Joint depth and color camera calibration with distortion correction

Wei Xiang

August 12, 2014

1 Basic

1.1 Color Camera Intrinsics

In this paper, the author uses pinhole model, i.e., given a point in color world coordinates $\mathrm{x}_c = [x_c, y_c, z_c]^T$ (the subscript c indicates 'color'), its projection to color image coordinate $\mathrm{p}_c = [u_c, v_c]^T$ is calculated through the following steps:

(1) First, the point is normalized by z coordinate: $\mathrm{x}_n = [x_n, y_n]^T = [x_c/z_c, y_c/z_c]^T$.

(2) Then perform distortion correction by using a similar intrinsic model proposed by Heikkila [1]:

$$\mathrm{x}_g = \begin{bmatrix} 2k_3 x_n y_n + k_4(r^2 + 2x_n^2) \\ k_3(r^2 + 2y_n^2) + 2k_4 x_n y_n \end{bmatrix} \tag{1}$$

$$x_k = (1 + k_1 r^2 + k_2 r^4 + k_5 r^6)\mathrm{x}_\mathrm{n} + x_g \tag{2}$$

where, $r^2 = x_n^2 + y_n^2$ and $\mathrm{k}_c = [k_1, ..., k_5]$ is a vector containing distortion coefficients.

(3) Finally, the image coordinates p_c are obtained:

$$\mathrm{p}_c = \begin{bmatrix} u_c \\ v_c \end{bmatrix} = \begin{bmatrix} f_{cx} & 0 \\ 0 & f_{cy} \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} u_{0c} \\ v_{0c} \end{bmatrix} \tag{3}$$

where $\mathrm{f}_c = [f_{cx}, f_{cy}]$ are focal lengths in x, y axes respectively and $\mathrm{p}_{0c} = [u_{0c}, v_{0c}]$ is the principle point. Again, subscript c means 'color'.

The author described above model by $\mathcal{L}_c = \{\mathrm{f}_c, \mathrm{p}_{0c}, \mathrm{k}_c\}$.

1.2 Depth Camera Intrinsics

The transformation between depth world coordinates $\mathrm{x}_d = [\mathrm{x}_d, \mathrm{y}_d, \mathrm{z}_d]^T$ and depth image coordinates $\mathrm{p}_d = [u_d, v_d]^T$ (where, subscript d indicates 'depth') follows a similar model to that used for the color camera with the respective parameter f_d and p_{0d}:

$$\mathrm{p}_d = \begin{bmatrix} u_d \\ v_d \end{bmatrix} = \begin{bmatrix} f_{dx} & 0 \\ 0 & f_{dy} \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} u_{0d} \\ v_{0d} \end{bmatrix} \tag{4}$$

where $[x_k, y_k]^T$ are coordinates of x_d after geometric distortion. Unlike color camera, whose distortion was performed in terms of forward model (world to image), the geometric distortion for depth camera is obtained by backward model (image to world), i.e., simply switching x_n and x_k in Eqs. (1) and (2). The backward model will be fully explained later in Section 2.2.4.

The raw data obtained from Microsoft Kinect device is actually a 11-bit number between 0-2047, called "disparity" (united via **kdu**–kinect disparity unit). To convert from disparity d to depth z_d, two parts are essential:

(1) **Distortion correction**: using an observed pattern to correct distortion. By taking depth images of the whole flat plane(an empty wall, for example), and then calculating the reprojection error with known camera parameters and designed distances to the wall, we can visually demonstrate the error residuals (kdu) at different depths. As shown in Fig. 1, the shape of the error pattern is constant but its magnitude decreases as the distance from the object increases, which is referred as "decay". Also, to have a better understanding of the pattern of decay, the author measured the reprojection errors from planes at several distances and normalized them by dividing all images by Fig. 1. As we can see from Fig. 2, the resulting medians of normalized error for every measured disparity, fits well to an exponential decay. Therefore, the distortion model can be constructed with per-pixel coefficients and decays exponentially with the increasing disparity, as the following equation:

$$d_k = d + D_\delta(u, v) \cdot \exp(\alpha_0 - \alpha_1 d) \tag{5}$$

where d is the distorted disparity as returned by the Kinect, D_δ contains the spatial distortion pattern, and $\alpha = [\alpha_0, \alpha_1]$ models the decay of the distortion effect.

Note that as we can see later in Section 2.2.1 and 2.2.3, the author added some images (as claimed, four is enough) of an empty wall for calibration and the purpose of doing this, is to ensure that all pixel in the depth images can be sampled to estimate the coefficients $D_\delta(u, v)$.

(2) **Scaled inverse**: using the following equation to obtain the estimated depth value:

$$z_d = \frac{1}{c_1 d_k + c_0} \tag{6}$$

where c_0 and c_1 are part of the depth camera intrinsic parameters to be calibrated and d_k is the undistorted disparity (i.e. the depth value after distortion correction). In the open-source code released by the author, $c_0 = 3.0938$ and $c_1 = -0.0028$.

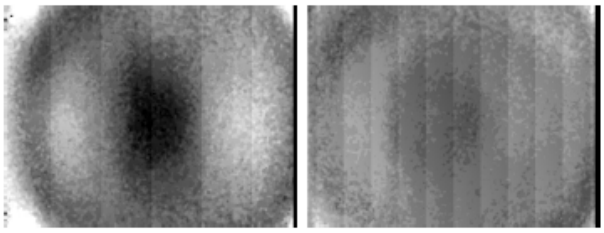


Figure 1: Error residuals (kdu) without distortion correction of a plane at 0.56m (left) and 1.24m (right).

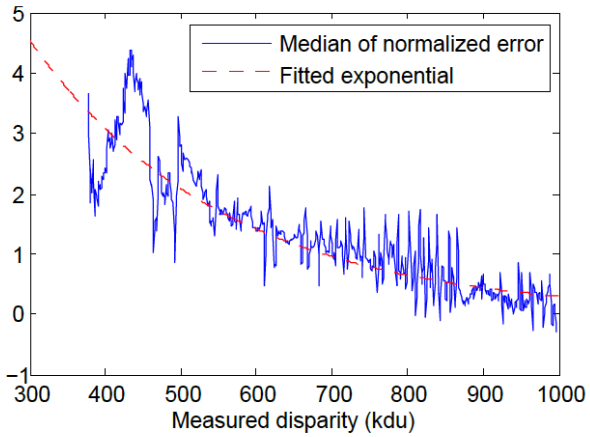


Figure 2: Distortion magnitude with increasing disparity.

