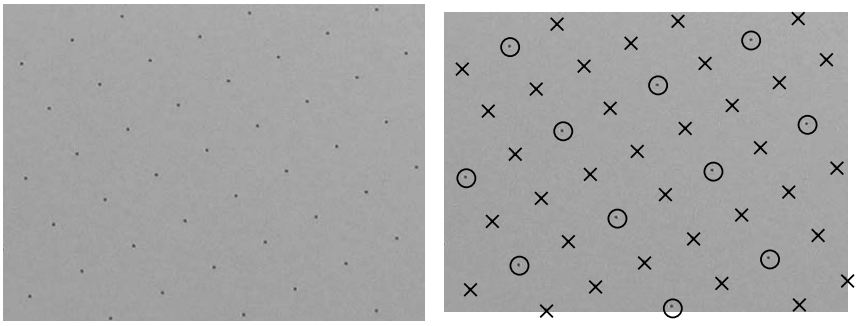

Figure 2: A closeup of one of the images. Right: calibration points are marked with circles and evaluation points with crosses.

auxiliary tensor $\mathbf{A}$ that is used for the construction of $\mathbf{K}$ in the case that non-ideal data is used. The performance of $\mathbf{K}$ in terms of 3D reconstruction error is also compared to the optimal method described in [9, 1]. At first sight, the optimal method is significantly better. However, by performing an iterative two-step adjustment of $\mathbf{K}$ to minimize the 3D error, $\mathbf{K}$ can be optimized to produce errors at the same level as the optimal method.

A 3D corner of three perpendicular planes, each covered with a regular grid of 150 points, is viewed from three distinct points with a 1944×2592 digital camera, see Figure 1 and Figure 2, left. The 450 3D points have positions which are measured manually with an accuracy of approximately 0.5 mm. The corresponding points in the three images are determined with integer value precision by means of local minima detection. These positions are affected both by estimation noise, which we can assume to be in the order of 1-2 pixels, and by geometric noise from the lens distortion. The latter is compensated for by assuming a radial tangens-based distortion [2] which is optimized to make the lines present in the point patters as straight as possible. The result is a set of distortion compensated image points in the three images $\{\mathbf{y}_1^{(l)}\}, \{\mathbf{y}_2^{(l)}\}$ and $\{\mathbf{y}_3^{(l)}\}$, corresponding to the set of 3D points $\{\mathbf{x}^{(l)}\}$. Approximately one fourth of the points are then used for calibration of cameras and later also for $\mathbf{K}$, and the remaining points for evaluation. See Figure 2, right. Using the calibration sets, the camera matrices are estimated with the Gold Standard algorithm [3], producing a mean and maximum 2D L_1-residual error at 0.50 and 1.5 pixels, respectively (although Gold Standard optimizes relative the L_2 error!).

To get a reference for the performance of the triangulation tensor, we start by applying the optimal three-view triangulation method [9, 1] on the evaluation sets. The L_1 residual error in the 3D space is measured and the mean and maximum is presented in the first row of Table 1. Next, we compute a triangulation tensor $\mathbf{K}$ according to Section 2.3. The auxiliary tensor $\mathbf{A}$ is here chosen simply as $\mathbf{p}_{00} \otimes \mathbf{p}_{00}$, where $\mathbf{p}_{00}$ is the plane which includes all three camera focal points. Given the configuration of the cameras, this plane is not visible in any of the three views and the corresponding $\mathbf{A}$, therefore, will not produce the degenerate case $\mathbf{KY} = \mathbf{0}$. This triangulation tensor is here denoted $\mathbf{K}_{00}$, and it is used to reconstruct the 3D points from the evaluation sets of image points with errors presented in the second row of Table 1. As cam be seen, the mean error and in particular the maximum error are significantly larger than for the optimal method.

