TABLE OF CONTENTS

Chapter 1 Introduction

1.1 RGBD Cameras

A Red-Green-Blue-Depth (RGB-D) camera is a sensing system that captures RGB images along with per-pixel depth information. Usually it is simply a combination of a RGB sensor and a depth sensor with an alignment algorithm. For instance, the PrimeSense's technology had been originally applied to gaming, with user interfaces based on gesture recognition instead of using a controller (also called Natural User Interface, NUI [1]). PrimeSense was best known for licensing the hardware design and chip used in Microsoft's first generation of Kinect motion-sensing system for the Xbox 360 in 2010 [2]. The PrimeSense sensor projects an infrared speckle pattern, which will then be captured by an infrared camera in the sensor. A special microchip is employed to compare the captured speckle pattern part-by-part to reference patterns stored in the device, which were captured previously at known depths. The final per-pixel depth will be estimated based on which reference patterns the captured pattern matches best [3]. Other than the first generation of Kinect camera, Asus Xtion PRO sensor, another consumer NUI application product, has also applied the PrimeSense's technology [4].

As a competitor [5] of PrimeSense Structured Light technology, time-of-flight technology had been applied into PMD[Vision] CamCube cameras and 3DV's ZCam cameras. Based on known speed of light, Time-of-Flight (ToF) camera resolves distance by measuring the "time cost" of a special light signal traveling between the camera and target for every single point. The "time cost" variable that ToF camera measures is the phase shift between the illumination and reflection, which will be translated to distance [6]. To detect the phase shifts, a light source is pulsed or modulated by a continuous wave, typically a sinusoid or square wave. The ToF camera illumination is typically from a LED or a solid-state laser operating in the near-infrared range invisible to human eyes. Fabrizio *et al.* [7] com-

pared the time-of-flight (PMD[Vision] CamCube) camera and PrimeSense (first generation Kinect) camera in 2011. He showed that the time-of-flight technology is more accurate and claimed that the time-of-flight technology will not only be extended to support colours and higher frame sizes, but also rapidly drop in price. In 2010, it was announced that Microsoft would acquire Canesta for an undisclosed amount [8]. And in 2013, Microsoft released the Xbox One, whose NUI sensor KinectV2 features a wide-angle Canesta ToF camera.

Unlike the PrimeSense's speckle pattern or KinectV2's ToF, Intel RealSense camera utilizes stereo vision [9]. Its sensor actually has three cameras: two IR cameras (left and right), and one RGB camera. Additionally, RealSense camera also has an IR laser projector to help the stereo vision recognize depth at unstructured surfaces. Compared with KinectV2 camera, RealSense camera is more like a desktop usage to capture faces or even finger gestures, whereas the KinectV2 could do better to capture the full body actions with all joints [10]. The effective distances of KinectV2 and RealSense hardwares are different. The KnectV2 is optimized to 0.5m 4.5m, while RealSense are designed for 0.2m 1.2m depends on different devices.

1.2 Human Computer Interface

Gesture recognition is one of the hottest sustained research activities in the area of HCI [11]. It has a wide area of application including human machine interaction, sign language, immersive game technology *etc*. Being a significant part in non-verbal communication, hand gestures are playing vital role in our daily life. Hand Gesture recognition system provides us an innovative, natural, user friendly way of interaction with the computer. By keeping in mind the similarities of human hand shape with four fingers and one thumb, Meenakshi [12] presents a real time system for hand gesture recognition on the basis of detection of some meaningful shape based features like orientation, center of mass (centroid), status of fingers, thumb in terms of raised or folded fingers of hand and their respective location in image. Since gestures based on hand and finger movements can be robustly un-

