CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

Department of Cybernetics

RGBD Kinect Camera Calibration

Bachelor Thesis

Study Programme: Cybernetics and Robotics
Branch of study: Robotics

Thesis advisor: Ing. Tomáš Pajdla, Ph.D.

Vojtěch Cvrček

Prague 2014

RGBD Kinect Camera Calibration

Vojtěch Cvrček

May 2014

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 23. května 2014 Vojtěch Cvrček

Acknowledgements

I would like to thank to my bachelor thesis supervisor Tomáš Pajdla for providing aid and hints during my work. I would also like to express gratitude to Daniel Herrera for publishing his toolbox. Finally, I would like to thank my family for their support during my work on the thesis.

Abstrakt

V této práci jsem zkoumal možnosti kalibrace RGBD kamery. Uvádím novou kalibrační metodu, která najednou kalibruje barevnou kameru, hloubkovou kameru, IR kameru a jejich vzájemnou orientaci. Nová metoda je založena na předchozích pracích. Soustředil jsem se na zvýšení přesnosti a robustnosti v porovnání se zkoumanými metodami. Součástí bakalářské práce je porovnání výsledků kalibračních metod. Výsledná kalibrace splňuje očekávání na reálných i simulovaných datech.

Abstract

I studied possibilities of calibrating RGBD camera in this thesis. I present new calibration method that simultaneously calibrates a color camera, a depth camera, a IR camera, and the relative pose between. The calibration is based on preceding calibrations. I concentrate on making method more accurate and robust than predecessor. Thesis also contains comparison with some of the preceding works. New calibration performs well both on simulated data and real data.

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Vojtěch C v r č e k

Studijní program: Kybernetika a robotika (bakalářský)

Obor: Robotika

Název tématu: Kalibrace RGBD kamery Kinect

Pokyny pro vypracování:

1. Prozkoumejte současné nové metody kalibrace RGBD kamer [2,3].
2. Porovnejte metody [2, 3] s metodou [1] a vysvětlete v čem se liší.
3. Navrhněte a implementujte vylepšení [1] na základě [2,3].
4. Demonstrujte Vaši implementaci na reálných datech.

Seznam odborné literatury:

[1] T. Pajdla, J. Smisek, and M. Jancosek: 3D with Kinect. In Andrea Fossati, Juergen Gall, Grabner Helmut, Xiaofeng Ren, and Kurt Konolige, editors, 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pages 1154-1160, Los Alamitos, USA, November 2011. IEEE Computer Society.
[2] C. Herrera, J. Kannala, H. Heikkila: Joint depth and color camera calibration with distortion correction. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 34(10). 2012.
[3] C. Raposo, J.P. Barreto, U. Nunes: Fast and Accurate Calibration of a Kinect Sensor. International Conference on 3DV, pp 342 - 349, 2013.

Vedoucí bakalářské práce: Ing. Tomáš Pajdla, Ph.D.

Platnost zadání: do konce letního semestru 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 10. 1. 2014

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

BACHELOR PROJECT ASSIGNMENT

Student: Vojtěch C v r č e k

Study programme: Cybernetics and Robotics

Specialisation: Robotics

Title of Bachelor Project: RGBD Kinect Camera Calibration

Guidelines:

1. Review the state of the art in RGBD camera calibration. Focus on new results [2,3] since [1].
2. Compare [2,3] and [1] and explain how they are different.
3. Suggest and implement an improvement of [1] based on [2,3].
4. Demonstrate your implementation on real data.

Bibliography/Sources:

[1] T. Pajdla, J. Smisek, and M. Jancosek: 3D with Kinect. In Andrea Fossati, Juergen Gall, Grabner Helmut, Xiaofeng Ren, and Kurt Konolige, editors, 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pages 1154-1160, Los Alamitos, USA, November 2011. IEEE Computer Society.
[2] C. Herrera, J. Kannala, H. Heikkila: Joint depth and color camera calibration with distortion correction. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 34(10). 2012.
[3] C. Raposo, J.P. Barreto, U. Nunes: Fast and Accurate Calibration of a Kinect Sensor. International Conference on 3DV, pp 342 - 349, 2013.

Bachelor Project Supervisor: Ing. Tomáš Pajdla, Ph.D.

Valid until: the end of the summer semester of academic year 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 10, 2014

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter brief history of Kinect and motivation for my work is presented.

1.1 History

Kinect was introduced as accessory for game console in 2011 by Microsoft, this accessory is basically RGBD camera, i.e. camera capturing both color and depth image. Afterward Kinect gain attention from computer vision community for its performance and became quickly popular. Device uses structured light projected by IR laser projector. Random speckle pattern is emitted by IR projector and captured by IR camera. This approach improved previous approaches, both in quality of reconstruction and decrease of processing time.

1.2 Motivation

Kinect is used for scene reconstruction. Key factor determining the quality of reconstruction is the calibration of device. Basic value of intrinsic parameters are precalibrated in factory and stored in the firmware. Intrinsic parameters approximate values are known [7]. But these values change notably from device to device. There exist many approaches to calibrate Kinect. I mainly concern with approaches which requires basic equipment. Studied approaches are [3],[4] and [6]. Each of these papers is built on a previous one and presents new improve calibration method. They are slight differences between camera models and significant differences between calibration algorithms. I believe that each method has some inconvenient feature. I decided to design a new calibration approach that is improving the calibration quality.

Chapter 2

Background

2.1 Pinhole model of camera

The model of camera is the same for RGB and IR camera (which is same as the depth camera with shifted principal point). Pinhole model of the perspective camera consists of camera center C and image plane π. Pinhole camera model is used in Smisek's [3], Herrera's [4] and Raposo's [6] work. Smisek [3] uses different formalism then Herrera's [4] and Raposo's [6] work that share the same formalism. Smisek et al. [3] present model of 3D point $\mathbf{X}$ projected into image point $\mathbf{u} = [u, v]^T$ in following form:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} x_k \\ y_k \\ 1 \end{bmatrix} \tag{2.1}$$

where K present camera matrix :

$$\mathbf{K} = \begin{bmatrix} f_{cx} & \alpha_c \cdot f_{cx} & p_x \\ 0 & f_{cy} & p_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

f_{cx}, f_{cy} present focal lengths, p_x, p_y is a principal point and α_c encodes the angle between image plane axes.
Coordinates of projection of point $\mathbf{X}$ w.r.t. camera reference frame is marked as $\mathbf{x}$ (Figure 2.1), after normalization and distortion the product is equal to $\mathbf{x}_k = [x_k, y_k]$. The normalization of $\mathbf{x}$ w.r.t. to camera coordinates:

$$\mathbf{x}_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} x_x/z_x \\ y_x/z_x \\ 1 \end{bmatrix} \tag{2.3}$$

now, the distortion:

$$\mathbf{x}_k = (1 + k_1 r^2 + k_2 r^4 + k_5 r^6) \begin{bmatrix} \mathbf{x}_n \\ 0 \end{bmatrix} + \begin{bmatrix} 2k_3 x_n y_n + k_4(r^2 + 2x_n^2) \\ 2k_4 x_n y_n + k_3(r^2 + 2y_n^2) \end{bmatrix} \tag{2.4}$$

$$r^2 = x_n^2 + y_n^2 \tag{2.5}$$

the relation between $\mathbf{X}$ and $\mathbf{x}$ is:

$$\mathbf{x} = \mathbf{R}(\mathbf{X} - \mathbf{t}) \tag{2.6}$$

where **R**, **t** are extrinsic parameters of the camera. The **R** is rotation between world reference frame and camera reference frame. The **t** is translation between same reference frames. The **t** in Equation 2.6 is same as camera center in world reference frame.

The eqs. 2.1 - 2.6 provide geometric model used by Smisek [3]. Smisek [3] also uses the same model for IR camera.

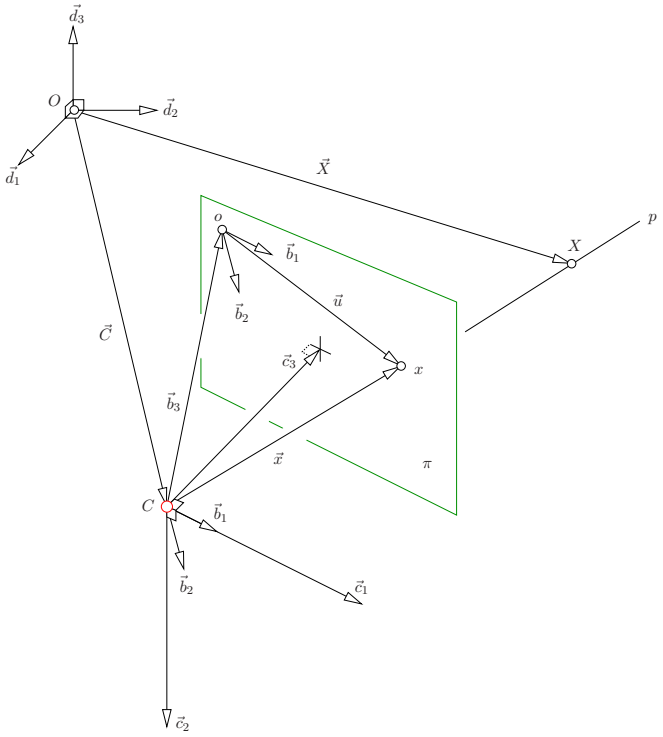

Figure 2.1: Pinhole model(Figure is taken from [1])

2.1.1 Distortion model

Distortion model used in calibrations is a model created by Brown in [14]. This distortion model is used in all three calibration procedure. Distortion of lens is represented by vector $\vec{k}$ where k_1, k_2, k_5 represent radial distortion and k_3, k_4 represent tangential distortion. Meaning of each parameter is visualized in the Figure B.1.

2.1.2 Differences between models of cameras in studied calibration approaches

The presented camera model is the same both for Herrera and Raposo including distortion model, the only difference is the notation.

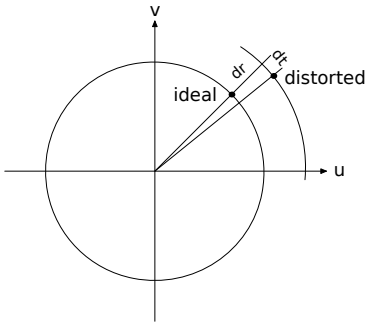

Figure 2.2: Illustration of effect of radial and tangential distortion (Figure is taken from [2])

2.2 Smisek depth model

Depth camera consists of an IR projector and an IR camera. The IR projector emits constant speckle pattern and the IR camera simultaneously detects a reflection of the pattern. Then the captured images are proceeded by the device and a raw disparity is produced. The guess of the procedure is presented here [12]. [12] also presents a way how to transform the raw disparity into a depth map. This approach originates in using a camera color pair and Smisek is using this approach. In fig. 2.3 the problem is illustrated. It's matter of a triangle similarities, where the triangle $\triangle|C_L C_R X|$ is similar to the $\triangle|ABX|$:

$$\frac{|C_L C_R|}{z} = \frac{|AB|}{z - f} \tag{2.7}$$

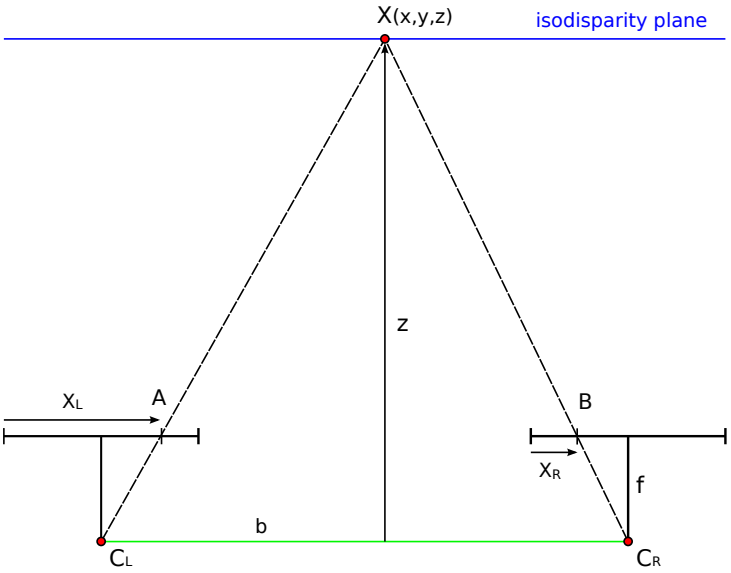

Figure 2.3: Pinhole model(Figure taken from [2])

where $b = |C_L C_R|$ represent a baseline, $\vec{f} = f$ represent the focal length and $|AB| = b - (X_L - X_R)$. After substitution $d_s = X_L - X_R$, where d_s is a disparity.