Equations (5) and (6) are used when transforming from measured disparity to depth, which is known as backward model. In order to calculate the reprojection error, the forward model, is also needed by calculating the inverse of Eqs. (5) and (6). Thus, the inverse of Eq. (6) is quite straightforward:

$$d_k = \frac{1}{c_1 z_d} - \frac{c_0}{c_1} \tag{7}$$

While the inverse of Eq. (5) is more mathematical involved because of the exponential. The author isolated exponential product by using two variable substitutions:

$$
\begin{aligned}
d_k &= d + D_\delta(u,v) \cdot \exp(\alpha_0 - \alpha_1 d) \\
y &= \exp(\alpha_0 - \alpha_1 d_k + \alpha_1 D_\delta(u,v) y) \\
&\text{where, } \mathrm{y} = \frac{\mathrm{d_k - d}}{\mathrm{D_\delta(u,v)}} \\
y &= \exp(\alpha_1 D_\delta(u,v) y) \exp(\alpha_0 - \alpha_1 d_k) \\
\frac{-\widetilde{y}}{\alpha_1 D_\delta(u,v)} &= \exp(-\widetilde{y}) \exp(\alpha_0 - \alpha_1 d_k) \\
&\text{where, } \widetilde{y} = -y\alpha_1 D_\delta(u,v) \\
\widetilde{y} \exp(\widetilde{y}) &= -\alpha_1 D_\delta(u,v) \exp(\alpha_0 - \alpha_1 d_k) \qquad (8)
\end{aligned}
$$

The product can be solved using the Lambert W function [2] due to that the Lambert W function is the solution to the relation $W(z) = \exp(W(z) = z)$.

Same with color camera, the author described the above model of depth camera by $\mathcal{L}_d = \{\mathrm{f}_d, \mathrm{p}_{0d}, \mathrm{k}_d, c_0, c_1, D_\delta, \alpha\}$ where the last four parameters are used to transform disparity to depth values.

1.3 Extrinsics and Relative Pose

In their paper, the author uses Kinect and one external color camera (with high resolution). Also, in this report, we refer to the color camera in Kinect as "color camera" while the external color camera as "external camera". To calibrate these three cameras, their extrinsics and relative pose are essential to know. Here, the author denotes the rigid transformation from one reference frame to another as $\mathcal{T} = \{\mathrm{R}, \mathrm{t}\}$, where R indicates the rotation matrix and t indicates the translation vector. Take a point x_w from world coordinates $\{W\}$ as an example, its transformation to color camera coordinates $\{C\}$ follows $\mathrm{x}_c = {}^W\mathrm{R}_C \mathrm{x}_w + {}^W\mathrm{t}_C$, where the rotation transformation from $\{W\}$ to $\{C\}$ is denoted as ${}^W\mathrm{R}_C$, and ${}^W\mathrm{t}_C$ for translation transformation.

Fig. 3 shows different references frames and corresponding transformations. Reference frame $\{V_i\}$ is actually the calibration plane of image (checkerboard) i. As we can see from Fig. 3, there are two relative poses: ${}^D\mathcal{T}_C$ and ${}^E\mathcal{T}_C$, both of which are poses to the color camera in Kinect. Also, each image has its own pose (extrinsics) from world to camera, they are ${}^{V_i}\mathcal{T}_D$, ${}^{W_i}\mathcal{T}_C$ and ${}^{W_i}\mathcal{T}_E$. Be aware

that ${}^{V_i}\mathcal{T}_D$ is essentially a pose for image i from reference $\{V\}$ (i..e calibration board) to depth camera. Later we can see in Section 2.2.1, to calculate ${}^{W_i}\mathcal{T}_E$, we only need to specify the area of calibration plane in the initialization stage. Besides, by design, the table and the checkerboard are coplanar but the full transformation between $\{V\}$and $\{W\}$ is unknown.

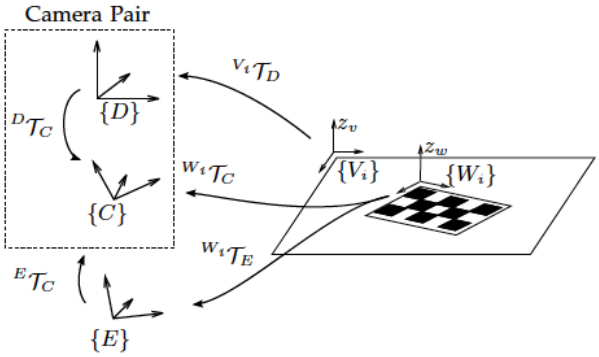


Figure 3: Reference frames and transformations: $\{D\}$, $\{C\}$, and $\{E\}$ are the depth, color, and external cameras. For image i, $\{V_i\}$ is attached to the calibration plane and $\{W_i\}$ is the calibration pattern.

2 Calibration

Figure 4 presents the calibration method in this paper. As separated by the dashed line in Fig. 4, their method can be divided into two parts: (1) Initialization. (2) Non-linear minimization.

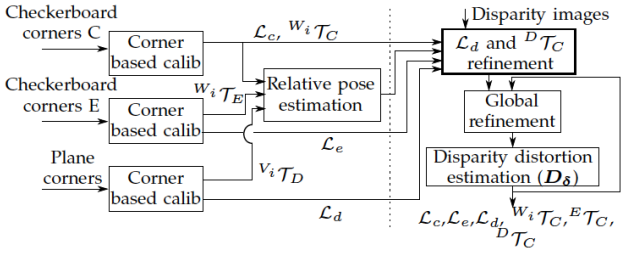


Figure 4: Calibration algorithm. Before dashed line: initialization. After dashed line: non-linear minimization.

2.1 Initialization

The initialization stage of calibration in their paper includes the initialization for three cameras: color camera, depth camera and external camera, respectively.

2.1.1 Initialization for color cameras

The problem of the calibration for color camera has been well studied. In their paper, they use Zhang's [3] method to initialize the two color cameras, which was done by the following steps:

(1) Anchor a checkerboard pattern to the calibration plane.

(2) Extract the corners of checkerboard automatically using the code from [4].

(3) Get the ground truth coordinates of corners with square size 4 cm, x axis inner corner number 10, and y-axis inner corner number 7 (depending on your checkerboard pattern).

(4) Estimate initial homography matrix H for each image. See Appendix A for details.

(5) Once homography matrix is estimated, both intrinsic and extrinsic parameters for each image (denoted as $\{\mathrm{A}, \mathrm{R}_i, \mathrm{t}_i\}$ for image i) are readily computed. See Appendix B for details.

