A Branch-and-Bound Algorithm for Globally Optimal Calibration of a Camera-and-Rotation-Sensor System [*]

Yongduek Seo, Young-Ju Choi, Sang Wook Lee
[yndk,wildpink,slee]@sogang.ac.kr
Department of Media Technology, Sogang University, Seoul, Korea

Abstract

We propose a branch-and-bound algorithm to obtain the globally optimal relative rotation between a camera and the rotation sensor attached to it. Compared to previous methods, our approach directly minimizes the image space error related to the measurements which is very natural for camera-based systems. Our algorithm is based on the observation that we may evaluate the residual when the rotation matrix is known. We propose a feasibility test algorithm for the branch-and-bound to efficiently reduce the search volume of the rotation domain. Experimental results are provided using synthetic and real data sets.

1. Introduction

This paper deals with the problem of optimally estimating the relative rotation between a camera and a rotation sensor attached to the camera. A goal of this research is to make use of a rotation sensor together with a camera and thus employ the well-developed L_∞ approaches.

Recent study on the L_∞ norm minimization method has opened a new way of geometric computation. In short, we may obtain *globally optimal* solutions to various geometric vision problems [3, 7, 8]. One major limitation of the L_∞ method has been the assumption of known rotation among the parameters to be estimated. For example, we need to know the rotation of the camera to obtain the global solution of the camera pose problem. This limitation is overcome by Hartley and Kahl [4] now for some of the problems. Their branch-and-bound algorithm provides an efficient method to deal with rotation components, too, through a search over the rotation space. It is now possible to obtain global solutions to the problems of camera pose and the essential matrix.

On the other hand, nowadays it is very common to have an external sensor that provides rotational information. For example, many platforms for robotic applications are equipped with hardware such as IMUs (Inertial Measurement Units) or motor-encoders that provide the rotational motion of the platforms. If we attach a rotation-sensor to our imaging camera, then we do not have to compute the rotation of the camera and it is then very natural to apply the L_∞ methods for geometric vision problems. However, the coordinate axes of the sensor and the camera do not coincide in practice. Therefore, we need to find the (rotational) displacement before using the camera-sensor pair.

Indeed, this belongs to the so-called *hand-eye calibration* problem which has been widely considered for robotic applications [12, 11, 2, 14, 1, 13, 6, 10, 9]. A solution to the displacement X may be achieved from the equation $AX = XB$ where A and B are known rigid motions of the camera and the sensor platform, respectively. In the previously proposed algorithms, the rotational component of X is estimated based on the rotational components of A and B. Translational component is estimated using the rotation and then a nonlinear optimization is done to refine get the whole solution. However, we observe that even though each step of any of the previous algorithms uses an optimal approach it does not guarantee optimality with respect to the re-projection error for the image measurements.

Our goal of this paper is to compute a globally optimal estimate of the relative rotation of the camera and the rotation-sensor. This paper exploits the fact that image correspondences are directly related by homography when the motion of the camera between two views is a pure rotation or can be well approximated as a pure rotation [5]. In this paper, we propose a branch-and-bound algorithm to overcome this non-convex problem in obtaining the rotational displacement based on Hartley and Kahl [4]. The method directly minimizes the residuals in the image space and provides a globally optimal estimate in the L_∞ sense.

Our paper is composed of several sections. Firstly, Section 2 describes the problem formulation in detail and the idea of the branch and bound algorithm. Section 3 derives the feasibility test for the branch-and-bound algo-

[*] This work was supported by the strategic technology development program of MCST/MKE/KEIT. [2008-F-030-01, Development of Full 3D Reconstruction Technology for Broadcasting Communication Fusion]

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

rithm. Then, Section 4 shows how image space error can be used instead of the angular error. The overall branch-and-bound algorithm is explained in Section 5. Section 6 gives some experimental results and practical aspects of the branch-and-bound algorithm. Finally, concluding remarks are given in Section 7.

2. Problem Formulation

As a camera-and-rotation-sensor system, we consider a calibrated camera equipped with a rotation sensor. We assume that the internal parameters such as focal length of the camera are known. In this case, the internal calibration matrix is the identity and we may consider image points as lying on a unit sphere rather than an image plane. An image measurement is therefore a unit vector $\mathbf{v}$ representing the direction from the camera center to its 3D point.

