A Linear Generalized Camera Calibration from Three Intersecting Reference Planes

Mai Nishimura* Shohei Nobuhara Takashi Matsuyama
Graduate School of Informatics, Kyoto University, Japan
{nisimura,nob,tm}@vision.kuee.kyoto-u.ac.jp

Shinya Shimizu Kensaku Fujii
NTT Media Intelligence Laboratories, NTT Corporation, Japan
{shimizu.shinya,fujii.kensaku}@lab.ntt.co.jp

Abstract

This paper presents a new generalized (or ray-pixel, raxel) camera calibration algorithm for camera systems involving distortions by unknown refraction and reflection processes. The key idea is use of intersections of calibration planes, while conventional methods utilized collinearity constraints of points on the planes. We show that intersections of calibration planes can realize a simple linear algorithm, and that our method can be applied to any ray-distributions while conventional methods require knowing the ray-distribution class in advance. Evaluations using synthesized and real datasets demonstrate the performance of our method quantitatively and qualitatively.

1. Introduction

3D analysis and motion measurement of underwater objects enable various applications such as discovering the process of fertilized egg development and kinematic analysis of fish motion, etc. Such applications are expected to contribute for bioinformatics or other industrial fields, however, there have been no general techniques which can handle complicated refraction, reflection, and attenuation of underwater environments. We therefore aim at establishing a new general technique and realizing multi-view camera system for underwater environment.

In computer vision, real cameras do not exactly follow ideal projection models such as perspective or orthographic projections, and captured images always show geometric distortions. For some special camera systems, such distortions can be modeled by displacements on the imaging plane[4, 1] in practice. However, such distortion models cannot fit with general cases such as distortions by catadioptric systems or refractive housings of unknown geometry as shown in Figure 1.

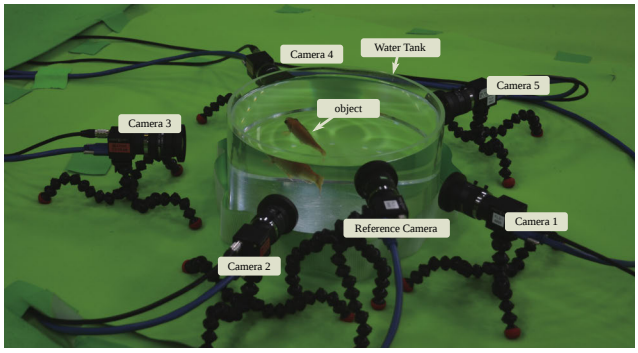

Figure 1. Multi-view camera system for underwater object capture

To handle such general cases involving unknown distortions, Grossberg and Nayer[3] have proposed the concept of generalized (or ray-pixel, raxel) camera model in which each pixel is associated with a 3D ray outside of the system as shown in Figure 2, and realized 3D ray modeling in the target space without modeling the refraction and/or reflection processes explicitly.

However, conventional methods for such 3D ray modeling require either calibration objects whose global 3D positions are given, or prior knowledge on the ray-distribution class (e.g. axial, central, etc.) of the system in order to switch the algorithm to be applied.

The goal of this paper is to propose a new linear calibration algorithm that can overcome these limitations, and the key idea on realizing such algorithm is use of intersections of calibration planes.

Our contribution is twofold: (1) a new linear calibration of generalized camera model that utilizes intersections

*Presently with NTT Software Innovation Center, NTT Corporation, Japan

of reference planes, and (2) a practical algorithm that detects intersections of reference planes from observed images. In what follows, we show that intersections of calibration planes can realize a distribution-independent formulation, and evaluate its performance in comparison with the state-of-the-art quantitatively and qualitatively with synthesized and real datasets.

2. Related Works

In order to model geometric distortions without explicitly modeling catadioptric systems or refractive housings of unknown geometry, Grossberg and Nayer have proposed the concept of generalized (or ray-pixel, raxel) camera model[3], and showed a calibration method using reference planes whose positions are given a priori. This idea is successfully applied for underwater photography to account for the refraction by housings or water tanks[9, 15, 2, 16]. However, these approaches require that the 3D geometry of reference planes in the target space, e.g., water, is given in a unified coordinate system, and they used extra markers exposed in the air or mechanical devices such as sliders or rotation tables.

Contrary to this approach, Ramalingam and Sturm have showed another approach utilizing three calibration planes of unknown postures [14, 11, 12, 10, 13]. These methods do not require such 3D geometry of calibration planes. Instead, they use collinearity constraints of 3D points projected to a same pixel in order to estimate the plane poses. Considering the distribution of rays in the target space, they showed that generalized cameras are classified into four classes (central, axial, crossed-slits, fully non-central), and proposed class-specific algorithms by assuming distribution-dependent degeneracies of the system. That is, they realized image-based calibrations, but such methods can be applied only to known ray-distributions.

Compared with these state-of-the-arts, our method does not require 3D geometry of the calibration objects and the knowledge on the ray-distribution class in advance. Similarly to them, our method also utilizes three reference planes of unknown postures, and estimates their postures from observed images in order to calibrate the rays in the target space.

The key difference is that our method utilizes not the collinearity but intersections of planes. That is, we estimate the relative postures of the planes by detecting the 3D locations on each plane where it intersects with another one. Since our method does not require the class-specific knowledge, our method can be applied to any unknown ray-distributions.

Besides, the state-of-the-arts[14, 11, 12, 10, 13] almost consist of linear steps but involve a non-linear process to seek a solution in null spaces inevitably. On the other hand, our method does not require such non-linear process and consists of a simple linear process.

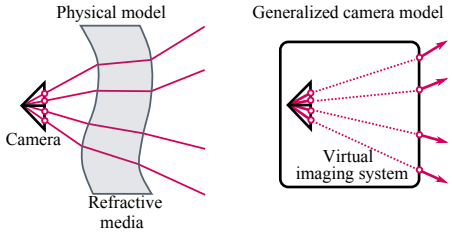

Figure 2. Generalized (or ray-pixel, raxel) camera model[3]

3. Measurement Model

The generalized (or ray-pixel, raxel) camera model[3] is defined as a set of 3D rays in the target space where an object exists, and each of the rays is associated with a pixel of the camera imager (Figure 2).