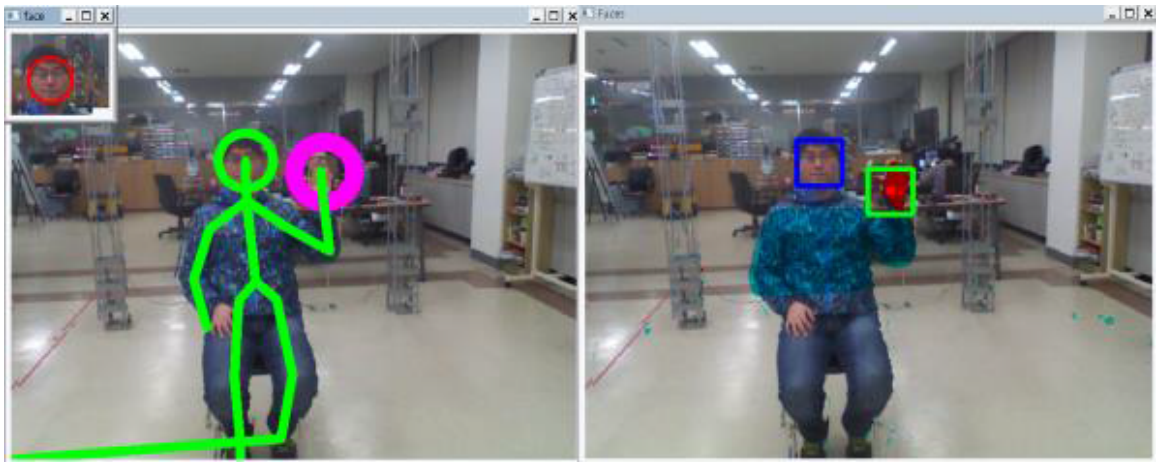


Figure 1.1: Calling Gesture Recognition Using Kinect [14]

derstood by computers by using a special 3D IR camera, users are allowed to play games and interact with computer applications in natural and immersive ways that improve the user experience.

Kam *et al.* [13] developed a real-time gesture-driven human computer interface using the KinectV1 camera and achieved close to 100% practical recognition rates. After Kam, a Kinect-based calling gesture recognition scenario is proposed by Xinshuang *et al.* [14] for taking order service of an elderly care robot. Its proposed scenarios are designed mainly for helping non expert users like elderly to call service robot for their service request. In order to facilitate elderly service, natural calling gestures are designed to interact with the robot. Figure 1.1 shows the evaluation of gesture recognition when sitting on chair. Individual subjects are segmented out from 3D point clouds acquired by Microsoft Kinect, skeletons are generated for each subject. And face detection is applied to identify whether the segment is human or not, and specific natural calling gestures are designed based on skeleton joints.

Dan *et al.* [11] proposed another smart and real-time depth camera based on a new depth generation principle. A monotonic increasing and decreasing function is used to control the frequency and duty-cycle of the NIR illumination pulses. The adjusted light pulses reflect off of the object of interest and are captured as a series of images. A recon-

figurable hardware architecture calculates the depth-map of the visible face of the object in real-time from a number of images. The final depth map is then used for gesture detection, tracking and recognition. Figure 1.2 shows an example extraction of hand skeleton. In 2013, Jaehong *et al*. [15] develop and implement a Kinect-based 3D gesture recognition system for interactive manipulation of 3D objects in educational visualization softwares.

1.3 Robust Vision

RGB-D cameras own great credits in mobile robotics, building dense 3D maps of indoor environments. Such maps have applications in robot navigation, manipulation, semantic mapping, and telepresence. Peter *et al*. [16] present a detailed RGB-D mapping system that utilizes a joint optimization algorithm combining visual features and shape-based alignment. Building on best practices in Simultaneous Localization And Mapping (SLAM) and computer graphics makes it possible to build and visualize accurate and extremely rich 3D maps with RGB-D cameras. Visual and depth information are also combined for view-