Some notations: $\mathtt{A}, \mathtt{B}, \mathtt{R}$, and $\mathtt{X}$ denote 3×3 rotation matrices in $\mathrm{SO}(3)$; $\mathbf{v}_i, \mathbf{u}_i \in \mathbb{R}^3$, $\|\mathbf{v}_i\| = \|\mathbf{u}_i\| = 1$, are, as the i-th correspondence, from the two images whose relative rotation of the sensor is $\mathtt{B}$; the norm $\|\cdot\|$ represents the L_2 or Euclidean norm; $B_\pi \subset \mathbb{R}^3$ is the ball of radius π, denoting the angle-axis parameterization of $\mathrm{SO}(3)$; $D(\sigma) \subset \mathtt{B}_\pi$ is a cube whose side length is 2σ; finally, $\boldsymbol{\beta}, \boldsymbol{\omega} \in B_\pi$ denote points in B_π.

A rotational motion $\mathtt{A} \in \mathrm{SO}(3)$ of the camera moves the measurement vector $\mathbf{v}_n$ to $\mathbf{u}_n$

$$\mathbf{u}_n = \mathtt{A}\mathbf{v}_n, \quad n = 1, ..., N, \tag{1}$$

where N is the number of matching pairs. Let $\mathtt{X} \in \mathrm{SO}(3)$ be the rotational displacement between the camera and the sensor; let $\mathtt{B}$ be the rotation of the sensor that induces the camera motion $\mathtt{A}$. Then, we have the well-known equation

$$\mathtt{A} = \mathtt{X}\mathtt{B}\mathtt{X}^\top \quad \text{or} \quad \mathtt{A}\mathtt{X} = \mathtt{X}\mathtt{B}. \tag{2}$$

Let us suppose that K pairs of images are given, and each pair of images have N point correspondences. The error e_{nk} between $\mathbf{u}_{nk}$ and $\mathtt{X}\mathtt{B}_k\mathtt{X}^\top\mathbf{v}_{nk}$ is defined to be the angle between the two vectors:

$$e_{nk} = \angle(\mathbf{u}_{nk}, \mathtt{X}\mathtt{B}_k\mathtt{X}^\top\mathbf{v}_{nk}), \tag{3}$$

where $\mathtt{B}_k$ is the sensor rotation between the two view of the pair. This paper minimizes the L_∞ norm of the errors e_{nk}.

$$\min_{\mathtt{X}} \max_{nk} \angle(\mathbf{u}_{nk}, \mathtt{X}\mathtt{B}_k\mathtt{X}^\top\mathbf{v}_{nk}) \ . \tag{4}$$

Since the minimization in (4) is performed over the rotation matrix $\mathtt{X}$, we could find the optimal rotation $\mathtt{X}$ if the error was examined for every $\mathtt{X}$. But, it is practically impossible because there are infinitely many rotation matrices. An alternative could be an algorithm like Levenberg-Marquardt. But it would converge to a local minimum depending on the initiation of the algorithm because of the non-linearity of the rotation matrix itself as well as the error function.

Our strategy, given below, is to obtain the global optimum through a branch-and-bound algorithm.

1. We choose a rotation $\mathtt{X}^* = \mathtt{X}_0$ and compute the cost $\epsilon_{\min} = \max_{nk} \angle(\mathbf{u}_{nk}, \mathtt{X}^*\mathtt{B}_k\mathtt{X}^{*\top}\mathbf{v}_{nk})$.

2. The rotation space is divided into cubic blocks $\{D_j\}$ whose half-side length is σ; repeat the following two steps.

(bnb-1). For each block D_j, We determine by a *test* whether there is a better rotation in D_j that may provide an error less than $\epsilon_{\min}$. If the answer is no, then D_j is excluded.

(bnb-2). Otherwise, we evaluate the cost function and update the cost $\epsilon_{\min}$ and the solution $\mathtt{X}^*$. Then, D_j is subdivided into eight smaller cubic sub-domains ($\sigma \leftarrow \sigma/2$).

Basically, the iteration of this algorithm is terminated when the size of the remaining cubic blocks is sufficiently small. The *test* problem in the third step is called the feasibility problem, which is detailed in the next section.

3. Feasibility Test for Branch-and-Bound

Let us consider a point $\boldsymbol{\beta} \in B_\pi$ representing the rotation whose axis is $\boldsymbol{\beta}/\|\boldsymbol{\beta}\|$ and angle $\|\boldsymbol{\beta}\|$. The rotation matrix $\mathtt{R}$ corresponding to $\boldsymbol{\beta}$ is given by the matrix exponential ($\exp$: $\boldsymbol{\beta} \rightarrow \mathtt{B}$):

$$\mathtt{B} = \exp[\boldsymbol{\beta}]_\times = \mathtt{I} + \frac{\sin\|\boldsymbol{\beta}\|}{\|\boldsymbol{\beta}\|}[\boldsymbol{\beta}]_\times + \frac{1-\cos\|\boldsymbol{\beta}\|}{\|\boldsymbol{\beta}\|}[\boldsymbol{\beta}]_\times^2 \tag{5}$$

where

$$[\boldsymbol{\beta}]_\times = \begin{bmatrix} 0 & -\beta_3 & \beta_2 \\ \beta_3 & 0 & -\beta_1 \\ -\beta_2 & \beta_1 & 0 \end{bmatrix}. \tag{6}$$

