Globally Optimal Hand-Eye Calibration Using Branch-and-Bound

Jan Heller, Michal Havlena, and Tomas Pajdla

Abstract—This paper introduces a novel solution to the hand-eye calibration problem. It uses camera measurements directly and, at the same time, requires neither prior knowledge of the external camera calibrations nor a known calibration target. Our algorithm uses branch-and-bound approach to minimize an objective function based on the epipolar constraint. Further, it employs Linear Programming to decide the bounding step of the algorithm.Our technique is able to recover both the unknown rotation and translation simultaneously and the solution is guaranteed to be globally optimal with respect to the L_∞-norm.

Index Terms—Hand-eye calibration, branch-and-bound algorithm, global optimization

1 Introduction

THE need to relate measurements made by a camera to a different known coordinate system arises in many engineering applications. It also appears in the connection with cameras mounted on robotic systems. The problem is commonly known as *hand-eye calibration* and has been studied abundantly in the past. Early solution methods searched for rotational and translational parts separately [2], [11], [15], [18], [19]. Such separation inevitably leads to propagation of the residual error of the estimated rotation into the translation estimation, and therefore methods for simultaneous estimation of both rotation and translation appeared [3], [8], [20], [21]. However, none of these methods work with camera measurements directly and require prior knowledge of the external camera calibrations instead. In [16], authors proposed a method based on tracking of image points. Still, an algebraic objective function, rather that a more geometrically meaningful criteria based on the original camera measurements, was minimized.

Recently, Heller et al. [6] proposed a method for optimal estimation of the translational part from camera measurements. It is a step towards a meaningfully defined optimality criterion, but it still requires prior knowledge of the relative camera rotations. Seo et al. [14] solved for the rotational part optimally but only for situation where the translational part is, or can practically expected to be, zero.

In this paper, we solve for the rotation and translation simultaneously by minimizing an objective function based on the epipolar constraint without any prior knowledge of the external camera calibration. Our method is based on the branch-and-bound (BnB) search over the space of rotations presented in [4] and it is guaranteed to converge to the optimum with respect to L_∞-norm. The presented work is an extension of [7] which appeared simultaneously with a similar approach to hand-eye calibration [12]. Even though both methods use different problem formulations and different objective functions, it can be shown that both methods provide an upper bound on the reprojection error. Here, we extend [7] with a novel more efficient cone intersection strategy and provide additional experimental results.

- J. Heller and T. Pajdla are with the Department of Cybernetics, Czech Technical University, Prague, Czech Republic. E-mail: {hellej1, pajdla}@cmp.felk.cvut.cz.
- M. Havlena is with the Institute of Geodesy and Photogrammetry, ETH Zurich, Switzerland. E-mail: michal.havlena@geod.baug.ethz.ch.

Manuscript received 13 July 2013; revised 23 July 2015; accepted 4 Aug. 2015. Date of publication 16 Aug. 2015; date of current version 8 Apr. 2016.
Recommended for acceptance by D. Forsyth.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2469299

2 Problem Formulation

Let us assume a camera that has been rigidly mounted on a robot's gripper. To find a hand-eye calibration is to determine a homogeneous transformation

$$\mathtt{X} = \begin{pmatrix} \mathtt{R}_\mathtt{X} & \mathbf{t}_\mathtt{X} \\ \mathbf{0}^\top & 1 \end{pmatrix},$$

such that rotation $\mathtt{R}_\mathtt{X} \in \mathrm{SO}(3) \subset \mathbb{R}^{3\times3}$ and translation $\mathbf{t}_\mathtt{X} \in \mathbb{R}^3$ transform points from the coordinate system of the gripper to the coordinate system of the camera.

Now, let's suppose that the gripper has been manipulated into $n+1$ poses resulting into n relative motions. These motions can be described by homogeneous transformations $\mathtt{B}_i$, $i = 1, \dots, n$ and are supposed to be known, e.g., obtained from the robot control software. The gripper motions give rise to n relative camera transformations $\mathtt{A}_i$, which are related to $\mathtt{B}_i$ through the unknown transformation $\mathtt{X}$ as

$$\mathtt{A}_i\mathtt{X} = \mathtt{X}\,\mathtt{B}_i,$$

see Fig. 1. This can be further decomposed to

$$\mathtt{R}_{\mathtt{A}_i}\mathtt{R}_\mathtt{X} = \mathtt{R}_\mathtt{X}\mathtt{R}_{\mathtt{B}_i} \quad \text{and} \quad \mathtt{R}_{\mathtt{A}_i}\mathbf{t}_\mathtt{X} + \mathbf{t}_{\mathtt{A}_i} = \mathtt{R}_\mathtt{X}\mathbf{t}_{\mathtt{B}_i} + \mathbf{t}_\mathtt{X},$$

where $\mathtt{R}_{\mathtt{A}_i}, \mathtt{R}_{\mathtt{B}_i} \in \mathrm{SO}(3)$ and $\mathbf{t}_{\mathtt{A}_i}, \mathbf{t}_{\mathtt{B}_i} \in \mathbb{R}^3$. By substituting $\mathbf{t}'_\mathtt{X} = -\mathtt{R}_\mathtt{X}^\top \mathbf{t}_\mathtt{X}$ and isolating $\mathtt{R}_{\mathtt{A}_i}$ and $\mathbf{t}_{\mathtt{A}_i}$, we get

$$\mathtt{R}_{\mathtt{A}_i} = \mathtt{R}_\mathtt{X}\mathtt{R}_{\mathtt{B}_i}\mathtt{R}_\mathtt{X}^\top \quad \text{and} \quad \mathbf{t}_{\mathtt{A}_i} = \mathtt{R}_\mathtt{X}\big((\mathtt{R}_{\mathtt{B}_i} - \mathtt{I})\mathbf{t}'_\mathtt{X} + \mathbf{t}_{\mathtt{B}_i}\big).$$

Further, suppose that the camera measured m correspondences $\mathbf{u}_{ij} \leftrightarrow \mathbf{v}_{ij}$, $j = 1, \dots, m$ in the ith motion. Through the rest of the paper we will assume that the camera's internal calibration is known [5] and that $\mathbf{u}_{ij}, \mathbf{v}_{ij} \in \mathbb{R}^3$ are unit vectors representing the directions to scene points from the respective camera positions. We will also assume that the correspondences satisfy the *cheirality condition* [5], i.e., that $\mathbf{u}_{ij}, \mathbf{v}_{ij}$ correspond to scene points $\mathbf{Y}_{ij} \in \mathbb{R}^3$ that lie in front of the cameras.

Let us consider an elementary fact from the geometry of stereo vision known as the *epipolar constraint* [5]. It states that vectors $\mathbf{u}_{ij}$ and $\mathbf{v}_{ij}$ form a correspondence for camera motion $\mathtt{A}_i$, if $\mathbf{t}_{\mathtt{A}_i}$ lies in the plane containing the two vectors. This fact is commonly expressed in the form $\mathbf{t}_{\mathtt{A}_i}^\top(\mathbf{v}_{ij} \times (\mathtt{R}_{\mathtt{A}_i}\mathbf{u}_{ij})) = 0$. Since we will work here with angular measurements, it is convenient to equivalently rephrase the constraint as

$$e_{ij} = \angle([\mathbf{v}_{ij}]_\times \mathtt{R}_{\mathtt{A}_i}\mathbf{u}_{ij}, \mathbf{t}_{\mathtt{A}_i}) - \tfrac{\pi}{2} = 0,$$

where $[\cdot]_\times$ denotes the 3×3 skew symmetric matrix such that $\forall \mathbf{u}, \mathbf{v} : [\mathbf{u}]_\times \mathbf{v} = \mathbf{u} \times \mathbf{v}$.