Method	Mean L_1	Max L_1
Optimal [9, 1]	0.25	0.79
$\mathbf{K}_{00}$, $\mathbf{A}$ based only on $\mathbf{p}_{00}$	1.61	15.0
$\mathbf{K}_{\text{opt}}$, optimizing $\mathbf{A}$	0.36	1.29
$\mathbf{K}_1$, optimized, iteration 1	0.33	0.94
$\mathbf{K}_{1,\text{readj}}$, enforcing the matching condition	0.24	0.82
$\mathbf{K}_2$, optimized, iteration 2	0.204	0.81
$\mathbf{K}_{2,\text{readj}}$, enforcing the matching condition	0.200	0.81

Table 1: 3D residual error.

The poor performance produced by $\mathbf{K}_{00}$ may be related to choosing $\mathbf{A}$ too arbitrary from the 7-dimensional space of feasible tensors described in Section 2.1. Instead we are in a position to determine which feasible $\mathbf{A}$ that minimizes the 3D L_1 residual error for the calibration set and choose that $\mathbf{A}$ for constructing an optimized triangulation tensor $\mathbf{K}_{\text{opt}}$, which then is applied on the evaluation sets. The corresponding errors are presented in the third row of Table 1. We see a significant reduction in the errors even though they are not at the same low level as for the optimal method. In fact, these figures are comparable to the optimal method for two-view triangulation[2].

In order to reduce the error even further we need to free ourselves from the idea that $\mathbf{K}$ necessarily has to be computed from the camera matrices which, themselves, have been estimated from the calibration data sets. We can simply try to determine the elements of $\mathbf{K}$ such that the resulting tensor minimizes the 3D L_1 residual error. This corresponds to a non-linear minimization problem which has to be solved with some iterative method, and we can use $\mathbf{K}_{\text{opt}}$ as the initial value for this optimization. Compare this to the Gold Standard method for camera calibration [3], with the difference that here we determine the multi-linear transformation from 2D+2D+2D image coordinates to 3D coordinates (instead of from 3D to 2D) and the error function uses the total 3D L_1-norm (instead of the total 2D L_2-norm). The resulting tensor is denoted $\mathbf{K}_1$ and the corresponding errors are presented in the fourth row of Table 1. It gives a reduction of the errors but it is still some distance to the optimal method.

So far we have focused on 3D errors, whereas the optimal method instead minimizes the total L_2-reprojection error in the image domain by adjusting the image coordinates so that they satisfy all three-view point matching constraints[3]. What about trying to do both? Given the three camera matrices, a basis of three-view point matching constraints can be determined [7], and a quick investigation of $\mathbf{K}_1$ shows that it does not satisfy the matching condition described in Section 2.2. In particular this is the case if we transform the images to the *normalized coordinates* described in [4]. We will use normalized coordinates since they provide a better correspondence between operations in 2D projective spaces and 2D Euclidean errors. Consequently, we enforce the matching condition on $\mathbf{K}_1$ by computing $\mathbf{K}'_1$ in accordance to Equation (16). The corresponding 3D errors for $\mathbf{K}'_1$ are presented in the fifth row of Table 1 and this operation clearly brings us to something that is comparable

[2]Optimal three-view triangulation gives approx. 30% better accuracy than optimal two-view triangulation on the same data.

[3]Even though the three-view point matching constraints never appear explicitly in [9], this is exactly what happens since all adjusted image points refer to the same 3D point.

to the optimal method.

What we have done so far can be described as two steps: (1) given some initial value of $\mathbf{K}$, its elements are adjusted to reduce 3D errors, (2) this $\mathbf{K}$ is again adjusted, now to meet the matching condition in normalized image coordinates. Since the latter step has reduced the target function of the former step, it seem reasonable that it can be even further reduced by using $\mathbf{K}'_1$ as the initial value of the first step to produce $\mathbf{K}_2$, and it may be even further reduced by again enforcing the matching condition in $\mathbf{K}'_2$. This is indeed the case, as can be seen in rows six and seven of Table 1. At least the mean error decreases below the optimal method, although the maximum error remains slightly higher. Furthermore, the step from $\mathbf{K}_2$ to $\mathbf{K}'_2$ gives only a minor improvement which implies that there is little to be gained by performing further iterations of the two steps.