The inverse map is given by the matrix logarithm which gives the 3-vector $\boldsymbol{\beta}$ corresponding to the rotation $\mathtt{R}$.

$$[\boldsymbol{\beta}]_\times = \log \mathtt{B} = \frac{\|\boldsymbol{\beta}\|}{2\sin\|\boldsymbol{\beta}\|}(\mathtt{B} - \mathtt{B}^\top) \tag{7}$$

where $1 + 2\cos\|\boldsymbol{\beta}\| = \mathrm{Trace}(\mathtt{B})$.

When a rotation $\mathtt{B}$ is obtained by taking the exponential of a point $\boldsymbol{\beta} \in D$, we use the notation $\mathtt{B} \in D$ to indicate that $\mathtt{B}$ is the exponential map of a point $\boldsymbol{\beta}$.

Lemma 1 *An identity.*

$$\log(\mathtt{X}\mathtt{B}\mathtt{X}^\top) = \mathtt{X}[\boldsymbol{\beta}]_\times\mathtt{X}^\top = [\mathtt{X}\boldsymbol{\beta}]_\times \ . \tag{8}$$

This is due to the fact that

$$\mathtt{X}\mathtt{B}\mathtt{X}^{-1} = \mathtt{X}\exp([\boldsymbol{\beta}]_\times)\mathtt{X}^{-1} = \exp(\mathtt{X}[\boldsymbol{\beta}]_\times\mathtt{X}^{-1}). \tag{9}$$

Lemma 1 is the theoretical basis of the method by Park and Martin [11] to solve the equation $AX = XB$.

Our branch-and-bound algorithm makes a series of subdivision of the rotation space B_π into small cubes, for which we enclose the ball B_π in a cube $C_\pi = [-\pi, \pi]^3$ as does [4]. Even though this cube provides a redundant representation, it does not matter for our branch-and-bound algorithm.

Now we provide an inequality for the feasibility problem. Let $D(\sigma)$ represent one of the cubes obtained by a subdivision of C_π.

Lemma 2 *Let $\bar{\mathtt{X}}$ be the rotation corresponding to the center of the cube $D(\sigma)$, and $\boldsymbol{\beta}$ be a 3-vector (representing a point in B_π, in particular). For any rotation $\mathtt{X} \in D(\sigma)$, we have the following inequality*

$$\|\mathtt{X}\boldsymbol{\beta} - \bar{\mathtt{X}}\boldsymbol{\beta}\| \leq 2\|\boldsymbol{\beta}\| \sin(\sqrt{3}\sigma/2)\ . \tag{10}$$

Proof. Note that $\|\mathtt{X}\boldsymbol{\beta} - \bar{\mathtt{X}}\boldsymbol{\beta}\| = \|\mathtt{X}(\boldsymbol{\beta} - \mathtt{X}^\top\bar{\mathtt{X}}\boldsymbol{\beta})\| = \|\boldsymbol{\beta} - \mathtt{X}^\top\bar{\mathtt{X}}\boldsymbol{\beta}\|$ because $\mathtt{X}$ is an isometry. The maximum possible rotation angle of $\mathtt{X}^\top\bar{\mathtt{X}}$ is $\sqrt{3}\sigma$ which is the distance from the center of $D(\sigma)$ to the farthest corners. Therefore, the maximum distance between the two vectors $\boldsymbol{\beta}$ and $\mathtt{X}^\top\bar{\mathtt{X}}\boldsymbol{\beta}$ is given by $l_{\max}/2 = \|\boldsymbol{\beta}\| \sin(\sqrt{3}\sigma/2)$. See Fig.1. □

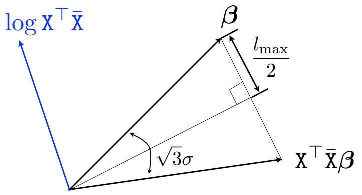
$\log \mathtt{X}^\top\bar{\mathtt{X}}$
$\boldsymbol{\beta}$
$\frac{l_{\max}}{2}$
$\sqrt{3}\sigma$
$\mathtt{X}^\top\bar{\mathtt{X}}\boldsymbol{\beta}$

Figure 1. The maximum possible distance $l_{\max}$ induced by $\mathtt{X}^\top\bar{\mathtt{X}}$.