Let $\mathcal{X}$ denote the target space. We assume rays $\boldsymbol{q}$ go straight in $\mathcal{X}$, and there exists only a single ray in $\mathcal{X}$ captured by each pixel by assuming a pinhole camera. The problem we address is to estimate the 3D geometry of such rays in $\mathcal{X}$ associated with pixels by observing some calibration objects in $\mathcal{X}$.

Suppose we capture a calibration plane under three unknown different postures Φ_0, Φ_1, and Φ_2 in $\mathcal{X}$, and the calibration plane has feature points $\boldsymbol{p}$ whose position on the plane local coordinate system is given.

Let $\boldsymbol{p}^{[k]} = (u, v, 0)^\top$ denote a point on the calibration plane of the kth posture. By denoting the rigid motion between Φ_k and Φ_0 by

$$R_k = \begin{pmatrix} r_{k1} & r_{k2} & r_{k3} \\ r_{k4} & r_{k5} & r_{k6} \\ r_{k7} & r_{k8} & r_{k9} \end{pmatrix}, \quad \boldsymbol{t}_k = \begin{pmatrix} t_{k1} \\ t_{k2} \\ t_{k3} \end{pmatrix}, \tag{1}$$

we can describe the points on Φ_1 and Φ_2 in the Φ_0 local coordinate system as

$$\boldsymbol{p}^{[0]} = R_1 \boldsymbol{p}^{[1]} + \boldsymbol{t}_1, \tag{2}$$

$$\boldsymbol{p}^{[0]} = R_2 \boldsymbol{p}^{[2]} + \boldsymbol{t}_2, \tag{3}$$

respectively.

4. Linear Generalized Camera Calibration using Three Intersecting Planes

Our calibration using three images of the calibration planes consists of three steps: (1) detection of the intersections of the planes, (2) estimation of the plane postures, and (3) ray generation using collinear points.

The last step is identical to the existing method[14]. That is, once recovered the plane postures in a single coordinate system, we can obtain the 3D ray corresponding to a pixel by detecting the points on the calibration planes such that they are projected to the pixel in question.

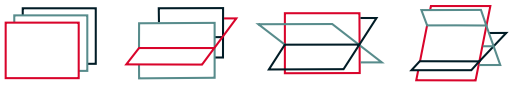

Figure 3. Special cases of three planes. (a)parallel to each other, (b)two parallel planes and the other cuts each in a line, (c)intersects in a line, (d)form a prismatic surface

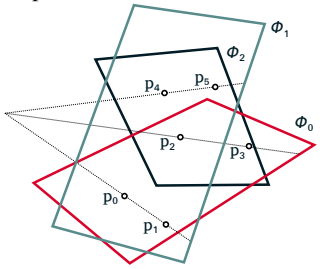

Figure 4. Intersections of three reference planes

In this section, we first introduce the second step, the core of our theoretical contribution, and then introduce the first step with our practical apparatus.

4.1. Pose Estimation of Three Intersecting Planes

Suppose we can detect (1) pixels in the captured images where the calibration planes intersect, and (2) corresponding 3D points on the calibration planes projected to the pixel. The key constraint we use in this section is that if a pixel corresponds to the intersection of two calibration planes, then the corresponding 3D points on the different planes are a coincident point in $\mathcal{X}$.

By ignoring special cases illustrated in Figure 3, three planes always have intersections each other, and such lines intersect at a single point by definition (Figure 4). The goal of this section is to estimate R_1, R_2, $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ by using coincident points on such intersections. Notice that the special cases of Figure 3 can be detected by verifying the rank of M of Eq (9) automatically as described later.

Given two points on each of the three intersecting lines as shown in Figure 4, coincident points $\boldsymbol{p}_0^{[k]}, \ldots, \boldsymbol{p}_5^{[k]}$ on such intersections provide the following equations:

$$\boldsymbol{p}_i^{[0]} = R_1 \boldsymbol{p}_i^{[1]} + \boldsymbol{t}_1, (i = 0, 1), \quad (4)$$

$$\boldsymbol{p}_i^{[0]} = R_2 \boldsymbol{p}_i^{[2]} + \boldsymbol{t}_2, (i = 2, 3), \quad (5)$$

$$R_1 \boldsymbol{p}_i^{[1]} + \boldsymbol{t}_1 = R_2 \boldsymbol{p}_i^{[2]} + \boldsymbol{t}_2, (i = 4, 5), \quad (6)$$

where $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$ are on the intersection of Φ_0 and Φ_1, $\boldsymbol{p}_2$ and $\boldsymbol{p}_3$ are on that of Φ_0 and Φ_2, and $\boldsymbol{p}_4$ and $\boldsymbol{p}_5$ are on that of Φ_1 and Φ_2.

This provides 18 linear constraints on 18 parameters which consist of R_1, R_2, $\boldsymbol{t}_1$, $\boldsymbol{t}_2$ to be estimated except r_{k3}, r_{k6}, r_{k9} $(k = 1, 2)$ corresponding to the z-axis. Here, other corresponding points observed on the intersection lines do not provide additional constraints mathematically. However they can contribute to make the system robust to noise.