(6) Suppose we are given n color images and there are m points on the model plane. By assuming that all the image points are corrupted by independent and identically distributed noise, the maximum likelihood estimate can be obtained by minimizing the following functional:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\left\|\mathrm{m}_{ij}-\widehat{\mathrm{m}}(\mathrm{A},\mathrm{k}_c,\mathrm{R}_i,\mathrm{t}_i,\mathrm{M}_j)\right\|^2 \tag{9}$$

where M_j is the ground truth coordinates of point j on checkerboard. $\widehat{\mathrm{m}}(\mathrm{A},\mathrm{R}_i,\mathrm{t}_i,\mathrm{M}_j)$ is the projection of point M_j in image i according to Eq. (3), and distortion coefficients k_c are introduced and initialized as $\mathrm{k}_c = [0,0,0,0,0]$. This minimization problem can be solved with the Levenberg-Marquardt Algorithm.

(7) After maximum likelihood estimation, we will invert $\{\mathrm{R}_i, \mathrm{t}_i\}$ if any translation vector t_i for image i is negative to ensure that camera always points forward.

(8) Set the color camera as the reference camera, i.e., ${}^{C}\mathcal{T}_C = \{{}^{C}\mathrm{R}_C, {}^{C}\mathrm{t}_C\}$, with

$${}^{C}\mathrm{R}_C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, {}^{C}\mathrm{t}_C = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

(9) For each image i taken by the external camera, calculate its relative pose to the reference camera (color camera) denoted by ${}^{E_i}T_C = \{{}^{E_i}\mathrm{R}_C, {}^{E_i}\mathrm{t}_C\}$, with

$$\begin{aligned} {}^{E_i}\mathrm{R}_C &= {}^{W_i}\mathrm{R}_C \, {}^{W_i}\mathrm{R}_E^T \\ {}^{E_i}\mathrm{t}_C &= {}^{W_i}\mathrm{t}_C - {}^{E_i}\mathrm{R}_C \, {}^{W_i}\mathrm{t}_E \end{aligned} \tag{10}$$

where, $\{{}^{W_i}\mathrm{R}_C, {}^{W_i}\mathrm{t}_C \,|i = 1, \ldots, n\}$ are the estimated extrinsic parameters for color camera, while $\{{}^{W_i}\mathrm{R}_E, {}^{W_i}\mathrm{t}_E \,|i = 1, \ldots, n\}$for external camera.

Please note that due to the rotation matrix is a normal matrix, i.e. mathematically $\mathrm{R}^T = \mathrm{R}^{-1}$ while geometrically ${}^{W_i}\mathrm{R}_E^T$ representing a rotation in exactly the opposite direction from ${}^{W_i}\mathrm{R}_E$, therefore we can say ${}^{W_i}\mathrm{R}_E^T = {}^{E}\mathrm{R}_{W_i}$.

(10) By design, the color camera and external camera are bind together, therefore, the relative pose of ${}^{E}T_C = \{{}^{E}\mathrm{R}_C, {}^{E}\mathrm{t}_C\}$ is constant, which means we are able to get the mean value of the relative poses from external camera to the reference camera by all images, and treat it as the relative pose of external camera, i.e.,

$$\begin{aligned} {}^{E}\mathrm{R}_C &= \frac{\sum_{i=1}^{n} {}^{E_i}\mathrm{R}_C}{n} \\ {}^{E}\mathrm{t}_C &= \frac{\sum_{i=1}^{n} {}^{E_i}\mathrm{R}_C}{n} \end{aligned} \tag{11}$$

(11) Perform a joint calibration between color camera and external camera. During calibration, extrinsic parameters ${}^{W_i}T_E = \{{}^{W_i}\mathrm{R}_E, {}^{W_i}\mathrm{t}_E\}$ for image i from external camera are calculated with $\{{}^{W_i}\mathrm{R}_C, {}^{W_i}\mathrm{t}_C\}$ and relative pose $\{{}^{E}\mathrm{R}_C, {}^{E}\mathrm{t}_C\}$ that we just obtained:

$$\begin{aligned} {}^{W_i}\mathrm{R}_E &= {}^{E}\mathrm{R}_C^T \, {}^{W_i}\mathrm{R}_C \\ {}^{W_i}\mathrm{t}_E &= {}^{E}\mathrm{R}_C^T ({}^{W_i}\mathrm{t}_C - {}^{E}\mathrm{t}_C) \end{aligned} \tag{12}$$

Then, same with what we did in Eq. (9), by using Levenberg-Marquardt Algorithm jointly with costs calculated for both color camera and external camera, the refined parameters $\{\mathrm{A}, \mathrm{k}_c, \mathrm{R}_i, \mathrm{t}_i \,|i = 1, \ldots, n\}$ for the two cameras are obtained.

2.1.2 Initialization for depth camera

As denoted in Section 1.2, the model of depth camera is described with $\mathcal{L}_d = \{\mathrm{f}_d, \mathrm{p}_{0d}, \mathrm{k}_d, c_0, c_1, D_\delta, \alpha\}$. For each parameter in $\mathcal{L}_d$, the author initialized their

values as:

$$
\begin{aligned}
\mathrm{f}_d &= \begin{bmatrix} 590 & 590 \end{bmatrix} \\
\mathrm{p}_{0d} &= \begin{bmatrix} 320 & 230 \end{bmatrix} \\
\mathrm{k}_d &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
c_0 &= 3.0938 \\
c_1 &= -0.0028 \\
D_\delta &= \mathbf{0}_{480,640}, \text{ with } \sum_{j=1}^{640}\sum_{i=1}^{480} a_{ij} = 0 \\
\alpha &= \begin{bmatrix} 1 & 1 \end{bmatrix}
\end{aligned}
$$

as well as the relative pose ${}^{D}\mathcal{T}_C = \left[{}^{D}\mathrm{R}_C, {}^{D}\mathrm{t}_C\right]$ for depth camera to reference camera:

$$
{}^{D}\mathrm{R}_C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{D}\mathrm{t}_C = \begin{bmatrix} -0.025 \\ 0 \\ 0 \end{bmatrix}
$$

2.2 Non-linear Minimization

Basically, the non-linear minimization part of their method includes four steps: (1) Obtain samples from disparity images. (2) Refine L_d and ${}^{D}\mathcal{T}_C$. (3) Refine D_δ independently, (4) Joint minimization by repeating global refinement until certain criterion has been met.

2.2.1 Samples from disparity

By asking user to manually select the corners of calibration plane for each image, we can get a corresponding mask. Later, all the points overlaid within mask from disparity image are sampled with no more than 60000 samples by default. The sampling method differs while the author simply chose to sample with an increasing step size until the number of points not exceed 60000.

