Robust Estimation of the Trifocal Tensor: A Comparative Performance Evaluation

Stuart B. Heinrich[a,*], Wesley E. Snyder[a]

[a]*Department of Computer Science, NC State University, Box 8206, Raleigh, NC 27695-8206*

Abstract

Keywords: trifocal tensor, projective reconstruction, RANSAC

1. Introduction

Bundle adjustment [1–3] is the maximum likelihood nonlinear improvement of a structure from motion (SfM) reconstruction, but being a nonlinear algorithm it requires a very good initialization. When dealing with uncalibrated monocular video or snap-shots, there are still no practical algorithms for direct metric reconstruction, and hence initialization is usually performed in projective space because the projective geometry of two or three views (as represented by the fundamental matrix and trifocal tensor, respectively) can be estimated minimally or linearly.

A reconstruction spanning an arbitrarily large number of views can be obtained by merging smaller projective reconstructions together. Although these partial reconstructions could be computed using either the fundamental matrix or trifocal tensor, there are many theoretical advantages to using the trifocal tensor:

1. Estimation of the fundamental matrix from the images of coplanar scene points is ill-conditioned, whereas the trifocal tensor is still uniquely determined [4].
2. There are additional types of constraints available for three views, so correspondence measurements provide stronger constraints on an estimate of the tensor than they do on an estimate of the fundamental matrix [5].
3. In order to merge two fundamental matrices using corresponding structure points, correspondences must be tracked through at least three frames already; thus, one might as well use the correspondences to their full potential by estimating the trifocal tensor.

It is well known that the trifocal tensor can be estimated either minimally from 6 points [6–9] or linearly from 7 or more points [10–12]. The linear method is over-determined, which provides robustness to noise, but does not enforce internal constraints so the result is not geometrically consistent. In contrast, the minimal algorithm implicitly enforces all internal constraints and requires fewer points, which theoretically means fewer iterations will be required when used within a RANSAC framework for robust estimation.

The importance of using minimal methods within RANSAC has been stressed [13], and in particular it has been concluded that the 6 point method should be used when estimating the trifocal tensor [2, 14], with empirical results showing that the 6 point method produces substantially lower error [14]. However, more recently developed quasi-linear methods improve the performance of the linear method and were not considered in that study. The purpose of this research was to determine whether or not the minimal or linear algorithm is better to use within RANSAC when state of the art techniques are employed; and, if the linear method is superior, then we also wanted to know which variation was most effective, and how many points to use for optimal performance.

We begin by introducing some basic mathematical background by showing how the tensor can be derived from corresponding line constraints in three images (Section 2) and how it relates to projection matrices (Section 2.1). We then discuss trifocal tensor estimation algorithms (Section 3), beginning with the minimal 6 point solution (Section 3.1), in which we introduce some minor tricks for improving robustness and disambiguiting between the multiple solutions. Next we introduce the basic linear method (Section 3.2), and discuss three alternative ways for enforcing the trilinear constraints (Section 3.2.1), as well as four methods for quasi-linear reestimation to enforce internal consistency constraints (Section 3.2.2). We also provide a discussion of additional estimation algorithms and explain why they were not included in our comparison (Section 3.3).

Out experiments (Section 5) begin with several tests designed to first find the best linear variation (Section 5.1) which we then compare to the minimal algorithm to see which has better performance (Section 5.2). Finally we investigate performance in RANSAC as a function of the number of points used, on both synthetic and real data (Section 5.3).

Our experimental results indicate several things: (a) we show that an older, lesser used, method of quasi-linear enforcement of the internal constraints actually performs best; (b) we could find no difference in performance between the various methods of trilinear constraint representation, which leads us to believe that it is best to stick with the simplest and fastest method; (c)

*Corresponding author
Email address: sbheinri@ncsu.edu (Stuart B. Heinrich)

we show that the best linear variation provides a substantially more accurate estimate than the minimal method, and is nearly a maximum likelihood estimate when estimated from more than 10 points; (d) contrary to popular belief, we show that using larger subset size in RANSAC is actually better because it allows a larger final consensus size to be reached, and in a shorter overall runtime, despite the fact that runtime for the minimal method by itself is substantially faster.

2. The Trifocal Tensor

The constraints on corresponding lines in three views were first derived for calibrated cameras in [15, 16]. These constraints were generalized to the uncalibrated case in [17], and formulated in terms of a trifocal tensor in [5, 10, 11]. It was shown in [18] that point constraints could also be represented using the tensor. In this section we summarize the derivation of the tensor from line constraints as described in [2].

Without loss of generality, the first projection matrix can be assumed canonical, so that the set of projection matrices for three views can be written as

$$\mathbf{P} = [\mathtt{I}|\mathbf{0}] \tag{1}$$

$$\mathbf{P}' = [\mathbf{a}_1 \dots \mathbf{a}_4] = [\mathbf{A}|\mathbf{a}_4] \tag{2}$$

$$\mathbf{P}'' = [\mathbf{b}_1 \dots \mathbf{b}_4] = [\mathbf{B}|\mathbf{b}_4]. \tag{3}$$

The tensor will be derived based on a correspondence between images of a line in 3D space. Let the three corresponding lines in the image plane be denoted as $\mathbf{l} \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$. The back projection of each line yields a plane,

$$\pi = \mathbf{P}^\mathsf{T}\mathbf{l} = (\mathbf{l}^\mathsf{T}, 0)^\mathsf{T} \tag{4}$$

$$\pi' = \mathbf{P}'^\mathsf{T}\mathbf{l}' = \begin{bmatrix} \mathbf{A}^\mathsf{T}\mathbf{l}' \\ \mathbf{a}_4^\mathsf{T}\mathbf{l}' \end{bmatrix} \tag{5}$$

$$\pi'' = \mathbf{P}''^\mathsf{T}\mathbf{l}'' = \begin{bmatrix} \mathbf{B}^\mathsf{T}\mathbf{l}'' \\ \mathbf{b}_4^\mathsf{T}\mathbf{l}'' \end{bmatrix}. \tag{6}$$

Because the lines were all images of a single 3D line, these back-projected planes must all intersect in a single 3D line that we write parametrically as a linear combination of two points $\mathbf{X}_1$ and $\mathbf{X}_2$,

$$\mathbf{X}(t) = t\mathbf{X}_1 + (1-t)\mathbf{X}_2. \tag{7}$$

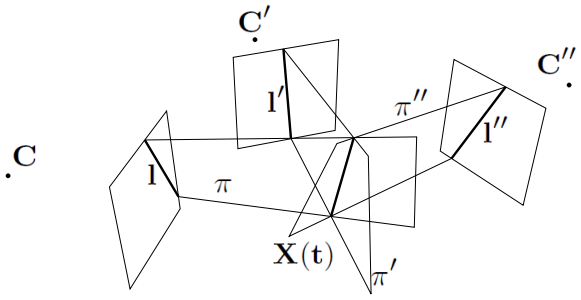

Figure 1. Diagram of trifocal line constraints. The first camera center is denoted by **C**. A parametric 3D line in space is given by $X(t)$. This line projects onto the first image plane as **l**. The line **l** back-projects to the plane π. Notation is similar with respect to the other two views.

This incidence relation is diagrammed in Fig. 1. Clearly, $\mathbf{X}(t)$ must be a point on each back-projected plane equation, so

$$\pi^\mathsf{T}\mathbf{X}(t) = \pi'^\mathsf{T}\mathbf{X}(t) = \pi''^\mathsf{T}\mathbf{X}(t) = 0. \tag{8}$$

If we concatenate these planes into a 4×3 matrix $\mathbf{M} = [\pi|\pi'|\pi'']$, then $\mathbf{M}^\mathsf{T}\mathbf{X}(t) = 0$. Substituting (4-6) into $\mathbf{M}$, we obtain

$$\mathbf{M} = \begin{bmatrix} \mathbf{l} & \mathbf{A}^\mathsf{T}\mathbf{l}' & \mathbf{B}^\mathsf{T}\mathbf{l}'' \\ 0 & \mathbf{a}_4^\mathsf{T}\mathbf{l}' & \mathbf{b}_4^\mathsf{T}\mathbf{l}'' \end{bmatrix}. \tag{9}$$

Because $\mathbf{M}^\mathsf{T}\mathbf{X}_1 = 0$ and $\mathbf{M}^\mathsf{T}\mathbf{X}_2 = 0$, $\mathbf{M}$ must have a 2-dimensional null space and is therefore rank 2 by the rank-nullity theorem. Thus, it follows that the first column can be written as a linear combination of the second two columns, so $\pi = \alpha\pi' + \beta\pi''$. From the bottom row we obtain

$$0 = \alpha\mathbf{a}_4^\mathsf{T}\mathbf{l}' + \beta\mathbf{b}_4^\mathsf{T}\mathbf{l}'', \tag{10}$$

which implies that $\alpha = k\mathbf{b}_4^\mathsf{T}\mathbf{l}''$ and $\beta = -k\mathbf{a}_4^\mathsf{T}\mathbf{l}'$ for some scalar k. Making these substitutions back into the top half of $\mathbf{M}$ provides a homogeneous equivalence constraint between the lines,

$$\mathbf{l} = \mathbf{b}_4^\mathsf{T}\mathbf{l}''\mathbf{A}^\mathsf{T}\mathbf{l}' - \mathbf{a}_4^\mathsf{T}\mathbf{l}'\mathbf{B}^\mathsf{T}\mathbf{l}'' \tag{11}$$

$$= \mathbf{l}''^\mathsf{T}\mathbf{b}_4\mathbf{A}^\mathsf{T}\mathbf{l}' - \mathbf{l}'^\mathsf{T}\mathbf{a}_4\mathbf{B}^\mathsf{T}\mathbf{l}''. \tag{12}$$

Introducing the notation $\mathbf{l} = (l_1, l_2, l_3)^\mathsf{T}$ and

$$\mathtt{T}_i = \mathbf{a}_i\mathbf{b}_4^\mathsf{T} - \mathbf{a}_4\mathbf{b}_i^\mathsf{T}, \tag{13}$$

it can be verified that (12) is equivalent to

$$l_i = \mathbf{l}'\mathtt{T}_i\mathbf{l}'' \ \forall i. \tag{14}$$

Thus, the relationship between cameras has been completely described by $\{\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3\}$. These three matrices, known as the *correlation slices*, can be represented by a single $3\times3\times3$ tensor

$\mathcal{T}$, allowing the above relations to be written equivalently in tensor notation as

$$\mathcal{T}_i^{jk} = a_i^j b_4^k - a_4^j b_i^k \tag{15}$$

$$l_i = l'_j l''_k \mathcal{T}_j^{jk}. \tag{16}$$

It should be noted that, similar to the fundamental matrix, the views are treated asymmetrically by the trifocal tensor. In other words, there are three different trifocal tensors for any trio of views depending on the order in which the views are considered. In the remainder of this work, we assume an implicit ordering of these views.