The above constraints do not enforce the rotation matrices be $SO(3)$. Consequently, Eq (4) ... Eq (6) do not give a unique solution and the rank of the above system always becomes 15. We therefore introduce additional constraints based on inner products as follows.

$$\overline{\boldsymbol{p}_0^{[1]}\boldsymbol{p}_1^{[1]}} \cdot \overline{\boldsymbol{p}_4^{[1]}\boldsymbol{p}_5^{[1]}} = \overline{\boldsymbol{p}_0^{[0]}\boldsymbol{p}_1^{[0]}} \cdot R_1 \overline{\boldsymbol{p}_4^{[1]}\boldsymbol{p}_5^{[1]}}, \quad (7)$$

$$\overline{\boldsymbol{p}_2^{[2]}\boldsymbol{p}_3^{[2]}} \cdot \overline{\boldsymbol{p}_4^{[2]}\boldsymbol{p}_5^{[2]}} = \overline{\boldsymbol{p}_2^{[0]}\boldsymbol{p}_3^{[0]}} \cdot R_2 \overline{\boldsymbol{p}_4^{[2]}\boldsymbol{p}_5^{[2]}}, \quad (8)$$

where $\overline{\boldsymbol{p}_i^{[k]}\boldsymbol{p}_j^{[k]}} = \boldsymbol{p}_j^{[k]} - \boldsymbol{p}_i^{[k]}$. The key point is that these are defined as linear constraints by utilizing inner products given in different local coordinates of the planes. By adding these two constraints, we have 20 linear equations that form a linear system $M\boldsymbol{x} = \boldsymbol{b}$ of rank 17 for 18 parameters, where

$$M = \begin{pmatrix} \tilde{\boldsymbol{p}}_0^{[1]\top} \otimes I_{3\times 3} & \mathbf{0}_{3\times 9} \\ \tilde{\boldsymbol{p}}_1^{[1]\top} \otimes I_{3\times 3} & \mathbf{0}_{3\times 9} \\ \mathbf{0}_{3\times 9} & \tilde{\boldsymbol{p}}_2^{[2]\top} \otimes I_{3\times 3} \\ \mathbf{0}_{3\times 9} & \tilde{\boldsymbol{p}}_3^{[2]\top} \otimes I_{3\times 3} \\ \tilde{\boldsymbol{p}}_4^{[1]\top} \otimes I_{3\times 3} & -\tilde{\boldsymbol{p}}_4^{[2]\top} \otimes I_{3\times 3} \\ \tilde{\boldsymbol{p}}_5^{[1]\top} \otimes I_{3\times 3} & -\tilde{\boldsymbol{p}}_5^{[2]\top} \otimes I_{3\times 3} \\ \overline{\boldsymbol{p}_4^{[1]}\boldsymbol{p}_5^{[1]}}^\top \otimes \overline{\boldsymbol{p}_0^{[0]}\boldsymbol{p}_1^{[0]}}^\top & \mathbf{0}_{1\times 9} \\ \mathbf{0}_{1\times 9} & \overline{\boldsymbol{p}_4^{[2]}\boldsymbol{p}_5^{[2]}}^\top \otimes \overline{\boldsymbol{p}_2^{[0]}\boldsymbol{p}_3^{[0]}}^\top \end{pmatrix}, \quad (9)$$

and

$$\boldsymbol{x} = \begin{pmatrix} r_{11} \\ r_{14} \\ r_{17} \\ r_{12} \\ r_{15} \\ r_{18} \\ t_{11} \\ t_{12} \\ t_{13} \\ r_{21} \\ r_{24} \\ r_{27} \\ r_{22} \\ r_{25} \\ r_{28} \\ t_{21} \\ t_{22} \\ t_{23} \end{pmatrix}^\top, \quad \boldsymbol{b} = \begin{pmatrix} \boldsymbol{p}_0^{[0]} \\ \boldsymbol{p}_1^{[0]} \\ \boldsymbol{p}_2^{[0]} \\ \boldsymbol{p}_3^{[0]} \\ \mathbf{0}_{6\times 1} \\ \overline{\boldsymbol{p}_0^{[1]}\boldsymbol{p}_1^{[1]}} \cdot \overline{\boldsymbol{p}_4^{[1]}\boldsymbol{p}_5^{[1]}} \\ \overline{\boldsymbol{p}_2^{[2]}\boldsymbol{p}_3^{[2]}} \cdot \overline{\boldsymbol{p}_4^{[2]}\boldsymbol{p}_5^{[2]}} \end{pmatrix}. \quad (10)$$

Here $I_{n\times n}$, $I_{2\times 3}$, and $\mathbf{0}_{n\times m}$ denote the $n \times n$ identity matrix, $\left(\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right)$, and the $n \times m$ zero matrix respectively. $\boldsymbol{p}_i^{[k]}(u_i, v_i, 0)^\top$ denotes the point on Φ_k, and $\tilde{\boldsymbol{p}}_i^{[k]} = (u_i, v_i, 1)^\top$. $\otimes$ denotes Kronecker product. Besides the rank of M indicates if the planes are in a special configuration (Figure 3) or not, since such special configurations introduces linearly-dependent equations.

This linear system $M\boldsymbol{x} = \boldsymbol{b}$ can be further decomposed

into two linear systems $M_1\boldsymbol{x}_1 = \boldsymbol{b}_1$ and $M_2\boldsymbol{x}_2 = \boldsymbol{b}_2$, where

$$M_1 = \begin{pmatrix} \tilde{\boldsymbol{p}}_0^{[1]} \otimes \boldsymbol{I}_{2\times 2} & \boldsymbol{0}_{2\times 6} \\ \tilde{\boldsymbol{p}}_1^{[1]} \otimes \boldsymbol{I}_{2\times 2} & \boldsymbol{0}_{2\times 6} \\ \boldsymbol{0}_{2\times 6} & \tilde{\boldsymbol{p}}_2^{[2]} \otimes \boldsymbol{I}_{2\times 2} \\ \boldsymbol{0}_{2\times 6} & \tilde{\boldsymbol{p}}_3^{[2]} \otimes \boldsymbol{I}_{2\times 2} \\ \tilde{\boldsymbol{p}}_4^{[1]\top} \otimes \boldsymbol{I}_{2\times 2} & -\tilde{\boldsymbol{p}}_4^{[2]\top} \otimes \boldsymbol{I}_{2\times 2} \\ \tilde{\boldsymbol{p}}_5^{[1]\top} \otimes \boldsymbol{I}_{2\times 2} & -\tilde{\boldsymbol{p}}_5^{[2]\top} \otimes \boldsymbol{I}_{2\times 2} \\ \overline{\boldsymbol{p}_4^{[1]}\boldsymbol{p}_5^{[1]\top}} \otimes I_{2\times 3}\overline{\boldsymbol{p}_0^{[0]}\boldsymbol{p}_1^{[0]\top}} & \boldsymbol{0}_{1\times 6} \\ \boldsymbol{0}_{1\times 6} & \overline{\boldsymbol{p}_4^{[2]}\boldsymbol{p}_5^{[2]\top}} \otimes I_{2\times 3}\overline{\boldsymbol{p}_2^{[0]}\boldsymbol{p}_3^{[0]\top}} \end{pmatrix}, \tag{11}$$