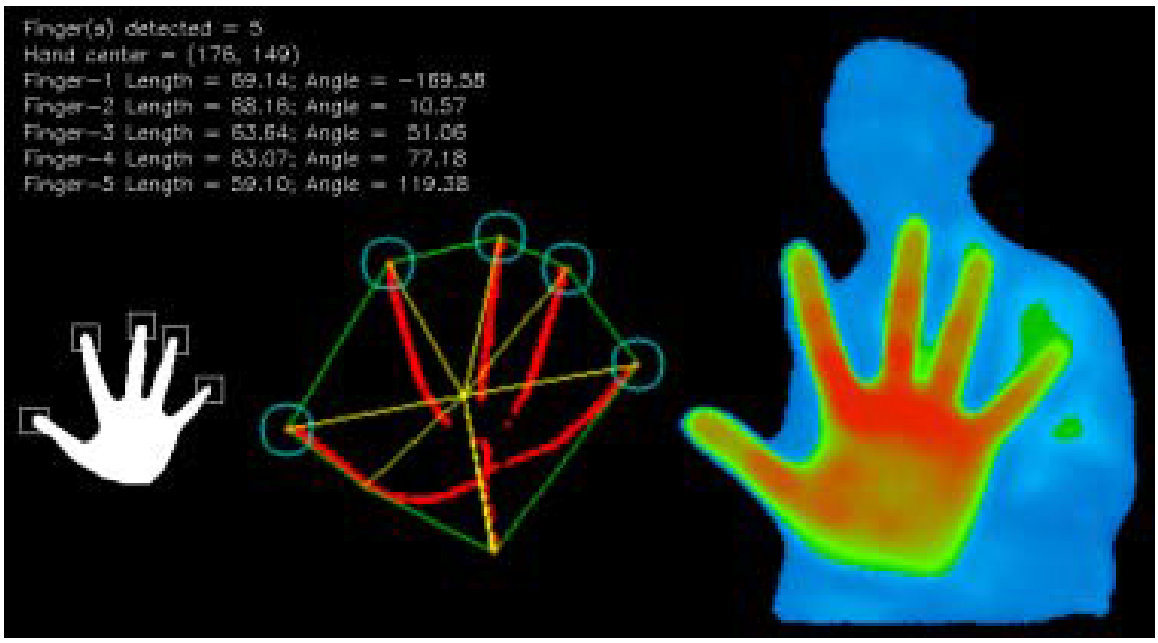


Figure 1.2: Finger Detection using Depth Data [11]

based loop closure detection, followed by pose optimization to achieve globally consistent maps. SLAM is the process of generating a model of the environment around a robot or sensor, while simultaneously estimating the location of the robot or sensor relative to the environment. SLAM has been performed in many ways, which can be categorized generally by their focus on localization or environment mapping [17]. SLAM systems focused on localizing the sensor accurately, relative to the immediate environment, make use of sparse sensor data to locate the sensor. Using range sensors such as scanning laser range-finders [18], LiDAR and SONAR [19], many robot applications use SLAM systems only to compute the distance from the sensor to the environment. SLAM systems focused on mapping use dense sensor output to create a high-fidelity 3D map of the environment, while using those data to also compute relative location of the sensor [20, 21]. Many modern SLAM algorithms combine both approaches, usually by extracting sparse features from the sensor and using these for efficiently computing the location of the sensor. This position is then used to construct a map from dense sensor data.

With a consumer RGB-D camera providing both color images and dense depth maps at full video frame rate, there appears a novel approach to SLAM that combines the scale information of 3D depth sensing with the strengths of visual features to create dense 3D environment representations, which is called RGB-D SLAM. Felix *et al.* [22] gives an open source approach to visual SLAM from RGB-D sensors, which extracts visual keypoints from the color images and uses the depth images to localize them in 3D. Maohai *et al.* [23] builds an efficient SLAM system using three RGBD sensors. As shown in Fig. 1.3, one Kinect looking up toward the ceiling can track the robot's trajectory through visual odometry method, which provide more accurate motion estimation compared to wheel motion measurement without being disturbed under wheel slippage. And the other two contiguous horizontal Kinects can provide wide range scans, which ensure more robust scan matching in the RBPF-SLAM framework.

Also using RGB-D sensor for SLAM, Kathia *et al.* [24] presents a constraint bundle

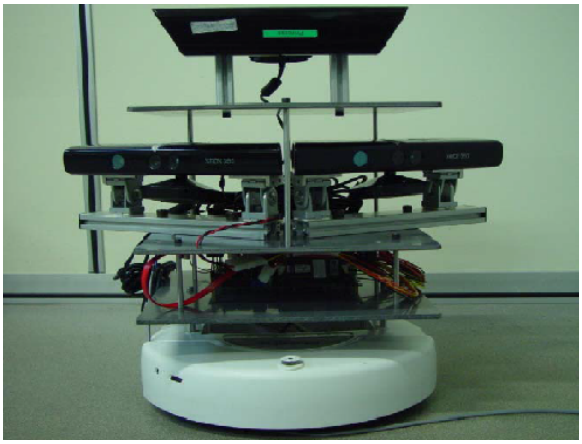
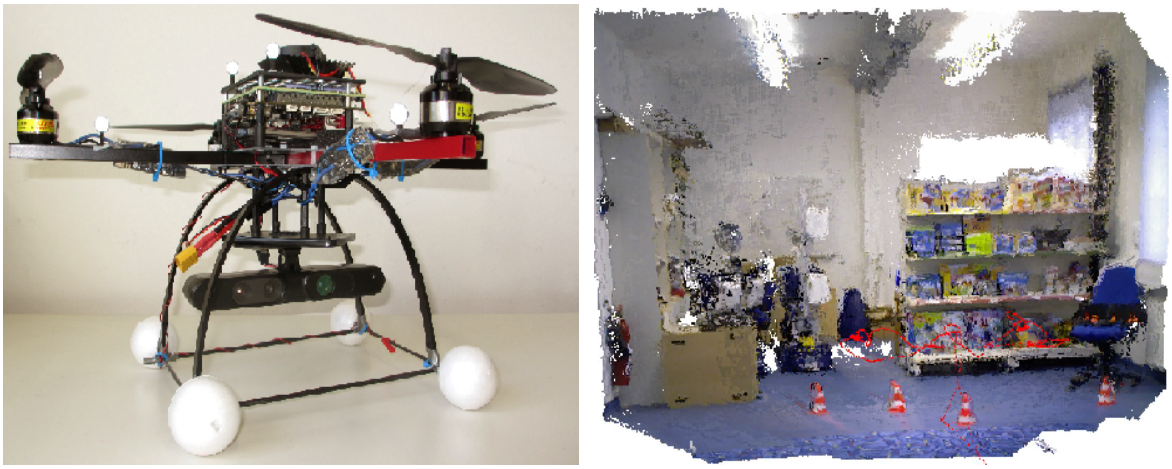


