Computer Vision and Image Understanding 113 (2009) 935–945

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

The triangulation tensor

Klas Nordberg

Computer Vision Laboratory, Department of Electrical Engineering, Link ping University, Linköping, Sweden

ARTICLE INFO

Article history:
Received 6 August 2006
Accepted 22 April 2009
Available online 18 May 2009

Keywords:
Triangulation
Reconstriction
Stereo
Estimation

ABSTRACT

This article presents a computationally efficient approach to the triangulation of 3D points from their projections in two views. The homogenous coordinates of a 3D point is given as a multi-linear mapping on its homogeneous image coordinates, a computation of low computational complexity. The multi-linear mapping is a tensor, and an element of a projective space, that can be computed directly from the camera matrices and some parameters. These parameters imply that the tensor is not unique: for a given camera pair the subspace K of triangulation tensors is six-dimensional. The triangulation tensor is 3D projective covariant and satisfies a set of internal constraints. Reconstruction of 3D points using the proposed tensor is studied for the non-ideal case, when the image coordinates are perturbed by noise and the epipolar constraint exactly is not satisfied exactly. A particular tensor of K is then the optimal choice for a simple reduction of 3D errors, and we present a computationally efficient approach for determining this tensor. This approach implies that normalizing image coordinate transformations are important for obtaining small 3D errors.

In addition to computing the tensor from the cameras, we also investigate how it can be further optimized relative to error measures in the 3D and 2D spaces. This optimization is evaluated for sets of real 3D + 2D + 2D data by comparing the reconstruction to some of the triangulation methods found in the literature, in particular the so-called *optimal method* that minimizes 2D L_2 errors. The general conclusion is that, depending on the choice of error measure and the optimization implementation, it is possible to find a tensor that produces smaller 3D errors (both L_1 and L_2) but slightly larger 2D errors than the optimal method does. Alternatively, we may find a tensor that gives approximately comparable results to the optimal method in terms of both 3D and 2D errors. This means that the proposed tensor based method of triangulation is both computationally efficient and can be calibrated to produce small reconstruction or reprojection errors for a given data set.

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 [4,6,2,7,9]. A common method is *optimal triangulation* that minimizes the total L_2 reprojection error in the image domains. This is 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 2D Euclidean distance measured in the image space, and where $\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2$ satisfy the epipolar constraint defined by the fundamental matrix. A non-iterative computational method for determining $\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2$ has been presented by Hartley and Sturm [3,6], and any standard technique can then be used to compute the 3D point from $\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2$. The statistical properties of this problem have been treated also by Kanatani [8]. Other standard methods found in the literature are the *mid-point* method, and the *homogeneous* and the *inhomogeneous* methods [4].

The optimal method is often described as the preferred approach to triangulation. It gives a maximum likelihood estimate of the 3D point, it is covariant to 3D projective transformations[1], and in general produces reconstructed 3D points of high accuracy although its optimality is defined in the image domains. The downside of the optimal method is the complexity of its computations, which include finding and comparing the roots of a sixth order polynomial. This makes the method less attractive for real-time applications where large sets of points are reconstructed, for example implemented in GPU hardware. A computationally less expensive approach is described by Kanatani et al. [9] where close approximations of $\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2$ are computed in few iterations and with relatively simple computations.

Optimal triangulation solves one optimization problem for every new pair of image points. Here, we explore the possibility of describing a parameterized reconstruction process that can be optimized

[1] This means that if the 3D coordinate system is transformed by a homography, the coordinates of the reconstructed 3D point always transform in the same way. This is sometimes also referred to as projective invariance.

E-mail address: klas@isy.liu.se

1077-3142/$ - see front matter
doi:10.1016/j.cviu.2009.04.003

for some calibration set of 3D and 2D data, and that can be applied to a large range of error measures. Once determined, we can use this process to reconstruct arbitrary points and produce small errors provided that the new points are sufficiently close to the calibration set. This article shows that such a reconstruction process indeed exists: it is a multi-linear mapping on the homogeneous image coordinates and, as such, it has a low computational cost.