After isolation of the variable z, we get the following relation:

$$z = \frac{bf}{d_s} \tag{2.8}$$

Smisek further improves eq. 2.8 by assuming a relation between the raw disparity d returned by the Kinect and the disparity in eq. 2.8:

$$d_s = c_1 d + c_0 \tag{2.9}$$

where c_1, c_0 are coefficients of a polynomial that transform the raw disparity d into the disparity d_s.

Smisek also models a residual error in a image by a spatially varying offset Z_δ. The final Smisek's relation between the raw disparity and depth is hence in the form:

$$z = \frac{bf}{c_1 d + c_0} + Z_\delta(u, v) \tag{2.10}$$

where u,v are a pixel coordinates.

2.3 Herrera depth model

Herrera uses the same model (parameters b, f are part of the polynomial) :

$$z = \frac{1}{c_1 d_k + c_0} \tag{2.11}$$

where d_k is an undistorted raw disparity. d_k is related to d by the spatially varying offset. Herrera excepts that the offset is decaying with a distance from the depth camera:

$$d_k = d + D_\delta(u, v) exp(\alpha_0 - \alpha_1 d) \tag{2.12}$$

this feature is supported by a experimental data in [4](corresponding Figure 2.4) author claims these data were taken from multiple disparity images of a flat wall. Later, in chapter 5 these data are confronted with my results.

Herrera also requires a backward model, which convert the depth into the distorted disparity. Converting depth into the undistorted disparity is simple, we only need to express d_k from Equation 2.11:

$$d_k = \frac{1}{c_1 z_d} - \frac{c_0}{c_1} \tag{2.13}$$

expressing d from Equation 2.12 is more complex, because of the exponential. Herrera [4] solve this problem using the Lambert W function:

$$\begin{aligned}
y &= exp(\alpha_0 - \alpha_1 d_k + \alpha_1 \mathbf{D}_\delta(u, v) y) && (2.14)\\
y &= \frac{d_k - d}{\mathbf{D}_\delta(u, v)} && (2.15)\\
y &= exp(\alpha_1 \mathbf{D}_\delta(u, v) y) exp(\alpha_0 - \alpha_1 d_k) && (2.16)\\
\frac{-\tilde{y}}{\alpha_1 \mathbf{D}_\delta(u, v)} &= exp(-\tilde{y}) exp(\alpha_0 - \alpha_1 d_k) && (2.17)\\
\tilde{y} exp(\tilde{y}) &= -\alpha_1 \mathbf{D}_\delta(u, v) exp(\alpha_0 - \alpha_1 d_k) && (2.18)
\end{aligned}$$

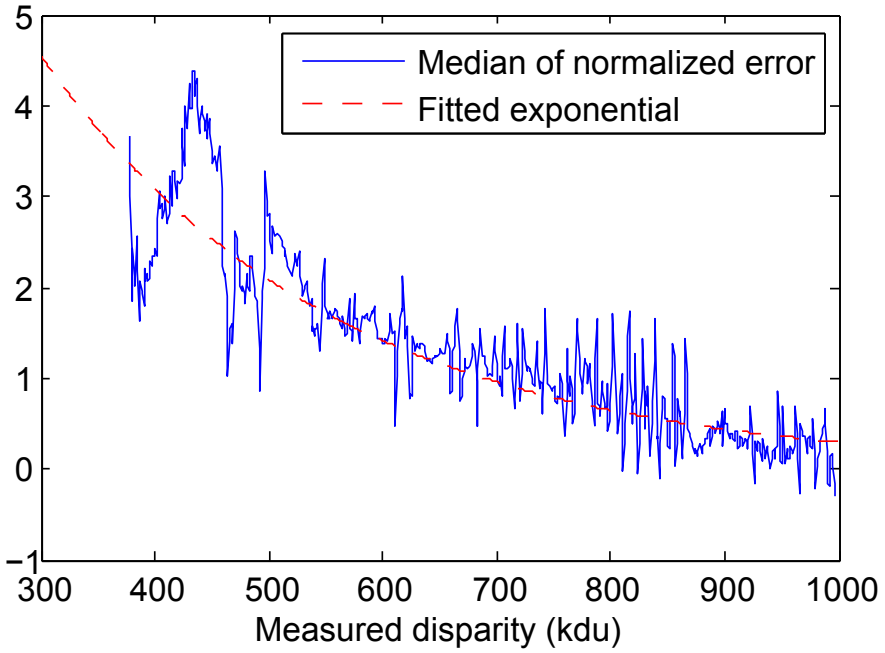

Figure 2.4: Relationship between the distortion magnitude and the measured disparity(Figure taken from [4])

now using the Lambert W function(solve relation in form W(z)exp(W(z)) = z) we solve our problem:

$$\tilde{y} = W(-\alpha_1 \mathbf{D}_\delta(u,v) exp(\alpha_0 - \alpha_1 d_k)) \tag{2.19}$$

$$(d - d_k)\alpha_1 = W(-\alpha_1 \mathbf{D}_\delta(u,v) exp(\alpha_0 - \alpha_1 d_k)) \tag{2.20}$$

$$d = d_k + \frac{W(-\alpha_1 \mathbf{D}_\delta(u,v) exp(\alpha_0 - \alpha_1 d_k))}{\alpha_1} \tag{2.21}$$

The Lambert W function is transcendental. Therefore, an analytic solution does not exist. The Solution is implemented in Matlab[15] and other mathematical packages, using an approximation(e.g. Newton's method).

2.4 Raposo depth model

Raposo uses basically the same model as Herrera, due to different calibration approach, there is the following difference:

$$d_k = d + W(u,v) exp(-\alpha_1 d) \tag{2.22}$$

where $W(u,v) = D_\delta(u,v) exp(\alpha_0)$ (do not confuse with the Lambert W function):

$$d = d_k + \frac{W(-\alpha_1 \mathbf{W}(u,v) exp(-\alpha_1 d_k))}{\alpha_1} \tag{2.23}$$

Raposo also does not require the backward model.

Chapter 3

State of the art

3.1 Related work

In this chapter are presented calibrations from Smisek [3], Herrera [3] and Raposo [6]. The Smisek's method precedes both Herrera's and Raposo's calibration. Probably the most innovative way to calibrate the Kinect is present in [6]. Where author calibrates both cameras at once, hence getting more information from each of them. Raposo's design is strongly influenced by Herrera and improves mainly time complexity.

3.2 Smisek calibration

The process of calibration suggested by Smisek in [3] is the first method that takes in account specification of RGBD camera. The approach requires additional IR source. Hence both external IR source and the IR projector interfere with each other. Therefore, it is better to switch between them. While other methods of calibration remove this inconvenience, they simultaneously present other. This method is fully automatic after the data are taken, provided that automatic corner detector is at disposal. Smisek calibrates the depth camera and the RGB camera separately.

3.2.1 Required data

Smisek [3] requires triples of disparity image, color image, and IR image of a calibration pattern. For each triple, following informations are needed:

1. 3D coordinates of corners in calibration pattern and their reprojection into RGB image plane.

2. 3D coordinates of corners in calibration pattern and their reprojection into IR image plane.

visualization of corners in RGB image is in Figure 3.2.

3.2.2 RGB/IR camera calibration

Smisek calibrates each camera separately. Calibration uses Bouget's calibration toolbox [13] on each set of triples . Because Smisek uses external IR source, he is capable of obtaining corners position from plane in IR image, making Bouget's calibration possible for IR images. Given model of the IR camera is treated as the depth camera model with constant pixel shift. Smisek estimates the pixel shift to be approximately 3 pixels in each dimension, giving following relation between pixel in depth and IR image:

$$X_d = X_{IR} + \left.\begin{bmatrix} -3 \\ -3 \end{bmatrix}\right\} \quad \text{offset between IR and depth image} \tag{3.1}$$

Smisek obtained this relation using Sobel's edge detector on a circular target projection. Projectins of fitted circles in th IR image and the disparity image are compared. The shift is then determined as a difference between centers in the IR image and the depth image (Figure 3.1).

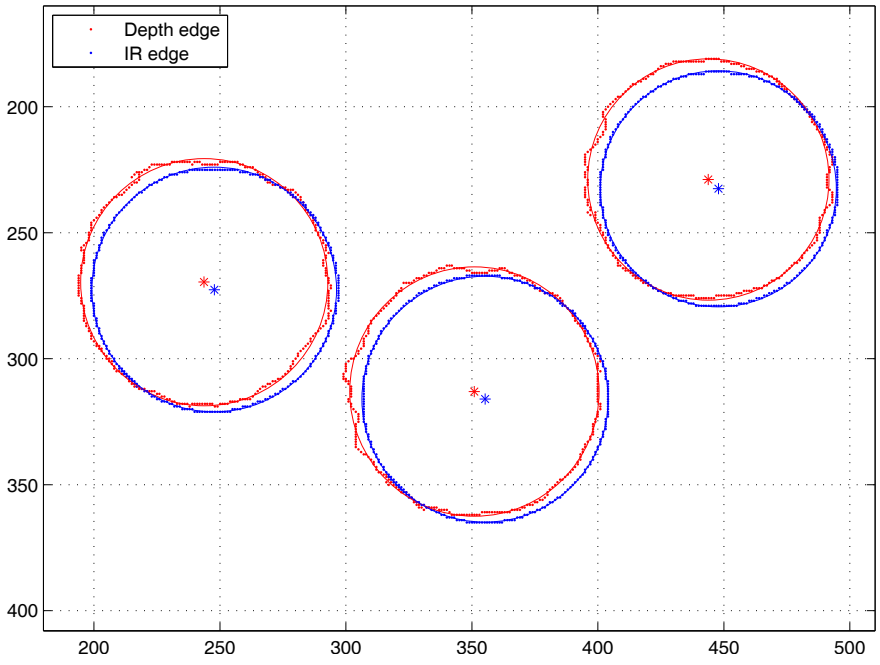

Figure 3.1: Shift of IR image against depth image (figure is taken from [3])

3.2.3 Depth camera calibration

Smisek [3] reconstructs the 3D coordination of every IR corner, using the model of the IR camera and Equation 3.1. Smisek obtains multiple mapping (z_{IR}, d) between the disparity and the depth. Parameters c_0, c_1 of the model mentioned in section 2.2 are estimated using (z_{IR}, d). Suggested approach is to minimize the least square error between z_{IR} and z_d, where z_d is the depth computed from disparity and estimated parameters c_0, c_1 according to Equation 2.10.

3.2.4 Disparity distortion estimation

Smisek's solution [3] consists of capturing multiple depth images of a flat wall perpendicular to axis z of the camera at known depth z^*, hence all disparity values should be the same. After a plane is fitted and the disparity measured, the difference between the fitted plane depths z^* and the depths from disparity (Equation 2.10) z_d still exhibits complex residual error.

$$z_{err}(u, v) = z^* - z_d(u, v) \tag{3.2}$$

Smisek's solution consists of adding offset to each pixel separately, using mean error from fitted planes and later uses this offset to adjust further measurements:

$$Z_\delta(u, v) = \frac{\sum_N (z^* - z_d(u, v))}{N} \tag{3.3}$$

Which completes calibration of Smisek's model and gives whole relation between depth and disparity Equation 2.10.

