A Linear Mapping for Stereo Triangulation

Klas Nordberg

Computer Vision Laboratory
Department of Electrical Engineering
Linköping University

Abstract. A novel and computationally simple method is presented for triangulation of 3D points corresponding to the image coordinates in a pair of stereo images. The image points are described in terms of homogeneous coordinates which are jointly represented as the outer products of these homogeneous coordinates. This paper derives a linear transformation which maps the joint representation directly to the homogeneous representation of the corresponding 3D point in the scene. Compared to the other triangulation methods this approach gives similar reconstruction error but is numerically faster, since it only requires linear operations. The proposed method is projective invariant in the same way as the optimal method of Hartley and Sturm. The methods has a "blind plane"; a plane through the camera focal points which cannot be reconstructed by this method. For "forward-looking" camera configurations, however, the blind plane can be placed outside the visible scene and does not constitute a problem.

1 Introduction

Reconstruction of 3D points from stereo images is a classical problem. The methods which address the problem fall coarsely into two classes; dense and sparse. The first class normally assumes a restricted camera configuration with camera viewing directions which are approximately parallel and a short camera baseline, resulting in approximately horizontal and relatively small displacements between corresponding points in the two images. As a consequence, a displacement or disparity field can be estimated between the two images for all points, although with low reliability for image points which, e.g. are located in a constant intensity region. Several methods for solving the disparity estimation problem have been proposed in the literature and an overview is presented, for example, in [9]. Given the estimated disparity field, the 3D reconstruction can be done using the inverse proportionality between depth in the scene and the disparity.

The second class allows general camera configurations, with the main restriction that they cameras depict a common scene. For a typical scene, however, this implies that only a smaller set of point may be visible in both images and can be robustly detected as corresponding points in both images. The proposed methods normally solve the reconstruction problem in two steps. First, two sets of points in each of the images are determined, and further investigated to produce point pairs where each pair corresponds to the same point in the 3D scene.

B.K. Ersbøll and K.S. Pedersen (Eds.): SCIA 2007, LNCS 4522, pp. 838–847, 2007.

A triangulation procedure can then be applied on each such pair to reconstruct the 3D point. In this paper we assume that the correspondence problem has been solved, and instead deal with the triangulation procedure.

In order to solve the triangulation problem, some related issues must first be addressed. Most solutions assume that the cameras involved in producing the stereo images can sufficiently accurately be modelled as pin-hole cameras. This implies that the mapping from 3D points to 2D coordinates in each of the two images can be described as a linear mapping on homogeneous representations of both 3D and 2D coordinates. Let $\mathbf{x}$ be a homogeneous representation of the coordinates of a 3D point (a 4-dimensional vector), let $\mathbf{y}_k$ be a homogeneous representation of the corresponding image coordinate in image k, (a 3-dimensional vector), and let $\mathbf{C}_k$ be the linear mapping which describes the mapping of camera k (a 3×4 matrix). The pin-hole camera model then implies that

$$\mathbf{y}_k \sim \mathbf{C}_k\,\mathbf{x}, \qquad k = 1, 2 \tag{1}$$

where $\sim$ denotes equality up to a scalar multiplication. Notice, that in this particular relation, the scalar in is only dependent on $\mathbf{x}$. The inverse mapping can be written as

$$\mathbf{x} \sim \mathbf{C}_k^{+}\mathbf{y}_k + \lambda_k\,\mathbf{n}_k, \qquad \lambda_k \in R \tag{2}$$

where $\mathbf{C}_k^{+}$ is the pseudo-inverse of $\mathbf{C}_k$ and $\mathbf{n}_k$ is the homogeneous representation of the focal point of camera k, i.e.,

$$\mathbf{C}_k\,\mathbf{n}_k = \mathbf{0} \tag{3}$$

From Equation (2) follows that the original 3D point must lie on a *projection line* through the image point and the focal point. In the following, we will assume that both camera matrices have been determined with a sufficient accuracy to be useful in the following computations.

