Dynamic 3D Scene Analysis and Modeling with a Time-of-Flight Camera

Dissertation

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
(Dr.-Ing.)
der Technischen Fakultät
der Christian-Albrechts-Universität zu Kiel

Dipl.-Ing. Ingo Schiller

Kiel
2011

1. Gutachter:	Prof. Dr.-Ing. Reinhard Koch
2. Gutachter:	Prof. Dr. Andreas Kolb

Datum der mündlichen Prüfung:	17.05.2011

Danksagung

Diese Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter in der Arbeitsgruppe "Multimediale Informationsverarbeitung“ von Prof. Reinhard Koch an der Christian-Albrechts-Universität zu Kiel. Ich möchte mich bei allen bedanken die mich während dieser Zeit in meiner Arbeit und meinem Privatleben unterstützt haben.

Meinem Doktorvater Prof. Reinhard Koch danke ich für die Möglichkeit unter seiner Betreuung zu promovieren, für die vielen spannenden Einblicke in die Welt der Bildverarbeitung, die wichtigen Impulse bei der Entwicklung der Arbeit und seine Führung der Arbeitsgruppe die ein stets hervorragendes Arbeitsklima ermöglichte. Dem Zweitgutachter Prof. Andreas Kolb von der Universität Siegen danke ich für seine Gutachter- und Prüfungs- Tätigkeit. Weiterhin danke ich den weiteren Mitgliedern der Prüfungskommission Prof. Manfred Schimmler und Prof. Reinhard von Hanxleden.

Den früheren Mitgliedern der Arbeitsgruppe Jan-Friso Evers-Senne, Felix Woelk, Daniel Grest und Kevin Köser danke ich für die gemeinsame Arbeit, die Einführung in die wissenschaftliche Arbeitsweise und für die vielen Anregungen und Hilfestellungen in den ersten Jahren. Christian Beder, Bogumil Bartczak und Falko Kellner danke ich für die konstruktive Zusammenarbeit von der diese Arbeit sehr profitiert hat. Den derzeitigen Mitgliedern der Arbeitsgruppe Daniel Jung, Anatol Frick, Sandro Esquivel, Anne Sedlazeck, Kristine Haase, Arne Petersen, Andreas Jordt, Robert Wulff, Lilian Zhang und Dan Chen danke ich ebenfalls für die angenehme und produktive Zusammenarbeit.

Meinen Eltern danke ich dafür, dass sie für meine gute Ausbildung gesorgt haben und damit das Fundament für meine weitere Entwicklung gelegt haben. Schließlich danke ich meiner Frau Christin Schiller für die Geduld, die Unterstützung und die Ermutigungen wärend des Verfassens der Promotionsschrift.

Abstract

Many applications in Computer Vision require the automatic analysis and reconstruction of static and dynamic scenes. Therefore the automatic analysis of three-dimensional scenes is an area which is intensively investigated. Most approaches focus on the reconstruction of rigid geometry because the reconstruction of non-rigid geometry is far more challenging and requires that three-dimensional data is available at high frame-rates. Rigid scene analysis is for example used in autonomous navigation, for surveillance and for the conservation of cultural heritage. The analysis and reconstruction of non-rigid geometry on the other hand provides a lot more possibilities, not only for the above-mentioned applications. In the production of media content for television or cinema the analysis, recording and playback of full 3D content can be used to generate new views of real scenes or to replace real actors by animated artificial characters.

The most important requirement for the analysis of dynamic content is the availability of reliable three-dimensional scene data. Mostly stereo methods have been used to compute the depth of scene points, but these methods are computationally expensive and do not provide sufficient quality in real-time. In recent years the so-called Time-of-Flight cameras have left the prototype stadium and are now capable to deliver dense depth information in real-time at reasonable quality and price. This thesis investigates the suitability of these cameras for the purpose of dynamic three-dimensional scene analysis. Before a Time-of-Flight camera can be used to analyze three-dimensional scenes it has to be calibrated internally and externally. Moreover, Time-of-Flight cameras suffer from systematic depth measurement errors due to their operation principle. This thesis proposes an approach to estimate all necessary parameters in one calibration step. In the following the reconstruction of rigid environments and objects is investigated and solutions for these tasks are presented. The reconstruction of dynamic scenes and the generation of novel views of dynamic scenes is achieved by the introduction of a volumetric data structure to store and fuse the depth measurements and their change over time. Finally a Mixed Reality system is presented in which the contributions of this thesis are brought together. This system is able to combine real and artificial scene elements with correct mutual occlusion, mutual shadowing and physical interaction.

This thesis shows that Time-of-Flight cameras are a suitable choice for the analysis of rigid as well as non-rigid scenes under certain conditions. It contains important contributions for the necessary steps of calibration, preprocessing of depth data and reconstruction and analysis of three-dimensional scenes.

Zusammenfassung

Viele Anwendungen des Maschinellen Sehens benötigen die automatische Analyse und Rekonstruktion von statischen und dynamischen Szenen. Deshalb ist die automatische Analyse von dreidimensionalen Szenen und Objekten ein Bereich der intensiv erforscht wird. Die meisten Ansätze konzentrieren sich auf die Rekonstruktion statischer Szenen, da die Rekonstruktion nicht-statischer Geometrien viel herausfordernder ist und voraussetzt, dass dreidimensionale Szeneninformation mit hoher zeitlicher Auflösung verfügbar ist. Statische Szenenanalyse wird beispielsweise in der autonomen Navigation, für die Überwachung und für die Erhaltung des Kulturerbes eingesetzt. Andererseits eröffnet die Analyse und Rekonstruktion nicht-statischer Geometrie viel mehr Möglichkeiten, nicht nur für die bereits erwähnten Anwendungen. In der Produktion von Medieninhalten für Film und Fernsehen kann die Analyse und die Aufnahme und Wiedergabe von vollständig dreidimensionalen Inhalten verwendet werden um neue Ansichten realer Szenen zu erzeugen oder echte Schauspieler durch animierte virtuelle Charaktere zu ersetzen.

Die wichtigste Voraussetzung für die Analyse von dynamischen Inhalten ist die Verfügbarkeit von zuverlässigen dreidimensionalen Szeneninformationen. Um die Entfernung von Punkten in der Szene zu bestimmen wurden meistens Stereo-Verfahren eingesetzt, aber diese Verfahren benötigen viel Rechenzeit und erreichen in Echtzeit nicht die benötigte Qualität. In den letzten Jahren haben die so genannten Laufzeitkameras das Stadium der Prototypen verlassen und sind jetzt in der Lage dichte Tiefeninformationen in vernünftiger Qualität zu einem vernünftigen Preis zu liefern. Diese Arbeit untersucht die Eignung dieser Kameras für die Analyse nicht-statischer dreidimensionaler Szenen. Bevor eine Laufzeitkamera für die Analyse eingesetzt werden kann muss sie intern und extern kalibriert werden. Darüber hinaus leiden Laufzeitkameras an systematischen Fehlern bei der Entfernungsmessung, bedingt durch ihr Funktionsprinzip. Diese Arbeit stellt ein Verfahren vor um alle nötigen Parameter in einem Kalibrierschritt zu berechnen. Im Weiteren wird die Rekonstruktion von statischen Umgebungen und Objekten untersucht und Lösungen für diese Aufgaben werden präsentiert. Die Rekonstruktion von nicht-statischen Szenen und die Erzeugung neuer Ansichten solcher Szenen wird mit der Einführung einer volumetrischen Datenstruktur erreicht, in der die Tiefenmessungen und ihr Änderungen über die Zeit gespeichert und fusioniert werden. Schließlich wird ein Mixed Reality System vorgestellt in welchem die Beiträge dieser Arbeit zusammengeführt werden. Dieses System ist in der Lage reale und künstliche Szenenelemente unter Beachtung von korrekter gegenseitiger Verdeckung, Schattenwurf und physikalischer Interaktion zu kombinieren.

Diese Arbeit zeigt, dass Laufzeitkameras unter bestimmten Voraussetzungen eine geeignete Wahl für die Analyse von statischen und nicht-statischen Szenen sind. Sie enthält wichtige Beiträge für die notwendigen Schritte der Kalibrierung, der Vorverarbeitung von Tiefendaten und der Rekonstruktion und der Analyse von dreidimensionalen Szenen.

Contents

Nomenclature

BFS	Breadth-First Search
BP	Belief Propagation
BRDF	Bidirectional Reflectance Distribution Function
BSP	Binary Space Partition
CCD	Charge-Coupled Device
CMOS	Complementory-Metal-Oxide-Semiconductor
CPU	Central Processing Unit
DLP	Digital Light Processor
DLT	Direct Linear Transform
DoF	Degree of Freedom
DP	Dynamic Programming
FoV	Field-of-View
GC	Graph Cuts
GPU	Graphics Processing Unit
HMI	Hierarchical Mutual Information
ICP	Iterative Closest Point
KLT	Kanade Lucas Tomasi
LBP	Loopy-Belief-Propagation
LDI	Layered Depth Image
LED	Light Emitting Diode
LIDAR	Light Detection And Ranging
MAD	Median Absolute Values
MoG	Mixture-of-Gaussians
MRF	Markov-Random-Fields
NCC	Normalized Cross Correlation
NIR	Near Infrared
pel	Picture Element (Pixel)
PMD	Photonic Mixing Device
PTU	Pan-Tilt Unit
RANSAC	Random Sample Consensus
RMS	Root Mean Square
SAD	Sum of Absolute Differences
SfM	Structure from Motion

SGM	Semi Global Matching
SL	Structured Light
SNR	Signal Noise Ratio
ToF	Time-of-Flight

List of Symbols

f_x, f_y	Focal length in x and y	4
c_x, c_y	Principle Point in x and y	4
K	Camera matrix	5
κ_1, κ_2	Radial distortion parameters	5
ρ_1, ρ_2	Tangential distortion parameters	5
R	Camera rotation matrix	6
C	Camera center	6
I_3	3×3 identity matrix	6
P	Projection matrix	6
λ	Wavelength	16
f_{mod}	Modulation frequency	16
λ_{mod}	Wavelength of modulation signal	16
c	Speed of light	17
L	Nonambiguity range	17
T_{int}	Integration time	17
ψ	Shift of the modulation signal	17
ω_{mod}	Angular modulation frequency	18
S	Correlation function	17
S_i	Samples of correlation function	18
φ	Phase shift of a PMD pixel	18
a	Correlation amplitude in PMD pixel	18
h	Incident light intensity in PMD pixel	18
d	Polar distance calculated per PMD pixel	18
I	Two-dimensional image, also $I(\boldsymbol{x})$	25
d^*	Wiggling corrected polar distance value d	27
d_c	Cartesian distance calculated from d	28
d_c^*	Wiggling corrected cartesian distance value	28
c_l	B-Spline controlpoints of wiggling depth error model	28
p_l	Polynomial wiggling depth error parameters	29
$\hat{h}$	Normalized reflectivity value	30
$\hat{d}_c$	Reflectivity and wiggling corrected cartesian depth	30
$\hat{d}$	Reflectivity and wiggling corrected polar depth	30
$\tilde{d}$	Filtered depth image value	50
G_σ	Gaussian kernel function	51
μ	Mean of a set of measurements	61
σ^2	Variance of a set of measurements	61
Σ	Covariance of a set of measurements	62
β	Background classification threshold for Mixture-of-Gaussian	62
γ	Update rate in Mixture-of-Gaussian	62

Δc	Mean color difference to background distributions	63
d_b^*	Background depth	65
Δd^*	Depth difference to background	65
Δd_n^*	Normalized depth difference Δd^*	65
Δc_n	Normalized color difference Δc	65
v_n	Normalized uncertainty	65
dv	Uncertainty filtered depth difference	65
cv	Uncertainty filtered color difference	65
α	Blending value for alpha matting	65
E	Expected value of a set of measurements	87
u	Signal amplitude	120
U	Electric current	120
C	Capacitor	120
c_a, c_b	Correlation functions	122

“Genius begins great works,
labor alone finishes them.”
Joseph Joubert (1754 -1824)

1 Motivation

The natural geometric world is a three-dimensional one. If time is taken into account it even is a four-dimensional one. If one wants to capture, reproduce and analyze this world all these dimensions have to be handled, otherwise the captured data, the analysis and the reproduction is incomplete. So more and more applications aim at including the third or fourth dimension. An example for an application which includes the third dimension is environment reconstruction, in which geometry and color of a static environment is reconstructed. More interesting and challenging are four-dimensional applications such as free-viewpoint video, in which the change of a scene is recorded and reproduced in all three dimensions over time. Visual media productions, such as three-dimensional television or mixed reality productions, are another challenging area in which real-time three-dimensional scene data is needed. Automotive applications, in which autonomous driving requires presence of continuous three-dimensional data, object recognition for robotic applications and human motion capture also rely on the availability of real-time three-dimensional scene information.

A fundamental requirement for such applications is the availability of real-time 3D-range measurements. Since many years the standard method to compute range information is the usage of a stereo-camera system consisting of two rigidly coupled standard cameras together with stereo algorithms. These algorithms search corresponding image points and use knowledge about the camera configuration to compute the distances of scene points to the cameras. For real-time applications efficient dense real-time stereo algorithms are needed. These algorithms consume a significant amount of CPU and/or GPU resources and suffer from unresolvable problems in sparsely textured scenes and at geometrical discontinuities. Besides stereo many other methods to acquire distance measurements exist, such as Laser Ranging, Structured Light approaches and others, which have been widely used and investigated. These approaches are capable to solve some of the problems stereo suffers from, but either require special hardware, a controlled environment or cannot handle dynamic scenes.

Within the last couple of years a new generation of active cameras has been developed based on the Time-of-Flight (ToF) principle. These so called Photonic Mixing Device (PMD)- cameras

[LSBS99] [XSH+98] emit modulated near infrared (NIR) light using LEDs with a modulation frequency in the range of several MHz, and measure the phase shift between the emitted modulated light and the received echo of the light using a special correlation sensor element. From the phase shift the distances of the scene points can be calculated. This new technique currently delivers dense depth maps at a resolution up to 204x204 pixel at frame rates up to 40Hz at no additional computational costs [KFM+04]. These cameras are suitable for range measurements in the near range, from ≈ 0.5 meters to ≈ 15 meters, depending on the modulation frequency. Different sensor and lens configurations with smaller opening angles and larger operating distance are also available and higher resolutions are already announced. The main advantage of these cameras is that they are mostly independent of the scene which is observed and the problems traditional depth measurement devices suffer from are widely reduced or eliminated. This new technique promises that sparsely textured areas and highly dynamic scenes are no longer a problem. Hence the topic of this thesis is the investigation of the usability of these new cameras for real-time dynamic 3D scene analysis.

The thesis will be structured in the following way. The following chapter 2 will introduce basic notation and the projective geometry and provides a review of the existing methods to acquire threedimensional data and discusses the advantages and disadvantages of the different approaches. Chapter 3 will introduce the principle of Time-of-Flight (ToF)- cameras and the calibration of ToF-cameras in combination with one or more conventional cameras, including multi-camera setups. Data preprocessing, image segmentation using ToF-cameras and data-structures for handling ToF-data are introduced in chapter 4. Scene analysis using a calibrated sensor of conventional- and ToF-cameras including pose estimation using depth and intensity is discussed in chapter 5. Chapter 6 shows the application of many of the presented methods in a real-time 3D Mixed-Reality System and finally, in chapter 7 the thesis is concluded. In this thesis I will refer to a rigidly coupled pair of ToF- and 2D- camera as a 2D/3D-camera.

"Everything has been said before,
but since nobody listens
we have to keep going back and
beginning all over again."
Andre Gide (1869 -1951)

2

Introduction

In this introductory chapter the notation used troughout the thesis is defined and the basics of projective geometry are introduced, followed by the discussion about traditional 3D scene acquisition methods and the contributions of this thesis.

2.1. Theoretical Basics

In this section the theoretical basics which are necessary to understand this thesis are introduced. It will start with the introduction of notations, continue with projective space and coordinates and introduce the perspective camera model used for color- and ToF-cameras.

2.1.1. Notation

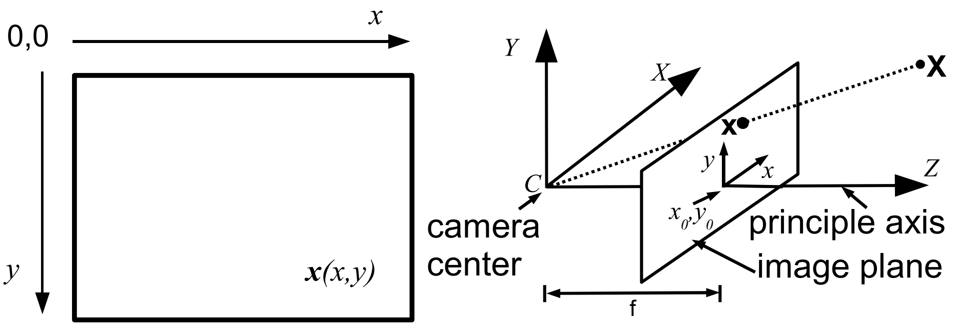
In this thesis I will use scalar values, two-dimensional image coordinates and points, two-dimensional projective points, three-dimensional points and matrices of different dimensions. The notation is summarized in table 2.1.

2.1.2. Projective Geometry

This section will introduce the projective geometry in short. The image coordinate system is defined as depicted in figure 2.1. The origin is located at the top left corner, x is to the right and y downwards. A point $\boldsymbol{x}(x, y)$ in an image, which is called a pixel (picture element), is a point of two dimensions in the euclidean space $\mathbb{R}^2$. The projective space $\mathbb{P}^2$ is constructed from the euclidean space by extending it by one dimension. This way an euclidean point $\boldsymbol{x}(x, y)$ is transformed in a homogeneous point $\mathbf{x}(x, y, w)$ in $\mathbb{P}^2$. The same is valid for euclidean spaces

Typeface	Description	Meaning
$abcdef$	normal italic	scalar values
$\boldsymbol{x}, \boldsymbol{y}$	small bold italic	two-dimensional point in $\mathbb{R}^2$
$\boldsymbol{X}, \boldsymbol{Y}$	capital bold italic	three-dimensional point in $\mathbb{R}^3$
$\mathbf{x}, \mathbf{y}$	small bold	homogeneous two-dimensional point in $\mathbb{P}^2$
$\mathbf{X}, \mathbf{Y}$	capital bold	homogeneous three-dimensional point in $\mathbb{P}^3$
$\mathit{\mathsf{R}}, \mathit{\mathsf{P}}$	capital italic	Matrices
K	capital straight	homogeneous Matrices

Table 2.1.: Typographic conventions



(a)

(b)

Figure 2.1.: Image coordinate system (a) and Pinhole camera geometry (b).

of higher dimensions. Especially the three-dimensional case is used in this thesis where a three-dimensional euclidean point $\boldsymbol{X}(x, y, z)$ in $\mathbb{R}^3$ is transformed into a homogeneous point $\mathbf{X}(x, y, z, w)$ in projective space $\mathbb{P}^3$.

Perspective Camera Model

As the ToF-camera uses traditional optics and a standard lens, the camera geometry is characterized by the standard camera matrix K (cf. [HZ04, p. 143], [MFW$^+$04, p. 229]):

$$\mathsf{K} = \begin{pmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.1}$$

Where f_x, f_y is the focal length in x and y, c_x, c_y is the principle point of the camera and s is the skew. Note that the skew is zero for most cameras and all cameras used in this thesis have zero skew.

A homogeneous 3D-point $\mathbf{X}_{cam}(x, y, z, 1)$ in the camera coordinate system is then projected to a homogeneous 2D point $\mathbf{x}$ on the image plane of the camera by:

$$\mathbf{x} = \mathsf{K}[\mathit{I}_3|\mathbf{0}]\mathbf{X}_{cam} \tag{2.2}$$

The ideal assumption that the imaging process is a linear one does not hold in practice. In general radial lens distortion is the most significant deviation from this assumption. In [HZ04] the lens distortion is modeled by a polynomial with at least degree two. In this work the lens distortion is modeled following the definition in [HS97] which uses radial (κ_1, κ_2) and tangential (ρ_1, ρ_2) distortion parameters which is also used in OpenCV [Ope10]. Before applying the radial undistortion the image points have to be normalized which is realized by applying the inverse camera matrix K^{-1}, which is equivalent to subtracting the principal point and dividing by the focal length:

$$\tilde{\mathbf{x}} = \mathsf{K}^{-1}\mathbf{x} = (\tilde{x}, \tilde{y}, 1) \tag{2.3}$$

The radially undistorted image coordinates $\hat{\boldsymbol{x}} = (\hat{x}, \hat{y})$ are then calculated with:

$$d_r = 1 + \kappa_1 r^2 + \kappa_2 r^4 \tag{2.4}$$

$$\begin{aligned} \hat{x} &= d_r\tilde{x} + \rho_1 2\tilde{x}\tilde{y} + \rho_2(r^2 + 2\tilde{x}^2) \\ \hat{y} &= d_r\tilde{y} + \rho_1 2\tilde{x}\tilde{y} + \rho_2(r^2 + 2\tilde{y}^2) \end{aligned} \tag{2.5}$$

with $r = \sqrt{\tilde{x}^2 + \tilde{y}^2}$ the current radial distance of the pixel $\tilde{\boldsymbol{x}}$ from the principle point. After applying the lens undistortion the normalization has to be inverted by applying the camera matrix K:

$$\mathbf{x} = \mathsf{K}\hat{\mathbf{x}} \tag{2.6}$$

Coordinate Transformation

The used cameras can also be located off the world coordinate system origin as depicted in figure 2.2. Taking the camera rotation R and translation $\mathbf{C}$ with respect to the world coordinate system into account a 3D point in world coordinates $\mathbf{X}$ is transformed to a 3D point in camera

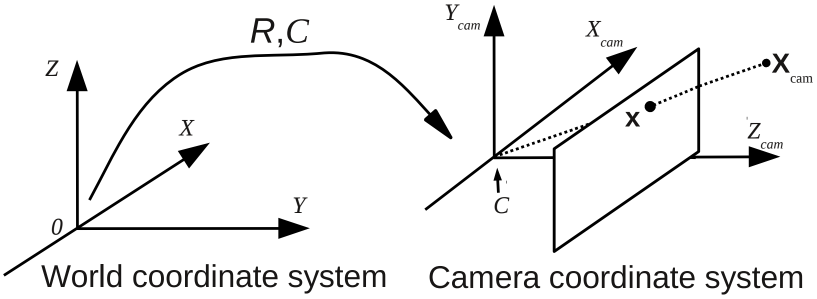


Figure 2.2.: Coordinate transformation from world to camera coordinate system.

coordinates $\mathbf{X}_{cam}$ by:

$$\begin{aligned} \mathbf{X}_{cam} &= R^{\mathsf{T}}[I_3| - \boldsymbol{C}]\mathbf{X} \\ \mathbf{X}_{cam} &= [R^{\mathsf{T}}| - R^{\mathsf{T}}\boldsymbol{C}]\mathbf{X} \\ \mathbf{X}_{cam} &= \begin{bmatrix} R^{\mathsf{T}} & -R^{\mathsf{T}}\boldsymbol{C} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix}\mathbf{X} \end{aligned} \tag{2.7}$$

Combining equation 2.2 with equation 2.7 results in:

$$\mathbf{x} = \mathsf{K}R^{\mathsf{T}}[I_3| - \boldsymbol{C}]\mathbf{X} \tag{2.8}$$

which leads to the projection Matrix P

$$\mathsf{P} = \mathsf{K}R^{\mathsf{T}}[I_3| - \boldsymbol{C}] \tag{2.9}$$

with K the camera matrix, R the rotation matrix, I_3 a 3×3 identity matrix and $\boldsymbol{C}$ the camera center. The projection matrix maps homogeneous 3D points $\mathbf{X}$ to homogeneous image coordinates $\mathbf{x}$ according to:

$$\mathbf{x} = \mathsf{P}\mathbf{X} \tag{2.10}$$

For every pixel $\mathbf{x}$ a corresponding ray $\mathbf{r}$ from the camera center through the pixel is calculated with:

$$\mathbf{r} = R\mathsf{K}^{-1}\mathbf{x} \tag{2.11}$$

The ray $\mathbf{r}$ is not normalized to units of focal length but can be of any length. Thus a normalization with $\sqrt{\mathbf{x}^T\mathsf{K}^{-T}\mathsf{K}^{-1}\mathbf{x}}$ is necessary in equation 2.12. Since the distance d of a scene point to the camera is measured in a pixel $\mathbf{x}$ by the ToF-camera, the homogeneous 3D point $\mathbf{X}$ can be calculated as:

$$\begin{aligned}\mathbf{X} &= d\frac{R\mathsf{K}^{-1}\mathbf{x}}{\sqrt{\mathbf{x}^T\mathsf{K}^{-T}\mathsf{K}^{-1}\mathbf{x}}} + \boldsymbol{C}\\ \mathbf{X} &= d\frac{\mathbf{r}}{\sqrt{\mathbf{x}^T\mathsf{K}^{-T}\mathsf{K}^{-1}\mathbf{x}}} + \boldsymbol{C}\end{aligned} \tag{2.12}$$

So for every pixel in a depth image a 3D point can be generated.

2.2. 3D Scene Acquisition Methods

Analyzing three-dimensional scenes is a research topic which has been widely worked on, and which is even more interesting nowadays as more and more real 3D applications are about to reach maturity such as for example 3D television. Most existing passive and active measurement techniques provide so-called depth-maps or depth images. These are images in which the distance of a scene point to the camera is stored, which is equivalent to the length of the ray from the camera center to the scene point. Obviously this is a 2.5D representation of the scene [Mar10], as the backside of the objects and possible occlusions are not included in the representation. Before the discussion of the contributions of this thesis I want to review existing techniques to acquire 3D scene data and evaluate them based on the following requirements:

1. Accuracy of measurement
2. Range of measurement
3. Ability to capture dynamics
4. Usability in real scenarios
5. Cost effectiveness

The most widely used method to acquire 3D data, at least in Computer Vision applications, is the usage of stereo algorithms which will be discussed in section 2.2.1. Less widely used, but comparably competitive are Structured Light approaches, which are introduced in section 2.2.2 and laser range scanners, so called LIDAR systems, which are discussed in section 2.2.3. All these methods provide 2.5D data in the form of depth images. A method to overcome this limitation and to record full 3D data is the combination of multiple sensors such as multiple cameras which observe the scene. These approaches are referred to as multi-camera systems and one of the techniques used is the multi-view stereo approach. Besides the isolated methods various combinations of methods have been proposed and discussed.

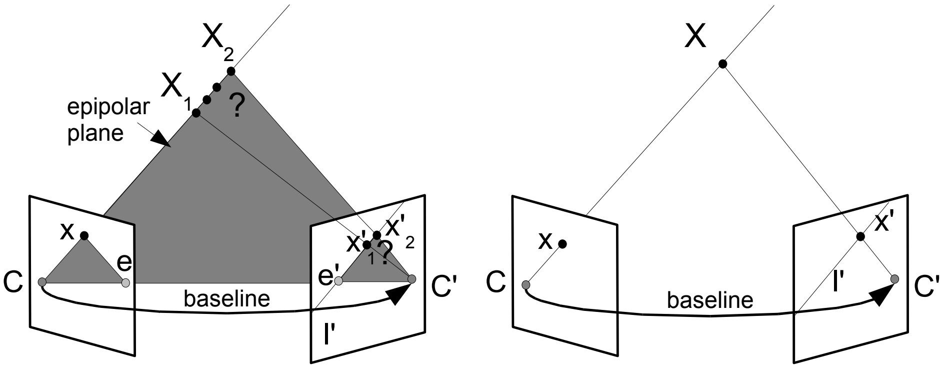


Figure 2.3.: The epipolar geometry: A 2D point $\mathbf{x}$, it's corresponding point $\mathbf{x}'$ and the corresponding 3D point $\mathbf{X}$ form a plane, the epipolar plane. From one camera it is not determinable on which depth a scene point is. In a second camera, the corresponding 2D point must lie on the epipolar line $\mathbf{l}'$. The correspondence has to be established and the corresponding 3D point is determinable. (After [HZ04])

2.2.1. Passive Stereo Approaches

Beginning in the 1970s two or more cameras are used together with stereo algorithms to estimate the distance of objects from the cameras. The developed and now mostly used stereo algorithms can roughly be categorized in three categories: The local correlation based approaches, semi-global approaches and global approaches.

The principle stereo setup to compute scene depth consists of two cameras observing the scene. The cameras are mounted with a certain distance to each other, called the baseline. If the internal camera parameters and the external transformation (rotation and translation) of the camera setup is known the distance of objects to the cameras can be estimated using the epipolar geometry constraints. Figure 2.3 shows the typical camera setup and the epipolar geometry.

A point $\mathbf{x}$ in image A, its corresponding point $\mathbf{x}'$ in image B and their common scene point $\mathbf{X}$ form a plane, the so called epipolar plane. The scene point $\mathbf{X}$ and the 2D point $\mathbf{x}$ are connected by the ray through $\mathbf{x}$ and the camera center $\mathbf{C}$. The distance of $\mathbf{X}$ to the camera center $\mathbf{C}$ is unknown. Knowing the epipolar geometry the corresponding point $\mathbf{x}'$ must lie on the epipolar line $\mathbf{l}'$ in image B which is the projection of the ray through $\mathbf{C}$ and $\mathbf{x}$. The search for correspondences in the images can therefore be limited to the epipolar line. If the correspondences are known, the disparity (= offset of corresponding points in image A and B in pixel) and distance of the scene points from the camera can be calculated.

Local stereo approaches are based on finding correspondences in images based on correlations of small image regions which are called matching windows. So for every image region the best matching window in the other image is searched for and the best match is taken as the corresponding point. From these matches the distance is calculated by triangulation. These methods assume constant distance to the camera within matching regions which is, for example at object boundaries, not correct. There are approaches to limit these errors, they can however not be avoided completely. A special local approach is the Dynamic Programming (DP) [VMVPVG02] approach in which correspondence search is executed pixel-wise on scanlines, which are defined by the epipolar geometry, as described in section 2.2.1. This avoids the false disparities at object boundaries but leads to other errors such as streaking.

Semi-global approaches such as Semi-Global Matching (SGM) [Hir05] use pixel-wise matching and an approximation of a global cost function with smoothness constraints. Global methods such as Graph Cuts (CG) [KZ01] or Belief Propagation (BP) [SZS03] are very memory intensive and require long computation times. For a review of stereo algorithms see [BBH03] and [SS02].

In [HS07] different correspondence and cost functions for stereo algorithms are compared concerning insensitivity to radiometric variations on the input images. In there Hirschmueller et al. compare six matching cost functions and three stereo methods. It is stated that local stereo methods usually use the SAD (Sum of Absolute Differences) over a small matching window to find correspondences while global methods compare values pixel-wise. The SAD is enhanced by using a sampling insensitive calculation of Birchfield and Thomasi. Other matching costs which are compared are image filters, Hierarchical Mutual Information (HMI) and Normalized Cross Correlation (NCC). The stereo methods compared are correlation stereo (Corr), Semiglobal Matching (SGM) and Graph Cuts (GC). The authors conclude that for the correlation based stereo (Corr) HMI and Rank-filters perform best and for the SGM and CG stereo the enhanced SAD is superior together with the HMI.

Multi-view stereo approaches have the goal of calculating depth images from multiple images and viewpoints and fusing these depth images to form a consistent model of the scene. The pairwise stereo is computed using one of the already discussed algorithms. The fusion of the depth maps in a model is carried out in different ways. The majority of approaches use voxels, polygon meshes or level-sets. A comparison of some state-of-the-art multi-view stereo algorithms for the purpose of reconstruction is given in [SCD$^+$06] in which six different methods are described and compared.

Using stereo to compute depth maps is a very flexible method to measure depth. It is usable in most environments as it is a passive system which is not affected by environmental influences such as direct sunlight. Taking stereo images is however not a simple task as the baseline (distance) of the camera centers has to be chosen in a way to fit to the scene depth. Illumination changes constitute a significant challenge as the cameras either have to automatically adapt shutter and gain, which have to be consistent in all cameras, or it has to be fixed which makes the system unusable in strongly changing conditions. Stereo also has significant problems in sparsely textured areas where no correspondences can be established. Assumptions for smoothness or planarity have to be made to fill these areas. Another problem are reflections on surfaces which cannot be detected. A significant problem that research concentrates on are discontinuities on which stereo systems tend to fail due to occlusions. In recent years, and due to the usage of the GPU, stereo algorithms have reached the real-time level on standard computer hardware making them usable for real-time applications. Still a significant amount of CPU/GPU power is used to calculate the depth maps. In [HS07] Hirschmueller reports a run-time between 20ms to 100ms for a 450×375 pixel image for the state-of-the-art approaches, while he reports 1.0s to 1.3s seconds in [Hir05] for SGM with HMI. For real-time reasons mostly local optimization methods are used. A comparison of these methods, run-times and accuracy of different cost functions on the GPU is found in [WGGY06]. Stereo approaches only require two conventional cameras and a standard computer for computation. This makes them very cost effective.

2.2.2. Structured Light Approaches

The most challenging part in depth computation with stereo systems is to find the correct correspondences between images. Especially in sparsly textured or untextured image regions correct correspondences are unavailable. Structured Light (SL) based approaches overcome this problem by projecting an artificial pattern on the scene to simplify correspondence search. This approach is used since many years. An early work on the color-encoded structured light approach was investigated by Boyer and Kak [BK87] in 1987. The principle of structured light depth acquisition is to project light in a known manner and pattern to an object and record the deformation of the projection with one or more cameras. In principle one of the cameras in the stereo setup is replaced by a projector and the search for correspondences is simplified by the projected pattern. Various patterns have been used, reaching from line patterns (this is for example used in conjunction with laser, see section 2.2.3) to color coded patterns. To project patterns to a surface either lasers or video projectors are used which are able to project a whole encoded pattern to a surface. To estimate the depth of the identified correspondence the external calibration of the camera relative to the projector has to be calibrated beforehand.

In [ZH04] high resolution (532×500) depth maps are computed using the structured light approach with a frame-rate of 40Hz using parallel implementation, a high speed digital light processor (DLP) projector and a high speed camera which both operate at 120Hz. The authors report an accuracy of $0.05mm$ RMS (root mean square) in an area of $266 \times 244mm$ which is a good result.

Approaches to include the temporal component in structured light systems exist, for example in [ZCS03] Zhang et al. compute a space-time stereo sequence using a structured light approach for correspondences. A comparable structured light approach for stereo is used in [SS03] to produce depthmaps to evaluate different stereo algorithms.

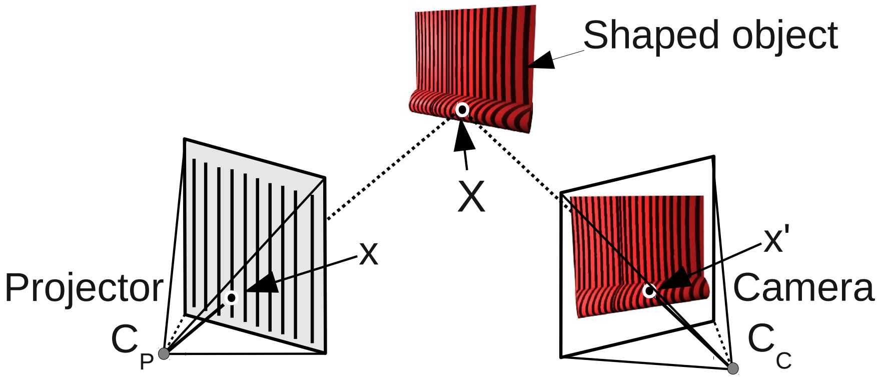


Figure 2.4.: The principle of structured light using a projected pattern. The pattern is projected with a projector to the surface of the scene while a camera observes the scene. For every 2D point $\mathbf{x}$ on the pattern it's corresponding point $\mathbf{x}'$ in the camera image is located and the 3D point $\mathbf{X}$ is calculated using triangulation.

For a long time Structured Light 3D scene acquisition systems have been too slow to capture dynamic scenes. Recent advances, such as described in [ZH04] seem to be able to cope with the real-time demand. The computational effort is less than for the stereo approaches as the search for correspondences is simplified. On the other hand SL approaches suffer from the need for special hardware as the projector and the cameras have to be synchronized and high-speed cameras have to be used if high framerates are required to capture dynamics. The method works best in controlled environments as a special hardware setup is required and the lighting conditions have to be controlled. The cost of SL approaches can be significant. One or more cameras and the projector are needed making SL approaches more expensive than stereo approaches. On the other hand is the accuracy much higher than the accuracy of stereo approaches, hence SL is often used as reference when comparing stereo methods. The Kinect camera[1] from Microsoft, which implements the reference design by PrimeSense[2], is a device offering real-time dense depth images using the Structured Light approach. It uses a pseudo-random pattern, projected onto the scene using a wavelength, which is not visible to the human eye, in the infrared spectrum. A monochrome CMOS sensor camera is used to capture this pattern. The Kinect camera was introduced in November 2010 and is offered at a very low price. It already has a very high impact on computer vision research.

[1] http://www.xbox.com/kinect/

[2] http://www.primesense.com

2.2.3. Active Depth Ranging

The difficult problem of correspondence search was simplified by the usage of artificial patterns projected onto the scene. A related approach is the range measurement using single light beams or lines for distance measurement. Because laser is characterized by a very narrow frequency spectrum and the parallelism of the emitted light, it is used in different wave lengths to measure distances to objects.

There are two different approaches of laser scanners, so called LIDAR- (Light Detection And Ranging) systems. The first class of laser scanner are the so called triangulation scanners which either use a point or a line laser. A camera is used to record the laser in the scene and triangulation algorithms are used to determine the distance of this point or line to the laser emitter. This is comparable to the Structured Light approaches.

The second class of laser scanners is based on the Time-of-Flight principle. It consists of a laser emitter, a rotating mirror and a receiver, either a solid state photo-detector or a photomultiplier, which measures the run-time of the laser beam and computes the distance of a scene point to the camera using the speed of light. An example of this principle is used by Yahav et al. in [YIM07]. The used laser scanner emits a square laser pulse of short duration. The reflection is measured with an imaging sensor (CCD). The emitted light pulse is reflected as a light wall in which 1mm depth difference corresponds to 7 picoseconds time-of-flight difference. Depending on the scene geometry different gray values are generated. The depth quality is limited by the bit depth of the CCD chip and the chosen depth window. Besides the computation of scene distance by Time-of-Flight the phase shift between the emitted and reflected signal is used to compute the scene depth. An early approach is investigated by Nitzan et. al in [NBD77], in which sinusoidal amplitude modulation is combined with a scanning mirror, a photomultiplier tube and a phase detector to detect the phase shift relative to the emitted signal which is relative to the distance.

While the Time-of-Flight lasers are used for higher distances because of the lower resolutions, the triangulation scanners are used for smaller distances and provide more accurate results. LIDAR systems typically use light with wavelengths in the ultraviolet, visible, or near infrared range. LIDAR systems are widely used in autonomous automotive applications, for example in [MBB^{+}08] and [TMD^{+}07], cultural heritage [EHBPG04] and large scale reconstruction [HYN03] [YHNF03] systems, as well as in satellites for topographical surveys.

The advantage of a laser scanner is the high distance which can be covered. The measurement accuracy is limited by the wavelength and the noise in the measurement data. Triangulation scanners are comparable to SL approaches concerning accuracy. On the other hand, laser scanning is a point- or line-based measuring method. This leads to measurement errors when measuring objects that are non-rigid or non-static. In this case the depth measurements tend to be blurred, especially at the object boundaries. Laser scanners for measuring distances up to 80m can be made eye-save, as for example the laser scanners (LMS211) from Sick[3]

[3] www.sick.de

used in [TMD+07]. High quality laser scanners are still very expensive, medium quality laser scanners are available at the cost of a good stereo camera system.

2.2.4. Conclusions

To conclude this chapter I want to summarize the mentioned 3D scene acquisition methods. First of all the overview is not complete. There are other means to acquire depth images, for example Shape-from-Shading, Shape-from-Defocus and various others. These are however less often used and less powerful and are therefore neglected in this comparison.

Stereo is a powerful distance measurement principle, but it requires a calibrated stereo camera system and sophisticated algorithms and a lot of processing power to produce real-time depth maps of acceptable quality. The latest state-of-the-art algorithms still fail for example in untextured areas in which no correspondences can be established and on images with highly repetitive texture. Furthermore object borders and occlusions are still sources of errors. Only assumptions about the scene can be made to fill holes but these assumptions do not always hold. Two active distance measurement principles were also discussed. The Structured Light approaches use a video projector or a laser to light objects and a camera to record the distortion of the projected light on the surface. This requires a lot of special hardware and a controlled environment. At least lighting has to be controlled. Recent advances in Structured Light approaches reach real-time and good quality. However, it is not always possible to control the environment and a method which works in all environments is desirable. The active laser ranging with continuous lasers reaches good quality but suffers from the fact that images are not taken at once but row-wise or even point-wise. The reconstruction of dynamic scenes is therefore very challenging and computational expensive as special algorithms for compensation of motion errors are needed.

