Robust Single-Shot Structured Light 3D Scanning

Robuste 3D-Vermessung mit strukturierter Beleuchtung in Einzelbildern

Der Technischen Fakultät der
Universität Erlangen–Nürnberg

zur Erlangung des Grades

DOKTOR–INGENIEUR

vorgelegt von

Christoph Schmalz

Erlangen — 2011

Als Dissertation genehmigt von der
Technischen Fakultät der
Universität Erlangen-Nürnberg

Tag der Einreichung:	29.08.2011
Tag der Promotion:	05.03.2012
Dekan:	Prof. Dr.-Ing. habil. Marion Merklein
Berichterstatter:	Prof. Dr.-Ing. Joachim Hornegger
	Prof. Dr.-Ing. Gerd Häusler

Abstract

In this thesis a new robust approach for Single-Shot Structured Light 3D scanning is developed. As the name implies, this measurement principle requires only one image of an object, illuminated with a suitable pattern, to reconstruct the shape and distance of the object. This technique has several advantages. It can be used to record 3D video with a moving sensor or of a moving scene. Since the required hardware is very simple, the sensor can also be easily miniaturized. Single-Shot Structured Light, thus, has the potential to be the basis of a versatile and inexpensive 3D scanner.

One focus of the work is the *robustness* of the method. Existing approaches are mostly limited to simple scenes, that is, smooth surfaces with neutral color and no external light. In contrast, the proposed method can work with almost any close-range scene and produces reliable range images even for very low-quality input images. An important consideration in this respect is the design of the illumination pattern. We show how suitable color stripe patterns for different applications can be created. A major part of the robustness is also due to the graph-based decoding algorithm for the pattern images. This has several reasons. Firstly, any color assessments are based on ensembles of pixels instead of single pixels. Secondly, disruptions in the observed pattern can be sidestepped by finding alternative paths in the graph. Thirdly, the graph makes it possible to apply inference techniques to get better approximations of the projected colors from the observed colors. For a typical camera resolution of 780×580, the whole decoding and reconstruction algorithm runs at 25Hz on current hardware and generates up to 50000 3D points per frame.

The *accuracy* of the recovered range data is another important aspect. We implemented a new calibration method for cameras and projectors, which is based on active targets. The calibration accuracy was evaluated using the reprojection error for single camera calibrations as well as the 3D reconstruction errors for complete scanner calibrations. The accuracy with active targets compares favorably to calibration results with classic targets. In a stereo triangulation test, the root-mean-square error could be reduced to a fifth. The accuracy of the combined Structured Light setup of camera and projector was also tested with simulated and real test scenes. For example, using a barbell-shaped reference object, its known length of 80.0057mm could be determined with a mean absolute error of 42µm and a standard deviation of 74µm.

The runtime performance, the robustness and the accuracy of the proposed approach are very competitive in comparison with previously published methods. Finally, endoscopic 3D scanning is a showcase application that is hard to replicate without Single-Shot Structured Light. Building on a miniature sensor head designed by Siemens, we developed calibration algorithms and apply the graph-based pattern decoding to generate high-quality 3D cavity reconstructions.

Kurzfassung

In dieser Arbeit wird ein neues robustes Verfahren zur 3D-Vermessung durch Strukturierte Beleuchtung in Einzelbildern entwickelt. Dieses Messprinzip benötigt nur ein einzige Aufnahme eines mit einem geeigneten Muster beleuchteten Objekts, um dessen Form und Abstand zu rekonstruieren. Diese Technik hat mehrere Vorteile. Sie kann benutzt werden, um 3D-Videos einer bewegten Szene oder mit einem bewegten Sensor aufzunehmen. Da sein Aufbau sehr einfach ist, ist der Sensor auch gut zur Miniaturisierung geeignet. Strukturierte Beleuchtung in Einzelbildern hat daher das Potential, als Grundlage für vielseitige und günstige 3D-Abtaster zu dienen.

Ein Schwerpunkt der Arbeit ist die *Robustheit* der Messmethode. Existierende Ansätze sind meistens auf einfache Szenen beschränkt, das bedeutet glatte Oberflächen in neutralen Farben und kein Fremdlicht. Im Gegensatz dazu kann die vorgeschlagene Methode mit fast jeder Szene im Nahbereich umgehen und zuverlässige Tiefenkarten auch aus Eingangsbildern mit sehr niedriger Qualität erzeugen. Eine wichtige Überlegung ist in dieser Hinsicht die Gestaltung des Beleuchtungsmusters. Wir zeigen, wie geeignete Farbstreifenmuster für verschiedene Anwendungen erzeugt werden können. Ein Großteil der Robustheit beruht auch auf dem graphenbasierten Dekodierungsalgorithmus für die Aufnahmen des Muster. Das hat mehrere Gründe. Erstens werden alle Farbeinschätzungen anhand von Gruppen von Pixeln anstatt Einzelpixeln vorgenommen. Zweitens können Störungen im beobachteten Muster umgangen werden, indem alternative Pfade im Graphen gefunden werden. Drittens erlaubt es der Graph, Folgerungstechniken anzuwenden, um bessere Näherungen für die projizierten Farben aus den beobachteten Farben zu erhalten. Mit einer üblichen Kameraauflösung von 780×580 läuft der gesamte Algorithmus zur Dekodierung und Rekonstruktion mit 25Hz und erzeugt bis zu 50000 3D-Punkte pro Bild.

Die *Genauigkeit* der gewonnenen 3D-Daten ist ein weiterer wichtiger Aspekt. Wir implementierten eine neue Kalibriermethode für Kameras und Projektoren, die auf aktiven Targets basiert. Die Kalibriergenauigkeit wurde sowohl anhand des Rückprojektionsfehlers für Einzelkamerakalibrierungen, als auch anhand des 3D-Rekonstruktionsfehlers für vollständige Systemkalibrierungen ermittelt. Mit aktiven Targets wird eine höhere Genauigkeit als mit klassischen Targets erreicht. Bei einem Test durch Triangulation mit zwei Kameras konnte der mittlere quadratische Fehler auf ein Fünftel reduziert werden. Die Genauigkeit des Aufbaus zur Strukturierten Beleuchtung aus Kamera und Projektor wurde ebenfalls ausgewertet. Die bekannte Länge eines hantelförmigen Referenzobjekts von 80.0057mm konnte mit einem mittleren Fehler von 42µm und einer Standardabweichung von 74µm bestimmt werden.

Die Rechenzeit, die Robustheit und die Genauigkeit der vorgeschlagenen Messmethode sind im Vergleich mit bisherigen Ansätzen sehr konkurrenzfähig. Eine Vorzeigeanwendung ist die endoskopische 3D-Abtastung, die ohne die Technik der Strukturierten Beleuchtung in Einzelbildern schwer umzusetzen ist. Aufbauend auf einem von Siemens entworfenen Miniatur-Sensorkopf entwickelten wir Kalibrierverfahren und wenden die graphenbasierte Musterdekodierung an, um hochqualitative 3D-Modelle von Hohlräumen zu erzeugen.

Acknowledgement

The present work is the result of my research at the Chair of Pattern Recognition of the University of Erlangen-Nuremberg and at Siemens CT T HW2 in Munich.

I would like to thank Prof. Dr. Joachim Hornegger for giving me the opportunity to become a member of the Pattern Recognition Lab and to work in such an excellent environment.

I am very grateful to my advisor Dr. Frank Forster for his support, his guidance and his valuable insights. The same is also true for Dr. Elli Angelopoulou, who accepted me in the Computer Vision group and always had advice for me.

Furthermore, I would like to thank my colleagues at the Pattern Recognition Lab and at Siemens for the memorable experiences shared over the past years. Special thanks go to Philip Mewes for his help in acquiring the pig stomach datasets, which was an experience in itself.

Christoph Schmalz

Contents

Chapter 1

Introduction

1.1 Motivation

The space around us is three-dimensional. Yet, the most common representations of the world have long been only two-dimensional. There is a long tradition of paintings, and later photographs and movies. We are very good at interpreting such two-dimensional views. After all, on the retina all our visual impressions turn into a 2D pattern. However, some information is irretrievably lost in the process. Especially for technical applications, the extra dimension is important in order to extract metric measurements from an image. Fortunately, it is possible to capture the distance to an object as well and build a true 3D model of its shape. These 3D models are often also used to generate enhanced 2D views from novel viewpoints. The direct and indirect application of 3D measurement technology is very common. It is used in the entertainment industry for movie production, gaming and virtual reality. It is used in medicine for patient positioning and surgery planning. It is further used in computer vision, quality control, reverse engineering, heritage conservation, crime scenes documentation and in security applications. The demands on the level of detail and realism are increasing steadily. Therefore 3D acquisition techniques must also improve.

Optical shape acquisition methods have the advantage that they are contactless and can work at a distance. A very successful branch of optical shape acquisition is Structured Light. It works by projecting suitable patterns onto the scene and observing them with a camera. However, areas that have been difficult so far are moving scenes and real-time 3D video. This work specifically seeks to improve *Single-Shot Structured Light 3D scanning*. In contrast to many other methods this approach requires a single image of the scene, illuminated with a single static projection pattern, to reconstruct the 3D data. Therefore our technique is suitable for measuring dynamic scenes. We also made sure that the underlying algorithms operate in real-time so that live feedback is possible. Another advantage of a Single-Shot system is that the required hardware is very simple. That means the setup can be very compact and inexpensive, using off-the-shelf components. Another focus is robustness. The system should work with a wide variety of objects, which has traditionally been hard to achieve. Experience also shows that in real life the image quality is often low, be it because of environment influences or uncooperative objects in the scene.

Being able to generate reliable 3D from such images in turn makes is possible to contemplate novel applications like endoscopic scanning. Because of limitations in the miniaturized hardware, the image quality in this case is necessarily relatively low. For applications in medicine this is exacerbated by effects like volume scattering and specular highlights, which are typical for biological tissue. The proposed pattern decoding algorithm is designed with such complications in mind. Also, the optimal illumination pattern is expected to vary with the application. The proposed algorithm is very flexible and not geared towards a specific pattern.

Once the pattern image has been successfully decoded, 3D data may be reconstructed. To that end the projector and the camera have to be calibrated, that is, all their relevant parameters have to be known. The calibration quality of a Structured Light system is a crucial step to assure the accuracy of the resulting 3D data. We use a calibration method based on active targets. The basic idea is to replace the noise-prone feature localization step in the classic calibration algorithms by a suitable encoding of the target surface with Structured Light patterns. We demonstrate that this approach offers a substantial improvement in accuracy.

1.2 Organization

The proposed system combines many aspects from different areas. In chapter 2 some basics are introduced. This includes the most important properties of the hardware for 2D imaging and projection as well the models used for the calibration of cameras and projectors. In a second part the existing body of work on image processing methods like edge detection and segmentation is summed up. This is important, as the accuracy of the proposed system rests on the ability to locate edges in the camera image. Also, the pattern decoding algorithm that will be introduced in chapter 5 is built on a watershed segmentation of the input images. Finally, we shortly introduce the area of GPU programming, which offers dramatic speedups for image processing. This helps to reach the goal of real-time operation for the proposed 3D scanning system.

Chapter 3 presents the state of the art in optical 3D shape measurement. We give particular attention to triangulation-based methods, but we also include other optical approaches. Naturally, other Structured Light-based methods are most closely related to our approach and are reviewed in detail. A short derivation of the expected measurement errors is also provided.

Chapter 4 is concerned with the design and calibration of Structured Light systems. The first part states the general design guidelines. The second part shows how the color stripe patterns used in our 3D scanning system are designed. The third and largest part introduces the *Active Calibration* approach. It is validated with a number of experiments and proves to yield superior results compared to the 'classic' calibration approach. It can not only be applied to cameras but also to the whole Structured Light system.

In chapter 5 the proposed graph-based pattern decoding algorithm is introduced. This part of the work shows how a *region adjacency graph* is built from a superpixel segmentation of each input image, and how it can be used to decode the observed stripe pattern. An additional feature that is made possible by the graph-based ap-

proach is the so-called *color enhancement* step. In this step, consistency between the potentially noisy color estimates for each superpixel is enforced, which results in a large improvement of the decoding performance.

Finally in chapter 6 we evaluate the performance of the proposed system. This is done in several different ways. The noise component of the measurement error is evaluated on synthetic test scenes with ground truth. The calibration error of a physical scanner setup is estimated with the help of a calibrated reference object. The decoding performance is compared to other systems. Finally, also the runtime performance on different hardware is tested.

The last chapter concludes with a summary and an outlook for future work.

Chapter 2

Basics

2.1 Image Formation

The following sections briefly introduce important building blocks for two-dimensional imaging which are essential for the eventual depth recovery. They are digital image sensors, color representation, illumination, sources of noise and finally camera calibration and 3D reconstruction. The calibration models also apply to projection devices, which can often be viewed as a "reverse" camera for modeling purposes.

2.1.1 Sensors

The two main types of digital image sensors are Charge Coupled Device (CCD) and Complementary Metal Oxide Semiconductor (CMOS) chips. The spectral sensitivity depends on the type of semiconductor used in the sensor. The most common types use doped silicon and have a sensitivity in the visible and near infrared range. Alternatives are silver-doped ZnS for ultraviolet light or GaInAs for infrared light.

Silicon has a bandgap of 1.12eV, which corresponds to a maximum wavelength of approximately 1100nm. Longer wavelengths do not have enough energy to create photoelectrons in silicon. The advantages of CMOS over CCD are lower power consumption, better integration of electronics for processing, no blooming artifacts, random access to image data and higher speed. The basic difference between the technologies is that in CCD sensors the charge in each pixel is read out and digitized later. In a CMOS sensor each pixel contains its own analog-digital converter (ADC) and digitization happens before read-out. CMOS pixels have a more complex structure than CCD pixels and their main drawback is a lower image quality due to a combination of lower fill factor and lower full-well capacity. Bigas et al. [Biga 06] give an overview of the recent developments for CMOS technology and contrast it to CCDs.

Another factor for image quality are the widely varying penetration depths for different wavelengths, from about 1µm for blue light to over 1000µm for near-IR. Owing to their production process, in CCD chips the light-sensitive area is in a layer below the readout electrodes. In the standard *front illumination* variant, short wavelengths thus never reach the actual sensor, while long wavelengths are only partially absorbed. *Deep depletion* sensor models improve this absorption ratio and offer higher

sensitivity in the red spectral range. *Backside illumination* requires additional manufacturing steps, but offers higher sensitivity, especially for short wavelengths. Figure 2.1 shows the resulting quantum efficiencies for different sensor variants.

Both CMOS and CCD sensors can be used without mechanical shutters. Low-end CMOS devices often feature a rolling shutter, where one column of the image is read out while the others are still being exposed. This causes artifacts with moving scenes. In CCDs the readout is usually realized by *interline transfer*. The charge in each pixel is simultaneously transferred to a separate, light-shielded storage area and read out from there. This way a true global shutter can be implemented. The additional structures between the light-sensitive area reduce the fill factor, but this can be mitigated with microlens arrays focusing the incoming light onto the non-shielded spots.

Sensors based on quantum dots [Kons 09] are a new development. The photosensitive silicon in traditional CCD or CMOS chips is replaced by nano-structured PbS crystals with an artificial bandgap in a polymer matrix. These quantum dots offer greater quantum efficiency of up to 95% and their sensitivity can be tuned within a wide spectral range from the far infrared to ultraviolet. Another interesting, if still largely experimental, parallel development is the Single Pixel Camera [Duar 08]. It makes use of Coded Aperture Imaging and the wider framework of Compressive Sensing by recording the inner product of the scene with a number of random test functions realized in a Digital Micromirror Device (DMD). This way, it is possible to reconstruct images from a number of samples which is lower than the Nyquist limit. Maybe in the future devices like this will become practical and allow more efficient imaging. For now, however, CCDs are the most common type of sensor for high-quality imaging applications.