Another issue is the so so-called epipolar constraint. It implies that two corresponding image points must satisfy the relation

$$\mathbf{y}_1^T\mathbf{F}\,\mathbf{y}_2 = 0 \tag{4}$$

where $\mathbf{F}$ is the so-called *fundamental matrix* which is determined from the two camera matrices [5]. Intuitively, we can think of this relation as a condition on $\mathbf{y}_1$ and $\mathbf{y}_2$ to assure that the corresponding projection lines intersect at the point $\mathbf{x}$. In practice, however, there is no guarantee that $\mathbf{y}_1$ and $\mathbf{y}_2$ satisfy Equation (4) exactly. For example, many point detection methods are based on finding local maxima or minima of some function, e.g., [3], typically producing integer valued image coordinates. As a consequence, the two projection lines do not always intersect.

Even if we assume that the cameras can be modeled as pin-hole cameras whose matrices are known with sufficient degree accuracy and that in some way or another all pairs of corresponding image points have been modified to satisfy Equation (4), we now face the ultimate problem of triangulation: finding the intersecting point of the two projection lines for each pair of corresponding image points. In principle, this is a trivial problem had it not been for the fact that Equation (4) is not always satisfied. The conceptually easiest approach is the *mid-point method* were we seek the mid-point of the shortest line segment which joins the projection lines of the two image points [1]. From an algebraic point of view, the problem can also be solved by using Equation (1) to obtain

$$\begin{aligned} \mathbf{y}_1 \times (\mathbf{C}_1\, \mathbf{x}) &= \mathbf{0} \\ \mathbf{y}_2 \times (\mathbf{C}_2\, \mathbf{x}) &= \mathbf{0} \end{aligned} \qquad (5)$$

where "×" denotes the vector cross product. These relations imply that we can establish six linear expressions in the elements of $\mathbf{x}$, an over-determined system of equations. On the other hand, if we know that $\mathbf{y}_1$ and $\mathbf{y}_2$ satisfy Equation (4) then there must be a unique solution $\mathbf{x}$ of Equation (5), disregarding a scalar multiplication. By rewriting Equation (5) as

$$\mathbf{M}\, \mathbf{x} = \mathbf{0} \qquad (6)$$

we can either find $\mathbf{x}$ as the right singular vector of $\mathbf{M}$ corresponding to the smallest singular value, or by solving for the non-homogeneous coordinates of the 3D point from the corresponding 6×3 inhomogeneous equation using a least squares solution. These are the *homogeneous* and the *inhomogeneous* triangulation methods [5].

All three methods appear to work in most situations, but they have some practical differences, notably that the resulting 3D point is not the same for all three methods in the case that Equation (4) is not satisfied. The implementations of all three methods are relatively simple but they do not provide a simple closed form expression for $\mathbf{x}$ in terms of $\mathbf{y}_1, \mathbf{y}_2$. Also, the basic form of both the mid-point method and the inhomogeneous method cannot provide a robust estimate of the 3D point in the case that it is at a large of infinite distance from the camera.

There is also a difference in the accuracy of each method. This can be defined in terms of the 3D distance between the resulting 3D point and the correct 3D point, but from a statistical point of view it can also be argued that the accuracy should be defined in terms of the Euclidean 2D distances of the projection of these 3D points. This leads to the *optimal method* for triangulation which seeks two subsidiary image points $\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2$ which are at total smallest squared distance from $\mathbf{y}_1, \mathbf{y}_2$, measured in the 2D image planes, which in addition satisfy $\hat{\mathbf{y}}_1^T \mathbf{F}\, \hat{\mathbf{y}}_2 = 0$. Once $\hat{\mathbf{y}}_1, \hat{\mathbf{y}}_2$ are determined, any of the above mentioned methods can then produce the "optimal" estimate of $\mathbf{x}$. A computational approach for determining the subsidiary image points is presented in [7]. Although this method is not iterative, it is of relatively high complexity and does not give $\mathbf{x}$ as a closed form expression. On the other hand, this method can be shown to be projective invariant, meaning that the resulting point $\mathbf{x}$ is invariant to any projective transformation of the 3D space.