The given overview evaluated the depth measurement techniques under various aspects. Stereo approaches suffer from high computational effort and difficult correspondence search. Structured Light approaches require significant hardware effort and controlled environment. The recording of dynamic scenes is only possible following significant restrictions. Laser-based methods only measure one point or line at a time and dynamic scenes are therefore difficult to capture. To model and analyze dynamic scenes based on depth information a depth computation method is required which is capable to deliver reliable real-time, image-at-once depth information, independent of scene geometry, texture and illumination. The ToF-cameras, which have recently become available at acceptable price and quality, promise to fulfill the desired requirements. In this thesis I will investigate these relatively new devices concerning calibration, reconstruction of rigid and non-rigid geometry and their suitability in a mixed reality system.

2.3. Contributions

This thesis introduces several contributions to the area of 3D scene analysis in conjunction with the usage of Time-of-Flight cameras:

1. Calibration of a ToF-camera: Using a ToF-camera as measuring device requires to calibrate this camera. I will introduce the calibration of external and internal camera parameters as well as a depth error calibration, compensating systematic measurement errors. The combination of ToF-cameras with additional cameras is essential and will also be covered in this contribution as well as multi-camera setups. This contribution was published in: "Calibration of a PMD camera using a planar calibration object together with a multi-camera setup" [SBK08] and "Time-of-Flight Sensor Calibration for Accurate Range Sensing" [LSKK10].

2. Rigid Reconstruction: ToF-cameras are predestined for the reconstruction of midsize environments. In this contribution it will be shown how ToF-cameras can be exploited to construct models of indoor environments with high accuracy. Additionally smaller objects can be reconstructed by estimating the camera pose and fusing multiple measurements. This contribution was published in: "Datastructures for Capturing Dynamic Scenes with a Time-of-Flight Camera" [SK09] and "Integration of a Time-of-Flight Camera into a Mixed Reality System for Handling Dynamic Scenes, Moving Viewpoints and Occlusions in Real-Time" [BSBK08].

3. 3D capturing and playback of dynamic scene content: How Space-Time Free Viewpoint Video is captured, stored and replayed is described in this contribution and was published in: "Datastructures for Capturing Dynamic Scenes with a Time-of-Flight Camera" [SK09].

4. Mixing of real and virtual content: This contribution uses the presented approaches to construct a mixed reality system including mutual occlusion and interaction with virtual objects. Published in: "Increasing Realism and Supporting Content planning for Dynamic Scenes in a Mixed Reality System Incorporating a Time-of-Flight Camera" [SBKK10], "Integration of a Time-of-Flight Camera into a Mixed Reality System for Handling Dynamic Scenes, Moving Viewpoints and Occlusions in Real-Time" [BSBK08] and "MixIn3D: 3D Mixed Reality with ToF-Camera" [KSB$^+$09].

"Any sufficiently advanced technology is indistinguishable from magic."

Arthur C. Clarke (1917 -2008)

3
2D/3D Camera Calibration

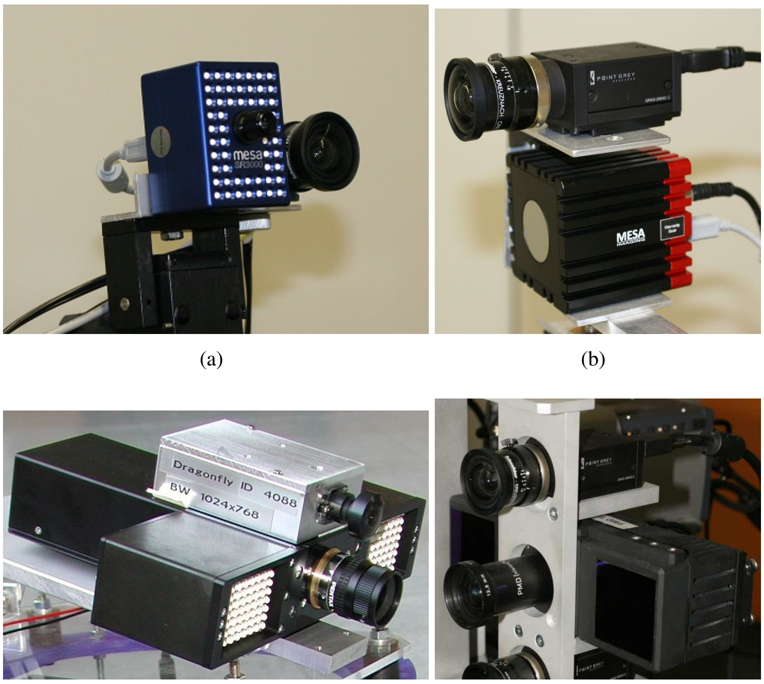
3.1. The Time-of-Flight Camera

In this section an overview of the physical operation principle of ToF- cameras based on the Photonic Mixer Device (PMD) is given. Four models of ToF-cameras are shown in figure 3.1. Each of them has a different resolution and field-of-view (FoV). The SwissRanger3000[1] features a resolution of 176×144 pixel and a FoV of $47.5° \times 39.6°$. The new Swissranger4000 has the same number of pixel, a FoV of $43.6° \times 34.6°$ but better SNR. The PMD[vision] 3k-S[2] has 64×48 pixel and a FoV of $40°$. The PMDTec CamCube has a resolution of 204×204 pixel and a FoV of $40° \times 40°$. The cameras offer both a range image with pixel-wise depth and a reflectance image with IR modulation intensities. Some additionally offer a low resolution intensity image as normal CCD cameras do.

Two different camera types have been used in this thesis, the Swissranger ToF-cameras, manufactured by Mesa-Imaging, and the PMD[vision] ToF-cameras manufactured by PMDTec. The differences are located on pixel level and concerning the demodulation and amplification. In the Swissranger cameras the measurement process is based on the so-called "Lock-In" method with a four-bucket solution, realized in CCD-/CMOS technique [SSVH95] [OLK$^+$03] [BL06] [BS08]. In the PMD[vision] cameras a slightly different approach [XSH$^+$98] [LSBS99] [Lua01] [Sch03] [KFM$^+$04] is used, it relies on two potential wells in which charge carriers are collected. To explain the measurement principle in detail I will concentrate in the Photonic Mixer Device (PMD) measurement principle used in the cameras of PMDTec.

[1] http://www.mesa-imaging.ch

[2] http://www.pmdtec.com



(a) (b)

(c) (d)

Figure 3.1.: Different models of Time-of-Flight cameras: SR3000 combined with high-resolution CMOS-camera (a), SR4000 combined with high-resolution CCD-camera (b), PMD[vision] 3k-S with CCD-camera (c) and CamCube with CCD-camera (d).

3.1.1. ToF-Measurement Principle

ToF-cameras are active cameras. They emit light and calculate the distance of objects to the cameras from the reflected signal. With a PMD-sensor the time of flight is computed from the phase shift between the emitted and the reflected signal in the sensor itself. Figure 3.2 shows the basic mode of operation. The illuminating units of the camera, consisting of special LEDs, emit intensity modulated near infrared light (NIR). The emitted signal s is reflected at the surface of objects and detected by the special pixel of the ToF-camera.

ToF-cameras use non-coherent modulated light for the measurement. The wavelength λ of the carrier signal, which is mostly chosen in the NIR spectrum (780 nm - 1400 nm), is not decisive as long as the wavelength of the modulation signal λ_{mod} is significantly longer. Typically a modulation frequency f_{mod} between 10 and 30MHz is chosen for distance measurement with

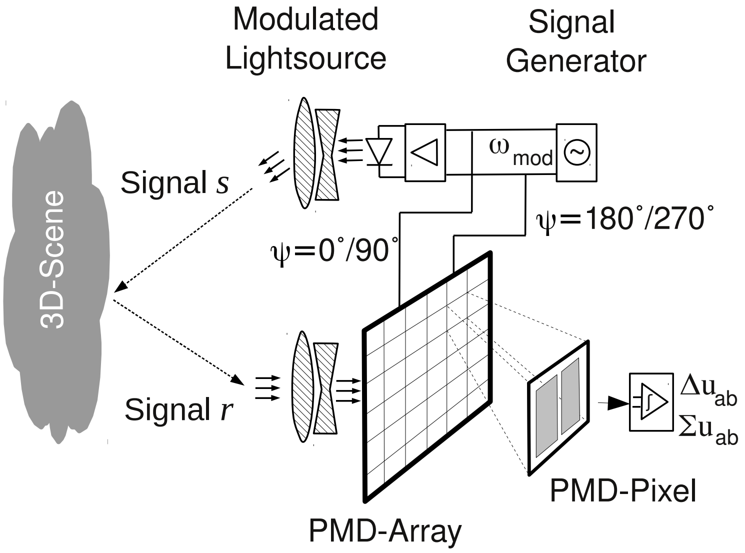


Figure 3.2.: The ToF-measurement principle with PMD-Sensor. (According to [Lua01]).

PMD-cameras. For a modulation frequency f_{mod} of 20MHz and $c \approx 299792458\frac{m}{s}$ the speed of light, the wavelength λ_{mod} results in:

$$\lambda_{mod} = c/f_{mod} = (299792458\frac{m}{s})/(20 \cdot 10^6 Hz) = 14.9896229m. \tag{3.1}$$

The measurement range L is limited by $\frac{\lambda_{mod}}{2}$, as with a phase shift bigger than $2L = \lambda_{mod}$ the phase delay is repeating and it cannot be distinguished whether it belongs to the first modulation period or a following one. The range L is called the nonambiguity range. The correlation between the detected optical signal r and the demodulation signal s is calculated in the smart pixel of the camera over the integration time T_{int}:

$$S = c(\psi) = r(t) \otimes s(t) = \lim_{T \to \infty} \frac{1}{T_{int}} \int_0^{T_{int}} r(t) \cdot s(t+\psi)\, dt. \tag{3.2}$$

with T_{int} the integration time, ψ an introduced phase delay of the reference signal and φ the phase shift of the reflected signal r. For s and r mostly sinusoidal signals are chosen:

$$s(t) = \sin(\omega_{mod} t) \tag{3.3}$$

$$r(t) = b + a\sin(\omega_{mod}t - \varphi) \tag{3.4}$$

with $\omega_{mod} = 2\pi f_{mod}$ the angular modulation frequency, b the background light and a the amplitude. This finally leads to:

$$S = c(\psi) = \frac{a}{2}cos(\varphi - \psi) \tag{3.5}$$

For further insights to the operation principle of the ToF-camera please consult appendix A.

Distance Calculation

By sampling the correlation function 3.2 four times:

$$\begin{aligned} &\psi_0 = 0^\circ, \psi_1 = 90^\circ, \psi_2 = 180^\circ, \psi_3 = 270^\circ \\ &S_i = c(\psi_i) \end{aligned} \tag{3.6}$$

which means taking four sequential images with an internal phase delay ψ_i and using simple trigonometry, it is possible to determine a pixel's phase shift φ between 0 and 2π, the correlation amplitude a and the incident light intensity h as

$$\begin{aligned} &\varphi = \mathrm{atan}\left(\frac{\mathrm{S}_3 - \mathrm{S}_1}{\mathrm{S}_0 - \mathrm{S}_2}\right) \qquad\qquad h = \frac{1}{4}\sum_{i=0}^{3} S_i, \\ &a = \frac{1}{2}\sqrt{(S_3 - S_1)^2 + (S_0 - S_2)^2}. \end{aligned} \tag{3.7}$$

The distance d to the corresponding object region is finally given by:

$$d = \frac{c}{4\pi\omega_{mod}}\varphi \tag{3.8}$$

where $c \approx 299792458\frac{m}{s}$ is the speed of light and ω_{mod} is the signal's angular modulation frequency. Following [Sch03] the decreasing accuracy with increasing distance can be explained with the SNR (Signal to Noise Ratio) of the Poisson-distributed number of electron carriers which are reflected by an object and measured by the ToF-camera. In the same way as the number of electron carriers decrease proportional to $\frac{1}{d^2}$ (d = distance), the SNR decreases, leading to increased measurement uncertainty and decreased accuracy. Rapp et. al [RFHJ08] provide a theoretical and experimental error analysis of the accuracy of the depth measurement. He shows that the variance of the depth error is approximately proportional to the squared distance.

Due to the active illumination with NIR light the FoV is limited and the resulting images are quite noisy and of low resolution. To overcome this shortcomings it is advisable to combine a ToF-camera with one or more high resolution CCD images as is depicted in figure 3.1.

This combination of ToF-cameras with standard CCD-cameras or other measurement devices is gaining more importance as the advantages of the absolute range measurement of the ToF-camera and the higher resolution of the CCD-camera can be combined. The accuracy of stereo systems and ToF-cameras was compared in [BBK07a] and a fused surface reconstruction was proposed in [BBK07b], which requires a reliable relative calibration of the stereo and the ToF-camera. In [KS06] a fusion of ToF and stereo was proposed, where the accuracy of the results is highly dependent on the quality of the calibration. The relative orientation of a ToF- and an optical camera for pose estimation is also required in [PMS$^+$08] and [SBKK07].

3.1.2. Accuracy Evaluation

Of more interest than the theoretical accuracy is the actual accuracy of the cameras as this defines the accuracy to which all applications are limited. The measurement accuracy is according to Schneider et. al [Sch03] in principle independent of distance, but according to Rapp et. al [RFHJ08] the distance accuracy is proportional to $1/d^2$. To evaluate this, average images with variance and standard deviations are calculated for different distances, modulation frequencies and integration times.

Figure 3.3 shows the variance analysis of the depth measurements with a SR4000 ToF-camera. In this experiment 50 images of a natural environment have been taken with the SR4000 camera with different integration times, different modulation frequencies and at different distances. The camera has been placed in front of a flat wall and the mean distance has been computed from 50 images. This is considered the true distance and the variances and standard deviations are given relative to this mean depth. The integration times used, which correspond to T_{int} of equations A.5, p.122 and A.6, p.122, are given for a complete depth image. The integration time for one sample image S_i (cf. equation 3.6) is therefore one forth of the given numbers. Using a modulation frequency of 30Mhz, as in the top of figure 3.3, limits the ambiguity range to ≈5m. Using lower modulation frequencies extends this ambiguity range to ≈10m as in figure 3.3 in the middle and ≈15m in figure 3.3 in the bottom.

The figures show that using the ideal and pre-calibrated modulation frequency of 30Mhz for the SR4000 camera results in the most reliable depth measurements with a standard deviation below 10mm till 3.5m object distance for higher integration times. Using lower integration times, the standard deviations increase rapidly above 4m object distance, suggesting that the camera is not suitable in such distances using this modulation frequency with low integration times. Higher integration times provide increased accuracy. The standard deviation of the measurements stays well below 10mm through the entire operation range for an integration time of 81.2ms.

Following Rapp et al. [RFHJ08] the standard deviation of the depth is approximately proportional to the squared distance.

$$\sigma_d \propto d^2 \tag{3.9}$$

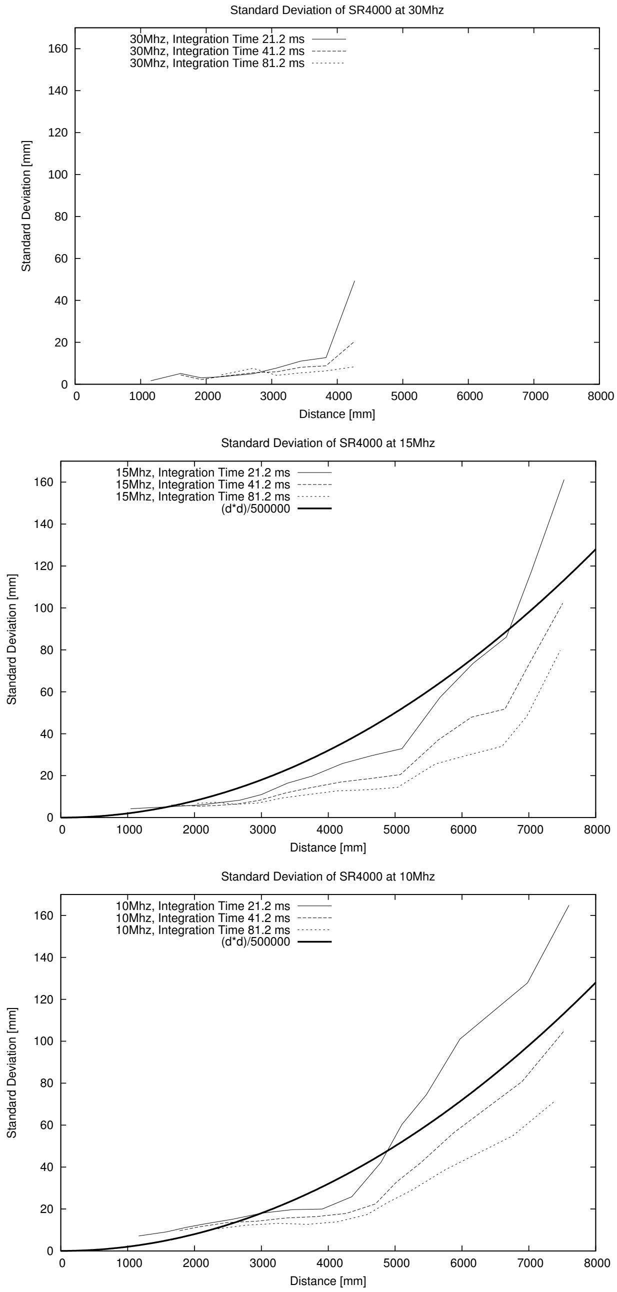


Figure 3.3.: Accuracy analysis of the SR4000 camera at 30MHz (top), 15Mhz (middle) and 10Mhz (bottom) modulation frequency and different integration times.

In the figures 3.3 and 3.4 the standard deviation is plotted and therefore a behavior proportional to the squared distance is expected. The squared distance, scaled with $1/500000$, is additionally shown in the plots for comparison. The results show, that the assumption of proportionality to the squared distance is approximately true. At small distances the standard deviation is increasing slower and at higher distances it is increasing faster. However, the increased measurement uncertainty with increasing distance is obvious.

Figure 3.4 shows the analysis of the standard deviation of depth measurements using the CamCube 2.0 from PMDTec. This camera can be operated using modulation frequencies of 19, 20 and 21 Mhz so a direct comparison to the SR4000 using the same modulation frequencies is not possible. The measured standard deviations of the CamCube are however comparable to the values of the SR4000. The standard deviation is small for small distances and increasing faster with growing distances. The values are approximately the same for all modulation frequencies, so it is not necessary to favor one frequency over another which makes the CamCube 2.0 suitable for multi-camera setups where different modulation frequencies have to be used.

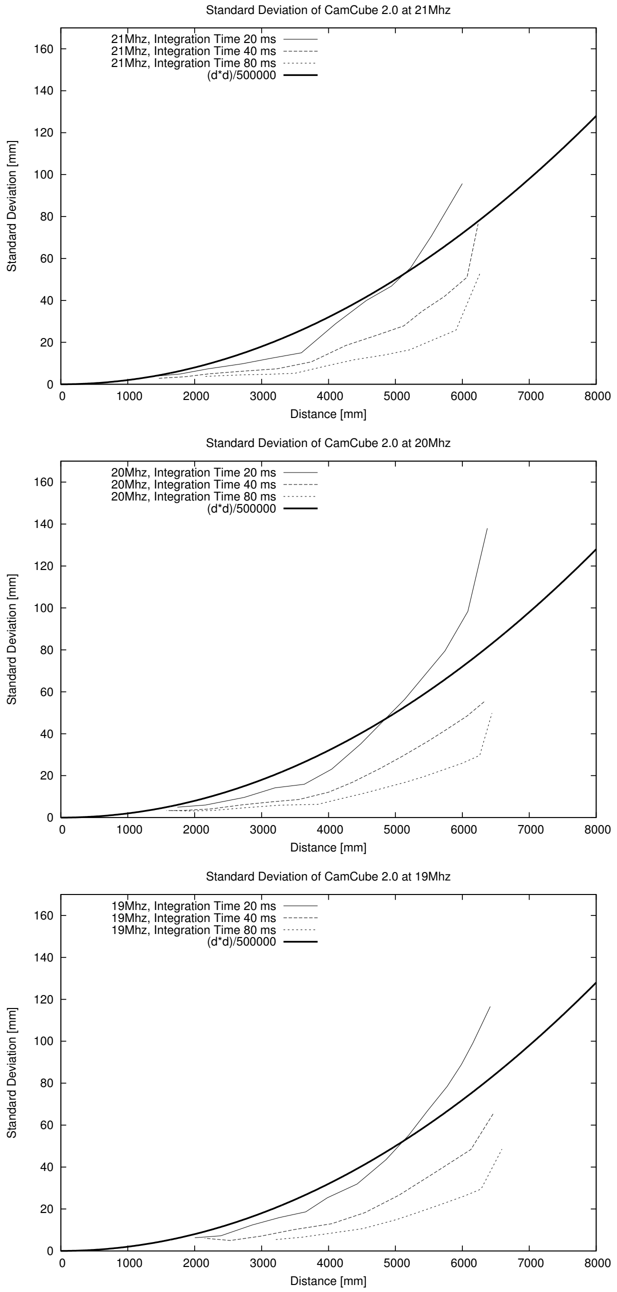


Figure 3.4.: Accuracy analysis of the CamCube 2.0 camera at 21 (top), 20Mhz (middle) and 19MHz (bottom) modulation frequency and different integration times.

3.2. ToF-Camera Calibration

An important issue when using ToF-cameras is the internal and external calibration of such cameras. Not only the standard perspective camera calibration including focal length, radial distortion and principle point is necessary but the calibration of the deviation of the depth measurement from the real depth. Different error sources for depth measurement errors are present, the most significant error is a systematic "wiggling" error caused by the imperfect sinusoidal modulation of emitted light.

3.2.1. Literature

Since ToF-cameras are used for Computer Vision applications the calibration of such devices is mandatory for correct measurement results. Previous approaches to calibrate ToF-cameras were described by Kuhnert et al. [KS06], Kahlman et al. [KRI06] and Lindner and Kolb [LK06]. The latter both use the calibration method of Zhang [Zha99] on the reflectance images to estimate internal and external orientation which is implemented in the OpenCV library [Ope10]. The depth image is not used for pose estimation, because the scope of those works also is the depth calibration of the cameras. The authors conclude that the poor quality of the low resolution reflectance images makes precise localization very difficult. Further calibration approaches are also found in [PHW$^+$06], [SF08] and [KCTT08]. While most calibration work with ToF-cameras only covered the calibration of single ToF-cameras the work of Kim et al. in [KCTT08] also discusses the calibration of multiple ToF-cameras. Tables 3.1 and 3.2 show a comparison of the used methods for calibration. The column "Pattern" describes what kind of calibration pattern is used, the column "Internal" specifies which approach is used to estimate the internal camera parameters and the column "Distance" specifies how the reference measurements for the distance calibration are obtained. The column "Depth Model" states what kind of error function is used to compensate depth measurement errors, the column "Para. Est." stands for the method the parameters are estimated and the last column tells whether a combination with one or more standard cameras is included in the calibration.

3.2.2. Calibration Approach

As stated in chapter 3.1 it is advisable and common to combine a ToF-camera with at least one other standard camera. In this section, I address the exact calibration of a ToF-camera in combination with at least one standard CCD-camera. This new calibration approach constitutes one of the contributions of this thesis as it is a novel approach for calibrating focal length, principle point, lens distortion and depth calibration for ToF- and standard cameras in a joint method. Simultaneously the relative external transformation between the ToF- and CCD-cameras is estimated and high accuracy is achieved by using multiple images from each

	Pattern	Internal	Distance
Kuhnert et al. [KS06]	Chessboard	not specified	not specified
Lindner et al. [LK06][LK07]	Chessboard	OpenCV [Zha99]	Trackline
Kahlman et al. [KRI06]	Pattern of NIR LEDs	Bundle Adjustment	Trackline
Fuchs et al. [SF08] [FH08]	Chessboard	OpenCV	Robot
Prasad et al. [PHW+06]	Chessboard	OpenCV	Measured
Kim et al. [KCTT08]	Chessboard	MATLAB Toolbox[Bou99]	MATLAB Toolbox + Angular Refinement
Schiller et al. [SBK08]	Chessboard	OpenCV + Refinement	OpenCV + Refinement

Table 3.1.: Comparison of calibration approaches for ToF-cameras I.

	Depth Model	Para. Est.	Comb. 2D/3D
Kuhnert et al. [KS06]	2nd Order Polynomial	not specified	No
Lindner et al. [LK06][LK07]	B-Spline + per pixel fixed pattern offset	not specified	No
Kahlman et al. [KRI06]	LUTs (different reflectivities) + per pixel fixed pattern offset	not specified	No
Fuchs et al. [SF08] [FH08]	Polynom/Splines (Wiggling error) + Linear (fixed pattern noise)	Nonlinear Least Squares	No
Prasad et al. [PHW+06]	Linear	not specified	Yes, Image Multiplier
Kim et al. [KCTT08]	6th Order Polynomial	not specified	Yes, Calibrated separately
Schiller et al. [SBK08]	3rd Order Polynomial or B-Spline	Nonlinear Least Squares	Yes, Calibrated simultaneously

Table 3.2.: Comparison of calibration approaches for ToF-cameras II

camera. This method uses the reflectance- and depth images provided by a ToF-camera and the reflectance images from the standard CCD-cameras. The reflectance image of the ToF-camera is also refered to as "amplitude"- or "modulation coefficients"- image in the literature. An analysis-by-synthesis approach is used in combination with non-linear optimization for parameter estimation.

Section 3.2.2 introduces the calibration approach and section 3.2.3 the depth calibration models. How the internal and external camera parameters are estimated is shown in section 3.2.5 using all images taken of a planar checkerboard calibration pattern. A new reflectivity calibration approach is presented in section 3.2.4. Finally the results are discussed in section 3.2.6.

In the calibration process the internal camera parameters for the single cameras as well as the external parameters are estimated. Typically 30 to 80 images per camera are used to calibrate a camera rig. A rigidly coupled camera rig is assumed and only the absolute external camera parameters of the first camera in the rig are estimated. The positions and rotations of the other cameras are estimated relatively to this first camera. E. g. with four cameras and 20 images per camera 20 absolute external camera parameters (20x translation and rotation) and three (3x translation and rotation) relative external camera parameters are estimated.

The cameras will from now on carry the indexes $k, k \in \{1, ..., K\}$, the images per camera will be indexed with indexes $j, j \in \{1, ..., M\}$ and a pixel in an image I is indexed with $i, i \in \{1, ..., N\}$.

The initial estimation of the external orientations and internal parameters is based on correspondences between a reference object (here a planar checkerboard pattern) and points in the camera reflectance images. The checkerboard is detected automatically if possible, otherwise the corners of the checkerboard have to be selected manually. Assuming that the checkerboard is located at the $Z = 0$ plane 2D-3D point correspondences can be established and the initial internal and external camera parameters are computed using standard computer vision methods from OpenCV [Ope10].

Estimation Model

The geometric camera model assumed follows the definition in section 2.1.2, p.4 equation 2.1 and 2.8. Note that one camera matrix K_k is used per camera. The nonlinear lens distortion is modeled by the radial and tangential parameters $(\kappa_1, \kappa_2, \rho_1, \rho_2)$. From the initial camera parameter computation an estimate for principal point $(c_{x,k}, c_{y,k})$, shear s_k, focal lengths $(f_{x,k}, f_{y,k})$ and nonlinear distortion parameters is available. For the ToF-camera depth images are available, so that for each pixel $\mathbf{x}_{ijk}$ in a ToF-image the corresponding distance $d_{ijk}(p_l)$ to the camera center is known. Note that the depth is dependent on the depth deviation parameters $p_l, l \in \{0, ..., 5\}$ (see section 3.2.3, p.27). Hence, for each pixel the corresponding 3D point

$\mathbf{X}_i$ can be computed as follows:

$$\mathbf{X}_i = d_{ijk}(p_l)\frac{\mathsf{R}_{jk}\mathsf{K}_k^{-1}\mathbf{x}_{ijk}}{\sqrt{\mathbf{x}_{ijk}^{\mathsf{T}}\mathsf{K}_k^{-\mathsf{T}}\mathsf{K}_k^{-1}\mathbf{x}_{ijk}}} + \boldsymbol{C}_{jk} \tag{3.10}$$

The assumption is made that the planar calibration object is located at the $Z = 0$ plane. So one can derive one constraint per pixel by requiring $X_z = 0$, with X_z the z-component of $\mathbf{X}_i$. Using equation (3.10) this can be expressed as:

$$d_{ijk}(p_l)\boldsymbol{r}_{z,jk}\mathsf{K}_k^{-1}\mathbf{x}_{ijk} + C_{z,jk}\sqrt{\mathbf{x}_{ijk}^{\mathsf{T}}\mathsf{K}_k^{-\mathsf{T}}\mathsf{K}_k^{-1}\mathbf{x}_{ijk}} = 0 \tag{3.11}$$

or equivalent:

$$d_{ijk}(p_l) = -\frac{C_{z,jk}\sqrt{\mathbf{x}_{ijk}^{\mathsf{T}}\mathsf{K}_k^{-\mathsf{T}}\mathsf{K}_k^{-1}\mathbf{x}_{ijk}}}{\boldsymbol{r}_{z,jk}\mathsf{K}_k^{-1}\mathbf{x}_{ijk}} = f_{ijk}^{(1)} \tag{3.12}$$

in which $\boldsymbol{r}_{z,jk} = (r_x, r_y, r_z)^{\mathsf{T}}$ (the last column of the rotation matrix R_{jk}) and $C_{z,jk}$ = z-component of $\boldsymbol{C}_{jk}$.

The depth constraint using only the planar reference object is obviously not sufficient for calibration. Hence the reflectance image has to be used as well. The second constraint that can be derived is, that the reflectance is assumed to be equal to the known reflectance of the reference object:

$$I_{ref}(\mathbf{X}_i) = I_{ref}\left(d_{ijk}(p_l)\frac{\mathsf{R}_{jk}\mathsf{K}_k^{-1}\mathbf{x}_{ijk}}{\sqrt{\mathbf{x}_{ijk}^{\mathsf{T}}\mathsf{K}_k^{-\mathsf{T}}\mathsf{K}_k^{-1}\mathbf{x}_{ijk}}} + \boldsymbol{C}_{jk}\right) = I(\mathbf{x}_{ijk}) \tag{3.13}$$

or by substituting $d_{ijk}(p_l)$:

$$I(\mathbf{x}_{ijk}) = I_{ref}\left(\boldsymbol{C}_{jk} - \frac{C_{z,jk}\mathsf{R}_{jk}\mathsf{K}_k^{-1}\mathbf{x}_{ijk}}{\boldsymbol{r}_{z,jk}\mathsf{K}_k^{-1}\mathbf{x}_{ijk}}\right) = f_{ijk}^{(2)} \tag{3.14}$$

in which I denotes the reflectance image and I_{ref} denotes the reference image which is the image of the smoothed checkerboard pattern.

For the ToF-camera both constraints (3.12) and (3.14) can be used to estimate the external orientation, the internal camera parameters and the depth deviation parameters. In the case of the standard CCD-cameras only the second constraint (3.14) can be used. I will now show how to use these constraints to precisely calibrate the external and internal camera parameters as well as the depth calibration of the ToF-camera in a joint approach. Figure 3.5 shows two input

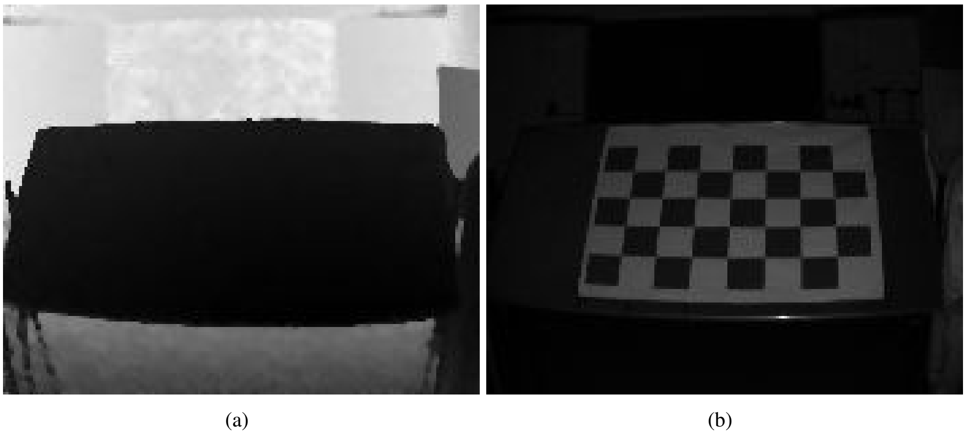


Figure 3.5.: Depth image (a) and gray value image (b) from SR4000-camera. Darker areas in (a) are closer to the camera.

images to the calibration procedure from a ToF-camera. On the left the depth measurement and on the right a reflectance image is shown.

3.2.3. Depth Calibration Model

The depth measurement with ToF-cameras suffers from systematic errors [SF08] which is mainly caused because the modulation of the light is not perfectly sinusoidal. The error is not only a constant offset but a higher order function [LK06]. Different error models are possible to model the depth deviation. In [SBK08] I proposed to use a polynomial model, but it is also possible to use a B-Spline model as proposed in [LK06] or to use look-up tables which requires intensive calibration effort. The combination of the Analysis-by-Synthesis approach and the B-Spline depth deviation model was published in collaboration with Lindner et al. in [LSKK10]. In this thesis I will introduce both models and compare them. In the following d is again the measured depth, d^* the corrected depth and x, y are the image coordinates. The depth images provided by the ToF-cameras encode ray lengths from the camera center to the 3D scene point which I will refer to as polar distance d. Therefore the wiggling error compensation function cannot be applied to the raw depth image data without considering the position in the image as this would change ray lengths at image borders more than in the image center and therefore distort scene geometry. Instead of considering the radial distance of a pixel to the principle point in the depth wiggling compensation function the depth images can be transferred into a Cartesian representation with Cartesian distance d_c:

$$
\begin{aligned}
d_c &= cos(\alpha_x) \cdot cos(\beta_{xy}) \cdot d \\
\alpha_x &= atan((x - c_x)/f_x) \\
\beta_{xy} &= atan\sqrt{(c_y - y)/(((x - c_x)\frac{f_y}{f_x})^2 + f_y^2)}
\end{aligned}
\tag{3.15}
$$

In the following the depth values will always be distorted and undistorted in Cartesian coordinates. After distortion or undistortion the Cartesian depth is transformed to depth in polar coordinates by applying the inverse of equation 3.15.

B-Spline Depth Error Model

Cubic B-Spline curves are very flexible in their shape and the fact that they are continuously differentiable make them an excellent choice to model the systematic depth deviation (wiggling) of the ToF-camera. The objective of the undistortion is to obtain a correction value for every measured depth value. Hence the cubic B-Spline curve parametrizes the depth deviation of the measured depth from the correct depth. This leads to the following definition of the depth correction function using cubic B-Splines:

$$
d_c^* = d_c - \sum_{l=0}^{m} c_l B_l^3(d_c) \tag{3.16}
$$

in which c_l are the control-points of the B-Spline function which are estimated during wiggling parameter calibration, $B_l^3(d_c)$ are the basis B-Splines, and m is the number of control points. In this calibration framework the knot-points of the B-Spline curve are evenly distributed in the available depth range of the calibration data. The wiggling corrected depth d_c^* is obtained by subtraction of the returned value of the B-Spline curve. In the following I will consider three different cases for the evaluation of wiggling error compensation. The first two schemes, named [A] and [B] will use the cubic B-Spline to compensate wiggling errors.

Polynomial Depth Error Model

Polynomials have also been used in the literature, e. g. in [KCTT08], to model the depth deviation. A polynomial of degree 3 is used in [SBK08] to model the depth deviation. I chose to model the corrected depth d_c^* as:

$$
d_c^* = p_0 + (1.0 + p_1) * d_c + p_2 * x + p_3 * y + p_4 * d_c^2 + p_5 * d_c^3 \tag{3.17}
$$

in which $p_l, l \in \{0, ..., 5\}$ are the parameters to estimate. The factors d_2 and d_3 define a tilt in the image plane of the ToF-camera with respect to the optical axis, which was observed in the measurement data. This wiggling error model will be used for evaluation in scheme [C].

3.2.4. Reflectivity Calibration

Empirical investigation showed that the measured depth is also dependent on the reflectivity of the observed surface. An approach to calibrate the reflectivity dependent depth error of the ToF-measurement together with the wiggling error was introduced by Lindner and Kolb in [LK07]. In this approach a combined reflectivity and wiggling error compensation function is formulated using cubic B-Splines. In a corporate development with Lindner and Kolb [LSKK10, Lin10] the reflectivity calibration approach was integrated in the calibration process presented in this thesis by using a modified calibration pattern as shown in figure 3.6. The pattern consists of a modified checkerboard pattern in which black squares have been replaced with squares of different reflectivity to cover the required reflectivity range. On the left the inner squares are printed with 100% black, decreasing in 4 steps to 20 % black on the right. In consequence, the synthesized checkerboard pattern used for non-linear parameter estimation is altered as well. In this corporate development the reflectivity error compensation is detached from the wiggling error compensation and the correction functions are formulated in a different manner which is elucidated below.

Figure 3.7 shows the effect of the different degrees of reflection and absorption of the checkerboard pattern with squares of different gray levels. The amplitude image of the ToF-camera in image (a), compared to (c) shows that the material and the ink used to print the checkerboard has a different reflectivity for different wavelengths λ. While the squares, printed in light gray, are not visible in the infrared reflectance image (a) they are visible in image (c), recorded with conventional CCD-camera. The depth dependency is visible in image (b) and (d) in which the black checkerboard squares are closer to the camera than the white squares. Another effect which has to be taken into account is the radial light fall off (vignetting) in the amplitude images of the ToF-camera. This effect is also visible in image (a) of figure 3.7 in which the white squares show a decrease in brightness as they are located closer to the image borders.

The incorporation of the modified checkerboard pattern for the calibration of reflectivity dependent depth errors was first introduced in [LSKK10]. To make the reflectivity values comparable, the reflectivity values have to be normalized as the values are not constant for the same object color over the operation range of the camera, as can be seen from a comparison of images (a) and (c) of figure 3.7.

The reflectivity values are normalized by determining the minimal $h_{min}(d_c^*)$ and maximal reflectivity $h_{max}(d_c^*)$ for every depth interval Δd_c of 100mm and using equation:

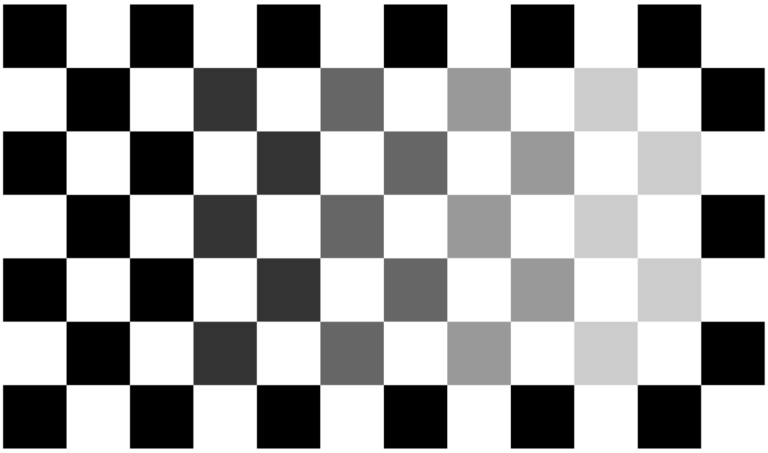


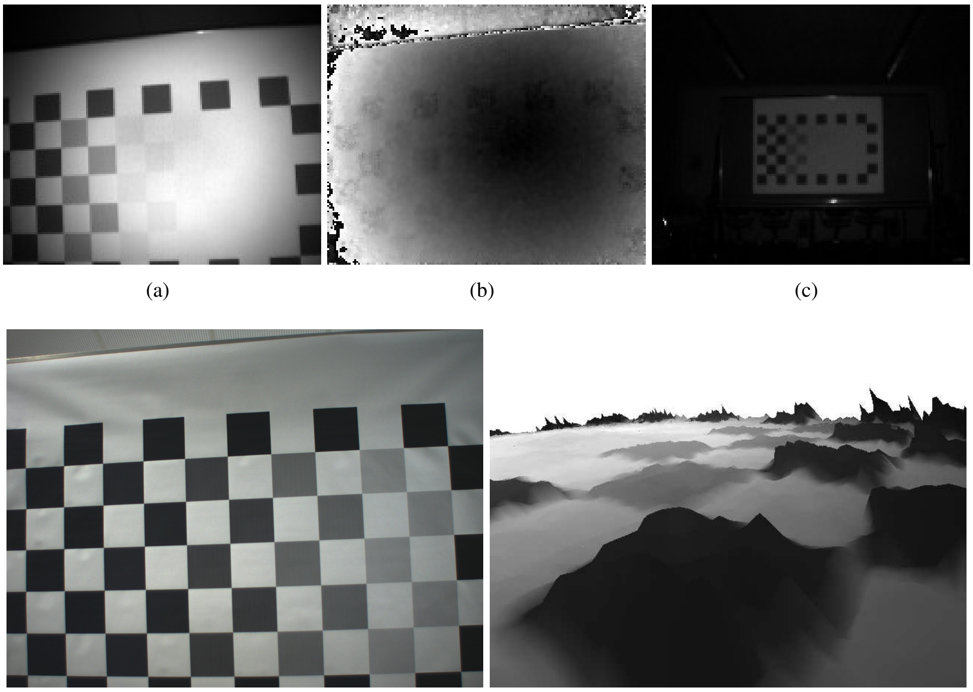
Figure 3.6.: The modified calibration checkerboard pattern used for reflectivity related calibration. Gray values are scaled from left to right in the center: 100% black, 80% black, 60 % black, 40 % black and 20 % black.

$$\begin{aligned}\hat{h}(h, d_c^*) &= norm(h, h_{min}(d_c^*), h_{max}(d_c^*)) \\ &= \frac{h - h_{min}(d_c^*)}{h_{max}(d_c^*) - h_{min}(d_c^*)}\end{aligned} \tag{3.18}$$

in which d_c^* represents the already wiggling corrected distance value. The final error adjustment consists of subtracting the reflectivity distance correction δ from the wiggling corrected depth d_c^*:

$$\hat{d}_c(d_c^*, \hat{h}) = d_c^* - \delta\left(\hat{h}(h, d_c^*)\right). \tag{3.19}$$

The resulting reflectivity and wiggling corrected depth is denoted $\hat{d}_c$ and in polar coordinates it is denoted $\hat{d}$. In this case, all three functions $\delta(\hat{h})$, $h_{min}(d_c^*)$ and $h_{max}(d_c^*)$ are modeled by



(a) (b) (c)

(d) (e)

Figure 3.7.: The amplitude image of the modified calibration pattern with different gray levels (a)+(c), to (a) corresponding depth image (b) and CCD intensity image for comparison (d). Textured triangle mesh of the ToF-measurement seen from the top left corner (e). Note the non-planarity in darker areas. Images taken with SwissRanger SR3000.

polynomials of degree 3, i. e.

$$\begin{aligned} h_{min}(d_c^*) &= \sum_{k=0}^{3} a_k^{\text{min}} {d_c^*}^k \\ h_{max}(d_c^*) &= \sum_{k=0}^{3} a_k^{\text{max}} {d_c^*}^k \\ \delta(\hat{h}) &= \sum_{k=0}^{3} a_k^{\delta} \hat{h}^k \end{aligned} \tag{3.20}$$

The radial light attenuation is incorporated by extending the normalization parameter by an additional radial attenuation:

$$\begin{aligned} h_{min}(d_c^*, r) &= \sum_{k=0}^{3} a_k^{\min} {d_c^*}^k + \sum_{l=0}^{3} b_l r^l \\ h_{max}(d_c^*, r) &= \sum_{k=0}^{3} a_k^{\max} {d_c^*}^k + \sum_{l=0}^{3} b_l r^l \end{aligned} \tag{3.21}$$

where r is the euclidean distance to the projection center (c_x, c_y) on the image plane (cmp. Eq. 2.1).