$$\boldsymbol{x}_1 = \begin{pmatrix} r_{11} \\ r_{14} \\ r_{12} \\ r_{15} \\ t_{11} \\ t_{12} \\ r_{21} \\ r_{24} \\ r_{22} \\ r_{25} \\ t_{21} \\ t_{22} \end{pmatrix}, \boldsymbol{b}_1 = \begin{pmatrix} I_{2\times 3}\boldsymbol{p}_0^{[0]} \\ I_{2\times 3}\boldsymbol{p}_1^{[0]} \\ I_{2\times 3}\boldsymbol{p}_2^{[0]} \\ I_{2\times 3}\boldsymbol{p}_3^{[0]} \\ \boldsymbol{0}_{4\times 1} \\ \overline{\boldsymbol{p}_0^{[1]}\boldsymbol{p}_1^{[1]}} \cdot \overline{\boldsymbol{p}_4^{[1]}\boldsymbol{p}_5^{[1]}} \\ \overline{\boldsymbol{p}_2^{[2]}\boldsymbol{p}_3^{[2]}} \cdot \overline{\boldsymbol{p}_4^{[2]}\boldsymbol{p}_5^{[2]}} \end{pmatrix}, \tag{12}$$

and

$$M_2 = \begin{pmatrix} \tilde{\boldsymbol{p}}_0^{[1]\top} & \boldsymbol{0}_{1\times 3} \\ \tilde{\boldsymbol{p}}_1^{[1]\top} & \boldsymbol{0}_{1\times 3} \\ \boldsymbol{0}_{1\times 3} & \tilde{\boldsymbol{p}}_2^{[2]\top} \\ \boldsymbol{0}_{1\times 3} & \tilde{\boldsymbol{p}}_3^{[2]\top} \\ \tilde{\boldsymbol{p}}_4^{[1]\top} & -\tilde{\boldsymbol{p}}_4^{[2]\top} \\ \tilde{\boldsymbol{p}}_5^{[1]\top} & -\tilde{\boldsymbol{p}}_5^{[2]\top} \end{pmatrix}, \boldsymbol{x}_2 = \begin{pmatrix} r_{17} \\ r_{18} \\ t_{13} \\ r_{27} \\ r_{28} \\ t_{23} \end{pmatrix}, \boldsymbol{b}_2 = \boldsymbol{0}_{6\times 1}. \tag{13}$$

The solution $\boldsymbol{x}_1$ is given as $\boldsymbol{x}_1 = (M_1^\top M_1)^{-1} M_1^\top \boldsymbol{b}_1$, and $\boldsymbol{x}_2$ is given up to a scale factor α using the 6th right singular vector $\boldsymbol{v}_6$ of M_2 corresponding to the null space of M_2:

$$\boldsymbol{x}_2 = \alpha \boldsymbol{v}_6. \tag{14}$$

This α can be trivially determined by using the orthogonality constraints on R_1 and R_2:

$$\boldsymbol{r}_{k,1}^\top \boldsymbol{r}_{k,2} = 0, \quad |\boldsymbol{r}_{k,1}| = 1, \quad |\boldsymbol{r}_{k,2}| = 1. \tag{15}$$

Up to this point, we obtain the first and the second column vectors of R_1 and R_2 as well as $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$.

Finally, by recovering the third columns of R_1 and R_2 as the cross products of each of the first and the second column vectors, we obtain the complete R_1, R_2, $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ up to scale.

Notice that the above linear system allows a mirrored solution such that $R_k = \begin{pmatrix} r_{k1} & r_{k2} & -r_{k3} \\ r_{k4} & r_{k5} & -r_{k6} \\ -r_{k7} & -r_{k8} & -r_{k9} \end{pmatrix}$ and $\boldsymbol{t}_k = \begin{pmatrix} t_{k1} \\ t_{k2} \\ -t_{k3} \end{pmatrix}$. Selecting the correct solution can be done by verifying the direction of the z-axis.

4.2. Intersection Detection from Distorted Images

The crucial point to realize our method is (1) how to obtain the 2D position on the calibration plane only from the captured images under unknown distortions, and (2) how to detect intersections of the calibration planes under different postures from their images.

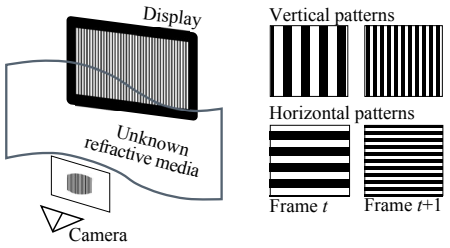

Figure 5. Gray code technique[3]

To this end, we employ a flat panel display as the calibration plane Φ, and utilize the gray code technique proposed in [3] (Figure 5). That is, we fix the display, and capture gray code patterns shown by it for each pose.

With this gray code technique, the first point is simply achieved by decoding the gray code representing the display pixel location as done by [3]. To achieve the second point, we propose a new method based on the difference of display pixel densities between captured images.

Suppose we have established display pixel to camera pixel correspondences for Φ_k by decoding the gray code images, and the correspondence is a one-to-many mapping where each display pixel is captured by multiple camera pixels. This assumption can be realized by making the camera to have effectively a higher resolution than the display by binning each $n \times n$ display pixels into a block.