In order to estimate the coefficients $D_\delta(u, v)$ to apply the distortion model for depth camera, the user is required to take several images which covers the whole flat plane by depth camera (a wall is an ideal object, therefore, we call them "wall images") with different distances. In their demo, the author obtained 16 images, and they also claimed in paper that 4 images are sufficient to calibrate with distortion. The season to take images from multiple distances is to ensure that all pixel in the depth images can be sampled to estimate $D_\delta(u, v)$.

Please note that all disparity images including wall images must be sampled before processed.

2.2.2 Relative pose of reference to depth

The initialization gives a very rough guess of the depth camera parameters and relative pose, whereas the color camera parameters have fairly good initial values. Thus, the first step in non-linear minimization as shown in Fig. 4 optimizes only L_d and ${}^{D}\mathcal{T}_{C}$ with all other parameters fixed.

The relative pose between the external and color cameras can be obtained directly because their pose with respect to the same reference frame $\{W\}$ is known. For the depth camera, however, only the pose with respect to $\{V\}$ is known, which is not aligned to $\{W\}$.

To obtain the relative pose ${}^{C}\mathcal{T}_{D}$ we take advantage of the fact that $\{V\}$ and $\{W\}$ are coplanar by design (see Fig. 3). We extract the calibration plane equation in both depth camera and reference camera and use it as a constraint. We define the calibration plane using the equation $\mathrm{N}^{T}\mathrm{x} - d = 0$ where N is the unit normal and d is the distance to the origin.

If we divide a rotation matrix into its colums $\mathrm{R} = [\mathrm{r}_1, \mathrm{r}_3, \mathrm{r}_3]$ and choose the parameters of the plane for both frames in world coordinates as $\mathrm{N} = [0, 0, 1]^{T}$ and $d = 0$ (Since we only care about depth and define the center of series of plane points as origin), therefore, the plane parameters in camera coordinates are:

$$\mathrm{N} = \mathrm{r}_3 \quad \text{and} \quad d = \mathrm{r}_3^{T}\mathrm{t} \tag{13}$$

where we are using ${}^{W_i}\mathcal{T}_{C} = \{{}^{W_i}\mathrm{R}_{C}, {}^{W_i}\mathrm{t}_{C}\}$ and ${}^{V_i}\mathcal{T}_{D} = \{{}^{V_i}\mathrm{R}_{D}, {}^{V_i}\mathrm{t}_{D}\}$ for both frames.

2.2.3 Extrinsics of depth images

For all the depth images, their extrinsic parameters ${}^{V}\mathcal{T}_{D} = \{{}^{V}\mathrm{R}_{\mathrm{D}}, {}^{V}\mathrm{t}_{\mathrm{D}}\}$ are readily computed to support calculation of expected plane depth, which will be introduced in Section 2.2.6:

(1) Since the points extracted from depth images are in raw kdu unit, they must be converted to depth first (see Section 1.2). Given an distorted disparity image coordinates $\mathrm{p}_d = [u_d, v_d]^{T}$ and its disparity value d_k, we are unable to estimate undistorted one d (see Eq. (8)) as D_δ was initialized with $\mathbf{0}_{480,640}$. Thus, the author treat the raw disparity value undistorted and directly convert it to depth z_d by Eq. (6).

(2) Trace optical ray directions of every depth point by reversing Eq. (4):

$$\mathrm{x}_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} (u_d - u_{0d})/f_{dx} \\ (v_d - v_{0d})/f_{dy} \end{bmatrix} \tag{14}$$

where $\mathrm{x}_n = [x_n, y_n]^T$ is the normalized point coordinates (see Section 1.1).

(3) For each depth image i, convert all its points from depth image coordinates to world coordinates $\mathrm{x}_w = [x_w, y_w, z_w]^T$ with the initialized relative pose ${}^D\mathcal{T}_C = \left[{}^D\mathrm{R}_C, {}^D\mathrm{t}_C\right]$, i.e.,

$$\mathrm{x}_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = {}^D\mathrm{R}_C \begin{bmatrix} x_n z_d \\ y_n z_d \\ z_d \end{bmatrix} + {}^D\mathrm{t}_C = \begin{bmatrix} x_n z_d - 0.025 \\ y_n z_d \\ z_d \end{bmatrix} \tag{15}$$

(4) Find the best fit plane for the set of points in world coordinates using PCA. The underlying principle is that "by performing PCA on world coordinates, the coefficients for the first two principal components define vectors that form a basis for the plane. The third principle component is orthogonal to the first two, and its coefficients define the normal vector of the plane. Because the first two components explain as much of the variance in the data as is possible with two dimensions, the plane is the best 2-D linear approximation to the data" (From http://www.mathworks.com/help/stats/examples/fitting-an-orthogonal-regression-using-principal-components-analysis.html). The normal is denoted as N, therefore, its distance from origin can be easily calculated by

$$d = \mathrm{N} \cdot \bar{\mathrm{x}}_w \tag{16}$$

where, $\bar{\mathrm{x}}_w$ denotes the center of all world coordinates x_w.

(5) With the calibration plane parameters (normal N and distance d from origin), we are able to extract the extrinsic parameters ${}^V T_D = \{\mathrm{R}, \mathrm{t}\}$ with

$$\begin{aligned} &\mathrm{R} = \left[\ \mathrm{v} \times \mathrm{N} \quad \mathrm{v} \quad \mathrm{N}\ \right] \\ &\text{where, } \mathrm{v}_{\mathrm{N}_1} = 0, \mathrm{v}_{\mathrm{N}_3} = \frac{\mathrm{N}_2}{\sqrt{\mathrm{N}_2^2 + \mathrm{N}_3^2}}, \mathrm{v}_{\mathrm{N}_2} = \sqrt{1 - \mathrm{v}_{\mathrm{N}_3}^2} \\ &[\mathrm{N}_1\ \mathrm{N}_2\ \mathrm{N}_3] \text{ are entries of N in decreased order.} \end{aligned} \tag{17}$$

and t with

$$\mathrm{t}_{\mathrm{N}_1} = \frac{d}{\mathrm{N}_1} \tag{18}$$

with other two entries are both zero. It means that only the third column of the rotation and one entry of the translation are constrained by the calibration plane parameters while the other values are arbitrary. Please refer to Section 2.2.2 for details on theoretical side.

(6) Solve the non-linear minimization problem with input N, d and cost function as well calculating error between measured disparity and reprojected disparity (see Section 2.2.4 for details) through Levenberg-Marquardt Algorithm. Finally, a refined N, d so that refined ${}^{V_i}T_D = \{{}^{V_i}\mathrm{R}_D, {}^{V_i}\mathrm{t}_D\}$ will come out as the estimated extrinsic parameters for depth image i.

(7) Repeat above steps for all depth images in order to obtain their refined extrinsic parameters.