2.1. Relationship to Projection Matrices

Because the trifocal tensor provides a complete description of the epipolar geometry for three views, it must be possible to extract a suitable set of projection matrices. However, it is not immediately obvious how one could factor a given tensor into the form of (13) to get back the original camera matrices. An algorithm is given in [2, Alg. 15.1] and is summarized here.

One begins by calculating the epipoles $\mathbf{e}'$ and $\mathbf{e}''$, which are the images of the focal point of the first camera in the other two views. This is achieved in two steps. First, denote the left and right null spaces of each $\mathtt{T}_i$ as $\mathbf{v}_i$ and $\mathbf{u}_i$ in

$$\mathtt{T}_i \mathbf{v}_i = \mathbf{0}, \quad i = 1 \dots 3 \tag{17}$$

$$\mathtt{T}_i^\mathsf{T} \mathbf{u}_i = \mathbf{0}, \quad i = 1 \dots 3. \tag{18}$$

Next, denote $\mathbf{U} = [\mathbf{u}_1|\mathbf{u}_2|\mathbf{u}_3]^\mathsf{T}$ and $\mathbf{V} = [\mathbf{v}_1|\mathbf{v}_2|\mathbf{v}_3]^\mathsf{T}$. Then the epipoles are given by the null spaces of $\mathbf{U}$ and $\mathbf{V}$,

$$\mathbf{U}\mathbf{e}' = \mathbf{0} \tag{19}$$

$$\mathbf{V}\mathbf{e}'' = \mathbf{0}. \tag{20}$$

Once the epipoles have been determined, one can recover the fundamental matrix between the first two views. Recall that the tensor was defined based on a correspondence between lines $\mathbf{l} \leftrightarrow \mathbf{l}' \leftrightarrow \mathbf{l}''$ in each image. If the third line $\mathbf{l}''$ back projects into a plane π'', then this plane induces a planar-homography mapping the first line $\mathbf{l}$ to the second line $\mathbf{l}'$.

A homography that transfers points according to $\mathbf{x}' = \mathbf{H}\mathbf{x}$ transfers lines according to $\mathbf{l}' = \mathbf{H}^{-\mathsf{T}}\mathbf{l}$. According to this definition, (14) implies that the homography transferring a line from the first to the second image induced by a line in the third image is given by

$$\mathbf{H}_{12} = [\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3]\mathbf{l}'', \tag{21}$$

where the notational convention of writing $\mathbf{A}[\mathbf{B}, \mathbf{C}, \mathbf{D}]\mathbf{E}$ is used as a shorthand for $[\mathbf{ABE}|\mathbf{ACE}|\mathbf{ADE}]$.

Given a point $\mathbf{x}$ in the first view, it is therefore transferred to $\mathbf{x}' = \mathbf{H}_{12}\mathbf{x}$ in the second view. The line between two points is given by the cross product, so the epipolar line $\mathbf{l}'_e$ corresponding to $\mathbf{x}$ is given by

$$\mathbf{l}'_e = \mathbf{e}' \times [\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3]\mathbf{l}''\mathbf{x}. \tag{22}$$

Thus, the fundamental matrix $\mathbf{F}_{12}$ from the first to the second view is given by

$$\mathbf{F}_{12} = [\mathbf{e}']_\times [\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3]\mathbf{l}''. \tag{23}$$

This formula holds for any $\mathbf{l}''$ as long as $\mathbf{l}''$ is not in the null space of any $\mathbf{T}_i$. One choice that avoids this degeneracy is $\mathbf{e}''$. Thus, one obtains

$$\mathbf{F}_{12} = [\mathbf{e}']_\times [\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3]\mathbf{e}''. \tag{24}$$

It is known that the fundamental matrix corresponding to a pair of cameras given by $\mathbf{P} = [\mathtt{I}|\mathbf{0}]$ and $\mathbf{P}' = [\mathbf{M}|\mathbf{m}]$ is equal to $[\mathbf{m}]_\times\mathbf{M}$. Therefore, a suitable choice for the first two camera matrices consistent with the tensor is given by

$$\mathbf{P} = [\mathtt{I}|\mathbf{0}] \tag{25}$$

$$\mathbf{P}' = [[\mathtt{T}_1, \mathtt{T}_2, \mathtt{T}_3]\mathbf{e}''|\mathbf{e}']. \tag{26}$$

The third camera matrix can now be determined from (13). Using the notation of (3),

$$\mathbf{a}_i = \mathtt{T}_i\mathbf{e}'', \quad i = 1 \dots 3 \tag{27}$$

$$\mathbf{a}_4 = \mathbf{e}' \tag{28}$$

$$\mathbf{b}_4 = \mathbf{e}'' \tag{29}$$

and substituting into (13) we obtain

$$\mathtt{T}_i = \mathtt{T}_i\mathbf{e}''\mathbf{e}''^\mathsf{T} - \mathbf{e}'\mathbf{b}_i^\mathsf{T} \tag{30}$$

$$\mathbf{e}'\mathbf{b}_i^\mathsf{T} = \mathtt{T}_i(\mathbf{e}''\mathbf{e}''^\mathsf{T} - \mathtt{I}). \tag{31}$$

If we choose the scale of $\mathbf{e}'$ such that $\mathbf{e}'^\mathsf{T}\mathbf{e}' = \|\mathbf{e}''\| = 1$, then we can left multiply by $\mathbf{e}'^\mathsf{T}$ to get

$$\mathbf{b}_i^\mathsf{T} = \mathbf{e}'^\mathsf{T}\mathtt{T}_i(\mathbf{e}''\mathbf{e}''^\mathsf{T} - \mathtt{I}) \tag{32}$$

$$\mathbf{b}_i = (\mathbf{e}''\mathbf{e}''^\mathsf{T} - \mathtt{I})\mathtt{T}_i^\mathsf{T}\mathbf{e}'. \tag{33}$$

Thus, a consistent choice for the third camera matrix is given by

$$\mathbf{P}'' = [(\mathbf{e}''\mathbf{e}''^\mathsf{T} - \mathtt{I})[\mathtt{T}_1^\mathsf{T}, \mathtt{T}_2^\mathsf{T}, \mathtt{T}_3^\mathsf{T}]\mathbf{e}'|\mathbf{e}'']. \tag{34}$$

3. Initial Tensor Estimation Algorithms

In this section we will review some of the known methods for estimating the trifocal tensor directly (i.e., without using non-linear methods). In section Section 3.1 we describe the minimal algorithm for estimating a tensor from 6 points, in section Section 3.2 we describe the basic linear approach to estimating a tensor from 7 or more points (with several variations), and in Section 3.3 we mention some other algorithms for estimating the trifocal tensor and explain why we did not consider them in our evaluation.

3.1. Minimal Solution

It has been shown that any projective reconstruction algorithm that works on n views of $m+4$ points can be transformed into a dual algorithm for doing a projective reconstruction from m views and $n+4$ points [7]. This observation is known as the Carlsson-Weinshall duality.

Thus, the relatively straightforward minimal reconstruction algorithm for 7 points in 2 views [2, sec. 11.1.2] may be used to compute the dual problem of a minimal reconstruction from 6 points in 3 views [6–9]. This is the basic idea behind the minimal approach, although some further tweaking is possible. The specific algorithm we use is given in [2, alg 20.1], which we summarize below (with two minor improvements).

We denote the 3 unknown camera matrices as $\mathbf{P}_j$, $j = 1\ldots3$, the 6 unknown structure points as $\mathbf{X}_i$, $i = 1\ldots6$, and the image of the ith point in the jth view as $\mathbf{x}_i^j$. Therefore, the projection constraints are written as

$$\mathbf{x}_i^j \propto \mathbf{P}_j\mathbf{X}_i \quad \forall i, j. \tag{35}$$

In the dual algorithm, it will be necessary to use the image measurements as basis vectors, but the method only works if one chooses a set of 4 points, no 3 of which are collinear in any of the views.

Rather than simply verifying that the selection is not collinear (within a threshold), we take this a step beyond [2, alg 20.1] by enumerating all 15 possible ways to pick the 4 points. For each way, we consider the 4 ways to pick a triangle out of the 4 points in each of the 3 views, and pick the set of 4 points so as to maximize the area of the triangle with minimal area in any view (our first improvement). Choosing the points in this way increases the stability of the remainder of the algorithm by ensuring that the points are as far from collinear as possible.

Using the fact that the area of a triangle is given by the determinant of the matrix constructed of homogeneous corner points as rows or columns, this maximization can be written as

$$\max_{\forall a,b,c,d}\left\{\min_{\forall j}\left\{\begin{array}{ll}\left|[\mathbf{x}_a^j|\mathbf{x}_b^j|\mathbf{x}_c^j]\right|, & \left|[\mathbf{x}_a^j|\mathbf{x}_b^j|\mathbf{x}_d^j]\right|, \\ \left|[\mathbf{x}_a^j|\mathbf{x}_d^j|\mathbf{x}_c^j]\right|, & \left|[\mathbf{x}_d^j|\mathbf{x}_b^j|\mathbf{x}_c^j]\right|\end{array}\right\}\right\}, \tag{36}$$

where $\{a, b, c, d\}$ is some combination of indices selected from $\{1, \ldots, 6\}$. Alternatively, one could maximize the residual error to the least squares line (the result would be much the same). In the remaining steps, for notational convenience we assume that the points are ordered such that the selected 4 come first.