Let $\mathtt{B}_1$ and $\mathtt{B}_2$ be two rotations. The distance $d_\angle(\mathtt{B}_1, \mathtt{B}_2)$ is the angle θ of the rotation $\mathtt{B}_1^\top\mathtt{B}_2$, lying in the range of $0 \leq \theta \leq \pi$. The following two lemmas are from [4] where their proofs are provided.

Lemma 3 *For any vector* $\mathbf{v}$,

$$\angle(\mathtt{B}_1\mathbf{v}, \mathtt{B}_2\mathbf{v}) \leq d_\angle(\mathtt{B}_1, \mathtt{B}_2). \tag{11}$$

Lemma 4 *Let $\mathtt{B}_i = \exp[\boldsymbol{\beta}_i]_\times$ be the rotation corresponding to the vector $\boldsymbol{\beta}_i \in B_\pi$. Then*

$$d_\angle(\mathtt{B}_1, \mathtt{B}_2) \leq \|\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2\| \tag{12}$$

with equality only when $\boldsymbol{\beta}_1 = \boldsymbol{\beta}_2$.

Now let us consider our feasibility problem for a small domain $D(\sigma)$.

Problem 1 *Do there exist any $\mathtt{X} \in D(\sigma)$ such that*

$$\max_{nk} \angle(\mathbf{u}_{nk}, \mathtt{X}\mathtt{B}_k\mathtt{X}^\top\mathbf{v}_{nk}) \leq \epsilon_{\min}? \tag{13}$$

This asks the possibility of existence of any rotation $\mathtt{X}$ in the domain $D(\sigma)$ that may result in the L_∞ error less than $\epsilon_{\min}$. It is not easy to give an answer to this problem directly. Instead we consider the following problem.

Problem 2 *Is the inequality below valid?*

$$\begin{aligned}\max_{nk} \angle(\mathbf{u}_{nk}, &\bar{\mathtt{X}}\mathtt{B}_k\bar{\mathtt{X}}^\top\mathbf{v}_{nk}) \\ &\leq \epsilon_{\min} + 2\boldsymbol{\beta}_{\max} \sin(\sqrt{3}\sigma/2),\end{aligned} \tag{14}$$

where $\boldsymbol{\beta}_{\max} = \max_k \|\boldsymbol{\beta}_k\|$, $\bar{\mathtt{X}} \in D(\sigma)$ is the rotation at the central point of $D(\sigma)$, σ the half-side length of the cube $D(\sigma)$, and $\boldsymbol{\beta}_k = \log \mathtt{B}_k$.

Lemma 5 *The two feasibility problems (Problem 1 and Problem 2) are related as follows.*
1. If Problem 1 has an affirmative answer then so does Problem 2.
2. If Problem 1 has a negative answer, then D can be split into subdomains D_i of sufficiently small half-side length such that Problem 2 has a negative answer on every D_i.

Proof. The proof for the first part of the lemma is given. Let $\mathtt{X}_{\text{opt}} \in D(\sigma)$ is a feasible solution for Problem 2. Then, we have the following result.

$$\begin{aligned}
&\max_{nk} \angle(\mathbf{u}_{nk},\ \bar{\mathtt{X}}\mathtt{B}_k\bar{\mathtt{X}}^\top\mathbf{v}_{nk}) \\
&\quad\leq \max_{nk} \angle(\mathbf{u}_{nk}, \mathtt{X}_{\text{opt}}\mathtt{B}_k\mathtt{X}_{\text{opt}}^\top\mathbf{v}_{nk}) \\
&\quad\quad + \max_{nk} \angle(\mathtt{X}_{\text{opt}}\mathtt{B}_k\mathtt{X}_{\text{opt}}^\top\mathbf{v}_{nk}, \bar{\mathtt{X}}\mathtt{B}_k\bar{\mathtt{X}}^\top\mathbf{v}_{nk}) && (15) \\
&\quad\leq \epsilon_{\min} + \max_{nk} d_\angle(\mathtt{X}_{\text{opt}}\mathtt{B}_k\mathtt{X}_{\text{opt}}^\top, \bar{\mathtt{X}}\mathtt{B}_k\bar{\mathtt{X}}^\top) && (16) \\
&\quad\leq \epsilon_{\min} + \max_{nk} d_\angle\big(\exp(\mathtt{X}_{\text{opt}}\boldsymbol{\beta}_k), \exp(\bar{\mathtt{X}}\boldsymbol{\beta}_k)\big) && (17) \\
&\quad\leq \epsilon_{\min} + \max_{nk} \|\mathtt{X}_{\text{opt}}\boldsymbol{\beta}_k - \bar{\mathtt{X}}\boldsymbol{\beta}_k\| && (18) \\
&\quad\leq \epsilon_{\min} + 2\max_{k} \|\boldsymbol{\beta}_k\| \sin(\sqrt{3}\sigma/2) && (19)
\end{aligned}$$

The second part of the proof is almost identical to the one provided in Hartley and Kahl [4] and omitted.