2.2.4 Joint minimization

As stated by the author, "The calibration method aims to minimize the weighted sum of squares of the measurement reprojection errors over all parameters $L_c, L_d, L_e, {}^{E}T_C, {}^{D}T_C, {}^{W_i}T_C$ for all images i". For the reprojection error calculated for color camera and external camera, they use Euclidean distance between measured corner position (Please refer to step (9) and (11) in Section 2.1.1), whereas for depth camera, they simply use the difference between the measured disparity and predicted disparity (Also, please refer to Section 2.2.6).

Because the errors have different unite,they are weighted using the inverse of corresponding measurement variance $(\sigma_C^2, \sigma_D^2, \sigma_E^2)$ before calibration:

$$c = \frac{\sum \left\| \widehat{m}_C - m_C \right\|^2}{\sigma_C^2} + \frac{\sum \left\| \widehat{d} - d \right\|^2}{\sigma_D^2} + \frac{\sum \left\| \widehat{m}_E - m_E \right\|^2}{\sigma_E^2} \tag{19}$$

where c is the resulting cost function. Due to that above function has highly nonlinear and depends on quite a lot parameters (For instance, calculating $\widehat{d} - d$ requires D_δ which contains $640 \times 480 = 307200$ entries), the author simplified the cost function by separating the optimization of disparity distortion parameters, i.e. calculate the residuals in undistorted disparity space instead of in measured disparity space:

$$c = \frac{\sum \left\| \widehat{m}_C - m_C \right\|^2}{\sigma_C^2} + \frac{\sum \left\| \widehat{d}_k - d_k \right\|^2}{\sigma_D^2} + \frac{\sum \left\| \widehat{m}_E - m_E \right\|^2}{\sigma_E^2} \tag{20}$$

After we optimized only L_d and ${}^{D}\mathcal{T}_C$ in Section 2.2.2, the optimization then continues iteratively with two alternating steps.

(1) Keep D_δ constant and minimize Eq. (19) with Levenberg-Marquardt algorithm over all other parameters, refer to Section 2.2.6 for calculation of $\widehat{d}_k$.

(2) Optimize D_δ independently for each pixel, refer to Section 2.2.5.

Note that the author claimed the initial values of the depth distortion model (α and D_σ) are not critical and initially assuming zero for both has proven to yield accurate results.

2.2.5 Disparity distortion estimation

Firstly, each disparity measurement d is undistorted using Eq. (5). Next, compute predicted disparity $\widehat{d}_k$ (see Section 2.2.6). Then, the following cost function is applied to obtain the distortion parameters:

$$c_d = \sum_{\text{images}} \sum_{u,v} \left(d + D_\delta(u,v) \cdot \exp(\alpha_0 - \alpha_1 d) - \widehat{d}_k \right)^2 \tag{21}$$

The optimal value of each D_σ can be obtained by solving a linear equation because above equation is quadratic in each $D_\delta(u, v)$, while for σ, the author still chose solving by Levenberg-Marquardt algorithm.

Note that due to a great number of disparity points we may get within the calibration plane, all disparity points must be sampled to ensure calculation performance, whereas in their released code, the author randomly sampled at most 70000 points.

2.2.6 Predict disparity

We can perform a backward calculation to estimate the expected depth of calibration plane in undistorted disparity space as follows:

(1) For depth image i with the refined extrinsic parameters ${}^{V_i}\mathcal{T}_D = \{{}^{V_i}\mathrm{R}_D, {}^{V_i}\mathrm{t}_D\}$, because ${}^{D}\mathcal{T}_C = \left[{}^{D}\mathrm{R}_C, {}^{D}\mathrm{t}_C\right]$ is known, the calibration plane parameters with respect to reference camera can be easily estimated using Eq. (13):

$$\mathrm{N_C} = {}^{W_i}\mathrm{R}_{C3}, \quad \text{and} \quad d_C = {}^{W_i}\mathrm{R}_{C3}{}^{W_i}\mathrm{t}_C$$

where, ${}^{W_i}\mathrm{R}_{C3}$ is the 3^{rd} column of ${}^{W_i}\mathrm{R}_C$.

(2) The relative pose from depth camera to reference camera is also straightforward to get:

$$ {}^{C}\mathrm{R}_D = {}^{D}\mathrm{R}_C^T \quad \text{and} \quad {}^{C}\mathrm{t}_D = -{}^{C}\mathrm{R}_D{}^{D}\mathrm{t}_C \tag{22}$$

(3) Project relative translation ${}^{C}\mathrm{t}_D$ upon the unit normal of calibration plane N_C which consists only the third column of rotation matrix ${}^{W_i}\mathrm{R}_C$, i.e., ${}^{C}\mathrm{t}_D^T\mathrm{N}_C$ that is actually the projected translation from depth camera to reference camera in z axis as well the compensated z coordinate due to the displacement of two cameras, with respect to depth camera.

(4) With known distance from calibration plane to origin (center of series of depth points) d_C, we are able to compute how far depth camera is to the calibration plane in terms of z axis, with respect to depth camera, i.e.,

$$d_D = {}^{C}\mathrm{t}_D^T\mathrm{N}_C + d_C \tag{23}$$

where d_D indicates the estimated distance from the origin of calibration plane to depth camera, in depth world coordinate system.

(5) Let $\mathrm{p}_d = [u_d, v_d]^T$ be the depth image coordinates within mask, by using Eq. (13) the optical ray directions of all depth points can be traced, i.e. $\mathrm{x}_n = [x_n, y_n]^T$ while skipping distortion correction for that k_d was initialized as $\mathbf{0}_{1,5}$. Therefore, we are able to obtain m directions by assuming that the calibration plane consists of m points.

(6) Transform unit normal from color world coordinate system to depth world coordinate system by

$$\mathrm{N}_D = {}^{D}\mathrm{R}_C^T \mathrm{N}_\mathrm{C} \tag{24}$$

(7) For each depth point in the calibration plane, using $\mathrm{N}_D^T \tilde{\mathrm{x}}_n$ (where $\tilde{\mathrm{x}}_n$ is the homogeneous form of x_n) to project the "shifted" unit normal which just turned to the view of depth camera, onto the optical ray direction of current point. Geometrically, $\mathrm{N}_D^T \tilde{\mathrm{x}}_n$ measures the "degree of spread" for current point to origin: the higher $\mathrm{N}_D^T \tilde{\mathrm{x}}_n$ is, the more they are closing to each other in terms of angle difference (and/or the optical ray coming closer to depth camera in terms of magnitude), and thus the closer this point is to depth camera.