3.3 Herrera calibration

Herrera calibrates his model, which is described in chapter 2. The calibration consists of several steps:

1. Initialization:
   (a) RGB camera initialization.
   (b) Depth camera initialization (or using preset depth model).
   (c) Relative pose initialization (or using preset depth model).
2. Nonlinear optimization of every parameter except disparity distortion.
3. Disparity distortion estimation.
4. Repeating 2. and 3. until maxim number of steps is exhausted or the model of the RGBD camera is not improving.

The corners w.r.t. the depth camera reference frame {D} are unknown. The information can not be simply extracted, because the projected speckle pattern from the IR projector creates a high noise in IR image. The position of corners w.r.t. the image reference frame and w.r.t. the world reference frame {W} is known only for the RGB image.

3.3.1 Required data

Herrera [4] requires pairs of disparity and color images of the calibration pattern, where for each pair following informations are needed:

1. 3D points of corners and their reprojection in the 2D camera plane.
2. A polygon covering the calibration plane in each disparity image.

Visualization of additional information is in Figure 3.2

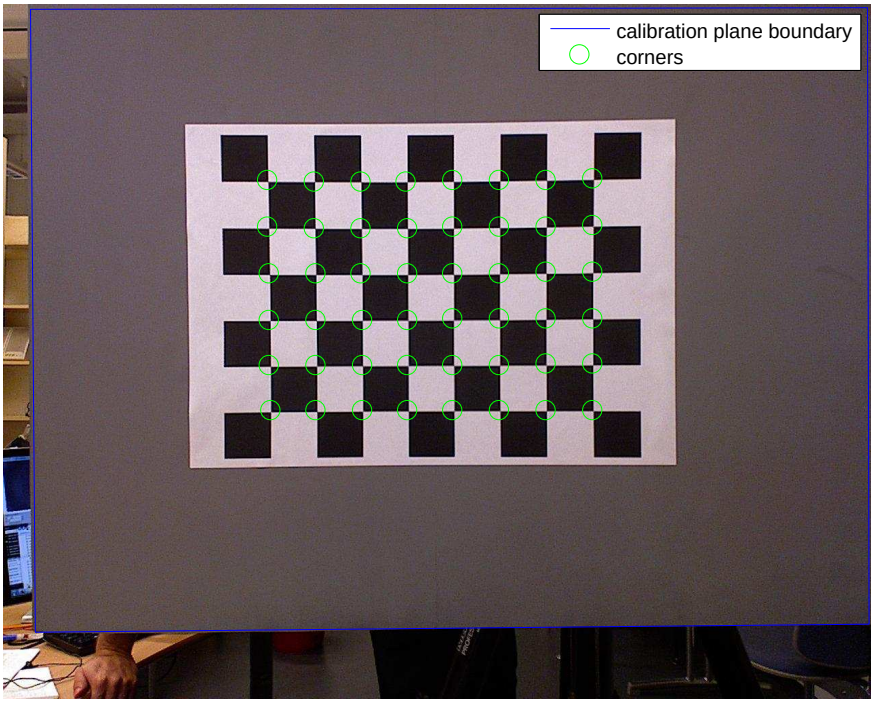

Figure 3.2: Visualization of data required for Herrera's calibration (The figure is taken from Herrera's toolbox [4]), notice that *calibration plane boundary* is actually corresponding to disparity image while this figure is color image.

3.3.2 Initialization

Corners based calibration of color camera

The initial guess is obtained through the method [10]. The procedure is very similar to initialization in Bouget's calibration and requires only pairs of corners in both {W} coordinates and image coordinates. Herrera also uses improvement of Zhang's method [10] by data normalization as suggested by Zhang himself.

Calibration of depth camera

Herrera suggests a method similar to initialization of the color camera. The checkerboard pattern is not visible in disparity image, hence initialization uses only four extreme corners in calibration plane. These four points are picked manually and the initial guess is very inaccurate. This is the reason why in the Herrera's toolbox itself this method is replaced by fixed initial values.

Relative pose estimation

Relative pose between the color camera and the depth camera is challenging to obtain. Mainly because the data for the depth camera describing the position of the calibration pattern in the calibration plane are unknown. Herrera uses coplanarity of {V} and {W}, hence both planes have the same equation $\mathbf{n}^T\mathbf{x} - \delta = 0$, where $\mathbf{n} = [0, 0, 1]^T$ and $\delta = 0$ in both {V} and {W}. Now using the relative pose between {W} and {C} ${}^W\mathbf{T}_C = \{{}^W\mathbf{R}_C, {}^W\mathbf{t}_C\}$ and dividing ${}^W\mathbf{R}_C$ into its

columns ${}^{W}\mathbf{R}_C = [\mathbf{r}_{c1}, \mathbf{r}_{c2}, \mathbf{r}_{c3}]$, we can transform the plane into the color camera reference frame:

$$\mathbf{n}_c = \mathbf{r}_{c3} \; and \; \delta_c = \mathbf{r}_{c3}^{T\,W}\mathbf{t}_C \tag{3.4}$$

with similar procedure, plane equation in the depth camera reference frame {D} is obtained.

$$\mathbf{n}_d = \mathbf{r}_{d3} \; and \; \delta_d = \mathbf{r}_{d3}^{T\,V}\mathbf{t}_D \tag{3.5}$$

Herrera is collecting plane equations for each images pair, thus getting following matrices:

$$\mathbf{M}_C = [\vec{n}_{c1}, \vec{n}_{c2}, ..., \vec{n}_{cn}] \tag{3.6}$$

$$\mathbf{M}_D = [\vec{n}_{d1}, \vec{n}_{d2}, ..., \vec{n}_{dn}] \tag{3.7}$$

$$\mathbf{b}_C = [\delta_{c1}, \delta_{c2}, ..., \delta_{cn}] \tag{3.8}$$

$$\mathbf{b}_D = [\delta_{d1}, \delta_{d2}, ..., \delta_{dn}] \tag{3.9}$$

The problem is therefore an overdetermined system and the solution is in form:

$$\begin{aligned} {}^{C}\mathbf{R}_D^{'} &= \mathbf{M}_D\mathbf{M}_C^T & (3.10)\\ {}^{C}\mathbf{t}_D &= (\mathbf{M}_C\mathbf{M}_C^T)^{-1}\mathbf{M}_C(\mathbf{b}_C - \mathbf{b}_D)^T & (3.11) \end{aligned}$$

where ${}^{C}\mathbf{R}_D^{'}$ tends not to be orthonormal. Valid rotation matrix can be obtained through SVD: ${}^{C}\mathbf{R}_D^{'} = UV^T$, where USV^T is the SVD of ${}^{C}\mathbf{R}_D^{'}$.

3.3.3 Non-linear optimization

Herrera uses non-linear optimization to improve both intrinsic and extrinsic parameters of both color and depth model. Only parameters that are optimized separately is the disparity distortion, due to waste amount of variables in the disparity distortion model. A non-linear cost function which is minimized is in form:

$$c = \frac{\sum |\hat{\mathbf{p}}_c - \mathbf{p}_c|^2}{\sigma_c^2} + \frac{\sum \left|\hat{d}_k - d_k\right|^2}{\sigma_d^2} \tag{3.12}$$

where $\hat{p}_c$ present the reprojected corner position from {W} to image plane and $\hat{d}_k$ is the undistorted disparity estimation from the fitted plane for the calibration pattern and p_c, d_k present measured values using the Kinect model. The task can be reformulate to following equation:

$$L_C, L_D, {}^{W_i}T_C, {}^{D}T_C = \underset{L_C, L_D, {}^{W_i}T_C, {}^{D}T_C}{\arg\min} \frac{\sum |\hat{\mathbf{p}}_c - \mathbf{p}_c|^2}{\sigma_c^2} + \frac{\sum \left|\hat{d} - d\right|^2}{\sigma_d^2} \tag{3.13}$$

The optimization is carry on by Levenberg-Marquardt algorithm using a function dedicated for non-linear optimization *lsqnonlin* from [15].

3.3.4 Disparity distortion estimation

Herrera calibrates the distortion of the depth model described in chapter 2 separately from the main non-linear minimization. However the disparity distortion estimation remain part of the same close-loop as the main non-linear minimization procedure. The target cost function, which Herrera tries to minimize [4]:

$$c_d = \sum_{image} \sum_{u,v} (\hat{d} + \mathbf{D}_\delta(u,v) exp(\alpha_0 - \alpha_1 \hat{d}) - d_k)^2 \tag{3.14}$$

where $\hat{d}$ presents the raw distortion captured by the Kinect and d_k is the predicted undistorted disparity. Following two steps are repeated until the disparity distortion model stop improving or the improvement is negligible.

$\mathbf{D}_\delta$ optimization

With fixed $\boldsymbol{\alpha}$, Herrera computes for each valid pixel(i.e. for each pixel which lay in the calibration plane) error for every measured and predicted pair $(\hat{d}, d_k)_i$ laying, in this particular pixel:

$$\hat{d}_i - d_{k,i} = \mathbf{D}_{\delta,i}(u,v) exp(\alpha_0 - \alpha_1 \hat{d}) \tag{3.15}$$

since $\boldsymbol{\alpha}$ is fixed, we get following expression:

$$\mathbf{D}_{\delta,i}(u,v) = \frac{\hat{d}_i - d_{k,i}}{exp(\alpha_0 - \alpha_1 \hat{d})} \tag{3.16}$$

Such $\mathbf{D}_{\delta,i}(u,v)$ is chosen, that least-square error for every measured pair in Equation 3.15 is minimized.

$\boldsymbol{\alpha}$ optimization

Second step consists of selecting a subset from all disparity data and perform nonlinear minimization using levenberg-marquardt algorithm with every modeled parameter fixed, except $\boldsymbol{\alpha}$.

3.4 Raposo calibration

Raposo [6] calibrates modified Herrera's model. Small difference between disparity distortion models is described in chapter 2. The calibration consists of several steps:

1. Initialization:
   (a) RGB camera initialization.
   (b) Preset Depth model.
   (c) Plane registration in dual space (relative pose initialization).

2. Nonlinear optimization of every parameter except the disparity distortion.

3. Disparity distortion estimation.

The corners w.r.t. the depth camera reference frame {D} are unknown. The information can not be simply extracted, because the projected speckle pattern from IR projector creates high noise in IR image. The position of corner points w.r.t. the image reference frame {C} and w.r.t. the world reference frame {W} is known only for the RGB image.

3.4.1 Required data

Raposo requires the same data as Herrera. Since he uses preset depth model, Raposo doesn't need four corners of the calibration plane in the disparity image.

3.4.2 Initialization

Corners based calibration of color camera

An initial guess is obtained with Bouguet's toolbox [13].

Calibration of depth camera

Initialization of the depth camera intrinsic is omitted. Raposo instead uses preset values. Similar to approach used in Herrera's toolbox.

Relative pose estimation

Raposo is seeking the transformation between {D} and {C} with another approach that he claims is robust. Briefly:

1. Select each triple of calibration planes w.r.t. the {D} and the {C} reference frame($\Pi_D^{(i)}, \Pi_C^{(i)}, i = \{1, 2, 3\}$)

2. For every triples of planes pair and corresponding transformation ${}^{D}T_C$, select the one that minimize Euclidean distance in dual space between the reprojected plane Π_C into {D} and the corresponding plane Π_D.