2.1.2 Color

Color imaging is realized either through multi-chip setups with tri-chroic beamsplitter prisms, or with a single chip and color filter arrays (CFAs) transmitting different wavelengths to different pixels. The latter are cheaper and have no alignment problems, but offer a reduced resolution and reduced sensitivity. A very common CFA is the Bayer pattern (see figure 2.2), which consists of separate filters for red, green and blue. Some modified designs also contain "white" (that is, transparent) or cyan filters instead of the second green pixel in each elementary cell. One alternative is the CYGM array with cyan, yellow, green and magenta filters. Figure 2.2 shows examplary spectral sensitivity curves for different CFA types. Since the filters transmission bands are partially overlapping, crosstalk can occur, which reduces the color contrast.

The design of filter arrays and the corresponding "demosaicing" algorithms still are an active area of research [Hao 10]. Common artifacts for naive linear interpolation include aliasing, color fringes and loss of high-frequency detail. More advanced methods work in frequency space or try to infer edges in one channel from the other channels. The rising pixel count of the sensor chips makes it also possible to generalize the filter concept and trade spatial resolution for other desirable information, like a higher dynamic range or plenoptic modeling [Ihrk 10].

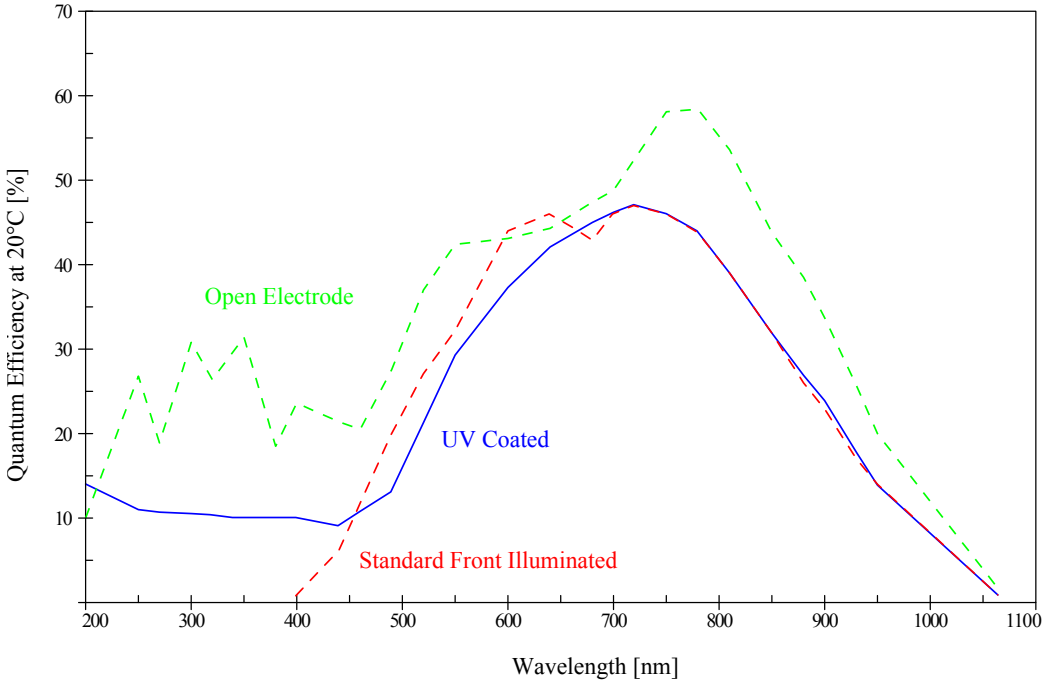
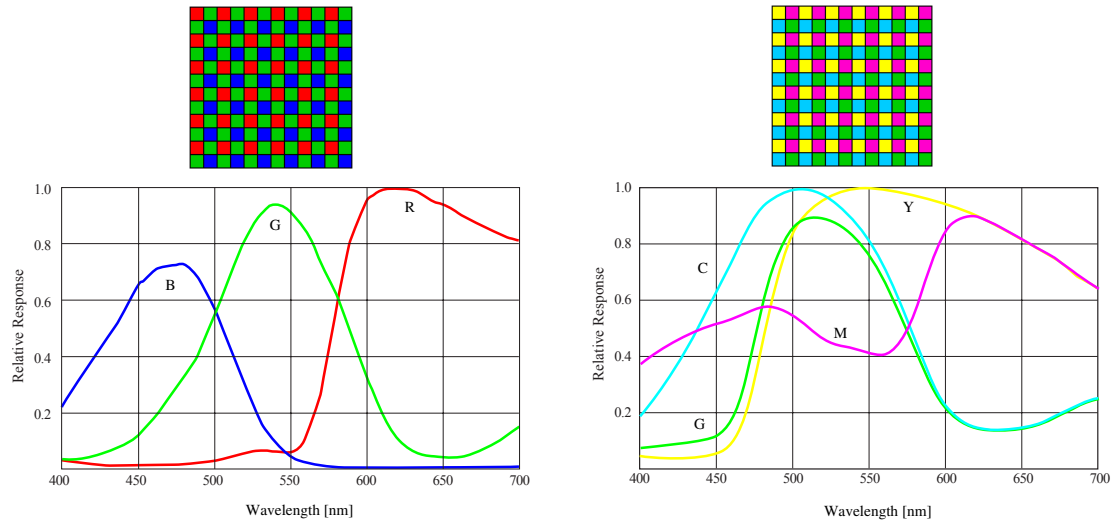


Figure 2.1: Quantum efficiency of a CCD chip for front-side illumination. Reproduced with adaptions from [e2v 03]. UV coating converts incident UV light to visible wavelengths with a larger penetration depth and thus increases the sensitivity in the UV range. Open Electrode sensors leave gaps in the electrodes covering the chip, which results in a large effective sensor area. Backside illumination (not shown) without any obstructing electrodes yields even higher efficiencies of up to 90%.

A third possibilty is the Foveon type of sensor [Hube 04]. It makes use of the different absorption rates for different wavelengths in silicon. Three layers of pixels for different colors are stacked vertically. Blue light is absorbed at the top, while red light reaches to the bottom. This sensor design thus is the middle ground between three-chip solutions with a prism and single-chip solutions with filter arrays. While the full spatial resolution is conserved and no light is lost in filters, the resulting color contrast is lower than with the other sensor types because of substantial crosstalk.

The human eye has three different kinds of photoreceptor cells for color vision as well. Their relative degrees of excitation correspond to a certain color impression. In particular, different mixtures of incoming wavelengths can create the same color impression. Three "primary" colors suffice to create others by mixing. The International Commission on Illumination (CIE) conducted a series of experiments with a "standard observer" to define a standard color space. In the CIE 1931 color space, the "imaginary" primaries are called X, Y and Z. All other colors can be represented by their mixtures. Other color spaces are derived from the XYZ model. In RGB color spaces, the primaries are red, green and blue light. The exact choice of the primary colors is a technical matter. Most displays implement the sRGB color space, which cannot reproduce the full color gamut (shown in figure 2.3). The human eye is very sensitive in the green spectral range, as can be seen in the wide spacings of the wavelengths along the monochromatic locus in figure 2.3 between 480nm and 580nm.



(a) RGB Bayer Filter. Reproduced with adaptions from Sony ICX285AQ datasheet.

(b) CYGM filter array. Reproduced with adaptions from Sony ICX419AKL datasheet.

Figure 2.2: Color Filter Arrays and their spectral responses. The color filters do not separate the colors perfectly. The result is so-called crosstalk, where for example incoming red light can also give a response in the green channel.

The eye can actually see wavelengths of up to 800nm, but with a very low sensitivity, so the CIE decided to end the locus at 700nm.

The most common digital representation of RGB colors is the RGBA 32 bit format with 8 bits per component (red, green, blue and alpha) that can differentiate 16.7 million colors. Hue-Saturation-Lightness (HSL) is another popular color space. It is based on a polar coordinate system as opposed to the cartesian RGB. For some applications it is more appropriate, as it separates the concept of color into hue and brightness. More information about color theory can be found in [Lee 05].

2.1.3 Light sources

As the name implies, Structured Light systems need some form of illumination. There are three common types of light sources: Thermal light sources like halogen lamps, lasers or Light Emitting Diodes (LEDs). The have very different properties and impose different limitations on the system.

Thermal light sources used to be the standard. We show a short estimation of their performance. According to the Stefan-Boltzmann law, the radiant flux of a grey-body light source with absolute temperature T, emissivity ϵ and area A is

$$\Phi(A,T) = \frac{2\pi^5 k^4}{15c^2h^3} \cdot \epsilon A T^4 = \sigma \epsilon A T^4 \tag{2.1}$$

with Stefan's constant σ. The emitted spectrum can be calculated with Planck's law. The radiant flux is measured in W. A light source with an area A illuminating a solid angle Ω has a radiance of

$$L = \frac{\Phi}{A\Omega} \tag{2.2}$$

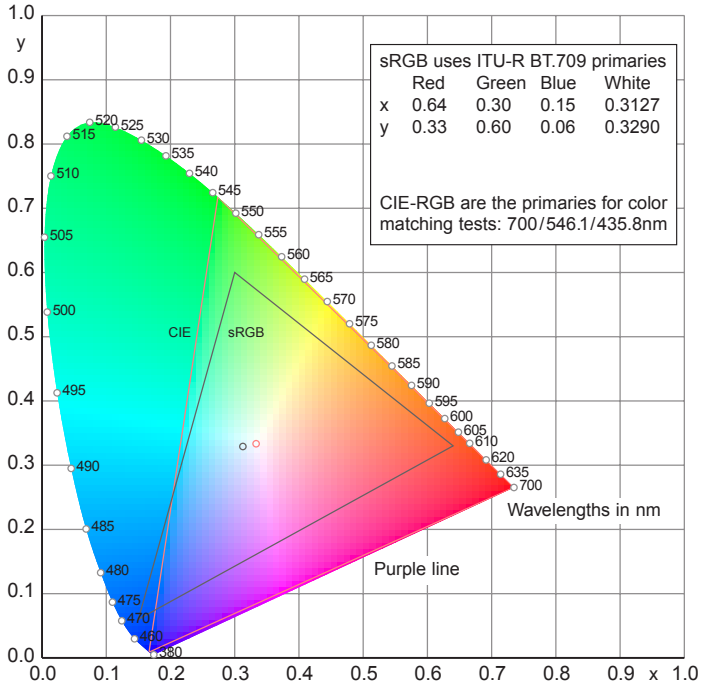


Figure 2.3: CIE RGB and sRGB color gamuts. The coordinates are the CIE xy chromaticity values. Not all colors can be reproduced by a given set of primaries. Reproduced with adaptions from [Hoff 08].

The radiance is measured in $W \cdot m^{-2} \cdot sr^{-1}$. Assuming an optical system with a spherical aperture of radius r at a distance d from the source, the irradiance is

$$E = L \cdot \frac{r^2 \pi}{d^2} \tag{2.3}$$

To see the limits imposed by the light source, consider a halogen lamp that is isotropic, has a temperature of about 3400K, a radius of 3mm and $\epsilon \approx 0.5$. According to equation 2.1 it then has a power of about 110W. A light guide with a radius of $\gamma = 3mm$ at a distance of $d = 5mm$ sees half the area of the source and picks up a power of about 5W. Additionally, only a fraction of the emitted spectrum is in a usable wavelength range. It is not possible to "compress" the light emitted in other directions into the light guide because of etendue conservation. Thermal light sources are therefore inefficient and require a lot of cooling. Since the required light power grows quadratically with the distance to the object, the practical range of a Structured Light system with a thermal light source is limited to a few meters. Additionally the lifetime of the bulbs is relatively limited, typically to a few thousand hours.

Laser sources can be extremely bright. They produce a collimated beam and are therefore more efficient than thermal sources. Furthermore they have a narrow-band spectrum, so it is easy to filter ambient light and the imaging system does not introduce chromatic aberrations. The disadvantage is that laser illumination on an optically rough surface produces speckles [Dain 84]. This makes it difficult to perform accurate spot or line detection.

In recent years, arrays of Light Emitting Diodes (LED) have been gaining a lot of popularity. They also offer quasi-monochromatic light. They are small and easy

to cool. They can be strobed. They require only a low voltage to operate and are therefore safe and easy to handle. LEDs have now become bright enough to replace other light sources in many applications, but are not available at all wavelengths.

2.1.4 Noise

An important concern for image quality is camera noise. There are different sources of noise. A systematic error is the fixed pattern noise. It is caused by a non-uniform response of different pixels to light. The camera manufacturers typically perform a flat-field correction to reduce or eliminate it. Another source is the read-out noise of the electronics used to convert the analog charge in the pixels to a digital value. It is typically in the range of a few electrons. The next source is the dark current. It occurs when high-energy electrons jump the band gap of the semiconductor without being excited by a photon. The number of electrons with sufficient energy is governed by the Maxwell distribution and depends strongly on the temperature. It approximately doubles for every 7K [Biga 06]. It may therefore be necessary to cool the sensor for better performance. The last major source of noise is the so-called shot noise. It is a consequence of the quantum nature of light. The number of photoelectrons generated by N photons arriving at a detector exhibits a Poisson distribution. The signal-to-noise ratio is then

$$\sigma = \frac{N}{\sqrt{N}} = \sqrt{N} \tag{2.4}$$

Shot noise is especially critical in low-light imaging. To increase N one can increase the illumination brightness or the exposure time. An example shows the physical limits of a Structured Light system. We assume a light source with a brightness rating of 200lm, which is typical for a small portable projector. By the definition of the unit lm, this is equivalent to $p = 0.3W$ at a wavelength of $\lambda = 540nm$. A single photon carries an energy of $E = \frac{\lambda}{hc}$. We can therefore assume a rate of $R_0 = \frac{p}{E} = 8 \cdot 10^{17}$ photons per second. Let the illuminated area on the object be $A = 0.1m^2$. Let the camera have a pixel size of $s = 8\mu m$, a lens with a focal length of $f = 10mm$ and an f-number of 8. The diameter of the entrance pupil is therefore $d = \frac{f}{8} = 1.25mm$. The spot size of a single pixel on a Lambertian object surface at distance g is $s' = \frac{sg}{f}$. This spot receives $R_1 = \frac{R_0 s'^2}{A}$ photons per second. We assume half are absorbed and a fraction $a = 0.5$ of them are scattered into a solid angle of 2π. The lens covers a solid angle of $\frac{d^2\pi}{g^2}$ and therefore receives

$$R_2 = \frac{R_1 a}{2\pi} \cdot \frac{d^2\pi}{g^2} = \frac{R_0 a s^2 g^2 d^2}{2f^2 g^2 A} = \frac{R_0 a s^2 d^2}{2f^2 A} \tag{2.5}$$

photons per second. The pixel has a fill factor of $b = 0.9$ and converts the captured photons to electrons with a quantum efficiency of $\epsilon = 0.6$ at a rate of $R_3 = R_2 b\epsilon$. Assuming a typical full well capacity of $C = 1.8 \cdot 10^4$ electrons (Sony ICX285AQ), it takes

$$t = \frac{C}{R_3} \approx 17ms \tag{2.6}$$

until the pixel is fully exposed. With additional ambient light the time is even shorter. The signal-to-noise ratio (due only to shot noise) of such a fully exposed pixel is

$$\sigma_{full} = \frac{C}{\sqrt{C}} = 134 : 1 \tag{2.7}$$

In practice saturation of the pixels needs to be avoided, so the signal to noise ratio is even lower. Note that for the given noise level it suffices to quantize the charge with 8 bits, or 256 intensity levels. The read-out noise alone is typically low enough that 12 bit digitalization makes sense, but with the shot noise in a standard sensor, an image can be stored in an 8 bit format without losing significant information. In low-light conditions, however, using more bits is useful, because here the shot noise is relatively low compared to the quantization steps computed from the full-well capacity.

2.2 Calibration

A calibrated camera makes it possible to relate measurements in images to metric quantities in the world. The calibration process is therefore fundamental and essential to computer vision tasks that involve image based metric measurements.

2.2.1 Coordinate systems

We first define some coordinate systems that will be used in all calibration related tasks. They are also illustrated in figures 2.4 and 2.5.