Let $f_k(q)$ denote the display pixel on Φ_k corresponding to a camera pixel q. Once obtained such one-to-many correspondences between display and camera pixels, we can obtain a display density $d_k(q)$ of each camera pixel q by counting the camera pixels sharing the same display pixel $f_k(q)$ (Figure 10).

The key idea on detecting the intersection is to use the difference of $d_k(q)$ and $d_{k'}(q)$ of Φ_k and $\Phi_{k'}$. Obviously $d_k(q)$ and $d_{k'}(q)$ are not equal or proportional to the real depth since it is affected by both the depth and the refraction/reflection process. However, if the values $d_k(q)$ and $d_{k'}(q)$ are the same, it is guaranteed that the corresponding 3D points are at a same depth as long as the refraction / reflection process is kept static. That is, if $d_k(q) = d_{k'}(q)$, then the corresponding 3D points $f_k(q)$ and $f_{k'}(q)$ on different planes Φ_k and $\Phi_{k'}$ are a coincident point and hence are on the intersection of the planes.

Based on this idea, we design our intersection detection as follows: (1) establish the display-camera pixel correspondences by decoding the gray code[3], (2) compute the density map $d_k(q)$, (3) find the pixels such that $d_k(q) = d_{k'}(q)$ for each pair of planes Φ_k and $\Phi_{k'}$ and return the corresponding $f_k(q)$ and $f_{k'}(q)$ as 3D points on the intersection. Section 5.3.2 shows how this process works in our real underwater environment.

5. Evaluations

In order to evaluate the performance of our method, this section first evaluates the performance quantitatively using synthesized datasets. Then we introduce our apparatus for an underwater environment and show a qualitative evaluation using real images with a 3D shape reconstruction in water.

5.1. Error Metrics

Rotation and Translation Errors In case that the ground truth is available, we evaluate the quality of the calibration by estimation errors of R and $\boldsymbol{t}$ w.r.t the ground truth since the ray directions are defined by the plane poses.

The estimation error of R is defined as the Riemmanian distance[8]. Let R_g, $\boldsymbol{t}_g$ be the ground truth of $R, \boldsymbol{t}$, and $\theta = \cos^{-1}(\frac{\mathrm{tr}R'-1}{2})$. The rotation matrix error E_R is defined as

$$E_R = \frac{1}{\sqrt{2}}||\log(R^\top R_g)||_F, \tag{16}$$

$$\log R' = \begin{cases} 0 & (\theta = 0), \\ \frac{\theta}{2\sin\theta}(R - R'^\top) & (\theta \neq 0), \end{cases} \tag{17}$$

where $||\cdot||_F$ denotes the Frobenius norm.

The translation vector error E_T is defined by the root mean square error (RMSE)

$$E_T = \sqrt{||\boldsymbol{t} - \boldsymbol{t}_g||^2/3}. \tag{18}$$

Re-Intersection Error Similarly to the reprojection error widely used in the perspective multi-view calibration, we can consider the re-intersection error that measures the distance between the observed and the synthesized points on the calibration planes.

By denoting the observed point on kth plane for camera pixel i by $\boldsymbol{p}_{k,i}$, and the intersection of the calibration plane and the corresponding ray by $\hat{\boldsymbol{p}}_{k,i}$, we can define the re-intersection error as

$$E_p = \frac{1}{|I|\,|K|}\sum_{i\in I}\sum_{k\in K}|\hat{\boldsymbol{p}}_{k,i} - \boldsymbol{p}_{k,i}|^2, \tag{19}$$

where I and K are the set of the pixels and the calibration planes, and $|I|$ and $|K|$ denotes the number of their elements respectively.

5.2. Evaluation using Synthesized Data

To evaluate the performance quantitatively, we used synthesized datasets which simulate our real setup shown in Figure 8. We set the camera and the display resolutions as 1280×960 and 1024×2048 respectively. The first display defines the world coordinate system, and the sizes of the other devices are described in the display pixel. The camera is placed at 2662.4 pixels from the center of the water cylinder of radius 2048 pixels. Notice that the refractive index of water is set to 1.3, and this virtual cylinder has no housings so that the rays coming from the air directly refracted at the boundary between water and the air.

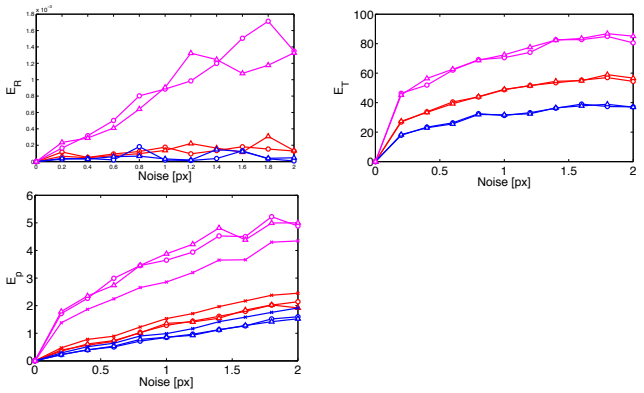

Figure 6. Plots of E_R, E_T, and E_p. Magenta: Ramalingam et al. [11] (60 points). Red: proposed (12 points). Blue: proposed (60 points). The cross, circle and triangle markers indicate the errors of the parameters of Φ_0,Φ_1 (i.e.$R_1, \boldsymbol{t}_1$) and Φ_2 (i.e.$R_2, \boldsymbol{t}_2$) respectively.

For each trial, we synthesized 3 different display postures Φ_0, Φ_1, Φ_2, and applied our calibration. Here the rotation matrices are generated using uniformly sampled random three Euler angles within $[-\pi/3 : \pi/3]$. The elements of the translation vectors are generated using uniformly sampled random values within $[-50 : 50]$ pixels.