3.4.3 Non-linear optimization

Raposo's non-linear optimization is based on Herrera non-linear optimization. Raposo's approach is removing the error that he notice under poor initialization. According to Raposo [6], the Herrera's method face with a depth drift. Illustration of the problem is in Figure 3.3. Raposo solves this problem by adding another member into the cost function(Equation 3.12).

$$c = \frac{\sum |\hat{\mathbf{p}}_c - \mathbf{p}_c|^2}{\sigma_c^2} + \frac{\sum \left|\hat{d}_k - d_k\right|^2}{\sigma_d^2} + \beta \left|\hat{\lambda} - \lambda\right|^2 \tag{3.17}$$

The parameters have the same meaning as in Equation 3.12. Additional parameters $\beta, \lambda, \hat{\lambda}$ are meant to prevent the depth drift. λ represent Euclidean distances between points of an object and $\hat{\lambda}$ their corresponding projection. β is only weighting factor, allowing to change importance of additional term.

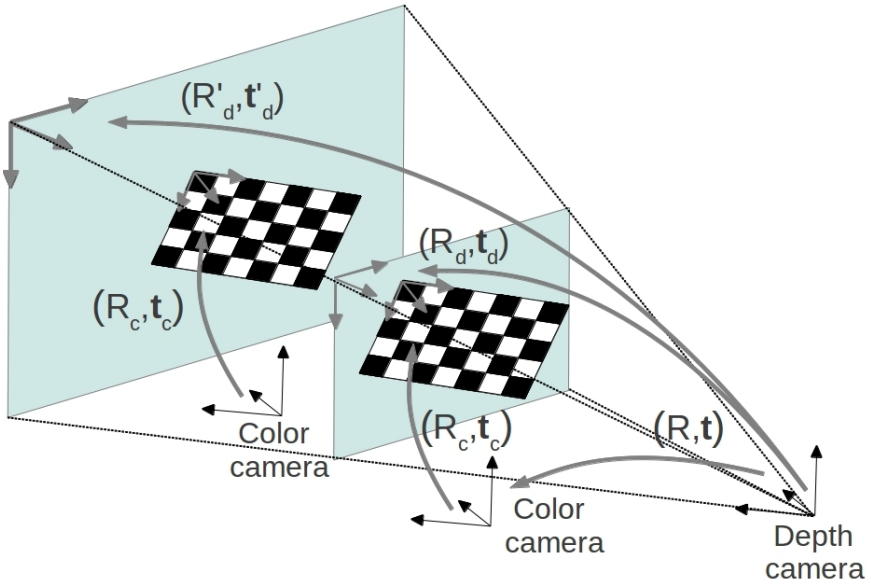

Figure 3.3: Visualization of the depth drift (transformation ${}^{D}t_{C}$ is correcting the misfitted depth model).

3.4.4 Disparity distortion estimation

Raposo's slightly different model is calibrated noticeably easier and faster. Mainly by eliminating the need of non-linear calibration. The model parameters are collected in two step in open loop(i.e. parameters are not use in main non-linear calibration procedure).

α_1 optimization

Given two measurements at the same pixel we get multiple pairs $(\hat{d}, d_k)_i$ that satisfies the following equation:

$$d_{k,i} - \hat{d}_i = \mathbf{W}_\delta(u,v) exp(-\alpha_1 \hat{d}_i) \tag{3.18}$$

for each pixel where we have two or more measurement, we take every subset of cardinality two and by substituting these value into Equation 3.21, we obtain system of equations:

$$\left.\begin{aligned} d_{k,1} - \hat{d}_1 &= \mathbf{W}_\delta(u,v) exp(-\alpha_1 \hat{d}_1) \\ d_{k,2} - \hat{d}_2 &= \mathbf{W}_\delta(u,v) exp(-\alpha_1 \hat{d}_2) \end{aligned}\right\} \quad \text{equation system} \tag{3.19}$$

from which is easy to compute α_1 by eliminating $W_\delta(u,v)$ in Equation 3.19:

$$\alpha_1 = \frac{ln \frac{d_{k,1} - \hat{d}_1}{d_{k,2} - \hat{d}_2}}{\hat{d}_2 - \hat{d}_1} \tag{3.20}$$

As estimation of α_1 Raposo [6] suggest average of all α_1 computed by Equation 3.20.

$\mathbf{W}_\delta$ optimization

With estimated α_1, we can easily compute W_δ from Equation 2.12:

$$\mathbf{W}_\delta(u, v) = (d_{k,i} - \hat{d}_i)/exp(-\alpha_1 \hat{d}_i) \tag{3.21}$$

Chapter 4

Contribution

According to goal of my work, Smisek's method was extend/changed by some of the Herrera's and Raposo's ideas. The results are available in chapter 5. Brief list of suggested changes:

1. Initialization using Bouget's calibration toolbox (basically Smisek's calibration).
2. Nonlinear optimization (modified Herrera).
3. Disparity distortion estimation (modified Raposo).

4.1 Required data

My method requires the same data as Smisek's method in chapter 3. TheOnly additional information is the plane mask (list of valid disparity points in each image).

4.2 Data collecting

I altered program Record (program for capturing RGB and depth image pair from library freenect) to better suit my purpose (capturing IR e.c.) and further improved by wrapping the modified program into the Matlab and equipping it with GUI. Short description of the capturing tool is in the Appendix A.

4.2.1 Artificial data

I also created an artificial data generator. The generator creates triples of images, expected extrinsic parameters for selected intrinsic parameters, and desired error. Artificial data can be used to measure robustness of calibration algorithm as well as it's precision. The result are available in chapter 5

Artificial data generating

The purpose of the artificial data generator is to convert intrinsic parameters into the calibration data (disparity and IR/RGB corners) and extrinsic parameters. Algorithm consist of following steps:

1. An input consists of intrinsic of both IR and RGB camera and depth model (without model of disparity distortion).

2. Generate rotation of image plane around axis x,y called $\mathbf{R}$.

3. Generate random distance d of camera (0.5-4 m) with uniform distribution.

4. Compute transition $\mathbf{t}$ such, that center of chessboard pattern will have coordinates $[0, 0, d, 1]^T = {}^W\mathbf{T}_C\mathbf{s}_W$, where ${}^W\mathbf{T}_C$ is transformation given by $\mathbf{R}$, $\mathbf{t}$ with same meaning as in chapter 2. The parameter $\mathbf{s}_W$ denotes center of calibration pattern.

5. Generate pairs of corners in image plane and in world reference frame using intrinsic model and obtain extrinsic parameters.

6. The same pair is obtained for the IR camera, using the relative pose between the IR camera and the RGB camera, and the IR camera intrinsic.

7. A plane spanning all corners is generated and by using IR camera parameters, offset between IR/depth camera, and relation between disparity and depth the disparity image is generated.

8. Apply noise on result. Each corner is randomly shift (vector with normal distribution of magnitude and uniform rotation). To each disparity value is added normaly distributed scalar.

4.3 Corner extraction

I combined some features of Bouget's toolbox [13] with Rufli toolbox [11], which makes corners selection fully automatic (i.e. user doesn't have to select four extreme corners). Selecting four extreme corners is needed only if automatic corner finder is unable to process images. There is no need to distinguish between IR and RGB image during corner extraction. An example of corners selected in image captured by previously noted capturing tool is in Figure 4.1.

4.4 Plane selection

I further enhanced plane selection using additional information from IR image. The approach consists of two steps:

1. Select initial plane by finding convex hull of corners extracted from IR image.

2. Do Smisek's calibration to obtain estimation of IR camera parameters.

3. Find initial plane in IR camera frame from parameters in step 2.

4. Find mean error and standard deviation for initial plane.

5. Initialize BFS (bread-first search) with convex hull.

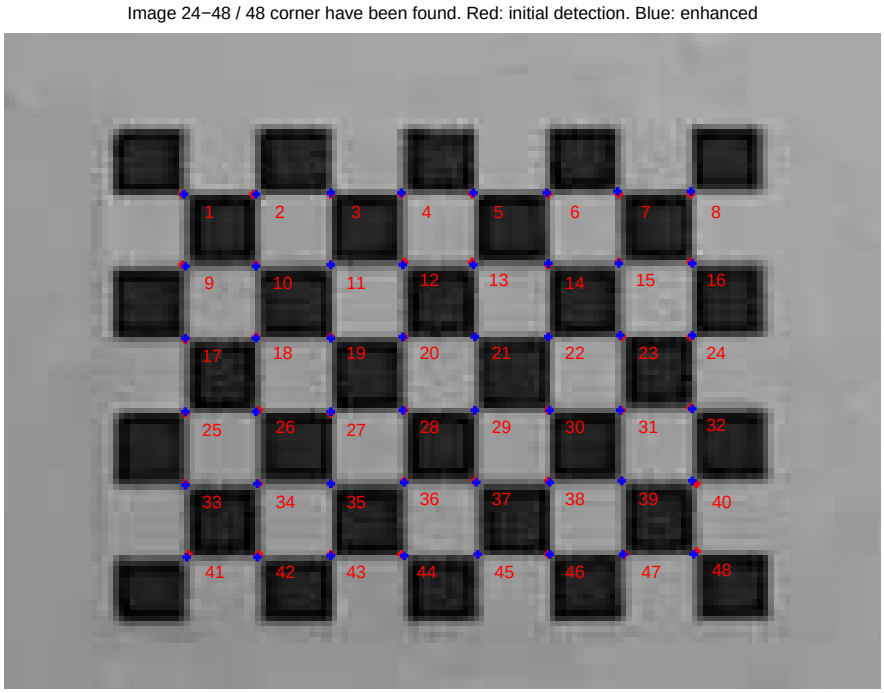

Figure 4.1: Sample output of corner detection program [11].

6. Search image using BFS and include those points whose error is reasonable close to plane (I used three times standard deviation of error for starting plane).

7. Return set of points found by BFS and set of points laying on initial plane as final plane incident with calibration pattern.

My approach allows to get more information from captured data while keeping procedure automatic and BFS assures continuos plane. Resulting plane mask does not have to be a polygon anymore. The roblem is that approach is not robust (can include/exclude wrong points forming wrong plane) and is computer time consuming.

4.5 Initialization

The initialization consists of calibrating using Smisek's method. Using both RGB and IR data in initialization step offers certain improvement over Herrera initialization. Such approach does not require static initialization model or does not rely on unstable depth image corners. Hence whole calibration is applicable on devices which differ noticeably from Kinect and also reduces chance of getting stuck in local optimum by bringing the initialized model close to final model (Especially useful for devices which differ from preset model of Depth camera).

4.5.1 Difference from Smisek's method

Obtaining transformation between color and depth camera

Both Raposo's [6] and Herrera's [4] way to get relative pose is obsolete, hence we obtained IR images which offer much simpler and accurate procedure to derive relative pose. Smisek's [3] calibration of RGBD camera is lacking approach to obtain relative pose between color and depth camera. I suggest following method to obtain relative pose between depth and color pair of cameras: The initialization step provide pair of transformation ${}^{W}\mathbf{T}_{C}$,${}^{W}\mathbf{T}_{D}$, for each pose:

$$ {}^{W}\mathbf{T}_{C} \cdot \mathbf{x}_{plane,w} = \mathbf{x}_{plane,c} \tag{4.1} $$

$$ {}^{W}\mathbf{T}_{D} \cdot \mathbf{x}_{plane,w} = \mathbf{x}_{plane,d} \tag{4.2} $$

our goal is transformation ${}^{D}\mathbf{T}_{C}$ for which should:

$$ {}^{D}\mathbf{T}_{C} \cdot \mathbf{x}_{plane,d} = \mathbf{x}_{plane,c} \tag{4.3} $$

by unfolding transformation ${}^{D}\mathbf{T}_{C}$

$$ {}^{W}\mathbf{T}_{C} \cdot {}^{D}\mathbf{T}_{W} \cdot \mathbf{x}_{plane,d} = \mathbf{x}_{plane,c} \tag{4.4} $$

hence transformation we seek is in form:

$$ {}^{D}\mathbf{T}_{C} = {}^{W}\mathbf{T}_{C} \cdot {}^{W}\mathbf{T}_{D}^{-1} \tag{4.5} $$

Such transformation is produced for each scene. Since the relative pose between depth camera coordinates {D} and {C} is constant for every scene, the one minimizing reprojection error is chosen:

$$ {}^{D}\mathbf{T}_{C}^{*} = \underset{{}^{D}\mathbf{T}_{C}}{\arg\min} \frac{\sum \left|\hat{\mathbf{p}}_{c} - \mathbf{p}_{c}\right|^{2}}{\sigma_{c}^{2}} + \frac{\sum \left|\hat{d} - d\right|^{2}}{\sigma_{d}^{2}} \tag{4.6} $$

this can be done by simply trying all obtained transformation.

