Available online at www.sciencedirect.com

ScienceDirect

Image and Vision Computing 26 (2008) 315–324

www.elsevier.com/locate/imavis

Triangulation for points on lines

Adrien Bartoli *, Jean-Thierry Laprest é

LASMEA – CNRS/Université Blaise Pascal, Clermont-Ferrand, France

Received 13 May 2006; received in revised form 23 February 2007; accepted 1 June 2007

Abstract

Triangulation consists in finding a 3D point reprojecting the best as possible onto corresponding image points. It is classical to minimize the reprojection error, which, in the pinhole camera model case, is nonlinear in the 3D point coordinates. We study the triangulation of points lying on a 3D line, which is a typical problem for Structure-From-Motion in man-made environments. We show that the reprojection error can be minimized by finding the real roots of a polynomial in a single variable, which degree depends on the number of images. We use a set of transformations in 3D and in the images to make the degree of this polynomial as low as possible, and derive a practical reconstruction algorithm. Experimental comparisons with an algebraic approximation algorithm and minimization of the reprojection error using Gauss–Newton are reported for simulated and real data. Our algorithm finds the optimal solution with high accuracy in all cases, showing that the polynomial equation is very stable. It only computes the roots corresponding to feasible points, and can thus deal with a very large number of views – triangulation from hundreds of views is performed in a few seconds. Reconstruction accuracy is shown to be greatly improved compared to standard triangulation methods that do not take the line constraint into account.

Keywords: Triangulation; Structure-From-Motion; Point; Line

1. Introduction

Triangulation is one of the main building blocks of Structure-From-Motion algorithms. Given image feature correspondences and camera matrices, it consists in finding the position of the underlying 3D feature, by minimizing some error criterion. This criterion is often chosen as the reprojection error – the Maximum Likelihood criterion for a Gaussian, centered and *i.i.d.* noise model on the image point positions – though other criteria are possible [5,9,10].

Traditionally, triangulation is carried out by some suboptimal procedure and is then refined by local optimization, see *e.g.* [7]. A drawback of this is that convergence to the optimal solution is not guaranteed. Optimal procedures for triangulating points from two and three views were proposed in [6,13].

We address the problem of triangulating points lying on a line, that is, given image point correspondences, camera matrices and a 3D line, finding the 3D point lying on the 3D line, such that the reprojection error is minimized.

Our main contribution is to show that the problem can be solved by computing the real roots of a degree-$(3n-2)$ polynomial, where n is the number of views. Extensive experiments on simulated data show that the polynomial is very well balanced since large number of views and large level of noise are handled. The method is valid whatever the calibration level of the cameras is – projective, affine, metric or Euclidean.

One may argue that triangulating points on a line only has a theoretical interest since in practice, triangulating a line from multiple views is done by minimizing the reprojection error over its supporting points which 3D positions are hence reconstructed along with the 3D line. Indeed, most work consider the case where the supporting points do *not* match across the images, see *e.g.* [3]. When one identifies correspondences of supporting points accross the

* Corresponding author.
E-mail address: Adrien.Bartoli@gmail.com (A. Bartoli).

0262-8856/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2007.06.003

images, it is fruitful to incorporate these constraints into the bundle adjustment, as is demonstrated by our experiments. This is typically the case in man-made environments, where one identifies, *e.g.* matching corners at the meet of planar facades or around windows. Bartoli et al. [2] dubbed Pencil-of-Points or 'POP' this type of features. In order to find an initial 3D reconstruction, a natural way is to compute the 3D line by some means (*e.g.* by ignoring the matching constraints of the supporting points, from 3D primitives such as the intersection of two planes, or from a registered wireframe CAD model) and then to triangulate the supporting point correspondences using point on line triangulation. The result can then be plugged into a bundle adjustment incorporating the constraints.

We review some related work in Section 2. Our triangulation method is derived in Section 3. A linear least squares method minimizing an algebraic distance is provided in Section 4. Gauss–Newton refinement is summarized in Section 5. Experimental results are reported in Section 6 and our conclusions in Section 7.