4. Minimizing the Pixel Space Error

The error norm in the inequality in Eq.(14) is defined by the angle between the two vectors $\mathbf{u}$ and $\hat{\mathbf{u}} = \bar{\mathtt{X}}\mathtt{B}\bar{\mathtt{X}}^\top\mathbf{v}$. In practice, instead of the angle distance we usually deal with the pixel distance which is defined by

$$d_p(\mathbf{u}, \hat{\mathbf{u}}) = f\|\mathbf{u}/u_3 - \hat{\mathbf{u}}/\hat{u}_3\| \tag{20}$$

where f is the focal length of the camera. We may use either the L_2 norm or L_∞ norm for the pixel distance d_p. In this paper, we adopt the L_∞ norm for the pixel distance

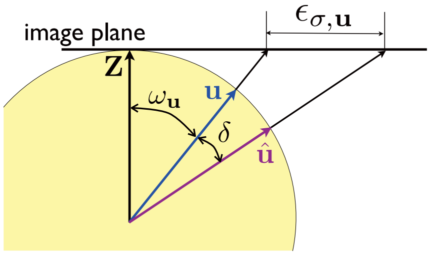

Figure 2. From angular error bound to the pixel error bound.

of a correspondence $(\mathbf{u}, \mathbf{v})$. Because the vector in the image plane is obtained by extending the unit-length vector, the pixel space distance can be attained accordingly. Fig.2 shows a two dimensional illustration for this geometrical relationship. Lemma 6 is achieved based on this drawing. Before we introduce the result, let $\omega_{\mathbf{u}}$ be the angle of $\mathbf{u}$ with respect to the viewing direction $\mathbf{Z} = [0, 0, 1]^\top$. Then, we have the following identity:

$$\tan \omega_{\mathbf{u}} \quad = \quad \sqrt{(u_1/u_3)^2 + (u_2/u_3)^2} \; . \tag{21}$$

Now let us use a new notation δ for simplicity:

$$\delta = 2\|\boldsymbol{\beta}_{\max}\| \sin(\sqrt{3}\sigma/2) \; . \tag{22}$$

Lemma 6 *The inequality for the feasibility test in the pixel space is given by*

$$\max_{nk} d_p(\mathbf{u}_{nk}, \bar{\mathtt{X}}\mathtt{B}_k\bar{\mathtt{X}}^\top \mathbf{v}_{nk}) \le \epsilon_{\min,\text{pixel}} + f\epsilon_\sigma \tag{23}$$

where $\epsilon_{\min,\text{pixel}}$ is the L_∞ residual for the current estimate $\mathtt{X}^*$, and ϵ_σ is the maximal deviation in the pixel space due to the variation of the rotation, given by

$$\epsilon_\sigma = \max_{nk} \epsilon_{\sigma,\mathbf{u}_{nk}} \tag{24}$$

$$= \max_{nk} \{\tan(\omega_{\mathbf{u}_{nk}} + \delta) - \tan \omega_{\mathbf{u}_{nk}}\} \tag{25}$$

$$= \max_{nk} \left\{ \frac{\tan \omega_{\mathbf{u}_{nk}} + \tan \delta}{1 - \tan \omega_{\mathbf{u}_{nk}} \tan \delta} - \tan \omega_{\mathbf{u}_{nk}} \right\} . \tag{26}$$

Note that we may easily compute the constant ϵ_σ which is necessary for the feasibility test. This lemma allows us to use the pixel distance instead of the angle distance.

5. The branch-and-bound Algorithm

To start the branch-and-bound algorithm, we need an initial rotation $\mathtt{X}^*$ to obtain $\epsilon_{\min}$. But we can choose any rotation for this initialization. Practically, we may choose the identity matrix $\mathtt{X}^* = \mathtt{I}$. In the loop of the branch-and-bound, we subdivide cubes $\{D_j(\sigma)\}$ into eight sub-cubes and therefore the number of cubes for the feasibility test increases eight times. Note that the population of remaining cubes does not explode by the octal subdivision because infeasible cubes are not considered in the next phase. Finally, the loop stops when the size of the cube is sufficiently small.