Analog Eq. 3.19 extends to

$$\begin{aligned} \hat{h}(h, d_c^*, r) &= norm(h, h_{min}(d_c^*, r), h_{max}(d_c^*, r)) \\ \hat{d}_c(d_c^*, \hat{h}) &= d_c^* + \delta\left(\hat{h}(h, d_c^*, r)\right) \end{aligned} \tag{3.22}$$

The reflectivity related calibration parameters are estimated by an additional step in the optimization (see section 3.2.5) using the same input data as for the intrinsic/depth deviation calibration. First, the normalization coefficients $a_k^{\min}$ and $a_k^{\max}$ are determined regarding the black and white squares of the checkerboard pattern. Therefore the depth range is divided into intervals of a given size (normally 100 mm) and for every interval the minimum and maximum reflectivity is determined. Note that correct data recording is crucial and has to take into account that the full intensity range for all depth intervals has to be available. The radial attenuation is also determined in this step. On the white squares of the calibration pattern the average of the reflectivity values is calculated for every radius r. Only natural numbers are used for r, resulting in a discretization of the radius with step-size 1. Finally, the coefficients of the actual deviation function $\delta\left(\hat{h}(h, d_c^*, r)\right)$ are estimated based on the normalized intensities as described above.

3.2.5. Optimization

This section describes the method to estimate the internal camera parameters and the external orientation from the depth and the reflectance image. From the estimate of the internal and external camera parameters the internal camera parameters as well as rotation $\mathsf{R}_{jk}^{(0)}$ and position $\boldsymbol{C}_{jk}^{(0)}$ of the cameras are approximately known. I will now show how to estimate those quantities starting from approximate initial values.

The rotation matrix can be represented using its Taylor expansion: (cf. [FW04, p.53])

$$\mathsf{R}_{jk}^{(\nu+1)} \approx \mathsf{R}_{jk}^{(\nu)} + \begin{pmatrix} 0 & -\kappa & \phi \\ \kappa & 0 & -\omega \\ -\phi & \omega & 0 \end{pmatrix} \tag{3.23}$$

For every optimization step the relevant parameters are collected in a parameter vector $\boldsymbol{p}$.

$$\boldsymbol{p} = (f_x, f_y, x_0, y_0, s, r_1, r_2, t_1, t_2, \omega, \phi, \kappa, C_x, C_y, C_z, ...)^{\mathsf{T}} \tag{3.24}$$

The depth image can be synthesized from the parameters using equation (3.12) as:

$$d_{ijk}(p_l) = f_{ijk}^{(1)}(\boldsymbol{p}) \approx f_{ijk}^{(1)}\left(\boldsymbol{p}^{(\nu)}\right) + \left.\frac{\partial f_{ijk}^{(1)}}{\partial \boldsymbol{p}}\right|_{\boldsymbol{p}^{(\nu)}} \Delta\boldsymbol{p} \tag{3.25}$$

and the reflectance image can be synthesized from the parameters using equation (3.14) as:

$$I(\boldsymbol{x}_{ijk}) = f_{ijk}^{(2)}(\boldsymbol{p}) \approx f_{ijk}^{(2)}\left(\boldsymbol{p}^{(\nu)}\right) + \left.\frac{\partial f_{ijk}^{(2)}}{\partial \boldsymbol{p}}\right|_{\boldsymbol{p}^{(\nu)}} \Delta\boldsymbol{p} \tag{3.26}$$

Note, that in this formulation no derivative of any observed noisy low-resolution depth or reflectance image is required for the Taylor expansion. An efficient way to synthesize depth and reflectance image is to render the checkerboard on the GPU. This possibility is exploited here as checkerboard- and depth image are rendered on the GPU including the full camera model. This significantly speeds up the calibration process.

Figure 3.9 shows the synthesized images generated with equations 3.12 and 3.14. On the left the synthesized depth image and on the right a synthesized reflectance image is shown. The synthesized images are overlaid on the real images.

The Jacobian is denoted with:

$$\mathsf{A}_{ijk}^{(1)} = \left.\frac{\partial f_{ijk}^{(1)}}{\partial \boldsymbol{p}}\right|_{\boldsymbol{p}^{(\nu)}} \qquad \mathsf{A}_{ijk}^{(2)} = \left.\frac{\partial f_{ijk}^{(2)}}{\partial \boldsymbol{p}}\right|_{\boldsymbol{p}^{(\nu)}} \tag{3.27}$$

and

$$\Delta\boldsymbol{l}_{ijk}^{(1)} = d_{ijk}(p_l) - f_{ijk}^{(1)}\left(\boldsymbol{p}^{(\nu)}\right) \tag{3.28}$$

$$\Delta\boldsymbol{l}_{ijk}^{(2)} = I(\boldsymbol{x}_{ijk}) - f_{ijk}^{(2)}\left(\boldsymbol{p}^{(\nu)}\right) \tag{3.29}$$

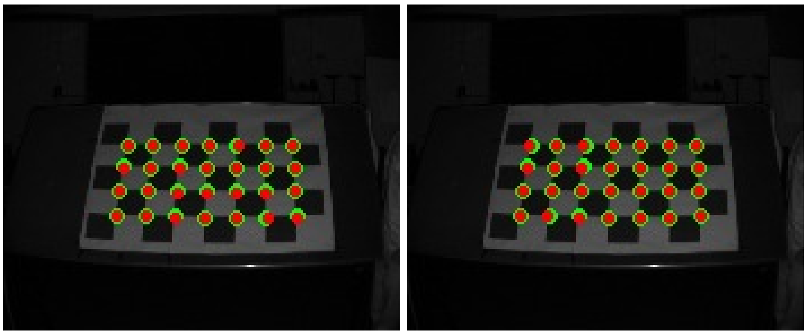
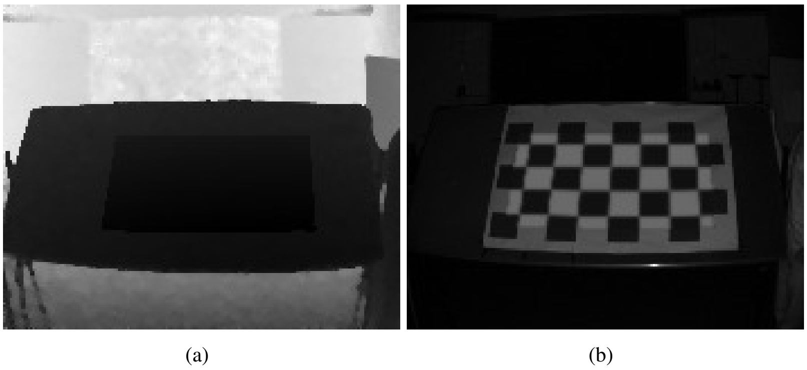


Figure 3.8.: Two amplitude images of the SR4000-camera with reprojected checkerboard corners during optimization. The detected checkerboard corners are marked in green and the projection of the 3D points is overlaid in red.



(a)

(b)

Figure 3.9.: Depth image (a) and corresponding amplitude value image (b) from SR4000-camera with projected depth plane in the center of the blackboard (a) and projected checkerboard in amplitude image (b).

one obtains the parameter covariance as (cf. [FW04], p.87)

$$\mathsf{C}_{\boldsymbol{pp}}^{(\nu+1)} = \left(\sum_{i}^{N} \sum_{j}^{M} \sum_{k}^{K} \frac{1}{\sigma_{jk}^{2}} \mathsf{A}_{ijk}^{T(\tau(k))} \mathsf{A}_{ijk}^{(\tau(k))} \right)^{-1} \tag{3.30}$$

where $\tau(k)$ indicates the current camera type. (1=ToF-camera, 2=CCD-camera) The variance factors σ_{jk} are initially set to 1.

From this the parameter update is computed as

$$\Delta \boldsymbol{p} = \boldsymbol{C_{pp}} \left(\sum_{i}^{N} \sum_{j}^{M} \sum_{k}^{K} \frac{1}{\sigma_{jk}^2} \boldsymbol{A}_{ijk}^{T(\tau(k))} \Delta \boldsymbol{l}_{ijk}^{(\tau(k))} \right) \tag{3.31}$$

yielding the improved parameter vector

$$\boldsymbol{p}^{(\nu+1)} = \boldsymbol{p}^{(\nu)} + \Delta \boldsymbol{p} \tag{3.32}$$

The sum of the squared residuals in a depth or reflectance image are given by:

$$\Omega_{jk} = \sum_{i}^{N} ||\boldsymbol{A}_{ijk}^{(\tau(k))} \Delta \boldsymbol{p} - \Delta \boldsymbol{l}_{ijk}^{(\tau(k))}||^2 \tag{3.33}$$

so that the variance factors can be updated according to (cf. [FW04], p.91)

$$\left(\sigma_{jk}^{(\nu+1)}\right)^2 = \left(\sigma_{jk}^{(\nu)}\right)^2 \frac{\Omega_{jk}}{R_{jk}} \tag{3.34}$$

with R_{jk} = the redundancy of the observations j, k. Starting with initial variance factors $\sigma_{jk}^{(0)} = 1$ this process is iterated until convergence. The convergence criterion is, that the update is smaller than 1% of the expected accuracy, i.e. $\Delta \boldsymbol{p}^{-T} \boldsymbol{C_{pp}^{-1}} \Delta \boldsymbol{p} < 0.01$.

3.2.6. Results and Discussion

For the discussion of the results I will investigate three different setups. Scheme [A] is the calibration of a ToF-camera without any additional CCD-cameras and B-Spline function for depth error compensation. Scheme [B] is the calibration of a ToF-camera with one additional CCD-camera and B-Spline depth error compensation and scheme [C] is as scheme [B] but with a polynomial error model for depth deviation. In the analysis the accuracy and correlations of the internal, external and depth deviation parameters will be investigated.

Error Analysis without Additional CCD Camera [A]

In this section the accuracy and correlations of the parameters of a ToF-camera are investigated, if calibrated without additional CCD-cameras from 42 ToF-images in the range of 1.2 - 6 meters, which is denoted scheme [A]. Tab. 3.3 shows the estimated values for the camera matrix in pixel [px]. Compared to the scheme with additional CDD camera (see. Tab. 3.7) the focal length is estimated too small, which was compensated by external and wiggling error parameters.

f_x [px]	f_y [px]	c_x [px]	c_y [px]
207.0551	205.1033	105.9832	85.6877

Table 3.3.: Estimated focal lengths and principle point for scheme [A].

The estimated accuracies for the internal parameters are shown in Tab. 3.4. It can be seen that these values have been estimated with good confidence. The comparison of the results with the results in section 3.2.6, which have been achieved using an additional CCD camera, however shows that the values are not correct, which is due to the ambiguity of focal length, external parameters and wiggling error which is reflected in the increased correlation of the parameters visible in table 3.6.

σ_{f_x} [px]	σ_{f_y} [px]	σ_{c_x} [px]	σ_{c_y} [px]
0.05881	0.06827	0.03207	0.03610

Table 3.4.: Accuracy of the internal camera parameters for scheme [A].

Tab. 3.5 states the accuracies of the external camera parameters. These have also been estimated with high precision, although the deviations of the translations are in the range of a few millimeters.

σ_X [mm]	σ_Y [mm]	σ_Z [mm]	σ_ω [°]	σ_ϕ [°]	σ_κ[°]
3.26	3.36	12.02	8.82e-4	8.94e-4	46.0e-4

Table 3.5.: Accuracy of the external camera parameters of the ToF-camera for scheme [A].

Tab. 3.6 shows the correlations of the external camera parameters for the calibration of a single ToF-camera. It can be seen that correlations between translation and rotation are quite high, although lower than the values in [BK08], which can be explained by the increased resolution and field-of-view of the ToF-camera.

Error Analysis with Additional CCD Camera [B]

In this second experiment, the calibration has been performed with the proposed combination of a high-resolution ($1600 \times 1200[px]$) CCD- and a ToF-camera (scheme [B]). The average reprojection error of the CCD-camera after initial guess by OpenCV is $1.24031[px]$ and the mean reprojection error of the ToF-camera is $0.167957[px]$. Note that for the initial guess every camera pose is estimated individually and no rigidity for the camera rig is enforced. After

	X	Y	Z	ω	ϕ	κ
X	1	0.017	0.086	0.005	**0.636**	0.002
Y	0.017	1	**0.213**	**0.482**	0.003	0.031
Z	0.086	0.213	1	0.020	0.080	0.028
ω	0.005	0.482	0.020	1	0.035	0.157
ϕ	0.636	0.003	0.080	0.035	1	0.024
κ	0.002	0.031	0.028	0.157	0.024	1

Table 3.6.: Computed correlations between the external ToF-camera parameters corresponding to the accuracy values shown in Tab. 3.5.

enforcing rigidity in the rig, the average reprojection error of the CCD-camera is $1.24021[px]$ and $0.708043[px]$ for the ToF-camera. After optimization on the checkerboard corners, the reprojection error is reduced to $1.16606[px]$ for the CCD-camera and $0.440916[px]$ for the ToF-camera.

f_x [px]	f_y [px]	c_x [px]	c_y [px]
210.9124	209.8890	100.6778	82.2850

Table 3.7.: Estimated focal lengths and principle point for scheme [B].

Tab. 3.8 shows the accuracies of the estimation of the internal camera parameters of the ToF-camera, which are shown in Tab. 3.7. By combining the ToF-camera with the CCD-camera, we gain one order of magnitude for focal length and radial distortion parameters and two orders of magnitude for the estimation of the principle point, compared to the values in Tab. 3.4.

σ_{f_x} [px]	σ_{f_y} [px]	σ_{c_x} [px]	σ_{c_y} [px]
0.00241	0.00330	0.00095	0.00083

Table 3.8.: Accuracy of the internal camera parameters of the ToF-camera for scheme [B].

According to Tab. 3.5, Tab. 3.9 shows the accuracies of the external camera parameters for the combined approach. Again a significant increase in confidence is visible in the order of two magnitudes. This matches the results of the correlation analysis which is shown in Tab. 3.10. The correlations between rotation and translation is now widely reduced.

Distance Deviation

An example of calibrated depth measurements for a SwissRanger3000 together with the mean errors and standard deviations is shown in Fig. 3.11, which corresponds to the uncalibrated

σ_X [mm]	σ_Y [mm]	σ_Z [mm]	σ_ω [°]	σ_ϕ [°]	σ_κ [°]
0.019	0.017	0.047	1.51e-6	1.14e-6	3.92e-6

Table 3.9.: Accuracy of the external camera parameters of the ToF-camera for scheme [B].

	X	Y	Z	ω	ϕ	κ
X	1	0.006	0.000	0.034	**0.101**	0.002
Y	0.006	1	**0.025**	**0.013**	0.027	0.013
Z	0.000	0.025	1	0.013	0.045	0.007
ω	0.034	0.013	0.013	1	0.014	0.006
ϕ	0.101	0.027	0.045	0.014	1	0.093
κ	0.002	0.013	0.007	0.006	0.093	1

Table 3.10.: Correlations between the external ToF-camera parameters corresponding to the accuracy values shown in Tab. 3.9

measurements shown in Fig. 3.10 in which the estimated B-Spline function, approximating the wiggling error, is shown as well. The horizontal axis shows the distance of the camera center to the reference plane, which is acquired using the vision-based optimization. Note that in scheme [A] the distances are estimated from the camera poses only, which has been calculated from the low-resolution ToF-images. This leads to a higher error and expanding (or shrinking) of the whole measurement range. The vertical axis shows the remaining error which is reduced below 50 mm throughout the whole operating range of the camera.

As introduced in section 3.2.3 different approaches of depth error compensation have been investigated. The in [SBK08] proposed polynomial model is compared to the B-Spline model introduced in [LSKK10].

Figure 3.12 shows the result of the fitting of a polynomial of degree three to the distorted depth measurements. Note that the approximation of the polynomial is not as precise as the approximation of the B-Spline function in figure 3.10.

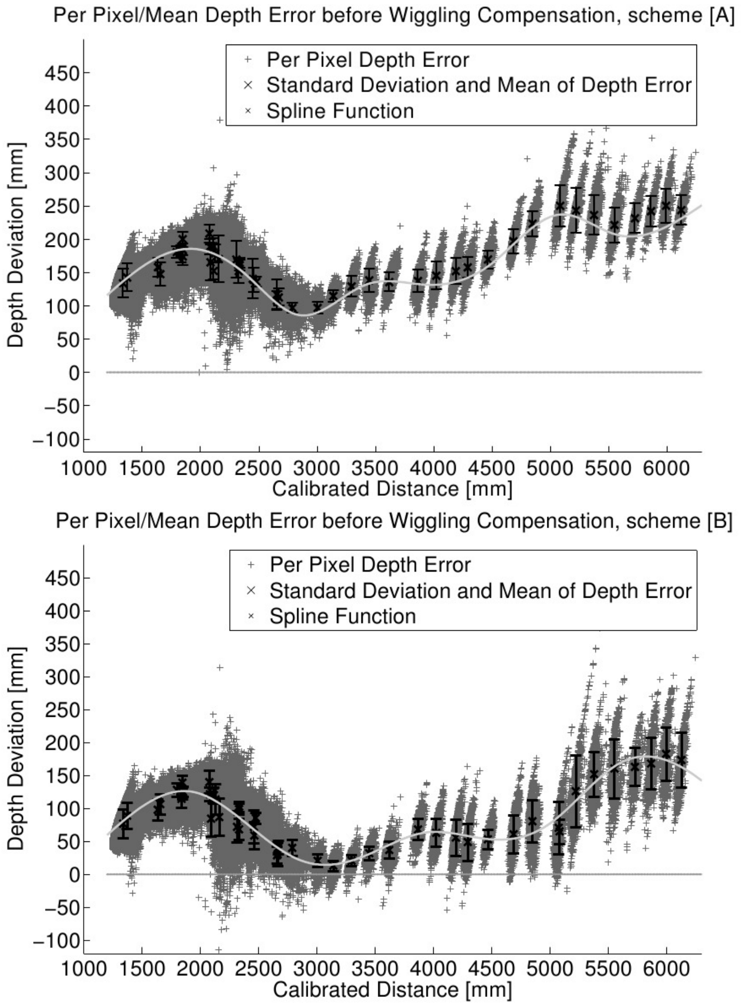


Figure 3.10.: Wiggling error before correction for scheme [A] (top) and [B] (bottom). The depth error of every ToF-image pixel is shown in gray, mean error and standard deviations are shown in black and the estimated B-Spline function is shown in light gray. Note that the error for scheme [A] is much higher due to the underestimated focal length and errors in pose estimation. Using an additional CCD-camera helps to minimize such errors.

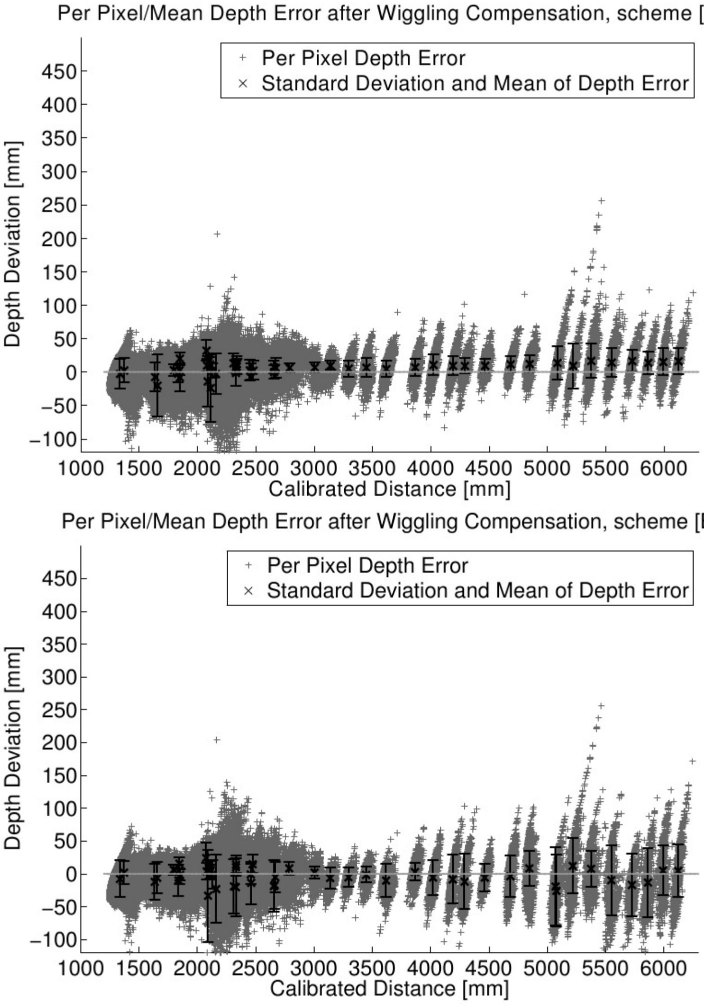


Figure 3.11.: Wiggling error after correction. The remaining depth error of every ToF-image pixel is shown in gray, mean error and standard deviations are shown in black.

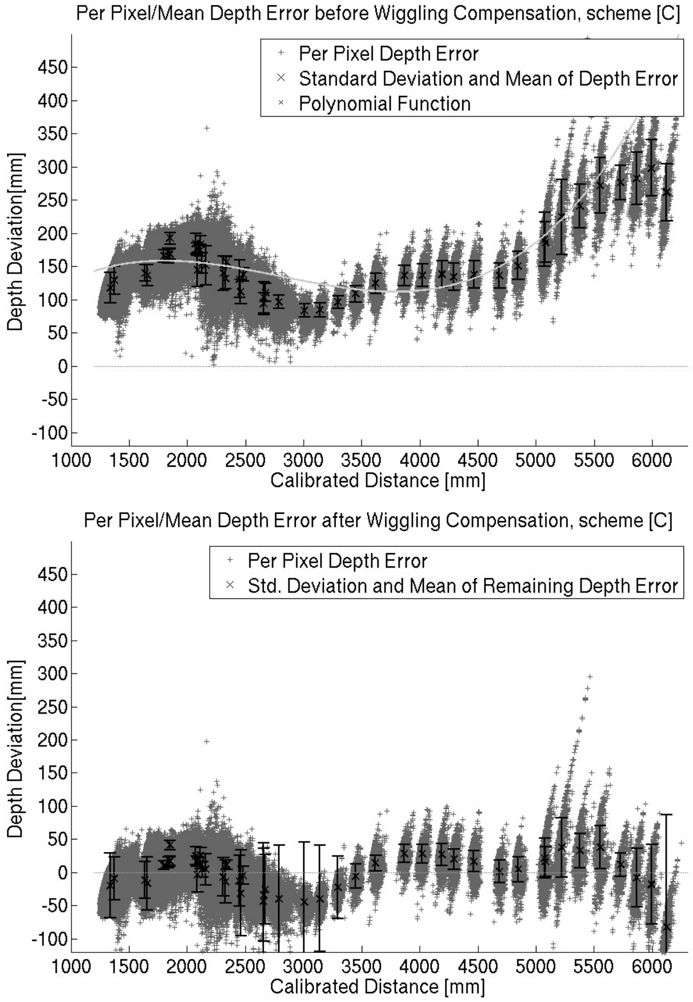


Figure 3.12.: Wiggling error of polynomial depth error correction model before and after correction. The depth error of every ToF-image pixel is shown in gray, mean error and standard deviations are shown in black. In comparison to the graphs in figure 3.11 the function does not fit the data very well and the remaining error is much higher.

Reflectivity Calibration

To investigate the suitability of reflectivity calibration the approach is verified on synthetic data. Therefore a sequence of images with the modified calibration pattern (cf. figure 3.6) was produced. The depth images have been distorted on the black and gray squares with a maximum distortion of 50mm, linearly decreasing in steps of 20% till 10mm. Additionally the radial light attenuation is simulated with a quadratic light attenuation from the center of the image to the borders. Example images of this sequence are shown in figure 3.13.

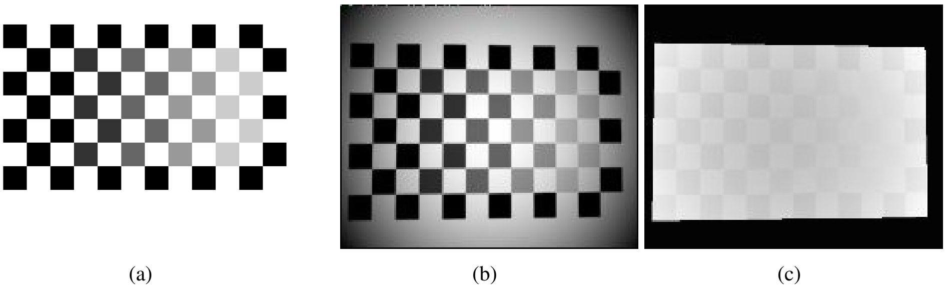
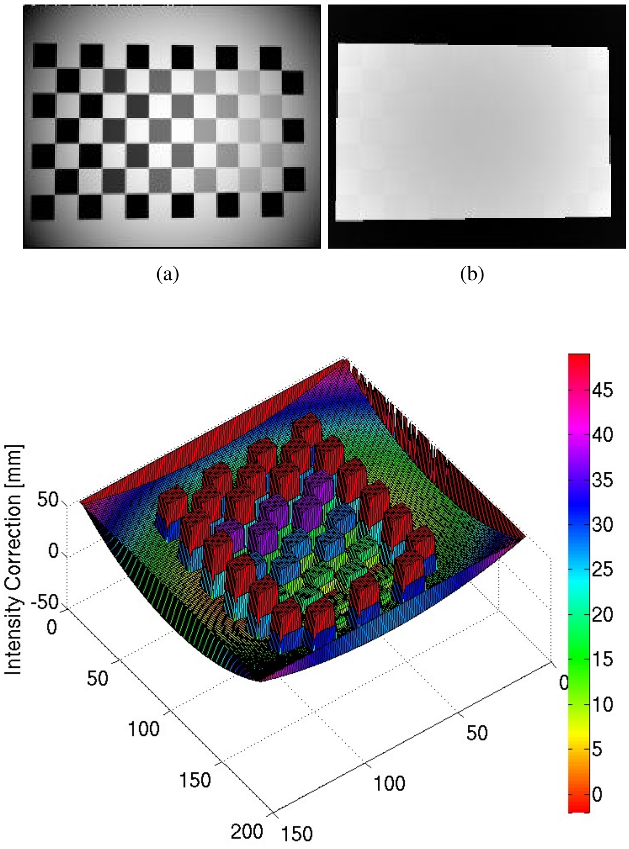


Figure 3.13.: Input images of synthetic scene for reflectivity calibration evaluation. (a) CCD image, (b) vignetted ToF-amplitude image, (c) reflectivity distorted depth image.

After optimization of the parameters and application to the distorted images the simulated effect could be widely reduced. The mean depth error of all pixel in all images was reduced from 47.34 mm to 8.10 mm using the estimated parameters. A visual result is shown in figure 3.14.

The effect of the reflectivity calibration on the real data is visualized in figure 3.15. Image (a) shows the ToF- amplitude image and image (b) shows the reflectivity corrected depth image. Image (c) visualizes the difference between wiggling corrected depth image and reflectivity corrected depth image. The depth correction is much smaller than in the simulated data. The different correction values for the different reflectivities of the checkerboard are visible as elevations in the graph. The reflectivity corrected depth image (b) shows a much smoother depth than the original depth image in figure 3.7 (b).

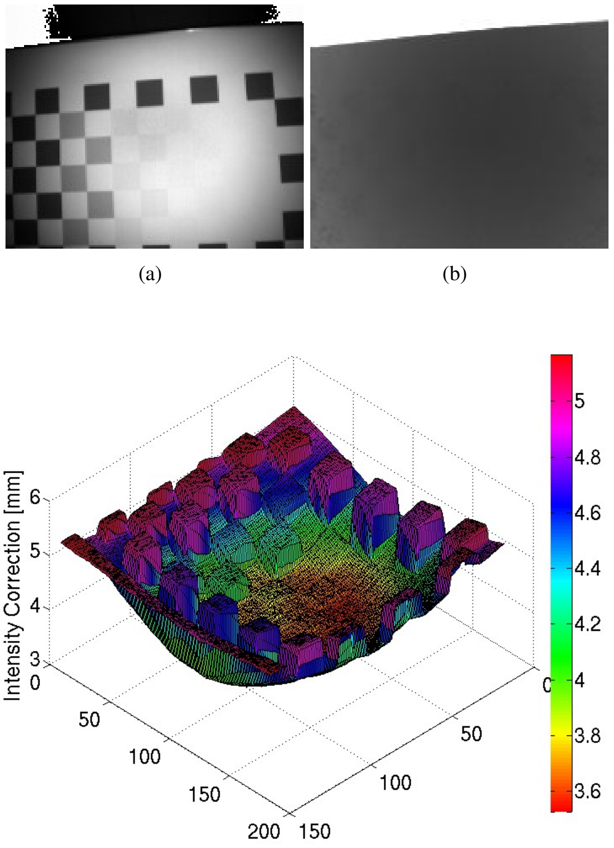


(a)

(b)

(c)

Figure 3.14.: Results of reflectivity calibration on synthetic scenes. (a) Difference image, (b) reflectivity undistorted depth image and 3D plot of difference between distorted and undistorted depth image. (Compare with image (c) in figure 3.13.)



(a)

(b)

(c)

Figure 3.15.: Results of reflectivity calibration on real data. (a) Difference image, (b) reflectivity undistorted depth image and 3D plot of difference between distorted and undistorted depth image. (Compare with image (b) in figure 3.7.) Note the different scale in comparison with figure 3.14 (c).

3.2.7. Calibration of Multi-ToF-Camera Setups

The reconstruction of deforming and moving objects suffers from incomplete data if the object is viewed by a single ToF-camera from a single perspective. The resulting reconstructed object is open at the backside for example. To overcome such incompleteness assumptions about the object can be made based on continuity of the object, e. g. cylindrical shapes could be assumed. This is however prone to errors as assumptions have to be made which may not be correct.

Another possibility is to use multiple ToF-cameras and to view the object from different perspectives, covering all areas of the object's space. Using multiple ToF-cameras however introduces several challenges and error sources which have to be considered. The ToF-cameras must not be operated using the same modulation frequency as the cameras influence each other and measurements are distorted by the active illumination of the other cameras. Additionally if the illumination units of other cameras are within the field of view of a ToF-camera, pixel of the ToF-camera are often saturated and reliable measurement is made impossible in this area.

Approach

The registration of cameras which face each other requires a suitable calibration procedure. Model-based calibration using a three-dimensional calibration object promises accurate calibration results. Hence the analysis-by-synthesis calibration approach as described in section 3.2.5 is exploited to calibrate the relative extrinsic parameters (rotation and translation) of the cameras, using a known threedimensional calibration object. The geometry of the calibration object has to be known and the visible parts of the objects have to be detectable in the images to calibrate the setup. For this reasons a marker-based calibration has been chosen as markers are distinguishable in the images and their size is known.

In a first step the exact geometry of the used markercube has to be calculated. Example input images are shown in figure 3.17. From these images and the knownledge about the marker geometry, encoded in the marker patterns, the geometry of the cube is calculated. An example marker cube reconstruction is shown in figure 3.18 (a). The cube is reconstructed with a mean reprojection error of the 3D points of the markers to the corresponding 2D points of 2.16784 pixel at a resolution of 3504×2336 pixel. This geometry is then used in the calibration as replacement for the checkerboard pattern in the analysis-by-synthesis calibration.

For the calibration of the external parameters in the second step, the calibration marker cube is placed at several positions in the interaction area (depicted in figure 3.16) and images are taken with all CCD- and ToF-cameras. In each image the markers are detected and to each of the eight detected 2D points on the marker borders, 3D points are assigned from the marker object. For the ToF-cameras, the intensity or amplitude images are used to detect the markers. The same optimization strategy as in section 3.2.5 is applied, but only external pose parameters

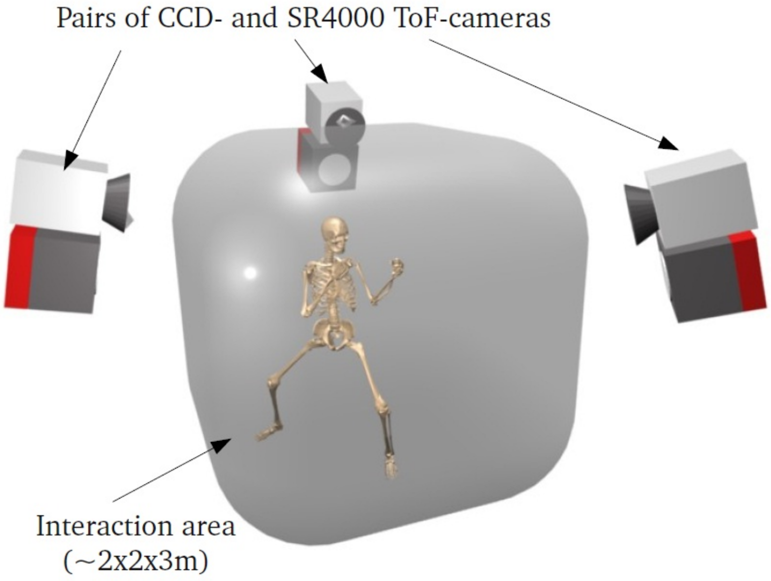


Figure 3.16.: Schematic setup for calibration of three ToF-and three CCD cameras which observe an interaction area. The transparent cube symbolizes the interaction area.

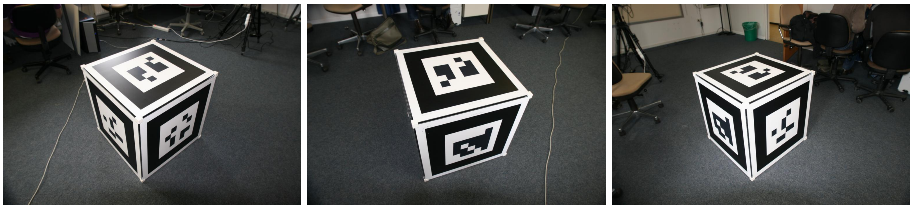
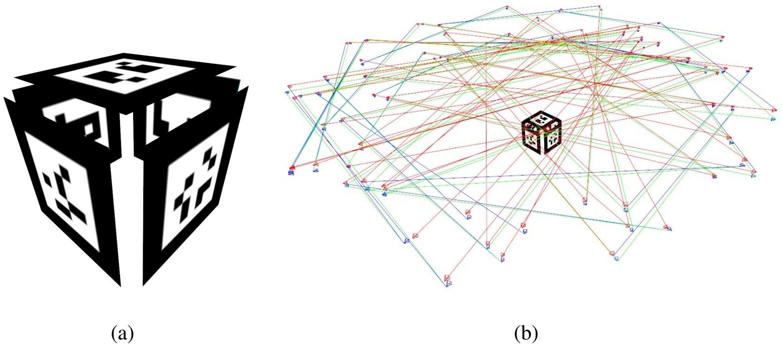


Figure 3.17.: Example input images for marker cube reconstruction.



(a)

(b)

Figure 3.18.: The reconstructed marker cube (a) and the calculated camera positions around the cube (b).

Camera	Initial error[px]	Rigidity Enforced[px]	Reprojection Optimized[px]
CCD-Camera 1	2.9417	2.9417	2.9746
CCD-Camera 2	2.0772	3.3700	2.1623
CCD-Camera 3	2.4400	3.2476	2.4742
ToF-Camera 1	0.3101	0.3780	0.3574
ToF-Camera 2	0.3438	0.7752	0.3899
ToF-Camera 3	0.2577	1.0236	0.3355

Table 3.11.: Reprojection errors of marker corners for a camera setup of 6 cameras.

are estimated. The internal and depth deviation parameters have been estimated separately for each pair of CCD and ToF-camera before. After optimization on the 2D/3D correspondences the second optimization step is performed by rendering the markers on the GPU and optimizing on real images of the marker cube with rendered views of the marker cube. As result of the calibration process the external camera parameters of all used cameras are available.

Results

Table 3.11 shows the reprojection errors of the 6 camera setup in pixel [px], consisting of 3 CCD cameras coupled with 3 ToF-cameras. The first column shows the reprojection error after initial individual pose estimation. In the second column the mean rigid transformation between the 6 cameras has been enforced which leads to an increase in reprojection error which is minimized in the last column by the optimization on the reprojected corners.

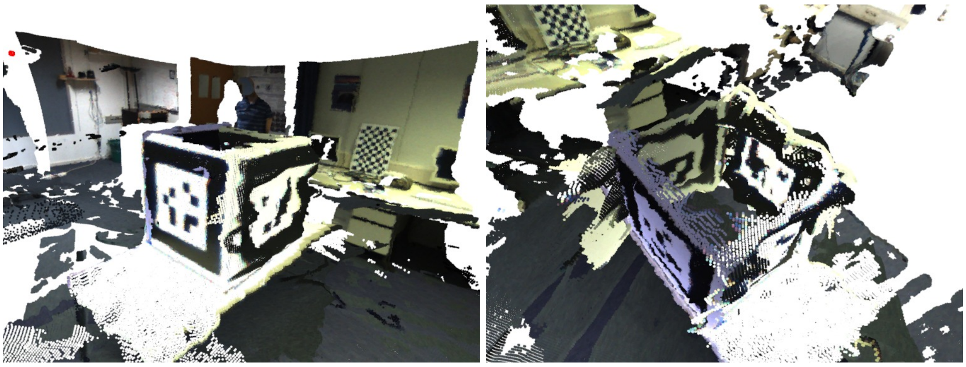


Figure 3.19.: The fused calibration object from 3 depth- and 3 CCD images.

Figure 3.19 shows the reconstruction of the calibration object using three depth and corresponding CCD- images of the calibration images. The depth values have been projected to 3D points using the calculated external camera parameters and corresponding color values have been taken from the CCD images by projecting the 3D points into the cameras. The depth values have been fused in a 3D volumetric Octree structure which is introduced in section 4.3.3, p.73. Additionally noise by flying pixel has been reduced by applying variance analysis of the depth values as described in section 4.1.3, p.54. The images show that the geometry of the marker cube is reconstructed very precisely.