Figure 1.3: SLAM system with Only RGBD Cameras [23]

adjustment which allows to easily combine depth and visual data in cost function entirely expressed in pixel. In order to enhance the instantaneity of SLAM for indoor mobile robot, Guanxi *et al.* [25] proposed a RGBD SLAM method based on Kinect camera, which combined Oriented FAST and Rotated BRIEF (ORB) algorithm with Progressive Sample Consensus (PROSAC) algorithm to execute feature extracting and matching. ORB algorithm which has better property than many other feature descriptors was used for extracting feature. At the same time, ICP algorithm was adopted for coarse registration of the point clouds, and PROSAC algorithm which is superior than RANSAC in outlier removal was employed to eliminate incorrect matching. To make the result more accurate, pose-graph optimization was achieved based on General Graph Optimization (g2o) framework. Figure 1.4 shows the 3D volumetric map of the lab, which can be directly used to navigate robots.

RGB-D camera is also famous in application of doing visual odometry on autonomous flight of a micro air vehicle (MAV), helping acquire 3D models of the environment and estimate the camera pose with respect to the environment model. Visual odometry generally has unbounded global drift while estimating local motion. To bound estimation error, it can be integrated with SLAM algorithms, which employ loop closing techniques to detect when a vehicle revisits a previous location. A computationally inexpensive RGBD-SLAM

Figure 1.4: 3D Map of RGBD-SLAM with ORB and PROSAC [25]



(a) RGBD UAV (b) Reconstruction and Estimated Trajectory

Figure 1.5: RGBD-SLAM for Autonomous MAVs [26]

solution tailored to the application on autonomous MAVs is discussed by Sebastian and Andreas [26], which enables our MAV to fly in an unknown environment and create a map of its surroundings completely autonomously, with all computations running on its on-board computer. Figure 1.5a shows the MAC with an RGB-D sensor (the first generation of Kinect) mounted. And Fig. 1.5b shows the reconstruction based on the full point clouds, with the estimated trajectory shown in red dots.

1.4 3D Scanning and Printing

RGB-D sensors can be used on a much smaller scale than SLAM to create more detailed, volumetric reconstructions of objects and smaller environments, which opens a new world to the fast 3D printing. 3D printing is an additive technology in which 3D objects are created using layering techniques of different materials, such as plastic, metal, *etc*. It has been around for decades, but only recently is available and famous among the general public. The first 3D printing technology developed in the 1980's was stereolithography (SLA) [27]. This technique uses an ultraviolet (UV) curable polymer resin and an UV laser to build each layer one by one. Since then other 3D printing technologies have been introduced. Nowadays, some companies like iMaterialise or Shapeways offer 3D printing services where you can simply upload your CAD model on-line, choose a material and in a few weeks your 3D printed object will be delivered to your address. This procedure is quite straight-forward when you got your CAD model. However, 3D shape design tends to be a long and tedious process, with the design of a detailed 3D part usually requiring multiple revisions. Fabricating physical prototypes using low cost 3D fabrication technologies at intermediate stages of the design process is now a common practice, which helps the designer discover errors, and to incrementally refine the design [28]. Most often, implementing the required changes directly in the computer model, within the 3D modeling software, is more difficult and time consuming than modifying the physical model directly using hand cutting, caving and sculpting tools, power tools, or machine tools. When one of the two models is modified, the changes need to be transferred to the other model, a process we refer to as synchronization.