In case of noisy measurements, however, the epipolar constraint will not hold, i.e., e_{ij} won't generally be zeros, and the values of $|e_{ij}|$ will encode the angular deviations of the estimated camera translation $\mathbf{t}_{\mathtt{A}_i}$ from the epipolar planes defined by correspondences $\mathbf{u}_{ij} \leftrightarrow \mathbf{v}_{ij}$. This leads us to defining an *ϵ-epipolar constraint*. The correspondences $\mathbf{u}_{ij} \leftrightarrow \mathbf{v}_{ij}$ and camera translation $\mathbf{t}_{\mathtt{A}_i}$ satisfy the ϵ-epipolar constraint with parameter $\epsilon > 0$, iff $|e_{ij}| \le \epsilon$. This constraint can be equivalently expressed as

$$\tfrac{\pi}{2} - \epsilon \le \angle([\mathbf{v}_{ij}]_\times \mathtt{R}_{\mathtt{A}_i}\mathbf{u}_{ij}, \mathbf{t}_{\mathtt{A}_i}) \le \tfrac{\pi}{2} + \epsilon,$$

i.e., $\mathbf{t}_{\mathtt{A}_i}$ has to lie outside of the double cone determined by axis $[\mathbf{v}_{ij}]_\times \mathtt{R}_{\mathtt{A}_i}\mathbf{u}_{ij}$ and aperture $\pi - 2\epsilon$, see Fig. 2. Note here that as a complement of a double cone ϵ-epipolar constraint is not a convex constraint. Let us rewrite the left inequality as

$$\tfrac{\pi}{2} - \epsilon \le \angle([\mathbf{v}_{ij}]_\times \mathtt{R}_{\mathtt{A}_i}\mathbf{u}_{ij}, \mathbf{t}_{\mathtt{A}_i}) \Leftrightarrow \tfrac{\pi}{2} + \epsilon \ge \angle(-[\mathbf{v}_{ij}]_\times \mathtt{R}_{\mathtt{A}_i}\mathbf{u}_{ij}, \mathbf{t}_{\mathtt{A}_i}).$$

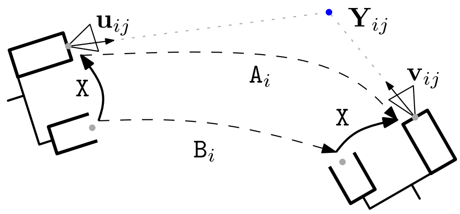


Fig. 1. A gripper-camera rig motion.

Now we can formulate hand-eye calibration as the minimization of the ϵ-epipolar constraint, i.e., as L_∞-norm minimization of the vector of residuals $\mathbf{e} = (|e_{11}|, \ldots, |e_{nm}|)$:

Problem 1.

$$(\hat{\mathtt{R}}_\mathtt{X}, \hat{\mathbf{t}}'_\mathtt{X}) = \arg\min_{\mathtt{R}_\mathtt{X}, \mathbf{t}'_\mathtt{X}} \max_{i,j} |e_{ij}| = \arg\min_{\mathtt{R}_\mathtt{X}, \mathbf{t}'_\mathtt{X}} \|\mathbf{e}\|_\infty.$$

The optimal residual error can be expressed as $\epsilon_{\min} = \|\mathbf{e}(\hat{\mathtt{R}}_\mathtt{X}, \hat{\mathbf{t}}'_\mathtt{X})\|_\infty$. After solving Problem 1, the optimal translation is determined as $\hat{\mathbf{t}}_\mathtt{X} = -\hat{\mathtt{R}}_\mathtt{X}\hat{\mathbf{t}}'_\mathtt{X}$. This substitution may seem superfluous, but it will allow us to prove Lemma 1 later.

3 The Space of Rotations

In order to solve Problem 1 we employ BnB optimization presented in [4] to search over the space of all rotations. The algorithm is based on the angle-axis parametrization of rotations, so let us first review relations between a rotation matrix and its angle-axis representations.

Let $\boldsymbol{\alpha} \in \mathbb{B}_\pi = \{\boldsymbol{\beta} : \boldsymbol{\beta} \in \mathbb{R}^3 \wedge \|\boldsymbol{\beta}\| \leq \pi\}$, then $\boldsymbol{\alpha}$ represents the rotation about axis $\boldsymbol{\alpha}/\|\boldsymbol{\alpha}\|$ by angle $\|\boldsymbol{\alpha}\|$. The corresponding matrix parametrization $\mathtt{R} \in \mathrm{SO}(3)$ can be obtained using *Rodrigues' formula* as $\mathtt{R} = \exp[\boldsymbol{\alpha}]_\times$ [5]. The inverse map is given by $[\boldsymbol{\alpha}]_\times = \log \mathtt{R}$.

For $\mathtt{R}_1, \mathtt{R}_2 \in \mathrm{SO}(3)$ we define the distance $d_\angle(\mathtt{R}_1, \mathtt{R}_2)$ as the angle θ of the rotation $\mathtt{R}_1^\top \mathtt{R}_2$, i.e., $[\boldsymbol{\alpha}]_\times = \log(\mathtt{R}_1^\top \mathtt{R}_2)$, such that $0 \leq \theta = \|\boldsymbol{\alpha}\| \leq \pi$. In the following, we will use the notation "$\mathtt{R} \in D \subset \mathbb{B}_\pi$" to mean

$$\mathtt{R} \in \{\mathtt{R}' \in \mathrm{SO}(3) : \exists \boldsymbol{\alpha} \in D \text{ such that } \mathtt{R}' = \exp[\boldsymbol{\alpha}]_\times\}.$$

4 Branch and Bound

Let us consider Problem 1 restricted to D_σ, where $D_\sigma \subset \mathbb{B}_\pi$ is a cubic block with side length 2σ:

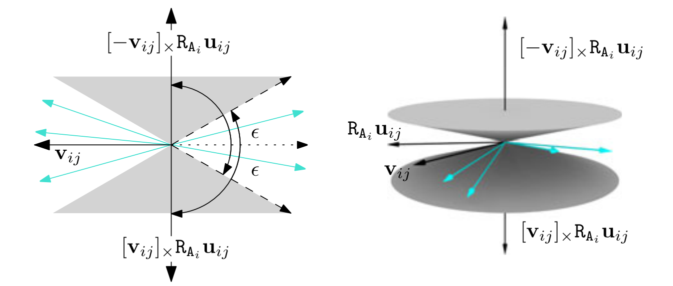
Problem 2.

$$(\hat{\mathtt{R}}_\mathtt{X}, \hat{\mathbf{t}}'_\mathtt{X}) = \arg\min_{\mathtt{R}_\mathtt{X} \in D_\sigma, \mathbf{t}'_\mathtt{X}} \max_{i,j} |e_{ij}|.$$

A schematic BnB algorithm for solving Problem 1 is now as follows:

1) Obtain an initial estimate of $\epsilon_{\min}$ for the optimal residual error of Problem 1.
2) Divide the space of rotations up into cubic subblocks D_σ^s, $s = 1, \ldots, 8$ and repeat the following steps:
   a) For each block D_σ^s, test whether there exists a solution to Problem 2 restricted on D_σ^s having the residual error smaller than $\epsilon_{\min}$. This test can be formulated as a *feasibility test*, see Section 5.
   b) If the answer to the test is no, throw the block away.
   c) Otherwise, evaluate the residual error ϵ for some rotation from block D_σ^s. If $\epsilon < \epsilon_{\min}$ then update the value $\epsilon_{\min} \leftarrow \epsilon$. Subdivide D_σ^s into eight cubic sub-blocks and continue to (a).

The iteration loop is terminated when the half-size of the blocks σ reaches a sufficiently small size $\sigma_{\min}$.



(a) (b)

Fig. 2. (a) Geometric interpretation of the ϵ-epipolar constraint with parameter ϵ imposed by the correspondence $\mathbf{u}_{ij} \leftrightarrow \mathbf{v}_{ij}$ on the position of $\mathbf{t}_{\mathtt{A}_i}$. The cyan vectors show examples of the admissible configurations of $\mathbf{t}_{\mathtt{A}_i}$. (b) 3D view of the same.

Note that although Problem 1 has six degrees of freedom (DOF), the BnB algorithm searches only over the three dimensional space of rotations. By limiting rotations in angle-axis parametrization to D_σ we are able to decide the feasibility test for Problem 2 effectively and optimally using Linear Programming (LP). The LP solution also provides $\mathbf{t}'_\mathtt{X}$ needed to compute the residual error in step (c) and thus there is no need to search over the space of translations.