Linear Calibration using Intersecting Reference Planes On each plane Φ, estimated intersections contain errors due to (a) zero-cross detection errors in the difference of pixel density maps, and (b) decoding errors on establishing the display to camera pixel correspondences. In this evaluation we focus on (b) since that also affects (a).

Suppose that we have the points on intersections calculated from synthesized posture data. To simulate the decoding errors in points on intersections, we added uniform random noise ranging between $[-\sigma : \sigma]$ $(\sigma = 0, \ldots, 2.0)$ pixels.

Figure 6 shows the results. Each point is the mean value of 100 trials at each noise level. The red and blue plots indicate the errors estimated with 12 and 60 points (2 and 10 points per intersection per plane, respectively). The magenta indicates the state-of-the-art[11] with 60 points (20 points per plane Φ). In this evaluation, [11] sampled points regularly on the image and ours sampled 2 or 5 points on each intersection as done in Figure 6[1].

[1]Our implementation is available at http://vision.kuee.kyoto-u.ac.jp/~nob/proj/iccv2015/.

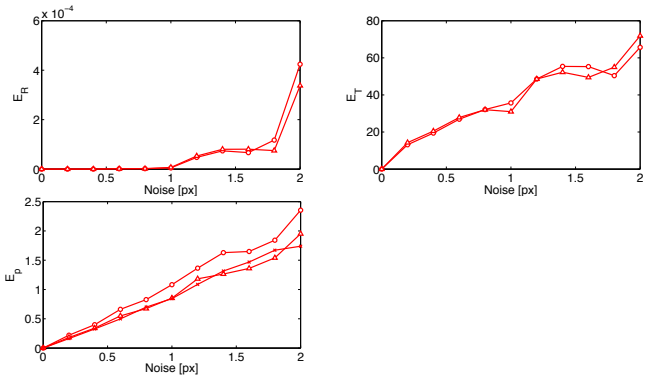

Figure 7. Plots of E_R, E_T, and E_p. The cross, circle and triangle markers indicate the errors of the parameters of Φ_0,Φ_1 and Φ_2 respectively.

From these results, we can observe that our method shows more robust performance than the state-of-the-art, while both methods return the exact results $E_R = E_t = 0$ at $\sigma = 0$.

Bundle Adjustment using Collinearity Constraints Using the linear calibration result as inputs, we applied the bundle adjustment using the collinearity constraint as done in the conventional methods[14]. That is, our method first utilizes the intersections to obtain the linear solution, and then the solution can be refined non-linearly using the collinearity constraint on other points, while the collinearity constraint itself do not provide a linear solution due to the ray-distribution dependent unknown degeneracy[12].

Figure 7 shows the average errors of 100 trials under different noise levels, where the red plots data in Figure 6 are used as initial inputs. From these results, we can conclude that our method can provide a reasonable initial values for the bundle adjustment, and the calibration errors are within a reasonable range in comparison with the injected noise.

5.3. Evaluation using Real Images

5.3.1 Apparatus for Underwater Environment

Figure 8 shows our setup. We used 4K cameras (Pointgrey FL3-U3-88S2C-C) to capture an object in an acrylic cylinder water tank of 300mm diameter and 5mm thick. The distance from the camera to the cylinder center was approximately 350mm. To calibrate the mapping between the camera pixels and corresponding rays in water, the cameras capture a waterproof tablet of 1366 × 768 resolution showing the gray code[3] as shown in Figure 5.

In this experiment, we exposed approximately the upper-half of the display in the air in order to obtain the reference calibration parameters using a baseline method[17] as shown as the green-boxed areas in Figure 9 (left). Notice

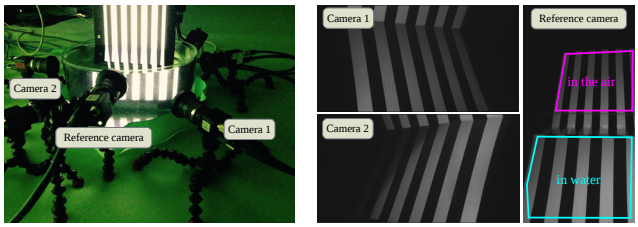

Figure 8. Experimental setup

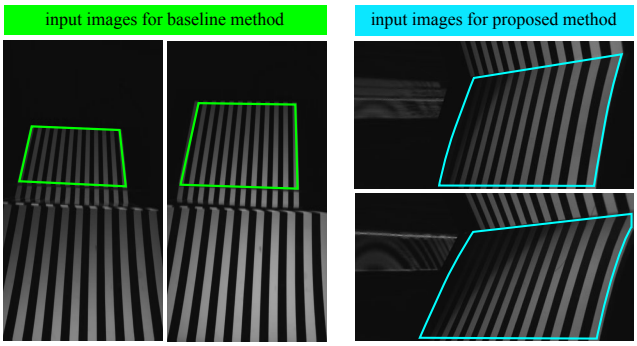

Figure 9. Examples of input images. Left: images for the baseline method captured by the reference camera. Right: images for the proposed method.

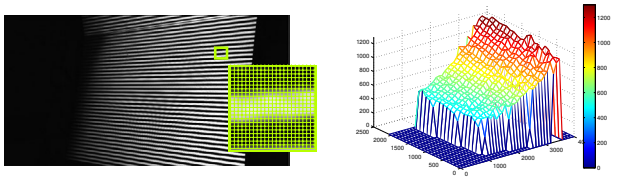

Figure 10. Pixel density estimation from images. Left: a captured image. Right: pixel density map.

that our calibration used only the lower-half of the display in water as shown as the cyan-boxed areas in Figure 9 (right).

5.3.2 Intersection Estimation

As described in Section 4.2, we performed intersection detection by creating the pixel density map. By computing camera pixels corresponding to 8 × 8 display pixel bins in the tablet from the captured images, we obtained pixel density maps as shown in Figure 10 (right).