The second step is to find projective transforms $\mathbf{T}_j$ for each view $j = 1\ldots3$ that transform the first 4 points in that view to a canonical basis for the projective space $\mathbb{P}^2$. In other words,

$$\mathbf{T}_j\mathbf{x}_i^j = \mathbf{e}_i, \quad i = 1\ldots4, \tag{37}$$

where $\mathbf{e}_i$ for $i = 1\ldots3$ are the standard basis vectors of $\mathbb{R}^3$ and $\mathbf{e}_4 = (1, 1, 1)^\mathsf{T}$. These $\mathbf{T}_j$ matrices can be calculated in closed form, as shown in [6]. Then, by the Carlsson-Weinshall duality [7], correspondences in the dual problem are given by

$$\hat{\mathbf{x}}_j \leftrightarrow \hat{\mathbf{x}}'_j, \quad j = 1\ldots3, \tag{38}$$

where

$$\hat{\mathbf{x}}_j = \mathbf{T}_j\mathbf{x}_5^j \tag{39}$$

$$\hat{\mathbf{x}}'_j = \mathbf{T}_j\mathbf{x}_6^j. \tag{40}$$

In the dual problem, there are 4 implicit correspondences given by $\mathbf{e}_i \leftrightarrow \mathbf{e}_i$ for $i = 1\ldots4$. The constraints $\mathbf{e}_i^\mathsf{T}\hat{\mathbf{F}}\mathbf{e}_i = 0$ for $i = 1\ldots3$ imply that the diagonal elements of $\hat{\mathbf{F}}$ are zero, and the constraint $\mathbf{e}_4^\mathsf{T}\hat{\mathbf{F}}\mathbf{e}_4 = 0$ means that the sum of the elements of $\hat{\mathbf{F}}$ is zero. Thus, the dual fundamental matrix can be parameterized as

$$\hat{\mathbf{F}} = \begin{bmatrix} 0 & p & q \\ r & 0 & s \\ t & -(p+q+r+s+t) & 0 \end{bmatrix}. \tag{41}$$

From the additional dual correspondences in (38), 3 linear constraints are imposed on the entries p, q, r, s, t using $\hat{\mathbf{x}}_j^\mathsf{T}\hat{\mathbf{F}}\hat{\mathbf{x}}'_j = 0$. This leaves a 2-dimensional basis for the null space, but due to the overall scale ambiguity, there is just 1 degree of freedom remaining. Thus, we can write

$$\hat{\mathbf{F}} = \lambda\hat{\mathbf{F}}_1 + \hat{\mathbf{F}}_2, \tag{42}$$

where $\hat{\mathbf{F}}_1$ and $\hat{\mathbf{F}}_2$ are solutions corresponding to the null space basis vectors. The free parameter λ is then determined using the the internal constraint that $\det\hat{\mathbf{F}} = 0$, a cubic equation for which there are 1 or 3 real solutions (complex/imaginary solutions can be ignored).

The next step is to retrieve a pair of reduced camera matrices compatible with the dual fundamental matrix. It is not known how these cameras might be formed directly from (41), but there is an alternative parameterization for the reduced fundamental matrix for which the answer is known. Specifically, if the reduced fundamental matrix is given by

$$\hat{\mathbf{F}} = \begin{bmatrix} 0 & b(d-c) & -c(d-b) \\ -a(d-c) & 0 & c(d-a) \\ a(d-b) & -b(d-a) & 0 \end{bmatrix}, \tag{43}$$

then a corresponding pair of reduced camera matrices is given by

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad \mathbf{P}' = \begin{bmatrix} a & 0 & 0 & d \\ 0 & b & 0 & d \\ 0 & 0 & c & d \end{bmatrix}. \tag{44}$$

The question is then how to determine a, b, c, d in (43) from p, q, r, s, t in (41). It turns out that this can be solved linearly. Three linearly independent constraints are provided by

$$\begin{bmatrix} p & r & 0 \\ q & 0 & t \\ 0 & s & l \end{bmatrix}\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \mathbf{0}, \tag{45}$$

and 3 more constraints (2 of which are linearly independent) are provided by

$$(d-a, d-b, d-c)\begin{bmatrix} 0 & p & q \\ r & 0 & s \\ t & -(p+q+r+s+t) & 0 \end{bmatrix} = \mathbf{0}. \tag{46}$$

These 6 constraints admit a least squares solution for a, b, c, d to be computed from p, q, r, s, t. Finally, back in the original measurement domain, the structure of the reconstruction is given by the dual of the dual reconstruction,

$$\mathbf{X}_i = \mathbf{E}_i, \quad i = 1 \ldots 4 \tag{47}$$

$$\mathbf{X}_5 = (1, 1, 1, 1)^\mathsf{T} \tag{48}$$

$$\mathbf{X}_6 = (a, b, c, d)^\mathsf{T}, \tag{49}$$

where $\mathbf{E}_i$ are the standard basis vectors of $\mathbb{R}^4$. The camera matrices $\mathbf{P}_j$ can be computed in the original measurement domain by resectioning [2, sec. 7.1], using the original measurements $\mathbf{x}_i^j$ and the reconstructed structure $\mathbf{X}_i$.

Because there may be up to 3 real solutions to (42) (only one of which is correct), it is recommended in [2, alg 16.4] to make the function for computing the tensor output 3 possible results, all of which must then be tested within the RANSAC framework. We find this solution undesirable because it not only results in three times as much wasted computation, but also makes the output of the function 'messy.'

We have noticed that the initial search for triplet correspondences typically involves computing the fundamental matrix between the first two views, $\mathbf{F}_{21}$, to rule out bad matches that do not need to be searched for in the 3rd view. Therefore, we pass $\mathbf{F}_{21}$ into the minimal triplet routine and use it to select the correct solution which has the same fundamental matrix when there are 3 unique solutions (our second improvement).

3.2. Linear Algorithm

The following algorithm is from Hartley [10], using the principles first developed in Shashua and Werman [11]. A correspondence of three points $\mathbf{x} \leftrightarrow \mathbf{x}' \leftrightarrow \mathbf{x}''$ that are the images of one structure point $\mathbf{X}$ in each of the respective views gives rise to 9 linear constraints on $\mathcal{T}$. These constraints are not easily written in matrix notation, but can be expressed in tensor notation as

$$x^i(x'^j\epsilon_{jpr})(x''^k\epsilon_{kqs})\mathcal{T}_i^{pq} = 0_{rs}, \tag{50}$$

where ϵ is the Levi-Civita symbol, x^i are the elements of $\mathbf{x}$, and a similar notation is used to denote the elements of $\mathbf{x}'$ and $\mathbf{x}''$. Only 4 of the 9 equations represented by (50) are linearly independent, so it is not necessary to use all of them. One choice of 4 linearly independent equations is given, after simplification, by

$$x^k(x'^i x''^l\mathcal{T}_k^{33} - x''^l\mathcal{T}_k^{i3} - x'^i\mathcal{T}_k^{3l} + \mathcal{T}_k^{il}) = 0, \quad \forall i, l \in \{1, 2\}. \tag{51}$$

These equations can be arranged into a homogeneous linear system,

$$\mathbf{A}\mathbf{t} = 0, \tag{52}$$

where $\mathbf{t}$ is a vector containing the elements of $\mathcal{T}$, and $\mathbf{A}$ is a constraint matrix containing 27 or more linearly independent rows. A least squares solution is obtained by minimizing $\|\mathbf{A}\mathbf{t}\|$ subject to $\|\mathbf{t}\| = 1$, which can be accomplished using SVD [2, alg A5.4].

There are primarily two limitations of this direct solution. First, none of the 8 internal constraints are enforced, so the tensor is not a consistent representation of any geometrical configuration. Second, the algebraic error that is minimized by SVD has no particular geometric meaning.

Because the error minimized by the linear solution has no particular geometric meaning, it is not surprising that the solution is not invariant to a scaling or translation of the image points. It has been noticed that normalizing the correspondence data generally leads to improved estimation accuracy [9, 12].

That is, instead of estimating $\mathbf{P}_i$ directly in $\mathbf{x} = \mathbf{P}_i\mathbf{X}$, it is recommended to replace $\mathbf{x}$ by $\tilde{\mathbf{x}} = \mathbf{H}_i\mathbf{x}$, where $\mathbf{H}_i$ is a 3×3 translation-scaling matrix constructed such that the distribution of points $\tilde{\mathbf{x}}$ in the ith image is centered around $(0, 0)$ and has a standard deviation of $\sqrt{2}$. Thus, one actually estimates $\tilde{\mathbf{P}}_i = \mathbf{H}_i\mathbf{P}_i$, and then maps the result back to $\mathbf{P}_i = \mathbf{H}_i^{-1}\tilde{\mathbf{P}}_i$.

3.2.1. Choosing Equations

In constructing the constraint matrix $\mathbf{A}$, there are a few different approaches that could be used. One option is to select only 4 of the 9 equations which are linearly independent, as in (51), for improved performance. However, it has been suggested that using all 9 constraints in (50) might give better results [10]. A theoretical argument for using all 9 constraints was given in [2, sec 17.7], where it was noted that the condition of the full set of equations is better, and therefore using all equations might help to avoid difficulties in near singular situations.