5 Feasibility Test

In this section the feasibility test based on Problem 2 is formulated. Let us first introduce the following notation

$$\bar{\mathtt{R}}_{\mathtt{A}_i} = \bar{\mathtt{R}}_\mathtt{X} \mathtt{R}_{\mathtt{B}_i} \bar{\mathtt{R}}_\mathtt{X}^\top, \; \bar{\mathbf{t}}_{\mathtt{A}_i} = \bar{\mathtt{R}}_\mathtt{X}\big((\mathtt{R}_{\mathtt{B}_i} - \mathtt{I})\mathbf{t}'_\mathtt{X} + \mathbf{t}_{\mathtt{B}_i}\big),$$
$$\hat{\mathtt{R}}_{\mathtt{A}_i} = \hat{\mathtt{R}}_\mathtt{X} \mathtt{R}_{\mathtt{B}_i} \hat{\mathtt{R}}_\mathtt{X}^\top, \; \hat{\mathbf{t}}_{\mathtt{A}_i} = \hat{\mathtt{R}}_\mathtt{X}\big((\mathtt{R}_{\mathtt{B}_i} - \mathtt{I})\mathbf{t}'_\mathtt{X} + \mathbf{t}_{\mathtt{B}_i}\big).$$

5.1 Feasibility Test Formulation

Problem 2 can be formulated as a feasibility test.

Problem 3.

$$\begin{array}{rl} \text{Given} & D_\sigma, \epsilon_{\min} \\ \text{do there exist} & \mathtt{R}_\mathtt{X} \in D_\sigma, \mathbf{t}'_\mathtt{X} \\ \text{such that} & \angle(\pm[\mathbf{v}_{ij}]_\times \mathtt{R}_{\mathtt{A}_i} \mathbf{u}_{ij}, \mathbf{t}_{\mathtt{A}_i}) \leq \frac{\pi}{2} + \epsilon_{\min} \\ \text{for} & i = 1, \ldots, n, \; j = 1, \ldots, m \; ? \end{array}$$

As a non-convex problem—its feasible set is an intersection of non-convex ϵ-epipolar constraints—Problem 3 is hard to solve. In order to do so, we start by bringing down the number of variables. We formulate a relaxation of Problem 3 where the rotation is fixed. Let $\bar{\mathtt{R}}_\mathtt{X}$ be the rotation represented by the center of cube D_σ and let $\boldsymbol{\beta}_i \in \mathbb{B}_\pi$ be such that $\exp[\boldsymbol{\beta}_i]_\times = \mathtt{R}_{\mathtt{B}_i}$.

Problem 4.

$$\begin{array}{rl} \text{Given} & D_\sigma, \epsilon_{\min}, \bar{\mathtt{R}}_\mathtt{X} \\ \text{does there exist} & \mathbf{t}'_\mathtt{X} \\ \text{such that} & \angle(\pm[\mathbf{v}_{ij}]_\times \bar{\mathtt{R}}_{\mathtt{A}_i} \mathbf{u}_{ij}, \bar{\mathbf{t}}_{\mathtt{A}_i}) \leq \frac{\pi}{2} + \epsilon_{\min} + \gamma_{ij} \\ \text{for} & i = 1, \ldots, n, \; j = 1, \ldots, m \\ \text{such that} & \angle(\pm\mathbf{v}_{ij}, \bar{\mathtt{R}}_\mathtt{A} \mathbf{u}_{ij}) \; > \; 2\|\boldsymbol{\beta}_i\| \sin(\sqrt{3}\sigma/2) \; ? \end{array}$$

Lemma 1 describes the relation between Problems 3 and 4 and its proof specifies the bounds γ_{ij}.

Lemma 1. *Relation between Problems 3 and 4.*

1) *If Problem 3 is feasible, so is Problem 4.*
2) *If Problem 3 is infeasible, then D_σ may be split into subdomains $D_{\sigma'}^s$ of sufficiently small half-side length σ' such that Problem 4 is infeasible in every $D_{\sigma'}^s$.*

Note that Problem 4 contains one more set of constraints on the angles $\angle(\pm\mathbf{v}_{ij}, \bar{\mathtt{R}}_\mathtt{A}\mathbf{u}_{ij})$. These are prerequisites of Lemma 9,

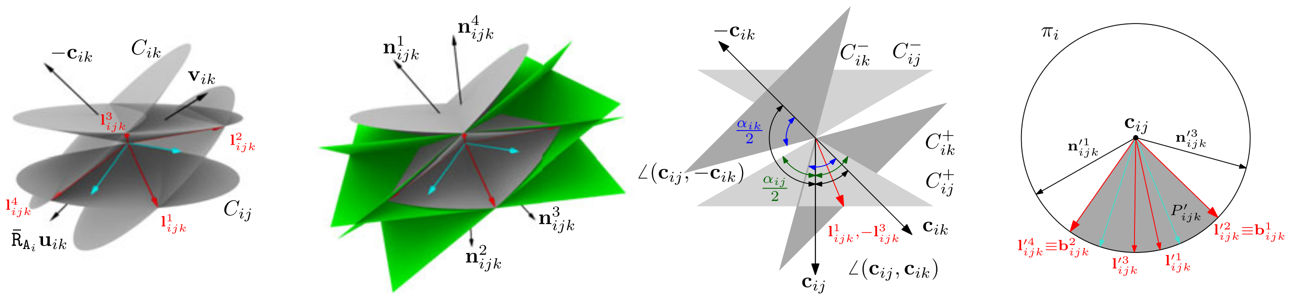


Fig. 3. (a) If the apertures α_j, α_k are sufficiently large, cones C_{ij} and C_{ik} intersect in up to four lines $\mathbf{l}^1_{ijk}, \mathbf{l}^2_{ijk}, \mathbf{l}^3_{ijk}, \mathbf{l}^4_{ijk}$. (b) The linear constraints imposed by correspondences $\mathbf{u}_{ij} \leftrightarrow \mathbf{v}_{ij}$, $\mathbf{u}_{ik} \leftrightarrow \mathbf{v}_{ik}$ are determined by the normals $\mathbf{n}^1_{ijk}, \mathbf{n}^2_{ijk}, \mathbf{n}^3_{ijk}, \mathbf{n}^4_{ijk}$. (c) Since Inequality 2 holds, the nappes C^+_{ij}, C^+_{ik} intersect in $\mathbf{l}^1_{ijk}$ and $-\mathbf{l}^3_{ijk}$. However, Inequality 3 does not hold and the cones C_{ij}, C_{ik} do not form the pyramid P_{ijk}. (d) The projection of the pyramid P_{ijk} into the plane π_i determined the boundary vectors $\mathbf{b}^1_{ijk}, \mathbf{b}^2_{ijk}$ (a 2D projection of the situation in (b)).

see Appendix in the supplementary material[1], which in turn is needed for the proof of Lemma 1. Since this proof is rather technical, it is deferred to Appendix, available in the online supplemental material. Lemma 1 justifies the replacement of the six DOF Problem 3 by the three DOF Problem 4 as the feasibility test for the BnB algorithm. Albeit easier, Problem 4 is still a non-convex problem. In the following section we will formulate another relaxation of Problem 4. This time, however, the relaxation will be convex and easily decidable by LP.

5.2 Intersecting ϵ-Epipolar Constraints

Let once again $D_\sigma \subset \mathbb{B}_\pi$ be a cubic block and $\bar{\mathtt{R}}_\mathtt{X}$ the rotation represented by the center of the block. In Problem 4, a correspondence $\mathbf{u}_{ij} \leftrightarrow \mathbf{v}_{ij}$ imposes an ϵ-epipolar constraint on $\bar{\mathbf{t}}_{\mathtt{A}_i}$ determined by the cone C_{ij} with the axis $\mathbf{c}_{ij} = [\mathbf{v}_{ij}]_\times \bar{\mathtt{R}}_{\mathtt{A}_i}\mathbf{u}_{ij}$ and the aperture $\alpha_{ij} = \pi - 2(\epsilon_{\min} + \gamma_{ij})$. Another correspondence $\mathbf{u}_{ik} \leftrightarrow \mathbf{v}_{ik}$, from the same ith motion, imposes another ϵ-epipolar constraint, this time determined by the cone C_{ik} with the axis $\mathbf{c}_{ik} = [\mathbf{v}_{ik}]_\times \bar{\mathtt{R}}_{\mathtt{A}_i}\mathbf{u}_{ik}$ and the aperture $\alpha_{ik} = \pi - 2(\epsilon_{\min} + \gamma_{ik})$.