- The world coordinate system is three-dimensional and right-handed. It is typically defined by a calibration body.

- The camera coordinate system is also a 3D right-handed system. It has its origin in the camera's projection center. The x-axis is parallel to the image plane and points "right". The y-axis is also parallel to the image plane and points "down". The z-axis is the cross of x and y. Ideally it is identical to the optical axis of the camera, that is the axis of rotational symmetry of the camera's optical path.

- The image coordinate system is two-dimensional. It has its origin in the upper left corner of the image. The x-axis points "right" and the y-axis points "down". Image coordinates are specified in pixels.

- Sensor coordinates are similar to image coordinates, except the origin is shifted to the principal point (the intersection of the optical axis with the image plane) and they are expressed in metric units. Conversely the sensor coordinates are camera coordinates in the $z = f$ plane.

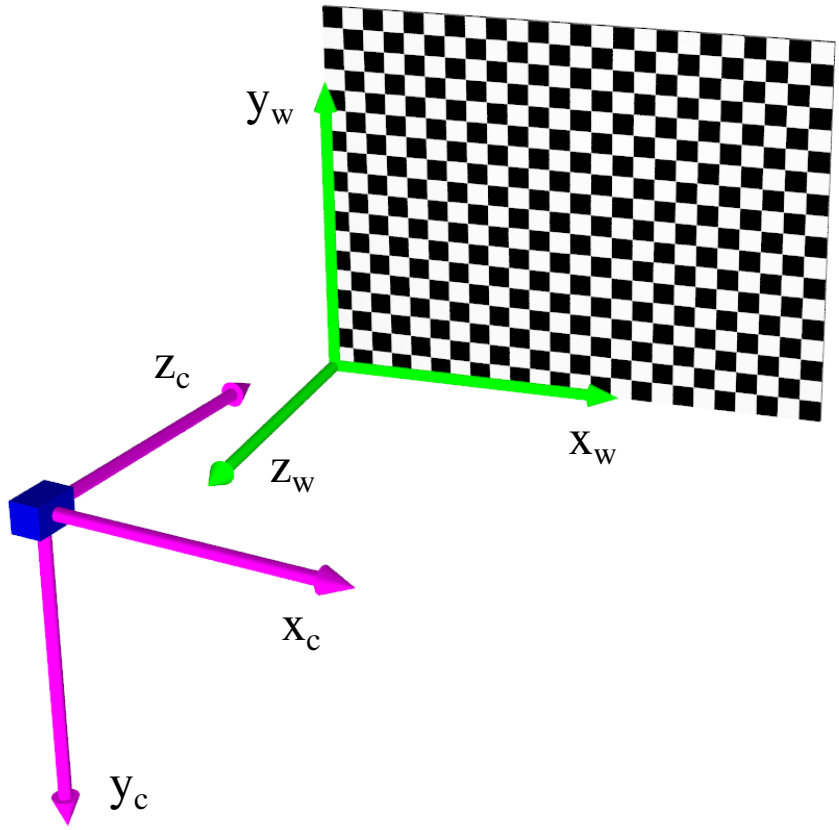


Figure 2.4: The world coordinate system (green) is often defined by the calibration target. The camera coordinate system (magenta) is defined by the sensor chip in the camera.

2.2.2 The pinhole camera

A result of a camera calibration can be expressed as a set of parameters. They are divided into two groups. The external calibration refers to the *camera pose* relative to a target. It maps the fixed world coordinate system to the camera coordinate system. The camera pose can be described by 6 parameters. The 3 rotation parameters are typically expressed as a rotation matrix $\mathbf{R}$ and the 3 translation parameters give the translation vector T. The second group are the intrinsic parameters. Their number and meaning depends on the camera model that is used. For the ideal pinhole camera, they are the principal point (u_0, v_0) and the focal length f. In homogenous coordinates the perspective projection mapping the point $[X_w, Y_w, Z_w, W_w]^T$ in the world coordinate system to the point $[u, v]$ in the image coordinate system can be written as

$$\begin{bmatrix} u \\ v \end{bmatrix} \backsimeq \begin{bmatrix} x_i \\ y_i \\ w_i \end{bmatrix} = \underbrace{\mathbf{A}\mathbf{P}_0 \begin{bmatrix} \mathbf{R} & T \\ 0 & 1 \end{bmatrix}}_{\mathbf{P}} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ W_w \end{bmatrix} \tag{2.8}$$

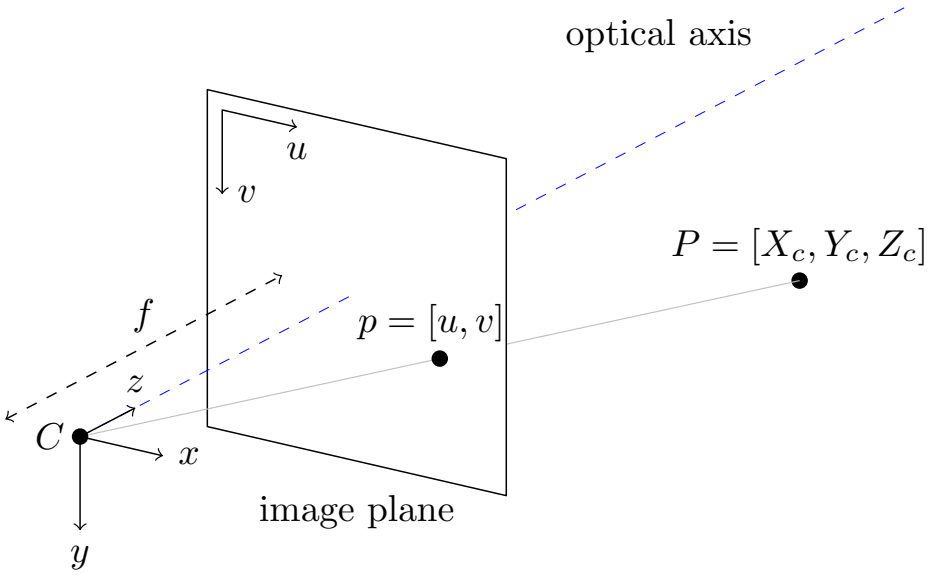


Figure 2.5: Perspective projection of point P in space to point p on the image plane. In a real camera the image plane is "behind" the projection center and the image is inverted.

$$\mathbf{A} = \begin{bmatrix} d_x & 0 & u_0 \\ 0 & d_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{P}_0 = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

with the camera internal matrix $\mathbf{A}$, the projection matrix P_0 and the coordinate transformation from world to camera coordinates $[\mathbf{R}|T]$. Their product $\mathbf{P}$ is also called the camera projection matrix. The parameters $d_x$and d_y are the pixel pitch in x and y direction. They are used to convert the metric sensor coordinates to image coordinates in pixels. Note that some authors introduce a skew parameter for the case of non-orthogonal axes in the sensor coordinate system. This used to be a concern with analog cameras, where the film transport could be uneven. For modern digital camera the skew is negligible. Given a camera matrix it is also possible to decompose it into the constituents [Truc 98]. More details about camera models can be found for example in [Faug 01, Hart 03].

2.2.3 Image distortion

The pinhole model is mathematically attractive, but the pinhole is necessarily very small, so very little light is admitted and the exposure time is very long. Real cameras therefore use lenses of a larger diameter to focus the incoming light. However, lenses are never perfect and exhibit different kinds of aberrations. One is chromatic aberration, which is due to the varying refractive index of the lenses for different wavelengths. There are so-called achromatic lenses which reduce the effect and correct it perfectly for two particular wavelengths, but it can never be eliminated completely for all wavelengths.

There are also five types of monochromatic aberrations, collectively known as Seidel aberrations, as they were first described by Ludwig von Seidel in 1857. They

are spherical aberration, coma, astigmatism, field curvature and geometric distortion. The first four effects degrade the point spread function (PSF) and thus the Modulation Transfer Function (MTF) of the optical system. This leads to blurred images. However, high quality lenses minimize these defects so that they can often be neglected.

Distortion, however, cannot be ignored, especially for wide-angle lenses. It causes straight lines to appear curved in the image. The distortion can further be split into so-called radial and tangential parts. To correct them, the pinhole model is extended with additional parameters. This augmented pinhole model is the most common camera model in use. Tsai presented a relatively simple variant [Tsai 92] with only two parameters for radial distortion. Heikkilä [Heik 97] uses two parameters for radial distortion and two for tangential distortion. The model (and calibration algorithm) proposed by Zhang [Zhan 00] is very popular, especially because it is available in the widely used OpenCV library [Brad 08] and as a Matlab toolbox [Boug 08]. These implementations support up to five parameters, three (k_1, k_2, k_3) for radial and two (p_1, p_2) for tangential distortion. Zhang's model maps undistorted image coordinates $[u_u, v_u]$ to distorted image coordinates $[u_d, v_d]$ via

$$u_d = u_u \cdot \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) + 2p_1 u_u v_u + p_2 \left(r^2 + 2u_u^2\right) \tag{2.9}$$

$$v_d = v_u \cdot \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) + 2p_2 u_u v_u + p_1 \left(r^2 + 2v_u^2\right) \tag{2.10}$$

where $u_u = u - u_0$, $v_u = v - v_0$ and $r^2 = u_u^2 + v_u^2$.

These models assume that the principal point is also the center of distortion and that all rays pass through the pinhole. This holds for typical applications with limited distortion, but not always [Will 93, Hart 07]. For wide-angle lenses, the pinhole model with the polynomial distortion does not accurately describe the imaging process. Other distortion models aim to remedy this [Kann 06, Clau 05, Deve 01], but they do not work for all types of cameras either.

2.2.4 Calibration algorithms

A camera model is only useful if there exists a robust and accurate way of determining its parameters. Many different algorithms for camera calibration have been developed [Remo 06]. So-called self-calibration methods do not assume any knowledge about the scene [Hema 03]. They are for example very useful for autonomous navigation. Metrology applications aim for highest accuracy and typically use dedicated calibration targets with well-known marks. Some methods use three-dimensional targets [Heik 00] but planar targets are more common as they are easier to build and handle. In 4.1 we propose the use of digital displays that can be found on everybody's desk.

The actual input data to perform a camera calibration are lists of correspondences between points on the calibration target (in world coordinates) and their image coordinates. Common calibration patterns are planar checkerboards. The corners of the checkers are the fiducial marks. They are typically localized by intersecting lines fitted to the sides of the checkers or by looking for a saddle point in the gradient. An alternative target type consists of an array of circular dots. The

centers of the dots are commonly computed via centroid methods, ellipse fitting to the contours, or deformable templates. However, for oblique viewing directions, the detected ellipse center is not the projection of the original circle's center and has to be corrected [Heik 97]. Image distortion also reduces the localization accuracy for dots [Mall 07]. For the usual planar calibration targets, measurements at several different poses of the camera relative to the target are required. From the correspondences between the markers' world coordinates and image coordinates a calibration algorithm [Tsai 92, Heik 97, Zhan 00] calculates the internal camera parameters and the coordinate transformation for each pose.

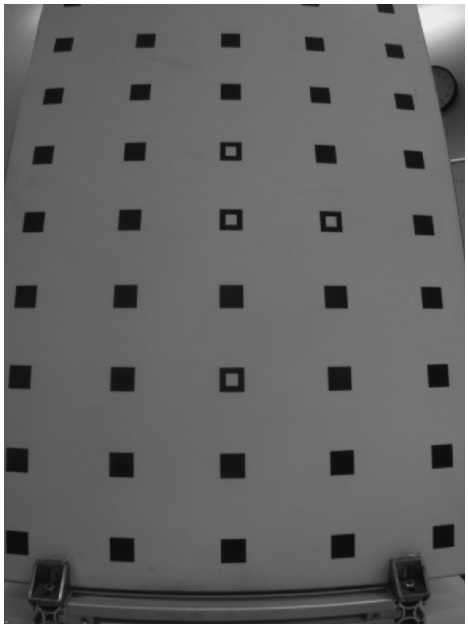


Figure 2.6: Checkerboard calibration target. The hollow squares can be used for determining the orientation and for identifying the marks automatically.

There is little comprehensive information in the literature about achievable feature localization accuracy. Shortis et al. [Shor 94] tested different algorithms for circular marks. They reported errors in the range of a few hundredths of a pixel, but did not include noise in the analysis. Heikkilä [Heik 00] shows lighting-dependent shifts of up to 0.5 pixels in the location of circular marks. Mohr et al.[Mohr 94] found errors of around 0.1 pixels in corner localization. White and Schowengerdt [Whit 94] examine the effect of the point spread function on edge localization accuracy and find errors of up to 0.2 pixels. Mallon and Whelan [Mall 07] show errors around 0.1 pixels for circular marks (without distortion bias) and up to 0.03 pixels for a checkerboard target. Chen and Zhang [Chen 05] give errors of about 0.05 pixels for checkerboard corner localization.

The final calibration errors are in the same range. Heikkilä [Heik 00] claims that an accuracy of 0.02 pixels is a realistic goal. He achieves it for synthetic images and reports 0.061 pixels on real images. Douxchamps and Chihara [Doux 08] even reach 0.0065 pixels on synthetic images and 0.045 on real images. However, in his widely known paper [Zhan 00], Zhang gives an RMS reprojection error of about 0.3 pixels. Albarelli et al. [Alba 09] achieve an error of 0.23 pixels up front but reduce it to 0.089

by additional bundle adjustment [Trig 00], as they assume imperfect knowledge of the target. Bundle adjustment is a non-linear minimization of the back-projection error which changes the camera parameters as well as the mark coordinates. Fiala and Shu [Fial 10] also reach values of around 0.2 pixels. The differences between these figures might be due to outlier removal steps, differences in image and target quality, or simply different pixel sizes and different lenses. An RMS reprojection error of 0.05 pixels seems to be a lower bound for a very careful calibration in an optimal environment, while errors up to 0.3 pixels are acceptable in everyday calibrations.

2.2.5 Catadioptric cameras

There are very useful extreme wide angle cameras based on fish-eye lenses or curved mirrors. The latter are also known as catadioptric cameras. This kind of camera does in general not conform to the pinhole model. This is because the effective viewpoint depends on the viewing direction (see figure 2.7). Only special setups of a perspective lens with a hyperbolic mirror or a telecentric lens with a parabolic mirror can be treated like pinhole cameras, because they do have a single effective viewpoint. However, any misalignment of camera and mirror destroys this property. In the general case, catadioptric cameras need to be calibrated with non-parametric camera models that simply store the ray of view for every single pixel.

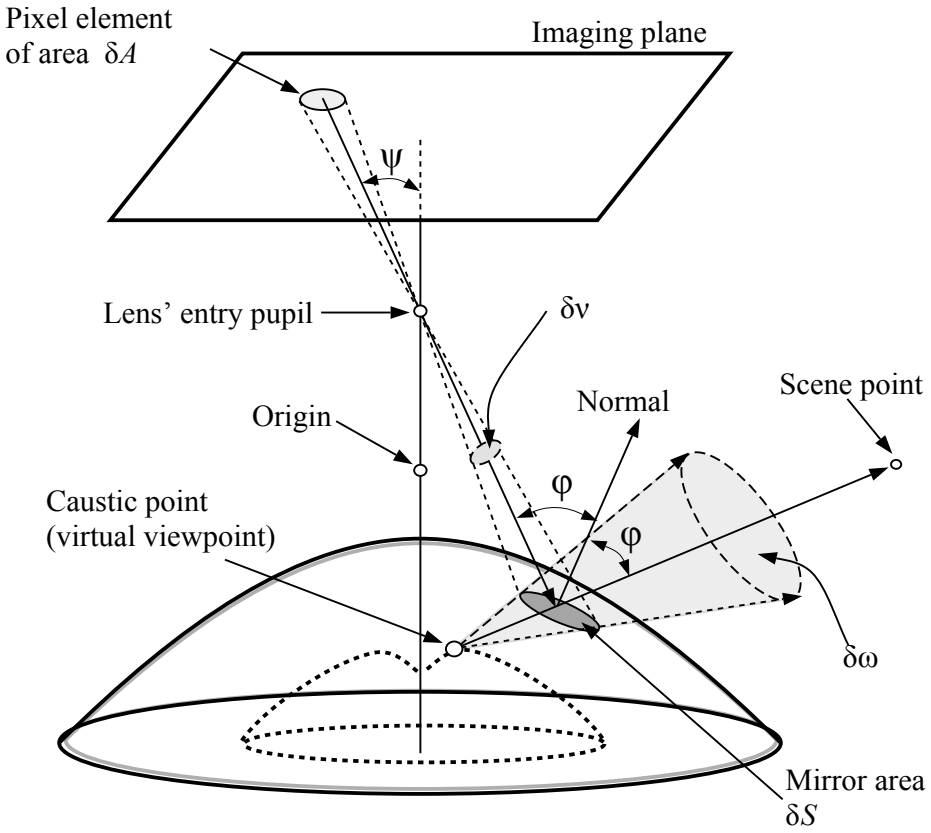


Figure 2.7: Non-single-viewpoint catadioptric camera. The dotted surface is the caustic surface formed by the different effective viewpoints of the camera. Reproduced from [Swam 02] with adaptions.