KinectFusion [29], a framework that allows a user to create a detailed 3D reconstruction of an object or a small environment in real-time using Microsoft Kinect sensor, has garnered a lot of attention in the reconstruction and modeling field. It enables a user holding and moving a standard Kinect camera to rapidly create detailed 3D reconstructions of an indoor scene. Not only an entire scene, a specific smaller physical object could also be

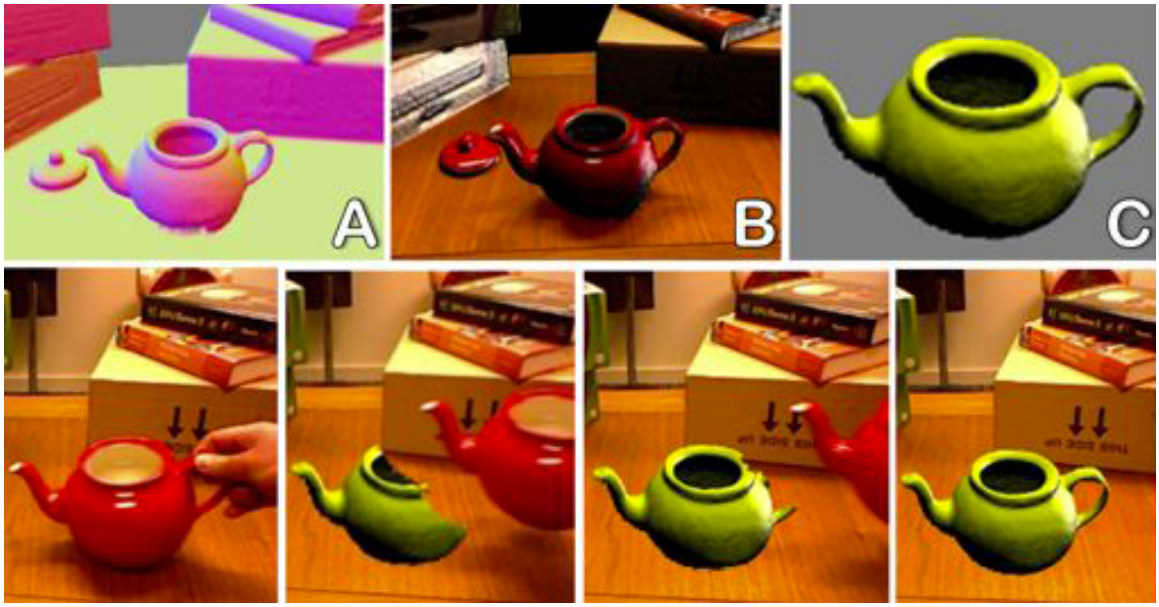
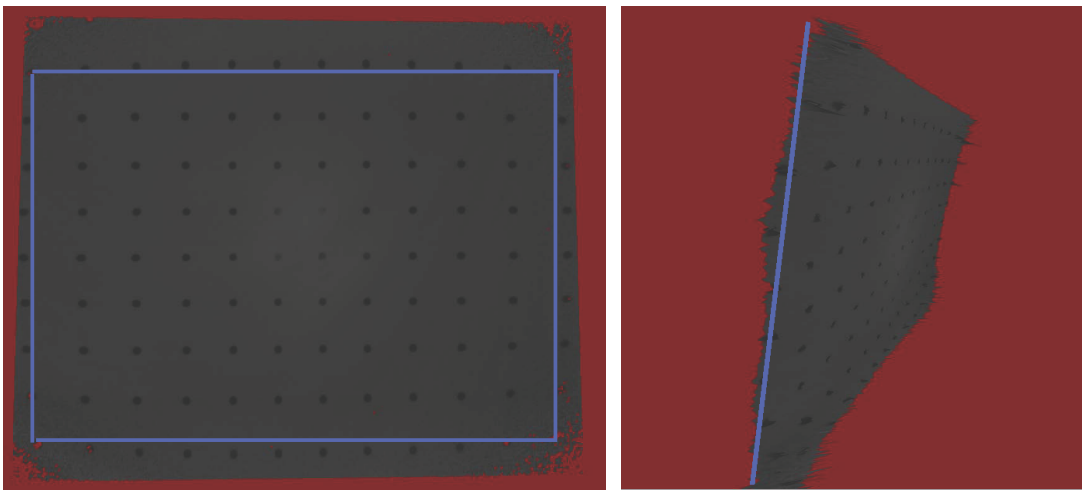