Now, let us consider the mutual configuration of the cones C_{ij} and C_{ik}. If the apertures α_{ij}, α_{ik} are sufficiently large, then for $\mathbf{c}_{ij} \neq \mathbf{c}_{ik}$ the two cones intersect in a non-zero vector, see Fig. 3a. Since the cones share the same apex, they intersect in up to four lines—generatrices of the cones—$\mathbf{l}^1_{ijk}, \mathbf{l}^2_{ijk}, \mathbf{l}^3_{ijk}$, and $\mathbf{l}^4_{ijk}$. These lines form the edges of a pyramid P_{ijk} in which every vector satisfying *both* ϵ-epipolar constraints must lie. The pyramid P_{ijk} has four faces lying in four planes, see Fig. 3b. These planes can be determined by their normals $\mathbf{n}^1_{ijk}, \mathbf{n}^2_{ijk}, \mathbf{n}^3_{ijk}$, and $\mathbf{n}^4_{ijk}$ as

$$\mathbf{n}^1_{ijk} = [\mathbf{l}^1_{ijk}]_\times \mathbf{l}^2_{ijk}, \;\; \mathbf{n}^2_{ijk} = [\mathbf{l}^2_{ijk}]_\times \mathbf{l}^3_{ijk},$$
$$\mathbf{n}^3_{ijk} = [\mathbf{l}^3_{ijk}]_\times \mathbf{l}^4_{ijk}, \;\; \mathbf{n}^4_{ijk} = [\mathbf{l}^4_{ijk}]_\times \mathbf{l}^1_{ijk}.$$

It is easy to see that every vector $\bar{\mathbf{t}}_{\mathtt{A}_i}$ that satisfies both ϵ-epipolar constrains, satisfies all of the following linear constraints as well

$$\bar{\mathbf{t}}^\top_{\mathtt{A}_i}\mathbf{n}^1_{ijk} \geq 0,\; \bar{\mathbf{t}}^\top_{\mathtt{A}_i}\mathbf{n}^2_{ijk} \geq 0,\; \bar{\mathbf{t}}^\top_{\mathtt{A}_i}\mathbf{n}^3_{ijk} \geq 0,\; \bar{\mathbf{t}}^\top_{\mathtt{A}_i}\mathbf{n}^4_{ijk} \geq 0. \quad (1)$$

Note that these constraints are also linear in $\mathbf{t}'_\mathtt{X}$.

On the other hand, if the apertures α_{ij}, α_{ik} are too small, cones C_{ij} and C_{ik} don't necessarily have to intersect in a non-zero vector and the ϵ-epipolar constraints cannot be replaced by the linear ones. Fig. 3c will help us to determine when exactly such a situation arises. Let $C^+_{ij}, C^-_{ij}, C^+_{ik}, C^-_{ik}$ be the nappes of the cones C_{ij}, C_{ik} determined by their cone axes $\mathbf{c}_{ij}, -\mathbf{c}_{ij}, \mathbf{c}_{ik}, -\mathbf{c}_{ik}$ respectively. The nappes C^+_{ij}, C^+_{ik} intersect in two line segments $\mathbf{l}^1_{ijk}$ and $-\mathbf{l}^3_{ijk}$ iff

$$\angle(\mathbf{c}_{ij}, \mathbf{c}_{ik}) < \tfrac{\alpha_{ij}+\alpha_{ik}}{2}. \quad (2)$$

In case of equality of the terms in Equation (2) the nappes are tangential and share only one common generatrix. Analogously, the line segments $\mathbf{l}^2_{ijk}$ and $-\mathbf{l}^4_{ijk}$ are the intersections of the nappes C^+_{ij}, C^-_{ik} iff

$$\angle(\mathbf{c}_{ij}, -\mathbf{c}_{ik}) < \tfrac{\alpha_{ij}+\alpha_{ik}}{2}. \quad (3)$$

Inequalities 2 and 3 are thus necessary and sufficient conditions for C_{ij}, C_{ik} to form the pyramid P_{ijk}.

Note that the intersection of cones C_{ij} and C_{ik} determines not only the pyramid P_{ijk}, but also the pyramid P'_{ijk} symmetrical to P_{ijk}, with the origin at the point of symmetry. However, since we assumed that the correspondences satisfy the cheirality condition, only constraints relevant to one of the pyramids are applicable. In the following we will assume, without loss of generality, that P_{ijk} forms the applicable constraints. Formulas for lines $\mathbf{l}^1_{ijk}, \mathbf{l}^2_{ijk}, \mathbf{l}^3_{ijk}, \mathbf{l}^4_{ijk}$ can be obtained by using elementary algebra.

Now, we can formulate a linear relaxation of Problem 4, which is a non-convex relaxation of Problem 3.

Problem 5.

$$\begin{array}{rl} \text{Given} & D_\sigma, \epsilon_{\min}, \bar{\mathtt{R}}_\mathtt{X} \\ \text{does there exist} & \mathbf{t}'_\mathtt{X} \\ \text{such that} & \bar{\mathbf{t}}^\top_{\mathtt{A}_i}\mathbf{n}^1_{ijk} \geq 0,\; \bar{\mathbf{t}}^\top_{\mathtt{A}_i}\mathbf{n}^2_{ijk} \geq 0 \\ & \bar{\mathbf{t}}^\top_{\mathtt{A}_i}\mathbf{n}^3_{ijk} \geq 0,\; \bar{\mathbf{t}}^\top_{\mathtt{A}_i}\mathbf{n}^4_{ijk} \geq 0 \\ \text{for} & i = 1,\ldots,n,\;\; j,k = 1,\ldots,m \\ \text{such that} & \angle(\pm\mathbf{v}_{ij}, \bar{\mathtt{R}}_\mathtt{A}\mathbf{u}_{ij}) > 2\|\boldsymbol{\beta}_i\| \sin(\sqrt{3}\sigma/2) \\ \text{and} & \text{Inequalities 2 and 3 hold?} \end{array}$$

Since Problem 5 is a relaxation of Problem 4, an analogy to Lemma 1 could be formulated. A proof of such a lemma would trivially follow from the geometrical construction of the linear constraints as the convex hull of the intersections of the ϵ-epipolar constraints.

5.3 Selecting ϵ-Epipolar Constraints

It is easy to see that a naive implementation of Problem 5 would lead to prohibitively large linear programs for problems with many correspondences. In this section we show how to reduce the number of linear constraints by selecting only a few of the available correspondences based on their relative positions.

Let us consider the ith motion for a cubic block $D_\sigma \subset \mathbb{B}_\pi$. First, let $\mathbf{C}_i$ be the list of indices of correspondences satisfying the assumptions of Lemma 9. Now, let us assume, without loss of generality, that $\mathbf{u}_{ij} \leftrightarrow \mathbf{v}_{ij}, j \in \mathbf{C}_i$ is the correspondence that forms the ϵ-epipolar constraint with the widest aperture α_{ij}. In the rest of this

1. Which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2015.2469299.

section, we will call the jth constraint the *base constraint*. Next, let's throw away the indices k from $\mathbf{C}_i$ that stand for correspondences $\mathbf{u}_{ik} \leftrightarrow \mathbf{v}_{ik}$ which do not intersect with the base constraint, i.e., indices for which Inequalities 2 and 3 do not hold. Further, let us order the list $\mathbf{C}_i$ according to the distance of the cones axes $|\mathbf{c}_{ij}^\top \mathbf{c}_{ik}|, k \in \mathbf{C}_i$ in the ascending order. The idea is to test the correspondences in $\mathbf{C}_i$ that are "more perpendicular" to the base constraint first, since they are more likely to form a tighter pyramid P_{ijk} and thus tighter linear bounds. Also, since the smaller the value $|\mathbf{c}_{ij}^\top \mathbf{c}_{ik}|$ gets, the more loose and skewed pyramid P_{ijk} becomes—it is reasonable to test only first $s \ll |\mathbf{C}_i|$ correspondences. In our experiments we set $s = 20$. We can think about this process as selecting correspondences whose epipolar lines are the most perpendicular to the epipolar line determined by the base constraint.