Notation. Vectors are written using bold fonts, *e.g.* $\mathbf{q}$, and matrices using sans-serif fonts, *e.g.* P. Almost everything is homogeneous, *i.e.* defined up to scale. Equality up to scale is denoted $\sim$. The inhomogenous part of a vector is denoted using a bar, *e.g.* $\mathbf{q}^\mathsf{T} \sim (\bar{\mathbf{q}}^\mathsf{T}\ 1)$ where T is transposition. Index $i = 1, \ldots, n$, and sometime j are used for the images. The point in the ith image is $\mathbf{q}_i$. Its elements are $\mathbf{q}_i^\mathsf{T} \sim (q_{i,1}\ q_{i,2}\ 1)$. The 3D line joining points $\mathbf{M}$ and $\mathbf{N}$ is denoted $(\mathbf{M}, \mathbf{N})$. The $\mathcal{L}_2$-norm of a vector is denoted as in $\|\mathbf{x}\|^2 = \mathbf{x}^\mathsf{T}\mathbf{x}$. The Euclidean distance measure d_e is defined by

$$d_e^2(\mathbf{x}, \mathbf{y}) = \left\| \frac{\mathbf{x}}{x_3} - \frac{\mathbf{y}}{y_3} \right\|^2 = \left(\frac{x_1}{x_3} - \frac{y_1}{y_3} \right)^2 + \left(\frac{x_2}{x_3} - \frac{y_2}{y_3} \right)^2. \tag{1}$$

2. Related work

Optimal procedures for triangulating points in 3D space, and points lying on a plane were previously studied, as summarized in Table 1. Hartley and Sturm [6] showed that triangulating points in 3D space from two views, in other words finding a pair of points satisfying the epipolar geometry and lying as close as possible to the measured points, can be solved by finding the real roots of a degree-6 polynomial. The optimal solution is then selected by straightforward evaluation of the reprojection error. Stewénius et al. [13] extended the method to three views. The optimal solution is one of the real roots of a system of 3 degree-6 polynomials in the three coordinates of the point. Chum et al. [4] show that triangulating points lying on a plane, in other words finding a pair of points satisfying an homography and lying as close as possible to the measured points, can be solved by finding the real roots of a degree-8 polynomial.

Error functions different from the reprojection error were considered in the literature. The directional error in two views is proposed in [10], along with a triangulation method for calibrated cameras. The $\mathcal{L}_\infty$-norm is considered in [5,9], instead of the usual $\mathcal{L}_2$-norm. A triangulation method for two views is given in [9], while it is shown in [5] that the n-view case can be cast as a convex optimization problem Table 2.

3. Minimizing the reprojection error

We derive our optimal triangulation algorithm for point on line, dubbed 'POLY'.

3.1. Problem statement and parameterization

We want to compute a 3D point $\mathbf{Q}$, lying on a 3D line $(\mathbf{M}, \mathbf{N})$, represented by two 3D points $\mathbf{M}$ and $\mathbf{N}$. The (3×4) perspective camera matrices are denoted P_i with $i = 1, \ldots, n$ the image index. The problem is to find the point $\hat{\mathbf{Q}}$ such that

$$\hat{\mathbf{Q}} \sim \arg \min_{\mathbf{Q} \in (\mathbf{M}, \mathbf{N})} \mathcal{C}_n^2(\mathbf{Q}),$$

where $\mathcal{C}_n$ is the n-view *reprojection error*

$$\mathcal{C}_n^2(\mathbf{Q}) = \sum_{i=1}^{n} d_e^2(\mathbf{q}_i, \mathsf{P}_i\mathbf{Q}). \tag{2}$$

We parameterize the point $\mathbf{Q} \in (\mathbf{M}, \mathbf{N})$ using a single parameter $\lambda \in \mathbb{R}$ as

$$\mathbf{Q} \sim \lambda\mathbf{M} + (1 - \lambda)\mathbf{N} \sim \lambda(\mathbf{M} - \mathbf{N}) + \mathbf{N}. \tag{3}$$

Introducing this parameterization into the reprojection error (2) yields

$$\mathcal{C}_n^2(\lambda) = \sum_{i=1}^{n} d_e^2(\mathbf{q}_i, \mathsf{P}_i(\lambda(\mathbf{M} - \mathbf{N}) + \mathbf{N})).$$

Defining $\mathbf{b}_i = \mathsf{P}_i(\mathbf{M} - \mathbf{N})$ and $\mathbf{d}_i = \mathsf{P}_i\mathbf{N}$, we get

$$\mathcal{C}_n^2(\lambda) = \sum_{i=1}^{n} d_e^2(\mathbf{q}_i, \lambda\mathbf{b}_i + \mathbf{d}_i). \tag{4}$$

Table 1
Different types of triangulation and methods minimizing the $\mathcal{L}_2$-norm reprojection error

Type of triangulation	Number of views	Polynomial system			Reference
		Number	Degree	Variables	
Point in 3D space	2	1	6	1	[6]
	3	3	6/6/6	3	[13]
Point on plane	2	1	8	1	[4]
Point on line	1	1	1	1	This paper
	2	1	4	1	
	3	1	7	1	
	4	1	10	1	
	n	1	$3n - 2$	1	

The number of polynomials to be solved, their degrees and the number of variables is given in the column 'Polynomial system'.