3.3. Conclusions

In this chapter I presented a calibration method for ToF-cameras in combination with multiple standard CCD-cameras. Standard computer vision algorithms are used for the initial parameter estimation. The parameters are optimized using an analysis-by-synthesis approach. The model used is a rendered smoothed checkerboard pattern. In contrast to point-based calibration methods a measurement for every pixel is obtained and the method is independent of the calibration model. The parameters are estimated using non-linear optimization. This method overcomes the limitations of the small opening angle todays ToF-cameras suffer from. The comparison of depth error compensation with a polynomial function and with a B-Spline function shows that the B-Spline function is clearly more suitable as it is capable to fit to the data more precisely and significantly reduces the error. The calibration of multiple ToF-cameras, which oberserve the same scene is another important issue for which a solution was presented in this chapter by using a threedimensional calibration pattern made of markers. The following chapters and the presented applications and examples all make use of the presented calibration approaches.

"Each problem that I solved became a rule, which served afterwards to solve other problems."

Rene Descartes (1596 -1650)

4

2D/3D Data Processing

In this chapter, preprocessing algorithms for noise reduction concerning Time-of-Flight depth images are presented and discussed in section 4.1.1. The combination of ToF-cameras with standard cameras and the transfer of the depth measurements into the perspective of additional cameras is presented in section 4.1.2. In section 4.2 segmentation algorithms are evaluated, comparing background image based approaches with approaches which combine color and depth information and in section 4.3 suitable datastructures to store ToF-depth data of three-dimensional scenes are discussed.

4.1. Depth Preprocessing

Different error sources distort the depth measurement of ToF-cameras. These are also described in [LSKK10], [LK06] and [LK07]. Like in almost every imaging-based measuring process, the distance measurement with ToF-cameras suffers from noise in the measured data. Noise in ToF- imaging can be reduced by increasing the integration time T_{int} (see equation 3.2, p.17) of the camera which leads to less frames per second and is not feasible for real-time applications. Therefore a good compromise between integration time and real-time processing is needed.

4.1.1. Noise Reduction

In the literature mostly median or mean filtering of the depth images is applied for denoising. Besides that some other approaches to denoise depth data can be found. Mure-Dubois [MDH07a] [MDH07b] focuses on the scattering compensation of ToF- cameras. Scattering describes the effect that the depth measurements change around an object if this object is introduced in the scene. More precicely, the measurements do not only change in the area in which the object actually is but also in other areas of the scene, mostly due to reflections. This effect is neglected in this thesis. Huhle et al. [HSJS08] use a NL-Means Filter to denoise ToF-depth data with additional color information and Matzka et al. [MPW07] advise the usage of temporal filtering over a number of frames. However, this leads to increasing errors if the camera is moving or if the scene changes rapidly. So they propose to use a small number of images for temporal filtering and a spatial Gaussian filter. In [HBMB08] Haker et al. also propose to use a Gaussian low-pass filter to reduce noise in the images.

Median/Mean/Gauss Filter

Noise reduction is crucial for every measured signal. To reduce the noise in a depth image the simplest solution is to apply a mean or median filter to such an image. In this thesis most of the images have been filtered with a median filter before further processing. Filtering images is a convolution with a filter-mask. The mean filter is a mask with constant values, so for every pixel $\boldsymbol{x}$ the mean depth value $\tilde{d}(\boldsymbol{x})$ of a small area is calculated:

$$\tilde{d}(\boldsymbol{x}) = \frac{1}{A} \sum_{i=-N}^{N} \sum_{j=-M}^{M} d(\boldsymbol{x}_{i,j}) \tag{4.1}$$

in which N and M are the sizes of the filter-mask so for $N = M = 1$ this is a 3×3 mean filter, and $A = (2N+1) \cdot (2M+1) = 9$.

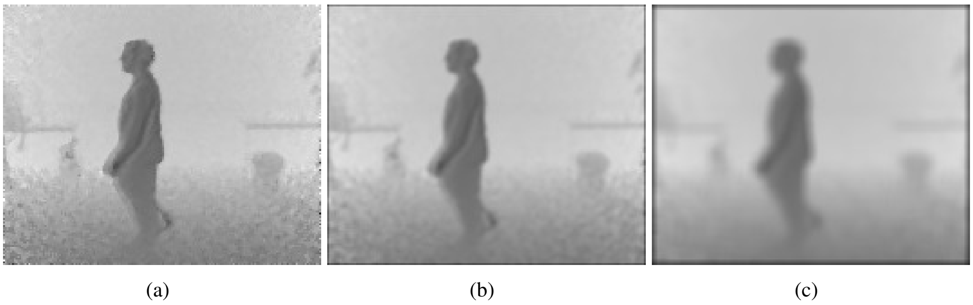
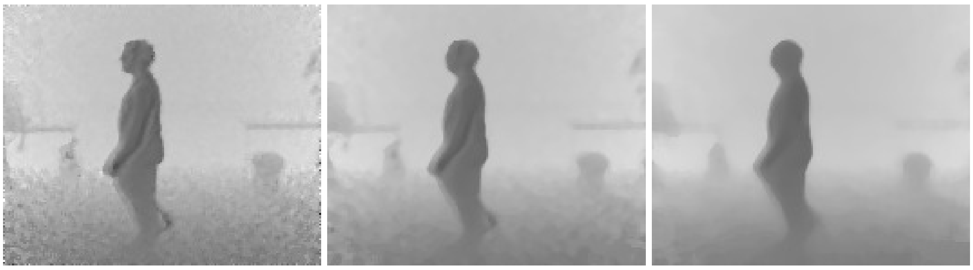


Figure 4.1.: Original (a) and mean filtered depth images with 5x5 (b) and 9x9 (c) pixel filter mask.

In a median filter the pixel values are replaced by the median of all pixel values in a surrounding with $N \times M$ pixel. The median is not realized as a convolution. Instead, for every depth pixel $d(\boldsymbol{x})$ in the image, the surrounding $N \times M$ pixel are sorted in an array in an increasing manner. The new value $\tilde{d}(\boldsymbol{x})$ of the output image is then the entry with index $(((N+2)\cdot(M+2))+1)/2$ in the sorted array. The median filter is more robust to outliers, as single pixel which strongly deviate from the rest of the pixel have less or even no influence on the result. Furthermore is the median filter computationally efficient as the complexity is $\mathcal{O}(N \cdot logN)$ or even $\mathcal{O}(1)$ [PH07].



(a) (b) (c)

Figure 4.2.: Original (a) and median filtered depth images with 5x5 (b) and 9x9 (c) pixel filter mask.

The spatial Gauss filter is a filter which applies a radial symmetric Gaussian kernel function:

$$G_\sigma(x,x_0)=\frac{1}{\sigma\sqrt{2\pi}}\cdot e^{-\frac{1}{2}\left(\frac{(x-x_0)^2}{\sigma^2}\right)} \tag{4.2}$$

in which $x-x_0$ is the Euclidean distance between two values x and x_0. If $\boldsymbol{x}$ is a two-dimensional vector this represents the Gaussian kernel in two dimensions, as e. g. an image. The filtered image is now calculated as:

$$\tilde{d}(\boldsymbol{x})=\sum_{i=-N}^{N}\sum_{j=-M}^{M}G_\sigma(\boldsymbol{x},\boldsymbol{x}_{i,j})\cdot d(\boldsymbol{x}_{i,j}) \tag{4.3}$$

Bilateral Filter

The Bilateral filter is a so called edge-preserving filter which makes it a superior alternative to the aforementioned filters. Mean and Gaussian filters tend to blur edges in the images which is fatal in depth images as the blur crosses sharp edges in depth and heavily distorts the measurements. The Bilateral filter uses two Gaussian kernels (see equation 4.2), one in the spatial domain which defines the influence of the surrounding pixel and one in the intensity/depth domain which limits the filtering effect on edges.

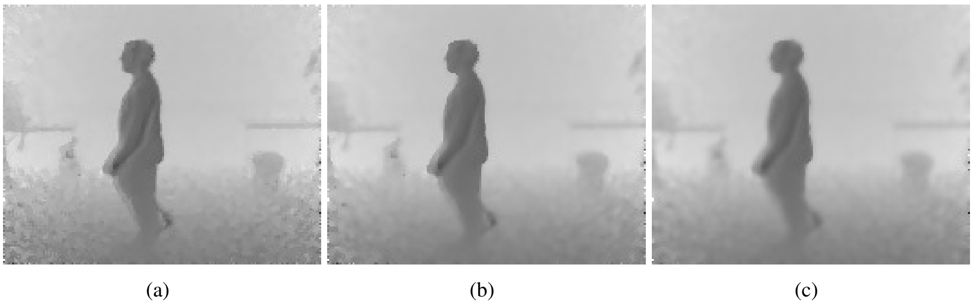


Figure 4.3.: Bilateral filtered depth images. Images have been generated with two iterations, a filter-size $N, M = 3 \ (7 \times 7)$ pixel, $\sigma_s = 5$ and $\sigma_d = 100$ (a), 300 (b) and 900 (c). Observe the preserved contours of the person in comparison to mean (figure 4.1) and median filtering (figure 4.2).

So the Bilateral filtered depth $\tilde{d}(x)$ is obtained using:

$$\tilde{d}(\boldsymbol{x}) = \frac{1}{A} \sum_{i=-N}^{N} \sum_{j=-M}^{M} G_{\sigma_s}(\boldsymbol{x}, \boldsymbol{x}_{i,j}) G_{\sigma_d}(d(\boldsymbol{x}), d(\boldsymbol{x}_{i,j})) \cdot d(\boldsymbol{x}_{i,j}) \tag{4.4}$$

in which $G_{\sigma_d}(d(\boldsymbol{x}), d(\boldsymbol{x}_{i,j}))$ is the Gaussian distribution in the depth domain and $G_{\sigma_s}(\boldsymbol{x}, \boldsymbol{x}_{i,j})$ is the Gaussian distribution in the spatial domain. If an intensity image with the same resolution and aligned content is available this can also be used to prevent the filter from smoothing across intensity edges, as done in [PSA$^+$04] for images taken with an extra flash, by rewriting equation 4.4 as:

$$\tilde{d}(\boldsymbol{x}) = \frac{1}{A} \sum_{i=-N}^{N} \sum_{j=-M}^{M} G_{\sigma_s}(\boldsymbol{x}, \boldsymbol{x}_{i,j}) G_{\sigma_I}(I(\boldsymbol{x}), I(\boldsymbol{x}_{i,j})) \cdot d(\boldsymbol{x}_{i,j}) \tag{4.5}$$

with $I(\boldsymbol{x})$ the intensity at pixel $\boldsymbol{x}$ (and $I(\boldsymbol{x}_{i,j})$ the intensity at pixel $\boldsymbol{x}_{i,j}$). This is then called a “Cross Bilateral Filter” or “Joint Bilateral Filter”. In [Wei06] efficient methods to speed up the bilateral filter are described and the parameter A is defined as:

$$A = \sum_{i=-N}^{N} \sum_{j=-M}^{M} G_{\sigma_s}(\boldsymbol{x}, \boldsymbol{x}_{i,j}) G_{\sigma_I}(I(\boldsymbol{x}), I(\boldsymbol{x}_{i,j})) \tag{4.6}$$

4.1.2. Depth Warping

For many applications it is necessary to combine the ToF-camera with one or multiple standard color cameras. Examples for such applications are environment reconstruction in depth and color, registration by feature matching and matting in depth and intensity. To use a ToF-camera together with a color camera, both have to be calibrated internally and the relative external translation and rotation between the cameras have to be known. The calibration approach presented in section 3.2, p. 23 was developed to serve for this purpose.

For applications which require high resolution depth maps it is necessary to transfer the depth image of the ToF-camera into the perspective of the color camera. An approach to do this is to project every pixel of the ToF-depth image to a 3D point using equation 2.12, p.7 and project these 3D points to the image plane of the color camera using equation 2.10, p.6. The resulting holes can be filled by applying image filters as proposed in [MFY$^+$09]. The big difference in resolution between ToF- and color camera causes large uncovered areas in the depth image in the perspective of the color camera. These areas are not filled with depth measurements from a projected 3D point. Filling these holes using filtering causes errors, and because large filter masks have to be used this approach is computationally expensive. The warping result using simple projection and hole filling with a dilation filter is illustrated in figure 4.4.

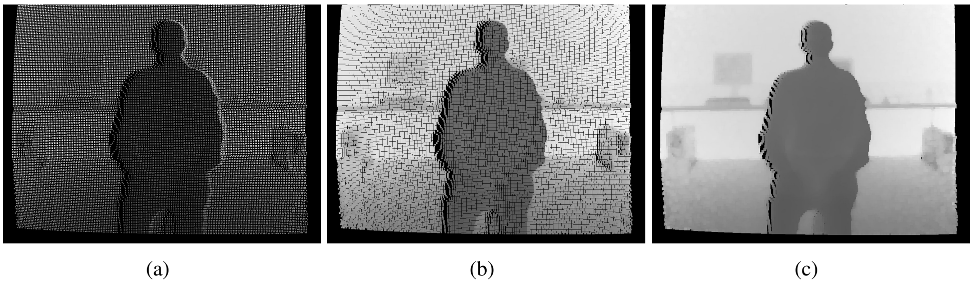


Figure 4.4.: Warped depth images generated by projection of depth values to 3D points and reprojection to the color camera image plane. Additionally a dilation filter with kernel size 7x7 (a), 11x11 (b) and 21x21 (c) is applied.

To circumvent these errors and high computation times, the efficient way to mesh the depth image on the GPU and to render it with the internal and external camera parameters of the color camera as described in [BSBK08] is used in this thesis. Recall that every pixel in a depth image captured by the ToF-camera represents a 3D scene point. This 3D point is reconstructed by scaling the ray $\mathbf{r}$ from the camera center through the pixel position $\boldsymbol{x}$ (cf. equation 2.11, p.6) with the depth value $d(\boldsymbol{x})$ at this pixel position as in equation 2.12, p.7. To render the geometry on the GPU, vertices have to be constructed. So neighboring pixel are connected to form a triangle-strip. If the internal camera matrix of the ToF-camera does not change, the mapping between pixel positions and rays can be precomputed, as well as the indices of the triangle-strip, which significantly speeds up calculation. The warping process is pictured in figure 4.5.

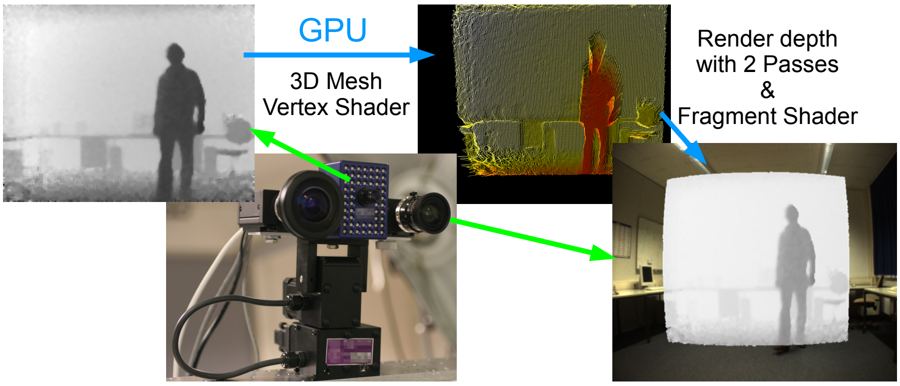


Figure 4.5.: Depth warping of ToF-depth-image into the perspective of the color camera by rendering as triangle mesh on the GPU.

For every new ToF-image the vertices of the triangle-strip are moved along the viewing rays of the camera and the geometry is rendered with the projection matrix and the external parameters of the camera to which the depth is transferred to. Applying this method a 2D/3D video stream is generated, consisting of two images for each time step with color and depth values for each pixel. In figure 4.6 a color image, a depth image and a warped depth image are shown. This is a convenient and computationally efficient method, but it also suffers from some problems. As observable in image (c) of figure 4.6, connections between fore- and background are present in the resulting image which leads to large errors in subsequent algorithms. Additional warping errors are visible at the left contour of the person. These are results of the meshing of fore- and background and the artifacts become visible because the two cameras do not share a common camera center. The strength of the error is therefore dependent on the distance between color- and ToF-camera and on scene depth. Analyzing the resulting image as in the next section 4.1.3 can limit the influence of the error. Additionally filtering with a bilateral filter can by applied as proposed by Frick et al. in [FKBK09] to align color and depth data (see also [KCLU07]).

4.1.3. Flying Pixel Removal

Another problem not solved in hardware are flying pixel. These are pixel which mostly occur at depth edges, positions in a depth image where a transition between fore- and background

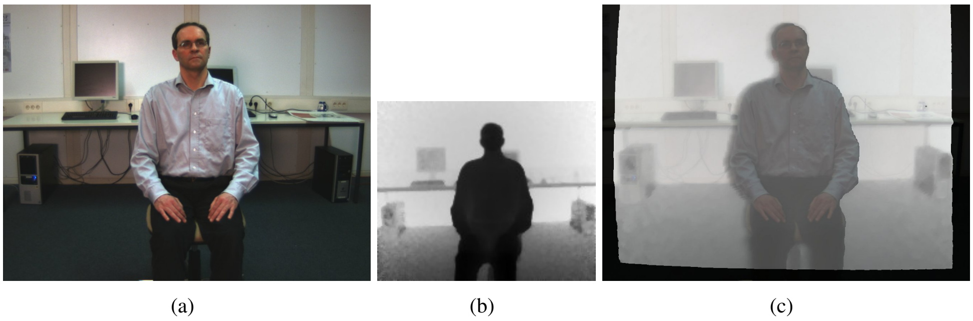


Figure 4.6.: CCD color image (1024x768 [px]) (a), depth image (176x144 [px]) (b), depth image warped into the view of the color camera including lens distortions (c). The warped image is overlayed on the CCD image to show the fit.

is present. These pixel are mostly located between fore- and background, but can also occur in front of the whole scene or randomly in the image. A possibility to indentify flying pixel is to analyze the amplitude information a (cf. equation 3.7, p.18) of the ToF-camera. However, this can be misleading as low amplitude values do not always indicate flying pixels. Correctly measured distances can also have low amplitude values due to surface and material effects. Most approaches do not explicitly handle flying pixel, but rely on a correction of these errors during the filtering or super-resolution of the depth image. Sabov and Krüger [SK10] describe three methods for the identification of flying pixel which they prefer to call flying surfels. These methods are: difference of absolute distances in a region around a pixel (see also [HJS08]), difference of normal directions and angular difference between mean surface normal and viewing direction. Besides the detection of flying pixel they also describe methods to correct these errors.

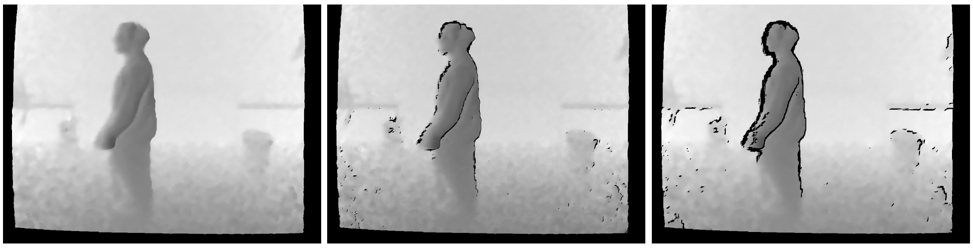


Figure 4.7.: Variance analysis for flying pixel removal. Original warped image (left) and variance filtered image (middle) with $\tau = 80mm$ and $\tau = 40mm$ (right).

In this thesis flying pixel are identified by analyzing the number of valid neighboring pixels. This is efficiently done by variance analysis of the neighboring pixel. If the variance σ^2 of the depth $d(\boldsymbol{x})$ of a pixel $\boldsymbol{x}$:

$$\sigma^2 = \frac{1}{(2N+1)\cdot(2M+1)} \sum_{i=-N}^{N} \sum_{j=-M}^{M} (d(\boldsymbol{x}_{i,j}) - d(\boldsymbol{x}))^2 \tag{4.7}$$

is above a threshold τ this pixel is invalidated and the depth set to zero.

Figure 4.7 shows the result of the variance analysis for flying pixel removal for $M = N = 2$ and $\tau = 80mm$ and for $\tau = 40mm$. Due to the limited resolution of the ToF-camera there are depth measurements connecting the person in the foreground with the background in the image on the left. These flying pixel have been invalidated by variance analysis. Note that the variance analysis for flying pixel removal is applied after warping the depth to the color cameras perspective. This is motivated by the warping algorithm as this algorithm is not designed to handle holes in the depth maps.

4.2. Segmentation of Dynamic Objects

Segmentation in this chapter describes the separation of foreground objects from a static background. Segmentation of dynamic objects is different from the detection of dynamic objects as detection mainly focuses on the detection that something is moving. The segmentation however needs to be pixel-wise and the object borders have to be segmented with high accuracy. Matting additionally aims at the computation of an (alpha-) "Matte", which is an image with values between zero and one, used to combine the segmented foreground objects with a new background. An additional demand in this thesis is that the computation has to be real-time capable which excludes many established methods for segmentation and matting.

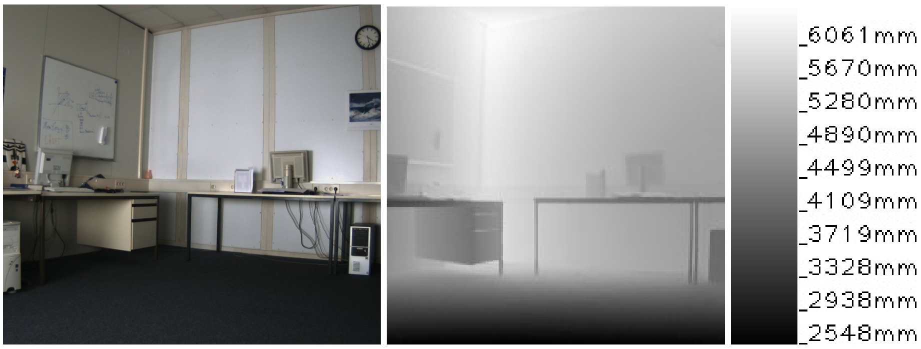
4.2.1. Prior Work

In the literature the two different goals, detection and segmentation, can be distinguished. Mixture-of-Gaussians is a well known method for foreground segmentation [SEG99],[XE01]. More recently the Graph Cuts [BVZ01] and Grab Cuts [RKB04] methods have been presented and gained a lot of attention in the research area of segmentation. One of the first who combined depth and color for segmentation was Gordon et al. in [GDHW99] who combined depth from stereo and color for background estimation. Using ToF-cameras, object detection and segmentation mostly employs thresholding of the depth images (e. g. [GHW$^+$06], [KPHB08]). Other authors use Trimaps from depth with Cross-Bilateral filtering [CTPD08], Graph Cut methods based on Trimaps from depth or infrared cameras and variants [WBB08] or fusion of stereo and ToF-camera with Loopy Belief Propagation (LBP) using Trimaps and Markov Random Fields (MRF) [ZLYP09]. These more advanced approaches are not applicable in real-time applications because of their computational complexity.

In this thesis precise and real-time capable segmentation is a mandatory part for the following approaches and algorithms. In [SK09] and [SBKK08] simple depth-keying on the GPU is used, described in section 4.2.2. In [SK11] this approach was extended to a combination with color mixture models to improve the segmentation. This approach is described in section 4.2.4.

4.2.2. Segmentation by Depth-Keying

A well known technique in media productions is the so called chroma-keying, also known as blue- or green screen. This is the principle in which some foreground objects are placed in front of a green or blue background. The foreground objects are easily segmented and the uniform background can be replaced by some other background. This approach can be adopted to the depth domain by using depth-keying as in [GKOY03]. Instead of using color to segment foreground objects the distance of an object to the camera can be used to segment objects and



(a) (b) (c)

Figure 4.8.: Background images, averaged over 100 images. a) Depth image, b) CCD image, c) depth image scale.

persons. Using depth makes the segmentation process independent of the color information and it is not required that foreground objects are of different color than the background.

Depth-Keying with Background Image

The most obvious way to detect moving objects is to do a comparison between some background image of the scene and the current depth image delivered by the ToF-camera. The background image can be created by averaging several ToF-images. Figure 4.8 shows an example of an averaged depth- and CCD image.

In the detection phase the depth $d^*(\boldsymbol{x})$ of every pixel $\boldsymbol{x}$ of a ToF-image is compared to the background image pixel's depth $d_b^*(\boldsymbol{x})$ and if:

$$d^*(\boldsymbol{x}) < d_b^*(\boldsymbol{x}) - \tau \tag{4.8}$$

the pixel $\boldsymbol{x}$ is classified as foreground or moving object. (τ is a threshold) Figure 4.9 shows a segmentation/keying result. The current depth image with a person in the scene is shown in the top row on the left together with the corresponding CCD-image on the right. As a segmentation in the domain of the CCD-camera is desired, because of the higher resolution and desired consistency between depth and color information, the background- and current depth image are warped to the CCD-camera's perspective using the warping described in section 4.1.2. In the bottom left corner of figure 4.9 the problem of depth-keying is visible. The selection of the used threshold is difficult due to the amount of noise in the data and the soft borders in the depth image resulting from the low resolution of the ToF-camera data. This

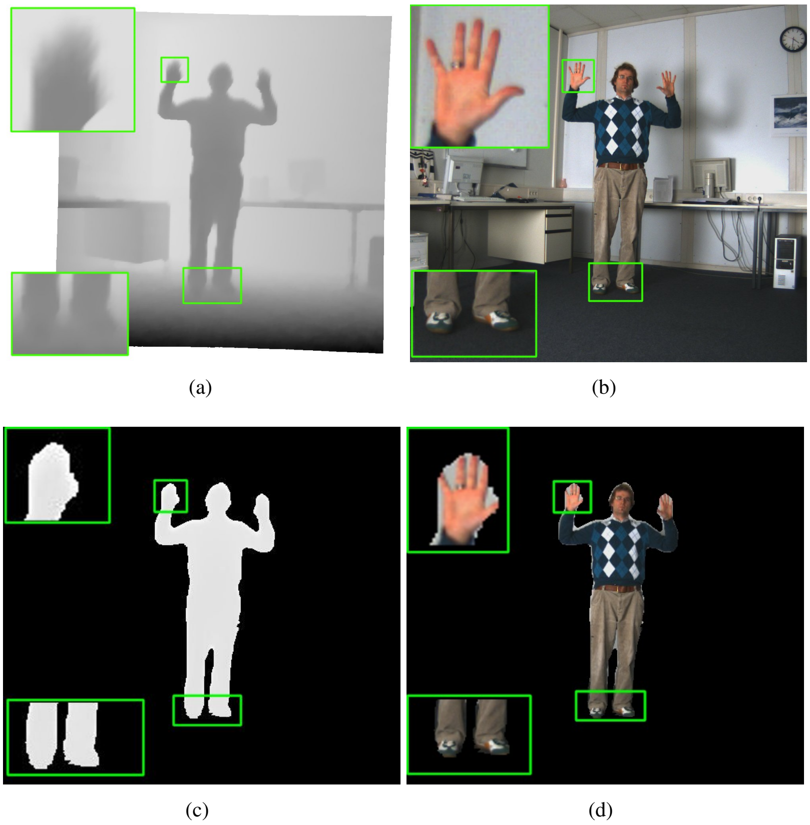


Figure 4.9.: Warped depth image with person (a), Corresponding CCD image (b), depth segmented image (c) and corresponding CCD image (d).

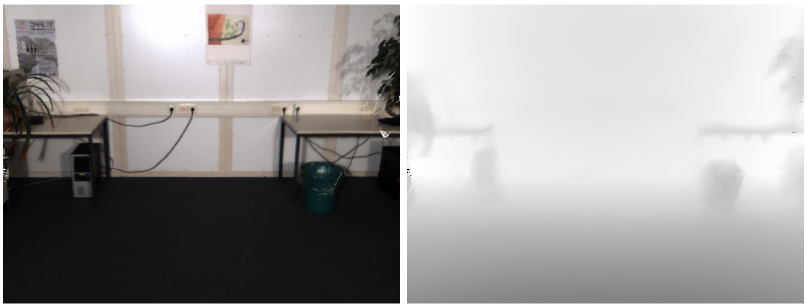


Figure 4.10.: The background model rendered in the perspective of the current camera's position.

causes that either parts of the foreground object are classified as background or that parts of the background are considered foreground.

This simple keying approach has many limitations. It is only applicable in case of a static camera and rigid background scene and the segmentation is not very precise as boundary errors are very frequent due to noise and the low depth resolution. Other, more flexible and reliable methods are therefore needed.

Depth-Keying with Background Model on GPU

In scenarios in which the camera is moving during recording, the above described depth keying with background image is not applicable. As a replacement for the background image a background model can be used which contains the geometry and the intensities of the environment. The background model can be generated as described in section 5.1.2, p.78. For keying, the background model is rendered on the GPU with the current camera parameters of the color camera. This way rendered images of depth and intensity in the same camera perspective as the real color camera are obtained. Figure 4.10 shows a color and a depth image of the rendered background model rendered with the camera parameters of the current color camera. The current ToF-depth image can be warped to these camera parameters using the method described in section 4.1.2. The same thresholding operation can now be applied as described in equation 4.8. As the images are already on the GPU as textures, the keying itself is a predestined application for the GPU which significantly speeds up computation. The keying is realized using GPU shaders in which the equation 4.8 is realized.

With this approach a segmentation can be performed although the camera is moving which is not possible using simple background image subtraction. Figure 4.11 shows the current

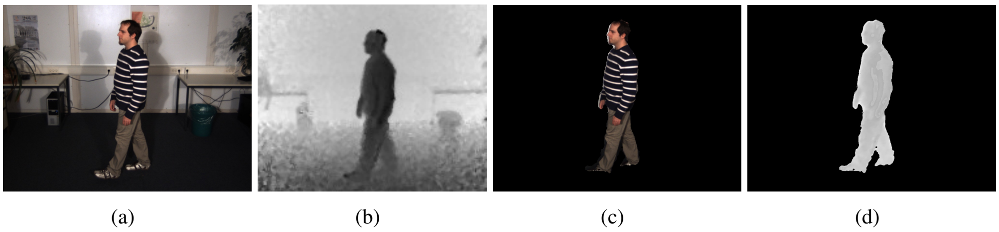


Figure 4.11.: The images of the color (a) and depth camera (b) and the segmented foreground person in color (c) and depth (d).

input images in color and depth and the segmented color and depth image using the depth-keying on the GPU with background model. The main problem of segmentation using depth as sole segmentation measure still remains. The small resolution leads to errors, especially at object borders. Adaptive background models are a well-known method in segmentation. This method can be transferred to depth images which is the scope of the next section.

4.2.3. Segmentation by Mixture-of-Gaussians

The detection of dynamic objects in a scene using an adaptive background mixture model relates to the work of Xu et al. [XE01] and Stauffer et al. [SEG99]. There a method is described that uses multiple Gaussian distributions to model each pixel of an image. Pixel values of subsequent images are tested against these Gaussian distributions and classified as fore- or background.

Mixture-of-Gaussians

In [XE01] it is defined that an intensity image is defined by the three color channels in RGB space $F(\boldsymbol{x}) = (f_R(\boldsymbol{x}), f_G(\boldsymbol{x}), f_B(\boldsymbol{x}))$ for every pixel $\boldsymbol{x}$. To simplify the notation I will omit the pixel index $\boldsymbol{x}$ in the following equations. Under the assumption that no channel is saturated this representation can be transformed in a normalized form $f = (f_r, f_g, f_b)$ with:

$$f_k = f_K / \sqrt{f_R^2 + f_G^2 + f_B^2} \tag{4.9}$$

and $k \in \{r, g, b\}$ and $K \in \{R, G, B\}$. It is stated that it is appropriate to model each color channel f_k using a Gaussian distribution. The weights ω, the mean values μ and the variances σ^2 are all defined per channel k, but to simplify the notation the channel index k is omitted in the following equations. Following Xu et al. the possibility of observing a value f_t at pixel $(\boldsymbol{x})$ and time t is given by:

$$P(f_t) = \sum_{i=1}^{N} \omega_{i,t} G_{\Sigma_{i,t}}(f_t, \mu_{i,t}) \tag{4.10}$$

where N is the number of distributions, $\omega_{i,t}$ is a weighting factor for each distribution at the position $\boldsymbol{x}$ and $G_{\Sigma_{i,t}}(f_t, \mu_{i,t})$ is the Gaussian probability density function. The covariance matrix $\Sigma_{i,t}$ is approximated by the sum of its diagonal elements $\sigma^2_{i,t}$ as the pixel can be assumed to be independent. Hence the Gaussian probability density function is formulated as:

$$G_{\sigma_{i,t}}(f_t, \mu_{i,t}) = \frac{1}{\sigma_{i,t}\sqrt{2\pi}} e^{-\frac{(f_t - \mu_{i,t})^2}{2\sigma^2_{i,t}}} \tag{4.11}$$

Initial values for $\sigma^2_{0,0}$ and $\mu_{0,0}$ are determined over a small image region Δx with n pixel:

$$\begin{aligned} \mu_{0,0}(\boldsymbol{x}) &= \frac{1}{n} \sum_{\Delta x} f_0(\boldsymbol{x} + \Delta x) \\ \sigma^2_{0,0}(\boldsymbol{x}) &= \frac{1}{n-1} \sum_{\Delta x} ||f_0(\boldsymbol{x} + \Delta x) - \mu_{0,0}(\boldsymbol{x})||^2 \end{aligned} \tag{4.12}$$

A pixel is classified as belonging to the background distribution i if:

$$||f_t - \mu_{i,t-1}|| < \beta \sigma_{i,t-1} \tag{4.13}$$

The value of $\beta \approx 3$ has been found experimentally by Xu et. al in [XE01]. If this condition is fulfilled the parameters are updated following equation 4.14, and the weight $\omega_{i,t}$ is increased while the weights of the not matched distributions $\omega_{j,t}$, with $j \neq i$ are decreased.

$$\begin{aligned} \mu_{i,t}(\boldsymbol{x}) &= (1-\gamma)\mu_{i,t-1}(\boldsymbol{x}) + \gamma f_t(\boldsymbol{x}) \\ \sigma^2_{i,t}(\boldsymbol{x}) &= (1-\gamma)\sigma^2_{i,t-1}(\boldsymbol{x}) + \gamma ||f_t(\boldsymbol{x}) - \mu_{i,t}(\boldsymbol{x})||^2 \end{aligned} \tag{4.14}$$

In equation 4.14 the parameter γ controls the update rate. If the current pixel does not match any of the existing distributions a new distribution is generated, and if the number of distributions exceeds the maximum, the distribution with the lowest weight is deleted. The distributions with the highest weights are regarded as background. Pixel that do not fall into these distributions are therefore classified as foreground and moving objects. The average distance of a pixel $\boldsymbol{x}$ to the background distributions is the color weight and denoted $\Delta c(\boldsymbol{x})$ in the

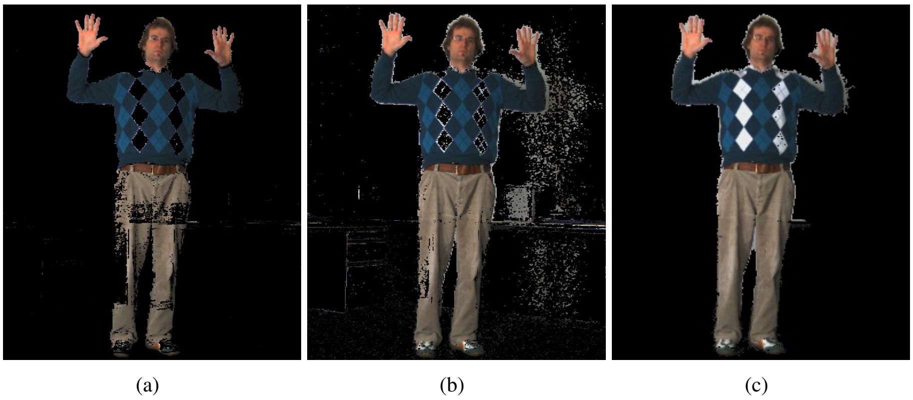


Figure 4.12.: MoG segmentation result: The color segmentation easily under- (a) or over-segments (b) foreground objects. MoG with depth as fourth channel improves the result, but borders remain erroneous (c).

following:

$$\Delta c(\boldsymbol{x}) = \frac{1}{N \cdot M} \sum_{i=1}^{N} \sum_{k=1}^{M} |f_{k,t} - \mu_{i,k,t}| \tag{4.15}$$

Recall that k is the current channel and M is the number of channels.

Figure 4.12 (a) and (b) show segmentation results if only the Mixture-of-Gaussian is used as segmentation clue. It can be seen that the algorithm tends to either under- (a) or over-segment (b) the person, especially if a challenging scene is chosen with many shadows and colors similar to the background. The advantage of this approach is that if an object moves into the scene it is detected as moving. If it then becomes stationary, it is integrated into the background after a number of frames. The number of frames after which the object is integrated into the background model depends on the update rate. If it starts moving again it is automatically classified as foreground again.

Extending Mixture-of-Gaussians to ToF-depth Images

The above described MoG approach is well-known for color images. It has however not been used on depth images since such high frame rate cameras taking depth images just became available. The adaption of the MoG algorithm to depth images is straightforward and promises even better results as differences in depth are more significant than differences in color and depth measurements are independent to lighting changes and shadows. Instead of using the color measurement F the depth measurement d^* is now used and no normalization is necessary. The color space therefore consists of only one channel: $f = (d^*)$. The possibility to observe a depth measurement $f_t = (d^*_t)$ is accordingly:

$$P(f_t) = \sum_{i=1}^{N} \omega_{i,t} G_{\Sigma_{i,t}}(f_t, \mu_{i,t}) \tag{4.16}$$

The other equations are easily adapted to using only one channel and depth instead of color information. The real strength of the approach is to use both, color and depth, to verify the detection of the one with the other. While depth measurements possibly can't detect very fine or small structures that change, color mostly can, but color detection may fail in uniform regions in which detection on depth measurements are the solution. The depth information is therefore added as forth channel to the MoG on color images. The vector on which the distributions are defined is then formed by the three color channels in rgb space and the depth value d^* as combined RGBD space: $f = (f_r, f_g, f_b, d^*)$. Note that the color values have been normalized following equation 4.9. Image 4.12 (c) shows that this can compensate some of the shortcomings of the pure MoG segmentation, but the borders of the foreground person are still erroneous.