Algorithm 1 The B&B for the camera-r-sensor calibration.

```
Input: $\mathtt{X}^* = \mathtt{X}_0$, $\epsilon_{\min} = e_\infty(\mathtt{X}^*)$
repeat
    Octal subdivision ($\sigma \leftarrow \sigma/2$) of every $D_j$.
    for each of $\{D_j\}$ do
        Check the feasibility by using (14) or (23).
        if infeasible then
            discard the domain $D_j$.
        else
            update $\epsilon_{\min}$ and $\mathtt{X}^* = \bar{\mathtt{X}}_j$ if $\epsilon_{\min} < e_\infty(\bar{\mathtt{X}}_j)$ .
        end if
    end for
until $\sigma < \sigma_{\min}$
```

6. Experiments

6.1. Experiments with synthetic data

To see the performance of our algorithm, we generated synthetically sets of image coordinates. Ten pairs of views were generated to test the calibration algorithm; the relative motion for each pair was a random rotation whose angle was chosen in the range of $[0, 8]$ degrees for both x and y axes. The unknown rotational displacement $\mathtt{X}$ was determined randomly in the range of $[0, 5]$ degrees for each of the three rotation axes. An image had 100 points, each of which were contaminated by a Gaussian noise of standard deviation $\tau = 0.5$ pixels

Fig. 3 shows the evolution of the $\epsilon_{\min,\text{pixel}}$ for a data set of ten pairs of views. Initially, we chose $\mathtt{X} = \mathtt{I}_{3\times 3}$ to compute the starting value of $\epsilon_{\min,\text{pixel}}$. We kept $\bar{\mathtt{X}}$ through the branch-and-bound iteration that yielded the smallest $\epsilon_{\min,\text{pixel}}$ as the best solution. The experiment shown in Fig.3 resulted in 1.84 pixels as the final estimation residual and took 21 seconds for the whole computation.

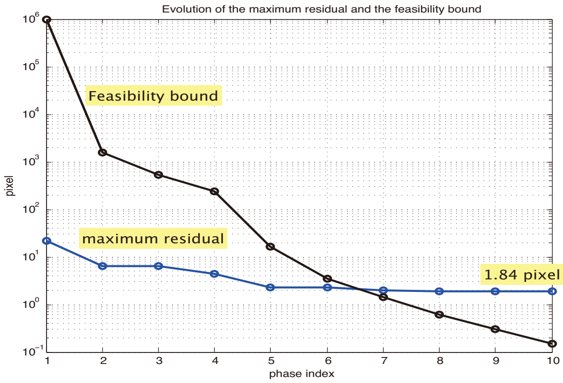

Figure 3. The evolution of the maximum residual $\epsilon_{\min,\text{pixel}}$ (blue line) and the maximum of the feasibility bounds $f\epsilon_\sigma$ (black line) through the branch-and-bound iteration. The ordinate is in log-scale.

Then, we generated a hundred sets of ten pairs of views and ran the branch-and-bound algorithm to see the statistics of the computation results. Figure 4 shows two histograms of residuals obtained. Left is the histogram of the maximum residuals $\epsilon_{\min,\text{pixel}}$; its average is 1.8 pixels even though some data set resulted in the maximum residual as large as 2.4 pixels. The level of the synthetic noise is $\tau = 0.5$, and the average maximum residual is around 3.6τ. Considering the distribution of the values of the random noise, we believe that this value is in a reasonable range. Right is the histogram of the rms (root-mean-squared) values for the computation. Even though we minimize the maximum residual to computed the L_∞ estimate $\mathtt{X}^*$, we want to see its rms performance. The rms is computed by using every residual due to $\mathtt{X}^*$. The average rms is 0.75 pixel and again we believe it is a reasonable result considering the noise level injected to each of the image coordinates.

Figure 5 shows histograms of computation time (left) and angle deviation (right), respectively. It took 13.5 seconds on the average and 32 seconds at the maximum. The deviation Δ of the rotational angle of the best estimate $\mathtt{X}^*$ with respect to $\mathtt{X}$ is defined as:$\Delta = \cos^{-1}((\text{trace}(\mathtt{X}^\top \mathtt{X}^*) - 1)/2)$. Right figure shows the histogram of Δ's Maximum deviation here is 1.6 degrees.

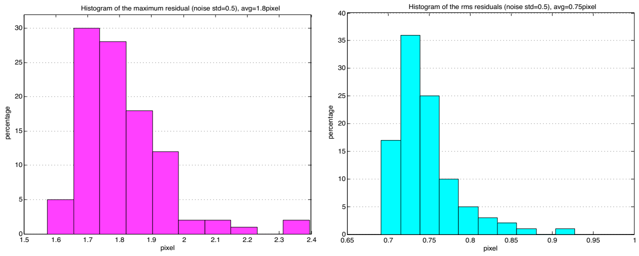

Figure 4. Histograms of the maximum residuals (left) and the rms residuals (right).