4.6 Non-linear minimization

Next step consists of choosing non-linear minimization. In this step I experienced problems with Herrera calibration model. Non-linear minimization allows depth model to significantly shift from optimal model while improving model performance, even for waste amount of image (~ 24). I inspired by Raposo's solution and removed this error by adding additional term to cost function. New cost function is in form:

$$ c = \frac{\sum \left|\hat{\mathbf{p}}_{c} - \mathbf{p}_{c}\right|^{2}}{\sigma_{c}^{2}} + \frac{\sum \left|\hat{d}_{k} - d_{k}\right|^{2}}{\sigma_{d}^{2}} + \frac{\sum \left|\hat{\mathbf{p}}_{ir} - \mathbf{p}_{ir}\right|^{2}}{\sigma_{ir}^{2}} \tag{4.7} $$

where last term express IR camera reprojection error. Compare to Raposo's approach this additional term is more accurate, remains dimensionless, and does not require additional information about plane. Performance of whole non-linear minimization is thus improved.

4.7 Disparity distortion estimation

Final step consists of calibrating disparity distortion estimation. I decided to choose Raposo's approach. Description is provided in chapter 3.

Chapter 5

Results

In this chapter are published results for three calibration methods discussed before.

5.1 Data set description

I captured two data sets and generate two artificial data sets.

5.2 Real data sets

Real data sets were taken using planar checkerboard pattern of size A4 on a wall. Sample image is in Figure 5.1. First data set consist of 18 images triples (RGB,IR,disparity), where each triple were taken in different position(either distance or orientation differ). First data set is noted data set I. Second data set consists of 25 image triple, where each triple were taken according to Herrera's toolbox documentation [5]. Herrera advice to obtain four types of images(frontal plane,plane rotated around the X axis,plane rotated around the Y axis, and full planar surface for distortion correction). Since Herrera claims that minimum number of images of each kind is five and I don't need the last type(most of the images contain flat wall), I capture the other types more than five times. The second data set is called data set II. I captured similar data set with better light condition called data set III. And finally I also added data set IV identical with data set used by Smisek in [2].

5.3 Artificial data sets

Artificial data sets generator is described in chapter 4, I captured two data sets, one of total 25 images pair and second of total 5 images pair. These image triples are used mainly for determining ability of calibrations to obtain original intrinsic.

5.4 Cross-validation

In one round a random training subset is obtained from whole data set and is used to calibrate RGBD camera model. Rest of data is used for validation (a

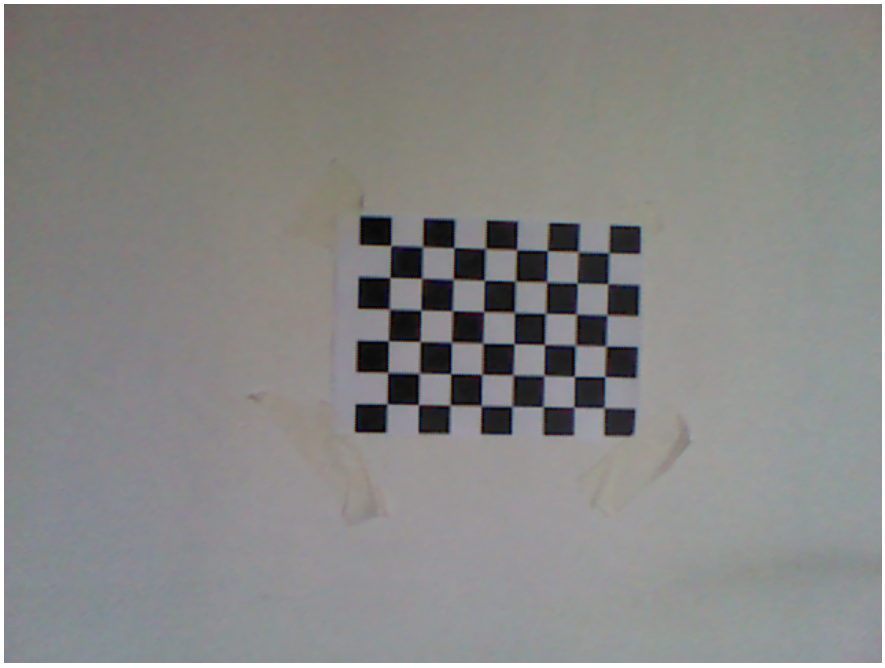

Figure 5.1: Sample of RGB image used for calibration procedure.

testing subset). In next rounds the exactly same procedure is repeated to reduce the variability in resulting errors evaluation.

5.5 Results structure

Results for each calibration(Smisek's [3], Herrera's [4], and my calibration), are in following format:

1. RGB camera model error
2. IR camera model error
3. Depth camera model error
4. RGB camera error box plot
5. Depth camera error box plot
6. Disparity distortion effect

For step 1-3, I expected the errors to follow normal distribution, and therefore present mean value μ and standard deviation σ.

5.5.1 RGB camera model error

RGB camera model error is Euclidean distance between measured corners and its reprojected position. Example with enhanced error scale(error is usually sub-pixel), can be seen in Figure 5.2. The resulting mean value and standard deviation is shown for each data set.

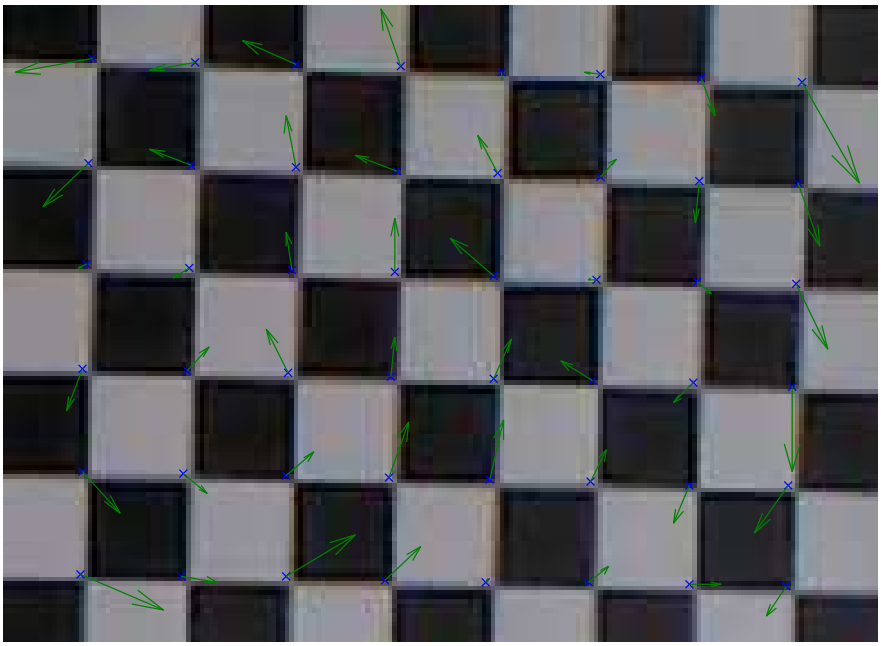

Figure 5.2: Sample of reprojected corners from {W} to image plane using RGB camera model.

5.5.2 IR camera model error

IR camera model error is evaluate in the same manner as RGB camera model error.

5.5.3 Depth camera model error

Depth camera model error is absolute distance between measured and expected disparity value. The results are present in similar way as IR/RGB camera model error. Only difference is in box plot and disparity distortion effect. I decided to omit absolute value to show orientation of error.

5.5.4 Camera error via box plot

Box plot from [15], allow to represent statistic information related to given image. The edge of the box are the 25th and 75th percentiles, the black line represent the most extreme data not considered outliers. The red crosses represent outliers.

5.5.5 Disparity distortion effect

Final statistic compare the error disparity-wise with and without disparity distortion correction. The resulting median for each disparity value is plotted.

5.6 Smisek's calibration results

The result of Smisek calibration are collected in following table:

Table 5.1: Smisek calibration results

Camera:	Color[px]		IR[px]		Depth[kdu]	
parameter:	μ	σ	μ	σ	μ	σ
data set I	0.175	0.086	1.084	0.713	2.446	1.798
data set II	0.238	0.172	1.171	0.839	5.568	3.578
data set III	0.180	0.106	0.394	0.374	1.539	1.117
data set IV	0.118	0.068	0.195	0.104	0.747	0.762

Intrinsic for data set III:

Table 5.2: Color camera intrinsic (Smisek calibration)

focal length[px]		principal point[px]		
524.453	524.488	305.704	245.834	
k_1	k_2	k_3	k_4	k_5
-0.062	0.576	-0.005	-0.006	0.000

Table 5.3: Depth camera intrinsic (Smisek calibration)

focal length[px]		principal point[px]		
588.372	588.082	307.321	229.502	
k_1	k_2	k_3	k_4	k_5
-0.245	1.562	-0.005	-0.008	0.000

Table 5.4: Depth camera intrinsic (Smisek calibration)

c_0	c_1
3.15856	-0.00291

Table 5.5: Smisek time complexity (Data set III)