4.2.4. Combining Mixture-of-Gaussians on Color and Depth Keying

The above described combination of color and depth already makes advantage of both modalities. MoG on combined rgbd space can apply different weights on color and depth, but does so for the whole image. The crucial decision is however, which information is more likely to be correct and if the color and depth modalities deliver different results which one to trust.
ToF-depth measurements contain a significant amount of noise, are of very low resolution and contain flying pixel which are located between fore- and background. This and the warping errors resulting from the change in perspective affect the segmentation especially at object borders and at points where fore- and background meet. A good example can be seen in figure 4.9 in which parts of the feet and the hands of the person are not detected as foreground. Therefore this segmentation approach aims at incorporating the information about depth discontinuities into the segmentation process. Two different approaches for a reliability measure are evaluated. The first is the usage of the variance in depth and the second is the usage of the amplitude information provided by the ToF-camera. The amplitude image quantifies the amount of light that is reflected from the object to the camera. The higher the values the more reliable is the measurement. At object discontinuities less light is reflected due to scattering effects. Therefore the inverse of the amplitude image is used to enable its usage in the same way as the variance image (see figure 4.13 (a)). Depth discontinuities for the first method can be detected by analyzing the variance in the original depth image. (Variance analysis is explained in section 4.1.3.) High variances are marked as shown in figure 4.13 (b). To be

able to compare the different modalities of depth and color they have to be normalized. So the current depth error compensated depth difference between current depth d^* and background depth d_b^*:

$$\Delta d^*(\boldsymbol{x}) = d^*(\boldsymbol{x}) - d_b^*(\boldsymbol{x}) \tag{4.17}$$

is normalized between the minimum Δd^*_{min} and the maximum depth difference Δd^*_{max} in that image and in the same way the weight of the color foreground pixel $\Delta c(\boldsymbol{x})$ (cf. section 4.2.3) is normalized between the minimum and maximum color weights Δc_{min} and Δc_{max}.

$$\Delta d_n^*(\boldsymbol{x}) = \frac{\Delta d^*(\boldsymbol{x}) - \Delta d^*_{min}}{\Delta d^*_{max} - \Delta d^*_{min}} \qquad \Delta c_n(\boldsymbol{x}) = \frac{\Delta c(\boldsymbol{x}) - \Delta c_{min}}{\Delta c_{max} - \Delta c_{min}} \tag{4.18}$$

The variance and inverted amplitude values are also normalized between zero and one to be comparable to the other measurement weights $\Delta d_n^*(\boldsymbol{x})$ and $\Delta c_n(\boldsymbol{x})$, and denoted the normalized uncertainty $v_n(\boldsymbol{x})$. In areas in which the depth uncertainty is high, the depth measurement is considered unreliable. Therefore the normalized depth difference $\Delta d_n^*(\boldsymbol{x})$ is weighted with the uncertainty $v_n(\boldsymbol{x})$, resulting in an uncertainty filtered depth difference $dv(\boldsymbol{x})$ which is scaled between zero and one. In contrast, the color is more reliable if the depth uncertainty is high. Hence the color weight $\Delta c_n(\boldsymbol{x})$ is multiplied with the depth uncertainty $v_n(\boldsymbol{x})$ and added to the color weight resulting in the uncertainty weighted color weight $cv(\boldsymbol{x})$. The result is that if the depth uncertainty is high the color weight is weighted even higher while at the same time the uncertainty filtered depth is weighted lower. To consider all measures in an adequate manner the following equations are proposed:

$$dv(\boldsymbol{x}) = (1 - v_n(\boldsymbol{x}))\Delta d_n^*(\boldsymbol{x}) \tag{4.19}$$

$$cv(\boldsymbol{x}) = (1 + v_n(\boldsymbol{x}))\Delta c_n(\boldsymbol{x}) \tag{4.20}$$

$$\alpha(\boldsymbol{x}) = \frac{dv(\boldsymbol{x}) + cv(\boldsymbol{x})}{2} \tag{4.21}$$

Figure 4.13 shows the weighting images of both approaches separately (c) and (d) and combined (e). Brighter values indicate higher weights. It is clearly visible that the color segmentation gains more importance on fine structures such as the hands and the feet of the person. For finally composing the segmented person with a new background B, blending between foreground color F and background color B is used and the new intensity I at pixel $\boldsymbol{x}$ is computed as:

$$I(\boldsymbol{x}) = \alpha(\boldsymbol{x})F(\boldsymbol{x}) + (1 - \alpha(\boldsymbol{x}))B(\boldsymbol{x}) \tag{4.22}$$

Note that the blending factor α is computed using either the variance of the depth image or the amplitude information of the Time-of-Flight camera as it is calculated in equation 4.21 using the variance or amplitude weighted depth difference $dv(\boldsymbol{x})$. Figure 4.17 shows the segmentation results of the combined approach. At some particular difficult points, in this

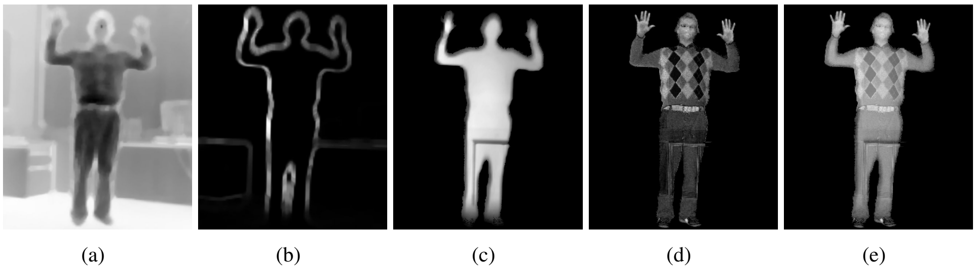


Figure 4.13.: Weighting images:(a)+(b) variance weight image and amplitude image (uncertainty) $v(\boldsymbol{x})$, (c) variance weighted depth difference $dv(\boldsymbol{x})$, (d) MoG weight $\Delta c_n(\boldsymbol{x})$ and (e) combined weight image $\alpha(\boldsymbol{x})$.

example the shoes of the person, the segmentation is not entirely correct, because the similarity between the white shoes and the gray floor is too high after color normalization. The improved segmentation is clearly visible at hands, hair, the silhouette and feet of the person. The current implementation is not optimized for speed, but to quantify the possibilities I will give some numbers of the current implementation. Warping the depth to the CCD image takes $\approx$ 20ms, applying MoG to a color image takes $\approx$ 140ms and segmenting the image on the GPU takes $\approx$ 20ms including uploading the images to textures and readout of textures to images. At the moment two shader passes are used which can be reduced to one. MoG, the limiting factor, is currently executed on the CPU, parallel to the final segmentation. Transferring it to the GPU will significantly speed up the process.

4.2.5. Results

To quantitatively evaluate the segmentation the presented approaches are compared in this section concerning their segmentation accuracy. The following five approaches are compared:

1. Segmentation by background subtraction (4.2.2)
2. Segmentation by Mixture-of-Gaussians on color (4.2.3)
3. Segmentation by Mixture-of-Gaussians on color and depth (4.2.3)
4. Combined segmentation by weighting Mixture-of-Gaussians on color and background subtraction on depth with discontinuity handling using the depth variance (4.2.4)
5. Combined segmentation by weighting Mixture-of-Gaussians on color and background subtraction on depth with discontinuity handling using the amplitude image of the ToF-camera (4.2.4)

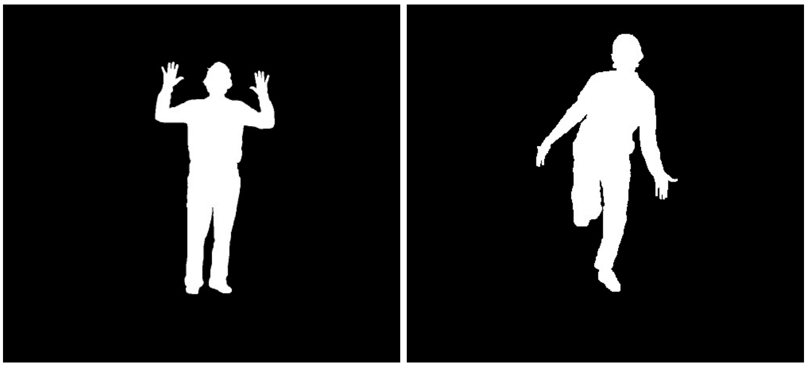


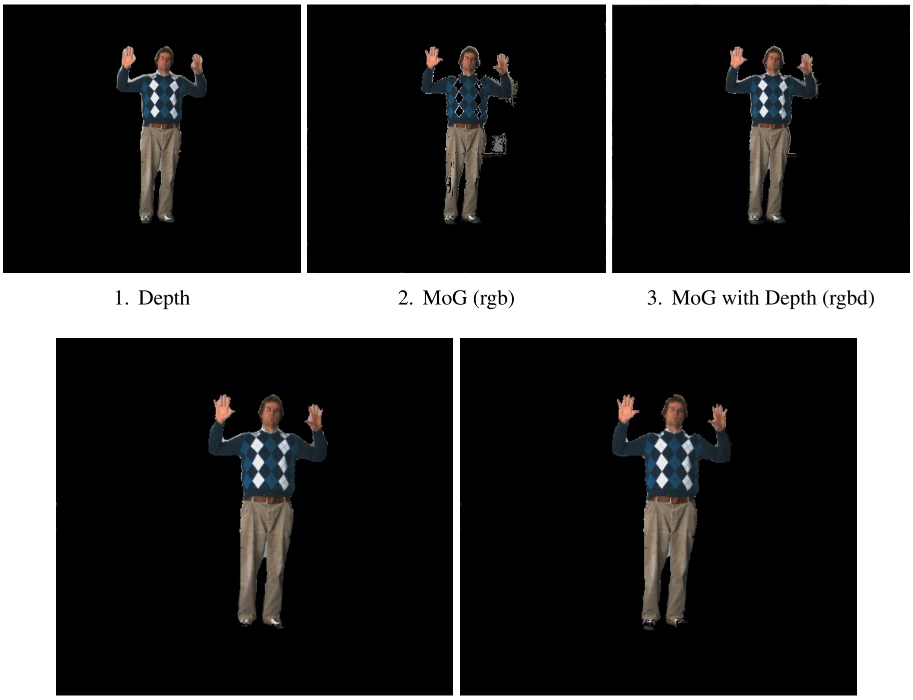
Figure 4.14.: Manually annotated ground truth silhouette images for quantitative evaluation in tables 4.1 and 4.2.

Approach Nr.	Correct pixel #/%	False positives #/%	False negatives #/%	Total error #/%
1.	223125 / 99.167	1542 / 8.287	333 / 1.79	1875 / 10.076
2.	221207 / 98.314	2320 / 12.468	1473 / 7.916	3793 / 20.384
3.	222098 / 98.710	2680 / 14.402	222 / 1.193	2902 / 15.595
4.	223782 / 99.459	599 / 3.219	619 / 3.327	1218 / 6.546
5.	223782 / 99.471	561 / 3.015	630 / 3.386	1191 / 6.4

Table 4.1.: Segmentation evaluation for image shown in figure 4.15. The image consists of 225000 pixel of which 18087 have been manually selected as foreground.

The approaches are evaluated with the depth maps warped to the domain of the color camera. Tables 4.1 and 4.2 show the evaluation results of the foreground segmentation for the different approaches. The image has been labeled by hand to allow quantitative evaluation. The table shows the number (#) of correctly classified pixel, which describes how many pixel have been correctly identified as fore- or background, how many false positives (detected as foreground, but belonging to background) and how many false negatives are produced by the different approaches. Percentages of matching pixel are relative to the number of pixel, percentages of false positives and negatives are given relative to the number of foreground pixel.

The approach based on depth thresholding mainly suffers from boundary errors which is an error introduced by low sensor resolution and the warping of the depth map to the domain of the color camera. Mixture-of-Gaussian mainly suffers from either under- or oversegmentation. In this evaluation three distributions and the best parameters that could be found have been used. While the approach delivers good results at the boundaries, the trousers and the white areas



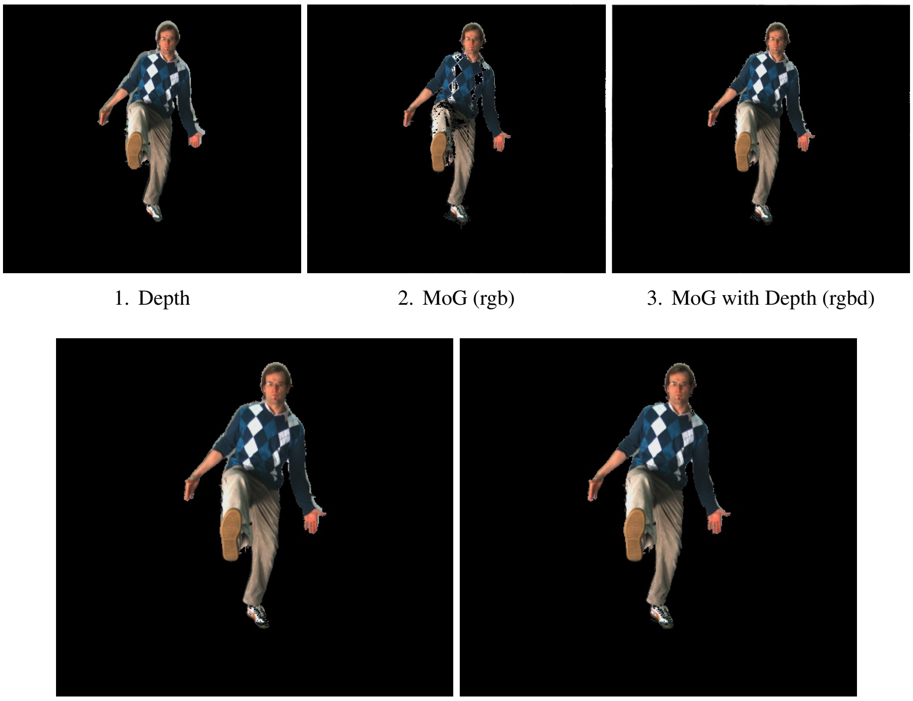
1. Depth 2. MoG (rgb) 3. MoG with Depth (rgbd)

4. MoG with variance weighted depth 5. MoG with amplitude weighted depth

Figure 4.15.: Segmented images, corresponding to table 4.1.

Approach Nr.	Correct pixel #/%	False positives #/%	False negatives #/%	Total error #/%
1.	221054 / 98.246	3512 / 16.837	434 / 2.081	3946 / 18.918
2.	221128 / 98.280	1944 / 9.320	1928 / 9.243	3872 / 18.563
3.	221522 / 98.454	3300 / 15.821	178 / 0.853	3478 / 16.674
4.	222704 / 98.980	1955 / 9.372	341 / 1.635	2296 / 11.007
5.	223297 / 99.243	1375 / 6.592	328 / 1.572	1703 / 8.164

Table 4.2.: Segmentation evaluation for images of figure 4.16. The image consists of 225000 pixel of which 20859 have been manually selected as foreground.



1. Depth

2. MoG (rgb)

3. MoG with Depth (rgbd)

4. MoG with variance weighted depth

5. MoG with amplitude weighted depth

Figure 4.16.: Segmented images, corresponding to table 4.2.

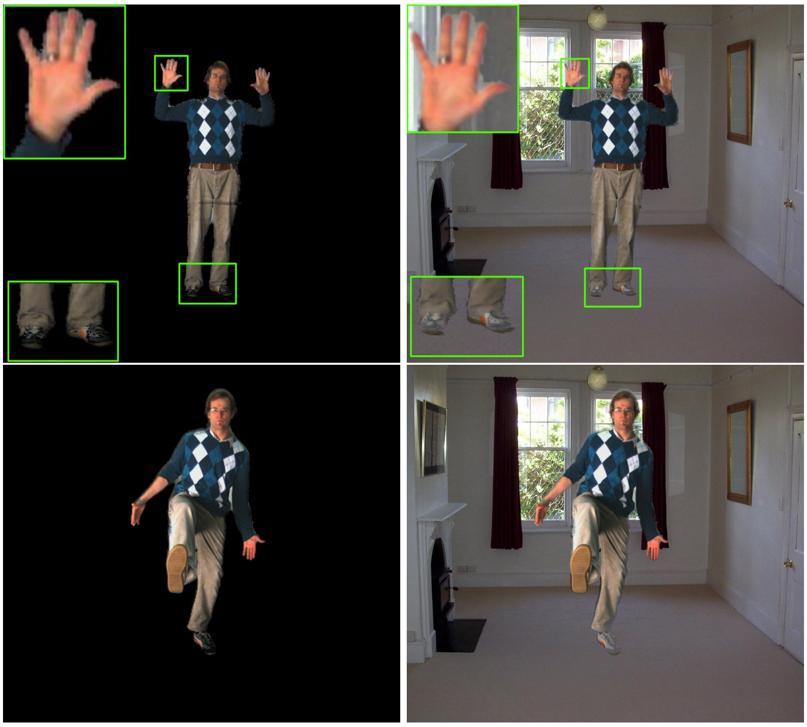


Figure 4.17.: Final result: Blended with black background and with new background image using equation 4.22. Improved regions are partially marked and enlarged.

on the body of the person are erroneous. In total it delivered the worst results. In combination with depth as forth channel the results could be improved which is mainly due to the decrease in false negatives. These are the areas which have been classified as background by the solely colorbased MoG-approach. The combined segmentation approach (see section 4.2.4) using the amplitude images of the ToF-camera as additional weighting factor outperforms the other approaches.

4.3. Datastructures for 2D/3D-Sensor Data

After calibration, noise reduction and optional flying pixel removal and segmentation, the ToF-camera measurements are now aligned with the color information and ready to be processed. As already mentioned, ToF-cameras are real-time capable and measure the distance to surface points. This also means that a geometric point or feature is measured multiple times by the camera delivering slightly different measures due to noise and environmental influences. Furthermore ToF-cameras record deformations of objects in real-time. The challenge is to represent, store and access the recorded data in an adequate manner. A suitable datastructure for holding ToF-camera data therefore has to have the following capabilities:

- Hold intensity and depth information simultaneously
- Represent full 3D geometry and occlusions
- Fuse multiple measurements inherently
- Represent dynamic scene content
- Hold different data for different spatial viewpoints
- Generate a dense surface
- Run-time and storage efficiency

4.3.1. Literature

At first I want to summarize what solutions are used in the literature. Simple, unstructured 3D point clouds are used in [HJS08], [WJH$^+$07], [JHS07], [JWB$^+$06] and [MFD$^+$09]). In [SBKK08], [BSBK08] and [PMS$^+$08] we chose the representation of the data as a 2.5 dimensional panoramic image that encodes color and depth in a 2.5D representation, for example in planar, cylindrical or spherical coordinates. Shade et al. [SGHS98] introduced the layered depth images (LDI). A layered depth image is an image in which at every position multiple depth and color measurements are stored corresponding to the line of sight through that pixel. Volumetric models divide the space into volumetric entities of a given size. The most widely known and used model is the Voxel representation as used in [CL96]. Several tree structures have been discussed (e. g. in [CCV85],[Sze93]), which provide a hierarchical nature to store and access image- and three-dimensional data.

4.3.2. Discussion on Datastructures

Point Cloud

Applying the projection matrix of the ToF-camera and equation 2.12, the delivered depth image can be transformed into an unstructured 3D point cloud. Considering a natural scene, the noise present in the measurements and the given limited accuracy of the ToF-camera, every scene point results in different measurements for consecutive frames, which creates multiple 3D points for the same scene point. This results in a large amount of points which do not have any neighboring relations. Additionally, closed surfaces are not represented as such but split up into a number of independent points. Furthermore, there is no updating of already measured scene points. Averaging over time or measuring points multiple times can increase robustness towards outliers. The Delaunay triangulation [Del34] is often used to construct a closed surface from such an unstructured point cloud. This is a very demanding task which can often not be solved to satisfaction.

2.5D Panoramic Image

A 2.5 dimensional panoramic image encodes color and depth in a 2D representation, for example in planar, cylindrical or spherical coordinates. With the representation as panoramic image, measurements which are taken with a rotating camera head can be fused. Multiple measurements of the same points are fused by averaging in the image, making the result more robust towards outliers. Unfortunately, this representation has some disadvantages. It is by nature only a 2.5 dimensional representation of a three-dimensional scene and occlusions cannot be represented. Furthermore, dynamics are not realizable in an efficient way. An advantage, however, is that from a panoramic depth image it is easy to construct a closed surface, such as a triangle mesh by connecting neighboring pixel in the panoramic images, which can be rendered efficiently on the GPU. An example of a 2.5D panoramic representation is discussed in section 5.1.2.

LDI, 2.5D Layered Depth Image

Shade et al. [SGHS98] introduced the layered depth images (LDI). A layered depth image is an image in which at every position multiple depth and color measurements are stored corresponding to the line of sight through that pixel. LDIs were developed for image-based rendering, which describes an approach to generate new interpolated views of a scene. Thus LDIs are capable of representing occlusions or dynamics, but not both at the same time. LDIs are constructed for a distinct camera position. Generating a LDI for a certain camera position includes the warping of all depth images to the viewpoint and internal camera parameters of this camera. LDIs can be viewed as generalization of 2.5D panoramic images to multiple occlusion layers. Rendering a scene from a different view requires to perform the incremental warping procedure. Chang et al. [CBL99] extended this approach to a hierarchical approach using Octrees.

The representation of multiple measurements as point cloud, panoramic image or LDI is not optimal. The main disadvantages are either the missing fusion of measurements and the lack of neighboring relations, or the missing possibility to represent occluded objects and dynamic content. The obvious step towards an optimal representation is to use a volumetric representation of the scene which also offers the possibility to represent neighboring relations and to fuse measurement.

Voxel Volumes

The voxel representation divides a predefined space in cells of a certain size. It offers a lot of possibilities, such as a real 3D representation of the scene and measurement fusion in the cells of the voxel volume. Furthermore neighboring searches and clustering can be executed in a straightforward way. The main disadvantage is, that the volume has to be defined and allocated entirely which leads to a large memory consumption. Volumes of large size and fine resolution require a significant amount of memory and are not efficient. The solution to this problem would be to allocate the memory only on demand. Tree-like datastructures implicitly provide this functionality. The generation of a closed surface is often solved by the usage of Marching Cubes [LC87] or related algorithms.

4.3.3. The Octree Datastructure

In conclusion a volumetric representation of data is needed to provide all required characteristics. The Octree representation [Mea82] combines the advantages of a volumetric model with a hierarchical data structure. Further advantages are ease of concept and implementation through recursion, storage efficiency and flexibility concerning volume content. A very early approach to represent depth images in an Octree is found in [Con84] and later in [LC94]. I will introduce the Octree structure, show how the fusion of measurements is realized and derive why it is an excellent choice to represent ToF-camera data for certain applications.

Building the Octree Structure

The Octree data structure is an oriented graph structure which represents a part of three-dimensional space. It is a recursive datastructure in which every Octree node has eight children and one father. Each Octree node consists of a position in 3D, a size and an Octree element, representing the information about space in this volume element. In general every Octree node can contain an Octree element but for most applications it is sufficient that the leafs of the tree have an Octree element. Leafs are Octree nodes which do not have any children and the size of the Octree leafs is the resolution of the Octree. The size of the Octree leaves do not have to be the same in the whole Octree, which allows different resolutions within the tree.

In the beginning of the Octree construction it only holds one element with a certain size containing the bounding volume to be modeled. The procedure of adding elements to an Octree is pictured in figure 4.18. Before adding measurements the spatial resolution of the Octree has

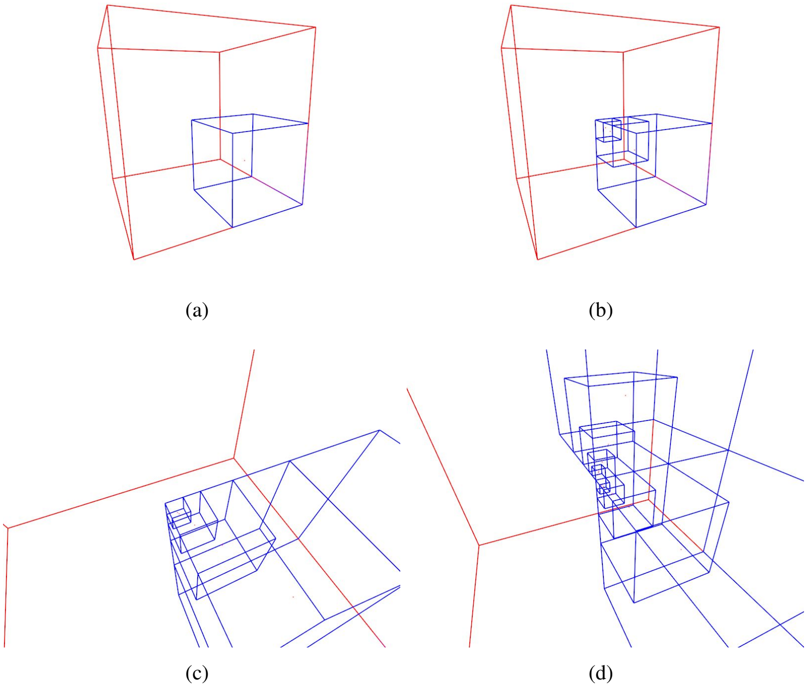


Figure 4.18.: Principle of Octree data structure: The space is subsequently divided in eight equal sized cubes until the desired cube-size is reached. Then the data is added to the Octree.

to be defined. This definition is made from knowledge we have about the used ToF-camera. For current ToF-cameras the manufacturers promote a repeatability between 5 and 200mm. So this can be used as guidance to select the minimal Octree cell size. In the experiments and examples I have mostly chosen a resolution of $25mm \times 25mm \times 25mm$. The resolution also affects other aspects such as memory consumption, time needed to insert data in the datastructure and rendering times. An exemplary comparison of the effects is listed in table 5.1, p. 84. To add measurements the algorithm starts at the top node by checking if the size of the node to add is bigger than half the size of this node. If this is the case a leaf is reached and the node is inserted here. If this is not the case the algorithm calculates to which of the eight child nodes the node to add belongs. (In figure 4.18 (a) the element to add does not have the size of the outer (red) cube, therefore a sub-cube in the front bottom right corner is created.) If the child node does not already exist it is created and this node is used in further branching. In figure

4.18 (b) and (c) the subsequent branching is shown. Sub-image (d) shows a second element added to the Octree.

Measurement Fusion

Depending on the requirements, different data can be stored in the elements of the Octree. This reaches from simple uncolored 3D points over colored points with normals up to small oriented surface patches with texture. Using natural scenes and lighting, objects can look different when viewed from different viewpoints. This can be included as well as different appearances depending on daytime or other factors. As every child of an Octree is an Octree itself, sub-trees can easily be added to the current scene. In contrast to simple point clouds where in general no measurement fusion is possible, the volumetric representation of the Octree allows to fuse the measurements while adding them. In the experiments simple colored 3D points with an additional radius component are used as Octree elements and to fuse multiple measurements the weighted average of the new and the already existing position and color is used.

Rendering / Surface Generation

Rendering Octrees is fast and straightforward. It consists of traversing the graph and rendering all active nodes in the Octree. How the Octree elements are rendered is mainly due to the intended usage. If only a sparse point cloud is needed, the Octree can be rendered as points with color and a certain size. This is shown in figure 4.19 (a) and (c). If a closed surface is needed, e. g. for depth testing many existing approaches are usable. For example point splatting (cf. [RL00]) is a feasible method exploiting GPU shader language. Examples of point splat rendering can be found in figure 4.19 (b) and (d). Not only point based rendering methods are applicable, for example in [Sam89] Samet shows how ray-tracing can be efficiently performed using Octrees. Additionally the surface reconstruction methods applicable to voxel volumes can also be used to construct closed surfaces for Octrees such as Marching Cubes [LC87].

4.4. Conclusions

Preprocessing depth maps is necessary due to the noise in the depth images which has not been handled by calibration. A filtering to remove independent noise has to be applied. Filters which do not mix foreground and background are favorable. The presented bilateral filter is a good choice as it is capable of smoothing foreground and background without merging them, which is in contrast to mean and Gaussian filtering. The presented warping algorithm on the GPU transfers the depth measurement in the perspective of any other camera observing the

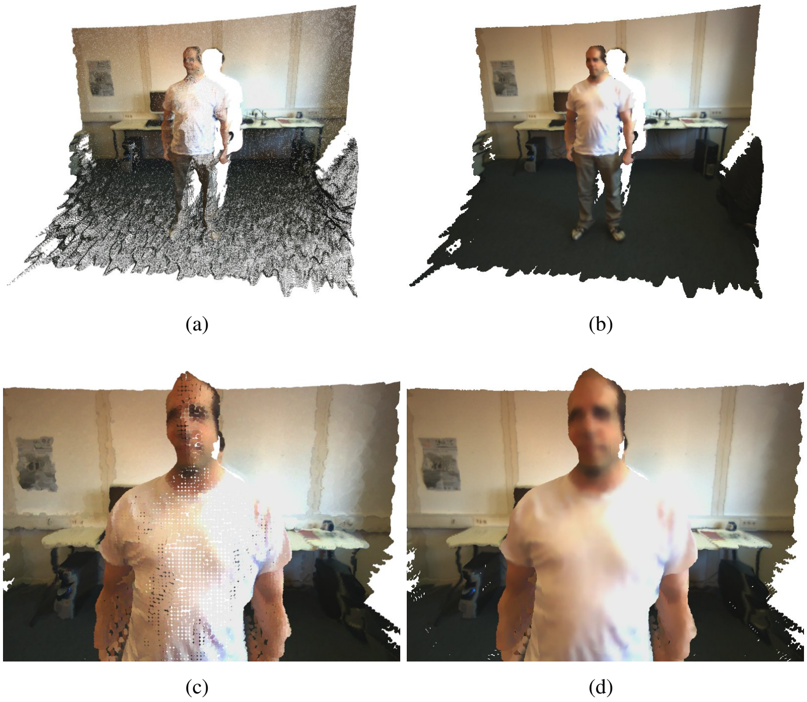


Figure 4.19.: Comparison of point rendering (a),(c) and point splatting (b),(d).

scene. This generates a higher resolution 2D/3D- camera image. The connections between fore- and background and flying pixel are another challenge in the usage of ToF-depth measurements. This is tackled by variance analysis and these false measurements can be widely reduced.

The choice of an appropriate datastructure for storing the ToF-data is crucial and dependent on the application. If fusion of measurements and multiple representations of geometric entities is required a volumetric representation using tree structures such as an Octree is advisable.

"The most incomprehensible thing about the world is that it is comprehensible."
Albert Einstein (1879 - 1955)

5 Scene Analysis with 2D/3D Sensor

The preceding chapters studied the calibration of ToF-cameras, the preprocessing of the delivered images and suitable datastructures to store combined depth and intensity information. This chapter will focus on the exploitation of the presented approaches and the ToF-camera for the analysis of rigid and non-rigid scenes.

5.1. Rigid Scene Modeling

Rigid environment reconstruction as well as object reconstruction is an extensively investigated and used method of computer vision. For many applications such as cultural heritage, architecture or navigation it is essential to capture geometry and intensity information of an object or environment and represent it as a full three-dimensional model which is correct on terms of scale and completeness. In section 2.2, p.7 the shortcomings of different approaches to compute correct depth information have been discussed. Correct depth information is a mandatory precondition for the reconstruction of scenes. ToF-cameras provide reliable dense depth maps of the environment. Often intensity information is also mandatory in the reconstructed model. Hence the ToF-camera is combined with a standard intensity camera which provides intensity information for the constructed three-dimensional model.

5.1.1. Prior Work

Well-known approaches for environment reconstruction solely based on camera images are Structure-from-Motion [PKVVG98] and Visual SLAM [Dav03] which uses additional sensors and builds only a sparse map and focuses more on the localization than on the reconstruction. The main problem of solely image based approaches is the computation of dense depth maps for final geometry generation. This problem is largely simplified by the usage of ToF-cameras.

In this literature review I will focus on publications which use ToF-cameras for reconstruction. ToF-cameras have a limited operation range and field of view due to their operation principle. Reconstruction of larger environments therefore relies on joining multiple measurements. Pioneering work in environment reconstruction or mapping with ToF-cameras was achieved in [PMS$^+$08] in which a fisheye camera is used together with the depth measurements from a ToF-camera to estimate the pose of a robot and multiple depth measurements are joined in a common model. Huhle and Jenke et al. [HJS08] [JHS07] [JWB$^+$06] use ICP on the point clouds produced by ToF-cameras in combination with an inertial sensor to estimate the camera movements and in order to reconstruct the interior of buildings. May et al. [MDH$^+$08] [MFD$^+$09] use the ICP algorithm with a relaxation strategy and frustum culling to register consecutive frames.

Guan et al. [GFP08] focus on the reconstruction of objects, viewed from different viewpoints with multiple ToF- and CCD-cameras. They fuse the measurements of ToF- and intensity by a probabilistic method using depth and silhouette information.

Kim et al. [KTD$^+$09] use a multi-camera setup consisting of ToF- and CCD cameras and use ToF-measurements as a coarse initialization model which they refine with stereo and silhouette information.

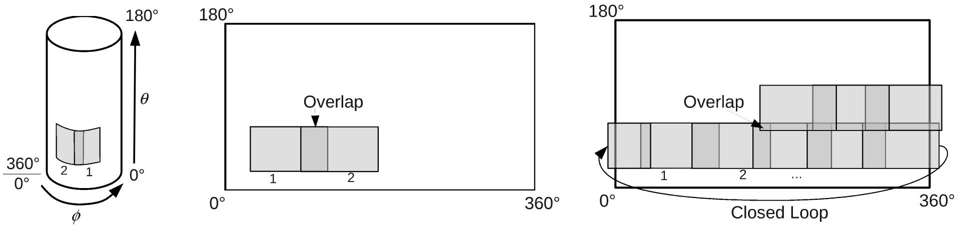
The systematic exploitation of ToF-cameras for precise indoor environment reconstruction with high quality depth and color has not been tackled so far. In this thesis I want to present two approaches for two model reconstruction cases. The first approach (see also [BSBK08]) uses a pan-tilt unit (PTU) instead of possibly erroneous camera pose estimation with ICP or variants together with panoramic images for accurate dense environment reconstruction. This is discussed in the following section 5.1.2. The second approach investigates the suitability of ToF-cameras for the reconstruction of smaller rigid objects without the usage of special tracking devices such as sensors or pan-tilt units. Instead, the pose of the camera relative to the object is computed from intensity features and depth, and the measurements are fused in a 3D Octree datastructure (see also [SK09]). The approach chosen in this thesis is presented in section 5.1.3.

5.1.2. Environment Reconstruction with Pan-Tilt Unit

Mandatory for the correct reconstruction of environments is the avoidance of drift during the reconstruction as this would distort the result. Hence this first approach for the reconstruction of indoor environments uses a ToF- and a CCD-camera mounted on a pan-tilt unit (PTU) as shown in figure 3.1 (a), p.16. The cameras are swept on a predefined path using the PTU. The fusion of measurements and the representation of reconstructed geometry can be realized in different ways of which two will be analyzed, the cylindrical 2.5D panoramic image and the Octree.

Reconstruction with 2.5D Panoramic Image

The idea behind the usage of 2.5D panoramic images for scene reconstruction is to project all depth and color information into a depth- and color- image, according to a cylindrical or spherical projection model, and to construct a 3D model from these depth- and color- images using triangulation. In this work a cylindrical representation of the 2.5D panoramic image and a cylindrical projection model is used, as with a cylindrical model less distortions are generated compared to a spherical representation. The underlying 2.5D panoramic image is represented as two images, one for color and one for depth. They form the hull of a cylinder and the resolution of the final model can be chosen as the resolution of the underlying images. In this exemplary case the resolution has been chosen as 2000×1000 pixel.
Figure 5.1 shows the principle of projecting multiple images to a cylindrical panoramic image. The rotation and translation of the underlying cylindrical projection are chosen similar to the initial position of the CCD-camera as a $4{\times}4$ matrix $\mathsf{P}_{ccd,ext} = [\mathsf{R}_{ccd,ext}|\mathbf{T}_{ccd,ext}]$, aligned with the initial position (0,0) of the PTU. The cylindrical projection model is formed by two parameters, the angle ϕ in horizontal direction, which is chosen as $0° - 360°$ and the angle θ in vertical direction which is chosen as $0° - 180°$. The cylindrical projection maps the rays of the camera coordinate system to pixel in the panoramic image.



(a) (b) (c)

Figure 5.1.: The principle of projecting images to a cylindrical image. In (a) the cylinder is shown with the angles ϕ (horizontal) and θ (vertical) and two exemplary images 1 and 2 projected to the hull of the cylinder according to the current rotation of the PTU. In (b) and (c) the cylinder is unrolled. The final values in overlapping regions are computed by weighted averaging of depth and intensity values.

The PTU moves the camera and the angular movements of the PTU are represented as a rotation matrix R_{ptu} with rotation of α around the $x-$ (tilt) and β around the $y-$ (pan) axis.

$$\begin{aligned}&\boldsymbol{R}_{tilt} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{pmatrix}, \boldsymbol{R}_{pan} = \begin{pmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{pmatrix} \\ &\boldsymbol{R}_{ptu} = \boldsymbol{R}_{pan}\boldsymbol{R}_{tilt}\end{aligned} \tag{5.1}$$

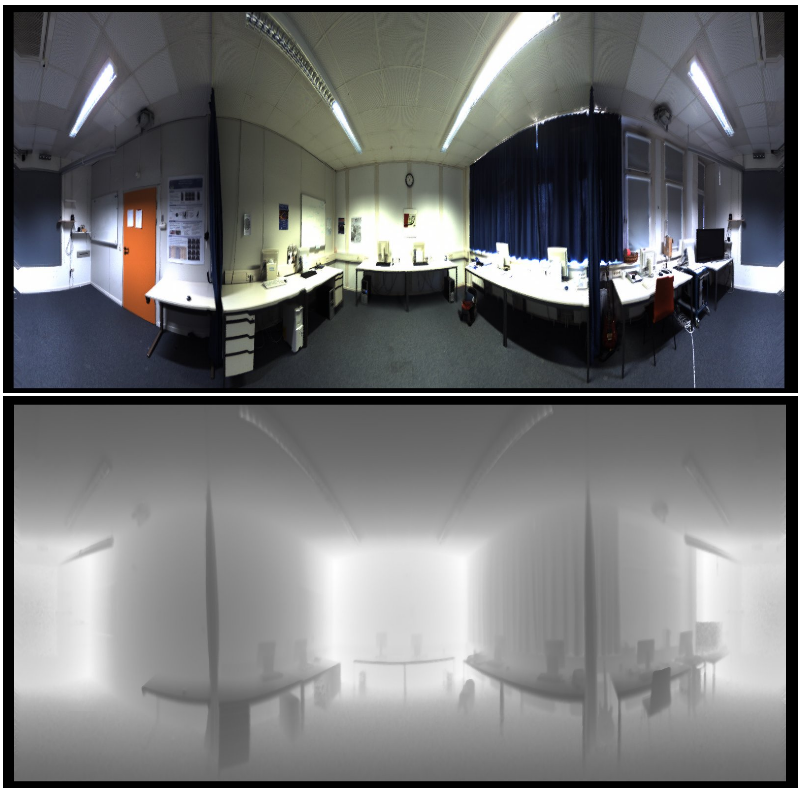


Figure 5.2.: 2.5D panoramic images for texture (top) and depth (bottom).

The origin of the coordinate system is set to the initial position of the PTU without rotation. As the tool center point (TCP) of the PTU and the camera centers are not identical this has to be taken into account for calculating the new projection. The offset between tool center and camera center is a transformation consisting of rotation $\boldsymbol{R}_{tcp}$ and translation $\boldsymbol{T}_{tcp}$. The rotation and translation of the TCP are combined in a 4×4 Matrix $\boldsymbol{P}_{tcp}$ and rotation and translation of the PTU are combined in $\boldsymbol{P}_{ptu}$. These are concatenated as $\boldsymbol{P}_{ptu,tcp}$:

$$\begin{aligned} \mathsf{P}_{tcp} &= [\mathsf{R}_{tcp}|\mathbf{T}_{tcp}] \\ \mathsf{P}_{ptu} &= [\mathsf{R}_{ptu}|\mathbf{I}_{1\times 4}] \\ \mathsf{P}_{ptu,tcp} &= \mathsf{P}_{ptu}\mathsf{P}_{tcp} \end{aligned} \tag{5.2}$$

In which $\mathbf{I}_{1\times 4} = (0,0,0,1)^T$ is a homogeneous vector, R_{ptu} and R_{tcp} are the homogenized versions of the rotation matrices R_{ptu} and R_{tcp} and $\mathbf{T}_{tcp}$ is the homogenized version of $\boldsymbol{T}_{tcp}$. The new external camera parameters $\mathsf{P}_{ccd,cyl}$ in the cylindrical coordinate system are then calculated by concatenating $\mathsf{P}_{ptu,tcp}$ and the external camera transformation $\mathsf{P}_{ccd,ext}$:

$$\mathsf{P}_{ccd,cyl} = \mathsf{P}_{ptu,tcp}\mathsf{P}_{ccd,ext} \tag{5.3}$$

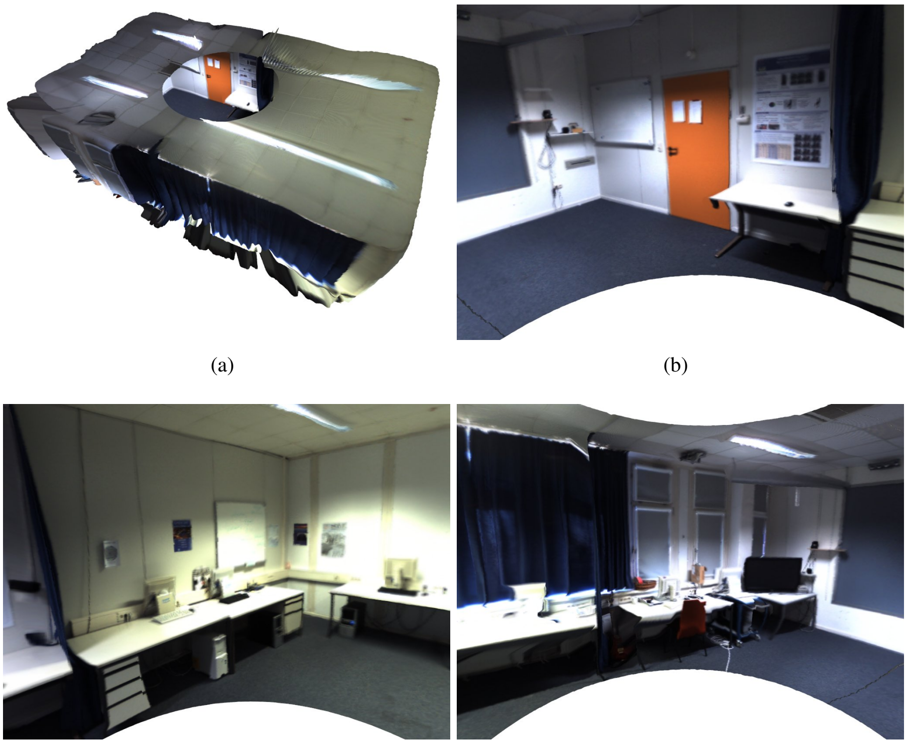
(Note that initially the external parameters of the cylindrical projection were chosen similar to the externals of the ToF-camera.) ToF- and intensity image do not share a common center. By warping the depth images to the CCD camera's perspective, combined depth and intensity images are obtained (see section 4.1.2, p.53). Every combined depth and intensity image is then projected to the cylindrical panorama using the external parameters of $\mathsf{P}_{ccd,cyl}$ and the internal parameters of the cylindrical projection. Because this is a forward projection not all pixel in the panorama will be filled with a new value by projection. The results in figure 5.2 where obtained by applying nearest neighbor interpolation. This means that in a certain range undefined pixel are filled with averaged valid surrounding pixel.

Besides holes due to forward projection also overlapping regions in the panorama occur. These are regions in which multiple measurements are taken. The final values in these regions are calculated using a weighted mean of all measurements. To smooth the effects of lighting changes the single images are additionally weighted, radially decreasing from the center of the image to the outer edges.