4 Summary

This paper presents a computational approach to three-view triangulation in terms of a multi-linear mapping $\mathbf{K}$ applied to the three homogeneous image coordinates which directly gives the homogeneous coordinates $\mathbf{x}$ of the corresponding 3D point, Section 2. We can interpret this as: $\mathbf{x}$ is given by a linear mapping on $\mathbf{Y}$, the tensor product of the image coordinates. This means that once $\mathbf{K}$ is determined the computation of $\mathbf{x}$ can be made very efficiently, in particular relative to optimal approaches [9, 1], making the proposed method relevant for real-time applications, e.g., in terms of GPGPU implementations. $\mathbf{K}$ can be computed from the three camera matrices and an auxiliary tensor $\mathbf{A}$ which lies in a 7-dimensional space, described in Section 2.1. Even if we choose $\mathbf{A}$ to minimize 3D L_1 residual errors, the resulting $\mathbf{K}_{\text{opt}}$ has larger errors than optimal triangulation, but it can be used as initial value for a two-step iterative approach which adjusts all elements of $\mathbf{K}$ to minimize both 3D and 2D errors. The first step adjusts the elements of $\mathbf{K}$ to reduce the 3D error. The second step readjusts the tensor to ensure that is satisfies the matching condition in normalized image coordinates, Section 2.2, implying that $\mathbf{Y}$ is adjusted to satisfy all three-view point matching constraints before being mapped by $\mathbf{K}$. These two steps can then be iterated a few time (here only two times) to reduce the 3D L_1 error to the same level as the optimal method. This means that we can perform a one-shot optimization of $\mathbf{K}$ based on a set of 3D/2D calibration points and obtain a tensor which efficiently and with 3D error comparable to the optimal method can reconstruct 3D points, assuming that the calibration set is representative for the points to be reconstructed in the later step.

The result presented here for three view triangulation corresponds well to the stereo case presented in [8]. In both cases is it possible to carry out a calibration of the triangulation tensor which produces 3D errors similar to what the optimal methods gives, even though the 2D reprojection error is significantly larger for the proposed method.

5 Acknowledgement

The work reported in this paper has been made within the VISCOS project, funded by the Swedish Foundation for Strategic Research (SSF). The Matlab code which implements the triangulation method described in [1] and used in the experiments presented in Section 3 has kindly been provided by Klas Josephson at the Mathematical Imaging Group, Lund University.

References

[1] Martin Byröd, Klas Josephson, and Kalle Åström. Improving numerical accuracy of Gröbner basis polynomial equation solvers. In *Proceedings of 11th International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007.

[2] Frédéric Devernay and Olivier Faugeras. Straight lines have to be straight: Automatic calibration and removal of distortion from scenes of structured environments. *Machine Vision and Applications*, 13:14–24, 2001.

[3] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision, 2nd ed*. Cambridge University Press, 2003.

[4] Richard I. Hartley. In defence of the 8-point algorithm. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 19(6):580–593, 1997.

[5] Richard I. Hartley and Peter Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.

[6] Klas Nordberg. A linear mapping for stereo triangulation. In *Proceedings of 15th Scandinavian Conference on Image Analysis (SCIA)*, pages 838–847, Aalborg, Denmark, June 2007. Springer LNCS Vol. 4522.

[7] Klas Nordberg. Point matching constraints in two and three views. In *Proceedings of Annual Symposium of the German Association for Pattern Recognition (DAGM) 2007*, Heidelberg, Germany, September 2007. Springer LNCS Vol. 4713.

[8] Klas Nordberg. Efficient triangulation based on 3d euclidean optimization. In *Proceedings of International Conference on Pattern Recognition*, December 2008.

[9] Henrik Stewénius, Fredrik Schaffalitzky, and David Nistér. How hard is 3-view triangulation really. In *Proceedings of 10th International Conference on Computer Vision*, San Diego, USA, 2005.