(8) For each of m points, let $\widehat{d}_k$ denote the depth of current point in depth world coordinates, therefore, by taking the degree of spread into account, we have

$$\widehat{d}_k = \frac{d_D}{\mathrm{N}_D^T \tilde{\mathrm{x}}_n} \tag{25}$$

Note that by diving d_D with the denominator above essentially performs a reverse to normalization.

3 Evaluation

The author captured three data sets (A1, A2, and B1). Two of them were captured with the same Kinect (A1 and A2) and one with a different Kinect (B1). For each set, the captured images were divided into calibration and validation groups with 60 and 14 images respectively. The calibration images were used to estimate all the parameters in the model, then the intrinsic parameters were kept fixed to estimate only the rigs pose (i.e. ${}^{W}\mathcal{T}_C$) for the validation images. All results presented here were obtained from the validation images.

Normally, the different calibrations (A1, A2, and B1) would produce slightly different error variances σ_C^2, σ_D^2, and σ_E^2. To compare the data sets the variances were kept constant $\sigma_C^2 = 0.18$px, $\sigma_D^2 = 0.9$kdu, and $\sigma_E^2 = 0.30$px.

3.1 Accuracy

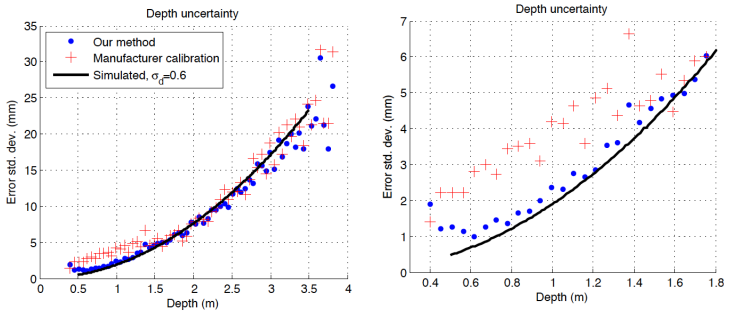
One highlight of this paper is the introduction of disparity distortion correction. The same idea was proposed by Smisek et al. [5]. Therefore, it is essential to compare the performance under both groups with correction and with no correction. As shown in Fig. 5 (table, actually), for each of the three cameras, the author compared performance with Smisek's method [5], and a no correction one of theirs.

		Color ±0.02 px	Depth ±0.002 kdu	External ±0.05 px
A1	No correction	0.42	1.497	0.83
	Smíšek [13]	0.32	1.140	0.72
	Our method	0.28	0.773	0.64
A2	No correction	0.36	1.322	0.83
	Smíšek [13]	0.33	0.884	0.85
	Our method	0.38	0.865	0.79
B1	No correction	0.56	1.108	0.97
	Smíšek [13]	0.62	1.300	0.91
	Our method	0.57	0.904	0.85

Figure 5: Calibration with different distortion models. Std.deviation of residuals with a 99% confidence interval.

3.2 Comparison with Manufacturer Calibration

The best way to prove practical application of their method is to compare with manufacturer calibration. The drivers provided by the manufacturer (Primesense) use factory calibrated settings to convert the disparity measurements to 3D points. They used these calibration parameters and compared manufacturer's performance to that of their calibration. They use A1 set to test performance and A2 set for calibration to avoid any bias. The error measurements are shown in Fig. 6 for both calibrations. The measurements were grouped by depth in 64 bins from 0.4m to 3.7m. For each bin, the standard deviation of the error was plotted.



(a) Full calibrated range (b) Zoom in

Figure 6: Measurement uncertainty for varying depths.

A Calculating homography

Here, I am introducing part of Zhang's method [3] on obtaining homography. Please note that the author uses different estimation method in Appendix A.2 from Zhang's.

A.1 Basic

Given a world 3D point $\mathrm{M} = [X, Y, Z]^T$ and its projection (2D point) to camera image coordinates $\mathrm{m} = [u, v]^T$, we use their homogeneous form $\widetilde{\mathrm{M}} = [X, Y, Z, 1]^T$ and $\widetilde{\mathrm{m}} = [u, v, 1]^T$ by adding 1 as the last element. Assuming in model plane of world coordinate system $Z = 0$, therefore, we have:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathrm{A} \begin{bmatrix} \mathrm{r}_1 & \mathrm{r}_2 & \mathrm{r}_3 & \mathrm{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \mathrm{A} \begin{bmatrix} \mathrm{r}_1 & \mathrm{r}_2 & \mathrm{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

where, s is an arbitrary scale factor and rotation matrix R and translation vector t are extrinsics, while A is called the intrinsic. This time, $\mathrm{M} = [X, Y]^T$ since Z is always equal to 0. In turn, $\widetilde{\mathrm{M}} = [X, Y, 1]^T$. Therefore, we can use homography to transform world coordinates $\widetilde{\mathrm{M}} = [X, Y, 1]^T$ to image coordinates $\widetilde{\mathrm{m}} = [x, y]^T$:

$$s\widetilde{\mathrm{m}} = \mathrm{H}\widetilde{\mathrm{M}}, \quad \text{with} \quad \mathrm{H} = \mathrm{A} \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

Again, the 3×3 matrix H is defined up to a scale factor s.

A.2 Estimation of homography

Let $\mathrm{p} = [X, Y, Z]^T$ be a 3D point in model plane, and its projection to camera image coordinates $\mathrm{m} = [u, v]^T$ that was captured by automatic corner finder. Besides, we denote the ground truth coordinates of corners as $\mathrm{g} = [g_x, g_y]^T$.

A.2.1 Normalization

On checkerboard image k with width w and height h, the set of coordinates m captured by corner finder is denoted as M, and its mean along x direction and y direction can be calculated by

$$\overline{M_x} = \frac{\sum_{j=1}^{h} \sum_{i=1}^{w} u_{ij}}{h \times w}, \quad \overline{M_y} = \frac{\sum_{j=1}^{h} \sum_{i=1}^{w} v_{ij}}{h \times w}$$

and its variance

$$d = \mathrm{var}(M) = \sqrt{\sum_{j=1}^{h} \sum_{i=1}^{w} \left(u_{ij} - \overline{M_x}\right)^2 + \left(v_{ij} - \overline{M_y}\right)^2}$$

In addition, a scale is defined:

$$\alpha = \frac{\sqrt{2}}{d}$$

Then create a normalize matrix:

$$\mathrm{N} = \begin{bmatrix} \alpha & 0 & -\alpha\overline{M_x} \\ 0 & \alpha & -\alpha\overline{M_y} \\ 0 & 0 & 1 \end{bmatrix}$$

which is later multiplied by all the point coordinates captured by the corner finder:

$$\mathrm{P}_n = \mathrm{N} \begin{bmatrix} u_{11} & u_{12} & \dots & u_{hw} \\ v_{11} & v_{12} & \dots & v_{hw} \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \alpha\left(u_{11} - \overline{M_x}\right) & \alpha\left(u_{12} - \overline{M_x}\right) & \dots & \alpha\left(u_{hw} - \overline{M_x}\right) \\ \alpha(v_{11} - \overline{M_y}) & \alpha(v_{12} - \overline{M_y}) & \dots & \alpha(v_{hw} - \overline{M_y}) \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

where, P_n is the set of point coordinates that have been normalized based on variance and means from original coordinates.