Calibration algorithms for general non-single-viewpoint cameras can be found in [Rama 05, Gros 01]. So-called axial cameras have the property that all viewpoints lie on the optical axis, which can be utilized as a constraint during calibration [Tard 09]. An important member of this subclass of catadioptric camera is the combination of a perspective camera with a spherical mirror. This camera type has practical advantages as spherical mirrors are relatively easy to manufacture and invariant to rotations, which reduces the effect of misalignments during assembly.

The generic camera models can in principle also be applied to narrow-angle imaging. However, in that regime they offer a lower level of accuracy [Dunn 07] than the regular augmented pinhole model.

2.2.6 Two-View Geometry

A Structured Light system consists of a camera and a projector. For purposes of calibration, a projector can be considered a reverse camera, so there is no difference between Stereo and Structured Light in this regard. Both work with two different perspectives on the same scene. The geometric relationships between two views of a scene are captured by the epipolar geometry. It can be computed from matches between corresponding points in both images. We give a short overview following [Hart 03] and [Faug 01]. It has to be noted that this treatment is only valid for perfect pinhole cameras without distortion. If distortion is present in the images, it has to be corrected first.

The basic setup is illustrated in figure 2.8. The baseline b connects the two camera projection centers C_A and C_B. The projection ray from a point X in space to C_A together with the baseline defines an epipolar plane Π_E. The projection ray from X to C_B must also lie in this plane. The intersection of the epipolar plane Π_E with the image plane of camera B yields the epipolar line l_{BX}. Conversely, the intersection of the epipolar plane with the image plane of camera A yields the epipolar line l_{AX}. The line l_{BX} is also the projection of the camera ray $\overline{C_A X}$ onto the image plane of camera B. Since X by definition lies on that camera ray, its image in camera B must lie on the epipolar line l_{BX}. This is very helpful for stereo problems, as the correspondence search for a given point x_A in image A can be restricted to the corresponding epipolar line l_{BX} in image B. The epipoles e_A and e_B are the intersections of the image planes and the baseline. Conversely, they are the projections of one camera center onto the image plane of the other camera. These intersection points are often outside of the actual image and can even be at infinity. The set of all epipolar planes is a pencil of planes around the baseline.

For the remainder of this section we use homogeneous coordinates. In this notation a line l through points m_1 and m_2 is expressed as $l \backsimeq m_1 \times m_2$ with the projective equality sign $\backsimeq$. Conversely, the intersection point m of two lines l_1 and l_2 is $m \backsimeq l_1 \times l_2$. Let $\mathbf{P}_A$ and $\mathbf{P}_B$ be the projection matrices for cameras A and B with non-identical projection centers C_A and C_B. Then the relations $\mathbf{P}_A C_A = 0$ and $\mathbf{P}_B C_B = 0$ hold. Let $\mathbf{P}_A^+$ and $\mathbf{P}_B^+$ be inverse projection matrices. Given the image point x_A we can write

$$x_B \backsimeq \mathbf{P}_B X \backsimeq \mathbf{P}_B \mathbf{P}_A^+ x_A \tag{2.11}$$

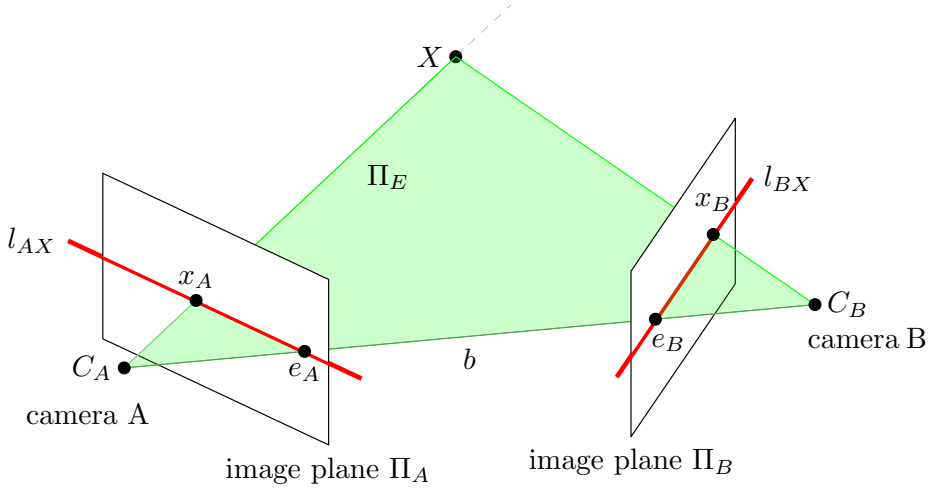


Figure 2.8: Epipolar geometry. The point X together with the centers of projection of the two cameras spans the epipolar plane Π_E..

Furthermore we know that the epipole e_B is the image of the optical center C_A on the image plane of camera B, that is

$$e_B \backsimeq \mathbf{P}_B C_A \tag{2.12}$$

The epipolar line l_{BX} passes through both of these points. Expressing the cross product as a skew-symmetric matrix $[\,]_\times$ we can write

$$l_{BX} \backsimeq e_B \times x_B = \underbrace{[\mathbf{P}_B C_A]_\times \mathbf{P}_B \mathbf{P}_A^+}_{\mathbf{F}} x_A \tag{2.13}$$

The fundamental matrix $\mathbf{F}$ describes the relationship between an image point x_A and its corresponding epipolar line l_{BX}. Since the point x_B lies on l_{BX} by definition, it follows that

$$x_B^T \mathbf{F} x_A = 0 \tag{2.14}$$

The fundamental matrix has rank 2. It can be computed from at least 7 corresponding point pairs in the two images. Details can be found in [Faug 01].

2.2.7 3D Reconstruction

Let two corresponding 2D image points x_A, x_B and the projection matrices $\mathbf{P}_A$, $\mathbf{P}_B$ be known. The 3D coordinates of X can then be recovered from the intersection of the camera rays. However, in practice the image coordinates are not known exactly, therefore typically the two conditions $x_A = \mathbf{P}_A X$ and $x_B = \mathbf{P}_B X$ cannot both be met exactly. A simple solution is to use the midpoint of the shortest line connecting the camera rays. Another possibility is to find the minimal correction to the image coordinates so that both equations can be satisfied. This approach is called optimal triangulation [Kana 08]. An estimation of the typical depth reconstruction errors can be found in section 3.4.3.

In Structured Light the situation is slightly different, as it does not use two cameras but one camera and one projector. For a stripe projector typically only one component of the image coordinates is known. This line in the projector image defines a plane in space, which again can be intersected with the camera ray. Parameterize the light plane illuminating the sought-after point m as

$$n \cdot m - \kappa = 0 \tag{2.15}$$

in a camera-centered coordinate system with the normal vector n and the distance κ to the origin. The operator $\cdot$ in this context is the dot product of two vectors. In the pinhole model, the equation of the camera ray for the pixel coordinates (u, v) from the origin 0 through m is

$$m = 0 + \lambda v = \lambda \begin{bmatrix} (u - u_0)\, d_x \\ (v - v_0)\, d_y \\ f \end{bmatrix} \tag{2.16}$$

with the pixel pitch (d_x, d_y), the focal length f, the principal point (u_0, v_0) and the free parameter λ. Plugging eq. 2.16 into eq. 2.15 and eliminating λ, the point m in camera coordinates is

$$m = \frac{\kappa}{n \cdot v}\, v \tag{2.17}$$

Avoiding the nonlinear division operation and plugging in the definition of v, this can be written in matrix form using homogeneous coordinates as

$$\begin{bmatrix} X_m \\ Y_m \\ Z_m \\ W_m \end{bmatrix} = \mathbf{A} \begin{bmatrix} \kappa d_x & 0 & 0 \\ 0 & \kappa d_y & 0 \\ 0 & 0 & \kappa f \\ n_x d_x & n_y d_y & n_z f \end{bmatrix} \begin{bmatrix} u - u_0 \\ v - v_0 \\ 1 \end{bmatrix} = \mathbf{AD} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \tag{2.18}$$

with the components of the light plane normal vector $n = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix}$ and a general 4×4 coordinate system transformation matrix $\mathbf{A}$. The matrix $\mathbf{AD}$ for each light plane can be cached and a 3D point in any coordinate system can be reconstructed with only one matrix multiplication.

2.3 Edge Detection and Localization

The detection and localization of edges is a fundamental task in digital image processing. The reliability and accuracy that can be achieved are of particular importance for Structured Light 3D scanning as well as for stereo systems. On one hand, calibration marks are typically located with the help of edges (a method without edge detection is proposed in section 4.3.1). On the other hand, the final 3D data in the proposed stripe projection system is computed from the observed stripe position in the image, which is determined with an edge detection filter. Stereo systems often depend on correspondences between edge points.

In the simplest definition, edges are extrema in the gradient image or correspondingly zero crossings of the second derivative. The most basic methods to detect them

are linear gradient filters that can be realized as simple convolution masks [Osko 10]. The most widely used of these is probably the Sobel filter, but there are also other examples like the Roberts, Kirsch and Prewitt operators. Each exists in several orientation variants. Scharr [Scha 00] proposed a variant of the Sobel kernel that gives better results on oblique edges. Torre and Poggio [Torr 84] made it clear that differentiation should always be preceded by a regularization step, that is, smoothing. Marr and Hildreth [Marr 80] proposed an isotropic Laplace-of-Gaussian (LoG) filter, inspired by physiological findings about the human vision system. It combines the second derivative Laplace operator with a Gaussian smoothing step. One characteristic of the Laplace filter is that the detected edges always form closed contours. Savitzky and Golay [Savi 64, Gorr 90] noted that a local least squares polynomial fit for smoothing and differentiation can be combined in one convolution operation. Shen and Castan [Shen 92] introduced the Infinite Symmetric Exponential Filter (ISEF). An overview of these approaches can also be found in [Pinh 97].

However, in the sense of higher level vision, not all extrema of the gradient image are edges. Many are due to noise or shading variations and do not reflect properties of the underlying scene. Therefore more advanced processing is necessary. The edge detector popularized by Canny [Cann 87] additionaly performs non-maxima suppression and hysteresis thresholding on the gradient image to form a binary result. Multi-scale approaches [Witk 84, Lind 90] can also be applied to filter out the most important edges.

In many applications, and in Structured Light in particular, it is desirable to localize edges with subpixel precision. One possibility are integral operators based on moments [Taba 84, Ying 05], but they need a large support region. Haralick [Hara 84] proposed to use the zero-crossing of the second derivative for subpixel localization. Blais and Rioux [Blai 86] presented formulas for edge localization based on linear interpolation of the gradient with different support regions (BR2, BR4 and BR8). Other methods based on the first derivative are parabolic and Gaussian interpolation. Fisher [Fish 96] compares the different gradient-based approaches and recommends Gaussian interpolation or Blais-Rioux (BR4). However, the former is relatively slow because logarithms have to be computed. The latter needs seven instead of three samples, which can be a disadvantage if several edges are close to each other.

There are also many works on the bounds of edge localization accuracy under noisy conditions. Kakarala [Kaka 92] considered continuous signals and derived a lower bound that scales linearly with the blur kernel size and quadratically with the inverse of the signal-to-noise ratio. Laboureux [Labo 01] showed that the localization accuracy for a feature is proportional to the local curvature. Koplowitz and Greco [Kopl 94] derived the probability density function of the localization error and concluded that in high-noise conditions arbitrarily large errors are possible. White and Schowengerdt [Whit 94] examined the influence of the point spread function of the imaging system. They obtained optimal results for a blur radius of 0.5 to 0.9 pixels, with localization errors of up to 0.2 pixels. Rockett [Rock 99] specifically tested the Canny edge detector with parabolic interpolation and found errors of a few hundredths of a pixel. Mikulastik [Miku 08] examined the parabolic estimation on a synthetic edge with gaussian blur. For the variants using five and seven samples,

they found systematic errors of up to 0.2 pixels. Only the three-sample parabolic interpolation is reasonably unbiased.

2.4 Image Segmentation

The proposed algorithm for decoding color stripe patterns makes use of a presegmentation step to reduce image complexity. Segmentation in general is a hard problem. It can be defined as the process of partitioning an image into disjoint and homogeneous regions. However, homogeneous in respect to which criterion? This depends on the problem. It could be brightness, color, texture or even semantic categories like buildings and vegetation. Consequently, many segmentation algorithms have been proposed. Some possible classifications are supervised vs. unsupervised methods, binary vs. multi-class, or feature domain vs. image domain. Surveys of segmentation methods for the important special case of color images have been done by Pal and Pal [Pal 93] and by Lucchese and Mitra [Lucc 01]. Noble [Nobl 06] gives an overview of segmentation methods with special regard for the notoriously difficult area of grayscale ultrasound image segmentation. In this section we highlight a few important techniques.

In the feature domain Otsu [Otsu 75] assumed a gray-level image with a bi-modal histogram and computed the optimal threshold value to perform a binary segmentation. For multi-channel images, clustering algorithms like k-means [Kanu 02] or mixture models [Figu 02] can be used. In unsupervised operation, there is an initialization problem: the cluster centers and the number of clusters are not known a priori. In that case it is possible to start with randomly initialized clusters and adapt the number until an optimum in the sense of information theory is reached [Pell 00]. Mean shift clustering [Coma 02] traces the density of points in the feature space and does not suffer from the initialization problem.

In the image domain so-called Split&Merge techniques [Wu 93] can be applied. The image is subdivided until each region satisfies the homogeneity criterion, then similar neighboring regions are merged. Seeded region growing [Mehn 97] adds neighboring pixels to predefined seed regions in the order of similarity. Regions are continuous, so in the final steps even dissimilar but "engulfed" unassigned pixels may be added to a region. There are boundary-based algorithms which use edge detectors as a preprocessing step. The Laplace filter has the convenient property that its zero-crossings always result in closed contours. This is not the case for edges computed with the Sobel filter. Level set methods [Seth 99] and Active Contours [Case 97] evolve a boundary according to some energy functional, which typically includes a data term and a curvature term.

Another family of algorithms interprets images as graphs and segmentation as a partitioning task. Each pixel becomes a node. The edge weights of the graph depend on the distances of neighboring pixels in some feature space. In the simplest case this is the intensity difference. Markov Random Fields [Pere 98] are frequently used. Each node can be in one of several states. There is a cost for each state and cost for each combination of neighboring states. The challenge is to find the global minimum of this cost function, which is an NP-hard problem. Szeliski et al. [Szel 07] compared different approximate optimization methods. Belief propagation [Yedi 03]

and maximum flow [Boyk 04] both deliver good results with acceptable computation time. Segmentation using explicit graph cuts was popularized by Shi and Malik [Shi 97] with the Normalized Cuts algorithm. It avoids the "small cuts" bias of other graph cut algorithms by normalizing the cut cost with the total weight of all edges in the cut region. However, the resulting problem is also NP-hard. Although the authors make approximations, the computation time is still high. Felzenszwalb and Huttenlocher presented an approach based on minimum spanning trees [Felz 04b]. Grady [Grad 06] introduced the random walker algorithm, which has a closed form solution and can handle non-binary segmentations. It is interesting to note that the graph cut, random walker and watershed segmentation algorithms have been unified in the so-called power watershed framework [Coup 10]. The watershed segmentation will be described in greater detail in the next subsection.