A third option is to translate the point-point-point correspondences into point-line-line correspondences [2, sec. 17.7]. Given a correspondence between a point $\mathbf{x}$ in the first view, which is known to lie on a line $\mathbf{l}' = (l'_1, l'_2, l'_3)^\mathsf{T}$ in the second view and $\mathbf{l}'' = (l''_1, l''_2, l''_3)^\mathsf{T}$ in the third view, then there is one constraint on the tensor given by

$$x^i l'_q l''_r \mathcal{T}_i^{qr} = 0. \tag{53}$$

For each point-point-point correspondence, it is easy to generate 4 linearly independent point-line-line correspondences

in the following manner: let $\mathbf{l}'^1$ and $\mathbf{l}'^2$ be two lines passing through $\mathbf{x}'$, and $\mathbf{l}''^1$ and $\mathbf{l}''^2$ be two lines passing through $\mathbf{x}''$. Then, the 4 constraints are given by

$$x^i l'^j_q l''^k_r \mathcal{T}_i^{qr} = 0, \quad \forall j,k \in \{1,2\}. \tag{54}$$

If $\mathbf{l}'^1$ and $\mathbf{l}'^2$ are orthonormal, and $\mathbf{l}''^1$ and $\mathbf{l}''^2$ are orthonormal, then the resulting constraint matrix will have the same SVD as if all 9 point-point-point constraints had been used, and therefore give the same solution for lower computational cost.

It was suggested to find these orthonormal lines using Householder matrices, but we note that it is simpler to just use the horizontal and vertical lines passing through the point. Given a point $(x, y, 1)^\mathsf{T}$ in the image, the vectors representing these lines are given by

$$\mathbf{l}_h = \frac{(1, 0, -x)^\mathsf{T}}{\sqrt{1+x^2}} \qquad \mathbf{l}_v = \frac{(0, 1, -y)^\mathsf{T}}{\sqrt{1+y^2}}. \tag{55}$$

3.2.2. Enforcing Internal Constraints

A reconstruction from projection constraints alone is, at best, ambiguous up to an arbitrary projective transform having 15 degrees of freedom (dof) in homogeneous space. Each projection matrix has 11 dof, so there are $11m-15$ dof to the projective geometry representing any configuration of m views [2, sec. 17.5]. Thus, the projective geometry of 3 views has 18 dof.

The tensor is a homogeneous entity with 27 elements, so it has 26 dof, and this means that a geometrically consistent trifocal tensor must satisfy $26 - 18 = 8$ independent algebraic constraints. These constraints are implicitly enforced by the minimal estimation algorithm (Section 3.1), but cannot be directly enforced in the linear method (52). However, when mapping the tensor into projection matrices for general use with the algorithm of Section 2.1, one naturally obtains a geometrically consistent representation because projection matrices have no internal constraints. The problem with this passive approach to constraint enforcement is that the estimation is adjusted to satisfy internal consistency without regard to the image correspondence constraints, and this could potentially result in a very large increase in reprojection error.

A better solution is to reestimate a consistent tensor in a second linear step by holding some aspects from the original estimation fixed. We refer to these as quasi-linear methods. The first such method was given in Hartley [10], where it was pointed out that if $\mathbf{a}_4$ and $\mathbf{b}_4$ from (13) are known, then the tensor may be expressed linearly in terms of the remaining elements of the projection matrices. Specifically, one can write

$$\mathbf{t} = \mathbf{E}\mathbf{a}, \tag{56}$$

where $\mathbf{a}$ contains all the elements of $\mathbf{a}_i, \mathbf{b}_i \; \forall i \in \{1,2,3\}$, and $\mathbf{E}$ is a constraint matrix based on the known $\mathbf{a}_4$ and $\mathbf{b}_4$. From (26-34) it can be seen that, without loss of generality, one can choose $\mathbf{a}_4 = \mathbf{e}'$ and $\mathbf{b}_4 = \mathbf{e}''$, which can be extracted from the initial linear estimate using (19-20). Plugging (56) into (52), one obtains

$$\mathbf{A}\mathbf{E}\mathbf{a} = 0. \tag{57}$$

Thus, the initial problem (52) of minimizing $\|\mathbf{A}\mathbf{t}\|$ subject to $\|\mathbf{t}\| = 1$ is analogous to minimizing $\|\mathbf{A}\mathbf{E}\mathbf{a}\|$ subject to $\|\mathbf{E}\mathbf{a}\| = 1$, but the latter guarantees a geometrically consistent result.

Because this is not in the traditional form that can be easily solved by taking the right singular vector of $\mathbf{A}\mathbf{E}$, and the $\|\mathbf{E}\mathbf{a}\| = 1$ constraint has no geometrical significance beyond preventing the trivial solution of $\mathbf{a} = 0$, one might instead minimize $\|\mathbf{A}\mathbf{E}\mathbf{a}\|$ subject to $\|\mathbf{a}\| = 1$. However, because $\mathbf{E}$ is not full rank, the solution vector would not be uniquely determined. In order to ensure a unique solution, it was suggested to use additional constraints of

$$\mathbf{a}_i \cdot \mathbf{a}_4 = 0 \quad i \in \{1,2,3\}, \tag{58}$$

which it was shown can be imposed without loss of generality. These additional constraints can be written as a system of linear equations by constructing an appropriate matrix $\mathbf{C}$ in

$$\mathbf{C}\mathbf{a} = 0, \tag{59}$$

and the minimization of $\|\mathbf{A}\mathbf{E}\mathbf{a}\|$ subject to $\|\mathbf{a}\| = 1$ and $\mathbf{C}\mathbf{a} = 0$ can be performed linearly using [2, alg A5.5].

It is not obvious if the addition of these latter constraints would actually be beneficial because the non-unique solutions would still be equivalent under the projective ambiguity, and the potential downside is that the degree to which the real trilinear constraints are violated must be increased in order to reduce the error on these artificial constraints.

Shortly thereafter in Hartley [12], it was shown that the problem of minimizing $\|\mathbf{A}\mathbf{E}\mathbf{a}\|$ subject to $\|\mathbf{E}\mathbf{a}\| = 1$ could be solved directly ([2, alg A5.6]), and this has become Hartley's recommended method in Hartley and Zisserman [2, alg 16.2].

To summarize, there are three interesting quasi-linear methods that may have similar performance, and we put all three variations to the test in our empirical comparison:

$$\min \|\mathbf{A}\mathbf{E}\mathbf{a}\| \text{ subject to } \|\mathbf{a}\| = 1 \tag{60}$$

$$\min \|\mathbf{A}\mathbf{E}\mathbf{a}\| \text{ subject to } \|\mathbf{a}\| = 1 \text{ and } \mathbf{C}\mathbf{a} = 0 \tag{61}$$

$$\min \|\mathbf{A}\mathbf{E}\mathbf{a}\| \text{ subject to } \|\mathbf{E}\mathbf{a}\| = 1. \tag{62}$$

3.3. Algorithms not Considered

There exist a number of iterative algorithms for estimating the trifocal tensor, such as using the Sampson approximation [2, sec 16.4.3] (first used for conic fitting by Sampson in [19]), iterative adjustment of the epipoles [2, sec 16.3], iterative adjustment of the image points [14], the nonlinear algorithm from [20], or nonlinear enforcement of internal constraints [12]. We do not consider these nonlinear algorithms because they all require an initial estimate (e.g., found by the linear method), and

once an initial geometrically valid tensor has been found, it can be converted into camera matrices without loss of information and then bundle adjustment is the maximum likelihood nonlinear improvement. Therefore, we concentrate our search only on finding the best geometrically consistent initialization.

We do not consider parameterizations of the linear algorithm using reduced affine coordinates such as [21] because none of the error is distributed onto the estimation in the first view, and this will necessarily yield inferior results in comparison to a solution that evenly distributes the error across all views.

We do not consider the linear Factorization method [22] (or its variations), because it assumes orthographic projection which is a crude approximation to perspective projection. Although there exist nonlinear methods to correct for perspective effects, such as in [23], the initial orthographic solution might not be in the basin of attraction of the perspective correct solution. Therefore, it is an inferior approach to the linear algorithm which properly accounts for perspective in the initial solution.

Finally, we do not consider globally optimal approaches to estimate the trifocal tensor using branch and bound [24] because the exponential time complexity of this approach admittedly makes it impractical for general use, much less integration into a framework requiring many repeated evaluations such as RANSAC.

4. Robust Estimation with RANSAC

In practice, a correspondence set will usually contain some mismatches (outliers) that would be inconsistent with the true reconstruction. A robust procedure for dealing with outliers in any model fitting problem is RANSAC [13], and is often applied to computation of the trifocal tensor, as in Torr and Zisserman [14]. There have been many improvements to the original RANSAC algorithm (see Raguram et al. [25] for a survey) but we mention only the basic algorithm here for simplicity.

The objective of RANSAC is to find the largest sample consensus; i.e., to find the model that is consistent with the largest subset of the data. This is achieved by picking many random subsets, creating an initial reconstruction from each subset, classifying inliers according to a threshold, and storing the model with the largest set of inliers that was found.

The usual way to choose the number of trials needed is by a probabilistic argument [13]: if the size of each random subset is s and the percent of inliers is p, then the probability of picking a subset of all inliers is p^s. If, after n trials, no trial subset has contained all inliers, then the overall result is failure. Thus, the probability of failure f is given by

$$f = (1 - p^s)^n. \tag{63}$$

Rearranging, one can solve for the minimum number of trials needed to meet any given probability of failure,

$$f = (1 - p^s)^n \tag{64}$$

$$n = \log_{(1-p^s)} f = \frac{\log f}{\log(1 - p^s)}. \tag{65}$$

Although p is not usually known in advance, it can be increased adaptively whenever a new larger sample consensus is found until the termination condition is exceeded.

It is clear that the number of iterations required grows exponentially with the size of the initial pick set, s. Thus, it is usually recommended to choose the smallest possible s. In the case of the trifocal tensor the minimal size is $s = 6$, using the minimal algorithm (Section 3.1).

5. Experimental Results

We start by trying to find the best variation of the linear algorithm by comparing the different ways to enforce internal constraints and represent the trilinear constraints (Section 5.1). Once we have identified the best linear variation, we compare the minimal 6 point algorithm to the best linear variation with 7 points in terms of accuracy and runtime performance (Section 5.2). Lastly, we investigate the effect of the number of points used (either 6 for the minimal algorithm ,or 7+ with the linear method) on the overall performance on RANSAC (Section 5.3).