To evaluate the quality of our intersection estimation, we compare our result with intersections calculated from the baseline method. Figure 11 shows the estimated intersections decoded onto each tablet of different postures. Blue to red color plots (*) indicate pixel-depth maps decoded to the display coordinate system, and Red and blue plots (o) indicate estimated intersections on each display, Φ_0 (left) and Φ_1 (right). Black lines (-) indicate the intersections of each

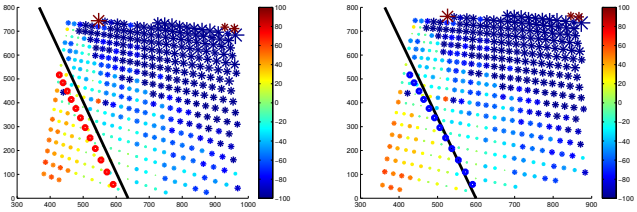

Figure 11. Estimated intersections on Φ_0 (left) and Φ_1 (right)

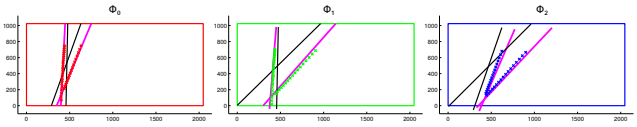

Figure 12. Intersection estimation results

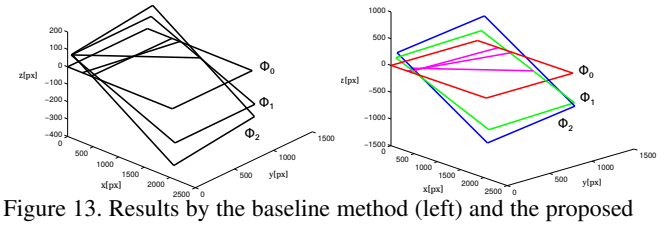

Figure 13. Results by the baseline method (left) and the proposed method (right)

display estimated by the baseline method. While the estimated intersections are not exactly coincident with the ones from the baseline method, and parallel to each other (Figure 11 left), or intersects each other (Figure 11 right), these could serve as a good enough estimation of intersections as shown in the next result.

5.3.3 Linear Calibration from Estimated Intersections

Figure 13 (left) shows the pose estimation result by the baseline method, and Figure 13 (right) shows the result by our proposed method. In Figure 12, black lines (-) indicate intersections calculated by the baseline method. Red, green, and blue markers ($\times$) indicate detected intersections from pixel density maps, and magenta lines (-) are intersections calculated by our proposed method. In our experiment, pixel density map-based intersection detection on the image plane did not perform perfectly in some images as shown in Φ_2 case in Figure 12 (right). This intersection-detection errors appeared as z-axial posture errors in (Figure 13 right).

Even so, these estimated pose parameters can be used as an initial values, and refined in the following multi-view calibration step with a global bundle adjustment.

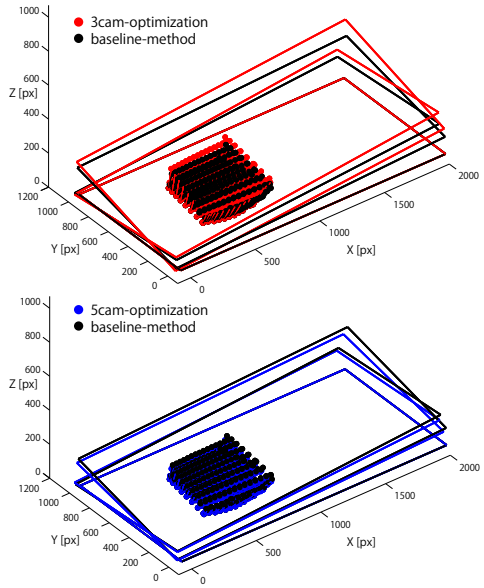

Figure 14. Calibration results compared to the baseline method. Top: the 3 cameras case. Bottom: the 5 cameras case.

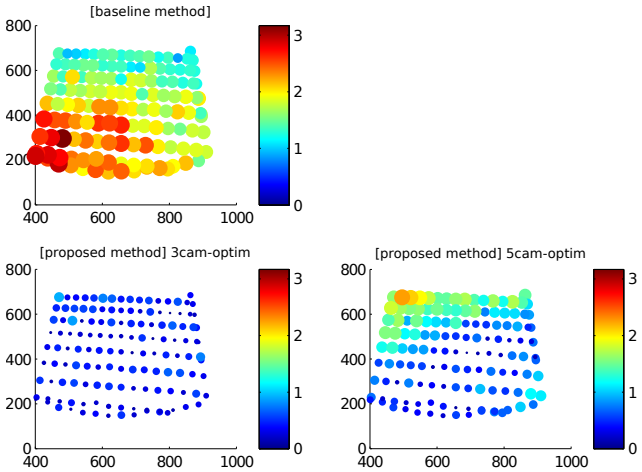

Figure 15. Re-intersection error distributions on the first plane. Top: the baseline method. Left: the 3 cameras case. Right: the 5 cameras case.

5.3.4 Multi-view Calibration and Underwater 3D Shape Capture

Figure 14 shows the calibration results of one of the cameras after the nonlinear refinement using 3 or 5 cameras, where 5 cameras form a closed loop around the cylinder tank while 3 cameras do not. Figures 15 and 16 show distributions of E_p at each display point on the first plane and the estimated rays in the cases of the 3 and 5 cameras together with the baseline result. In Figure 16, the dots illustrate the intersections with the planes and their color indicate the re-intersection error E_p. The average re-intersection error over

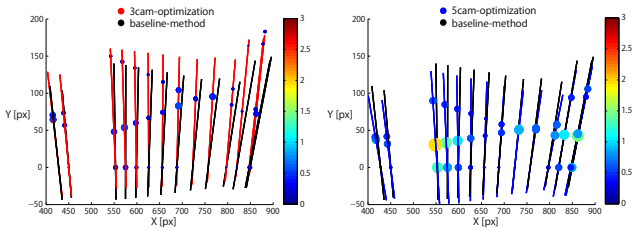

Figure 16. Estimated rays in the 3 cameras case (left) and the 5 cameras case (right). The position of the dots indicate the intersection with the planes, and their colors indicate the re-intersection errors.