2.4.1 Watershed Segmentation

The watershed segmentation has the advantages that it is fast, unsupervised and parameter free. Therefore it is a good choice as a preprocessing step in a real-time 3D scanning system. The details of this application are presented in section 5.1. Here we give a more general overview.

The watershed segmentation is formally defined as a morphological transform [Roer 00]. The underlying idea is very intuitive. Rain falls on a landscape whose heightmap is given by a grayscale image, typically the magnitude of the gradient of the image to be segmented. The watersheds are the dividing lines between different domains of attraction for the water, also called basins. Next to the rainfalling implementations [Stoe 00] there are also immersion-type implementations [Vinc 91], where the water seeps in from below. The former is also called tobogganing and is typically faster [Lin 06]. A watershed transform of a non-synthetic, noisy image typically produces severe oversegmentation. There are many shallow basins. Most perceptual regions in the image correspond to more than one basin. Postprocessing can correct this [Blea 00]. In other applications, like for example stereo [Zitn 07] or model search [Mori 05], the oversegmentation is performed on purpose as a preprocessing step. In that case, the resulting regions are called superpixels. They offer a simple way of reducing the image complexity and give a perceptually more meaningful representation than the simple pixels, which are viewed as artifacts of the imaging process. Next to the watershed transform, there are specialized segmentation algorithms [Radh 10, Levi 09] that produce superpixels.

2.5 GPU Programming

Real-time performance is an important feature for a practical Structured Light decoding algorithm. Historically, the increase in processor speed tended to solve this issue automatically. However, there has been a shift in the development of computing hardware in the recent years. Individual processors hardly get faster any more, but their number in a given device increases. As an example, a 2010 Intel Nehalem CPU has six cores running at 2.8GHz. The same clock frequency was already reached by a single-core Pentium 4 in 2002. When designing a high-performance algorithm,

it is therefore necessary to keep in mind the aspect of parallelization so that the implemenation can benefit from the latest hardware developments.

Graphics processing units (GPU) are massively parallel devices with several hundreds of cores. The NVidia Tesla M2050 consists of 448 processors running at 575MHz. The individual so-called stream processors are throughput-oriented where traditional processors were mostly latency-oriented. They offer only little cache and control logic but emphasise arithmetic performance on workloads with inherent data-parallelism. The programming model to make efficient use of this computing power is called stream programming.

The current standard for many-core and heterogeneous computing is OpenCL [Tsuc 10] (not to be confused with OpenGL). Thus, we outline its most basic principles. Functions (called kernels) are executed by many threads in parallel. The threads are organized in so-called work-groups sharing a certain amount of memory. Additionally, there is thread-local and global memory. Accessing global memory is slow. This latency can be masked if multiple threads perform coalesced reads or writes, that is if they access consecutive global memory adresses at the same time. Synchronization between different threads and workgroups is a bottleneck and therefore discouraged. Divergent branches for threads within the same workgroup massively degrade performace. Optimizing memory access patterns and avoiding synchronization are the major parts of performance tuning for GPU algorithms.

Existing algorithms may have to be modified to reach maximum performance. Not all algorithms are suitable for a GPU implementation. However, it is advantageous to have the complete processing pipeline on the same device to avoid copying data back and forth over slow buses with a relatively low bandwidth. The transfer time can be a substantial part of the total processing time. The exact speedup factor that can be reached with a GPU implementation versus a CPU implementation of an algorithm depends on the problem type. The ratio of computations per memory word transferred is the arithmetic intensity. The higher it is, the more an algorithm stands to gain in performance on the GPU. Inherently parallel problems also naturally profit more than predominantly serial problems.

OpenCL code can, once written, be run on the CPU as well as on the GPU. However, the source code portability of OpenCL currently does not mean that optimal performance is automatically reached on all possible devices. Commonly reported speedup factors of GPU versus CPU are in the single-digit to three-digit range [Ryoo 08, Park 11]. Many image processing tasks are perfect for GPUs as the computations for each pixel can be performed independently of all others.

Chapter 3

State of the Art in Optical 3D Shape Acquisition

This chapter presents the state of the art in optical 3D imaging in general and Structured Light in particular. The techniques for acquiring the 3D shape of objects have improved greatly in the last 20 years. Early surveys were done by Jarvis [Jarv 83] and Besl [Besl 88], while Blais gives an overview of the more recent developments in [Blai 04].

There are many methods for 3D data acquisition and many ways to categorize them. Technical criteria are for example the difference between point, line, area or volume measurements or the surface types the sensors are suited for. Another possibility is to differentiate between active methods with dedicated illumination and passive ones without. The taxonomy in figure 3.1 is based on the underlying principle of the method. Keep in mind that we are only concerned with optical methods for recording the 3D surface shape of object. Direct tactile methods and imaging modalities like Ultrasound, Optical Coherence Tomography (OCT), Magnetic Resonance Imaging (MRI), Computed Tomography (CT), Single Photon Emission Computed Tomography (SPECT) and Positron Emission Tomography (PET) that produce volume data are ignored. Under this constraint the most common methods for practical applications are based either on runtime measurements (in interferometry) or direction measurements (in triangulation). Häusler and Leuchs [Haus 97] compared these two approaches with regard to their scaling behavior and surface type compatibility. In particular they note that the depth error δz of triangulation based methods is proportional to the square of the object distance z^2, while for interferometry it is independent of z. Triangulation-based systems are therefore best suited for close-range applications.

The next four sections introduce the state of the art in the four subgroups of optical range imaging techniques that have been identified in 3.1. Section 3.5 is special in that it contains an overview of methods proposed for endoscopic measurements, irrespective of their underlying principle. There is also a large body of work on the calibration of Structured Light systems. This topic is not covered here but in section 4.3. The typical postprocessing tasks of registration, meshing and stitching are outside of the scope of this work and are therefore not included in the overview.

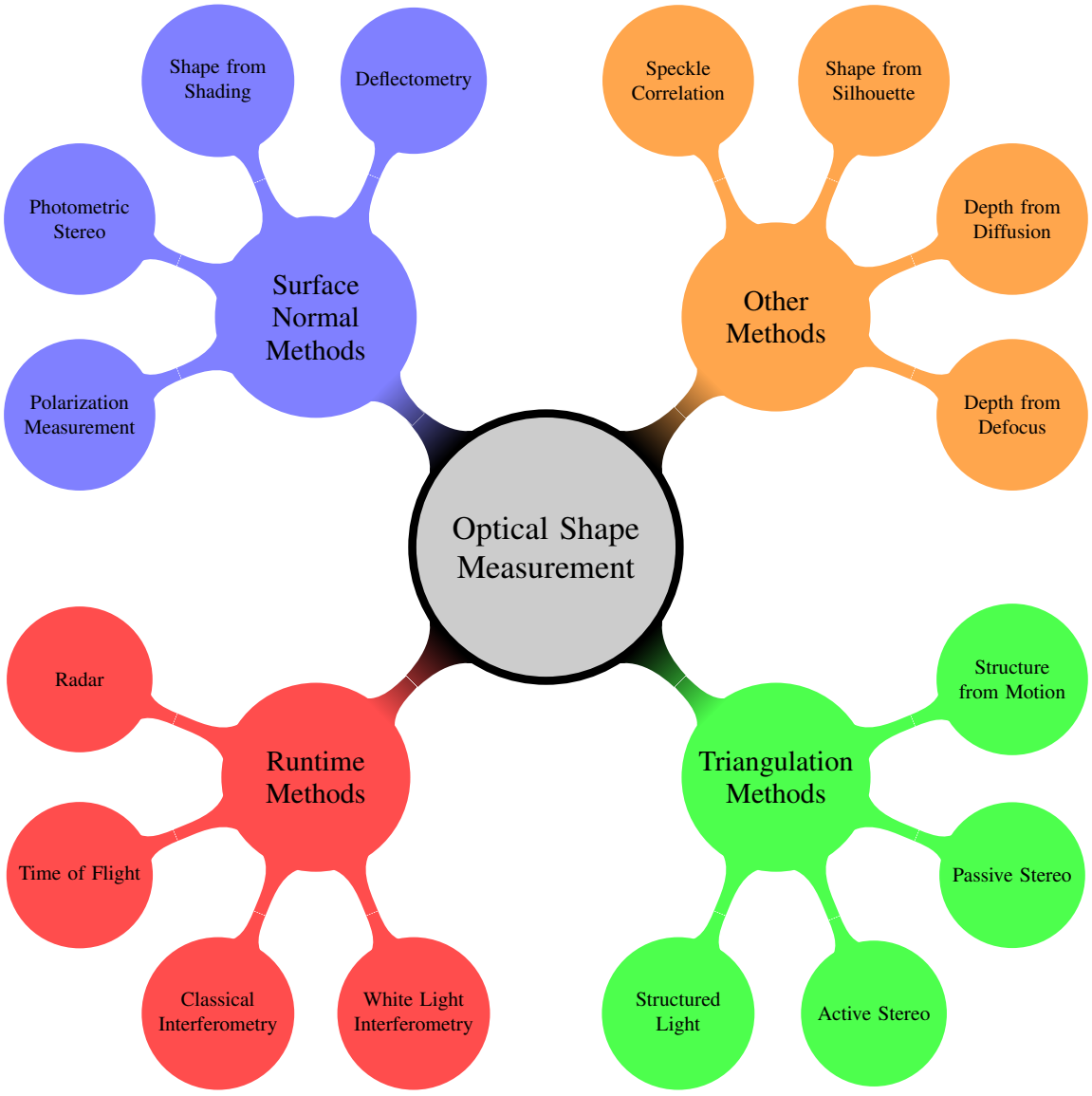


Figure 3.1: A simple taxonomy of optical 3D shape measurement techniques based on the underlying principle. Depth-from-Defocus can also be seen as an indirect triangulation method.

3.1 Runtime Measurement

Runtime based 3D acquisition methods measure the time it takes for light to travel from a controlled source to the object and come back to the sensor. This can be done directly for long-distance targets, as for example in pulsed radar. For close range objects, the better alternative is usually a phase measurement. *Classical Interferometry* [Hari 06] uses coherent illumination that is split into a reference wave and an object wave, which interfere in the detector. It can reach the extremely high precision of a thousandth of the used wavelength. However, because the phase is ambiguous, the working range is very small. Interferometers therefore are mostly set up to detect deviations from a reference instead of true 3D data. *White Light Interferometry* [Dres 92, Wyan 02] or *Speckle Interferometry* is a variation that employs a light source with finite coherence length. Fringes can only occur at a well-know distance, which solves the ambiguity, but the object (or the reference) has to be translated in a controlled way to get a full scan. White Light Interferometry also works on optically rough, diffuse surfaces. Since illumination and observation are coaxial, there are no occlusions. White Light Interferometry therefore can, for example, be used to examine boreholes. While the precision of interferometers is unsurpassed, their sensitivity means they must be built as stable as possible to counter external vibrations. This makes them expensive and not very robust. Therefore interferometry in general is a technique for close range measurements in controlled environments.

Time-of-Flight (ToF) based 3D cameras exist in two basic variants [Hebe 92]. One uses pulsed illumination and measures the runtime of the pulses. This is done with a very short exposure window [Meng 02]. A long-running pulse will partially miss the shutter and appear darker. A second pulse with a longer exposure window is used to correct for object reflectivity. Ambient light can be compensated by an "empty" exposure without an illuminating pulse. The other ToF variant uses amplitude-modulated continuous-wave illumination and measures the phase shift of the reflections. As in interferometry, this phase is not unique. The ambiguity interval for a modulation frequency of 25Mhz is 6m, but it can be extended using multiple different modulation frequencies. ToF has a number of advantages [Gokt 05]. There are no occlusion artifacts as the illumination source and the sensor are directly next to each other. Therefore ToF systems also can be very compact. The depth data can be calculated in each pixel independently. The illumination is typically in the NIR range and thus unobtrusive for humans. Assuming constant brightness, the measurement error is approximately independent of distance. However, constant brightness is hard to achieve as the intensity of the backscattered pulse drops quadratically with distance (from a diffuse surface). There are also other drawbacks [Kolb 08]. Multiple sensors can disturb each other. The lateral resolution is limited because of the relatively large amount of circuitry required for each pixel. These large pixels also cause artifacts in the form of "mixed pixels" at object borders. Calibration is very intricate. There are systematic depth errors up to several centimeters [Lind 08] and drift effects depending on temperature. There are also scene-dependent depth errors caused by interreflections, object texture and volume scattering. All in all, ToF is a promising technology for the meter range. For longer ranges the required light sources are not eye safe, for shorter ranges triangulation-based systems yield better performance.

3.2 Surface Normal Methods

Instead of measuring the shape of an object, an indirect approach is to measure its surface slope and calculate the shape by integration. *Shape-from-Shading* [Zhan 99] typically assumes a known reflectance, orthographic projection and a known lighting direction. Often an additional assumption of Lambertian reflectance is also used. It is then possible to reconstruct the shape of the object from a single image. Unfortunately even under these fairly restricted conditions the shape is not unique [Belh 99], so regularizing assumptions have to be made. Additional images with different lighting directions also help in finding the correct reconstruction.

Photometric Stereo is a technique of more general applicability. It requires multiple images from a fixed viewpoint, but with changing lighting conditions. Early formulations [Wood 80] assumed knowledge of the object's reflectance properties and the lighting directions, but later works have successfully relaxed these requirements [Basr 07, Bars 03, Geor 03]. For a Lambertian surface three images are needed to solve for two gradients and the albedo in each pixel. For uniformly colored objects, these three images can be combined into a single one with three different-colored light sources [Drew 00]. Helmholtz stereopsis [Zick 02] offers a way to measure any object, regardless of its reflectance properties, but it requires an exchange of camera and light source positions. Photometric stereo has also been combined with Structured Light [Neha 05] for impressive results. The latter yields low-resolution depth values which are used as anchors for the integration of the high-resolution normals produced by the former.

Taking a different approach, Winkelbach [Wink 02] directly computed the slope of the object surface from the observed angle and width of projected stripes. For smooth dielectric surfaces it is also possible to recover the surface normal using polarization imaging [Atki 06]. The lighting direction does not have to be known, but again, this works only for simple objects.

All methods presented so far have difficulties with specular surfaces. In contrast, *Deflectometry* [Knau 04] is designed especially for this class of objects. The underlying problem is that a specular surface itself is "invisible" - it only mirrors the environment. Hence, the solution is to control the environment. Instead of a point light source a large screen is used. Depending on the local slope of the surface, a different part of the screen is observed by the camera. The method is extremely sensitive and can detect local height changes of a few tens of nanometers, which is a precision typically only reached by interferometers. Furthermore, in typical applications like lens testing, the quantity of interest is the local refractive power, which is related to the curvature. To get it from an interferometric measurement, one has to differentiate twice, which amplifies noise considerably. When measuring the slope, only one differentiation is necessary.

3.3 Other methods

Depth from Defocus [Naya 96, Fava 05] and the closely related *Depth from Focus* require at least two images of a scene taken with different focus settings. The imaging system must be telecentric, that is the magnification must be independent of the

object distance [Wata 97]. A depth estimate can then be computed from the local blur or sharpness. Advantages over triangulation based methods are that there are no occlusions and no correspondence problem (see section 3.4). They work only on textured objects, but a texture can be projected if necessary. However, there are artifacts at occlusion boundaries as the focus measure can only be computed on windows with a non-negligible spatial extent. Also the achievable accuracy is rather low. In fact, Schechner [Sche 00] shows that the accuracy is equivalent to a stereo system with a baseline as large as the lens diameter. *Depth from Diffusion* [Zhou 10a] is a new variant. In this case the focal setting does not change. Instead blur is created by a diffusor in front of the object. This has the advantage that the accuracy now depends on the scattering angle of the diffusor instead of the lens aperture. Additionally the diffusor can be placed close to the object while the camera is far away, without loss of accuracy. *Confocal Stereo* [Hasi 09] varies the aperture and the focus of the camera and can reconstruct depth values for single pixels. It is therefore suitable for scenes with very fine detail that pose problems for other methods.