Furthermore, the author did the same to normalize ground truth data, and the set of normalized coordinates is denoted as G_n.

Note that the above normalization prevents the offsets of checkerboard corner coordinates and helps normalize them based on variance, which gives good metric in estimation of homograph.

A.2.2 Nonlinear minimization

Due to the measurement noise, the model points M and image points m do not satisfy Eq. (3). In Zhang's paper, he presented a technique based on maximum likelihood criterion to estimate the homography. By assuming (which is important in his idea) all points in image i are corrupted by Gaussian noise with mean 0 and covariance matrix Λ_{m_i}, the maximum likelihood estimation of homography H can be obtained by minimizing the following functional:

$$\sum_i (\mathrm{m}_i - \hat{\mathrm{m}}_i)^T \Lambda_{\mathrm{m}_i}^{-1} (\mathrm{m}_i - \hat{\mathrm{m}}_i)$$

$$\text{where, } \hat{\mathrm{m}}_\mathrm{i} = \frac{1}{\bar{\mathrm{h}}_3^\mathrm{T}} \begin{bmatrix} \bar{\mathrm{h}}_1^T \mathrm{M}_i \\ \bar{\mathrm{h}}_2^T \mathrm{M}_i \end{bmatrix}, \quad \text{with } \bar{\mathrm{h}}_i, \text{ the } i^{\mathrm{th}} \text{ row of H}$$

In practice, they assume $\Lambda_{m_i} = \sigma^2 \mathrm{I}$ for all image i, cause it is reasonable if points are extracted independently with the same procedure. Therefore, the above problem becomes a nonlinear least-square one, i.e., $\min_\mathrm{H} \sum_i \|\mathrm{m}_i - \hat{\mathrm{m}}_i\|^2$. In their paper, Zhang preferred to conduct the nonlinear minimization via Levenberg-Marquardt Algorithm, which requires an initial guess as introduced in the next section.

A.2.3 Initial guess

Let $\mathrm{x} = [\bar{\mathrm{h}}_1^T, \bar{\mathrm{h}}_2^T, \bar{\mathrm{h}}_3^T]$, Zhang constrained x based on Eq. (3) via:

$$\begin{bmatrix} \widetilde{\mathrm{M}}^T & 0^T & -u_0\widetilde{\mathrm{M}}^T \\ 0^T & \widetilde{\mathrm{M}}^T & -\upsilon\widetilde{\mathrm{M}}^T \end{bmatrix} \mathrm{x} = 0$$

where $\widetilde{\mathrm{M}}^T$ is the homogeneous form of M^T. When we are given n points, we have n above equations, which can be written in matrix equation as $\mathrm{Lx} = 0$, where L is a $2n \times 9$ matrix. As x is defined up to a scale factor, the solution is well known to be the right singular vector of L associated with the smallest singular value (or equivalently,the eigenvector of $\mathrm{L}^T\mathrm{L}$ associated with the smallest eigenvalue).

However, in their released code, the author constrained x by:

$$\begin{bmatrix} 0^T & -\mathrm{P}_z\mathrm{G}_n^T & \mathrm{P}_x\mathrm{G}_n^T \\ \mathrm{P}_z\mathrm{G}_n^T & 0^T & -\mathrm{P}_y\mathrm{G}_n^T \end{bmatrix} \mathrm{x} = 0$$

where, P_x and P_y are values of x, y coordinates for one point of P_n denoted in Appendix A.2.1. Note that the above constraints are derived as follows:

$$\begin{cases} \mathrm{P}_z\widehat{\mathrm{P}}_y = \mathrm{P}_x\widehat{\mathrm{P}}_z \\ \mathrm{P}_z\widehat{\mathrm{P}}_x = \mathrm{P}_y\widehat{\mathrm{P}}_z \end{cases} \Rightarrow \begin{cases} \mathrm{P}_z(\bar{\mathrm{h}}_2\mathrm{G}_n) = \mathrm{P}_x(\bar{\mathrm{h}}_3\mathrm{G}_n) \\ \mathrm{P}_z(\bar{\mathrm{h}}_1\mathrm{G}_n) = \mathrm{P}_y(\bar{\mathrm{h}}_3\mathrm{G}_n) \end{cases}$$
$$\Rightarrow \begin{cases} (\mathrm{P}_z\mathrm{G}_n^T)\bar{\mathrm{h}}_2^T = (\mathrm{P}_x\mathrm{G}_n^T)\bar{\mathrm{h}}_3^T \\ (\mathrm{P}_z\mathrm{G}_n^T)\bar{\mathrm{h}}_1^T = (\mathrm{P}_y\mathrm{G}_n^T)\bar{\mathrm{h}}_3^T \end{cases}$$

where, $\widehat{\mathrm{P}} = [\widehat{\mathrm{P}}_x, \widehat{\mathrm{P}}_y, \widehat{\mathrm{P}}_z]^T$ indicates the image coordinates transformed from ground truth 3D point G_n with homography x.

A.2.4 Calculate homography

The above $\mathrm{Lx} = 0$ form is solved by calculating the eigenvector of $\mathrm{L}^T\mathrm{L}$ associated with the smallest eigenvalue. Moreover, due to the homography x is measured with different metrics in captured image coordinates and ground truth coordinates (where the result of solution to $\mathrm{Lx} = 0$ is within), i.e.,

$$\mathrm{N}_p\mathrm{x}_p = \mathrm{N}_{\mathrm{gt}}\mathrm{x}_{\mathrm{gt}}$$

where, N_p and N_{gt} are normalization matrix for coordinates of captured corner points and ground truth points, respectively. x_p denotes homography in metric of image coordinates and x_{gt} of ground truth coordinates. Therefore, we have

$$\mathrm{x}_p = \mathrm{N}_p^{-1}\mathrm{x}_{\mathrm{gt}}\mathrm{N}_{\mathrm{gt}}$$

The author transformed the measurement unit from two different metrics to obtain more accurate homography based on the above equation.

Finally, a normalized homography x_n is obtained by:

$$\mathrm{x}_n = \frac{\mathrm{x}_p}{|\mathrm{h}_1|}$$

where, $|\mathrm{h}_1|$ is the l^2-norm of first column of x_p.