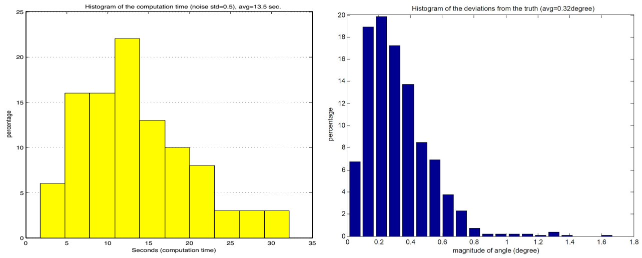

Figure 5. (Left) Histogram of the computation times in second. (Right) Histogram of the angles of $\mathtt{X}^\top \mathtt{X}^*$ in degree, computed by $\Delta = \cos^{-1}((\text{trace}(\mathtt{X}^\top \mathtt{X}^*) - 1)/2)$.

6.2. Experiments with Real Data Sets

Calibration using a real data set. In order to test our camera-sensor calibration algorithm for a set of real images, we first calibrated the internal parameters of the camera. As the pair of camera and rotation-sensor system, we used a camera mounted on a pan-tilt. The rotation angle applied to the pan-tilt between two views was recorded and used to calculate the corresponding rotation matrix $\mathtt{B}_i$. In total, we took five pairs of images for the calibration. Figure 6 shows a pair of images among them. We tried to keep the distance from the camera to the scene as far as possible to achieve a configuration of the pure-rotation. Our branch-and-bound algorithm resulted in the L_∞ residual 10.7pixels. The average of residuals using the estimated parameters were calculated to be 8.1pixels. Figure 7 shows a histogram of all the residuals, $e_{nk} = d_p(\mathbf{u}_{nk}, \mathtt{X}\mathtt{B}_k\mathtt{X}^\top \mathbf{v}_{nk})$ defined in (20). We observe that the residual level of this experiment is rather high compared to the results attained in the synthetic experiments. However, when we used this result for a 3D reconstruction with the metod of the L_∞ structure and motion, the recoved scene and the re-projection level were both satisfactory. It is provided below.

The L_∞ 3D reconstruction using the calibration. We then captured another triplet of images for a three-dimensional reconstruction with the method of the L_∞ structure and motion. We chose these images (Figure 8) because the result of 3D reconstruction could be examined easily due to its planar structure. The camera-pan-tile system was moved so that no extra rotation was induced between the views except for the rotation given by the control command to the pan-tilt. Extracted were the matching points among the three views, and the scene points and the camera location were reconstructed using the L_∞ reconstruction algorithm [7]. The camera rotations were set as $\mathtt{X}\mathtt{B}_i\mathtt{X}^\top$ where $\mathtt{B}_i$ was the rotation obtained from the rotation sensor between view 0 and view i. The first camera's rotation term was set to $\mathtt{I}_{3\times3}$. Figure 9 shows a histogram

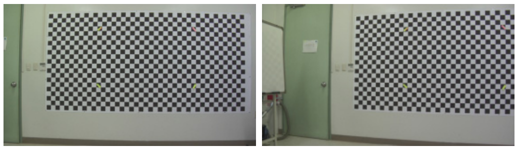

Figure 6. A pair among five of input images used for the calibration of rotational displacement.

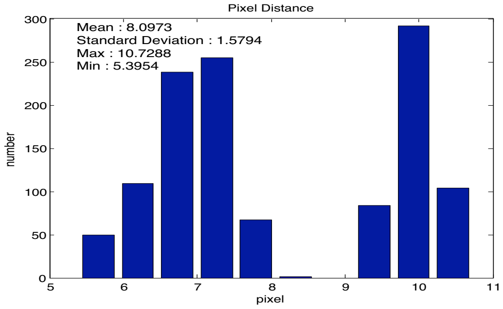

Figure 7. Histogram of residuals obtained from a real data set.

of the re-projection errors of the reconstruction. The maximum residual was obtained to be 11.9pixels and the average of the residuals was 4.3pixels as shown in the figure.

Figure 10 shows two three-dimensional views of the reconstructed scene points and the three locations of the camera. Note that all the scene points are aligned in a planar shape. Especially, the bird-eye view shows that the recovered scene points form a linear structure when viewed from above.