Another method is *Shape from Silhouette*, also known as *Space Carving* or *Visual Hull* [Kutu 00]. In this approach, several images of a scene from different viewpoints are used to constrain the volume that is occupied by convex opaque objects. *Coded Aperture Imaging [Levi 07]* can also reconstruct the depth, though with relatively low accuracy.

3.4 Triangulation Methods

In triangulation-based methods, the position of a point in space is determined by the intersection of two or more rays of light. The basic geometry is illustrated in figure 3.2. The uncertainty in the recovered position depends on the triangulation angle. To obtain good accuracy, the two viewpoints must be separated by a certain distance. However, this causes occlusions, that is areas which are not visible in both cameras, and for which no depth can be recovered. For distant objects, the large baseline required for high absolute accuracy also makes the system unwieldy. Therefore triangulation is mostly used for close to medium range distances. An estimation of the errors for triangulation-based measurements is provided in section 3.4.3.

3.4.1 Stereo and Structure-from-Motion

One very common application of the triangulation principle is stereo vision and in particular binocular stereo with two cameras. In the simplest case we assume a point in space which is projected onto two different cameras. If the cameras are calibrated and the image coordinates in both images are known we can calculate the projection rays. By intersecting the projection rays, the 3D position of the point can be reconstructed. However, we have to be sure that the image coordinates in both images actually belong to the same point in space. This is the *correspondence problem*. If the stereo rig is calibrated, that is, the relative positions of the cameras are known, the correspondence search for a feature in one camera can be constrained

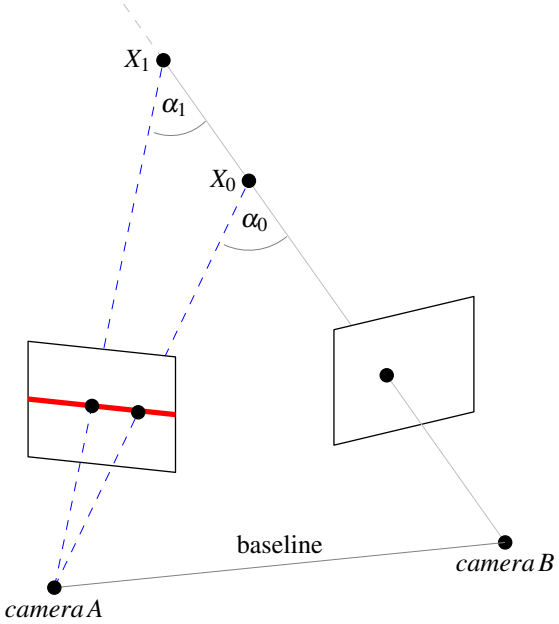


Figure 3.2: Principle of Stereo Triangulation. Given the projection of a point X in camera B, the image coordinates in camera A determine the 3D position of X. The triangulation angle α varies with the position of X. The epipolar line (see section 2.2.6) is marked in red.

to the appropriate epipolar line in the other camera (figure 3.2). There are numerous publications on a wide variety of stereo methods. A basic distinction can be made between feature-based and area-based approaches. The former try to identify feature points in the two images. This can be done using for example Scale-Invariant Features (SIFT) [Lowe 99], Speeded-Up Robust Features (SURF) [Bay 06], Maximally Stable Extremal Regions (MSER) [Mata 04] or simple edge features. Area-based or dense stereo methods assume locally smooth surfaces and match whole windows to get a dense depth map. There are many variants of the matching cost functions [Hirs 07], cost aggregation schemes [Gong 07] and optimization methods [Szel 08]. Also there are various preprocessing steps that can be applied, for example foreshortening correction [Xion 02], oversegmentation [Lei 06] or differentiation [Guan 09]. The benchmark for dense stereo algorithms is the Middlesbury Stereo database [Scha 02] that provides a set of test images with ground truth. Nowadays, with the help of GPU processing, many state-of-the-art algorithms are capable of real-time operation. However, the results are often quantized to just a few distinct depth levels. Furthermore, the results typically contain a small percentage of erroneous depth values. Stereo works best on scenes with cooperative textures. Such textures can also be created by projection [Tomb 09], thus yielding Active Stereo. Stereo has also been unified with Structured Light techniques in a framework called Spacetime Stereo [Davi 05].

Some common assumptions made in stereo algorithms are worth stating explicitly:

- Uniqueness: There should be at most one match for an item from one image in the other image. Reflections and translucent objects are not included in the model.

- Color Constancy: The photometric properties of two pixels forming a pair should be similar. This condition is typically violated by specular surfaces.

- Smoothness: Area-based algorithms, in particular, model the scene as mainly smooth, with only few localized discontinuities.

- Ordering: An object appearing to the left of another in one image should also be to the left in the other image. This is true if the scene is a single continuous surface, but it will not always hold if the scene has holes or small objects in front of others.

The case of triangulation between different views of a single, but moving camera is known as *Structure from Motion* (SfM) [Nist 05]. This is typically more difficult as the camera motion is unconstrained. No epipolar constraint exists to restrict the search space for potential matches. Most approaches are focused on navigation and produce only sparse depth data, but it is also possible to fill the gaps using optical flow techniques [Newc 10].

3.4.2 Structured Light

It is possible to replace one of the cameras of a stereo system with a projection device. Triangulation can then be performed between the projector light rays and the camera viewing rays. In this case one has to solve the correspondence problem between the camera image and the known projected pattern. This approach is also called Active Triangulation, as the pattern is actively projected onto the scene, opposed to passive triangulation that uses only features which are naturally present in the scene.

A very common form of Active Triangulation are so-called light sectioning sensors. Here the illumination takes the form of a line, which can be generated by focusing a laser beam with a cylindrical lens [Haus 88]. The object points are reconstructed by intersecting the illumination plane with the projection rays corresponding to the illuminated camera pixels. The number of points per image is limited by the number of rows (respectively columns) offered by the sensor chip. The laser illumination has a large depth of field and the environment light can be very effectively blocked by narrow-band filters matched to the laser wavelength. This enhances the robustness of the method. However, speckles are a nuisance that make precise localization of the line in the camera image difficult. They occur when coherent light is reflected from an optically rough surface. To get a full-field measurement, the illuminating line has to be swept over the object. This is also a source of imprecision due to mechanical tolerances. Still, line scanners are a good fit for quality control tasks on conveyor belts, since in that case no moving parts are required in the sensor itself.

It is a natural step to project not only one line but a two-dimensional light pattern to increase the number of 3D points that can be generated. However, this means the correspondence between projected and observed pattern is not obvious anymore. Therefore the projected pattern has to be encoded in some way. There are many ways to do this. Salvi et al. and Gorthi and Rastogi [Salv 10, Gort 10] recently published surveys of the various techniques. Some considerations to take into account are:

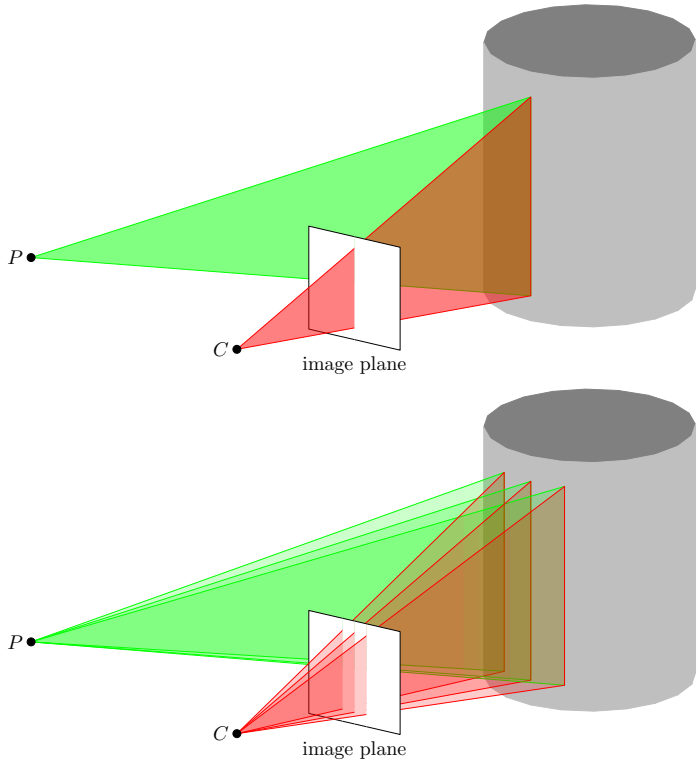


Figure 3.3: Different Structured Light principles: Line sensor (top) and area sensor (bottom).

- Is the coding temporal, spatial or mixed? In a temporal pattern multiple images have to be acquired. The sequence of values of a given pixel in the image is unique, given that there is no movement in the scene. In a spatial pattern the neighborhood of a given pixel in a single image is unique, given that the object surface is smooth.

- Is the pattern black & white, grayscale or color? This influences the robustness of the decoding. Black and white has maximum contrast. Color patterns may have difficulties with colored objects. Horn and Kiryati [Horn 97] and Caspi et al. [Casp 98] consider the problem of choosing the optimal pattern for a given level of robustness.

- Are both directions of the pattern coded or only one? Two-dimensional arrays of dots or checkers have been used, but stripe patterns are more common. This is because the second direction is not strictly necessary and stripe patterns are easier to design and decode.

- Is the encoding periodic or absolute? Periodic patterns need an additional unwrapping step to generate unique depth data, but are are easier to design.

There are some common assumptions Structured Light algorithms make about the scene. Not all assumptions are shared by all algorithms.

- Spatial Smoothness: The scene should have no depth discontinuities.

- Spatial Continuity: A weaker version of smoothness. There should be a smooth path between any two points in the scene.
- Local Smoothness: The scene does not exhibit discontinuities on scales smaller than the building blocks of the pattern.
- Reflectivity Smoothness: The scene should have approximately constant reflectivity.
- Color Neutrality: The scene should be in a shade of gray (achromatic).
- Temporal Smoothness: The scene should not move.
- No Ambient Light: The only light in the scene should be the Structured Light.

In the next subsections some examples of the most common coding techniques are presented. First, we define some important concepts related to Structured Light Patterns.

- *Primitives* are the smallest building blocks of a pattern, typically lines, rectangles or other simple geometric shapes. The *alphabet* is the set of all given primitives.
- A *pattern* in the most general case is a three-dimensional array of primitives. The first two dimensions are the spatial pattern coordinates, the third is the temporal coordinate. A particular primitive can thus be denoted as $P(x, y, t)$.
- A *code word* is a particular three-dimensional array of primitives. The first two dimensions are the spatial size, the third is the temporal size. For example, a specific $2 \times 2 \times 2$ array of colored dots might be considered a code word. Code words have a certain Hamming Distance, that is the number of positions in which they are different. The minimal distance for uniqueness is one. Larger distances allow error detection and potentially error correction.
- *Encoding* is the process of generating a pattern containing unique code words. Note that it is also possible to build a pattern from non-unique code words. We consider these as *unencoded patterns*. If only some codewords are unique, the pattern is *sparsely* encoded.
- *Decoding* is the process of identifying code words in observed patches of the pattern. The uniqueness of the codewords solves the correspondence problem.
- The *resolution* of a pattern is the product of the number of codewords in the x and y directions contained in the pattern. This is unrelated to the number of pixels used to draw a pattern and also different from the resolution of the final depth data to be computed. The maximum resolution is defined by the number of legal codewords. This in turn is defined by the size of the alphabet, the size of the code words and their minimal mutual Hamming distances.

The resolution of the projection device, the camera resolution and the system geometry also play a role in the design of the pattern. Features smaller than a few pixels cannot be reliably detected in the camera image. This dictates a lower limit for the size of the pattern primitives on the object. Otherwise they should be packed as densely as possible to recover a dense depth map and reduce the smoothness demands on the object surface. It is wasteful to design a pattern that has 200 codewords when only 100 can be projected and observed at reasonable size. The better solution is to adapt alphabet size, code word size or minimal distance so that only 100 codewords exist.

What is the mimimal size of a primitive in the camera image to be resolved reliably? For a quick estimation, we assume a simple stripe pattern. According to the Sampling Theorem, the sampling frequency has to be at least twice as high as the highest frequency in the pattern. We therefore need at least 2 pixels per stripe to recover the "base frequency" of the pattern. Furthermore, if the pattern is imaged with a color camera having a Bayer Pattern, its effective resolution is halved. Therefore a stripe should occupy at least 4 pixels in the camera image. A typical camera with a resolution of 780×580 pixels thus can resolve about 200 vertical or 150 horizontal stripes.

There are two basic encoding possibilities, 1D or 2D. Note that the pattern itself is always a 2D image. The difference is only in the encoding, that is in the form of the primitives used. One-dimensional patterns consist of lines, which correspond to planes in space. Depth data is calculated via ray-plane intersection. In two-dimensional patterns the primitives can have any 2D shape. Knowing their location allows depth computation by ray-ray intersection. This allows for additional error checks, as the two rays do not typically meet exactly. A large distance between the rays can indicate miscalibration or a false correspondence. However, 2D patterns have a disadvantage. Assume a 1D pattern of size s with code word size c. The "density" is $\frac{c}{s} \ll 1$. For a 2D pattern, the corresponding code word size is c^2 and the pattern size is s^2. The density is therefore $\frac{c^2}{s^2}$, which is much smaller than $\frac{c}{s}$. To put it differently, the number of possible codewords grows more slowly than the area to be covered. Therefore 1D stripe patterns are most common for color patterns. With spatial binary patterns 1D encoding is difficult, however, because the stripe order of black-white-black is predetermined. The only choice is varying the projected stripe widths. However, the observed stripe widths may be distorted by oblique surfaces and occlusions in the scene and are therefore not reliable. 2D encoding is therefore common for spatial binary patterns after all, but as previously stated, their density is relatively low.

It is tempting to view the pattern design as a classic coding problem, for which there are efficient solutions like Turbo Codes or LDPC codes [MacK 03]. However, there are fundamental differences: In Structured Light the sent message (the projected pattern) is two-dimensional and already known at the receiver (the camera). While symbol errors do occur, the main problem is the synchronization, that is, finding the spatial mapping of sent to received codewords. In classical coding theory the symbol error probability is minimized by adding redundancy to the codewords. However, in Structured Light, the codewords need to be as compact as possible to avoid

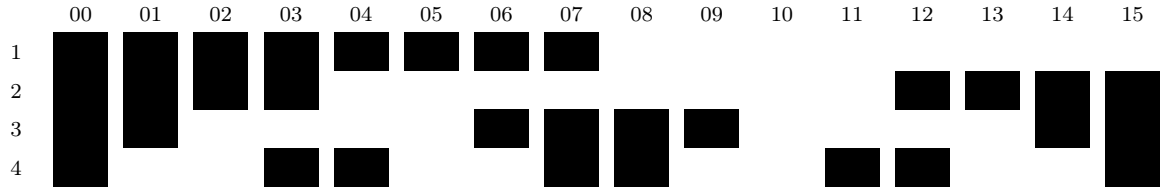


Figure 3.4: Counting from 0 to 15 with a 4 bit Gray Code. Only single bits change between successive numbers. In Structured Light, each number is a stripe index and each bit is realized as one projected pattern.

the situation where codewords are undecodable because they cover discontinuities in the scene.