In summary, there exist a larger number of methods for solving the sparse triangulation problem, even besides the "standard" methods presented above. For an overview of sparse triangulation methods, [7,5] serve as good starting points. Most of the recent work in this area derives reconstruction methods based on various cost functions, for example see [4,8].

This paper takes a slightly different view on triangulation by leaving cost functions and optimization aside and instead focus on the basic problem of finding an algebraic inverse of the combined camera mapping from a 3D point to the two corresponding image points. It proves the existence of a *closed form expression* for the mapping from $\mathbf{y}_1$ and $\mathbf{y}_2$ to $\mathbf{x}$. The expression is in the form of a *second order polynomial* in the elements of $\mathbf{y}_1$ and $\mathbf{y}_2$, or more precisely, a linear mapping on the outer product $\mathbf{y}_1\mathbf{y}_2^T$. Beside its simple computation, another advantage is that it can be shown to be projective invariant, although this aspect is not discussed in detail here. An experimental evaluation of the proposed method shows that, on that specific data set, it has an reconstruction error which is comparable to the other standard methods, including optimal triangulation.

2 Derivation of a Reconstruction Operator

2.1 The Mapping to Y

Let $\mathbf{y}_k$ be the homogeneous representation of a point in image k, given by Equation (1). We can write this expression in element form as

$$y_{ki} = \alpha_k(\mathbf{x})\, \mathbf{c}_{ki} \cdot \mathbf{x} \tag{7}$$

where $\mathbf{c}_{ki}$ is the i-th row of camera matrix k and the product in the right-hand side is the inner product between the vectors $\mathbf{c}_{kl}$ and $\mathbf{x}$. Now, form the outer or tensor product between $\mathbf{y}_1$ and $\mathbf{y}_2$. The result can be seen as a 3×3 matrix $\mathbf{Y} = \mathbf{y}_1\mathbf{y}_2^T$. The elements of $\mathbf{Y}$ are given by

$$\begin{aligned} Y_{ij} &= y_{1i}\, y_{2j} = \alpha_1(\mathbf{x})\, \alpha_2(\mathbf{x})\, (\mathbf{c}_{1i} \cdot \mathbf{x})(\mathbf{c}_{2i} \cdot \mathbf{x}) = && (8)\\ &= \alpha_1(\mathbf{x})\, \alpha_2(\mathbf{x})\, (\mathbf{c}_{1i}\mathbf{c}_{2j}^T) \cdot (\mathbf{x}\mathbf{x}^T) = \alpha_1(\mathbf{x})\, \alpha_2(\mathbf{x})\, (\mathbf{c}_{1i}\mathbf{c}_{2j}^T) \cdot \mathbf{X} && (9) \end{aligned}$$

We can interpret this as: element Y_{ij} is given by an inner product between $\mathbf{c}_{1i}\mathbf{c}_{2j}^T$ and $\mathbf{X} = \mathbf{x}\mathbf{x}^T$, defined by the previous relations. Notice that $\mathbf{X}$ is always a symmetric 4×4 matrix, which means that we can rewrite Y_{ij} as

$$Y_{ij} = \frac{\alpha_1(\mathbf{x})\, \alpha_2(\mathbf{x})}{2}\, (\mathbf{c}_{1i}\mathbf{c}_{2j}^T + \mathbf{c}_{2j}\mathbf{c}_{1i}^T) \cdot \mathbf{X} = \alpha_1(\mathbf{x})\, \alpha_2(\mathbf{x})\, \mathbf{B}_{ij} \cdot \mathbf{X} \tag{10}$$

where each $\mathbf{B}_{ij} = (\mathbf{c}_{1i}\mathbf{c}_{2j}^T + \mathbf{c}_{2j}\mathbf{c}_{1i}^T)/2$ is a symmetric 4×4 matrix.