1.1. Mathematical preliminaries

This work is based on the multi-view pin-hole camera model where the relation between 2D image and 3D world coordinates is described as:

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

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 two Greek indices. Unless explicitly stated, we assume Einstein's summation convention, i.e., summation is made over the appropriate range for any index that is repeated in an expression. The matrix/vector notation provides us with a more intuitive interpretation of an object as a linear mapping when it is placed to the left of another object while the index based notation is more precise in terms of implementation. Outer products between vector $\mathbf{u}, \mathbf{v}$ are sometimes denoted as $\mathbf{u}\mathbf{v}^T$ when it is clear that the result represents a matrix, and sometimes as $\mathbf{u} \otimes \mathbf{v}$ when the result is simply a tensor for which a matrix representation is not required. Inner products are denoted either as $\mathbf{u}^T\mathbf{v}$ or $\mathbf{u} \cdot \mathbf{v}$. Both operations are sometimes used also for $\mathbf{u}$ and $\mathbf{v}$ that are not described as vectors, e.g., matrices. In this case, outer products simply means the collection of products between all possible pairs of elements in the two factors, and an inner product is the scalar formed as the sum of all products between pairs of elements in the two factors that have the same index.

The homogeneous coordinates of camera center k is denoted as $\mathbf{n}_k$ and satisfies $\mathbf{C}_k\mathbf{n}_k = \mathbf{0}$ (no summation over k here!). For two cameras, the resulting image coordinates must satisfy the epipolar constraint

$$\mathbf{y}_1^T\mathbf{F}\mathbf{y}_2 = \mathbf{F} \cdot (\mathbf{y}_1\mathbf{y}_2^T) = \mathbf{F} \cdot (\mathbf{y}_1 \otimes \mathbf{y}_2) = 0 \tag{2}$$

where $\mathbf{F}$ is the 3×3 fundamental matrix related to $\mathbf{C}_1$ and $\mathbf{C}_2$, [4]. In the following sections, we will sometimes treat $\mathbf{F}$ as a nine-dimensional vector, and in order to make a clear distinction between its uses as a matrix or a vector, the latter case is denoted by $\mathbf{f}$.

1.2. Multi-linear reconstruction from normalized coordinates

The possibility of reconstructing $\mathbf{x}$ as a bilinear combination of $\mathbf{y}_1$ and $\mathbf{y}_2$ can be derived from the work on the essential matrix by Longuet-Higgins [11]. He assumed normalized cameras, i.e., the inner calibration parameters are represented by the identity matrix. The exterior parameters are represented by a 3×3 rotation matrix $\mathbf{Q}$ and a 3D translation vector $\mathbf{t}$ such that $\bar{\mathbf{x}}_2 = \mathbf{Q}(\bar{\mathbf{x}}_1 - \mathbf{t})$, where $\bar{\mathbf{x}}_k$ are the camera centered 3D coordinates of camera k. From Eqs. (31) and (32) in [11] it follows directly that

$$\mathbf{x}_1 = \begin{pmatrix} \bar{\mathbf{x}}_1 \\ 1 \end{pmatrix} \sim \begin{pmatrix} \mathbf{y}_1\mathbf{t}^T\mathbf{Q}^T\mathbf{H}\mathbf{y}_2 \\ \mathbf{y}_1^T\mathbf{Q}^T\mathbf{H}\mathbf{y}_2 \end{pmatrix} \quad \text{where } \mathbf{H} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \tag{3}$$

which means that the elements of $\mathbf{x}$ can be computed as a linear mapping on $\mathbf{y}_1 \otimes \mathbf{y}_2$.

Multiplication of $\mathbf{y}_k$ with a suitable inner calibration matrix and transformation of $\mathbf{x}$ to an arbitrary 3D coordinate system leads to a more general formulation of the multi-linear mapping from 2D to 3D. This type of derivation, however, has a number of limitations. For example, there is no easy way to know if this is a unique mapping or how large the space of such mappings may be. It also does not clearly reveal the various interesting properties this mapping has, some of which are described in Sections 3 and 4.