Temporal Patterns - Binary Coding

Temporal binary Structured Light encoding was first used by Posdamer and Altschuler [Posd 82]. They projected a series of time-coded black-and-white dots on the object. The most common binary coding scheme is the Gray Code [Sava 97]. It is named after Frank Gray, who patented it in 1953. In opposite to regular binary counting, in a binary Gray Code only one bit changes between successive numbers (figure 3.4). This is a useful property as there are no intermediate states that can be misinterpreted. In Structured Light, Gray-coded patterns are mostly used for stripe indexing of non-unique phase shift patterns. With N patterns 2^N stripes can be indexed. The threshold to differentiate white from black pixels can be defined by observing the maximum and minimum intensities in each pixel over the whole sequence. To improve robustness, inverted patterns can be projected as well [Sato 86]. This simplifies the classification of whether a given pixel is "on" or "off" when interreflections are present in the scene. A typical Gray Code sequence consists of about 10 patterns. However, because of their binary nature they can be projected very fast with appropriate hardware. Using a custom DMD (Digital Micromirror Device) engine kit and a high-speed camera, Takei et al. [Take 07] built a system that can perform over 3000 3D measurements per second.

Hall-Holt and Rusinciewicz [Hall 01] proposed so-called "stripe boundary codes". They track the boundaries between black-and-white stripes over four patterns. A boundary can have four states: b-w, w-b, b-b, w-w. This allows indexing $4^4 = 256$ boundaries. The complication is that the latter two states are "invisible". The algorithm thus hypothesizes about their location. Consecutive invisible states are not allowed. This scheme even allows limited movement of less than half a stripe width in the scene and can still encode 110 boundaries. Reflectivity smoothness must be assumed, otherwise spurious boundaries can be detected. In another work, Young et al. [Youn 07] showed that well-placed additional cameras can reduce the number of patterns needed for a unique encoding.

Temporal Patterns - Phase Shifting

In interferometry, phase shifting is a very old technique [Crea 86], but it can also be used in Structured Light. Originally, the phase shifting was performed mechanically.

Therefore a common source of errors were small deviations of the shift interval. With *digital* fringe projection those do not occur, but quantization and non-linear projector brightness can lead to artifacts. There are many variations of the basic phase shifting algorithm. Wiora [Wior 01] has a useful overview. The most common forms are the so-called three-bucket phase shift with three images and the four-bucket phase shift with four images. The projected patterns are defined as

$$I_i = A \cdot cos(\phi - i\Delta) + \mathrm{B} \tag{3.1}$$

where I_i is the intensity in image i at a given pixel, A is the amplitude (typically $\frac{255}{2}$ for an 8 bit image), B is an offset (also typically $\frac{255}{2}$ for an 8 bit image), Δ is the phase shift interval and finally ϕ is an arbitrary phase. For the three-bucket variant, Δ is $\frac{2\pi}{3}$, for the four-bucket variant Δ is $\frac{\pi}{2}$. To recover the phase ϕ from the images the following equations can be used

$$\phi_3 = atan\left(\frac{\sqrt{3}(I_1 - I_2)}{2I_0 - I_1 - I_2}\right) \tag{3.2}$$

$$\phi_4 = atan\left(\frac{I_1 - I_3}{I_0 - I_2}\right) \tag{3.3}$$

The general equation for N images with a relative phase shift of $\Delta = \frac{2\pi}{N}$ is

$$\phi_N = atan\left(\frac{\sum_{n=0}^{N-1} I_n sin\left(\frac{2\pi n}{N}\right)}{\sum_{n=0}^{N-1} I_n cos\left(\frac{2\pi n}{N}\right)}\right) \tag{3.4}$$

However, the phase can only be recovered modulo 2π. All values lie in the interval $[0, 2\pi[$. To generate a continuous surface, *unwrapping* has to be performed. Multiples of 2π have to be added to the phase to reconstruct the correct period of the phase shift pattern. One way to do this is to use the aforementioned Gray Code to index the periods [Sans 99]. Other possibilities include heuristic unwrapping [Take 96, Gold 88], Bayesian methods [Biou 07, Nico 00] or multi-wavelength phase shifting methods [Li 08a, Gass 03].

An advantage of fringe projection is that the object reflectivity cancels out naturally. The resulting phase map is therefore independent of object texture, except in cases where the reflectivity is not constant within the area observed by a single pixel. Additionally, sinusoidal patterns are robust against defocusing. A rectangular pattern contains high frequency components that are attenuated differently by the transfer function of the system. Therefore the pattern changes for different levels of defocusing. A sinusoidal pattern contains only a single frequency. Thus, only the contrast reduces if the pattern is defocused. This property is especially useful since off-the-shelf projectors typically have a small depth of focus. Their aperture is very large to achieve maximum brightness.

The different phase shifting methods differ mainly by the number of images they require. As a rule of thumb, every additional image increases the robustness of the result. For additive noise this is obvious, but with more images it is also possible to compensate different kinds of systematic errors. For digital phase shifting a common source of systematic error is nonlinear projector brightness. For the three-bucket

phaseshift it is important to correct it [Pan 09]. More-bucket algorithms are less sensitive [Hibi 95, Liu 10]. The effect of quantization was examined by Zhao and Surrel [Zhao 97]. They concluded that for a projection system with a dynamic range of 8 bits or more the quantization errors are not significant.

Wang et al. [Wang 10] derived patterns with maximal signal-to-noise ratio. They turn out to have trapezoidal intensity profiles. Interestingly, the same patterns were used on an empirical basis in [Zhan 04] since they offer faster evaluation. No inverse tangens has to be computed (in contrast to equation 3.4). They can also be seen as extensions of the original intensity ratio depth sensor by Carrihill and Hummel [Carr 85]. Another idea is to generate the sinusoidal profiles by defocusing binary patterns [Lei 10]. The advantage of this approach is that nonlinear projector brightness does not influence the measurement.

Phase shifting offers a simple and largely texture-independent way to generate dense, high quality depth data. Unfortunately, it comes at the cost of having to acquire multiple images. Naturally, there have been many attempts to reduce the number of images and speed up the image acquisition process so that dynamic scenes can be measured as well. Wust and Capson [Wust 91] combined three phase shift patterns in a single color image. For non-gray objects this approach needs additional reference images under white and black illumination to correct the observed brightness for each phase, so three images are required after all. Taking advantage of the way DMD-projectors work, Huang [Huan 03] proposed to remove the color wheel so that three phase shift patterns can be projected with 240Hz. Using a synchronized camera they acquire three phase shift images per measurement. This approach has no problem with colored objects since the projected sinuoidal patterns are white. However, the unwrapping problem remains. Huang used only a few fringe periods and assumed a continuous object surface. Weise et al. [Weis 07] used the same projection scheme but solved the unwrapping problem with a second camera to eliminate the false matches. While "real-time" 3D is possible with these sensors, some motion artifacts remain. Also, the three-bucket phase shift is not very robust, so additional phase corrections are necessary [Zhan 07]. Wissmann et al. [Wiss 11] use a four-bucket phase shift with a superimposed coded pattern to assist the unwrapping process. Using a custom projection device they can acquire depth data at 50Hz.

Another possibility for a single-shot phase shift is to combine the patterns using different carrier frequencies [Guan 03] or to use only a single pattern and recover the phase using the Fourier transform [Take 83, Quan 10]. However both approaches work well only for a limited range of objects. Discontinuities and textures pose difficulties, as they introduce frequency components into the image which cannot be separated from the projected pattern.

Spatial Patterns - Monochrome coding

True single-shot systems are possible with spatially coded patterns. Binary black and white patterns are popular because they offer maximum contrast. Early work was done by Vuylsteke and Oosterlinck [Vuyl 90], who project a chess board pattern (figure 3.6/1) with specially encoded corners. The resolution is 64×64. Assuming limited distortion, the observed pattern can be decoded after simple image processing steps. Hu and Stockmann [Hu 89] proposed an unencoded black-and-white grid

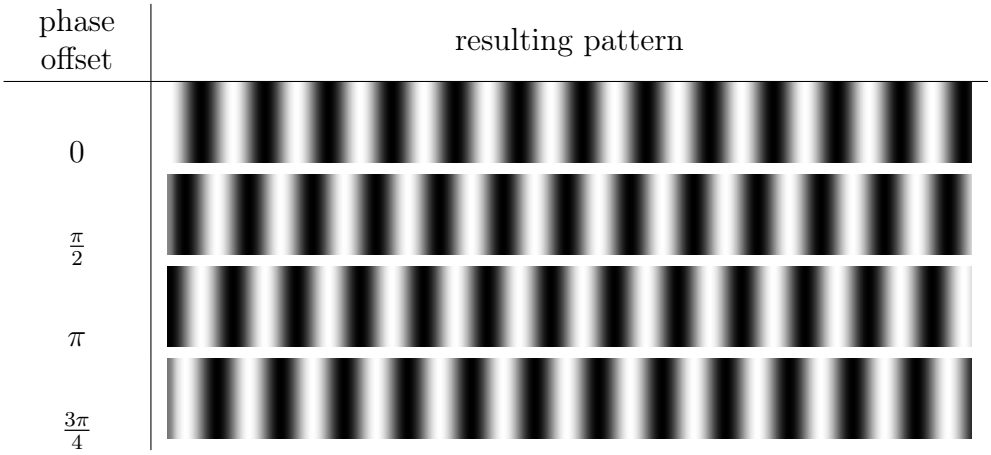


Figure 3.5: Example four-bucket phase shift patterns

pattern. With the help of general geometric constraints they were able to exclude many possible object shapes, but "a small degree of ambiguity remains". Proesmans [Proe 96] uses a very similar pattern viewed under orthographic projection. The intersection points are labeled relative to some starting position. The claim is that wrong labelings are avoided by using the most reliable path to reach other intersections. Only relative depth is recovered and discontinuous objects still pose difficulties. Koninckx [Koni 05a] proposes a black and white line pattern with a diagonal colored line for disambiguation via the epipolar constraint (figure 3.6/2). Kawasaki [Kawa 08] also uses a grid pattern (figure 3.6/3) and reconstructs the depth with the help of coplanarity constraints on the observed intersections. Brink et al. [Brin 08] present an algorithm to index a black and white stripe pattern with the help of maximum spanning trees. In general, unencoded grids suffer from a lack of robustness. They do not provide error detection, so errors during the identification process can propagate. Textures and discontinuities in the scene are therefore hard to handle. Additionally, the possibilties for confusion grow with the number of stripes and crossings in the pattern, so the realistically achievable resolution is relatively low. To a lesser degree, this is also true for sparsely encoded patterns.

Morita et al. [Mori 88] proposed a dot pattern based on pseudorandom arrays [Mitc 95, Etzi 88, Dene 90]. These so-called M-arrays have the property that all subarrays of a given size appear only once in the full arrays. That is, every such subarray is a unique code word. Observing such a subarray solves the correspondence problem. Griffin et al. [Grif 92] also used a two-dimensional encoding with 4 geometric primitives arranged in code words of size 5 (figure 3.6/4). Their coding is optimal, that is, every possible code word occurs exactly once in the pattern. The resolution however is only 32×32. Maruyama and Abe [Maru 93] projected a pattern of discontinuous lines (figure 3.6/5). The gaps are randomly arranged. The code words are read along epipolar lines. Another approach was proposed by Devernay et al. [Deve 02], who project a pseudo-random noise pattern. They then try to find parts of it in the camera image by cross-correlation. While this is simple and reasonably fast, strong deformations of the pattern as well as object texture can cause problems.

The "Kinect" 3D controller system released by Microsoft in fall 2010 is based on a single-shot triangulation sensor. It uses a near-infrared "speckle" pattern to illuminate

the scene (figure 3.6/6). The observed pattern is compared to a reference view by means of local cross-correlation. The displacements of the correlation windows give the depth. The illumination is monochromatic, so environment light can be filtered very effectively. The baseline is relatively short at 75mm, compared to a working distance of approximately 1000mm to 3000mm. The camera therefore sees the pattern almost without distortions, which makes the correlation more robust. On the other hand, the short baseline limits the accuracy of the sensor. The sensor is unobtrusive to human eyes and works reliably on a variety of surfaces at a frame rate of up to 60Hz. There is only a small amount of published material about the measurement principle apart from two patents [Shpu 07, Zale 06]. There is also an older paper by the same authors on depth reconstruction with a related method [Garc 08]. This older method works without triangulation merely by observing a projected speckle pattern, which varies in z direction, and comparing it to stored reference images. In the final product the speckle pattern was tweaked to be constant in z-direction (in the far field). Also, the intensity distribution was changed to be effectively binary (see figure 3.6). It is unknown how exactly this pattern is created. The advantage of this type of illumination is its very high depth of field. Additionally, the pseudo-speckle peaks are extremely bright compared to the background. This high contrast overcomes object textures and makes the correlation step more robust. Since this single-shot sensor is commercially available we compare it to our results in section 6.1.4.

Spatial Patterns - Color coding

Compared to monochrome patterns, color patterns can be encoded three times more densely, since they have three channels. However, there is a cost: the observed color depends not only on the projected color, but also on the scene color. Therefore color encoded systems tend to have difficulties with strongly textured scenes. The earliest 3D sensor based on color coding was proposed by Boyer and Kak [Boye 87]. The pattern, composed of stripes of different colors, contains unique blocks of stripes. However, not all blocks were unique. Later authors [Monk 93, Hugl 89] introduced patterns based on De Bruijin sequences[Mitc 96, Anne 97]. These are in principle one-dimensional pseudorandom arrays. Subsequences of a certain length are unique. Observing them solves the correspondence problem. Zhang et al. [Zhan 02] also used De Bruijin sequences for their pattern (figure 3.7/1). Their contribution is an elegant recursive decoding algorithm. It works on each scanline individually, so the noise robustness is limited. They have to explicitly assume the ordering constraint. The authors claim, however, that scenes violating this constraint can still be measured using multiple decoding passes.