2.2 The Set $\mathbf{B}_{ij}$

Let S denote the vector space of symmetric 4×4 matrices. Notice that $\mathbf{X} \in S$ and $\mathbf{B}_{ij} \in S$. The question now is: what kind of a set is $\mathbf{B}_{ij}$? Obviously, it cannot be a basis of S; the space is 10-dimensional and there are only 9 matrices. Recall that $\mathbf{n}_k$ is the homogeneous representation of the focal point for camera k, i.e.,

$\mathbf{C}_k\,\mathbf{n}_k = \mathbf{0}$. We assume that $\mathbf{n}_1 \neq \mathbf{n}_2$ (as projective elements). It then follows that

$$\mathbf{B}_{ij} \cdot (\mathbf{n}_k \mathbf{n}_k^T) = (\mathbf{c}_{1i} \cdot \mathbf{n}_k)(\mathbf{c}_{2j} \cdot \mathbf{n}_k) = 0 \tag{11}$$

i.e., $\mathbf{Q}_k = \mathbf{n}_k \mathbf{n}_k^T$ is perpendicular to all matrices $\mathbf{B}_{ij}$ for $k = 1, 2$. This implies that the 9 matrices at most span an 8-dimensional space, i.e., there exists at least one set of coefficients $\tilde{F}_{ij}$ such that

$$\sum_{ij} \tilde{F}_{ij}\,\mathbf{B}_{ij} = \mathbf{0} \tag{12}$$

Consider the expression

$$\sum_{ij} \tilde{F}_{ij} Y_{ij} = y_{1i}\,y_{2j}\,\tilde{F}_{ij} = \mathbf{y}_1^T \tilde{\mathbf{F}}\,\mathbf{y}_2 \tag{13}$$

where $\tilde{\mathbf{F}}$ is a 3 × 3 matrix with elements $\tilde{F}_{ij}$. We can now insert Equation (10) into the left-hand side of the last equation and get

$$\alpha_1(\mathbf{x})\,\alpha_2(\mathbf{x}) \sum_{ij} \tilde{F}_{ij}(\mathbf{B}_{ij} \cdot \mathbf{X}) = \alpha_1(\mathbf{x})\,\alpha_2(\mathbf{x})\,(\sum_{ij} \tilde{F}_{ij}\mathbf{B}_{ij}) \cdot \mathbf{X} = \mathbf{0} \cdot \mathbf{X} = 0 \tag{14}$$

Consequently, the right-hand side of Equation (13) vanishes, which is equivalent to the statement made in Equation (4), i.e., we can identify $\tilde{\mathbf{F}}$ with the fundamental matrix $\mathbf{F}$. Since this matrix is unique (disregarding scalar multiplications), it follows that the set of 9 matrices $\mathbf{B}_{ij}$ spans an (9 - 1 = 8)-dimensional subspace of S, denoted S_c. The matrices $\mathbf{B}_{ij}$ therefore form a *frame* [2] rather than a basis of S_c. Furthermore, $\mathbf{Q}_1$ and $\mathbf{Q}_2$ span the 2-dimensional subspace of S which is perpendicular to S_c.

2.3 The Dual Frame

Let us now focus on the subspace S_c. First of all, any matrix $\mathbf{S} \in S_\mathrm{c}$ can be written as a linear combination of the 9 frame matrices $\mathbf{B}_{ij}$. Since these matrices are linearly dependent such a linear combination is not unique, but a particular linear combination can be found as

$$\mathbf{S} = \sum_{ij} (\mathbf{S} \cdot \tilde{\mathbf{B}}_{ij})\,\mathbf{B}_{ij} \tag{15}$$

where $\tilde{\mathbf{B}}_{ij}$ is the dual frame relative to $\mathbf{B}_{ij}$. In this case, we compute the dual frame in the following way:

1. Reshape each $\mathbf{B}_{ij}$ to a 16-dimensional vector $\mathbf{a}_I$ with label $I = i + 3j - 3$: , i.e., there are 9 such vectors.
2. Construct a 16×9 matrix $\mathbf{A}$ with each $\mathbf{a}_I$ in its columns.