Figure 1.6: Object Segmentation in KinectFusion [29]

cleanly segmented from the background model simply by moving the object directly. Figure 1.6 shows how the interested object (a teapot) is accurately segmented from the background by physically removed. The sub-figure (A) shows surface normals, and sub-figure (B) is the texture mapped model. Nadia *et al.* [30] proposed and introduced a from-Sense-to-Print system that can automatically generate ready-to-print 3D CAD models of objects or humans from 3D reconstructions using the low-cost Kinect sensor. Further, Ammar and Gabriel [28] addresses the problem of synchronizing the computer model to changes made in the physical model by 3D scanning the modified physical model, automatically detecting the changes, and updating the computer model. A new method is proposed that allows the designer to move fluidly from the physical model (for example his 3D printed object, or his carved object) to the computer model. In the proposed process the physical modification applied by the designer to the physical model are detected by 3D scanning the physical model and comparing the scan to the computer model. Then the changes are reflected in the computer model. The designer can apply further changes either to the computer model or to the physical model. Changes made to the computer model can be synchronized to the physical model by 3D printing a new physical model.

1.5 RGB-D Cameras' Calibration and 3D Reconstruction on GPU

As discussed above, applications like RGBD-SLAM and KinectFusion apply 3D reconstruction techniques using an RGBD camera, in which an RGB sensor offers color values and a depth sensor measures objects' distances (D or Z^C). RGBD cameras, e.g. KinectV2, offer the horizontal and vertical field of view (FoV)s of the sensors based on a pinhole camera model, from which a proportional per-pixel beam equation (from Z^C to X^C/Y^C) could be derived. It is not hard to do 3D reconstruction in camera space naturally on GPU with the help of the proportional per-pixel beam equations; however, the 3D reconstructed image in that case will be deformed a lot by distortions. Figure 1.7 shows the KinectV2 NearIR 3D reconstruction in camera space, when observing a canvas hung on a flat wall printed with uniform grid ground-dots pattern. In the front view, a blue rectangle is drawn based on four corner dot-clusters, which reflects lens distortions (the uniformed distribution of the captured dot-clusters). While in the side view, a blue straight line added on the side of 3D reconstruction, which shows the unflatness of captured "flat wall". The deformation in the side view is probably caused by the various resolutions of depth sensor on per-pixel basis, which we will call as *depth distortion* in this thesis. In order to get undistorted 3D



(a) Front View (b) Side View

Figure 1.7: KinectV2 NearIR 3D Reconstruction in Camera Space

images, camera calibration is necessary before a camera being employed.

Camera calibrations usually use calibration objects, which could be assigned world space coordinates ($X^W/Y^W/Z^W$) to help remove distortions. For decades, much work on camera calibration has been done, starting from the photogrammetry community [31, 32], to computer vision ([33, 34, 35] to cite a few). And the combination of a pinhole-camera matrix with a distortion removal vector (which contains five high-order polynomial parameters) are widely known as important tools in camera calibration. However, there needs to be a lot calculations in the GPU fragment shader based on those parameters from both pinhole-camera matrix and distortion removal vector. And we would like to find a simple method with fewer calculations when generating the 3D coordinates. Similar with the per-pixel *proportional* beam equations in camera space reconstruction on GPU, Kai [36] derived more common *linear* beam equations (from X^W to Y^W/Z^W) directly from the pinhole-camera matrix, on the basis of per-pixel. That *linear* beam equations make it possible to show world space 3D reconstruction naturally on GPU, but it did not contains infos about lens distortion correction.

Our goal is to draw undistorted 3D reconstruction on GPU with the fewest calculations. Inspired by Kai, we will build up a rail calibration system to support the per-pixel D to Z^W mapping, such that Kai's per-pixel X^W to Y^W/Z^W linear mappings could be applied during the 3D reconstruction on GPU. We will call this new method *per-pixel calibration*. As shown in Fig. 1.8, the camera observing the uniform grid dots pattern is mounted on a rail, which is perpendicular to pattern on the wall. A laser distance measurer will be used to supply accurate per-frame Z^W, so that the per-pixel D to Z^W mapping could handle *depth distortion*. As long as the undistorted dense X^W/Y^W could be acquired, we will be able determine the parameters of per-pixel *linear* beam equations.