μ [s]	σ [s]
13.311	0.691

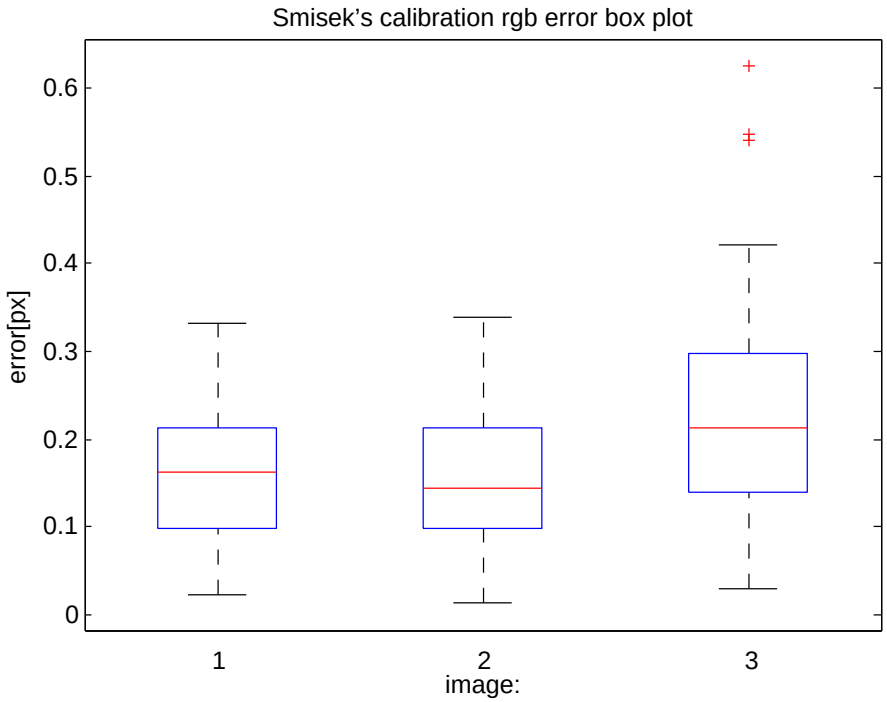

Figure 5.3: Box plot from [15], description of figure is in subsection 5.5.4

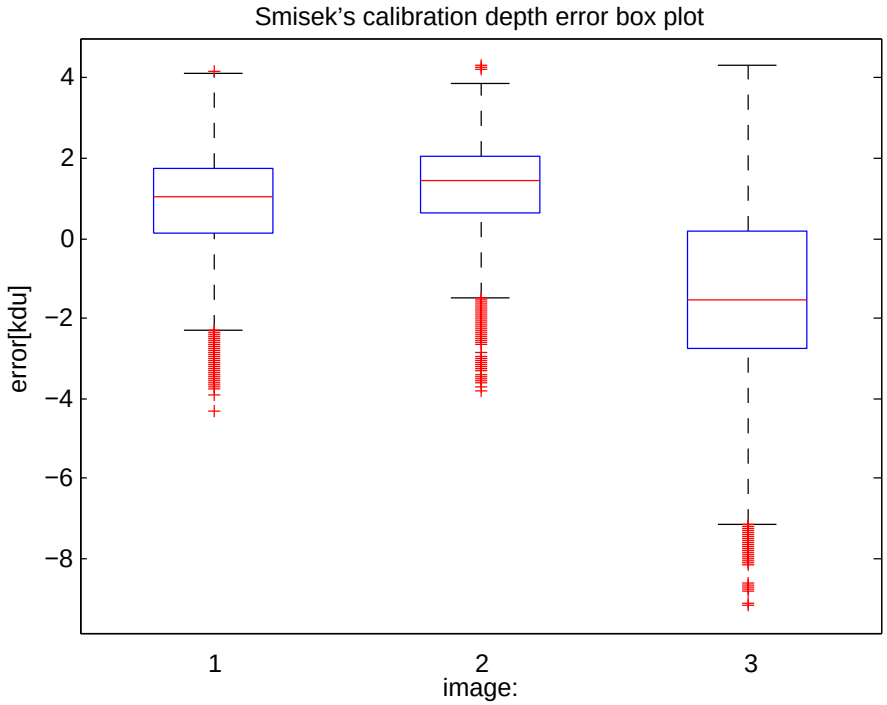

Figure 5.4: Box plot from [15], description of figure is in subsection 5.5.4

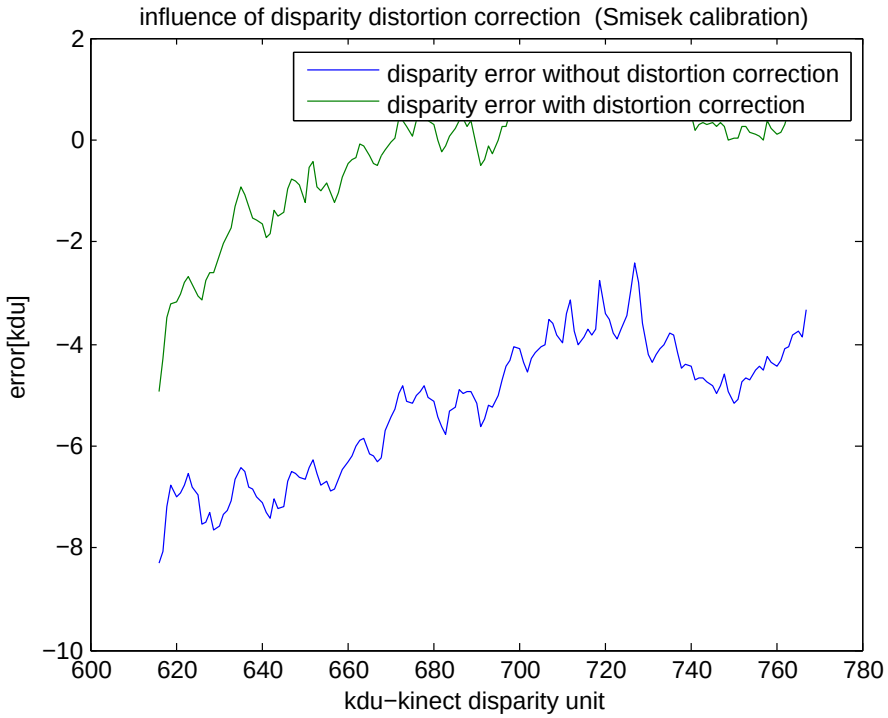

Figure 5.5: The dependency of disparity error on measured disparity, both with and without disparity distortion correction. Error for each disparity value is median from errors on this particular disparity value.

5.7 Herrera's calibration results

The result of Smisek calibration are collected in following table:

Table 5.6: Herrera calibration results

Camera:	Color[px]		IR[px]		Depth[kdu]	
parameter:	μ	σ	μ	σ	μ	σ
data set I	0.178	0.088	75.590	97.268	0.731	0.590
data set II	0.248	0.192	2.250	1.167	0.962	1.986
data set III	0.181	0.107	1.631	1.132	0.652	0.511
data set IV	0.118	0.067	2.556	1.853	0.525	0.408

Intrinsic for data set III:

Table 5.7: Color camera intrinsic (Herrera calibration)

focal length[px]		principal point[px]		
522.233	521.977	306.908	247.344	
k_1	k_2	k_3	k_4	k_5
-0.029	0.265	-0.002	-0.006	-3.547

Table 5.8: Depth camera intrinsic (Herrera calibration)

focal length[px]		principal point[px]		
562.229	584.050	334.965	239.949	
k_1	k_2	k_3	k_4	k_5
0.000	0.000	0.000	0.000	0.000

Table 5.9: Depth camera intrinsic (Herrera calibration)

c_0	c_1
3.17800	-0.00293

Table 5.10: Herrera time complexity (Data set III)

μ [s]	σ [s]
94.379	11.337

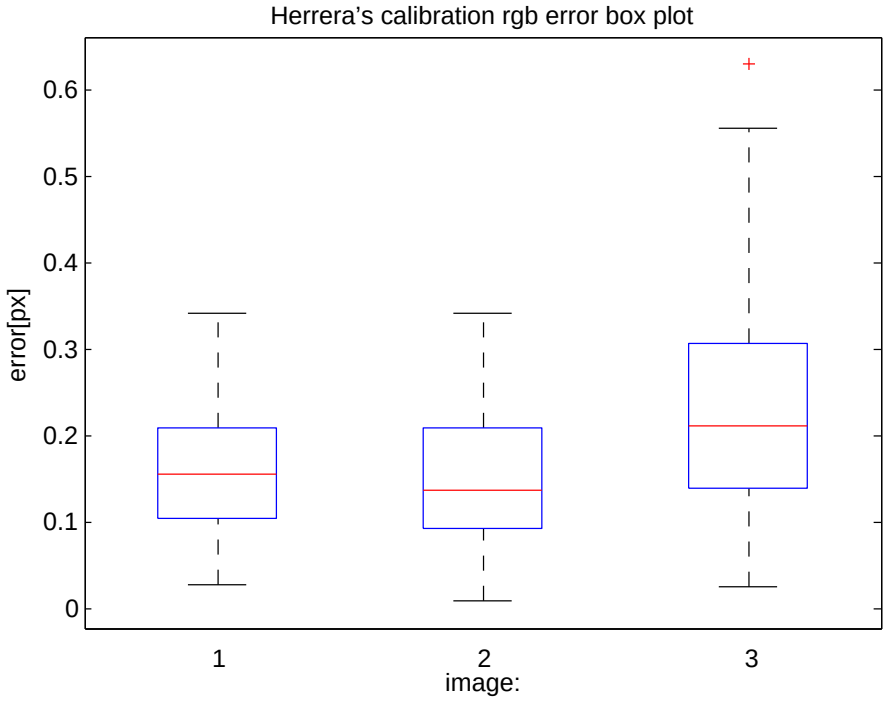

Figure 5.6: Box plot from [15], description of figure is in subsection 5.5.4

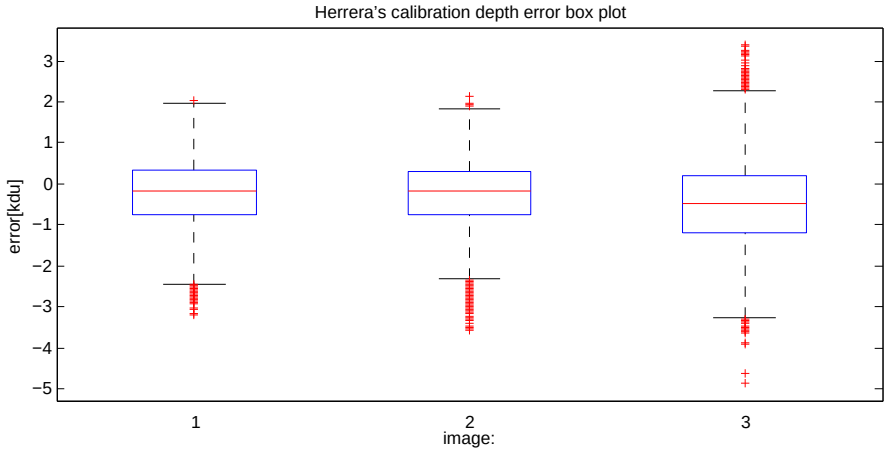

Figure 5.7: Box plot from [15], description of figure is in subsection 5.5.4

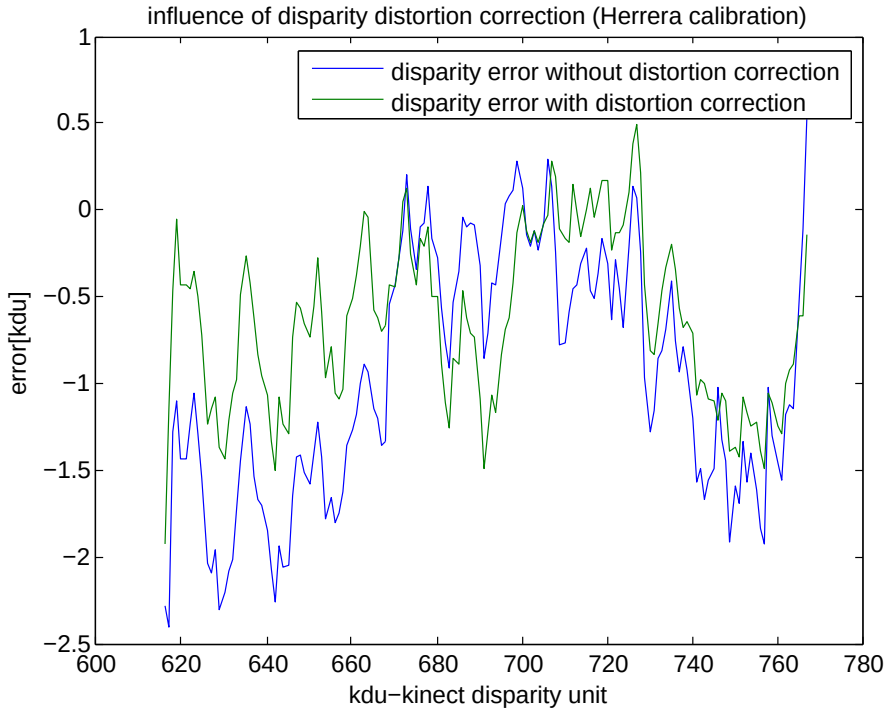

Figure 5.8: The dependency of disparity error on measured disparity, both with and without disparity distortion correction. Error for each disparity value is median from errors on this particular disparity value.