The 2.5D panoramic image is converted into a triangle-mesh using triangulation of neighboring pixel and a three-dimensional surface representation is generated as seen in figure 5.3. As the camera head is rotating in the middle of the room, the advantages of the 2.5D representation outperform the disadvantages in this case.

Reconstruction with Volumetric Octree

To prove the suitability and usability of the volumetric Octree representation for the purpose of environment reconstruction, the combined depth and color images of the camera head were also used to construct the Octree model shown in figure 5.4 for which a minimum cell size of $25mm$. Note that the size specifies the size of one side of the cell and the volume of one cell is therefore $25mm \times 25mm \times 25mm$. Additionally this experiment was chosen to



(a)

(b)

(c)

(d)

Figure 5.3.: Triangle mesh generated from 2.5D panoramic image. (a) View of the complete environment model from outside, (c) -(d): Views of the interior.

compare the reconstruction approaches. Using Octrees requires a merging strategy for Octree cells. Unoccupied leaves are initialized with a weight of 1 if a measurement is added. Every additional measurement which falls into this cell is in this case added with a weight of w, resulting in the weighted average of old and new measurement. The fusion itself fuses color $\boldsymbol{c}$ as well as 3D position $\boldsymbol{X}$. The new cell content is calculated as:

$$\begin{aligned} \boldsymbol{X} &= w\boldsymbol{X}_{new} + (1-w)\boldsymbol{X}_{old} \\ \boldsymbol{c} &= w\boldsymbol{c}_{new} + (1-w)\boldsymbol{c}_{old} \end{aligned} \tag{5.4}$$

in which w is chosen as 0.5.

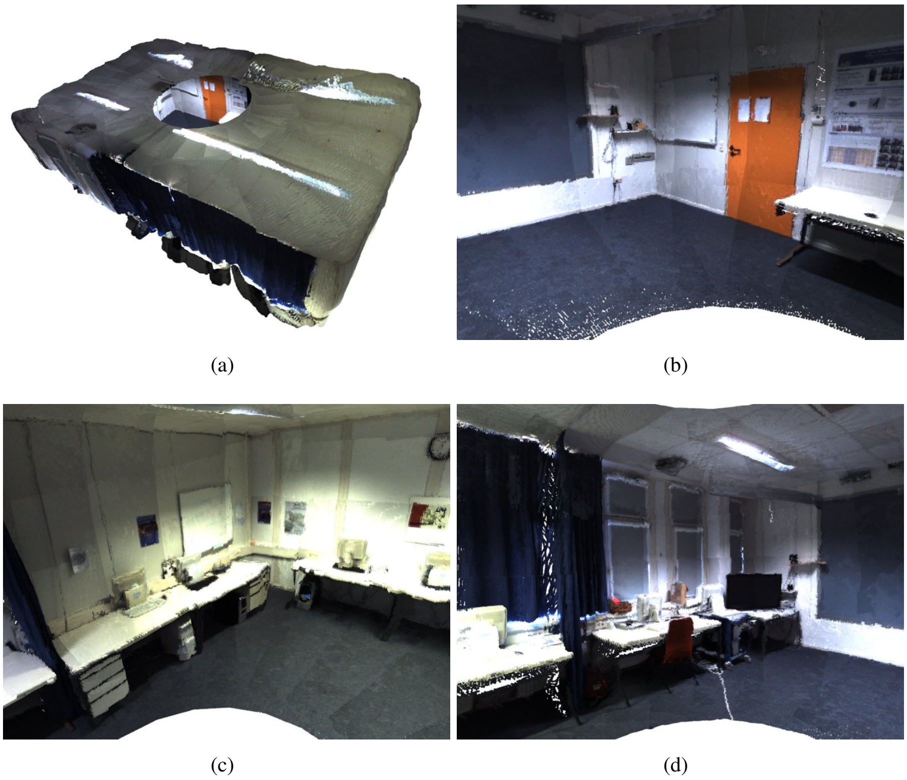


Figure 5.4.: Environment model as Octree, fused from 115 images, cell-size $25mm$. In comparison to figure 5.3 a slight degradation due to point rendering is visible.

Evaluation

A visual comparison of both models after rendering shows that the surface mesh is somehow smoother than the Octree rendering, but both models are comparable in visual quality. The Octree processing can also be compared to simple 3D point cloud construction and rendering. Table 5.1 compares the performance of the Octree for the reconstruction of the indoor scene for different cell sizes and simple 3D point cloud processing. The scene has been constructed for different cell sizes. Insertion of one image with 176×144 pixel and merging it with existing content is performed in 44ms ($100mm$) to 69ms ($25mm$). This includes the computation of 3D points which is also necessary for the point cloud. So the real traversal in the tree and the merging takes between 6ms ($100mm$) and 30ms ($25mm$). Traversing the tree and collecting all valid elements takes between 11ms and 192ms and rendering all points takes 2ms to 32ms. From these numbers it is observable that at a cell size of $50mm$ the Octree rendering is comparable to pure point cloud rendering concerning speed. The tests have been carried out using an Intel Core2 CPU 6600 @ 2.40GHz with 4GB RAM and a NVidia GeForce 8800 GTS GPU. The indoor reconstruction scenario shows the suitability of the Octree datastructure to represent ToF-measurements.

	Pointcloud	Octree $25[mm]$	Octree $50[mm]$	Octree $100[mm]$
Elements	2.744.772	1.617.140	470.230	98.102
Insert in Octree	38,48 ms	68,80 ms	51,46 ms	44,38 ms
Collect Elements	-	191,55 ms	51,32 ms	10,80 ms
Render Elements	59,83 ms	32,30 ms	9,69 ms	2,15 ms

Table 5.1.: Comparison of Octree cell size and 3D point cloud performance for the environment model. See text for details.

Besides the performance is the storage efficiency a crucial point concerning the choice for a data representation. Comparing image based representations such as 2.5D panoramic images or LDIs and 3D representations it is obvious that volumetric representations will require more storage because instead of one value for depth, three values for the 3D position are saved. The storage comparison is given in table 5.2.

	Panorama	Mesh	Pointcloud	Octree	Octree	Octree
		(VRML)		$25[mm]$	$50[mm]$	$100[mm]$
Elements	2 000 000	2 305 030	2 744 772	1 617 140	470 230	98 102
In RAM[MB]	-	1 085.85	115.55	735.26	200.02	41.68
On disk[MB]	13.67	136.37	174.18	344.64	100.22	21.19

Table 5.2.: Comparison of storage requirements. “Elements” denotes the number of points or triangles, The row denoted “In RAM” denotes the real storage consumption after loading the data into memory and the row “On disk” is the file size for saving the data to disk as images (binary), VRML or Octree (ASCII).

Two images are used for the panoramic image, one holding the depth values and one for the intensity information. For depth float values and for intensity RGB values in unsigned char with one byte per channel are used, which results in 4 bytes for a float and 3 bytes for intensity for every pixel. The generated triangle mesh is bigger as for every pixel the 3D position and connection information is saved as well. The texture is saved as an image as above and projected on the geometry. The storage requirement is dependent on the texture resolution. In this case full resolution (2000×1000) has been used and the storage usage is over 1 GB. Lower numbers are observable for the pointcloud as only the 3D position and intensity values are saved. For the octree the 3D position of the cell, the 3D position of the point, four cornerpoints of each cell, the cell size and the pointers which connect tree items are necessary.This produces a far higher memory usage than the representation as pointcloud, but for an octree with a cell side length of $50mm$ the memory consumption is only moderately increased. This shows that large data sets which cover large environments, in this case approximately $5 \times 7 \times 3$ meters and larger, are manageable with a resolution of $25mm$ and smaller using octrees.

For a qualitative rating of the reconstruction accuracy the original room geometry has been measured with a laser distance measurement device and compared to the environment model generated with the 2.5D panoramic image approach. Measurement of the model has been carried out from wall to wall at several location and the results have been averaged. As can be seen in table 5.3, the mean of the measurements differs 44-177mm from the real geometry. This is equivalent to an average error of 1.25- 2.03%.

Dimensions:	length$[mm]$	height$[mm]$	width$[mm]$
room size (ground truth):	8528	2985	5528
model size (mean):	8705	3029	5598
mean error:	177	44	70
mean error(%):	2.03%	1.45%	1.25%

Table 5.3.: Top Lines: Room size measured in real scene and mean model size of several measurements taken in the reconstructed model. Bottom lines: Mean error between reconstructed model and ground truth room.

5.1.3. Object Reconstruction with Pose Estimation

The already mentioned second reconstruction objective is the reconstruction of smaller rigid objects such as persons, furniture or machine parts. This is an often requested process in the manufacturing industry, for quality supervision and film industry and hence intensively investigated. The goal in this part of the work is therefore the reconstruction of a model of a real person or object using a ToF-camera and a true 3D representation with a closed surface is requested. Images from different perspectives shall be added to the model and the object or person can additionally contain self-occlusions, which requires a suitable representation and makes it impossible to use for example 2.5D panoramic images. Hence a real volumetric representation is needed which is capable to represent the three-dimensional data. Additionally the measurements from different perspectives have to be fused in a single model in this volumetric datastructure.

The reconstruction of objects with a single camera requires either to take images from different perspectives or to rotate the object in front of the camera. Estimating the movement of the camera is equivalent to estimating the movement of the object if the static background is neglected.

Pose Estimation with ToF- and CCD-Images

Pose estimation is investigated since a couple of decades in Computer Vision. Very successful approaches using only standard monocular cameras exist (e. g. SLAM in [Dav03]), as well as approaches using stereo camera rigs. Using ToF-cameras for pose estimation offers one crucial advantage, the real-time availability of depth measurement and dense depth maps.

Standard pose estimation algorithms and reconstruction approaches need to determine the distance of a world point to the camera center which is mostly determined only up to scale. Classical Structure-from-Motion approaches ([PKVVG98]) need a certain baseline, a distance between the first two cameras, to triangulate 3D points from 2D correspondences. Without knowledge of the baseline the distance of the 3D points to the camera is only up to scale. Figure 2.3, p.8 depicts these principles. Using multiple cameras of known relative external geometry and internal camera parameters this issue can be solved and the pose of the camera rig can be estimated in metric scale.

The pose estimation of a camera is typically based on the tracking of significant points over an image sequence or matching of significant points between images which picture the same objects. The most widespread feature is the gradient based KLT-Feature which was introduced in [ST94]. In the following I will discuss the pose estimation with CCD- and ToF-camera.

In section 4.1.2, p.53 it was introduced how depth measurements are transferred into the view of an additional color camera. In this section it is assumed that the images are precomputed in this way and call this a 2D/3D image (pair). In a first 2D/3D image pair 2D-KLT-Features are detected on the intensity image and for every 2D feature $\boldsymbol{x}$ the distance $d(\boldsymbol{x})$ to the camera center is extracted from the warped and filtered depth image. Using equation 2.12, p.7 a 3D point $\boldsymbol{X}$ is generated for every feature. From these correspondences, the camera pose can be estimated using the standard camera pose estimation scheme DLT (cf. [HZ04] p.173 and p.73).

The process of pose estimation is distorted by uncertainties at different steps in the algorithm: during the detection of 2D (KLT-) feature points and during the physical measurement of depth using the ToF-camera. This noise is modeled using covariances and predicted to subsequent frames in the process.

Uncertainty Modeling

This section introduces basic variance and covariance notation and shows how the uncertainty of a measurement of a 2D feature and 3D point measurement is modeled. Let $\mathbf{X}$ be a set of measurements $[X_0, ..., X_n]^\mathsf{T}$. The expected value of a measurement X_i, which is equivalent to the mean, is denoted:

$$\mu_i = E(X_i) \tag{5.5}$$

and the covariance $\Sigma_{i,j}$ is calculated as:

$$\Sigma_{i,j} = cov(X_i, X_j) = E[(X_i - \mu_i)(X_j - \mu_j)] \tag{5.6}$$

The covariance matrix Σ_N of a set of N measurements is then:

$$\begin{aligned}\Sigma_N &= \begin{pmatrix} E[(X_1-\mu_1)(X_1-\mu_1)] & ... & E[(X_1-\mu_1)(X_n-\mu_n)] \\ . & & . \\ . & & . \\ . & & . \\ E[(X_n-\mu_n)(X_1-\mu_1)] & ... & E[(X_n-\mu_n)(X_n-\mu_n)] \end{pmatrix} \\ &= E\left[(\boldsymbol{X} - E[\boldsymbol{X}])(\boldsymbol{X} - E[\boldsymbol{X})^\mathsf{T}]\right]\end{aligned} \tag{5.7}$$

The variances σ^2 are located on the diagonal of the covariance matrix Σ_N which are defined as:

$$\sigma_i^2 = var(X_i) = E[(X_i - \mu_i)^2] \tag{5.8}$$

The uncertainty of a 2D point in x and y is directly related to the accuracy of the feature detector. The feature detector (e. g. the KLT-corner detector [ST94]) computes the covariance of a feature from the structure tensor of the image. So the 2×2 covariance matrix of a 2D feature $\boldsymbol{x}$ is:

$$\Sigma_{\boldsymbol{x}} = \eta^2 \begin{pmatrix} \Delta_x^2 & \Delta_x \Delta_y \\ \Delta_y \Delta_x & \Delta_y^2 \end{pmatrix} \tag{5.9}$$

with Δ_i the gradient in direction i and η a scale factor.

To model the uncertainty of the depth measurement by a ToF-camera, every constructed 3D point is additionally modeled by a covariance matrix. The covariance of a 3D point is best modeled as an ellipse in 3D and represented as it's 3×3 covariance matrix. As normally only one measurement of a 3D point at a certain time step is available, the variances $\sigma_i^2, i\epsilon\{x, y, z\}$ of the covariance matrix $\Sigma_{\boldsymbol{X}}$ of a 3D point $\boldsymbol{X}$ are defined in advance.
The biggest uncertainty of the depth measurement is along the viewing ray of the camera. Eventual errors in the measurement affect the distance to the camera and therefore the z component. The uncertainty in the direction of the viewing ray is directly related to the value of the amplitude image of the ToF-camera. The amplitude image expresses the reflectivity of the observed surface. A good reflecting surface delivers high amplitudes and provides a reliable measurement, a scattering or less reflective surface delivers low amplitudes and results in unreliable measurements. Therefore σ_z can alternatively be set from the values of the amplitude image of the ToF-camera. The values have to be scaled to be in the correct range to be used as uncertainty. The ellipse has to be oriented with the ray through the camera center and the 3D point as the direction of the highest uncertainty. Hence the ray r_z through camera center and 3D point $\boldsymbol{X}$ is calculated using equation 2.11, p.6. Rays r_x and r_y, perpendicular to this ray, are constructed for $\boldsymbol{x}$ and $\boldsymbol{y}$ using the cross product:

$$r_d = (0, 1, -1) \tag{5.10}$$
$$r_y = r_z \times r_d \tag{5.11}$$
$$r_x = r_z \times r_y \tag{5.12}$$
$$\tag{5.13}$$

The 3×3 matrices $\boldsymbol{R}_x, \boldsymbol{R}_y, \boldsymbol{R}_z$ are constructed using the outer product:

$$\boldsymbol{R}_x = r_x \cdot r_x^T \tag{5.14}$$
$$\boldsymbol{R}_y = r_y \cdot r_y^T \tag{5.15}$$
$$\boldsymbol{R}_z = r_z \cdot r_z^T \tag{5.16}$$

The 3×3 covariance of a 3D point $\mathbf{X}$ can then be defined as:

$$\Sigma_{\boldsymbol{X}} = \sigma_x^2 \boldsymbol{R}_x + \sigma_y^2 \boldsymbol{R}_y + \sigma_z^2 \boldsymbol{R}_z \tag{5.17}$$
$$\Sigma_{\boldsymbol{X}} = \begin{pmatrix} \Sigma_{xx} & 0 & 0 \\ 0 & \Sigma_{yy} & 0 \\ 0 & 0 & \Sigma_{zz} \end{pmatrix} \tag{5.18}$$

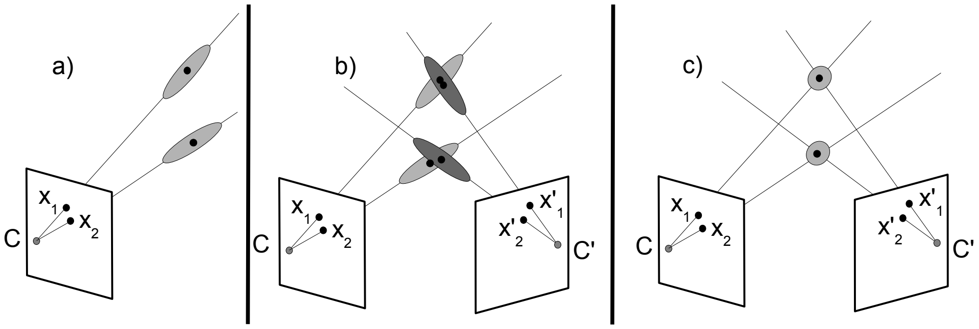


Figure 5.5.: Covariance update method with filtering: a) Calculated covariances for 3D points. b) Second measurement of 3D points and covariances which differ from first due to noise. c) Updated points and covariances.

As the 2D correspondences are tracked over image sequences every 3D point is seen multiple times. Every new measurement decreases the uncertainty of the measurement if it is consistent with the previous measurements. The consistency measurement in this case is the uncertainty ellipsoid. If the new measurement lies within the covariance ellipsoid $\Sigma_{\boldsymbol{X}}$ the corresponding 3D point $\boldsymbol{X}$ and the covariance are updated.

The updated position and covariance of a 3D point is calculated using a Kalman Filter [WB95] together with the Unscented Transform [SJ97].

Measurement Fusion

Figure 5.6 shows the input data for the reconstruction of the person model. While the person is turning on a swivel chair in front of the cameras, the pose of the camera is estimated relative to the person. For the reconstruction 64 image pairs of depth and intensity have been used.

The foreground object is segmented by depth keying from background, since the depth gives an easy cue to object segmentation (see section 4.2.2, p.58). After estimating the object's pose for every image, the depth and intensity elements are added into an Octree. Figure 5.7 shows the resulting fused model. The top left image shows a result after five images, the other images show the integration of 64 images. Note that the person swayed a little during recording, which leads to some errors in the fused model. The bottom image shows the reconstructed person and the estimated camera position as pyramids, as seen from the rigid object coordinate system. The fused model still contains some errors, but the objective is to show the advantages of the volumetric representation compared to other data structures and the feasibility of the approach.

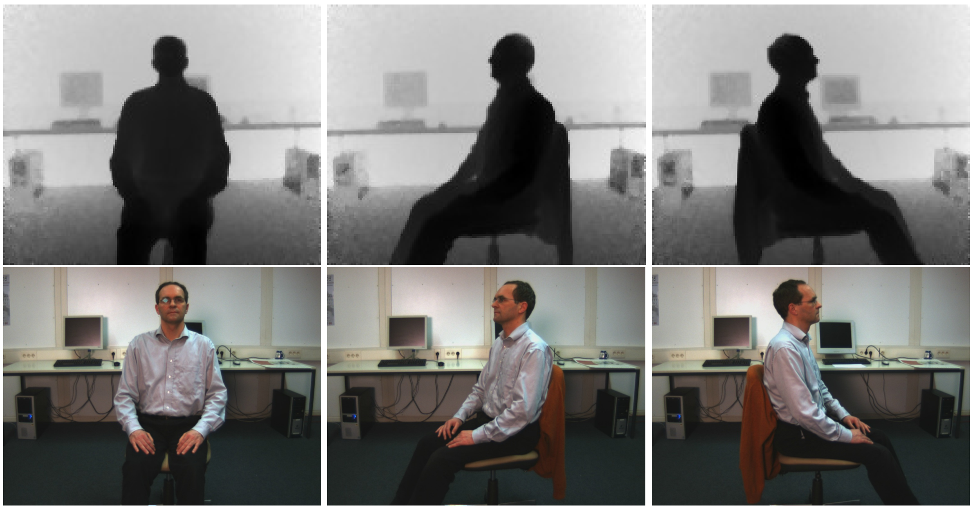


Figure 5.6.: Depth and color input images for the reconstruction of a person. Depth and color images have different resolutions (176×144 pixel and 1024×768 pixel).

5.1.4. Conclusions

The presented approaches show that the ToF-camera is well suited to reconstruct rigid geometry. Two modeling approaches have been discussed and it could be shown that important applications of computer vision could be solved using ToF-cameras. The first is the modeling of indoor environments in which the camera is actively controlled and the measurements are fused in a 2.5D panoramic image which is then transformed to a real 3D model by triangulation. The reached accuracy lies well in the expected range of the accuracy of the ToF-camera. It is bound by the physical operation principle of the ToF-cameras and was found to be within 44-177mm which corresponds to 1.25-2.03% of the room size. The second method is the reconstruction of smaller objects, by either walking around the object and estimating the camera pose or by turning the object in front of the camera. In the second case it is not possible to use a panoramic image to store the values. Instead, a volumetric Octree representation is chosen which inherently fuses multiple measurements of the same physical entities. This shows that ToF-cameras can be used to reconstruct rigid geometry such as complete rooms with good accuracy and that is it possible to capture and reconstruct smaller rigid objects using ToF-cameras and suitable reconstruction algorithms.

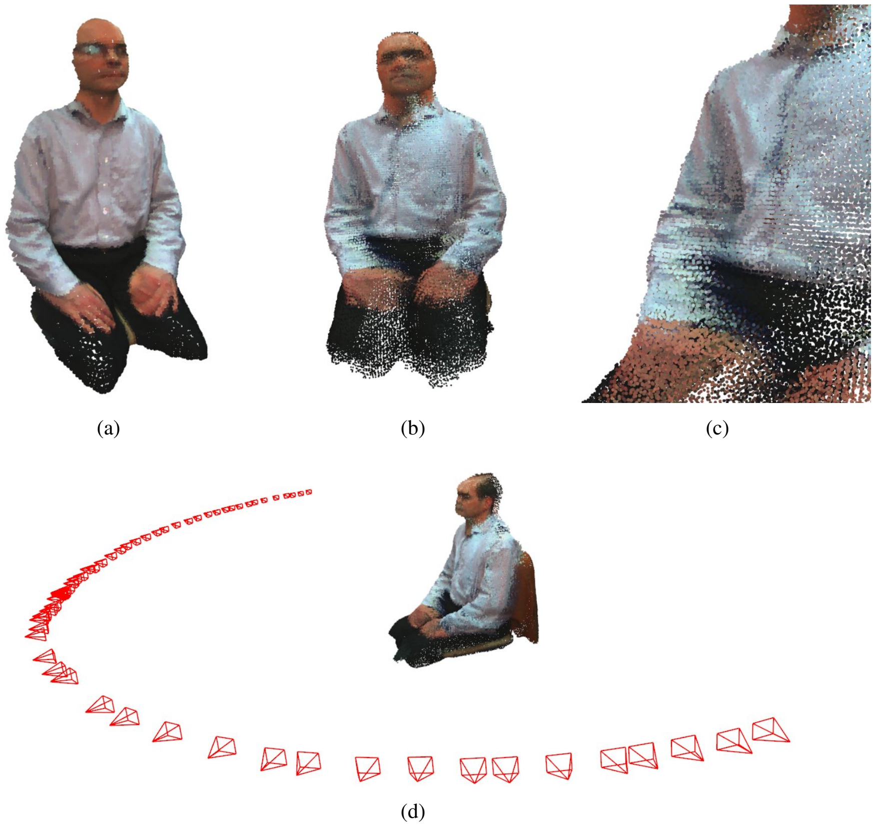


Figure 5.7.: Person model as Octree, fused from multiple image pairs. Octree fused from 5 images (a). Person model fused from 64 image pairs (b)+(c). The miss-registration is mainly due to local motion of the person during recording. Octree cells were rendered as points without splatting. Fused person model with camera poses drawn as pyramids (d).

5.2. Non-Rigid Object Modeling

More challenging and interesting than the rigid object modeling is the non-rigid object modeling. The possibility to capture and reproduce deformations in real-time is an ability which offers advanced opportunities in many areas such as the production of film and media content, movement analysis in health care and human-computer interaction. Due to the real-time capabilities of the Time-of-Flight camera technology, these cameras are dedicated to capture deformations of objects. Hence this chapter discusses the reconstruction of non-rigid and deformable objects with ToF-cameras.

5.2.1. Literature

There are different approaches in Computer Vision which focus on the reconstruction of dynamics and deformation of objects. A research area exists which focuses on the reconstruction of surface deformations and there is a different area which focuses on the reconstruction of movements of articulated objects, such as persons or animals. An articulated object consists of rigid parts, connected by joints which have different Degrees of Freedom (DoF). A typical articulated object is a skeleton model of a person. The objective in the reconstruction of articulated objects is the estimation of limbs lengths and joint states, such as rotation and translation. A good survey in this interesting area can be found in [MHK06]. Gavrila et al. used multiple intensity cameras to reconstruct human body poses in [GD96]. Bray et al. [BKT06] presented an approach (PoseCut) for integrated segmentation and human body pose estimation based on intensity cameras and Markov Random Fields. A comparable approach using voxels and shape from silhouette is used by Wan et al. in [WYM08]. Gall et al. [GSdA$^+$09] recover the skeleton movements and the non-rigid surface deformation of articulated persons and animals.

One of the first to use range images from multiple perspectives were Curless and Levoy in [CL96] but the approach is limited to rigid models. Guan et al. [GFP08] use multiple ToF- and intensity cameras to reconstruct 3D objects. They focus on rigid objects and on the fusion of silhouette and depth information. Pekelny and Gotsman [PG08] use a single depth camera to recover the body pose of an articulated object and reconstruct the surface piecewise. They divide the object into pieces and apply ICP to find the transformations for each limb. Knoop et al. [KSD09] use a ToF- and an intensity camera to track movements of articulated bodies. They use a body template consisting of cylinders and joints and a combination of depth and intensity features to register to the model using ICP.

In contrast to the interpretation of objects as articulated objects, dynamic scenes also contain local object deformation which must be considered. The reproduction of a scene from a different viewpoint is possible using the image-based rendering approaches as in [ESK05] and [MB95] but these lack the possibility to represent non-rigid objects.
An approach which is capable to represent dynamic scenes and to replay the scene from a

different viewpoint is Space-Time-Video, which does not only encode the three-dimensional geometry of the scene but also records the changes in time as 4D representation. The replay of Space-Time-Video should allow to separate 3D space and time, which allows to render the scene from any viewpoint at any time. This requires that for every time step the full 3D geometry is known. Based on this knowledge further data analysis can be executed such as deformation parameter estimation. An approach in which human motion is recorded and actions are represented as Space-Time Shapes can be found in [GBS$^+$05]. There, actors are segmented using silhouette information, and shapes are recorded over time. These shapes are stored in a volumetric representation and classification of motions is applied.

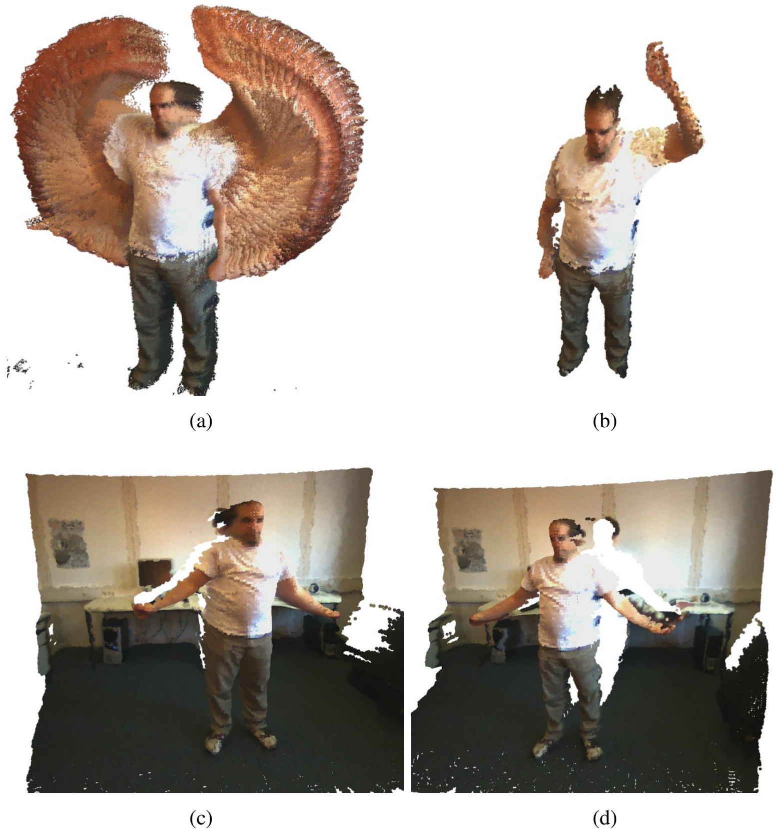
5.2.2. Deformation as Space-Time-Video

As ToF-cameras are capable of recording real-time depth image sequences, these cameras are well suited for recording deformations over time. To record and replay Space-Time Video a suitable way to store and reproduce the volumetric data is needed. It requires to store 3D geometry as well as change over time. The presented Octree data structure (cf. section 4.3.3, p.73) is capable to store 3D volumetric data and the reproduction of this data is possible by rendering it on the GPU. To enable the representation of dynamic scenes the Octree datastructure is extended to store in every cell the time at which it is visible and the corresponding color. If only foreground objects are to be considered, the reconstruction requires a reliable segmentation of the deformable object which is discussed in section 4.2, p.57. Using only one 2D/3D-camera and using foreground segmentation also means that the geometry in the back of the person is incomplete. This restricts the camera movements, which however can be solved by using additional ToF-cameras that observe the scene from different viewpoints.

When adding a new image to the Octree datastructure, the current image number is used as a time-code. If the volume element is already occupied, the geometric information is calculated as an average of old and new measurement and the timestamps at which the element is visible are updated.

As a volumetric representation of the scene is at hand, the camera can freely move around and render the Octree by selecting only those cells which were visible at a given time stamp. Figure 5.8 shows examples of such a Space-Time-Video sequence, including camera movement. A person is standing in front of the camera head, swinging the arms. The left image shows all Octree elements which have been filled during the animation sequence which consists of 375 images. The second image shows a selected frame of the sequence. Image three and four show one moment in time, recorded by the ToF-Camera and rendered from different views. By including the object in the environment Octree model, the background can be filled as well.

Figure 5.9 shows the color-coded accumulated hit-counts for the above sequence. The number of times an Octree volume element is hit up to the current time frame is saved and analyzed. This provides information about which parts of the scene are static and which are dynamic.



(a) (b)

(c) (d)

Figure 5.8.: Space-Time Video: All elements in the Octree (a), selected time instance of the animation sequence (b), rendered views with background (c)+(d).

In the above sequence, the arms of the person are moving and therefore distinguished from the body. 4D Space-Time representation also allows to combine 3D information which has been taken at different points in time. This is the selection of different time slices and the combination in one 3D shape. An example is shown on the right of figure 5.9 in which three points in time have been selected, forming a person with six arms.

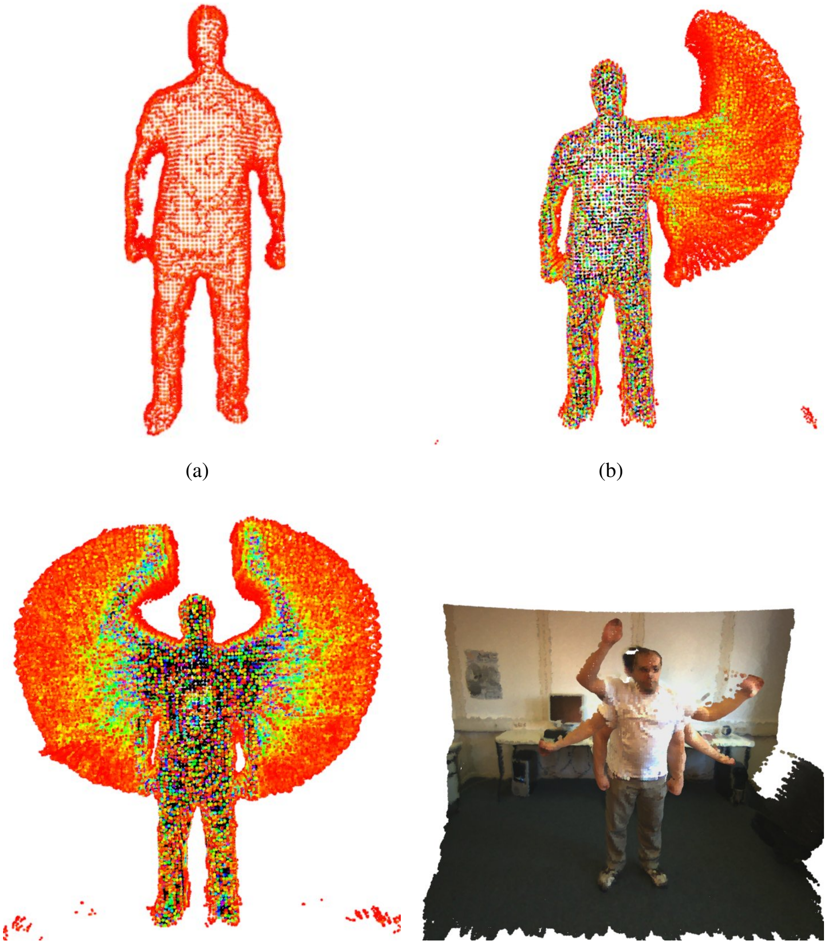


Figure 5.9.: Left: Accumulated hit-count for video sequence over time, color coded: red (< 60) - black (>= 360). Frame 1 (a), integration of frame 1 to 120 (b), integration of frame 1 to 375 (c). Rendering by selection of 3 time frames of the Space-Time Video (d).

5.2.3. Performance Description

Capturing and storing dynamic three-dimensional scenes at a high resolution and framerate produces a large amount of data which has to be stored and eventually retrieved and displayed. As mentioned in the previous section the nodes of the Octree datastructure have been extended for this purpose by additional elements for visibility at a certain time-stamp and corresponding color information. To store the change of a scene over time, the data in the Octree doubles if the whole scene changes, compared to a static representation of the scene, with every additional frame. This results in a very high amount of data which will easily excess the memory capacity of current computers. For example the Octree of the environment model in section 5.1.2, p.81 with 25mm cell side length consumes 735.26 MB of data as shown in table 5.2, p.84. Approximately half of the memory is occupied by the data in the Octree and half by the Octree itself. Every additional frame in which every point of the Octree changes would therefore add approximately 370 MB to the Octree. After a couple of frames the entire memory is occupied.

To avoid this, new geometric and intensity information is only stored if the change exceeds a predefined threshold. Otherwise already contained geometric and intensity information is linked to the current time stamp. Rendering times also increase with increasing sequence length as for every frame the nodes to display and the color information (and eventually additional information) has to be determined by searching the timestamps of every node. Alternatively volumetric datastructures can be implemented on the GPU which allows faster rendering and e.g. real-time editing as in [KCK09].

5.2.4. Conclusions

In this section I have presented an approach to capture, fuse and reconstruct non-rigid geometry in real-time using a ToF-camera coupled with a CCD-color camera. In this approach it has been shown how 2D/3D- camera can be used to capture a persons geometry while the person is moving in the scene. Furthermore a space-time video approach has been developed to encode movements of the person in an Octree. The resulting person model is of high spatial quality, but covers only the front half of the person.

"The Guide is definitive. Reality is frequently inaccurate."

Douglas Adams (1952 - 2001)

6 Application in Mixed Reality System

This chapter shows an application of the methods developed in this thesis in a real-time Mixed Reality system. The system and the results have been published in these publications: [SBKK10] [SBKK08] [BSBK08] [KSB^{+}09]. Key parts of the system have been developed in this thesis and it uses most of the methods and approaches presented in this work. It uses the presented calibration, preprocessing, environment reconstruction and segmentation approaches.

6.1. System Overview

This section gives an overview of the whole Mixed Reality system. A cheap and flexible usage of the system is desired. Hence only images taken by standard- and ToF- cameras and no chroma-keying facilities are used. It uses three cameras which are shown in figure 6.3, a ToF-camera combined with a CCD-camera, both rigidly coupled and mounted on a Pan-Tilt unit and a spherical CCD-camera. The ToF-camera delivers depth images, but from a different viewpoint and with different intrinsic parameters than the perspective CCD-camera. The perspective CCD-camera is also denoted target camera because the final composed image is generated for this camera's perspective and resolution. The input images of the three cameras are depicted in figure 6.1 in which the fisheye images are shown on the left, the target camera images are shown in the middle and the ToF-depth images are shown on the right.

Figure 6.2 gives an overview of the system. As a first step it requires that the cameras are internally and externally calibrated. Therefore the calibration from section 3.2 was extended to handle spherical cameras, parametrized with the model developed by Scaramuzza et al. [SMS06]. As target camera and ToF-camera do not share the same center of projection and have different perspective camera parameters a warping of the depth measurements into the target view has to be performed. This is more closely explained in section 4.1.2.

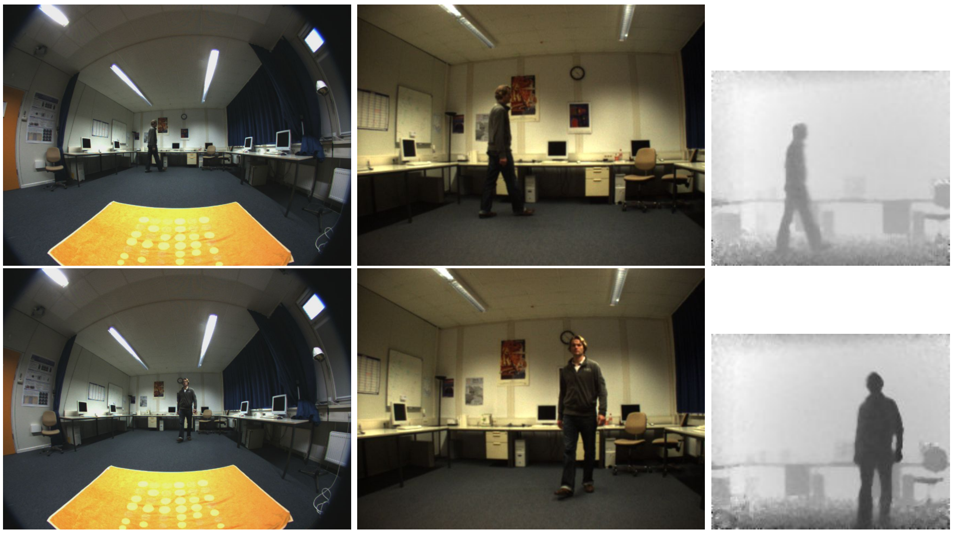


Figure 6.1.: Input images of fisheye-, target- and ToF-camera. Note that the ToF-depth images have a much lower resolution than the other images.

The system requires methods to estimate the camera movement and to replace the chroma-keying facilities. These requirements can be met by using a model of the environment and hence a background model is generated as described in section 5.1.2. A segmentation method is needed which is capable to handle moving camera viewpoints. Hence the GPU-based segmentation with background model (cf. section 4.2.2) is utilized. The GPU is also used for the correct combination of real and virtual content. This is presented in section 6.3. For a realistic appearance of the composed scene, shadowing of real and virtual objects is additionally computed and added to the final mixing result (see section 6.4).

A correct rendering and composing is only possible if the current target view camera pose is known in the coordinate system of the virtual content. To calculate the current pose of the CCD- target camera the spherical CCD-camera is used as a pose sensor. The analysis-by-synthesis background model based tracking approach, discussed in the next section 6.2, is used to track the spherical camera pose. The orientation and translation between fisheye- and target camera is known from calibration.

Once the background model has been created, the alignment of virtual content can be performed by means of 3D modeling tools. The trajectory of a moving object in the scene can be determined from the segmentation which is computed on the GPU. The received trajectory can now be used to assist in the placement of virtual objects in the scene. The complete system and the working flow in the system is shown in figure 6.2.