3. Compute an SVD of $\mathbf{A}$, $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$, where $\mathbf{S}$ is a 9×9 diagonal matrix with the singular values. According to the discussion above, exactly one singular value must vanish since $\mathbf{f}\,\mathbf{A} = \mathbf{0}$ where $\mathbf{f}$ is the is the fundamental matrix reshaped to a 9-dimensional vector. Consequently, $\mathbf{A} = \tilde{\mathbf{U}}\,\tilde{\mathbf{S}}\,\tilde{\mathbf{V}}^T$, where $\tilde{\mathbf{S}}$ is an 8×8 diagonal matrix with only non-zero singular values.
4. Form the 16×9 matrix $\tilde{\mathbf{A}} = \tilde{\mathbf{U}}\,\tilde{\mathbf{S}}^{-1}\,\tilde{\mathbf{V}}^T$.
5. The columns of $\tilde{\mathbf{A}}$ now contains the dual frame. The corresponding matrices $\tilde{\mathbf{B}}_{ij}$ can be obtained by doing the inverse operations of steps 2 and 1.

By constructing $\tilde{\mathbf{A}}$ in this way it follows that $\mathbf{A}^T\tilde{\mathbf{A}}$ is an identity mapping on the subspace S_c, when the elements of this space are reshaped according to step 1 above.

2.4 Triangulation

We will now choose $\mathbf{S}$ in particular ways. Let $\mathbf{p}$ be the dual homogeneous representation of a plane which passes through the focal points of the two cameras. This implies that $\mathbf{p}\cdot\mathbf{n}_1 = \mathbf{p}\cdot\mathbf{n}_2 = 0$. Let $\mathbf{r}$ be an arbitrary 4-dimensional vector, and form the 4×4 matrix

$$\mathbf{S} = \mathbf{p}\mathbf{r}^T + \mathbf{r}\mathbf{p}^T \tag{16}$$

Clearly, $\mathbf{S} \in S$, but it is also the case that $\mathbf{S} \in S_\mathrm{c}$. This follows from

$$\mathbf{S}\cdot\mathbf{Q}_k = \mathbf{S}\cdot(\mathbf{n}_k\mathbf{n}_k^T) = 2\,(\mathbf{n}_k\cdot\mathbf{p})(\mathbf{n}_k\cdot\mathbf{r}) = 0 \tag{17}$$

which implies that $\mathbf{S}$ is perpendicular to $\mathbf{Q}_1$ and $\mathbf{Q}_2$. Consequently, this $\mathbf{S}$ can be written as in Equation (15). Consider the expression $\mathbf{S}\cdot\mathbf{X}$. By inserting this into Equation (15) and with the help of Equation (10) we get

$$\mathbf{S}\cdot\mathbf{X} = \sum_{ij}(\mathbf{S}\cdot\tilde{\mathbf{B}}_{ij})(\mathbf{B}_{ij}\cdot\mathbf{X}) = \frac{2}{\alpha_1(\mathbf{x})\,\alpha_2(\mathbf{x})}\sum_{ij}(\mathbf{S}\cdot\tilde{\mathbf{B}}_{ij})\,Y_{ij} \tag{18}$$

If we instead insert it into Equation (16), we get

$$\mathbf{S}\cdot\mathbf{X} = (\mathbf{p}\mathbf{r}^T + \mathbf{r}\mathbf{p}^T)\cdot(\mathbf{x}\mathbf{x}^T) = 2\,(\mathbf{x}\cdot\mathbf{p})(\mathbf{x}\cdot\mathbf{r}) \tag{19}$$

and by combining Equations (18) and (19)

$$\sum_{ij}(\mathbf{S}\cdot\tilde{\mathbf{B}}_{ij})\,Y_{ij} = \alpha_1(\mathbf{x})\,\alpha_2(\mathbf{x})\,(\mathbf{x}\cdot\mathbf{p})(\mathbf{x}\cdot\mathbf{r}) \tag{20}$$