1.3. This article

In this article we present a purely projective derivation of multi-linear triangulation mappings in the form of a tensor. As a result, the 3D coordinates of the reconstructed point can be computed with only a small set of multiplications in a single iteration. The original derivation of the tensor was presented in [12] and is extended here by a detailed presentation of its 3D and 2D covariance properties together with a derivation of the internal constraints in Section 3. The perturbation analysis of the reconstructed 3D point with respect to noise in the image coordinates presented in Section 4 is also novel and is applied to the optimization procedures described in Section 5. The preliminary results of the optimization procedure, originally presented in [14], are here extended by a method where the enforcement of the so-called *epipolar condition*, described in Section 3.1, is integrated into the optimization. The main conclusion drawn from experimental results on real data in Section 6 is that the proposed method can produce almost as good 2D reconstruction errors and even better 3D errors as the optimal method, given that a careful estimation of the triangulation tensor has been made.

2. Reconstruction of 3D points from two views

In this section we derive the two-view triangulation operator $\mathbf{K}$. The starting point is the (multiplicative) joint image coordinate[2] of corresponding image points, given by the outer product or tensor product of the two homogeneous image coordinates:

$$\mathbf{y}_1 \otimes \mathbf{y}_2 \sim (\mathbf{C}_1\mathbf{x}) \otimes (\mathbf{C}_2\mathbf{x}) \quad [y_{1,\alpha}y_{2,\beta} \sim C_{1,\alpha\gamma}x_\gamma C_{2,\beta\delta}x_\delta]. \tag{4}$$

A reorganization of the factors gives us with a more operational description of the mapping from $\mathbf{x}$ to $\mathbf{y}_1 \otimes \mathbf{y}_2$:

$$\mathbf{y}_1 \otimes \mathbf{y}_2 \sim (\mathbf{C}_1 \otimes \mathbf{C}_2)(\mathbf{x} \otimes \mathbf{x}) \quad [y_{1,\alpha}y_{2,\beta} \sim C_{1,\alpha\delta}C_{2,\beta\epsilon}x_\delta x_\epsilon] \tag{5}$$

$$\mathbf{Y} \sim \mathbf{C}\mathbf{X} \quad [Y_{\alpha\beta} \sim C_{\alpha\beta\delta\epsilon}X_{\delta\epsilon}] \quad [Y_I \sim C_{IJ}X_J]. \tag{6}$$

In the last equation, we can see the joint camera mapping $\mathbf{C} = \mathbf{C}_1 \otimes \mathbf{C}_2$ as a 9×16 matrix that maps $\mathbf{X} = \mathbf{x} \otimes \mathbf{x}$ (reshaped as a 16-dimensional vector) to the joint image coordinate $\mathbf{Y} = \mathbf{y}_1 \otimes \mathbf{y}_2$ (a nine-dimensional vector).

The next step is to make use of the fact that $\mathbf{X} = \mathbf{x} \otimes \mathbf{x}$ is an element of the 10-dimensional subspace of completely symmetric second order tensor on $\mathbb{R}^4$, denoted by $\mathscr{X}$. Let $\mathbf{P}$ denote the projection operator of $\mathscr{X}$, i.e., $\mathbf{P}\mathbf{X} = \mathbf{X}$ for $\mathbf{X} \in \mathscr{X}$ and $\mathbf{P}\mathbf{X}' = \mathbf{0}$ for $\mathbf{X}' \in \mathscr{X}^\perp$. We can represent this $\mathbf{P}$ as a 16×16 matrix, and inserted into Eq. (6) it gives

$$\mathbf{Y} \sim \mathbf{C}\mathbf{P}\mathbf{X} = \mathbf{M}\mathbf{X} \; [Y_I \sim C_{IJ}P_{JL}X_L = M_{IL}X_L]. \tag{7}$$

[2] Compare this multiplicative construction of a joint image coordinate of corresponding image point with the additive joint image formed by a direct vector sum (concatenation) of the homogeneous image coordinates described by Triggs [16].