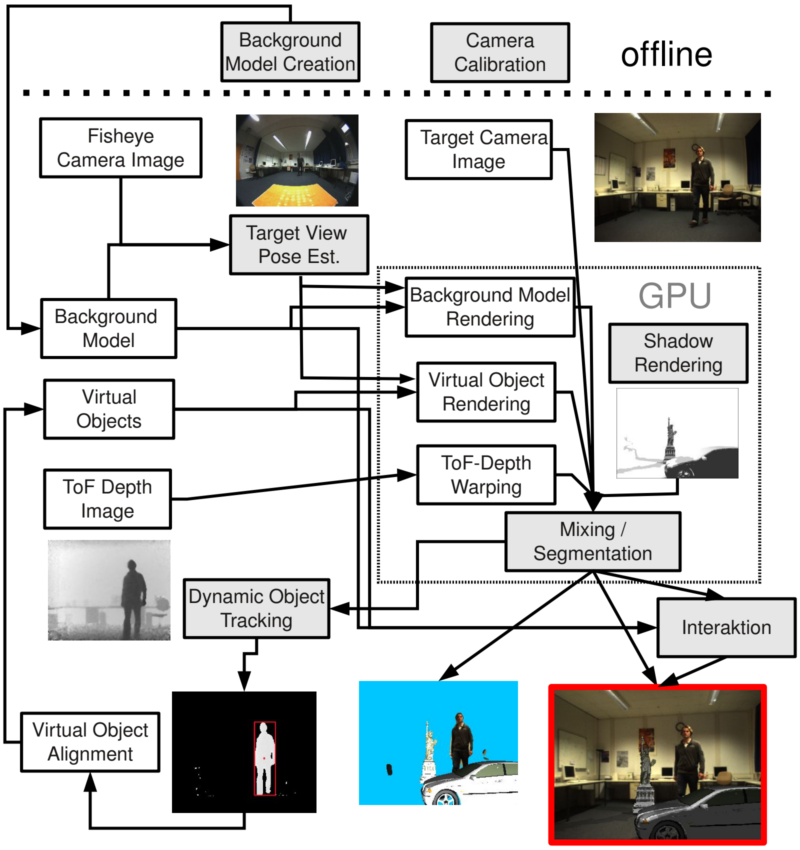


Figure 6.2.: Mixed Reality System components and interactions. The boxes represent algorithms and components of the system. The light gray boxes represent algorithms which have been developed in this thesis. The finally composed image is marked with a red border.

6.2. Pose Estimation with Environment Model

Many studio installations use rigid camera installations or cameras controlled by robots to obtain information about the current position and orientation. This is however very inflexible and expensive. Hence an automatic computation of the current position and rotation of the camera constitutes a huge simplification and allows to reduce the costs of media productions.

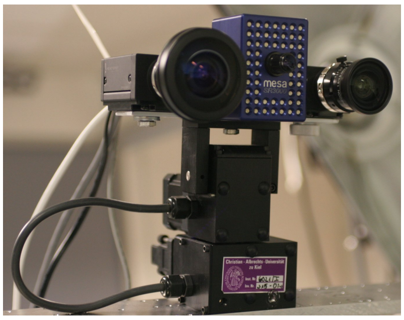


Figure 6.3.: The setup for pose estimation with environment model. Fisheye CCD-camera, ToF-camera and perspective CCD-camera, mounted on a pan-tilt unit.

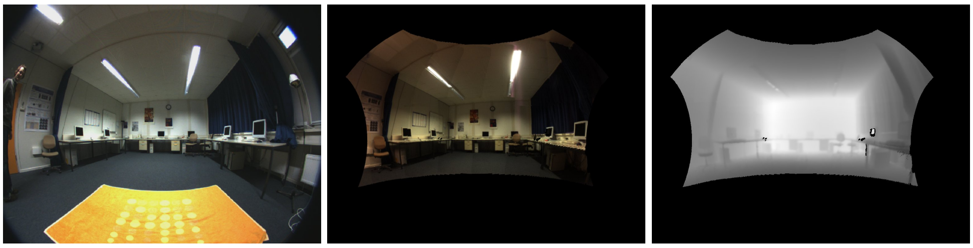
One possibility is to use the pose estimation with CCD- and ToF- depth images as discussed in section 5.1.3. This raises several problems, originating from the small opening angle and the noise in the ToF-images if operated at high frame rates. These problems can be partially overcome by the solutions presented in this section. The idea is to use a environment model, generated with the approach from section 5.1.2, as an absolute reference and to estimate the current camera pose with an analysis-by-synthesis approach relative to this environment model.

The pose estimation follows and extends the analysis-by-synthesis approach presented in [KBK07]. The accuracy of the pose estimation is directly dependent on the accuracy of the environment model. In section 5.1.2 an evaluation of the accuracy of the environment reconstruction is given. The average error of the reconstruction was found to be $1.25 - 2.03\%$ or $44 - 177$mm for an environment of size $\approx 3 \times 5.5 \times 8.5$m, typical for the usage of the system (cf. table 5.3). The error of the pose estimation is expected to be in the same order of magnitude.

As large parts of the scene can be occluded by objects or actors during the recording process a robust pose estimation method has to be used which can handle occlusions. Hence a fisheye camera with a spherical lens and an opening angle of ≈ 180 degrees is used as a pose sensor in the proposed setting. The pose estimation is carried out using the images of this fisheye camera together with the background model. The idea of the pose estimation is to track the current camera relative to the background model of the environment. Current camera position and model orientation can be very different in the beginning. Therefore the current camera position has to be registered relative to the model using robust feature matching. After this registration the camera motion is assumed to be small. Therefore less robust and less computationally expensive features are used to establish correspondences between model- and current fisheye image.

6.2.1. Registration

The pose estimation starts with the registration of the current fisheye image to the generated background model. Assuming that the camera is close to the position from which the model was generated, the background model is rendered with the intrinsic parameters of the fisheye camera and the extrinsic parameters (rotation and translation) with which the model was generated. Original fisheye image and rendered intensity and depth images are depicted in figure 6.4.



(a) (b) (c)

Figure 6.4.: Image of fisheye camera (a) and view of the environment model of intensity (b) and depth (c) rendered with fisheye camera parameters.

Due to the possible large displacement between current camera position and rendered model pose, features which are robust to scale change and rotation have to be used. Hence the gradient orientation based scale invariant SIFT-features (cf. [Low04]) are detected in the rendered intensity image 6.4 (b) and 3D points are generated for these features using the depth information from the corresponding rendered depth image 6.4 (c). These SIFT-features are matched against features extracted in the current fisheye camera's image 6.4 (a).

The camera pose is estimated on these 2D/3D correspondences using the standard DLT algorithm ([HZ04] p.73) in combination with RANSAC [FB81]. Outliers are removed after an initial pose estimation by deleting correspondences which do not comply with the estimated pose and the pose estimation is repeated on the remaining inliers. A feature does not comply with the estimated pose if the distance between the 2D SIFT-feature and the projection of the 3D point to the image plane with the estimated camera pose exceeds a threshold, which is typically set to 1 pixel. The detected SIFT-features in both images are visualized in figure 6.5 together with the correspondences. After this registration the background model is aligned to the current image.

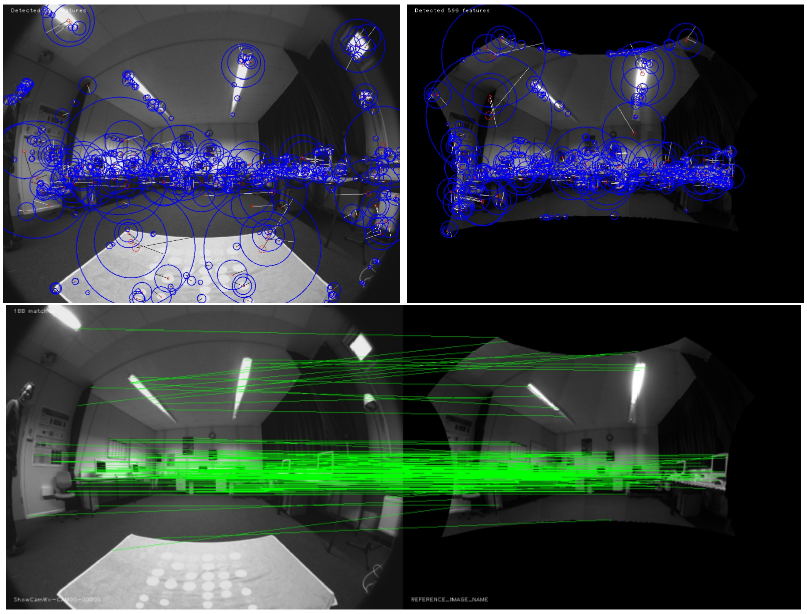


Figure 6.5.: Visualization of the detected SIFT-Features on input and rendered model image and the computed correspondences visualized as lines between corresponding features.

6.2.2. Camera Pose Tracking

After registration the camera motion is assumed to be small between subsequent images. Therefore the pose estimation can be based on tracking points of interest, in this case KLT-features [ST94], from the model's image to the current fisheye image. To start the camera pose tracking initial KLT-features are detected on the current fisheye image. As fisheye camera and virtual camera of the rendered model are identical after initial registration, 3D points are constructed for each 2D KLT-feature by depth look-up in the rendered depth image of the background model.

In the consecutive tracking steps the detected 2D KLT-features are tracked between the rendered model's intensity image and the current fisheye image. To support the feature tracking the pose the background model is rendered with is predicted to the current frame. A linear prediction of the pose using up to three previous poses is used for this purpose.

To detect occlusions, the current depth measurements of the ToF-camera and the established 3D points are checked against the corresponding depth values in the rendered model's depth image. If the difference exceeds a threshold, this point is considered an outlier and is not taken into account for pose estimation. The remaining points which are considered inlier are marked in green in figure 6.6 on the right. Additionally a feature's validity is checked by a robust photometric measure in order to detect features which are occluded by a dynamic object before it contributes to the estimation. The used photometric measure is called the X84 rule which was introduced by Fusiello et al. in [FTTR99]. This rule compares the distribution of the image values in the area of the feature in the current frame with the distribution of image values of the same feature in the previous frame. It enforces that features with distributions, which differ more than k Median Absolute Deviations (MAD) from the median, are rejected (k is set to 5.2).

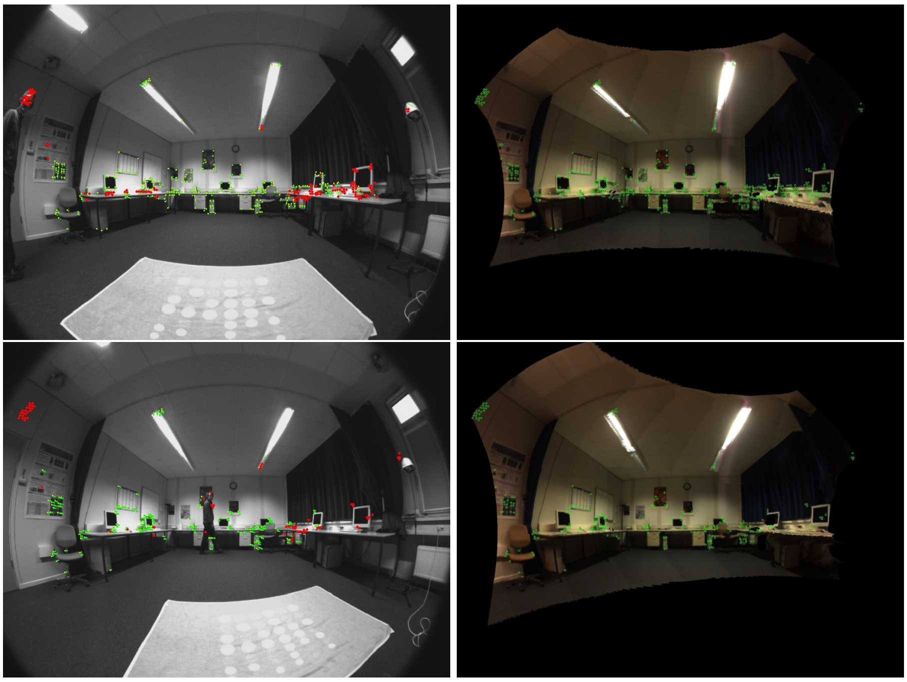


Figure 6.6.: Tracked KLT-Features on input and rendered model image for the first image (top) and after 150 images (bottom).

Points for which the tracking between model- and fisheye image was not successful are marked red in the left images of figure 6.6. Based on the established 2D/3D correspondences the current camera pose is estimated for each image using the same combination of DLT and RANSAC as in the initial registration phase described in section 6.2.1. The pose estimation minimizes the differences between the 2D points in the image and the projection of the corresponding 3D points into this image using the estimated camera pose. This projection error between 2D and 3D points is depicted in figure 6.7 for the initial pose estimation and after 150 images.

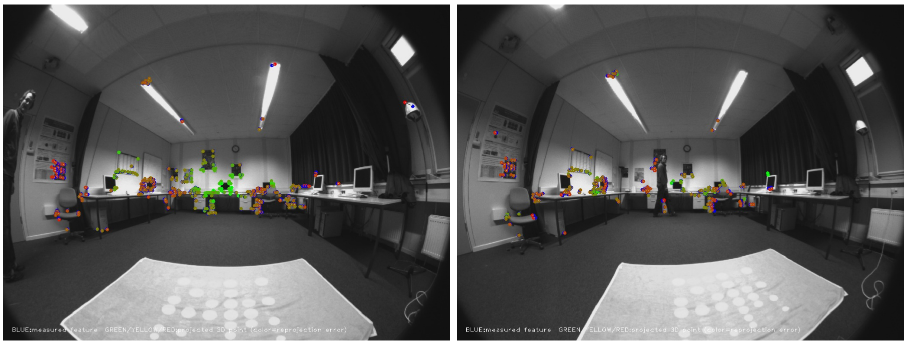

BLUE:measured feature GREEN/YELLOW/RED:projected 3D point (color=reprojection error)
BLUE:measured feature GREEN/YELLOW/RED:projected 3D point (color=reprojection error)

Figure 6.7.: Projection error of pose estimation with 2D/3D correspondences. Green describes a small reprojection error, increasing over blue to red which indicates a high projection error.

The fisheye camera's extended FoV always provides sufficient visible features for reliable tracking, even if large parts of the used background model are occluded. Using the background model as reference the known problem of error accumulation (drift), which is for example encountered in the solely depth image based pose estimation in section 5.1.3, is avoided.

6.3. Combination of Real and Virtual Content

For the combination of real and virtual content, the depth images of the ToF-camera, the background model and the virtual objects are used. For all the computations on images during the composition of the final images GPU shaders are used, which allows to perform all computations in real-time. The principle of the final composition is as follows. At every pixel in the image the decision which color the final image is assigned, is based on the corresponding depth values in the depth images shown in figure 6.8. In all these depth images, the current depth image, the background depth image and the virtual depth image, the smallest distance

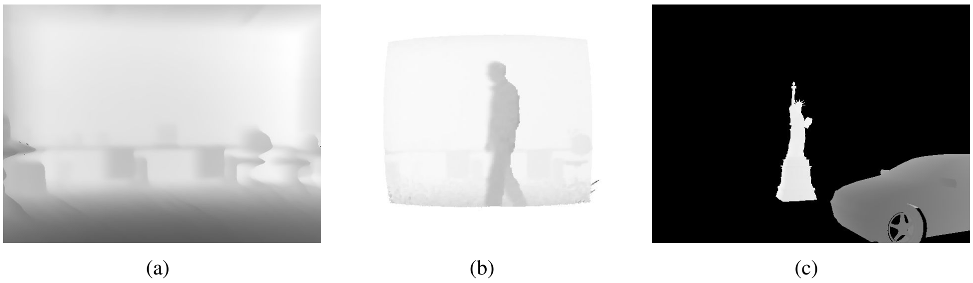


Figure 6.8.: Depth images for composition with correct occlusion handling. (a) Background depth image, (b) warped current ToF-depth image, (c) virtual depth image.

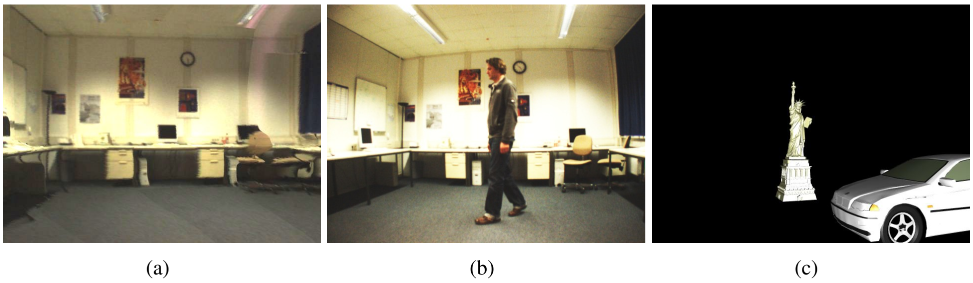


Figure 6.9.: Color images for composition with correct occlusion handling. (a) Background color image, (b) real color image, (c) virtual color image.

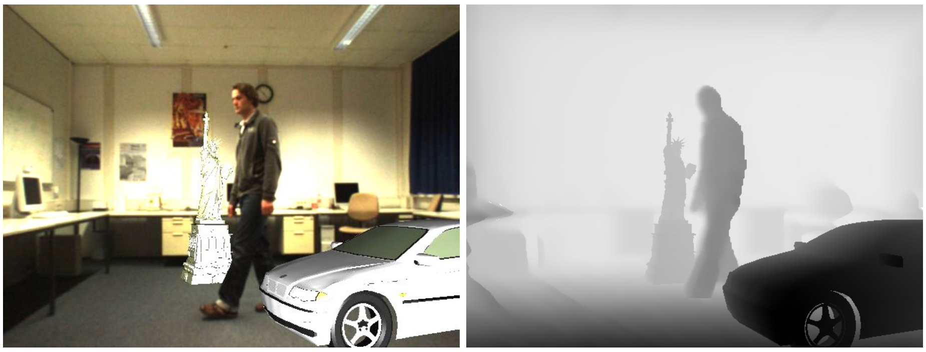
to the camera is searched. The color value of the corresponding color image is selected and assigned to the finally composed image.

Let $d_b(\boldsymbol{x})$ be the depth in the rendered background depth image 6.8 (a) at pixel $\boldsymbol{x} = (x, y)$, $d_c(\boldsymbol{x})$ the depth in the current warped ToF- depth image (b), $d_v(\boldsymbol{x})$ the depth in the rendered depth image (c) and $d_f(\boldsymbol{x})$ the finally composed depth in image 6.10 (b). Let $c_b(\boldsymbol{x})$ be the color in the background color image 6.9 (a), $c_c(\boldsymbol{x})$ the color in the current color image (b), $c_v(\boldsymbol{x})$ the color in the rendered virtual image (c) and $c_f(\boldsymbol{x})$ the final composed color in image 6.10 (a). How the final composed image is calculated is shown in the following algorithm 1:

The achieved mixing quality and how real the perception of the mixed images is, is directly dependent on the segmentation accuracy. This accuracy has been evaluated in section 4.2.5 in which it could be shown that the number of wrongly classified pixel is below 2% for the depth thresholding and if the proposed approach with MoG and weighted depth clues is used it is

Algorithm 1 : Algorithm for mixing real and virtual content based on depth.

```
for $\boldsymbol{x} = 0$ to $nrOfPixel$ do
  $c_f(\boldsymbol{x}) = c_b(\boldsymbol{x})$ {initialize final color with background color}
  $d_f(\boldsymbol{x}) = d_b(\boldsymbol{x})$ {initialize final depth with background depth}
  if $d_c(\boldsymbol{x}) < d_f(\boldsymbol{x})$ then
    $d_f(\boldsymbol{x}) = d_c(\boldsymbol{x})$
    $c_f(\boldsymbol{x}) = c_c(\boldsymbol{x})$ {if current depth is smaller than background set to current}
  end if
  if $d_v(\boldsymbol{x}) < d_f(\boldsymbol{x})$ then
    $d_f(\boldsymbol{x}) = d_v(\boldsymbol{x})$
    $c_f(\boldsymbol{x}) = c_v(\boldsymbol{x})$ {if rendered depth is smaller than current set to virtual}
  end if
end for
```



(a)

(b)

Figure 6.10.: Color- (a) and depth (b) image, composed of original images and virtual objects by depth mixing.

below 1% of the number of image pixel (cf. tables 4.1 and 4.2).

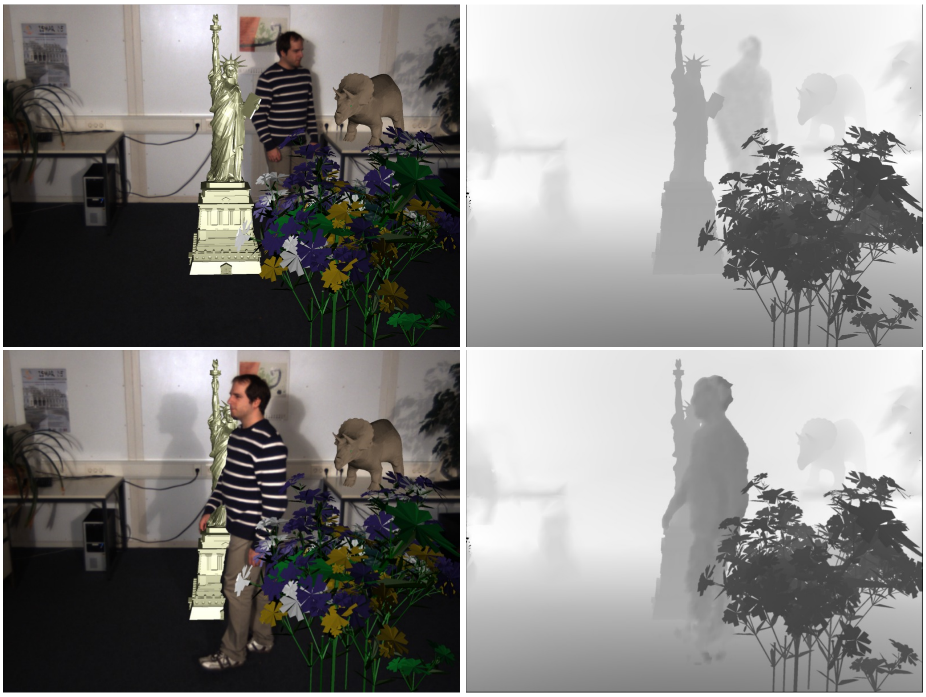


Figure 6.11.: Original image of a person, augmented by virtual objects with mutual occlusion (left). Corresponding mixed depth images (right).

Figure 6.11 shows another result of the depth based mixing approach in which the real image is enhanced by virtual objects of a statue, a dinosaur and a plant in the front. Due to the quality of the segmentation of the person a realistic perception of mutual occlusion is achieved. The real person and the real plant in the right corner cast shadows in the back wall. The virtual objects do not cast shadows on the wall which disturbs the perception of the scene. How this is solved by adding shadows as discussed in the next section.

6.4. Shadow Rendering

The correct and stable placement of the synthetic content in the images, which is guaranteed by the previously described camera pose tracking, is the first part to ensure the quality of the finally composed image. The other component that significantly increases the quality of the augmentation is a correct shadow casting of the virtual objects. The images in figure 6.11 show an augmentation without shadow calculation for the virtual objects for example.

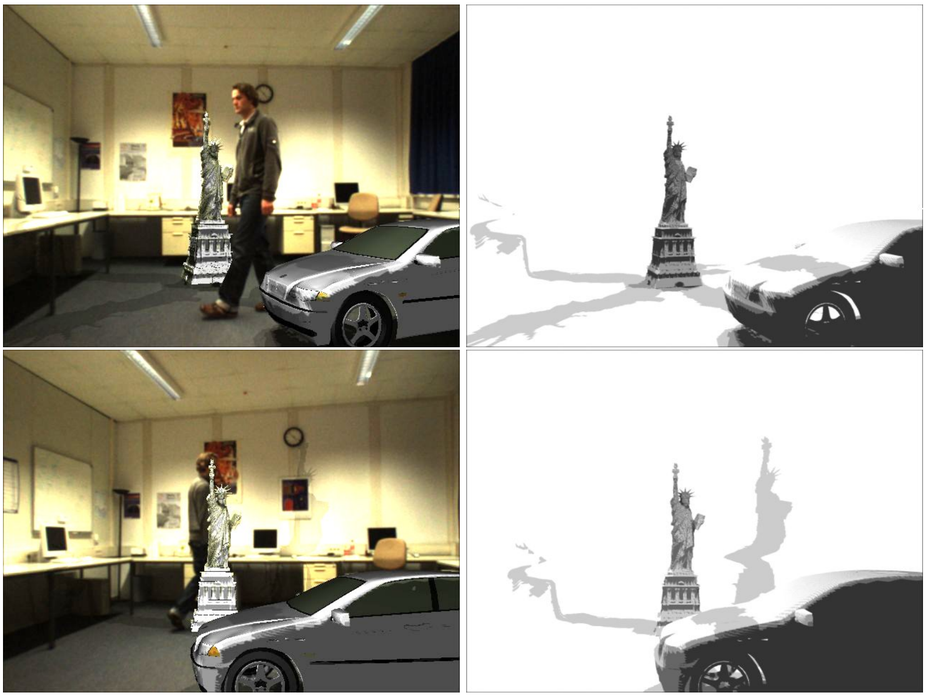


Figure 6.12.: Final composition with shadow mapping. Left: Images with shadows computed from light maps. Right: Light maps computed from light sources at the ceiling (top) and with alternative light source positions (bottom).

If the virtual objects are correctly aligned to the floor, the viewer should always have this impression even if the camera is moving. Sometimes however the virtual objects seem to hover slightly over ground. This impression is resolved by adding shadows, as shown in the lower images of figure 6.12. In order to add shadows of virtual objects to the real images, so called light maps are calculated for each video frame. These maps in principle encode how much light is reaching the part of a scene pictured by a particular pixel if virtual content is present. Each pixel in the light map contains a factor $0 \leq s \leq 1$, which is used to scale the RGB color values in the augmented image. A scale factor of 1 corresponds to no shadowing, 0 renders a pixel black and values in between model partial shadowing.

The light maps are generated using the well-known shadow mapping technique [Wil78]. Therefore a depth map is rendered for each light source containing all virtual objects that should throw shadows. Additionally the background model and all virtual objects are rendered from the target camera's point of view, texturing the scene with the calculated depth maps of the corresponding lights using projective texturing. This way for each pixel in the target image the distance values R encoded in the depth maps of the light sources can be compared to the distances D between a light source and the 3D point corresponding to the pixel.

As the light's depth map provide the distance between the light source and the first intersection of the light ray with the scene geometry, it can be decided whether the pixel is in shadow ($R < D$) or receives light from the light source ($D = R$). Evaluating all light sources and adding an ambient light offset leads to the target view dependent light map used for shadow generation as shown in figure 6.12.

This algorithm automatically adapts to different scene geometries, which allows to take full advantage of the background model's geometric information for shadow rendering. This is demonstrated in the images in figure 6.12. Observe how consistent results are achieved by not only casting shadows on the floor and the side walls but also on tables, taking the given background geometry into account. Moreover, the background model can aid in defining the appropriate positions and orientations of light sources in the scene, because the real light sources are visible in its texture. The result presented in the lower row on the left of figure 6.12 was generated using four point-lights, that were positioned on the real light sources according to the light sources in the background model. This is of course not sufficient to simulate the reality but already increases the augmentation's perceived quality. However, spending more resources for rendering more light sources in conjunction with light map smoothing will already increase the realism without much alteration of the proposed processing scheme.

Figure 6.13 shows another example for the increased augmentation perception if compared to the images in figure 6.11. Two real and also two artificial light sources have been used to allow consistent shadow casting of the virtual objects. The two real light sources have been placed next to the camera to generate shadows on the wall in the back. Figure 6.11 shows images of the sequence with natural shadows, only created by the real light sources. To copy this natural shadow casting, two artificial point light sources have been placed virtually in the scene for the shadow computation. Note how the virtual objects cast shadows on the real person and the background in 6.13. The described shadow casting technique does not use any antialiasing technique for the light maps which results in course shadows. To improve the effect and generate a more realistic perception with soft shadow edges, the light maps are filtered using a Gaussian filter.

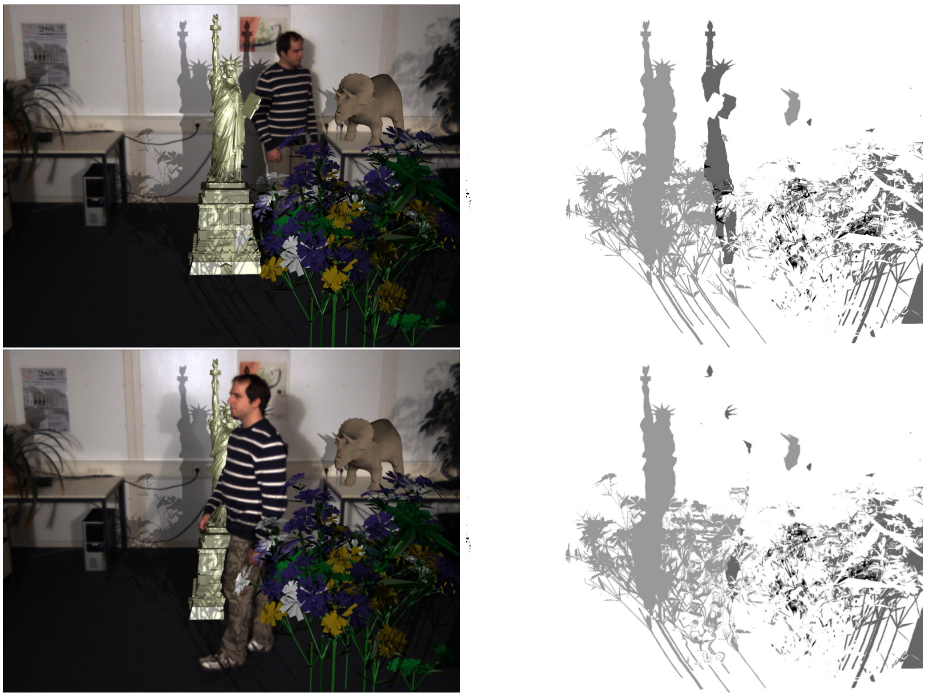


Figure 6.13.: Final composition with shadow mapping. Mixed images with added shadows (left) and light maps for two light sources (right).

6.5. Tracking Segmented Objects

For many purposes it is helpful to analyze the movements of dynamic objects in the scene. In the presented example (figure 6.14) a person appears in the rigid scene, and is segmented using depth-keying with background model as described in section 4.2.2. To detect individual moving objects in the images a simple and fast clustering algorithm, BFS (Breadth-First Search) is used. It is known as a graph search algorithm which can easily be applied to images.

Starting with a single pixel which belongs to the segmented object, all neighboring pixel which belong to the foreground are added to the current object. For the sake of robustness to small segmentation errors a minimum desired object size is defined. For each detected object a center-of-mass is calculated by averaging its pixel coordinates. The result of this tracking can be seen in figure 6.14 in which only one object is present. Projecting the detected pixel coordinate of the center-of-mass with the depth measurement from the ToF-camera to a 3D point results in the 3D point of the center-of-mass of the object.

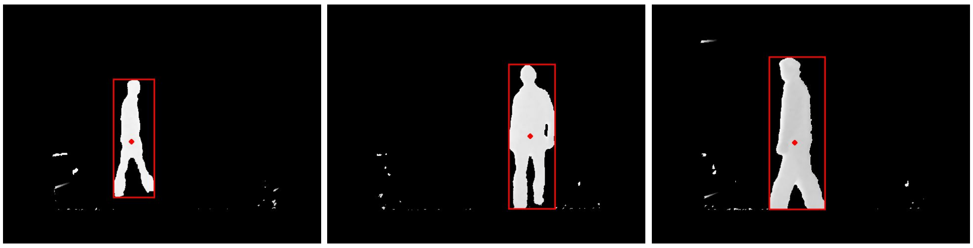


Figure 6.14.: Tracking of dynamic content. The moving person is detected and marked with a rectangle, the center-of-mass is marked with a dot.

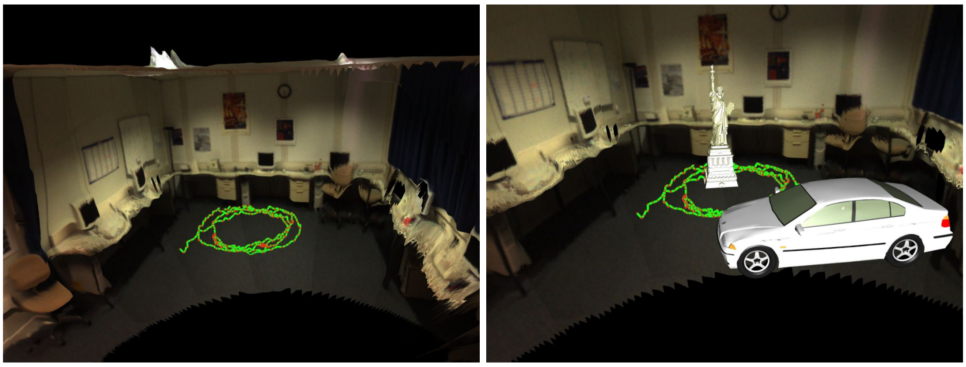
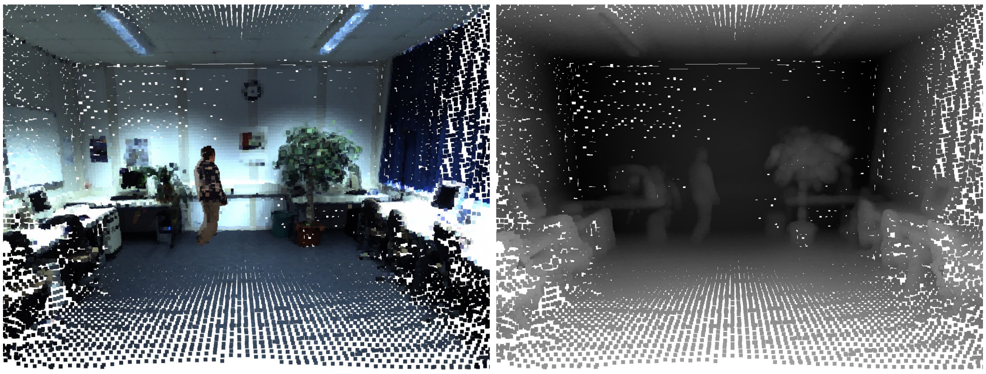


Figure 6.15.: Background model with detected trajectory (projected to the ground plane) of the moving person and aligned virtual models.

Projecting the detected 3D point of the center-of-mass to the ground plane of the background model directly yields the trajectory of the moving person on the floor, as shown in figure 6.15 at the top. This information can be used for a variety of applications. For example in live processing to place the models in the scene. It can additionally be used in an offline step to plan the placement of the virtual models. In the exemplary results shown in figure 6.15 the person entered the model from the left and walked round the room two and a half times, turned around and walked in the opposite direction. Note that this tracking of moving scene content is also used to simulate interaction, e.g. collisions between real and virtual content as described in the next section.



(a)

(b)

Figure 6.16.: Octree of the environment. Shown are the colored (a) and the depth-coded (b) voxels.

6.6. Collision Detection and Geometric Interaction

Since the full 3D geometry of all objects is available, it is possible to compute geometric collisions between the objects, especially the geometric interaction between virtual elements, the real background model and the dynamic foreground objects. This allows a realistic interaction with the environment and is basis for interactive productions and interactive contact free games. There are two possibilities to compute collisions and the physics of interaction, based on two different representations of the scene. The first is to represent the scene entirely as an Octree and compute collisions between Octree cells, the second is to represent all content as triangles and compute collisions between bounding collision shapes approximated on the triangles of the objects. The two possibilities are discussed in the following two sections.

6.6.1. Collision Detection Using Octree

For collision detection and geometric interaction using Octrees, the environment model is converted into an Octree representation that is adapted to the measurement uncertainty. A volume of $8 \times 8 \times 8m^3$ is processed with a minimum voxel size of $(25mm)^3$ bounding box for collision detection. While the environment model is converted only once, the dynamic person model is updated in each frame to reflect the object motion. Figure 6.16 shows the octree volume representation for one of the frames, including the person's model. As an example for interaction with virtual content, colored balls are dropped into the scene while the real person is walking around.

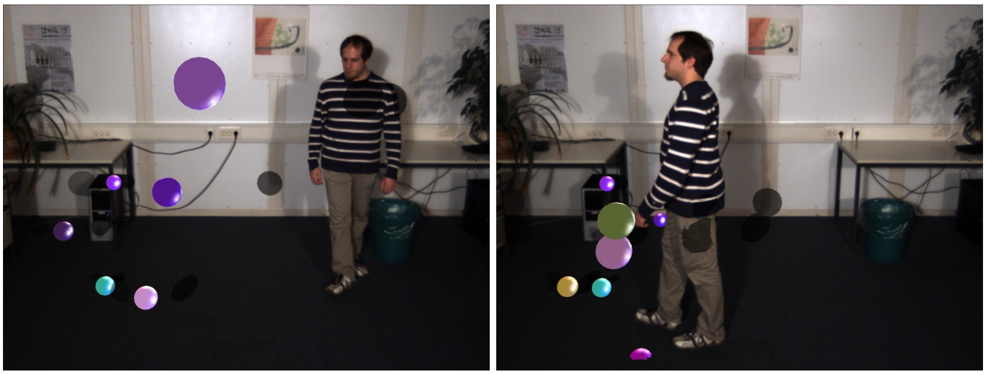


Figure 6.17.: Two frames of the animation sequence using Octrees for collision detection.

The background model is assigned a very large mass and zero velocity while the tracked person is assigned a large mass and a velocity from the tracking as described in the previous section 6.5. The balls have a lower assigned weight and an initial velocity. Collision is now computed between the bounding boxes of the Octree cells of background and person and the bounding box of every virtual object. The virtual balls are colliding and bounce off the real scene objects. Since the camera observes the frontal object surface only, some collisions at the person's back side are missed, but that does not really harm the visual effect. Figure 6.17 shows two frames of the resulting animation sequence. The balls are reflected and even stirred up by the legs of the walking person.

A far more substantial problem is that bounding boxes of the Octree cells as well as the bounding boxes of the objects do not approximate the real surface of the objects close enough. To achieve realistic collisions the normals of the objects have to be considered to compute the angle in which objects bounce off each other. So an approximation of the surface at the colliding surface points would have to be computed. This is neglected and to compute the angle of incidence the trajectory of the virtual objects is used and the reflection is computed by setting the angle of incidence equal to the angle of reflection.

6.6.2. Collision Detection Using Triangles

Collision detection using the Octree datastructure has a significant drawback as discussed above. This drawback can be overcome by using the convex triangle hull of objects as collision shapes. For this purpose a state of the art collision detection engine can be used without further modification to the algorithms by creating sets of triangles of the background once and for the moving objects in every new frame. For the collision detection purpose the Open Source physics engine "Bullet" [Bul10] is used. Figure 6.18 demonstrates the collision of

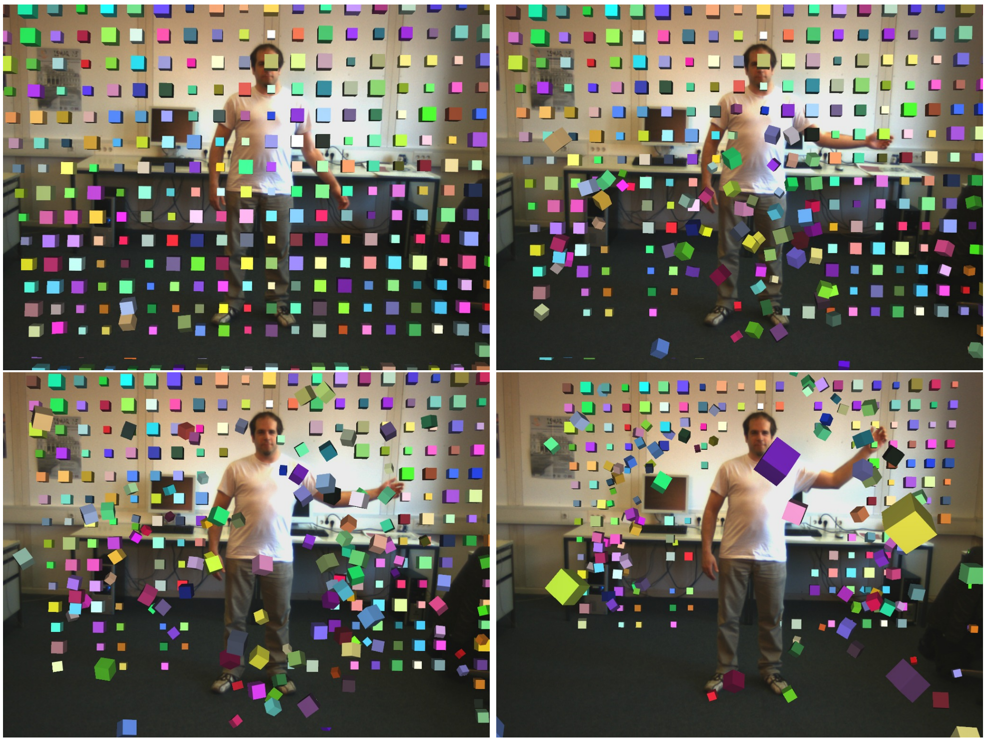


Figure 6.18.: Animation sequence using triangles and Bullet for collision detection.

small colored boxes with the person in the foreground. The boxes have their gravity vector oriented to the backside wall for demonstration purpose. They fly towards the person and bounce of the collision shape of the segmented person. This principle can also be used to design interactive games in this Mixed Reality system. Figure 6.19 shows such a game in which the player has to collect items (the cylinders) that approach him from the front to earn points. The player has to avoid other items (the morning stars) which decrease his score when they hit the player.