Let $\mathbf{e}_l$ be the standard basis of R^4: $\mathbf{e}_l\cdot\mathbf{x} = x_l$, where x_l is the l-th element of $\mathbf{x}$. Define 4 matrices $\mathbf{S}_l$ according to

$$\mathbf{S}_l = \mathbf{p}\mathbf{e}_l^T + \mathbf{e}_l\mathbf{p}^T \tag{21}$$

where $\mathbf{r}$ now is replaced by $\mathbf{e}_l$ to produce $\mathbf{S}_l$ from $\mathbf{S}$ in Equation (16). As a consequence, Equation (20) becomes

$$\sum_{ij}(\mathbf{S}_l\cdot\tilde{\mathbf{B}}_{ij})\,Y_{ij} = (\mathbf{x}\cdot\mathbf{p})(\mathbf{x}\cdot\mathbf{e}_l) = \alpha_1(\mathbf{x})\,\alpha_2(\mathbf{x})\,(\mathbf{x}\cdot\mathbf{p})\,x_l \tag{22}$$

Notice that the factor $\alpha_1(\mathbf{x})\,\alpha_2(\mathbf{x})\,(\mathbf{x}\cdot\mathbf{p})$ is independent of l. Set $K_{lij} = \mathbf{S}_l \cdot \tilde{\mathbf{B}}_{ij}$. This is a $4\times 3\times 3$ array of scalars which can be seen as a linear transformation or an operator which maps $\mathbf{Y}$ to $\mathbf{x}$:

$$\mathbf{K}\,\mathbf{Y} = \alpha_1(\mathbf{x})\,\alpha_2(\mathbf{x})\ \ (\mathbf{x}\cdot\mathbf{p})\,\mathbf{x} \tag{23}$$

The factor $\alpha_1(\mathbf{x})\ \alpha_2(\mathbf{x})\ (\mathbf{x}\cdot\mathbf{p})$ vanishes when the 3D point $\mathbf{x}$ is in the plane $\mathbf{p}$. Assuming that this is not the case, we can disregard this factor and write $\mathbf{K}\,\mathbf{Y} \sim \mathbf{x}$. It should be noticed, however, that the existence of this factor in the derivations implies that the proposed method cannot reconstruct 3D points if they lie in or sufficiently close to the *blind plane* $\mathbf{p}$.

The blind plane can be a problem if the camera configuration is such that the cameras "see" each other, that is, the 3D line (base-line) which intersects both focal points is visible to both cameras. In this case, a plane $\mathbf{p}$ cannot be chosen which is not visible to both cameras. On the other hand, if both cameras are "forward-looking" in the sense that they are not seeing each other, it is possible to choose $\mathbf{p}$ so that it is not visible to the cameras. In this case, reconstruction of points in the blind plane will never occur in practice.

The reconstruction formula $\mathbf{x} \sim \mathbf{K}\,\mathbf{Y}$ is interesting since it means that $\mathbf{X}$ can be computed only by multiplying a 4×9 matrix on the 9-dimensional vector $\mathbf{Y}$ which, in turn is given by reshaping the outer product of $\mathbf{y}_1$ and $\mathbf{y}_2$. Alternatively, $\mathbf{x}$ can be computed by first reshaping $\mathbf{K}$ as a 12×3 matrix that is multiplied on $\mathbf{y}_1$, resulting in a 12-dimensional vector $\mathbf{x}'$. $\mathbf{x}$ is then obtained by reshaping $\mathbf{x}'$ as a 4×3 matrix and multiply it on $\mathbf{y}_2$. The computational complexity for computing $\mathbf{x}$ is therefore not more than 32 additions and 45 (or 48) multiplications, depending on which approach is used.