In most of our tests we have used synthetic data where the levels of noise can be precisely controlled to more accurately investigate the dependence on noise. We generate synthetic correspondences from uniformly distributed 3D structure points in a $[-50, 50]^3$ volume imaged by camera views on a circle having random radius uniformly distributed in the range $(200, 1000)$. Each camera has a 45° field of view with principal point in the center of the image, and the separation between each camera on the circle is uniformly distributed in the range of $(0.01, 5)$ degrees. Correspondences are generated by projecting the structure points into each image plane and adding uniformly distributed random noise in the range $(-\varepsilon, \varepsilon)$ pixels, for a noise level of ε.

5.1. Best Linear Variation

We start by comparing the three methods of quasi-linear reestimation to enforce internal constraints. We first plotted the median of the mean reprojection error from 1000 repetitions as a function of noise (Fig. 2a). Note that we have only considered $\varepsilon < 1$, because generally the precision of a correspondence finder is limited by the image discretization. However, we also note that effective noise is relative to object distance.

Our results indicate that (60) actually increased the error in comparison to the passive method (Section 2.1), whereas minimizing either (61) or (62) reduced the error by roughly 50%, with no noticeable difference between the two. However, we observed some sensitivities when minimizing (62), and a plot of reconstruction error as a function of the SVD precision tolerance (Fig. 2b), with the noise level fixed at $\varepsilon = 0.5$, shows that in fact (62) is a much less stable algorithm. Specifically, (62) demands a precision of at least 1×10^{-15} to give good results, whereas (61) was insensitive to the SVD precision, requiring fewer iterations for the same accuracy. We therefore conclude that (61) is the superior way to enforce internal constraints, despite it being an older and lesser known method.

Next, we considered the various methods for representing trilinear constraints described in Section 3.2.1. These methods

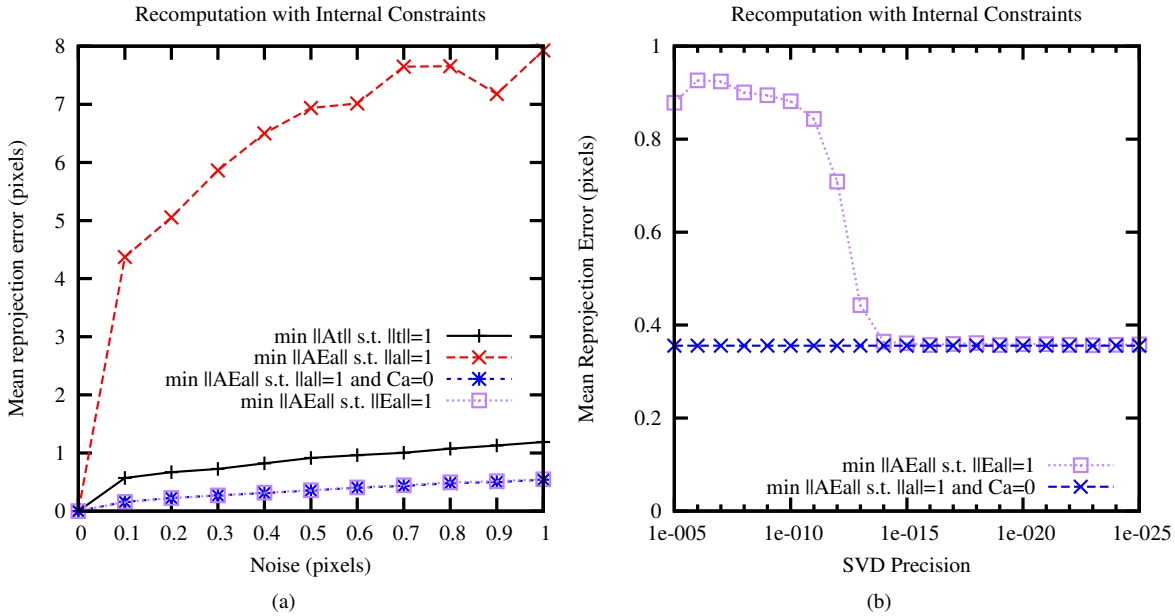

Figure 2. Comparison of methods for enforcing internal constraints in the linear algorithm by quasi-linear reestimation. The minimization of $\|\mathbf{At}\|$ s.t. $\|\mathbf{t}\| = 1$ is the basic linear algorithm, and constraint enforcement is done passively when mapping back to projection matrices (Section 2.1); the minimization of $\|\mathbf{AEa}\|$ s.t. $\|\mathbf{a}\| = 1$ and $\mathbf{Ca} = 0$ ((61)) is the quasi-linear re-estimation method from Hartley [10]; the minimization of $\|\mathbf{AEa}\|$ s.t. $\|\mathbf{a}\| = 1$ ((60)) investigates the necessity of the $\mathbf{Ca} = 0$ constraint; the minimization of $\|\mathbf{AEa}\|$ s.t. $\|\mathbf{Ea}\| = 1$ ((62)) is the method from Hartley [12]. (a) the mean reprojection error for each estimation method is shown as a function of correspondence noise, with the median over 1000 trials is plotted. (b) although the difference between (61) and (62) is imperceptible from (a), the error as a function of the SVD precision tolerance, with $\varepsilon = 0.5$, shows that (62) is much less stable and requires more iterations for a reliable result. This plot is also the median over 1000 trials.

were tested on configurations of 100 structure points using the 7 point linear method. We measured the reprojection error for the 7 fitted points (Fig. 3, left), as well as the remaining 93 points (Fig. 3, middle) by looking at the residual errors from maximum likelihood triangulation. As before, we plotted the median results over 1000 trials for each level of noise.

From previous theoretical arguments (Section 3.2.1), one would expect to see equivalent results using either all 9 point-point-point constraints (`9ppp`) or the 4 point-line-line constraints (`4pll`), and slightly worse results using just the 4 linearly independent point-point-point constraints (`4ppp`). However, our results showed no significant difference in median performance. In order to see if there was a difference in worst-case performance, we also analyzed the histogram of performance from 100,000 random configurations (Fig. 3, right). Surprisingly, the distribution of performance appears exactly identical. Therefore, we conclude that there is no justification for using the more computationally expensive `9ppp` method, and there is also no need to complexify the implementation by translating the point-point-point constraints into point-line-line constraints; in other words, we conclude that it is best to simply use the four linearly independent point-point-point constraints (`4ppp`).

5.2. Minimal vs. Linear

Having identified the best linear variation, we are now prepared to compare the performance of the linear method to the minimal 6 point method. This was done by generating configurations of 100 points and then reconstructing from a subset of 6 points using the minimal method or 7 points using the linear method.

We first plot the median of the mean reprojection error on the fitted data (Fig. 4, left) as an indicator of precision. Because the minimal 6 point method is an exact solution it is expected to have zero error, and this is confirmed in the plot. We measured actual error on the order of 1×10^{-12} which is due only to limited numerical precision. The linear algorithm has non-zero error that increases with noise because it is over-determined, and the fact that the reconstruction error remains proportional to and slightly less than the noise indicates that it is capable of fitting the data well.

A second graph showing the median of the mean reprojection error for additional testing correspondences after maximum likelihood triangulation (Fig. 4, middle) shows how accurate the reconstruction actually was; here we see that the minimal 6 point algorithm, while precise, is much less accurate because it does not fit the testing points nearly as well as the linear algorithm for any non-zero level of noise.

A third graph shows the median of the mean error on all available data after bundle adjustment, which shows that even though the minimal initialization was worse, both methods are usually in the basin of attraction of the global minimum.

Runtime performance between the minimal 6 point method and the linear with 7 or more points was compared with a plot of the mean reconstruction runtime for 1000 random configurations (Fig. 5). We observed that the linear method exhibits $O(n)$ performance, at least when n (the number of points) was

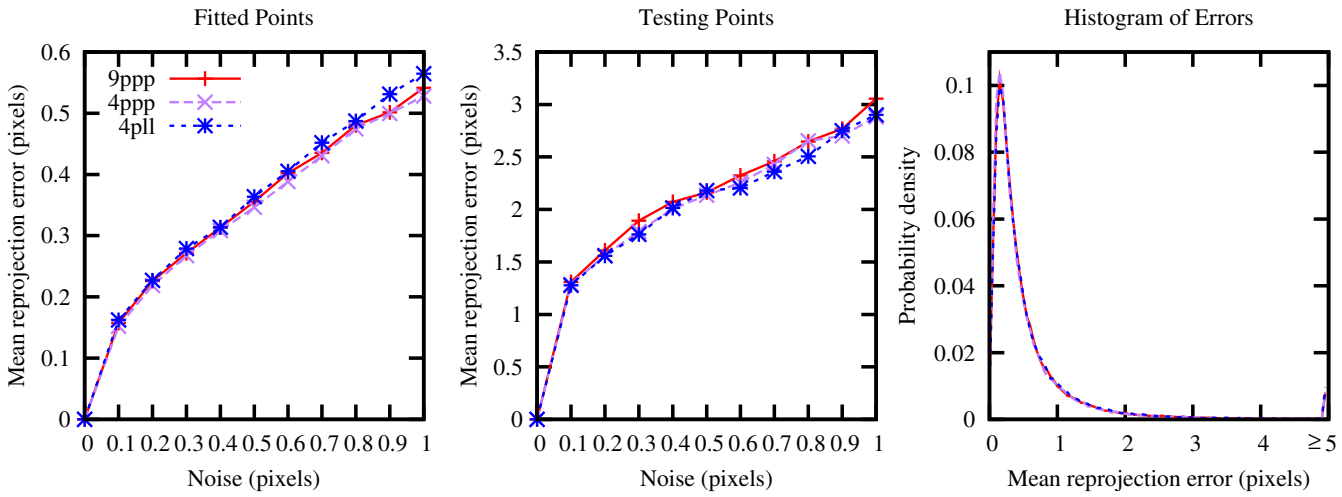

Figure 3. Effect of choosing different linear constraints in using the 7 point linear algorithm. Data sets were generated from 100 points. Left: mean reprojection error for the fitted data (first 7 points). Plot shows the median over 1000 trials. Middle: mean reprojection error for the testing data set (remaining 93 points), determined by triangulation minimizing the L_2-norm of reprojection errors. Plot shows the median over 1000 trials. Right: comparison between empirical PDF of mean reprojection error on the fitted data for each method, determined from 100,000 trials. Correspondence noise was set to $\varepsilon = 0.5$.