6.7. Conclusion

This chapter fused most of the presented methods shown in the preceding parts of the thesis. It utilizes the calibration, extended by the spherical camera model, to calibrate a camera setup consisting of a fisheye- a CCD- and a ToF-camera mounted on a pan-tilt unit. This setup was used to construct a full 3D model of the environment and used model-based camera pose estimation with the fisheye camera to estimate the movement of the camera. The current ToF-depth measurements are used, together with the rendered background model to segment dynamic objects in the images which is used in the augmentation state to combine virtual and real content under correct mutual occlusion on the GPU. The background model and the geometry of dynamic objects was additionally used to calculate shadows which cast on these objects to improve the visual perception. The last contribution, the contact-less interaction with virtual objects, allows to generate interactive games and new applications for visual media productions.

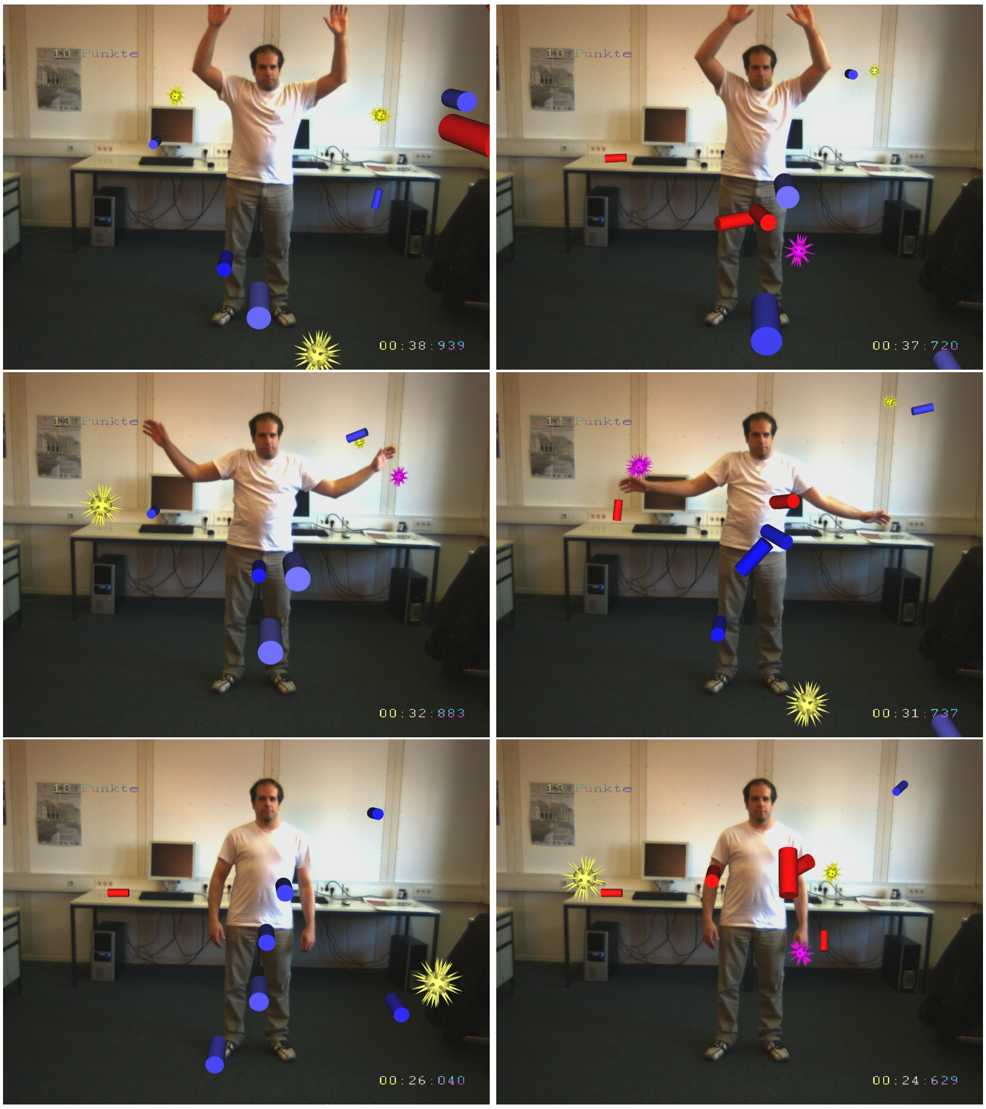


Figure 6.19.: Screen-shots of a game developed in the Mixed Reality system.

"No amount of experimentation can ever prove me right; a single experiment can prove me wrong."

Albert Einstein (1879 -1955)

7 Summary and Outlook

The topic of this thesis is the three-dimensional analysis of static and dynamic scenes using a combination of ToF- and conventional CCD- cameras. The analysis of dynamic scenes in 3D and the reconstruction of dynamic scenes or objects is one of the most challenging tasks in Computer Vision. Many approaches for object reconstruction and dynamic scene modeling exist as it is a widely investigated topic. Reconstruction of rigid- as well as deformable geometry requires dense depth information about the scene or the object which is necessary prerequisite for the reconstruction algorithm. If a real-time scene analysis is required the depth data also has to be available in real-time. Many conventional methods, such as stereo approaches, fail to provide dense depth maps, or fail on dynamic scenes such as triangulation laser scanners. Recently the Time-of-Flight cameras, based on the Photonic Mixer Device sensors, have reached series-production readiness and deliver reliable dense depth information of complex scenes in real-time which makes them suitable for the analysis of deformable three-dimensional scenes. This thesis investigates the usability of ToF-cameras for 3D scene analysis.

The first part of this work studies the calibration of ToF-cameras in combination with additional CCD-cameras, which is one of the most crucial parts to be able to use a ToF-camera for 3D scene analysis. The presented approach is suitable for the calibration of multiple CCD- and ToF-cameras, estimating all relevant intrinsic and extrinsic camera parameters. As a checkerboard pattern is also detectable in the amplitude images of ToF-cameras, standard calibration approaches could be exploited for initial parameter estimation. The checkerboard pattern is used to establish correspondences and non-linear least-squares optimization is used to refine the parameters. ToF-cameras suffer from noise and systematic depth errors. To compensate the systematic depth errors a suitable error model was found and the depth deviation parameters of the model are estimated using an analysis-by-synthesis approach. The approach uses synthesized views of the checkerboard depth plane and compares these to real depth images of the ToF-camera. The depth error compensation using different parametrizations as polynomial or cubic B-Spline is evaluated and the B-Spline compensation was found to outperform other parametrizations. Covariance analysis shows that the combined calibration of ToF- and CCD-cameras helps to increase accuracy as rigidly coupled CCD-cameras reduce parameter correlations, especially for parameters that are highly correlated due to the low sensor resolution and the small field-of-view of the ToF-camera. With the presented approach ToF-cameras can be calibrated with high accuracy, making them usable for applications in Computer Vision. Besides the calibration of single Tof-cameras is the calibration of camera systems consisting of several ToF-cameras an important issue for many applications. The presented calibration of multi-ToF-camera systems uses a marker-based calibration object, placed in the middle of the camera system at different locations. From the known geometry of the calibration object all camera positions can be calibrated with high accuracy.

Environment and rigid object modeling is an important topic which offers many possibilities for many applications. The presented 3D scene modeling evaluates the suitability of ToF-cameras for the accurate reconstruction of rigid scenes for environment modeling. The ToF-camera is dedicated for environment modeling as it provides dense depth maps at real-time. For rigid scenes a modeling approach using a pan-tilt unit is presented using a panoramic 2.5D image representation of the data. The contribution shows, that precise dense 3D reconstruction including texture is possible with a combination of ToF- and CCD- camera. Additional to the environment reconstruction, the reconstruction of smaller objects including the estimation of the current pose of the ToF-camera is evaluated. A suitable datastructure, the Octree was exploited to store volumetric data and provide the necessary data fusion of multiple measurements.

The recording and analysis of dynamic three-dimensional scenes is even more challenging than the reconstruction of rigid scenes. The presented dynamic 3D scene capturing and reconstruction approach uses a volumetric data representation. The volumetric representation is realized using an Octree to store the scene data which is extended by one dimension to represent the change over time. In the Octree leaves it is stored which element is visible at which point in time and what color information is valid for this time step. Adding the time as fourth dimension, the full 4D representation of the scene makes it possible to replay animations from different viewpoints and to generate new content. The suitability of ToF-cameras to capture data for this purpose could be manifested.

The introduced Mixed Reality system is a system which allows the composition of real and virtual content. In this work it also allows correct mutual occlusions, shadowing and interaction. It was shown that the contributions of this thesis provided crucial elements of the system. It assembles most of the developed approaches, using camera pose estimation on a generated background model, object segmentation on the GPU using the background model and collision detection for interaction simulation. Full 3D content can be generated of combined real and virtual scenes making the approach suitable for future display technologies such as auto-stereoscopic three-dimensional displays.

7.1. Outlook

Besides the achievements which have been reached in this thesis there are many opportunities for further improvements and investigations. For dynamic 3D scene analysis a reliable and correct segmentation of dynamic content is crucial. With the combination of depth- and color matting using Mixture-of-Gaussians, the segmentation accuracy was increased, but severe problems remain in areas such as the segmentation of a persons feet from the floor at low color difference.

The presented environment modeling with pan-tilt unit is very precise as it does not suffer from registration- or tracking errors as other approaches are prone to. It is however also less flexible due to the usage of the pan-tilt unit. Therefore an environment reconstruction using pose estimation with an additional inertial sensor for a good rotation estimation is an interesting and desired extension to the presented approach. Other researchers are already actively investigating this subject, mostly using ICP (Iterative Closest Point) algorithms and variants for the registration of point clouds.

The presented space-time approach of dynamic scene recording and playback suffers from the low resolution of the ToF-sensor and the quality is at the moment not sufficient. So further work will focus on further decreasing the outliers in the measurement data and on increasing the resolution of the ToF-sensor, for example using super-resolution approaches. Point-based rendering approaches also contribute to the low visual quality. The current point splatting algorithm will perform better with increased resolution, but further improvements in rendering have to be achieved.

Using the multi-camera setup, full 3D models of any subject are available. For many applications the movements of the limbs of a person or animal is of interest. The estimation of the parameters of an articulated model of the subject is therefore the logical next step which is already under research.

A
PMD Operation Principle

Two different camera types have been used in this thesis, the Swissranger ToF-cameras, manufactured by Mesa-Imaging [SSVH95, OLK$^+$03], and the PMD[vision] ToF-cameras manufactured by PMDTec [XSH$^+$98, LSBS99, Sch03]. As already mentioned are the differences located on pixel level and concerning the demodulation and amplification. For the explanation of the measurement principle on pixel level I will discuss the PMD-principle. Figure A.1 shows the simplified two-gate structure of a PMD pixel. The complete mixing process of optical and electrical signal takes place in each pixel. Each pixel is a five terminal device with two semi-transparent modulation electrodes (*am* and *bm*) which serve as optical input window. These are isolated from the substrate by an oxide layer. Two pn-junction diodes are located on both sides of the gates which are contacted and covered by metal electrodes. These diodes are connected to the readout circuitry. The operation mode of a PMD pixel core is based on a charge coupling effect which allows the PMD to have overlapping gates with high charge-transfer efficiency.

The movement of generated charge carriers is controlled by the amplitude of the reference signal u_m applied to the modulation electrodes *am* and *bm*. This way it can be influenced if the charge carriers move to the right or the left pn-junction. If the reference signal is a rectangular signal and the received incident signal is constant, the same amount of charge carriers will be collected on each side of the pn-junctions. If the received incident light is modulated with the same reference signal as the signal u_m applied to the modulation electrodes and there is no phase delay between the incident signal and reference signal all carriers will be moved to one side. For other phase delays the difference of the output voltage will be different, depending on this phase delay (cf. [XSH$^+$98]).

A read out circuit is connected to every pixel of the PMD-sensor. This circuit evaluates the charges of the two electrodes. Therefore the charge, which is accumulated over a certain integration time T_{int}, after which a reset of the circuit is executed, is saved in a capacitor C. A distance correlated charge difference can, according to [Lua01], be measured with the electric streams U_{ak} and U_{bk} of the two electrodes.

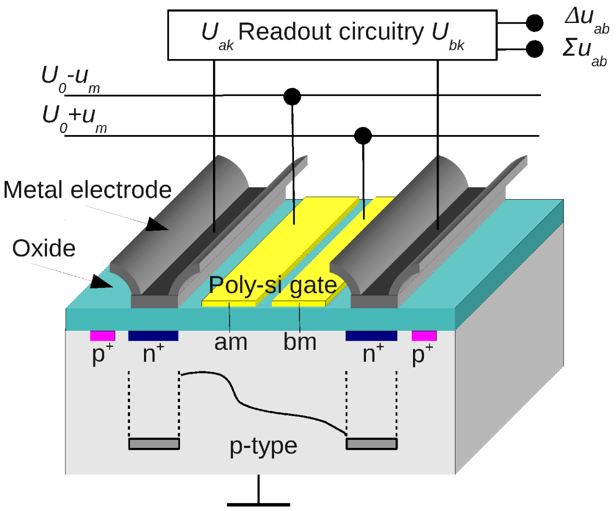


Figure A.1.: The simplified two-gate PMD structure, called a "smart-pixel", according to [XSH+98].

$$\Delta u_{ab} = \frac{T_{int}}{C}(U_{ak} - U_{bk}). \tag{A.1}$$

Simultaneously the typical CCD intensity value is available for every pixel, which can be determined by the sum of the charge of the two electrodes:

$$\Sigma u_{ab} = \frac{T_{int}}{C}(U_{ak} + U_{bk}). \tag{A.2}$$

The intensity image however is not always delivered by the ToF-camera. For the mathematical model of the correlation of the input and output signal it is assumed in the following that the sent out light is modulated with the sine-function. To steer the electron swing, charges U_{ak} and U_{bk} are applied to the electrons, which correspond to symmetric counter-signals to the modulation signal with the same frequency ω_{mod}. This charge consists of a constant charge U_0, a constant amplitude u_m and a selectable shift of the modulation signal by ψ degrees:

$$\begin{aligned} s_{ak}(t) &= U_{ak}(t,\psi) = U_0 + u_m \cdot \sin(\omega t + \psi) \\ s_{bk}(t) &= U_{bk}(t,\psi) = U_0 - u_m \cdot \sin(\omega t + \psi). \end{aligned} \tag{A.3}$$

The light which hits a PMD-element can be described as:

$$r(t) = G_0 + Rcos(\omega_{mod}t - \varphi), \tag{A.4}$$

where G_0 is a factor corresponding to the surrounding light and R is a factor respecting the reflection properties of the object. The result $c(\varphi, \psi)$, the so called auto-correlation of the two signals Φ and U_a can be calculated by multiplication and integration over the time $[0, T_{int}]$, in which T_{int} is a natural multiple of the period $\frac{2\pi}{\omega_{mod}}$ (see [Sch03]).

$$c_a(\psi) = r(t) \otimes s(t) = \lim_{T \to \infty} \frac{1}{T_{int}} \int_0^{T_{int}} r(t) \cdot s_{ak}(t + \psi)dt \tag{A.5}$$

And analogous to equation A.5 it yields:

$$c_b(\psi) = r(t) \otimes s(t) = \lim_{T \to \infty} \frac{1}{T_{int}} \int_0^{T_{int}} r(t) \cdot s_{bk}(t + \psi)dt \tag{A.6}$$

The factor c_0 is a value for the sensitivity of the semiconductor, which is however not relevant for the upcoming calculations. Equation A.5 can be simplified according to [Sch03] as:

$$c_a(\psi) = H + Mcos(\varphi - \psi) \tag{A.7}$$

analog is valid for the symmetric counter signal:

$$c_b(\psi) = H - Mcos(\varphi - \psi), \tag{A.8}$$

with $\varphi = f(d, \omega)$ a variable phase shift, an unmodulated background signal $H = f(U_0, G_0, T_{int})$ and a modulated signal $M = f(U, R, T_{int})$.

Please continue reading at section 3.1.1, subsection “Distance Calculation”.

Bibliography

[BBH03] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 25(8):993–1008, 2003. ISSN: 0162-8828.

[BBK07a] C. Beder, B. Bartczak, and R. Koch. A Comparison of PMD-Cameras and Stereo-Vision for the Task of Surface Reconstruction using Patchlets. In *IEEE/ISPRS Workshop BenCOS 2007*, 2007.

[BBK07b] C. Beder, B. Bartczak, and R. Koch. A combined approach for estimating patchlets from pmd depth images and stereo intensity images. In Fred Hamprecht, Christoph Schnörr, and Bernd Jähne, editors, *Pattern Recognition*, volume 4713 of *Lecture Notes in Computer Science*, pages 11–20. Springer Berlin / Heidelberg, 2007.

[BK87] K. L. Boyer and A. C. Kak. Color-encoded structured light for rapid active ranging. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 9(1):14–28, 1987.

[BK08] C. Beder and R. Koch. Calibration of focal length and 3d pose based on the reflectance and depth image of a planar object. *International Journal of Intelligent Systems Technologies and Applications*, 5(3-4):285–294, 2008.

[BKT06] M. Bray, P. Kohli, and P. H. S. Torr. Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph-cuts. In *European Conference on Computer Vision*, pages 642–655, 2006.

[BL06] B. Büttgen and F. Lustenberger. Demodulation pixel based on static drift fields. In *IEEE Transactions on Electron Devices*, volume 53, NO. 11, November 2006.

[Bou99] J.-Y. Bouguet. *Visual methods for three-dimensional modelling*. PhD thesis, California Institute of Technology, Pasadena, CA, USA, 1999.

[BS08] B. Büttgen and P. Seitz. Robust optical time-of-flight range imaging based on smart pixel structures. In *IEEE Transactions on Circuits and Systems*, volume 55, No. 6. IEEE, July 2008.

[BSBK08] B. Bartczak, I. Schiller, C. Beder, and R. Koch. Integration of a time-of-flight camera into a mixed reality system for handling dynamic scenes, moving viewpoints and occlusions in real-time. In *Proceedings of the 3DPVT Workshop*, Atlanta, GA, USA, June 2008.

[Bul10] Bullet. Bullet open source physics library, version 2.77. `http://bulletphysics.org`, visited on August 17th, 2010.

[BVZ01] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 23:1222–1239, 2001.

[CBL99] C.-F. Chang, G. Bishop, and A. Lastra. Ldi tree: a hierarchical representation for image-based rendering. In *ACM Trans. Graph. (Proc. SIGGRAPH)*, pages 291–298, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[CCV85] I. Carlbom, I. Chakravarty, and D. Vanderschel. A hierarchical data structure for representing the spatial decomposition of 3-d objects. *Computer Graphics and Applications, IEEE*, 5(4):24–31, April 1985.

[CL96] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *ACM Trans. Graph. (Proc. SIGGRAPH)*, pages 303–312, New York, NY, USA, 1996. ACM.

[Con84] C. Connolly. Cumulative generation of octree models from range data. In *Robotics and Automation. Proceedings. IEEE International Conference on*, volume 1, pages 25–32, 1984.

[CTPD08] R. Crabb, C. Tracey, A. Puranik, and J. Davis. Real-time foreground segmentation via range and color imaging. In *Computer Vision and Pattern Recognition Workshops, CVPRW, IEEE Computer Society Conference on*, pages 1–5, June 2008.

[Dav03] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings International Conference Computer Vision (ICCV), Nice*, 2003.

[Del34] B. N. Delaunay. Sur la sphère vide. *Bulletin of Academy of Sciences of the USSR*, 7(6):793–800, 1934.

[EHBPG04] S. F. El-Hakim, J.-A. Beraldin, M. Picard, and G. Godin. Detailed 3d reconstruction of large-scale heritage sites with integrated techniques. *IEEE Comput. Graph. Appl.*, 24(3):21–29, 2004.

[ESK05] J.-F. Evers-Senne and R. Koch. Image-based rendering of complex scenes from a multi-camera rig. In *IEE Proceedings Vision Image and Signal Processing*, volume 152, Number 4, August 2005.

[FB81] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, June 1981.

[FH08] S. Fuchs and G. Hirzinger. Extrinsic and depth calibration of tof-cameras. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–6, June 2008.

[FKBK09] A. Frick, F. Kellner, B. Bartczak, and R. Koch. Generation of 3d-tv ldv-content with time-of-flight camera. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2009*, pages 1 –4, may 2009.

[FTTR99] A. Fusiello, E. Trucco, T. Tommasini, and V. Roberto. Improving feature tracking with robust statistics. *Pattern Analysis and Applications*, 2:312–320, 1999.

[FW04] W. Förstner and B. Wrobel. Mathematical concepts in photogrammetry. In J.C.McGlone, E.M.Mikhail, and J.Bethel, editors, *Manual of Photogrammetry*, pages 15–180. ASPRS, 2004.

[GBS$^+$05] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *International Conference on Computer Vision*, 2:1395–1402, 2005.

[GD96] D. M. Gavrila and L. S. Davis. 3-d model-based tracking of humans in action: a multi-view approach. In *IEEE Conf. on Computer Vision and Pattern Recognition*, page 73, Washington, DC, USA, 1996. IEEE Computer Society. ISBN: 0-8186-7258-7.

[GDHW99] G. Gordon, T. Darrell, M. Harville, and J. Woodfill. Background estimation and removal based on range and color. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 459–464, 1999.

[GFP08] L. Guan, J.-S. Franco, and M. Pollefeys. 3D Object Reconstruction with Heterogeneous Sensor Data. In *International Symposium on 3D Data Processing, Visualization and Transmission*, Atlanta États-Unis d'Amérique, 2008.

[GHW$^+$06] S.E. Ghobadi, K. Hartmann, W. Weihs, C. Netramai, O. Loffeld, and H. Roth. Detection and classification of moving objects-stereo or time-of-flight images. In *Computational Intelligence and Security*, pages 11–16, Center for Sensor Systems, University of Siegen, 2006. IEEE.

[GKOY03] R. Gvili, A. Kaplan, E. Ofek, and G. Yahav. Depth keying. *Stereoscopic Displays and Virtual Reality Systems X*, 5006(1):564–574, 2003.

[GSdA$^+$09] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 0, pages 1746–1753, Los Alamitos, CA, USA, 2009. IEEE Computer Society. ISBN: 978-1-4244-3992-8.

[HBMB08] M. Haker, M. Bohme, T. Martinetz, and E. Barth. Scale-invariant range features for time-of-flight camera applications. In *Computer Vision and Pattern Recognition Workshops, CVPRW, IEEE Computer Society Conference on*, pages 1–6, June 2008.

[Hir05] H. Hirschmueller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 807–814, Washington, DC, USA, 2005. IEEE Computer Society. ISBN: 0-7695-2372-2.

[HJS08] B. Huhle, P. Jenke, and W. Straßer. On-the-fly scene acquisition with a handy multisensor-system. *International Journal of Intelligent Systems Technologies and Applications*, 5, No.3/4:255 – 263, 2008.

[HS97] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1106–1112, Washington, DC, USA, 1997. IEEE Computer Society.

[HS07] H. Hirschmueller and D. Scharstein. Evaluation of cost functions for stereo matching. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[HSJS08] B. Huhle, T. Schairer, P. Jenke, and W. Strasser. Robust non-local denoising of colored depth data. In *Computer Vision and Pattern Recognition Workshops, CVPRW, IEEE Computer Society Conference on*, pages 1–7, June 2008.

[HYN03] J. Hu, S. You, and U. Neumann. Approaches to large-scale urban modeling. *IEEE Comput. Graph. Appl.*, 23(6):62–69, 2003.

[HZ04] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2. Edition, 2004.

[JHS07] P. Jenke, B. Huhle, and W. Strasser. Self-localization in scanned 3dtv sets. In *3DTV CON - The True Vision*, pages 1–4. Univ. of Tubingen, Tubingen, 2007.

[JWB$^+$06] P. Jenke, M. Wand, M. Bokeloh, A. Schilling, and W. Straßer. Bayesian point cloud reconstruction. *Computer Graphics Forum*, 25(3):379–388, 2006.

[KBK07] K. Koeser, B. Bartczak, and R. Koch. Robust gpu-assisted camera tracking using free-form surface models. *Journal of Real Time Image Processing*, 2:133–147, 2007.

[KCK09] M. Keller, N. Cuntz, and A. Kolb. Interactive dynamic volume trees on the gpu. In *Proc. Vision, Modeling and Visualization*, pages 165–176, 2009.

[KCLU07] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 26, July 2007.

[KCTT08] Y. M. Kim, D. Chan, C. Theobalt, and S. Thrun. Design and calibration of a multi-view tof sensor fusion system. In *Computer Vision and Pattern Recognition Workshops, CVPRW, IEEE Computer Society Conference on*, pages 1–7, June 2008.

[KFM$^+$04] H. Kraft, J. Frey, T. Moeller, M. Albrecht, M. Grothof, B. Schink, H. Hess, and B. Buxbaum. 3d-camera of high 3d-frame rate, depth-resolution and background light elimination based on improved pmd (photonic mixer device)-technologies. In *6th International Conference for Optical Technologies, Optical Sensors and Measuring Techniques (OPTO)*, May 2004.

[KPHB08] E. Kollorz, J. Penne, J. Hornegger, and A. Barke. Gesture recognition with a time-of-flight camera. *International Journal of Intelligent Systems Technologies and Applications*, 5(3-4):334–343, 2008.

[KRI06] T. Kahlmann, F. Remondino, and H. Ingensand. Calibration for increased accuracy of the range imaging camera SwissrangerTM. In *ISPRS Commission V Symposium Image Engineering and Vision Metrology, IEVM06*, 2006.

[KS06] K.-D. Kuhnert and M. Stommel. Fusion of stereo-camera and pmd-camera data for real-time suited precise 3d environment reconstruction. In *International Conference on Intelligent Robots and Systems, IEEE/RSJ*, pages 4780–4785, Oct 2006.

[KSB$^+$09] R. Koch, I. Schiller, B. Bartczak, F. Kellner, and K. Koeser. Mixin3d: 3d mixed reality with tof-camera. In Andreas Kolb and Reinhard Koch, editors, *Dynamic 3D Imaging*, volume 5742 of *Lecture Notes in Computer Science*, pages 126–141. Springer Berlin / Heidelberg, 2009.

[KSD09] S. Knoop, S.Vacek, and R. Dillmann. Fusion of 2d and 3d sensor data for articulated body tracking. *Robotics and Autonomous Systems*, 57(3):321 – 329, 2009. Selected papers from 2006 IEEE International Conference on Multisensor Fusion and Integration (MFI 2006), 2006 IEEE International Conference on Multisensor Fusion and Integration.

[KTD$^+$09] Y.M. Kim, C. Theobalt, J. Diebel, J. Kosecka, B. Micusika, and S. Thrun. Multi-view image and tof sensor fusion for dense 3d reconstruction. In *IEEE Workshop on 3-D Digital Imaging and Modeling (3DIM)*, 2009.

[KZ01] V. Kolmogorov and R. Zabih. Proceedings of the computing visual correspondence with occlusions via graph cuts. In *International Conference on Computer Vision*, pages 508–515, 2001.

[LC87] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM Trans. Graph. (Proc. SIGGRAPH)*, SIGGRAPH '87, pages 163–169, New York, NY, USA, 1987. ACM.

[LC94] A. Li and G. Crebbin. Octree encoding of objects from range images. *Pattern Recognition*, 27(5):727 – 739, 1994.

[Lin10] M. Lindner. *Calibration and Realtime Processing of Time-of-Flight Range Data*. PhD thesis, University of Siegen, 2010.

[LK06] M. Lindner and A. Kolb. Lateral and depth calibration of pmd-distance sensors. In *International Symposium on Visual Computing (ISVC)*, volume 2, pages 524–533. Springer, 2006.

[LK07] M. Lindner and A. Kolb. Calibration of the intensity-related distance error of the PMD ToF-camera. In *SPIE, Intelligent Robots and Computer Vision*, volume 6764, 2007. doi:10.1117/12.752808.

[Low04] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[LSBS99] R. Lange, P. Seitz, A. Biber, and R. Schwarte. Time-of-flight range imaging with a custom solid-state imagesensor. In *EOS/SPIE Laser Metrology and Inspection*, volume 3823, 1999.

[LSKK10] Marvin Lindner, Ingo Schiller, Andreas Kolb, and Reinhard Koch. Time-of-flight sensor calibration for accurate range sensing. *Journal on Computer Vision and Image Understanding*, 114(12):1318 – 1328, 2010. Special issue on Time-of-Flight Camera Based Computer Vision.

[Lua01] X. Luan. *Experimental investigation of Photonic Mixer Device and development of TOF 3D ranging systems based on PMD technology*. PhD thesis, Department of Electrical Engineering and Computer Science, Universität Siegen, 2001.

[Mar10] *Vision - A Computational Investigation into the Human Representation and Processing of Visual Information*. MIT Press, 2010.

[MB95] L. McMillan and G. Bishop. Plenoptic modeling: an image-based rendering system. In *ACM Trans. Graph. (Proc. SIGGRAPH)*, pages 39–46, New York, NY, USA, 1995. ACM.

[MBB$^+$08] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen,

I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun. Junior: The stanford entry in the urban challenge. *Journal of Field Robotics*, 25(9):569–597, 2008.

[MDH07a] J. Mure-Dubois and H. Hügli. Optimized scattering compensation for time-of-flight camera. In *Proc. Conf. Two- and Three-Dimensional Methods for Inspection and Metrology V, Proc. SPIE*, volume 6762-0H, 2007.

[MDH07b] J. Mure-Dubois and H. Hügli. Real-time scattering compensation for time-of-flight camera. In *Proc. Workshop on Camera Calibration Methods for Computer Vision Systems - CCMVS2007*, Bielefeld, Germany, 2007.

[MDH$^+$08] Stefan May, David Droeschel, Dirk Holz, Christoph Wiesen, and Stefan Fuchs. 3D Pose Estimation and Mapping with Time-of-Flight Cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on 3D Mapping*, Nice, France, October 2008.

[Mea82] D. Meagher. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 19(2):129 – 147, 1982.

[MFD$^+$09] Stefan May, Stefan Fuchs, David Droeschel, Dirk Holz, and Andreas Nüchter. Robust 3D-Mapping with Time-of-Flight Cameras. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1673–1678, St. Louis, Missouri, USA, October 2009.

[MFW$^+$04] C. J. Mugnier, W. Förstner, B. Wrobel, F. Paderes, and R. Munjy. The mathematics of photogrammetry. In J.C.McGlone, E.M.Mikhail, and J.Bethel, editors, *Manual of Photogrammetry*, pages 181–316. ASPRS, 2004.

[MFY$^+$09] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto. View generation with 3d warping using depth information for ftv. *Signal Processing: Image Communication*, 24(1-2):65 – 72, 2009. Special issue on advances in three-dimensional television and video.

[MHK06] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Journal on Computer Vision and Image Understanding*, 104(2-3):90 – 126, 2006. Special Issue on Modeling People: Vision-based understanding of a person's shape, appearance, movement and behaviour.

[MPW07] S. Matzka, Y. R. Petillot, and A. M. Wallace. Fast motion estimation on range image sequences acquired with a 3-d camera. In *Proceedings of the Britisch Machine Vision Conference, BMVC*, 2007.

[NBD77] D. Nitzan, A.E. Brain, and R.O. Duda. The measurement and use of registered reflectance and range data in scene analysis. *Proceedings of the IEEE*, 65(2):206 – 220, feb. 1977.

[OLK$^+$03] T. Oggier, M. Lehmann, R. Kaufmann, M. Schweizer, M. Richter, P. Metzler, G. Lang, F. Lustenberger, and N. Blanc. An all-solid-state optical range camera for 3d real-time imaging with sub-centimeter depth resolution. In *Proceedings of SPIE.*, volume SPIE-5249, pages 534–545, 2003.

[Ope10] OpenCV. OpenCV: Open source Computer Vision library, version 2.1. `http://opencv.willowgarage.com`, visited on August 17th, 2010.

[PG08] Y. Pekelny and C. Gotsman. Articulated object reconstruction and markerless motion capture from depth video. *J. Computer Graphics Forum*, volume 27(2), 2008.

[PH07] S. Perreault and P. Hebert. Median filtering in constant time. *Image Processing, IEEE Transactions on*, 16(9):2389 –2394, sep. 2007.

[PHW$^+$06] T.D.A. Prasad, K. Hartmann, W. Wolfgang, S.E. Ghobadi, and A. Sluiter. First steps in enhancing 3d vision technique using 2d/3d sensors. In V. Chum, O.Franc, editor, *11. Computer Vision Winter Workshop 2006*, pages 82–86, University of Siegen, 2006. Czech Society for Cybernetics and Informatics.

[PKVVG98] Marc Pollefeys, Reinhard Koch, Maarten Vergauwen, and Luc Van Gool. Metric 3d surface reconstruction from uncalibrated image sequences. In Reinhard Koch and Luc Van Gool, editors, *3D Structure from Multiple Images of Large-Scale Environments*, volume 1506 of *Lecture Notes in Computer Science*, pages 139–154. Springer Berlin / Heidelberg, 1998.

[PMS$^+$08] A. Prusak, O. Melnychuk, Ingo Schiller, H. Roth, and R. Koch. Pose estimation and map building with a pmd-camera for robot navigation. *International Journal of Intelligent Systems Technologies and Applications*, 5(3-4):355–364, 2008.

[PSA$^+$04] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 23(3):664–672, 2004.

[RFHJ08] H. Rapp, M. Frank, F. A. Hamprecht, and B. Jahne. A theoretical and experimental investigation of the systematic errors and statistical uncertainties of time-of-flight-cameras. *International Journal of Intelligent Systems Technologies and Applications*, 5(3/4):402–413, 2008.

[RKB04] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. In *ACM Trans. Graph. (Proc. SIGGRAPH)*, pages 309–314, New York, NY, USA, 2004. ACM.

[RL00] S. Rusinkiewicz and M. Levoy. Qsplat: a multiresolution point rendering system for large meshes. In *ACM Trans. Graph. (Proc. SIGGRAPH)*, pages 343–352, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[Sam89] H. Samet. Implementing ray tracing with octrees and neighbor finding. *Computers And Graphics*, 13:445–460, 1989.

[SBK08] I. Schiller, C. Beder, and R. Koch. Calibration of a pmd camera using a planar calibration object together with a multi-camera setup. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume Vol. XXXVII. Part B3a, pages 297–302, Beijing, China, 2008. XXI. ISPRS Congress.

[SBKK07] B. Streckel, B. Bartczak, R. Koch, and A. Kolb. Supporting structure from motion with a 3d-range-camera. In *Scandinavian Conference on Image Analysis*, June 2007.

[SBKK08] I. Schiller, B. Bartczak, F. Kellner, and R. Koch. Increasing realism and supporting content planning for dynamic scenes in a mixed reality system incorporating a time-of-flight camera. In *Proceedings of the European Conference on Visual Media Production, CVMP*, volume 5, London, UK, 2008.

[SBKK10] I. Schiller, B. Bartczak, F. Kellner, and R. Koch. Increasing realism and supporting content planning for dynamic scenes in a mixed reality system incorporating a time-of-flight camera. *Journal of Virtual Reality and Broadcasting 7, CVMP 2008 Special Issue*, no. 4, August 2010. urn:nbn:de:0009-6-25786, issn 1860-2037.

[SCD$^+$06] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 519–528, 2006.

[Sch03] B. Schneider. *Der Photomischdetektor zur schnellen 3D-Vermessung für Sicherheitssysteme und zur Informationsübertragung im Automobil*. PhD thesis, Department of Electrical Engineering and Computer Science, Universität Siegen, 2003.

[SEG99] C. Stauffer, W. Eric, and L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2246–2252, 1999.

[SF08] S. May S. Fuchs. Calibration and registration for precise surface reconstruction with time-of-flight cameras. *International Journal of Intelligent Systems Technologies and Applications*, 5. No.3/4:274–284, 2008.

[SGHS98] J. Shade, S. Gortler, L. He, and R. Szeliski. Layered depth images. In *ACM Trans. Graph. (Proc. SIGGRAPH)*, pages 231–242, New York, NY, USA, 1998. ACM.

[SJ97] Julier S.J. and Uhlmann J.K. A new extension of the kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls 3*, 1997.

[SK09] I. Schiller and R. Koch. Datastructures for capturing dynamic scenes with a time-of-flight camera. In Andreas Kolb and Reinhard Koch, editors, *Dynamic 3D Imaging*, volume 5742 of *Lecture Notes in Computer Science*, pages 42–57. Springer Berlin / Heidelberg, 2009.

[SK10] A. Sabov and J. Krüger. Identification and correction of flying pixels in range camera data. In *Proceedings of the 24th Spring Conference on Computer Graphics*, SCCG '08, pages 135–142, New York, NY, USA, 2010. ACM.

[SK11] I. Schiller and R. Koch. Improved video segmentation by adaptive combination of depth keying and mixture-of-gaussians. In Anders Heyden and Fredrik Kahl, editors, *Image Analysis*, volume 6688 of *Lecture Notes in Computer Science*, pages 59–68. Springer Berlin / Heidelberg, 2011. 10.1007/978-3-642-21227-7_6.

[SMS06] D. Scaramuzza, A. Martinelli, and R. Siegwart. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *Proceedings of IEEE International Conference of Vision Systems*. IEEE, January 2006.

[SS02] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002. ISSN: 0920-5691.

[SS03] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 195–202, June 2003.

[SSVH95] T. Spirig, P. Seitz, O. Vietze, and F. Heitger. The lock-in ccd-two-dimensional synchronous detection of light. *IEEE Journal of Quantum Electronics*, 31(9):1705–1708, Sep 1995.

[ST94] J. Shi and C. Tomasi. Good features to track. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, June 1994. IEEE.

[Sze93] R. Szeliski. Rapid octree construction from image sequences. *CVGIP: Image Underst.*, 58(1):23–32, 1993.

[SZS03] J. Sun, N.-N. Zheng, and H.-Y. Shum. Stereo matching using belief propagation. *IEEE Trans. Pattern Anal. and Mach. Intell.*, 25(7):787–800, 2003.

[TMD$^+$07] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Stanley: The robot that won the darpa grand challenge. 36:1–43, 2007.

[VMVPVG02] G. Van Meerbergen, M. Vergauwen, M. Pollefeys, and L. Van Gool. A hierarchical symmetric stereo algorithm using dynamic programming. *International Journal of Computer Vision*, 47(1-3):275–285, 2002.

[WB95] G. Welch and G. Bishop. An introduction to the kalman filter. Technical Report 95-041, University of North Carolina, Chapel Hill, NC, USA, 1995.

[WBB08] Q. Wu, P. Boulanger, and W.F. Bischof. Automatic bi-layer video segmentation based on sensor fusion. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, Dec. 2008.

[Wei06] B. Weiss. Fast median and bilateral filtering. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 25(3):519–526, 2006.

[WGGY06] L. Wang, M. Gong, M. Gong, and R. Yang. How far can we go with local optimization in real-time stereo matching. In *Proceedings of the International Symposium on 3D Data Processing Visualization and Transmission*, volume 0, pages 129–136, Los Alamitos, CA, USA, 2006. IEEE Computer Society. ISBN: 0-7695-2825-2.

[Wil78] L. Williams. Casting curved shadows on curved surfaces. *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 12(3):270–274, 1978. ISSN 0097-8930.

[WJH$^+$07] M. Wand, P. Jenke, Q. Huang, M. Bokeloh, L. Guibas, and A. Schilling. Reconstruction of deforming geometry from time-varying point clouds. In *SGP '07: Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 49–58, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[WYM08] C. Wan, B. Yuan, and Z. Miao. Markerless human body motion capture using markov random field and dynamic graph cuts. *The Visual Computer*, Volume 24:373 – 380, 2008.

[XE01] M. Xu and T. Ellis. Illumination-invariant motion detection using colour mixture models. In *British Machine Vision Conference (BMVC)*, pages 163–172, 2001.

[XSH$^+$98] Z. Xu, R. Schwarte, H. Heinol, B. Buxbaum, and T. Ringbeck. Smart pixel - photonic mixer device (pmd). In *M2VIP - International Conference on Mechatronics and Machine Vision in Practice*, pages 259 – 264, 1998.

[YHNF03] S. You, J. Hu, U. Neumann, and P. Fox. Urban site modeling from lidar. In *Proceedings of the 2nd Int. Workshop Computer Graphics and Geometric Modeling (CGGM*, page 588, 2003.

[YIM07] G. Yahav, G.J. Iddan, and D. Mandelboum. 3d imaging camera for gaming application. In *International Conference on Consumer Electronics (ICCE). Digest of Technical Papers*, pages 1–2, January 2007.

[ZCS03] L. Zhang, B. Curless, and S. M. Seitz. Spacetime stereo: Shape recovery for dynamic scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 367–374, June 2003.

[ZH04] S. Zhang and P. Huang. High-resolution, real-time 3d shape acquisition. In *Computer Vision and Pattern Recognition Workshops, CVPRW, IEEE Computer Society Conference on*, volume 3, page 28, Washington, DC, USA, 2004. IEEE Computer Society.

[Zha99] Z. Zhang. Flexible Camera Calibration by Viewing a Plane from Unknown Orientations. In *International Conference on Computer Vision*, pages 666–673, Corfu, Greece, 1999.

[ZLYP09] J. Zhu, M. Liao, R. Yang, and Z. Pan. Joint depth and alpha matte optimization via fusion of stereo and time-of-flight sensor. *IEEE Conf. on Computer Vision and Pattern Recognition*, 0:453–460, 2009.