3 Experiments

A set of 72 3D points is defined by a calibration pattern placed on three planes at straight angels to each other. These points are viewed by two cameras which

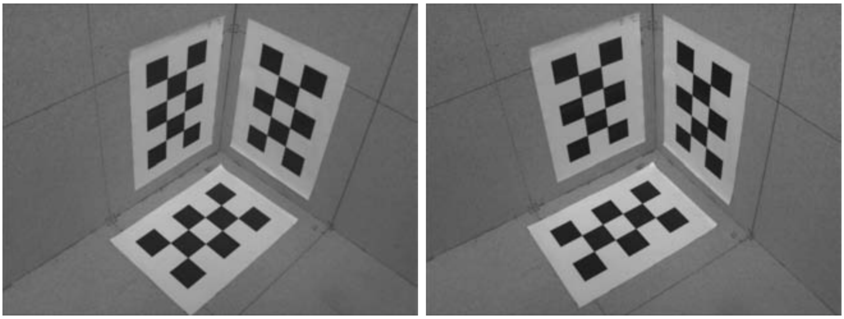

Fig. 1. Images from the two cameras viewing the calibration pattern

Table 1. Reconstruction error for the triangulation operator **K**, the optimal method (O), the mid-point method (MP), the homogeneous method (H), and the inhomogeneous method (IH). All units are in mm. The last row shows a rough figure for the computation time relative to the proposed method.

	K	O	MP	H	IH
$\bar{e}$	1.08	1.07	1.07	1.12	1.12
e_{max}	2.74	2.79	2.79	3.44	3.45
σ_e	0.62	0.62	0.62	0.66	0.66
Time	1	34	15	5	3

are assumed to satisfy the pinhole camera model, Equation (1), see Figure 1. The calibration points are manually identified and measured in terms of their image coordinates in each of the two images. As a result, we have one set of 3D points $\{\mathbf{x}_k\}$ and two sets of *corresponding* 2D coordinates in the two images $\{\mathbf{y}_{1,k}\}$ and $\{\mathbf{y}_{2,k}\}$.

From these data sets, the two camera matrices $\mathbf{C}_1$ and $\mathbf{C}_2$ are estimated using the normalized DLT-method [5]. From $\mathbf{C}_1$ and $\mathbf{C}_2$, the corresponding focal points $\mathbf{n}_1$ and $\mathbf{n}_2$ can be determined using Equation (3), and a suitable blind plane $\mathbf{p}$ can be set up by choosing a third point so that this $\mathbf{p}$ is approximately perpendicular to the viewing directions of the two cameras. Given $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{p}$, a triangulation operator $\mathbf{K}$ is computed according to the algorithm described in Section 2. This computation is made using coordinates which are normalized according to [6], followed by a proper re-normalization.

The triangulation operator $\mathbf{K}$ is evaluated on the data set together with the standard methods described in Section 1. The homogeneous and the inhomogeneous method is computed on normalized data and the result is transformed back to the original coordinates. For each of the 72 corresponding image points a 3D point in reconstructed, and the norm of the Euclidean reconstruction error is computed as $e = \|\mathbf{x}'_i - \mathbf{x}'_{i,rec}\|$ where $\mathbf{x}'_i$ is the i-th 3D coordinate and $\mathbf{x}'_{i,rec}$ is the corresponding reconstructed 3D coordinate. From the entire set of such errors, the mean $\bar{e}$, the maximum e_{max} and the standard deviation σ_e are estimated. The resulting values are presented in Table 1. All calculations are made in Matlab.

A few conclusions can be drawn from these figures. First, the triangulation operator $\mathbf{K}$ appears to be *comparable* in terms of accuracy with the best of the standard methods, which for this data set happens to be the optimal method. Given that the accuracy of the original 3D data is approximately ±1 mm. and the 2D data has an accuracy of approximately ±1 pixel, a more precise conclusion than this cannot be made based on this data. The last row of Table 1 shows very rough figures for the computation time of each of the methods as given by Matlab's profile function. The proposed method has a computational time which is significantly less than any of the standard methods, although it provides a comparable reconstruction error, at least for this data set.