5.8 My calibration results

The result of Smisek calibration are collected in following table:

Table 5.11: My calibration results

Camera:	Color[px]		IR[px]		Depth[kdu]	
parameter:	μ	σ	μ	σ	μ	σ
data first set	0.182	0.095	0.659	0.523	0.794	0.567
data big 1	0.253	0.223	0.389	0.316	1.187	1.403
data big 2	0.197	0.116	0.315	0.359	0.726	0.529
Smisek data set	0.121	0.067	0.196	0.114	0.553	0.417

Intrinsic for data set III:

Table 5.12: Color camera intrinsic (My calibration)

focal length[px]		principal point[px]		
510.850	507.605	303.564	253.058	
k_1	k_2	k_3	k_4	k_5
0.108	1.544	0.002	-0.023	-14.390

Table 5.13: Depth camera intrinsic (My calibration)

focal length[px]		principal point[px]		
574.610	574.164	328.351	232.072	
k_1	k_2	k_3	k_4	k_5
-0.251	1.659	-0.005	-0.007	0.000

Table 5.14: Depth camera intrinsic (My calibration)

c_0	c_1
3.19066	-0.00293

Table 5.15: My calibration time complexity (Data set III)

μ [s]	σ [s]
89.191	9.464

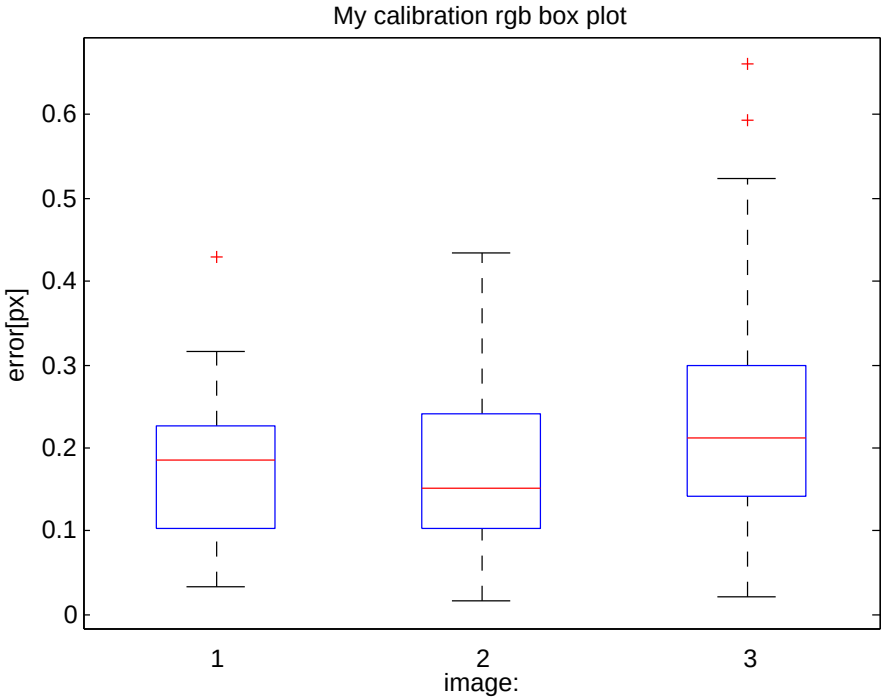

Figure 5.9: Box plot from [15], description of figure is in subsection 5.5.4

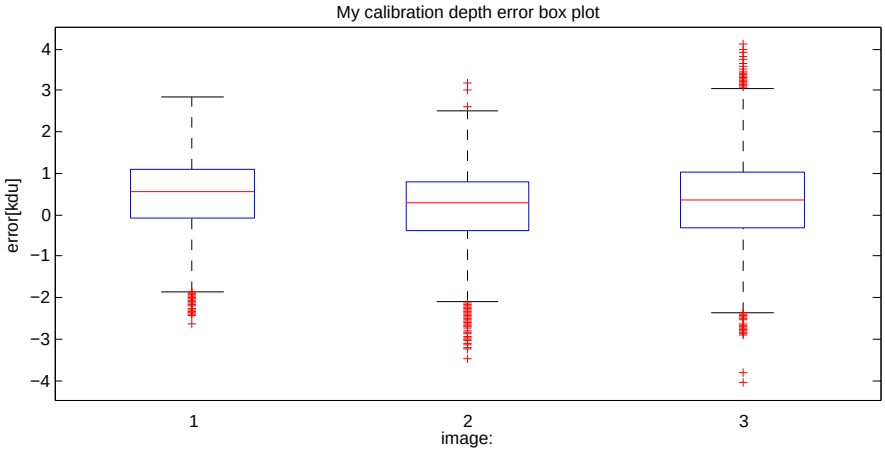

Figure 5.10: Box plot from [15], description of figure is in subsection 5.5.4

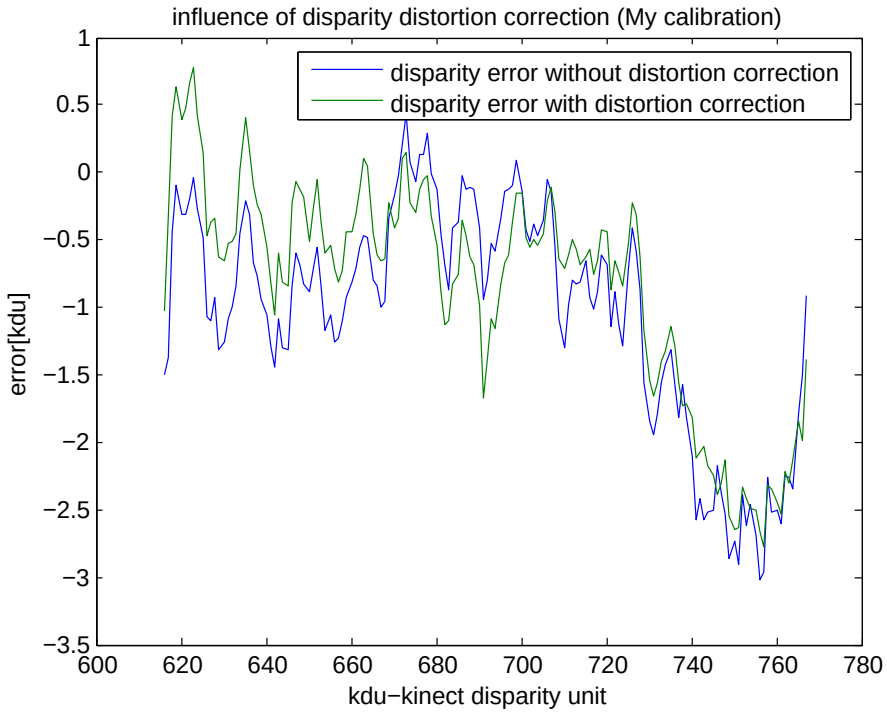

Figure 5.11: The dependency of disparity error on measured disparity, both with and without disparity distortion correction. Error for each disparity value is median from errors on this particular disparity value.

5.9 Tests aiming at comparing Herrera's and my approach

I performed multiple test with artificial data and data set III. First is presented visualization of cost function progression during calibration process in each calibration step. Later, I am going to show how does Herrera's and my modified calibration deal with artificial data. Artificial data input intrinsic are similar to ground truth and present good basis for such validation.

5.9.1 Cost function

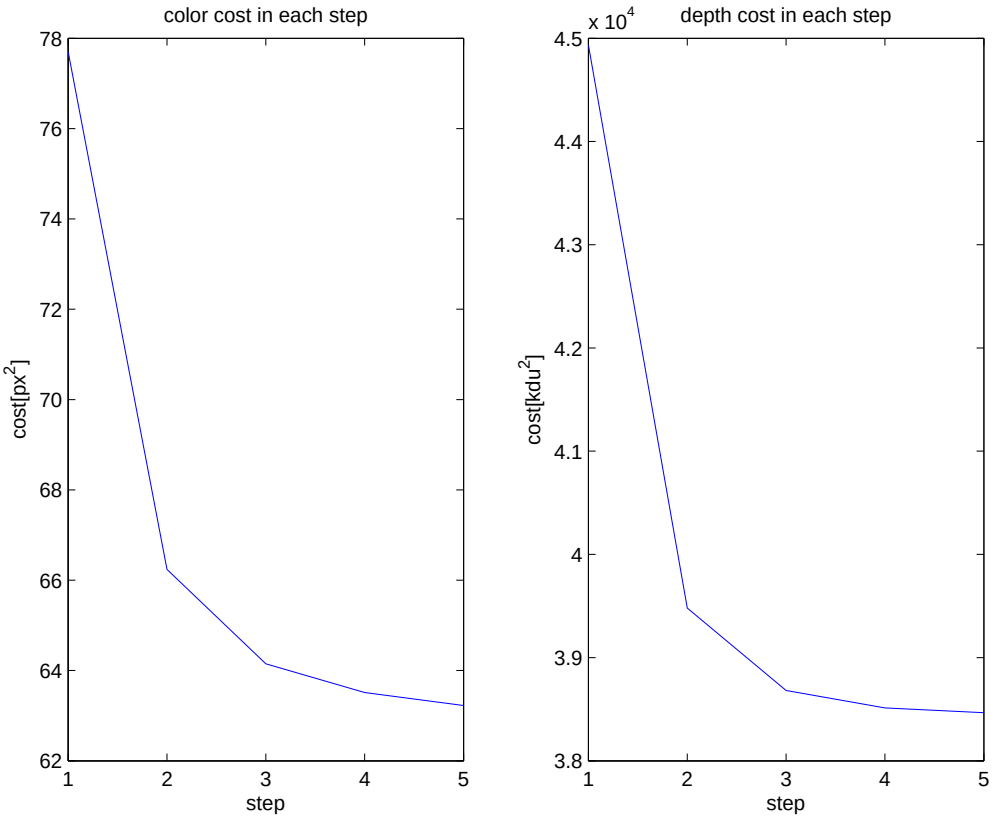

Figure 5.12: Graphs of improvement of cost in each step. My calibration for same data, take only one step to get similar result. Graph obtained using data from data set III.

5.9.2 Artificial data

Next test consist of calibrating data constructed artificially, as described previously. The input intrinsic are:

Table 5.16: Color camera intrinsic (generating intrinsic)

focal length[px]		principal point[px]		
500	500	310	240	
k_1	k_2	k_3	k_4	k_5
-0.008	-0.029	0	-0.002	0.000

Table 5.17: Depth camera intrinsic (generating intrinsic)

focal length[px]		principal point[px]		
580	580	320	240	
k_1	k_2	k_3	k_4	k_5
-0.103	0.434	0.005	0.003	0.000

Table 5.18: Depth camera intrinsic (generating intrinsic)

c_0	c_1
3.12000	-0.00286

The noise standard deviation is $\sigma_{rgb} = 0.2[px]$, $\sigma_{ir} = 0.2[px]$, $\sigma_{ir} = 0.7[kdu]$. The noise mean value is 0.