Algorithm 1. SelectEpipolarConstraints (Section 5.3)

Require: $i, \mathbf{C}_i$
$\mathit{feasible} \leftarrow 1, s \leftarrow 0, \mathbf{L}_i, \mathbf{b}_i^1, \mathbf{b}_i^2 \leftarrow \emptyset, j \leftarrow \mathrm{argmax}_k \alpha_{ik}$
$\mathbf{C}_i \leftarrow \mathbf{C}_i \backslash \{j\}$ sorted s.t. $\forall k < |\mathbf{C}_i| : |\mathbf{c}_{ij}^\top \mathbf{c}_{ik}| \leq |\mathbf{c}_{ij}^\top \mathbf{c}_{ik+1}|$
while $(|\mathbf{C}_i| > 0) \wedge (s < \texttt{MAX\_CONSTRAINTS})$ **do**
$k \leftarrow \mathrm{PopFront}(\mathbf{C}_i)$
if $\angle(\mathbf{c}_{ij}, \pm\mathbf{c}_{ik}) < (\alpha_{ij} + \alpha_{ik})/2$ **then**
$(\mathbf{b}_{ijk}^1, \mathbf{b}_{ijk}^2) \leftarrow$
$\mathrm{GetBounds}(\mathbf{l}_{ijk}^1, \mathbf{l}_{ijk}^2, \mathbf{l}_{ijk}^3, \mathbf{l}_{ijk}^4, \mathbf{n}_{ijk}^1, \mathbf{n}_{ijk}^3)$
if $\mathbf{b}_i^1, \mathbf{b}_i^2 = \emptyset$ **then**
$(\mathbf{b}_i^1, \mathbf{b}_i^2) \leftarrow (\mathbf{b}_{ijk}^1, \mathbf{b}_{ijk}^2)$
$\mathbf{L}_i \leftarrow \mathbf{L}_i \cup \{[i, \mathbf{n}_{ijk}^1], [i, \mathbf{n}_{ijk}^2], [i, \mathbf{n}_{ijk}^3], [i, \mathbf{n}_{ijk}^4]\}$
else
$(\mathbf{b}_i^1, \mathbf{b}_i^2) \leftarrow \mathrm{IntersectBounds}(\mathbf{b}_i^1, \mathbf{b}_i^2, \mathbf{b}_{ijk}^1, \mathbf{b}_{ijk}^2)$
if $\mathbf{b}_i^1, \mathbf{b}_i^2 = \emptyset$ **then**
$\mathit{feasible} \leftarrow 0$, **return** $[\mathit{feasible}, \mathbf{L}_i]$
else
$s \leftarrow s + 1, \ \mathbf{L}_i \leftarrow \mathbf{L}_i \cup \{[i, \mathbf{n}_{ijk}^1], [i, \mathbf{n}_{ijk}^3]\}$
end if
end if
end if
end while
return $[\mathit{feasible}, \mathbf{L}_i]$

The second reason for intersecting all ϵ-epipolar constraints with the base constraints is that by projecting pyramids P_{ijk} into the plane π_i, defined by vectors $\bar{\mathtt{R}}_{\mathtt{A}_i}\mathbf{u}_{ij}$ and $\mathbf{v}_{ij}$, a simple test based on 2D feasibility of $\mathbf{t}_{\mathtt{A}_i}$ can be constructed. Let $\mathbf{l}'^1_{ijk}, \mathbf{l}'^2_{ijk}, \mathbf{l}'^3_{ijk}, \mathbf{l}'^4_{ijk} \in \mathbb{R}^2$ be normalized 2D projections of the edges of the pyramid P_{ijk} into the plane π_i, see Fig. 3d. It is a trivial observation that the projection of the pyramid P'_{ijk} will be bounded by two vectors $\mathbf{b}_{ijk}^1, \mathbf{b}_{ijk}^2$, which will coincide with two of the projected edges. To decide which edges will form the boundaries, projections $\mathbf{n}'^1_{ijk}, \mathbf{n}'^3_{ijk} \in \mathbb{R}^2$ of the pyramid's faces need to be considered as well. Since the face $\mathbf{n}_{ijk}^1$ is defined by edges $\mathbf{l}_{ijk}^1, \mathbf{l}_{ijk}^2$, one of the projections $\mathbf{l}'^1_{ijk}, \mathbf{l}'^2_{ijk}$ that forms the largest angle with the projection $\mathbf{n}'^1_{ijk}$ will form the boundary. In the situation shown in Fig. 3d, $\mathbf{l}'^{1\top}_{ijk}\mathbf{n}'^1_{ijk} > \mathbf{l}'^{2\top}_{ijk}\mathbf{n}'^1_{ijk}$ and so the $\mathbf{b}_{ijk}^1$ coincides with $\mathbf{l}'^2_{ijk}$. Analogously, we can use $\mathbf{n}'^3_{ijk}$ to decide that $\mathbf{b}_{ijk}^2$ coincides with $\mathbf{l}'^4_{ijk}$. Faces $\mathbf{n}_{ijk}^2$ and $\mathbf{n}_{ijk}^4$ cannot be used in this manner, since their projections to π_i are equally distant from the projection of the respective edges forming them. Since P_{ijk} forms constraints on the position of $\mathbf{t}_{\mathtt{A}_i}$, it is easy to see that P'_{ijk} forms constraints on the projection $\mathbf{t}'_{\mathtt{A}_i}$. From this it follows that the intersection of all $P'_{ijk}, k \in \mathbf{C}_i$ must not be empty for block D_σ to be feasible. The implication in the opposite direction does not hold (two projections $P'_{ijk}, P'_{ij\ell}, \ k, \ell \in \mathbf{C}_i$ can intersect, even though P_{ijk} and $P_{ij\ell}$ do not), so the block can still be infeasible even if the intersection is not empty. However, this simple pre-test can decide infeasibility for *ca.* 30 percent of infeasible cubes. Practically, the correspondences are processed sequentially in the order given by $\mathbf{C}_i$ and the "running intersection" of projections is kept as $\mathbf{b}_i^1, \mathbf{b}_i^2$, compared, and updated with every upcoming projection, see Algorithm 1.

6 The Hand-Eye Calibration Algorithm

This section sums up the BnB algorithm for hand-eye calibration in a more comprehensible pseudo-code form.

First, let us review the hand-eye feasibility test, see Algorithm 2. For every cubic block D_σ the algorithm solves Problem 5. Due to the quite strict assumptions—Lemma 9 and Inequalities 2, 3—not all correspondences produce linear constraints. Indeed, for a large block there might be no feasible correspondences at all. Feasibility of such a block cannot be decided by Problem 5 and it has to be declared feasible by default. However, the smaller the blocks get, the more correspondences can produce linear constraints and Problem 5 is more likely to be decidable.