4 Summary

This paper demonstrates that there exists a linear transformation $\mathbf{K}$ which maps the tensor or outer product $\mathbf{Y}$ of the homogeneous representations of two corresponding stereo image points to a homogeneous representation of the original 3D point. The main restriction of the proposed method is that the resulting linear transformation is dependent on an arbitrarily chosen 3D plane which includes the two focal points of the cameras, and only points which are not in the plane can be triangulated.

The proposed method is not derived from a perspective of optimality in either the 2D or 3D domains. The experiment presented above suggests, however, that at least the 3D reconstruction error is comparable even to the optimal method. In addition to this, the proposed method offers the following advantages

- It is can be implemented at a low computational cost; 32 additions and 45 multiplications for obtaining the homogeneous coordinates of the reconstructed 3D point. This is a critical factor in real-time applications or in RANSAC loops involving 3D matching.
- The existence of the reconstruction operator $\mathbf{K}$ implies that closed form expressions from homogeneous 2D coordinates to various homogeneous representations in 3D can be described. For example, given two pairs of corresponding points with their joint representations $\mathbf{Y}_1$ and $\mathbf{Y}_2$, two 3D points can be reconstructed as $\mathbf{x}_1 = \mathbf{K}\,\mathbf{Y}_1$ and $\mathbf{x}_2 = \mathbf{K}\,\mathbf{Y}_2$. The 3D line which passes through these two points can be represented by the anti-symmetric matrix $\mathbf{L} = \mathbf{x}_1 \otimes \mathbf{x}_2 - \mathbf{x}_2 \otimes \mathbf{x}_1$. A combination of these expressions then allows us to write $\mathbf{L} = (\mathbf{K} \otimes \mathbf{K})(\mathbf{Y}_1 \otimes \mathbf{Y}_2 - \mathbf{Y}_2 \otimes \mathbf{Y}_1)$, i.e., we can first combine the joint representations of corresponding points in the image domain and then transform this combination to the 3D space and directly get $\mathbf{L}$. This may only be of academic interest, but implies that $\mathbf{K}$ serves as some kind of stereo camera inverse (an inverse of a combination of both camera matrices).
- Although this property is not proven here, it follows from the construction of $\mathbf{K}$ that it is projective invariant in the sense described in [7]. This implies that the reconstructed point $\mathbf{x}$ is invariant, that is, it is the same point in space, independent of projective transformations of the coordinate system.

Acknowledgement

This work has been made within the VISCOS project funded by the Swedish Foundation for Strategic Research (SSF).

References

1. Beardsley, P.A., Zisserman, A., Murray, D.W.: Navigation using affine structure from motion. In: Eklundh, J.-O. (ed.) ECCV 1994. LNCS, vol. 801, pp. 85–96. Springer, Heidelberg (1994)

2. Christensen, O.: An Introduction to Frames and Riesz Bases. Birkhäuser (2003)
3. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. 4th Alvey Vision Conference, Manchester, UK pp. 147–151 (1988)
4. Hartley, R., Schaffalitzky, F.: minimization in geometric reconstruction problems. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. I, pp. 769–775 (2004)
5. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2003)
6. Hartley, R.I.: In defence of the 8-point algorithm. IEEE Trans. on Pattern Recognition and Machine Intelligence 19(6), 580–593 (1997)
7. Hartley, R.I., Sturm, P.: Triangulation. Computer Vision and Image Understanding 68(2), 146–157 (1997)
8. Kahl, F., Henrion, D.: Globally optimal estimates for geometric reconstruction problems. In: Proceedings of International Conference on Computer Vision, vol. 2, pp. 978–985 (2005)
9. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal in Computer Vision, 47(1–3):7–42 (2002)