Undistorted world space coordinates $X^W/Y^W/Z^W$ with D together will be collected and saved onto local drives, during which lens distortions will be removed. Instead of using the combination of pinhole-camera matrix and distortion removal vector, we will determine a

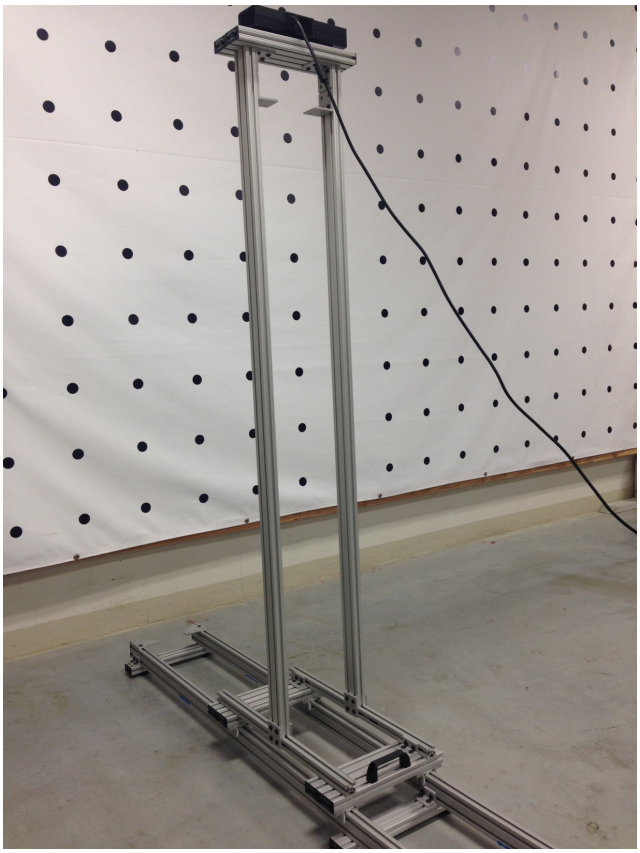


Figure 1.8: KinectV2 Calibration System

best-fit two–dimensional high-order polynomial mapping that can directly map from *Row* and *Column* in image space to X^W and Y^W, and the high-order polynomial mapping will handle the lens distortions. After the data collection, we will determine a best-fit mapping model between per-pixel D and Z^W, and then process the collected data, and finally generate per-pixel mapping parameters, which can make up a look-up table that will help draw undistorted 3D reconstruction on GPU in real-time.

In Chapter **??**, a pinhole-camera-model based calibration method is discussed in detail, including the lens distortions analysis and its removal. Chapter **??** will introduce how to draw the camera space 3D reconstruction on GPU, introduce a rail calibration system's set-up, and then talk about the proposed per-pixel calibration method and simple 3D reconstruction on GPU in detail. Chapter **??** will explain how the two polynomial mapping models in the proposed calibration method are determined, and then show the calibrated

results about how well the lens distortions and *depth distortion* are corrected. Chapter **??** will conclude this thesis and talk about the future work of RGB-D cameras calibration.

In line: RGBD3. In math mode:

$$RGBD3$$

List of Symbols

RGBD2 description here. 14

Bibliography

[1] Daniel Wigdor and Dennis Wixon. Brave NUI World: Designing Natural User Interfaces for Touch and Gesture. *1 edition*, Apr. 2011.

[2] Dean Takahashi. Beyond Kinect, PrimeSense wants to drive 3D sensing into more everyday consumer gear. *Venturebeat*, Jan. 2013.

[3] Krystof Litomisky. Consumer RGB-D Cameras and their Applications. University of California, Riverside. 2012.

[4] Csaba Kertesz. Physiotherapy Exercises Recognition Based on RGB-D Human Skeleton Models. *Modelling Symposium (EMS)*, Nov 2013.

[5] Manuel Gesto Diaz, Federico Tombari, Pablo Rodriguez-Gonzalvez, and Diego Gonzalez-Aguilera. Analysis and Evaluation Between the First and the Second Generation of RGB-D Sensors. *Sensors*, 15(11), Jul. 2015.

[6] Larry Li. Time-of-Flight Camera – An Introduction. *Texas Instruments Technical White Paper*, SLOA190B, 2012.

[7] Three Depth-Camera Technologies Compared. F. Pece and J. Kautz and T. Weyrich". *First BEAMING Workshop*, June 2011.

[8] Erica Ogg. Microsoft to acquire gesture control maker Canesta. *CNET*, Oct. 2010.

[9] Colleen C. Introducing the Intel RealSense R200 Camera (world facing). *Intel Developer Zone*, May 2015.

[10] Liberios Vokorokos, Juraj Mihalov, and Lubor Lescisin. Possibilities of depth cameras and ultra wide band sensor. *International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Jan 2016.