My calibration resulting intrinsic

Table 5.19: Herrera's calibration results

Camera:	Color[px]		IR[px]		Depth[kdu]	
parameter:	μ	σ	μ	σ	μ	σ
small data set	0.163	0.107	0.201	0.134	0.585	0.458
big data set	0.000	0.000	0.000	0.001	0.243	0.146

Table 5.20: Color camera intrinsic-results of calibrating small data set

focal length[px]		principal point[px]		
499.769	499.203	317.815	233.722	
k_1	k_2	k_3	k_4	k_5
-0.316	11.293	-0.004	0.004	-120.754

Table 5.21: Depth camera intrinsic-results of calibrating small data set

focal length[px]		principal point[px]		
580.103	579.559	315.418	238.142	
k_1	k_2	k_3	k_4	k_5
-0.071	-0.233	0.001	0.010	0.000

Table 5.22: Depth camera intrinsic-results of calibrating small data set

c_0	c_1
3.11299	-0.00285

Table 5.23: Color camera intrinsic-results of calibrating big data set

focal length[px]		principal point[px]		
500.002	500.002	309.999	240.000	
k_1	k_2	k_3	k_4	k_5
-0.008	-0.031	-0.000	-0.002	-0.000

Table 5.24: Depth camera intrinsic-results of calibrating big data set

focal length[px]		principal point[px]		
580.000	580.000	317.000	237.000	
k_1	k_2	k_3	k_4	k_5
-0.103	0.434	0.005	0.003	0.000

Table 5.25: Depth camera intrinsic-results of calibrating big data set

c_0	c_1
3.12002	-0.00286

Herrera resulting intrinsic

Test consist of calibrating data constructed artificially, as described previously. The input intrinsic are:

Table 5.26: Herrera's calibration results

Camera:	Color[px]		IR[px]		Depth[kdu]	
parameter:	μ	σ	μ	σ	μ	σ
small data set	0.174	0.110	108.336	203.170	0.532	0.424
big data set	0.000	0.000	0.087	0.055	0.244	0.146

Table 5.27: Color camera intrinsic-results of calibrating small data set

focal length[px]		principal point[px]		
499.023	499.182	310.889	232.310	
k_1	k_2	k_3	k_4	k_5
-0.035	0.538	-0.005	-0.001	-0.070

Table 5.28: Depth camera intrinsic-results of calibrating small data set

focal length[px]		principal point[px]		
583.495	587.161	308.449	237.279	
k_1	k_2	k_3	k_4	k_5
0.000	0.000	0.000	0.000	0.000

Table 5.29: Depth camera intrinsic-results of calibrating small data set

c_0	c_1
3.16712	-0.00290

Table 5.30: Color camera intrinsic-results of calibrating big data set

focal length[px]		principal point[px]		
499.962	499.976	309.919	240.020	
k_1	k_2	k_3	k_4	k_5
-0.003	-0.191	-0.000	-0.002	1.400

Table 5.31: Depth camera intrinsic-results of calibrating big data set

focal length[px]		principal point[px]		
580.451	580.719	317.642	237.151	
k_1	k_2	k_3	k_4	k_5
0.000	0.000	0.000	0.000	0.000

Table 5.32: Depth camera intrinsic-results of calibrating big data set

c_0	c_1
3.11936	-0.00285

Chapter 6

Discussion

This chapter summarizes chapter 5.

6.1 Summary

I compared both approach and rewrite following advantages/disadvantages of my calibration compare with Herrera's calibration:

6.1.1 Calibration improvement

Designed calibration brings following improvements:

1. The resulting difference between f_x , f_y in depth model is smaller. The difference is noticeable (check Table 5.13 and Table 5.9). I notice similar pattern in each tested data set. Herrera's method is more vulnerable to over-fitting.

2. The time complexity of non-linear minimization is improved greatly. But overall time complexity is improved only slightly. The reason is disparity distortion correction. Improvement of non-linear minimization is direct consequence of better initial calibration.

3. The final calibration done on measurements shows much better calibration of IR camera. This is valid for every tested data set. I anticipate this behavioral, since my calibration takes in account IR data.

4. The results on artificial data show that even small data set allows good calibration of depth camera. Herrera's calibration struggles with local optimum.

5. Eventually Herrera's calibration may not work with calibration parameters far from optimum (preset depth model), Smisek's calibration used for initialization does not suffer such problem. Therefore, my calibration is better suited for different RGBD cameras.

6.1.2 My calibration disadvantage

1. The method does not improve depth camera performance. Results are better than Smisek's calibration, but not better than Herrera's calibration. The reason is probably combined calibration of depth camera with IR and depth data. Such approach creates more constrains for non-linear optimization algorithm. On the other hand, mean values and especially standard deviations are similar.

2. On small artificial data set, method has problem with color camera distortion.

3. Method needs data triples instead pairs. Additional data are also difficult to obtain (process requires external source of IR light).

4. Offset between IR and depth camera remains constant. This doesn't allow calibration to optimally calibrate on device with different offset.

6.2 Recommendation

I transformed some of the disadvantages and other problem I occurred into following recommendation for improvement:

1. The IR camera non-linear term in Equation 4.7 could be removed and strategy similar to the Neyman-Pearson task could be used (i.e. minimize original Equation 4.7 under condition that the IR term will not exceed some predefined value).

2. Alternative could be variable offset, so far, I was using the constant shift between the IR and the depth camera. Adding other two parameters (shift vector) into the non-linear minimization could achieve two improvements. First, even wrongly given offset could reach its optimum. Second, the IR and the depth camera model would obtain 2 degree of independence that would likely improve the calibration results. On the other hand, this feature may reduce the ability of the IR term to prevent the depth model from shift towards locale optimum.

3. Raposo's disparity distortion model is achieving slightly better result then Herrera's (at least in open loop). However the selection of dumping parameter α_1 in subsection 3.4.4 is inconvenient. After careful analyse, I found that the result vary significantly on selected disparity points subsets. I believe that with correct data preproccesing, the method could be more stable. I tried rejecting extreme values and median without success (i.e. stabilise result no matter the selected subset). I abandon this matter without any improvements.

4. After analyzing captured data, I came to conclusion that parameters α_1 is related to rounding error in relation Equation 2.11. It follows that error in greater distance is more likely to be rounded (i.e. the more far the error is, it is more likely it will be neglected because of quantization). This explain the

dumping tendencies of residual depth error and could be used to compute dumping parameters directly or at least obtain guess.

5. The Smisek data offers better calibration quality! A3 calibration plane is possibly better.

Chapter 7

Conclusions

I compared three calibration concepts and created new calibration method. My calibration improves its predecessor in accuracy (Smisek's calibration) and robustness (Herrera's calibration). The calibration uses Smisek's calibration in initialization step and takes into account the overall performance (i.e. minimizing the error for IR, depth and color data simultaneously). I also automized plane selection, hence make preprocessing of calibration data simpler.The algorithm requires a simple planar checkerboard pattern on a flat surface and an external IR source.

I applied the disparity distortion model of Raposo and achieved slightly better results then Herrera's model. The final calibration was tested on artificial data and proved ability to better fit model that differ from preset depth camera model used by Herrera. I believe that suggested calibration is better suited for RGBD cameras with unknown intrinsic estimation.

All three calibrations, automatic corner finder, visualization methods, statistic computation, exporting tools, and artificial data generator are part of electronic attachment of this thesis. I also included the capturing tool. Basic information are in appendix. I concluded that requirements of my thesis were met.

Bibliography

[1] T. Pajdla, Elements of Geometry for Computer Vision[online], 2013 [22.5.2014], URL: https://cw.felk.cvut.cz/courses/GVG/2013/Lecture/GVG-2013-Lecture.pdf.

[2] J. Smíšek, 3D Camera Calibration, Praha: CTU 2011, Master's thesis, CTU, Faculty of electrical engineering, Department of Cybernetics.

[3] J. Smíšek, M. Jančošek, T. Pajdla, 3D with Kinect, in: IEEE Workshop on Consumer Depth Cameras for Computer Vision, 2011.

[4] D. Herrera C., J. Kannala, J. Heikkilä, Joint depth and color camera calibration with distortion correction, TPAMI, 2012.

[5] Kinect Calibration Toolbox, [online], D. Herrera C., [22.5.2014], URL: http://www.ee.oulu.fi/ dherrera/kinect/.

[6] C. Raposo, J.P. Barreto, U. Nunes. Fast and Accurate Calibration of a Kinect Sensor. International Conference on 3DV, pp 342 – 349, 2013.

[7] Kinect calibration, [online], N. Burrus, 2011, [22.5.2014], URL: http://nicolas.burrus.name/index.php/Research/KinectCalibration.

[8] I. Dryanovski, W.Morris, and S. Magnenat. kinect_node. http://ros.org/wiki/kinect_node, 2014

[9] Prime Sense LTD., METHOD AND SYSTEM FOR OBJECT RECONSTRUCTION, ZALEVSKY Zeev, SHPUNT Alexander, MAIZELS Aviad, GARCIA Javier, 19.04.2007, Pub. No.: WO/2007/043036, [22.5.2014],URL: http://patentscope.wipo.int/search/en/WO2007043036

[10] Z. Zhang, Flexible camera calibration by viewing a plane from unknown orientations in: ICCV, 1999, pp. 666-673.

[11] Rufli, M., Scaramuzza, D., and Siegwart, R., Automatic Detection of Checkerboards on Blurred and Distorted Images,Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008), Nice, France, September 2008.

[12] Technical description of Kinect calibration [online], K. Konolige, P. Mihelich, [22.5.2014], URL: http://wiki.ros.org/kinect_calibration/technical#Depth_calculation.

[13] Camera calibration toolbox [online], J.Y. Bouguet, [22.5.2014], URL: http://www.vision.caltech.edu/bouguetj/calib_doc/.

[14] D. Brown, Close-range camera calibration. PHOTOGRAMMETRIC ENGINEERING, 37(8):855–866, 1971.

[15] MATLAB Release 2013b, The MathWorks, Inc., Natick, Massachusetts, United States.

Appendices

Appendix A

Capturing tool

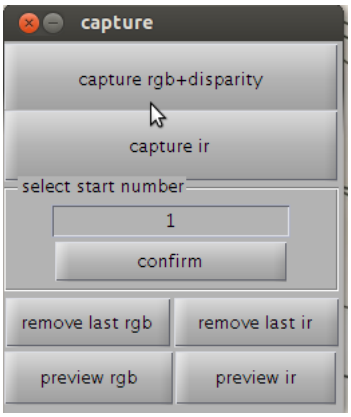

Figure A.1: capturing tool GUI screenshot

Capturing tool allows to capture IR and RGB+disparity image, change numbering, delete captured image, and previewing images before capturing.

Appendix B

Calibration toolbox

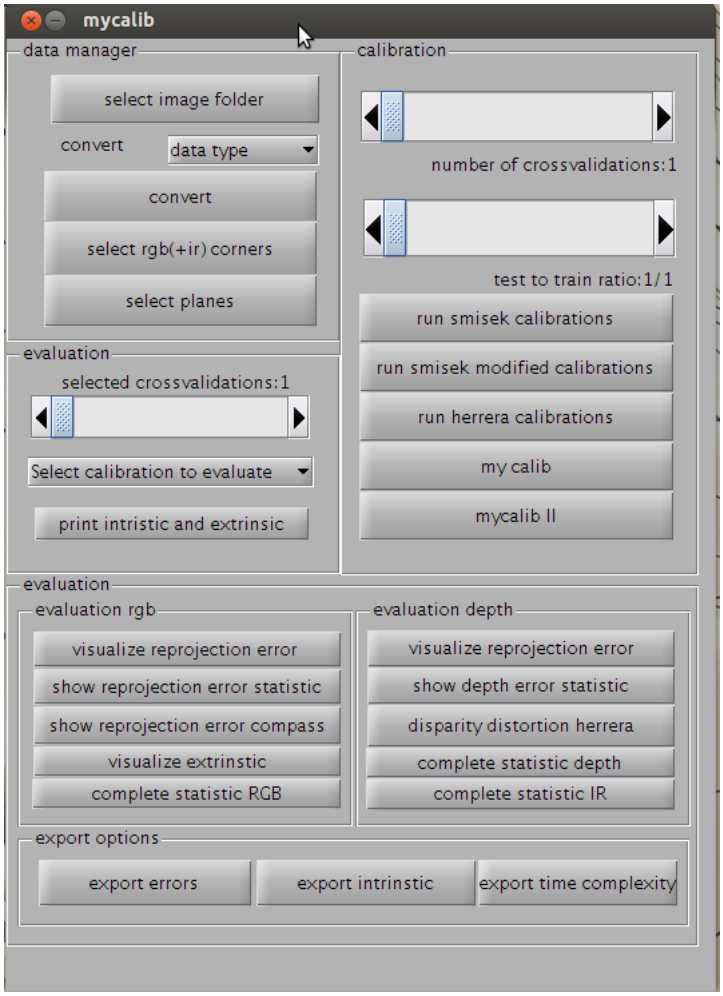

Figure B.1: calibration toolbox GUI screenshot

Calibration GUI was designed to help me with my work. GUI allows to quickly change data folder, and cross-validation options. It has some options to visualize results (plotting position of camera, reprojection image, etc.) and computing error. Exporting options are meant to simplify table writing in LaTeX.