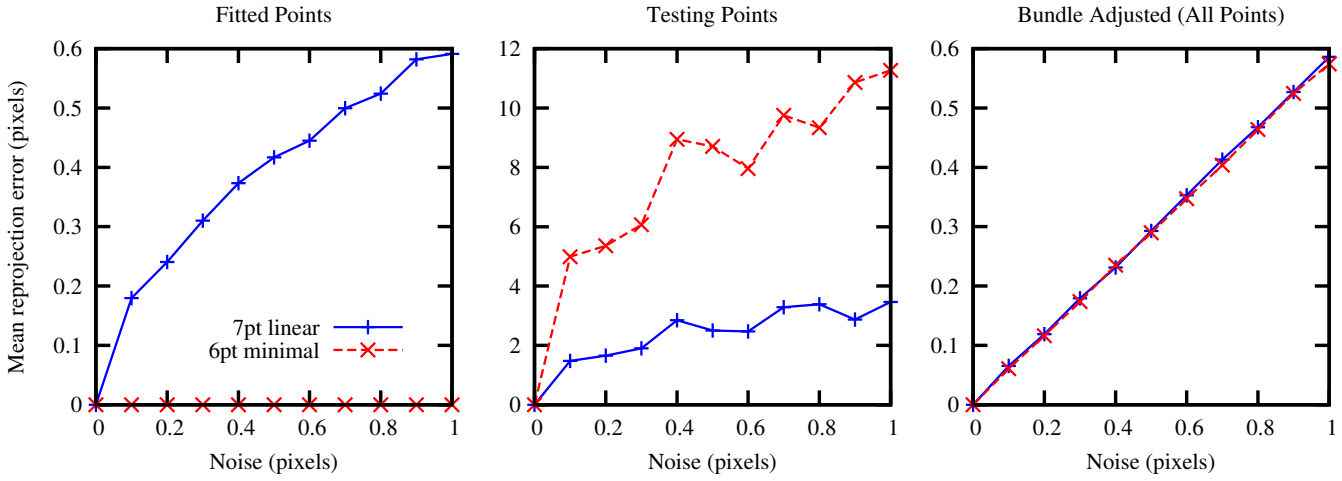

Figure 4. Comparison between minimal 6 point algorithm and best-performing variation of the linear 7 point algorithm. The left panel shows errors on the fitted data, indicating the level of precision. The middle panel shows reprojection errors after triangulation on the 100 testing correspondences, indicating the accuracy of the reconstruction. The right panel shows the result of finalizing with bundle adjustment on all available data.

less than 80, despite the fact that the computational cost of SVD is $O(n^3)$. A linear regression gives shows that the performance of our implementation of the n-point linear method is about $0.096125n+0.503315$ microseconds on the testing machine (Intel Core i7 920), compared to 0.124317 microseconds for the 6 point minimal method. In other words, the minimal algorithm is significantly faster, but the linear algorithm is still quite fast for practical purposes.

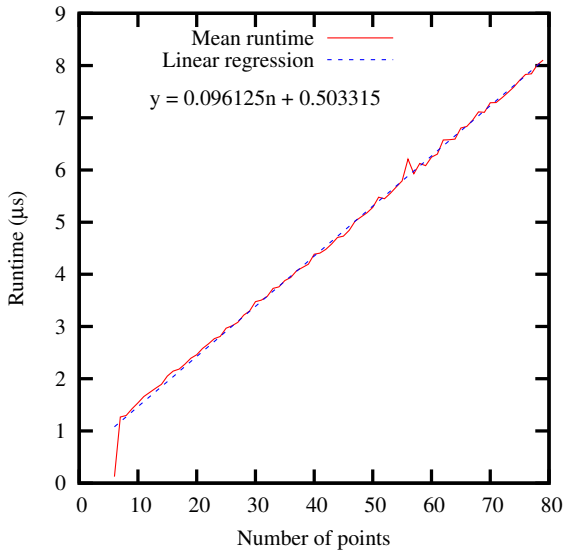

Figure 5. Mean reconstruction runtime as a function of the number of points used. The minimal algorithm is used for 6 points and the linear algorithm is used for 7 or more points.

It has been assumed that the minimal method will require the fewest iterations for RANSAC convergence, and in addition we have shown that the minimal algorithm by itself is significantly faster than the linear algorithm. However, we have speculated that the robustness to noise gained by the over-determined nature of the linear algorithm may actually lead to superior performance. We first investigated this by analyzing the size of the largest consensus size as a function of RANSAC trials using each of the minimal, 7 point linear, and 15 point linear algorithms on a random configuration (Fig. 6). The configuration consisted of 100 structure points with 80% inliers. The experiment was repeated at three noise levels for $\varepsilon = \{0, 0.5, 1\}$, and the RANSAC inlier threshold was fixed at $\tau = 1.75$. The most interesting observation from these results is that, in the presence of noise, using a larger number of points allowed RANSAC to converge to a larger final consensus size.

5.3. Subset size in RANSAC

The following experiments were designed to examine the effect of varying subset size in RANSAC in further detail. First, we looked at the accuracy of the linear method as a function of the number of points n, for $n = 7, \ldots, 80$, to see how many points are necessary before one reaches diminishing returns in the accuracy of the reconstructed tensor. We fixed the correspondence error level at $\varepsilon = 0.5$ and generate correspondences from configurations of 100 points.

Looking at the mean reprojection error on just the fitted data before and after bundle adjustment (Fig. 7, left) indicates how close the linear estimate is to the maximum likelihood estimate. We see that the maximum likelihood estimate is significantly better for $n < 10$ points, but with about 15 or more points, the linear estimate is almost as good as a maximum likelihood estimate. We also looked at how well the remaining data points fit with this model before and after bundle adjusting using all data (Fig. 7, right). Here, we see also that a linear estimate from 10-15 points is typically capable of fitting all the remaining points very well. Using more points in the initial linear estimate causes an asymptotic convergence to the true configuration, but the returns are diminishing.

The ideal number of points to use in RANSAC depends upon both the inlier fraction and the noise distribution. To demonstrate this we define the RANSAC Rerformance Ratio (PR) to be the total number of inliers divided by the total runtime. We then calculated the subset sizes that empirically optimize the performance ratio over 100 courses of running RANSAC at various combinations of noise and inlier percentages (Fig. 8). Inliers were corrupted with normally distributed noise having σ at the specified level whereas outliers were corrupted with noise having $\sigma = 50$ pixels. The inlier threshold was set automatically at $\max(0.5, 3\sigma)$ and RANSAC was run until at least 95% of the inliers were found.

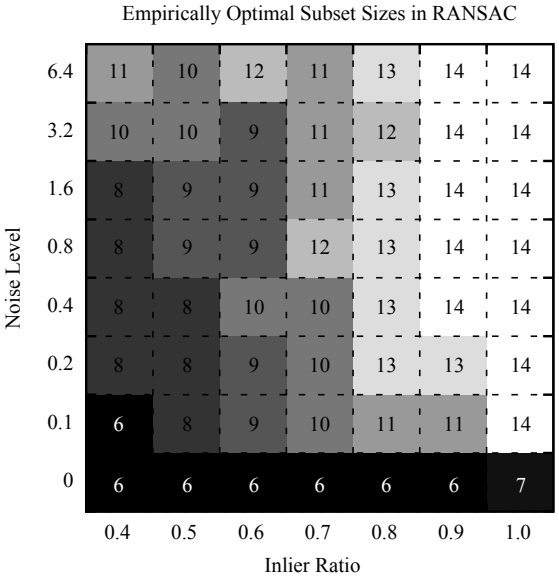

Figure 8. Empirically optimal subset sizes that maximize the summed performance ratio of final consensus sizes divided by total runtimes after running RANSAC 100 times. The minimal 6 point algorithm has a better performance ratio when there is zero noise, but a linear algorithm using more points gives superior performance when noise is introduced.

In these synthetic tests, we see clearly that the minimal 6 point algorithm maximizes the performance ratio only for zero noise. Even with an unrealistically low level of noise (σ0.1 pixels), the linear method has a better performance ratio. In general, as either the noise level or the inlier fraction is increased, the benefits of using a larger subset size are increased. However, we note that the noise is relative to scene configura-

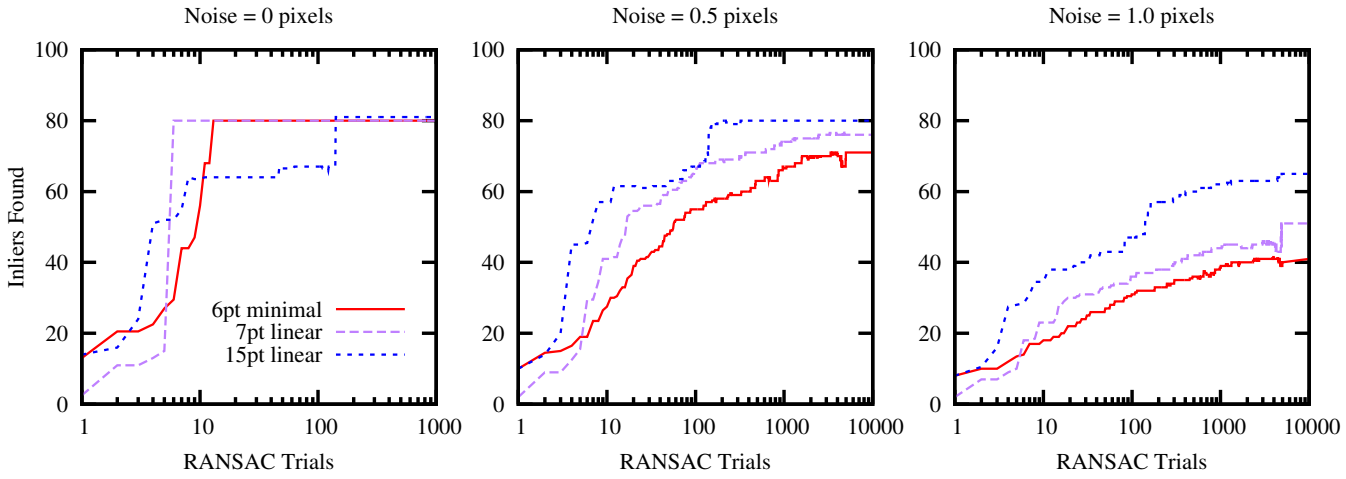

Figure 6. Comparison of RANSAC convergence using the 6 point algorithm, and the best linear variation from 7 and 15 points. The data set contained 100 points, of which 20 were outliers($p = 0.8$). The experiment is repeated using correspondence noise levels of $\varepsilon \in \{0, 0.5, 1\}$. The inlier threshold was fixed at $\tau = 1.75$ pixels. The median size of the largest consensus set over 100 random data sets is plotted as a function of RANSAC iterations.