B Computing camera parameters

Zhang's paper has elaborated the computation of camera parameters (for both intrinsic and extrinsic ones) based on estimated homography:

B.1 Closed-form solution

Here, we use the same notations with Appendix A.1.

Let

$$\begin{aligned}\mathrm{B} =&\mathrm{A}^{-T}\mathrm{A}^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{c}{\alpha^2\beta} & \frac{cv_0-u_0\beta}{\alpha^2\beta} \\ -\frac{c}{\alpha^2\beta} & \frac{c^2}{\alpha^2\beta^2}+\frac{1}{\beta^2} & -\frac{c(cv_0-u_0\beta)}{\alpha^2\beta^2}-\frac{v_0}{\beta^2} \\ \frac{cv_0-u_0\beta}{\alpha^2\beta} & -\frac{c(cv_0-u_0\beta)}{\alpha^2\beta^2}-\frac{v_0}{\beta^2} & \frac{(cv_0-u_0\beta)^2}{\alpha^2\beta^2}+\frac{v_0^2}{\beta^2}+1 \end{bmatrix}\end{aligned}$$

Note that B is symmetric, defined by a 6D vector

$$\mathrm{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T$$

Let the i^{th} column vector of H be $\mathrm{h}_i = [h_{i1}, h_{i2}, h_{i3}]^T$. Then, we have

$$\begin{aligned}\mathrm{h}_i^T\mathrm{B}\mathrm{h}_j &= v_{ij}^T\mathrm{b} \\ \text{with, } v_{ij} &= [h_{i1}h_{j1}, h_{i1}h_{j2}+h_{i2}h_{j1}, h_{i2}h_{j2}, \\ &\quad h_{i3}h_{j1}+h_{i1}h_{j3}, h_{i3}h_{j2}+h_{i2}h_{j3}, h_{i3}h_{j3}]^T\end{aligned}$$

Let's denote homography by $\mathrm{H} = [\mathrm{h}_1\ \mathrm{h}_2\ \mathrm{h}_3]$, thus, we have

$$\begin{bmatrix} \mathrm{h}_1 & \mathrm{h}_2 & \mathrm{h}_3 \end{bmatrix} = \lambda\mathrm{A}\begin{bmatrix} \mathrm{r}_1 & \mathrm{r}_2 & \mathrm{t} \end{bmatrix}$$

where λ is an arbitrary scalar. Using the knowledge that r_1 and r_2 are orthonormal, we have

$$\begin{aligned}&\mathrm{h}_1^T\mathrm{A}^{-T}\mathrm{A}^{-1}\mathrm{h}_2=0 \\ &\mathrm{h}_1^T\mathrm{A}^{-T}\mathrm{A}^{-1}\mathrm{h}_1=\mathrm{h}_2^T\mathrm{A}^{-T}\mathrm{A}^{-1}\mathrm{h}_2\end{aligned}$$

The above are two basic constraints on the intrinsic parameters, given one homography H. Therefore, the two fundamental constraints given homography, can be rewritten as 2 homogeneous equations in b:

$$\begin{bmatrix} \mathrm{v}_{12}^T \\ (\mathrm{v}_{11} - \mathrm{v}_{22})^T \end{bmatrix} \mathrm{b} = 0$$

If n images of the model plane are observed, by stacking n such equations as above we have

$$\mathrm{Vb} = 0$$

where V is a $2n \times 6$ matrix, If $n \geq 3$, we will have in general a unique solution b defined up to a scale factor. If $n = 2$, we can impose the skewless constraint $c = 0$, i.e., $[0, 1, 0, 0, 0, 0]\,\mathrm{b} = 0$, which is added as an additional equation to $\mathrm{Vb} = 0$. The solution to above equation is well known as the eigenvector of $\mathrm{V}^T\mathrm{V}$ associated with the smallest eigenvalue (equivalently, the right singular vector of V associated with the smallest singular value).

B.2 Intrinsic parameters

Once b is estimated, the camera intrinsic matrix A is readily computed without difficulty. As defined in Appendix B.1, the matrix B is defined up to a scalar, i.e., $\mathrm{B} = \mathrm{A}^{-T}\mathrm{A}^{-1}$ with λ an arbitrary scale, thus, we can uniquely extract the intrinsic parameters from matrix B as follows:

$$\begin{aligned}
\upsilon_0 &= (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \\
\lambda &= B_{33} - [B_{13}^2 + \upsilon_0(B_{12}B_{13} - B_{11}B_{23})]/B_{11} \\
\alpha &= \sqrt{\lambda/B_{11}} \\
\beta &= \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \\
c &= -B_{12}\alpha^2\beta/\lambda \\
u_0 &= c\upsilon_0/\alpha - B_{13}\alpha^2/\lambda
\end{aligned}$$

B.3 Extrinsic parameters

From

$$s\widetilde{\mathrm{m}} = \mathrm{H}\widetilde{\mathrm{M}}, \quad \text{with} \quad \mathrm{H} = \mathrm{A}\begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

referred in Appendix A.1., we have

$$\mathrm{r}_1 = \lambda\mathrm{A}^{-1}\mathrm{h}_1, \mathrm{r}_2 = \lambda\mathrm{A}^{-1}\mathrm{h}_2, \mathrm{r}_3 = \lambda\mathrm{A}^{-1}\mathrm{h}_3, \mathrm{t} = \lambda\mathrm{A}^{-1}\mathrm{h}_3$$

with $\lambda = 1/\left\|\mathrm{A}^{-1}\mathrm{h}_1\right\| = 1/\left\|\mathrm{A}^{-1}\mathrm{h}_2\right\|$.

Note that the author used SVD(Singular Value Decomposition) in their code on the matrix R to determine the orthogonal matrix O closest to R. The solution is known as the product UV^*.

References

[1] J. Heikkila. Geometric camera calibration using circular control points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10):1066–1077, Oct 2000.

[2] D.A Barry, J.-Y Parlange, L Li, H Prommer, C.J Cunningham, and F Stagnitti. Analytical approximations for real values of the lambert w-function. *Mathematics and Computers in Simulation*, 53(12):95 – 103, 2000.

[3] Zhengyou Z. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 666–673 vol.1, 1999.

[4] M. Rufli, D. Scaramuzza, and R. Siegwart. Automatic detection of checkerboards on blurred and distorted images. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3121–3126, Sept 2008.

[5] Jan Smisek, Michal Jancosek, and Tomas Pajdla. 3d with kinect. In *Consumer Depth Cameras for Computer Vision*, pages 3–25. Springer, 2013.