Algorithm 2. FeasibilityTest

Require: $D, \epsilon_{\min} > 0$
$\bar{\mathtt{R}}_{\mathtt{X}} \leftarrow$ rotation represented by the center of cube D
$\sigma \leftarrow$ half-side length of D
$\mathbf{L} \leftarrow \emptyset$
for $i = 1$ **to** number of motions **do**
// Collect feasible correspondences s.t. Lemma 9
$\mathbf{C}_i \leftarrow \emptyset$
for $j = 1$ **to** number of correspondences **do**
if $\angle(\pm\mathbf{v}_{ij}, \bar{\mathtt{R}}_{\mathtt{A}_i}\mathbf{u}_{ij}) > 2\|\boldsymbol{\beta}_i\| \sin(\sqrt{3}\sigma/2)$ **then**
$\mathbf{C}_i \leftarrow \mathbf{C}_i \cup \{j\}$
end if
end for
$[\mathit{feasible}, \mathbf{L}_i] \leftarrow \mathrm{SelectEpipolarConstraints}(i, \mathbf{C}_i)$
if not $\mathit{feasible}$ **then return** 0
else $\mathbf{L} \leftarrow \mathbf{L} \cup \mathbf{L}_i$ **end if**
end for
if $\mathbf{L} = \emptyset$ **then**
$\mathit{feasible} \leftarrow 1$, **return** $\mathit{feasible}$
end if
// Solve Problem 5
$\mathbf{t}'_{\mathtt{X}} \leftarrow \mathbf{t}'_{\mathtt{X}}$ such that $\forall [i, \mathbf{n}] \in \mathbf{L} : \bar{\mathbf{t}}_{\mathtt{A}_i}^\top \mathbf{n} \geq 0$
if such a $\mathbf{t}'_{\mathtt{X}}$ does not exist **then**
$\mathit{feasible} \leftarrow 0$, **return** $\mathit{feasible}$
else
$\mathit{feasible} \leftarrow 1, \epsilon \leftarrow \max_{i,j} |\angle([\mathbf{v}_{ij}]_\times \bar{\mathtt{R}}_{\mathtt{A}_i}\mathbf{u}_{ij}, \bar{\mathbf{t}}_{\mathtt{A}_i}) - \frac{\pi}{2}|$
end if
// Solve Problem 6
$\tilde{\mathbf{t}}'_{\mathtt{X}} \leftarrow \min_{\mathbf{t}'_{\mathtt{X}}} \sum_{i,j} \left(\angle([\mathbf{v}_{ij}]_\times \bar{\mathtt{R}}_{\mathtt{A}_i}\mathbf{u}_{ij}, \bar{\mathbf{t}}_{\mathtt{A}_i}) - \frac{\pi}{2}\right)^2$
$\tilde{\epsilon} \leftarrow \max_{i,j} |\angle([\mathbf{v}_{ij}]_\times \bar{\mathtt{R}}_{\mathtt{A}_i}\mathbf{u}_{ij}, \bar{\mathbf{t}}_{\mathtt{A}_i}(\tilde{\mathbf{t}}'_{\mathtt{X}})) - \frac{\pi}{2}|$
if $\epsilon \leftarrow \tilde{\epsilon}$ **then** $\epsilon \leftarrow \tilde{\epsilon}, \mathbf{t}'_{\mathtt{X}} \leftarrow \tilde{\mathbf{t}}'_{\mathtt{X}}$ **end if**
return $[\mathit{feasible}, \epsilon, \bar{\mathtt{R}}_{\mathtt{X}}, \mathbf{t}'_{\mathtt{X}}]$

Further, because Problem 5 is a feasibility problem, an LP solver will generally provide a basic feasible solution that does not minimize the nonlinear objective function $\|\mathbf{e}\|_\infty$. In order to speed up the convergence of the algorithm, we use a feasible solution to Problem 5 as an initial estimate for the following non-linear problem:

Problem 6.

$$\begin{aligned} \text{Given} \quad & \bar{\mathtt{R}}_{\mathtt{X}}, \text{ the initial estimate } \mathbf{t}'_{\mathtt{X}} \\ \text{minimize} \quad & \|\mathbf{e}\|_2^2 = \textstyle\sum_{i,j} (\angle([\mathbf{v}_{ij}]_\times \bar{\mathtt{R}}_{\mathtt{A}_i} \mathbf{u}_{ij}, \bar{\mathbf{t}}_{\mathtt{A}_i}) - \frac{\pi}{2})^2 \\ \text{for} \quad & i = 1, \ldots, n, \; j = 1, \ldots, m. \end{aligned}$$

Since Problem 6 minimizes $\|\mathbf{e}\|_2$, it does not necessarily have to provide a solution with better $\|\mathbf{e}\|_\infty$. However, we observed that it helps to accelerate the overall convergence.

Finally, let us describe the BnB part of the algorithm, see Algorithm 3. The algorithm is initialized by the cubic block $D_\pi = \langle -\pi, \pi \rangle^3$. Such a block is redundant since $\mathbb{B}_\pi \subsetneq \langle -\pi, \pi \rangle^3$, but this fact does not impair the algorithm, since blocks $D_\sigma \cap \mathbb{B}_\pi = \emptyset$ can be easily detected and thrown away. Feasible blocks are divided into 8 cubic subblocks and stored in a queue $\mathbf{Q}$, making the algorithm breadth-first search. The search is terminated when the size of the blocks σ reaches a sufficiently small size $\sigma_{\min}$.

Algorithm 3. Branch and Bound

Require: initial estimate of $\epsilon_{\min}$, stopping crit. $\sigma_{\min}$
  $D_\pi \leftarrow \langle -\pi, \pi \rangle^3, \sigma \leftarrow 2\pi$
  PushBack$(\mathbf{Q}, D_\pi)$
  while $\sigma > \sigma_{\min}$ **do**
    $D \leftarrow$ PopFront$(\mathbf{Q})$, $\sigma \leftarrow$ half-side length of D
    $\{feasible, \epsilon, \bar{\mathtt{R}}_{\mathtt{X}}, \mathbf{t}'_{\mathtt{X}}\} \leftarrow$ FeasibilityTest$(D, \epsilon_{\min})$
    if $feasible \equiv$ **true then**
      if $\epsilon < \epsilon_{\min}$ **then** $\hat{\mathtt{R}}_{\mathtt{X}} \leftarrow \bar{\mathtt{R}}_{\mathtt{X}}, \hat{\mathbf{t}}'_{\mathtt{X}} \leftarrow \mathbf{t}'_{\mathtt{X}}, \epsilon_{\min} \leftarrow \epsilon$ **end if**
      PushBack$(\mathbf{Q}$, SubdivideBlock$(D))$
    end if
  end while
  $\hat{\mathbf{t}}_{\mathtt{X}} \leftarrow -\hat{\mathtt{R}}_{\mathtt{X}} \hat{\mathbf{t}}'_{\mathtt{X}}$
  return $\{\hat{\mathtt{R}}_{\mathtt{X}}, \hat{\mathbf{t}}_{\mathtt{X}}, \epsilon_{\min}\}$

The proposed algorithm is easily parallelizable by running the feasibility test in multiple threads consuming a mutual queue $\mathbf{Q}$. Note that the parallel processing of the cubes can theoretically result in accepting some cubes that would be rejected in the single thread mode, making $\mathbf{Q}$ larger. However, the experiments show that this is not a practical issue and that the computation time reduction is in practice linear in the number of threads used.

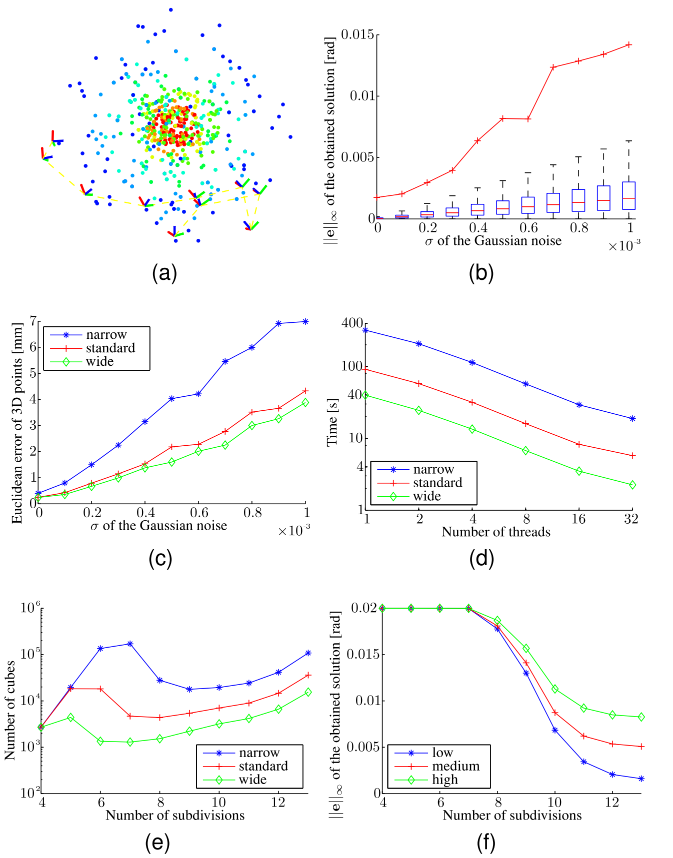


Fig. 4. *Ball experiment.* (a) Camera positions in the experiment, different colors encode positions of the 3D points at different FOV levels. (b) The maximum residual error of the obtained solutions for the various values of σ (red line) and the distribution of the measured errors over all correspondences (boxes). (c) The mean Euclidean distance between the 3D points transformed to the gripper's coordinate systems using ground truth $\mathtt{X}$ and the 3D points transformed to the gripper's coordinate systems using the estimated $\mathtt{X}$. The eight FOV levels were merged into three groups. (d) Loglog plot of the computational time as a function of the number of threads. (e) The mean number of remaining cubes plotted against the number of subdivision phases. Note that the computation starts after the fourth subdivision. (f) The mean residual error at the beginning of the respective subdivision phase. Different noise levels were clustered into three groups.