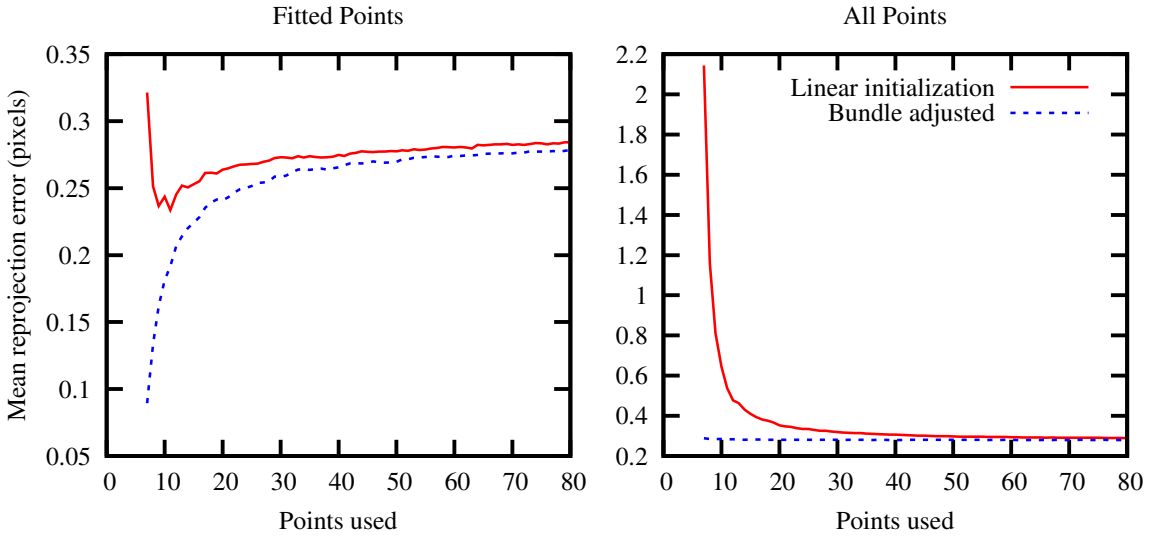

Figure 7. Dependence of linear reconstruction quality on the number of points used (median over 200 trials). The left panel shows reprojection errors on the fitted data, before and after bundle adjustment. The right panel shows reprojection errors on all available data (from 100 points), where additional points are initialized by triangulation, before and after bundle adjustment.

tion, and therefore this table should not be used as a reference for choosing the subset size of real configurations based on the noise level and inlier fraction.

In order to see what size performs best on real image data we generated correspondences by automatically matching Harris [26] corner points using the Normalized Cross Correlation. For the *Bookshelves* scene (Fig. 9), we found a total of 1369 triplet correspondences. We plotted the overall consensus size and runtime of RANSAC as a function of the number of points (Fig. 9a), with the corresponding performance ratio plotted in (Fig. 9b). The best performance was found using the 8 point linear method. We obtained a final consensus size of $1172/1369$ points, and after bundle adjustment, the mean squared reprojection error was reduced to 0.159661 pixels (relative to the 1148×764 images).

Our results on another real scene, the *Desk* scene, (Fig. 10), shows similar results; this time we found a total of 1340 triplet correspondences. The overall consensus size and runtime of RANSAC as a function of the number of points is plotted in Fig. 10a, with the corresponding performance ratio plotted in (Fig. 10b). The best performance was again found using the 8 point linear method. We obtained a final consensus size of $1003/1340$ points, and after bundle adjustment, the mean squared reprojection error was reduced to 0.192335 pixels (relative to the 1024×768 images).

6. Conclusions

We have introduced two small improvements to the minimal 6 point algorithm, one being a method for selecting points that form a stable basis in order to ensure precise results, and the other being a method for selecting the correct solution when there are 3 possible solutions.

We have also examined several variations of the linear algorithm in order to the determine the most accurate and efficient variation. We have shown that an older, lesser used, method of quasi-linear enforcement of the internal constraints actually performs best, and that there appears to be no difference in performance between the various methods of trilinear constraint representation, which leads us to believe that it is best to stick with the simplest and fastest method.

Contrary to previous results, we show that the linear method can provide a substantially more accurate estimate than the minimal method, and is nearly a maximum likelihood estimate when estimated from more than 10 points. We also show that using larger subset size in RANSAC with the linear method allows a larger final consensus size to be reached, and in a shorter overall runtime, despite the fact that runtime for the minimal method by itself is substantially faster.

Acknowledgements

We thank Dr. Margaret J. Eppstein for her useful comments.

References

[1] B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon, Bundle Adjustment – A Modern Synthesis, in: B. Triggs, A. Zisserman, R. Szeliski (Eds.), Vision Algorithms: Theory and Practice, vol. 1883 of *Lecture Notes in Computer Science*, Springer-Verlag, 298–372, URL `http://lear.inrialpes.fr/pubs/2000/TMHF00`, 2000.

[2] R. I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, ISBN: 0521540518, second edn., 2004.

[3] M. Lourakis, A. Argyros, The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm, Tech. Rep. 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, available from http://www.ics.forth.gr/ lourakis/sba, 2004.

[4] D. Nistér, An Efficient Solution to the Five-Point Relative Pose Problem, IEEE Trans. Pattern Anal. Mach. Intell. 26 (6) (2004) 756–777, ISSN 0162-8828, doi:http://dx.doi.org/10.1109/TPAMI.2004.17.

[5] B. Triggs, Matching constraints and the joint image, in: ICCV '95: Proceedings of the Fifth International Conference on Computer Vision, IEEE Computer Society, Washington, DC, USA, ISBN 0-8186-7042-8, 338, 1995.

[6] L. Quan, Invariants of Six Points and Projective Reconstruction From Three Uncalibrated Images, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1) (1995) 34–46, ISSN 0162-8828, doi: http://dx.doi.org/10.1109/34.368154.

[7] S. Carlsson, D. Weinshall, Dual Computation of Projective Shape and Camera Positions from Multiple Images, Int. J. Comput. Vision 27 (3) (1998) 227–241, ISSN 0920-5691, doi: http://dx.doi.org/10.1023/A:1007961913417.

[8] R. Hartley, G. Debunne, Dualizing Scene Reconstruction Algorithms, in: SMILE'98: Proceedings of the European Workshop on 3D Structure from Multiple Images of Large-Scale Environments, Springer-Verlag, London, UK, ISBN 3-540-65310-4, 14–31, 1998.

[9] R. Hartley, N. Dano, Reconstruction from six-point sequences, in: Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, vol. 2, 480 –486 vol.2, doi:10.1109/CVPR.2000.854888, 2000.

[10] R. Hartley, A linear method for reconstruction from lines and points, Computer Vision, IEEE International Conference on 0 (1995) p. 882, doi: http://doi.ieeecomputersociety.org/10.1109/ICCV.1995.466843.

[11] A. Shashua, M. Werman, Trilinearity of three perspective views and its associated tensor, in: ICCV '95: Proceedings of the Fifth International Conference on Computer Vision, IEEE Computer Society, Washington, DC, USA, ISBN 0-8186-7042-8, 920, 1995.

[12] R. Hartley, Minimizing algebraic error in geometric estimation problems, in: Computer Vision, 1998. Sixth International Conference on, 469 –476, doi:10.1109/ICCV.1998.710760, 1998.

[13] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (1981) 381–395, ISSN 0001-0782, doi:http://doi.acm.org/10.1145/358669.358692, URL `http://doi.acm.org/10.1145/358669.358692`.

[14] P. H. S. Torr, A. Zisserman, Robust parameterization and computation of the trifocal tensor, Image and Vision Computing 15 (8) (1997) 591 – 605, ISSN 0262-8856, doi:DOI: 10.1016/S0262-8856(97)00010-3, british Machine Vision Conference.

[15] M. E. Spetsakis, J. Aloimonos, Structure from Motion Using Line Correspondences, International Journal of Computer Vision 4 (3) (1990) 171–183, doi:10.1007/BF00054994, URL `http://dx.doi.org/10.1007/BF00054994`.

[16] J. Weng, T. S. Huang, N. Ahuja, Motion and Structure from Line Correspondences; Closed-Form Solution, Uniqueness, and Optimization, IEEE Trans. Pattern Anal. Mach. Intell. 14 (3) (1992) 318–336, ISSN 0162-8828, doi:http://dx.doi.org/10.1109/34.120327.

[17] A. Shashua, Algebraic Functions For Recognition, IEEE Trans. Pattern Anal. Mach. Intell. 17 (8) (1995) 779–789, ISSN 0162-8828, doi: http://dx.doi.org/10.1109/34.400567.