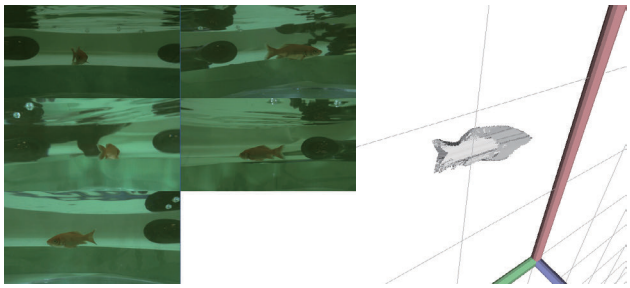

Figure 17. The input images (left) and the reconstructed 3D shape (right)

the all points of the all planes were 0.21 and 1.59 pixels for the 3 cameras and the 5 cameras setups while the average reprojection error for the baseline method as 0.88 pixels on the camera screen.

From these results, we can observe that the 5 camera case shows a comparable result to the baseline method while the 3 camera case results in a smaller re-intersection error but much different 3D plane postures. This fact indicates that the ray-pixel modeling can produce an overfitted result to the observation because of its flexibility, and multi-view generalized (or ray-pixel, raxel) camera calibration with a close loop can balance such flexibility and the robustness.

3D Shape Reconstruction To evaluate the calibration accuracy qualitatively, we captured a goldfish in the tank as shown in Figure 1, and applied the shape-from-silhouette method[5] to reconstruct the 3D shape. Figure 17 shows the input images taken by the camera 1 to 5 and the reconstructed 3D shape. This result demonstrates that our calibration can serve as a practical technique to realize a multi-view generalized (or ray-pixel, raxel) camera system for 3D measurement under unknown refractions.

6. Conclusion

In this paper, we proposed a new linear generalized (or ray-pixel or raxel) camera calibration algorithm. The key idea was to use intersections of the reference planes and to build a linear system utilizing coincident points on the intersections. Our method first utilizes the ray-independent intersections found where the collinearity constraint degenerates, and then our linear solution can be refined non-linearly using the collinearity constraint found other points on the plane as done in the conventional studies. Also we proposed a new practical method to detect such intersections from images with unknown distortions by exploiting the difference of pixel-density maps.

Compared with the state-of-the-arts[14, 11, 12, 10, 13] that involve non-linear solution finding processes in null spaces of their systems, our method requires only solving a simple linear system and can be applied without knowing the ray-distribution class in advance, since our method does not require adding class-specific constraints to solve the problem.

Limitations Our model does not require any regularizers such as continuities between 3D rays associated with neighboring pixels. This design maximizes its ability to model complex refraction / reflection processes, but such flexibility can also introduce unstable or erroneous results in the bundle adjustment utilizing the collinearity constraint as discussed in the evaluation. Hence integrating smoothness constraints between rays and/or pixels[7, 6] remains to be investigated.

Acknowledgement

This research is partially supported by NTT Media Intelligence Laboratories and by JSPS Kakenhi Grant Number 26240023.

References

[1] D. Claus and A. W. Fitzgibbon. A rational function lens distortion model for general cameras. In *Proc. CVPR*, volume 1, pages 213–219, 2005. 1

[2] J. Gregson, M. Krimerman, M. B. Hullin, and W. Heidrich. Stochastic tomography and its applications in 3d imaging of mixing fluids. In *Proc. SIGGRAPH*, pages 52:1–10, 2012. 2

[3] M. Grossberg and S. Nayar. The raxel imaging model and ray-based calibration. *IJCV*, 61(2):119–137, 2005. 1, 2, 4, 6

[4] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 1

[5] A. Laurentini. How far 3d shapes can be understood from 2d silhouettes. *TPAMI*, 17(2):188–195, 1995. 8

[6] P. Miraldo and H. Araujo. Calibration of smooth camera models. *TPAMI*, 35(9):2091–2103, 2013. 8

[7] P. Miraldo, H. Araujo, and J. Queiró. Point-based calibration using a parametric representation of the general imaging model. In *Proc. ICCV*, pages 2304–2311, 2011. 8

[8] M. Moakher. Means and averaging in the group of rotations. *SIAM Journal on Matrix Analysis and Applications*, 24(1):1–16, 2002. 5

[9] S. Narasimhan, S. Nayar, B. Sun, and S. Koppal. Structured light in scattering media. In *Proc. ICCV*, pages 420–427, 2005. 2

[10] S. Ramalingam, S. K. Lodha, and P. Sturm. A generic structure-from-motion framework. *CVIU*, 103(3):218–228, Sept. 2006. 2, 8

[11] S. Ramalingam, P. Sturm, and S. K. Lodha. Generic calibration of axial cameras. Technical Report RR-5827, INRIA, 2005. 2, 5, 8

[12] S. Ramalingam, P. Sturm, and S. K. Lodha. Towards complete generic camera calibration. In *Proc. CVPR*, pages 1093–1098, 2005. 2, 6, 8

[13] P. Sturm and J. P. Barreto. General imaging geometry for central catadioptric cameras. In *Proc. ECCV*, pages 609–622. 2008. 2, 8

[14] P. Sturm and S. Ramalingam. A generic concept for camera calibration. In *Proc. ECCV*, pages 1–13, 2004. 2, 6, 8

[15] B. Trifonov, D. Bradley, and W. Heidrich. Tomographic reconstruction of transparent objects. In *Proc. of Eurographics Conf. on Rendering Techniques*, pages 51–60, 2006. 2

[16] T. Yano, S. Nobuhara, and T. Matsuyama. 3D shape from silhouettes in water for online novel-view synthesis. *IPSJ Trans. on CVA*, 5:65–69, Jul 2013. 2

[17] Z. Zhang. A flexible new technique for camera calibration. *TPAMI*, 22:1330–1334, 1998. 6