7 Experimental Results

We evaluated the performance of the proposed algorithm using both synthetically generated and real world data measurements. The values of the initial estimate $\epsilon_{\min}$ and the stopping criterion $2\sigma_{\min}$ were set to 0.02 rad and 0.001 rad respectively. We use GLPK [1] to solve Problem 5 and levmar [10] to solve nonlinear Problem 6. All the reported times were achieved using a C++ implementation on a 32×2.6 GHz AMD Opteron based computer running 64-bit Linux. The source code is available at `http://cmp.felk.cvut.cz/~hellej1/bbhec/`.

7.1 Experiments with Synthetic Data

Ball experiment. A synthetic scene consisting of 100 3D points was generated into a ball of radius 1,000 mm. 10 absolute camera poses were set up so that (i) the centers of the cameras were outside of the ball but close to its "surface", (ii) the centers were positioned so that the offsets of the camera motions were ~500 mm and (iii) the cameras faced approximately the center of the ball. In order to simulate the effect of decreasing field of view (FOV), seven progressively smaller balls were generated inside the initial ball so that the balls shared the centers and the volume of the newly created ball was half the volume of the previous ball. This defined 8 FOV levels, namely 119, 85, 66, 52, 40, 31, 24, and 19 degree. Additional 3D points were generated inside each of the smaller balls in order to have exactly 100 3D points at each FOV level measured in the respective cameras giving correspondences $\mathbf{u}_{ij} \leftrightarrow \mathbf{v}_{ij}$.

Further, fifty random transformations $\mathtt{X}$ were generated and the optimization tasks composed of the known correspondences $\mathbf{u}_{ij} \leftrightarrow \mathbf{v}_{ij}$ and nine motions $\mathtt{B}_i$—computed from the known absolute camera poses and the generated $\mathtt{X}$—were constructed for each of the eight FOV levels. Finally, the correspondences were corrupted with Gaussian noise in the angular domain using 11 noise levels, $\sigma \in \langle 0, 10^{-3} \rangle$ in 10^{-4} steps resulting to 88 tasks per transformation.

Fig. 4a shows the setup of the experiment, Figs. 4b, 4c, 4d, 4e, 4f show the results. Notice that the median of the measured errors over all correspondences—denoted by red horizontal lines in Fig. 4b—is approximately one order of magnitude smaller than the maximum residual error. The actual errors of the calibration, i.e., the Euclidean distances between the 3D points transformed to the gripper coordinate systems using the ground truth $\mathtt{X}$ and the 3D points transformed to the gripper coordinate systems using the estimated $\mathtt{X}$, are higher for narrow FOVs. The reason is that the wide FOVs situations contain correspondences that produce tighter linear bounds and thus result in higher accuracy. Considering the computational times, the solutions are found faster for wide FOVs as the correspondences of narrow FOVs camera pairs

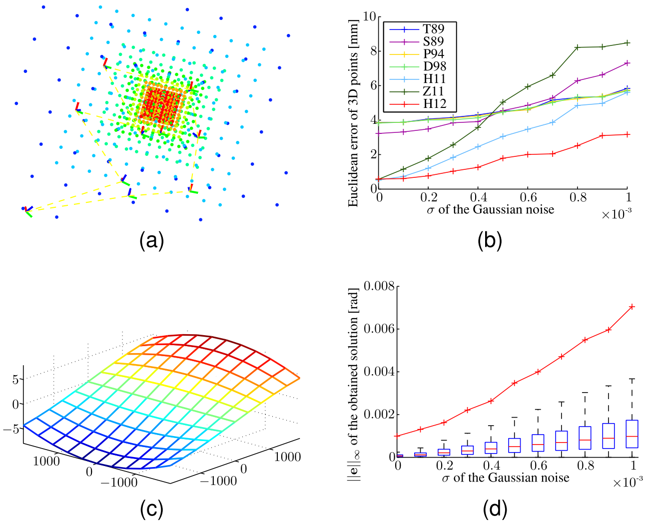


Fig. 5. *Planar experiment.* (a) Camera positions in the experiment, different colors encode positions of the 3D points at different FOV levels. (b) Comparison of the proposed method with previous methods using Euclidean distance measure. (c) Grid deformation (not in scale). (d) The maximum (red line) and median (boxes) residual error of the obtained solutions for the various values of σ.

do not generate enough linear constraints for large blocks and more blocks need to be subdivided.

Planar experiment. In order to compare our method to the previously proposed ones, we generated a planar calibration device consisting of a rectangular 11×11 grid with known 3D position, see Fig. 5a. The camera poses were set up so that camera centers were ~1,000 mm away from the calibration device and the correspondences were corrupted by the same amount of Gaussian noise as in the Ball experiment. This time we generated twenty random transformations $\mathtt{X}$ for three standard FOV levels 64, 47 and 35 degree (by varying calibration device size). In order to simulate the manufacturing and structural errors introduced while using a calibration device we also corrupted the 3D positions of the grid points with systematic errors. We multiplied the grid by 1.005 along the y-axis and deformed it using the function $\sin x \cos y$ along the z-axis, again, so to differ by no more that 5‰ of the grid's side length from the original, see Fig. 5c. Since most of the previous methods require also the additional knowledge of the external camera poses, these were recovered using EPnP algorithm [9]. Fig. 5b shows the results of the comparison using the same Euclidean distance measure as in the Ball experiment. The labels "T89", "S89", "P94", "D98", "H11", and "Z11" stand for methods [3], [6], [11], [15], [19], and [21], respectively. Label "H12" stands for the proposed method.

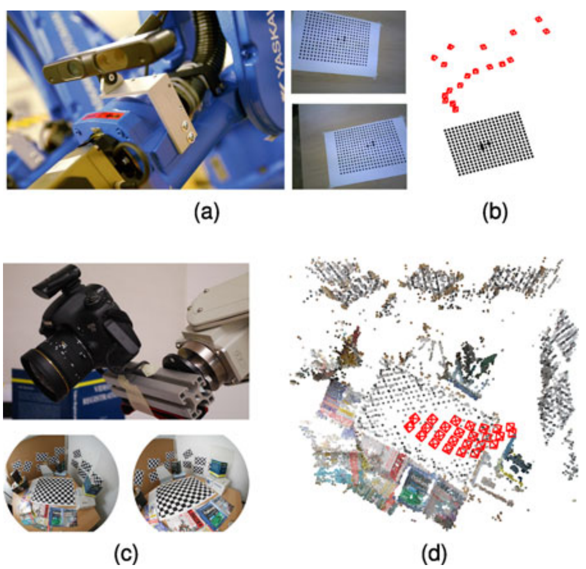


Fig. 6. *Real data experiment.* (a-b) Motoman MA1400. Close up of the camera-gripper rig, sample images from the sequence; Camera poses reconstruction (c-d) Mitsubishi MELFA-RV-6S. Close up of the camera-gripper rig, sample images from the sequence; Model resulting from SfM, cameras are denoted by red pyramids.

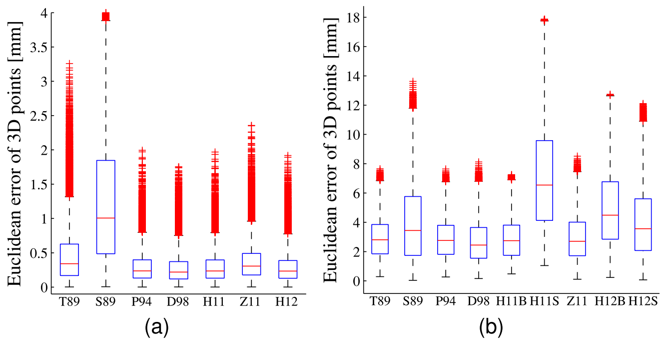


Fig. 7. *Real data experiment error.* (a) Motoman MA1400 (b) Mitsubishi MELFA-RV-6S.