[18] R. I. Hartley, Lines and Points in Three Views and the Trifocal Tensor, Int. J. Comput. Vision 22 (2) (1997) 125–140, ISSN 0920-5691, doi: http://dx.doi.org/10.1023/A:1007936012022.

[19] P. Sampson, Fitting Conic Sections to 'Very Scattered' Data: An Iterarive Refinement of the Bookstein Algorithm, Computer Vision, Graphics, and Image Processing 18 (1) (1982) 97–108.

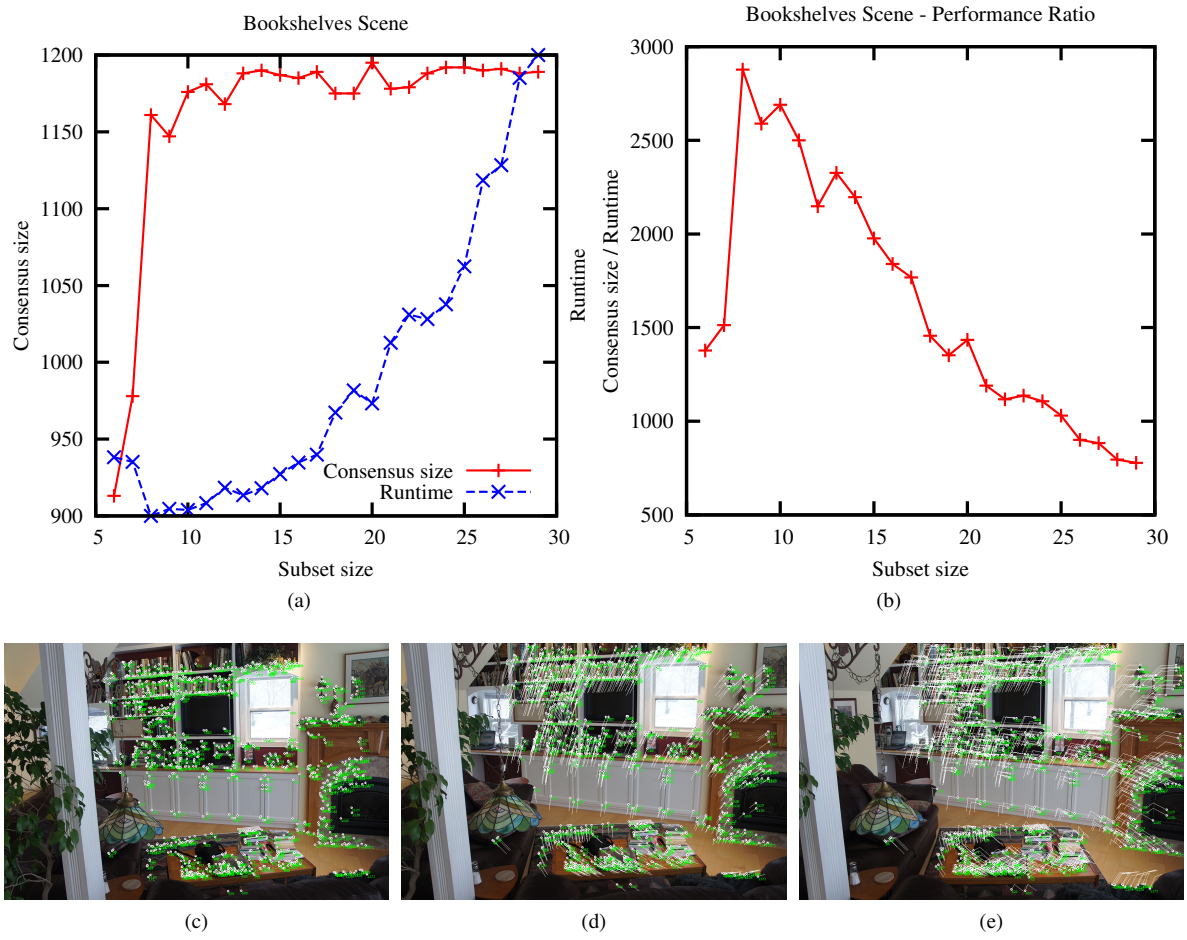

Figure 9. Example reconstruction using the trifocal tensor. The inlier threshold was automatically determined at 1.01605 pixels, and 1172 out of 1369 triplet correspondences were found as inliers. The mean squared reprojection error is 0.159661 pixels (in comparison, the image size is 1148×764 pixels).

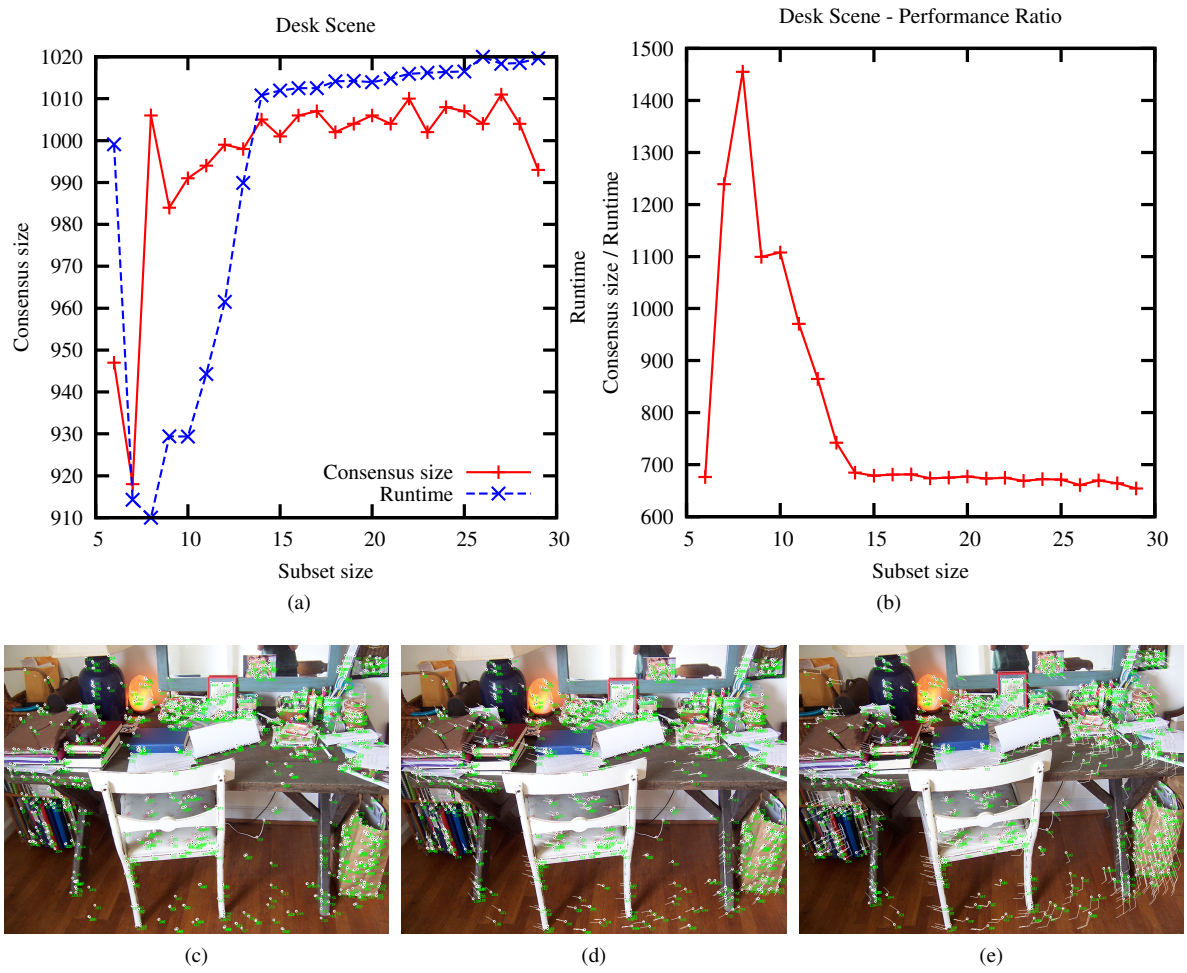

Figure 10. Example reconstruction using the trifocal tensor. The inlier threshold was automatically determined at 1.09729 pixels, and 1003 out of 1340 triplet correspondences were found as inliers. The mean squared reprojection error is 0.192335 pixels (in comparison, the image size is 1024×768 pixels).

[20] O. Faugeras, R. Keriven, Complete Dense Stereovision using Level Set Methods, in: in Proc. 5th European Conf. on Computer Vision, 379–393, 1998.

[21] A. Heyden, Reduced Multilinear Constraints: Theory and Experiments, Int. J. Comput. Vision 30 (1) (1998) 5–26, ISSN 0920-5691, doi: http://dx.doi.org/10.1023/A:1008020228557.

[22] C. Tomasi, T. Kanade, Shape and motion from image streams under orthography: a factorization method, Int. J. Comput. Vision 9 (2) (1992) 137–154, ISSN 0920-5691, doi:http://dx.doi.org/10.1007/BF00129684.

[23] S. Christy, R. Horaud, Euclidean Shape and Motion from Multiple Perspective Views by Affine Iterations, IEEE Trans. on Pattern Analysis and Machine Int. 18 (11).

[24] R. I. Hartley, F. Kahl, Global Optimization through Rotation Space Search, Int. J. Comput. Vision 82 (1) (2009) 64–79, ISSN 0920-5691, doi:http://dx.doi.org/10.1007/s11263-008-0186-9.

[25] R. Raguram, J.-M. Frahm, M. Pollefeys, A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus, in: Proceedings of the 10th European Conference on Computer Vision: Part II, Springer-Verlag, Berlin, Heidelberg, ISBN 978-3-540-88685-3, 500–513, 2008.

[26] C. Harris, M. Stephens, A Combined Corner and Edge Detector, in: Proceedings of The Fourth Alvey Vision Conference, 147–151, 1988.