[11] Dan Ionescu, Viorel Suse, Cristian Gadea, and Bogdan Solomon. A Single Sensor NIR Depth Camera for Gesture Control. In *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, May 2014.

[12] Meenakshi Panwar. Hand gesture recognition based on shape parameters. In *2012 International Conference on Computing, Communication and Applications*, Feb 2012.

[13] Kam Lai, Janusz Konrad, and Prakash Ishwar. A gesture-driven computer interface using Kinect. In *Image Analysis and Interpretation (SSIAI)*, Apr 2012.

[14] Xinshuang Zhao, Ahmed M. Naguib, and Sukhan Lee. Kinect based calling gesture recognition for taking order service of elderly care robot. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, Aug 2014.

[15] Jaehong Lee, Heon Gul, Hyungchan Kim, Jungmin Kim, Hyoungrae Kim, and Hakil Kim. Interactive manipulation of 3D objects using Kinect for visualization tools in education. In *Control, Automation and Systems (ICCAS)*, Oct 2013.

[16] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments. In *Experimental Robotics*, 2014.

[17] Jodie Wetherall, Matthew Taylor, and Darren Hurley-Smith. Investigation into the Effects of Transmission-channel Fidelity Loss in RGBD Sensor Data for SLAM. In *International Conference on Systems, Signals and Image Processing (IWSSIP)*, Sep 2015.

[18] Henrik Kretzschmar, Cyrill Stachniss, and Giorgio Grisetti. Efficient information-theoretic graph pruning for graph-based SLAM with laser range finders. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep 2011.

[19] Maurice F. Fallon, John Folkesson, Hunter McClelland, and John J. Leonard. Relocating Underwater Features Autonomously Using Sonar-Based SLAM. *IEEE Journal of Oceanic Engineering*, 38(3), Jul 2013.

[20] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, and Andrew J. Davison. KinectFusion: Real-time dense surface mapping and tracking. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Oct 2011.

[21] Cheng-Kai Yang, Chen-Chien Hsu, and Yin-Tien Wang. Computationally efficient algorithm for simultaneous localization and mapping (SLAM). In *IEEE International Conference on Networking, Sensing and Control (ICNSC)*, Apr 2013.

[22] Felix Endres, Jurgen Hess, Nikolas Engelhard, Jurgen Sturm, Daniel Cremers, and Wolfram Burgard. An evaluation of the RGB-D SLAM system. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2012.

[23] Maohai Li, Rui Lin, Han Wang, and Hui Xu. An efficient SLAM system only using RGBD sensors. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec 2013.

[24] Kathia Melbouci, Sylvie Naudet Collette, Vincent Gay-Bellile, Omar Ait-Aider, Mathieu Carrier, and Michel Dhome. Bundle adjustment revisited for SLAM with RGBD sensors. In *IAPR International Conference on Machine Vision Applications (MVA)*, May 2015.

[25] Guanxi Xin, Xutang Zhang, Xi Wang, and Jin Song. A RGBD SLAM algorithm combining ORB with PROSAC for indoor mobile robot. In *International Conference on Computer Science and Network Technology (ICCSNT)*, Dec 2015.

[26] Sebastian A. Scherer and Andreas Zell. Efficient onbard RGBD-SLAM for autonomous MAVs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013.

[27] C. W. Hull. âĂIJApparatus for production of three-dimensional objects by stereolithography,âĂİ. *US Patent US4575330 A*, Mar 1986.

[28] Ammar Hattab and Gabriel Taubin. 3D Modeling by Scanning Physical Modifications. In *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, Aug 2015.

[29] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, and Richard Newcombe. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011.

[30] Nadia Figueroa, Haiwei Dong, and Abdulmotaleb El Saddik. From Sense to Print: Towards Automatic 3D Printing from 3D Sensing Devices. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, Oct 2013.

[31] D. C. Brown. Close-range camera calibration. *Photogrammetric Engineering*, 37(8), 1971.

[32] W. Faig. Calibration of close-range photogrammetry systems: Mathematical formulation. *Photogrammetric Engineering and Remote Sensing*, 41(12), 1975.

[33] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4), Aug 1987.

[34] Olivier Faugeras. Three-Dimensional Computer Vision. *MIT Press*, Nov 1993.

[35] Zhengyou Zhang. Camera Calibration. (Chapter 2). *Emergin Topics in Computer Vision*, 2004.

[36] Kai Liu. *Real-time 3-D Reconstruction by Means of Structured Light Illumination*. PhD thesis, University of Kentucky, 2010.