7.2 Experiment with Real Data

Motoman MA1400 experiment. In the first real data experiment, an Asus Xtion Pro sensor was rigidly attached to the fifth link of a Motoman MA1400 serial 6-DOF manipulator, see Fig. 6a. The end-effector was manipulated into 18 poses $\mathtt{B}'_i$ and in each pose a 640×480 picture of a calibration target consisting of 315 distinguishable dots was taken. An optimization task consisting of nine relative movements and totaling 2,835 correspondences was constructed. The algorithm converged to a solution with residual error $\|\mathbf{e}\|_\infty = 0.003$ rad (~ 0.17 degree) in 85 seconds (running in eight threads).

In order to compare the result to the previous methods, absolute camera poses $\mathtt{A}'_i$ were recovered using EPnP algorithm [9], see Fig. 6b. Next, hand-eye transformations $\mathtt{X}_k$, $k = 1, \ldots, 6$ were computed by methods "T89", "S89", "P94", "D98", "H11", and "Z11" using the same relative movements. Fig. 7a shows a distribution of the Euclidean distances between the 3D points of the calibration device C and the 3D points of the calibration device transformed by the relative movements $\mathtt{A}'^{-1}_j \mathtt{X}_k \mathtt{B}'^{-1}_j \mathtt{B}'_i \mathtt{X}^{-1}_k \mathtt{A}'_i$, for $i, j = 1, \ldots, 18$. Notice that this error measure is different from the one used in the synthetic experiment, since here the ground truth transformation $\mathtt{X}$ is not known. Also notice that lower error values do not necessarily mean "better" $\mathtt{X}$, since both camera and robot calibrations can be slightly noisy. In this case, however, medians for all the methods—except for "S89"—are well below 0.5 mm, validating the result by the proposed method labeled as "H12".

Mitsubishi MELFA-RV-6S experiment. A Mitsubishi MELFA-RV-6S serial manipulator with a Canon 7D digital SLR camera and a Sigma 8 mm lens (pixel size ~ 0.063 degree, FOV ~ 130 degree) were used to acquire data for the second real experiment. The robot was instructed to move the gripper along the surface of a sphere of radius ~700 mm centered in the middle of the scene. The position of the gripper was adjusted to reach 25 different locations at four different pitch angles and the gripper was set to face the center of the sphere, see Fig. 6c. The internal calibration of the camera was obtained from several images of a checkerboard using OCamCalib [13]

First, SfM software [17] was used to automatically generate correspondences, see Fig. 6d. We used seven motions $\mathtt{B}_i$ and with 447 randomly selected correspondences to construct the optimization task. The algorithm converged to a solution with residual error $\|\mathbf{e}\|_\infty = 0.006$ rad (~ 0.34 degree) in 8 seconds (eight threads). The result is labeled as "H12S" in Fig. 7b. We used the same optimization task to recover the calibration using the method [6], labeled as "H11S".

Next, we detected the grid pattern in every image. Using the same seven relative movements, a calibration task consisting of 1,155 correspondences was constructed. The algorithm converged to a solution with residual error $\|\mathbf{e}\|_\infty = 0.005$ rad (~ 0.29 degree) in 56 seconds (eight threads). The result is labeled as "H12B" in Fig. 7b. Again, method [6] labeled "H11B" was used with the same task. Finally, since the grid's dimensions were known, the camera poses were recovered in scale using EPnP algorithm and calibration was performed using methods "T89", "S89", "P94", "D98", and "Z11".

We can see, Fig. 7b, that the resulting calibrations are relatively worse than the results in the MA1400 experiment. We explain it by the fact that the MELFA-RV-6S robot was slightly miscalibrated, showing the fact that the proposed method is more sensitive to the robot calibration. This is probably due to the fact that the error cannot be compensated by the known camera positions as in the previous approaches. However, when the robot is properly calibrated, it can deliver comparable or better results than its competitors.

8 Conclusion

In this paper, we removed the requirement for known camera extrinsics from the hand-eye calibration problem. Since the presented algorithm is completely independent on the scene geometry and scale, there is no need for a known calibration device and the calibration can be performed solely from a general scene. Not only this makes the method immune to the errors introduced by calibration device manufacturing process but also increases the calibration space to virtually arbitrary size. The theory shows that the algorithm is guaranteed to be globally optimal with respect to L_∞-norm. Further, the experiments show that the algorithm is also practically useful, since it is highly parallelizable and competitive in situations where a calibration device is lacking in precision or is impractical due to the calibration space requirements.

Acknowledgments

The authors were supported by the EC FP7-SPACE-2012-312377 PRoViDE, by the EC FP7-288553 CloPeMa, and by the Grant Agency of the CTU Prague SGS10/277/OHK3/3T/13 projects.

References

[1] GNU linear programming kit version 4.47, 2011.

[2] J. C. K. Chou and M. Kamel, "Finding the position and orientation of a sensor on a robot manipulator using quaternions," *Int. J. Robot. Res.*, vol. 10, no. 3, pp. 240–254, 1991.

[3] K. Daniilidis, "Hand-eye calibration using dual quaternions," *Int. J. Robot. Res.*, vol. 18, pp. 286–298, 1998.

[4] R. Hartley and F. Kahl, "Global optimization through rotation space search," *Int. J. Comput. Vis.*, vol. 82, no. 1, pp. 64–79, 2009.

[5] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[6] J. Heller, M. Havlena, A. Sugimoto, and T. Pajdla, "Structure-from-motion based hand-eye calibration using L_∞ minimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 3497–3503.

[7] J. Heller, M. Havlena, and T. Pajdla, "A branch-and-bound algorithm for globally optimal hand-eye calibration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 1608–1615.

[8] R. Horaud and F. Dornaika, "Hand-eye calibration," *Int. J. Robot. Res.*, vol. 14, no. 3, pp. 195–210, 1995.

[9] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, pp. 155–166, 2009.

[10] M. I. A. Lourakis, "levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++," Jul. 2004, http://www.ics.forth.gr/~lourakis/levmar/

[11] F. C. Park and B. J. Martin, "Robot sensor calibration: Solving AX=XB on the Euclidean group," *IEEE Trans. Robot. Automation*, vol. 10, no. 5, pp. 717–721, Oct. 1994.

[12] T. Ruland, T. Pajdla, and L. Kruger, "Globally optimal hand-eye calibration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 1035–1042.

[13] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *Proc. Int. Conf. Intell. Robots Syst.*, 2006, pp. 5695–5701.

[14] Y. Seo, Y.-J. Choi, and S. W. Lee, "A branch-and-bound algorithm for globally optimal calibration of a camera-and-rotation-sensor system," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 1173–1178.

[15] Y. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB," *IEEE Trans. Robot. Automation*, vol. 5, no. 1, pp. 16–29, Feb. 1989.

[16] H. Stewenius and K. Aström, "Hand-eye calibration using multilinear constraints," in *Proc. Asian Conf. Comput. Vis.*, 2004, pp. 760–763.

[17] A. Torii, M. Havlena, and T. Pajdla, "Omnidirectional image stabilization for visual object recognition," *Int. J. Comput. Vis.*, vol. 91, no. 2, pp. 157–174, 2011.

[18] R. Tsai and R. Lenz, "Real time versatile robotics hand/eye calibration using 3d machine vision," in *Proc. Int. Conf. Robot. Automation*, 1988, vol. 1, pp. 554–561.

[19] R. Tsai and R. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," *IEEE Trans. Robot. Automation*, vol. 5, no. 3, pp. 345–358, Jul. 1989.

[20] H. Zhang, "Hand/eye calibration for electronic assembly robots," *IEEE Trans. Robot. Autom.*, vol. 14, no. 4, pp. 612–616, Aug. 1998.

[21] Z. Zhao, "Hand/eye calibration using convex optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 2947–2952.