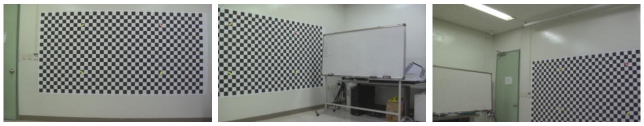

Figure 8. The three input images used for the L_∞ structure and motion reconstruction.

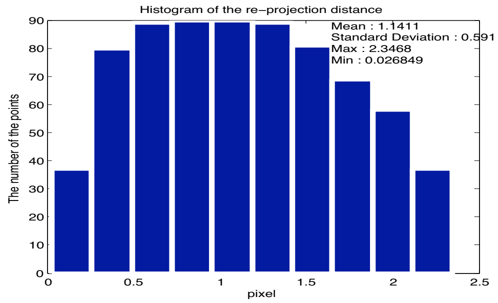

Figure 9. Histogram of residuals produced by the L_∞ structure and motion. The rotation matrices $\mathtt{XB}_i\mathtt{X}^\top$ of the camera were provided as input ($\mathtt{B}_i$s were from pan-tilt control commands and $\mathtt{X}$ were given by our branch-and-bound algorithm).

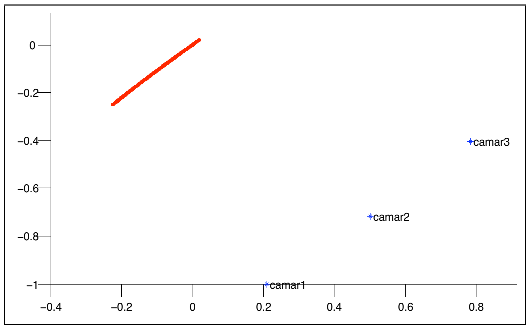

Figure 10. Three dimensional views of the result of the L_∞ reconstruction. One may observe that the reconstructed scene points are all contained in a plane. The bird-eye view shows that the the reconstructed 3D points are all aligned along a line, which provides the accuracy qualitatively of the camera-rotation-sensor calibration.

7. Conclusion

This paper proposed a method for globally optimal estimation of the relative rotation between a camera and the rotation sensor attached to it. For this, we employed a branch-and-bound algorithm in the L-infinity sense. Compared to previous methods, our approach minimizes the image space error directly related to the measurements and provides the global opimum. Our algorithm is based on the observation that the residual function can be evaluated when the rotation matrix is known. To efficiently reduce the search volume in the rotation space, we proposed a novel feasibility test for the branch-and-bound algorithm based on the theory of Lie-algebra. The experimental results show that our algorithm reliably finds the optimal solutions on both synthetic and real data sets.

References

[1] K. Daniilidis. Hand-eye calibration using dual quaternions. *Int. J. Robot. Res.*, 18(3):286–298, 1999.

[2] F. Dornaika and R. Horaud. Simultaneous robot-world and hand-eye calibration. *IEEE Journal of Robotics and Automation*, 14(4):617–622, 1998.

[3] R. Hartley and F. Schaffalitzky. L_∞ minimization in geometric reconstruction problems. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.

[4] R. I. Hartley and F. Kahl. Global optimization through rotation space search. *Int. J. Computer Vision*, 82:64–79, 2009.

[5] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Press, 2004.

[6] J. D. Hol, T. B. Schon, and F. Gustafsson. Relative pose calibration of a spherical camera and an IMU. In *IEEE International Symposium on Mixed and Augmented Reality*, 2008.

[7] F. Kahl and R. Hartley. Multiple-view geometry under the L_∞-norm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(9):1603–1617, 2008.

[8] Q. Ke and T. Kanade. Quasiconvex optimization for robust geometric reconstruction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(10):1834–1847, 2007.

[9] J. Lobo and J. Dias. Relative pose calibration between visual and inertial sensors. *International Journal of Robotics Research*, 26(6):561–575, 2007.

[10] F. M. Mirzaei and S. I. Roumeliotis. A Kalman filter-based algorithm for IMU-Camera calibration: Observability analysis and performance evaluation. *IEEE Transactions on Robotics*, 24(5):1143–1155, 2007.

[11] F. Park and B. Martin. Robot sensor calibration: solving $AX = XB$ on Euclidean group. *IEEE Journal of Robotics and Automation*, 10(5):717–721, 1994.

[12] Y. Shiu and S. Ahmad. Calibration of a wrist-mounted robotic sensors by solving homogeneous transformation equation of the form $AX = XB$. *IEEE Journal of Robotics and Automation*, 5(1):16–29, 1989.

[13] K. H. Strobl and G. Hirzinger. Optimal hand-eye calibration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[14] H. Zhuang. Hand/eye calibration for electronic assembly robots. *IEEE Journal of Robotics and Automation*, 14(4):612–616, 1998.