In a parallel line of development, Tajima [Taji 90], Häusler [Haus 93] and Geng [Geng 96] used a rainbow color spectrum to encode the pattern without any explicit stripes. However, it is hard to reliably distinguish so many colors in the camera image, even if the scene is color-neutral. Therefore, Häusler proposed a complex setup to exclude the green part of the spectrum that cannot be reliably mapped to the wavelength with a standard camera. Tajima even advised to average over 10 images to reduce noise, but of course this is in conflict with the single-shot principle. Ambient illumination poses additional problems and has to be compensated with a second image. Because of these limitations, spectrum-based approaches did not

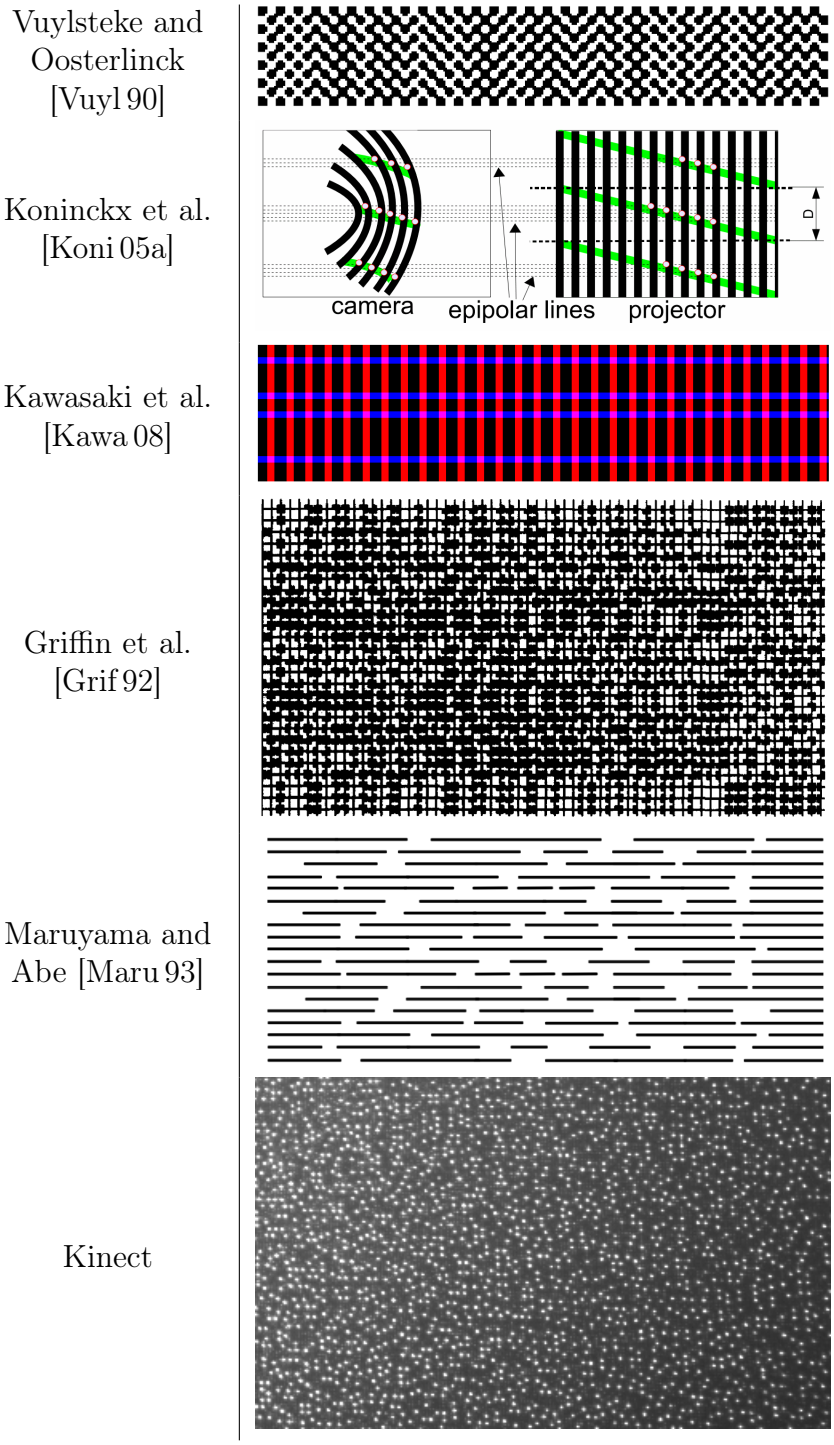


Figure 3.6: Monochrome patterns used by different authors

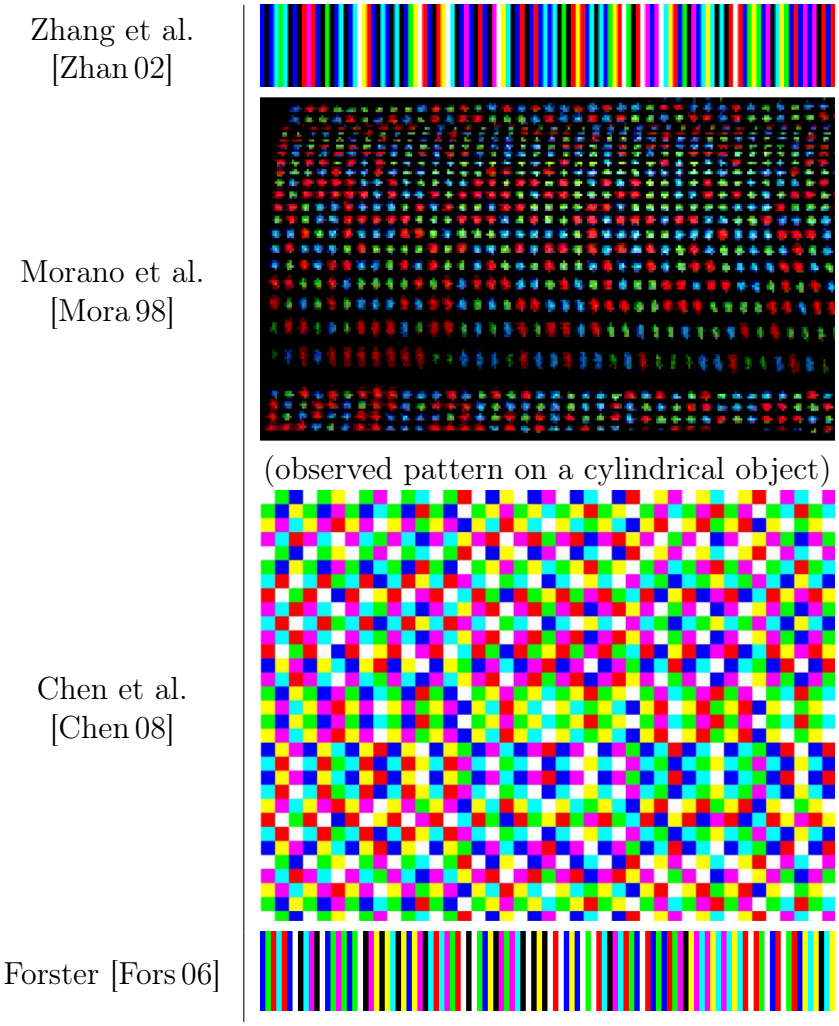


Figure 3.7: Color patterns used by different authors

become widely used. However, the principle of the measurement method is attractive and might be worth revisiting with modern hyperspectral cameras.

Morano et al. [Mora 98] used a 2D-encoded pattern (figure 3.7/2). The code is based on M-arrays. They also proposed the use of error-correcting codes, but did not use them in their actual implementation. Chen et al. [Chen 08] used a 2D pattern as well (figure 3.7/3), but with a non-formal construction. They analyze the time complexity of their decoding algorithm in great detail, but there is no accuracy evaluation.

Forster [Fors 06] employed a stripe pattern with error-correction (figure 3.7/4). The decoding step works per scanline, but propagates successful identifications along the stripes. The algorithm was designed to be as robust as possible and the implementation is fast. This system can be viewed as a predecessor to the work described in this thesis.

Summary of triangulation based systems

Stereo methods offer dense depth maps in real time, and they do so passively and unobtrusively. However, they work only for cooperative, texture-rich scenes. Also, dense in this context is used in the computer vision sense. In the physical sense the depth data is only plausibly interpolated between known points of support. Additionally, the accuracy is often rather low as only few distinct disparity levels are used to speed up the computation. A small baseline makes the stereo matching easier, but also reduces the accuracy. Wide baseline stereo methods are considered a different branch of research as they have to assume that an object is seen from very different perspectives. Outliers are an additional problem. The Middlesbury Stereo benchmark [Scha 02] established that even the best algorithms produce a small percentage of outliers. What's worse, they can be almost arbitrarily far off, so the error in the RMS sense can be very high.

Structured Light is similar to Active Stereo, but the hardware is even simpler, since only one camera is used together with the projector. The temporal coding scheme combining Gray Code and Phase Shifting is very well established and works reliably on almost any object. The drawbacks are that it is unsuitable for dynamic scenes. This can be mitigated with expensive hardware, for example in the system described by Takei [Take 07], but the principal problem remains. Along the same direction, using multiple patterns always requires a complex projection device that can quickly switch between different patterns and does not introduce positioning errors. Mass-produced digital projectors are not very expensive, but have limited resolution and intensity quantization. Using analog projection solves this, but typically introduces mechanical instabilities if more than one pattern has to be used.

Single-shot methods are capable of measuring dynamic scenes. They are also very elegant: The hardware is reduced to the bare minimum. There are no moving parts, only a slide projector and a camera. Binary single-shot patterns are mostly sparsely encoded or encoded in 2D with geometric shapes as primitives. Sparsely encoded patterns, in particular, are sensitive against error propagation. Geometric encoding has a low density. Therefore it needs smooth surfaces and offers only limited resolution. The robustness is low since it is hard to incorporate error correction mechanisms. Color single-shot patterns offer more flexibility. The patterns based on pseudorandom sequences have a relatively high density. By maintaining a minimum Hamming Distance between the code words, error recognition and error correction can be introduced. The challenge is to cope with textured and colored objects.

3.4.3 Error estimation for triangulation-based systems

There are several sources of errors in a triangulation-based 3D acquisition system. The following collection of error sources is valid for binocular Stereo as well as Structured Light. In the latter case, we assume a projector is an "inverted" camera.

- Matching Error: It occurs when triangulation is performed between falsely identified rays. The resulting errors can be arbitrarily large. It is the responsibility of the preceding stages of the Stereo or Structured Light processing algorithm to exclude such false matches.

- Localization Error: Even when features have been correctly matched, their location is not known with perfect accuracy. There is a small difference between Structured Light and Stereo. The projected pattern is theoretically exactly known, but depending on the type of projector used, small uncertainities remain.

- Calibration Error: The position, orientation, distortion and other parameters of the cameras are not precisely known. Any inaccuracy propagates into the triangulation. While there are highly accurate calibration algorithms, the stability of the calibration is a practical problem. Vibrations during transport may necessitate re-calibration. Temperature changes cause components to expand or shrink and can therefore change the parameters of the system.

- Model error: The calibration model may not fully represent the actual physical system. A prime example is the pinhole camera model that breaks down for ultra-wide-angle lenses. Using an inadequate model causes additional errors.

We present a simple derivation of the localization error that must be expected in a triangulation-based 3D acquisition system. We make some simplifying assumptions. The first is that the image planes are coplanar. This is indeed often the case. If not, it can be achieved synthetically by stereo rectification [Gluc 01]. The second is that the left and the right camera (or projector) have equal focal length f. Again this occurs commonly in stereo setups, and if not, we can virtually rescale the image plane and the associated uncertainity to the necessary focal length. The idealized setup is visualized in figure 3.8. A more extensive treatment can for example be found in [Blos 87].

We express the coordinates of a 3D point P with respect to the left camera coordinate system. Similar triangles yield:

$$x_p = \frac{-x_L \cdot b}{x_R - x_L} \qquad y_p = -\frac{-y_L \cdot b}{y_R - y_L} \qquad z_P = \frac{f \cdot b}{x_R - x_L} \tag{3.5}$$

The uncertainity in z is the difference between the maximum and minimum values.

$$\Delta z = \frac{fb}{(x_R - \delta x_R) - (x_L + \delta x_L)} - \frac{fb}{(x_R + \delta x_R) - (x_L - \delta x_L)} \tag{3.6}$$

Introducing the disparity $d = x_R - x_L$ and the localization error $\delta x = \delta x_R + \delta x_L$ gives

$$\Delta z = \frac{fb}{d - \delta x} - \frac{fb}{d + \delta x} = \frac{2fb\delta x}{d^2 - \delta x^2} \tag{3.7}$$

With $d = \frac{fb}{z}$ and the approximation $d^2 \gg \delta x^2$ this becomes

$$\Delta z = \frac{2z^2\delta x}{fb} \tag{3.8}$$

The approximation is valid as triangulation-based systems are close-range systems. The disparity values are therefore several pixels, compared to a localization uncertainty that is only fractions of a pixel with appropriate subpixel localization

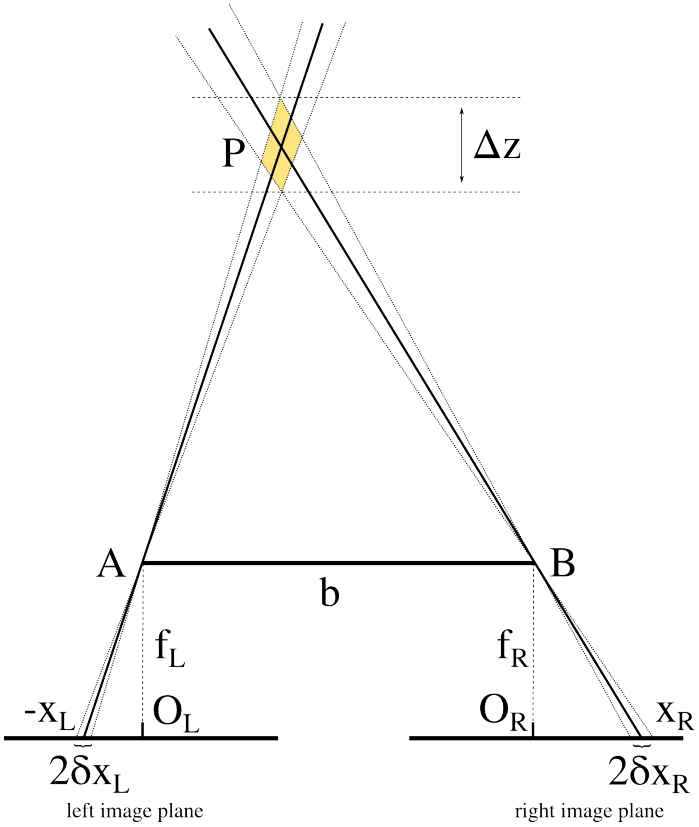


Figure 3.8: Standard triangulation geometry for two cameras A and B separated by a baseline of length $b = |O_R - O_L|$. The image planes are coplanar and the focal lengths are equal ($f_L = f_R = f$). Only the $y = 0$ plane is shown. The projection of a point P in the left camera coordinate systems is at x_L, while its projection in the right camera coordinate system is at x_R. Uncertainities δx in their position result in an uncertainity Δz in the triangulated depth.

algorithms. It can be seen that for triangulation-based measurement systems the error Δz increases quadratically with the distance z.

For Δx (and analogously Δy) we get

$$\Delta x = \frac{-(x_L - \delta x_L) \cdot b}{(x_R - \delta x_R) - (x_L - \delta x_L)} - \frac{-(x_L + \delta x_L) \cdot b}{(x_R + \delta x_R) - (x_L + \delta x_L)} \tag{3.9}$$

$$\approx \frac{2z^2 (x_R \delta x_L - x_L \delta x_R)}{f^2 b} \tag{3.10}$$

To get an idea what this means in practice, consider a typical close-range stripe projection system with a pixel size of 8.3µm and focal length of 8.5mm for the camera. The projector has a pixel size of 14µm and 15.5mm focal length. The baseline length is 100mm and the working distance 300mm. The localization error is more elusive. For the camera we can assume 0.2 pixels (see section 2.3). If the projector is of DMD type, the fill factor is 90% and the edges between the stripes are well defined. We therefore assume 0.1 pixels. On the projector side we additionally have to scale the resulting uncertainity to a virtual focal length of 8.5mm. All in all this gives

$\mathbf{\Delta z = 0.51mm}$. Assuming $x_p = \frac{b}{2}$, we get a lateral uncertainity $\mathbf{\Delta x = 0.085mm}$. For a typical setup the depth error thus is much larger than the lateral error. The relative depth error is "only" **0.17%** of the working distance. One has to keep in mind, however, that this is a lower bound that does not include calibration errors. The total error may be larger, especially on uncooperative surfaces, where the edges are more difficult to locate. Different system geometries can also change the analysis. Our result is in accordance with Trobina [Trob 95], who modelled the errors of a stripe projection system and reported relative depth errors of about 0.15%.

Zhao and Nanhakumar [Zhao 96] examined the effects of various types of miscalibration. In particular these are incorrect roll/pitch/yaw angles between the cameras and incorrect distortion parameters. They found that roll and pitch errors are relatively benign. The resulting errors are less than 1/4000 for 1 degree of misalignment. On the contrary, a yaw angle off by 1 degree causes about 1/20 relative depth error. Tangential distortion in the form of a sensor plane rotated by 1 degree causes up to 1/100 of depth error. Applying radial distortion also resulted in up to 1/20 of relative depth error. The **yaw angle** and the **radial distortion** are therefore the most critical parameters.

A related and complementary measurement principle extracts information about surface slope and curvature from the observed widths and angles of a projected stripe pattern. This was for example proposed by Winkelbach and Wahl [Wink 02]. However, Yang and Wang [Yang 96] showed that this direct computation technique for the slope gives large errors, compared to computing the slope from distance data.

3.5 Endoscopic 3D scanning

Endoscopes are used to examine the inside of potentially very narrow cavities through relatively small openings. Thus, their diameter is an important factor. Typical endoscopes are therefore monocular and provide only 2D images with few depth cues. However, for many applications in medical, as well as industrial settings, metric 3D data is desirable. Such a 3D endoscope would permit synthesizing wide baseline stereoscopic images for surgeons, providing them with an intuitive 3D visualization rather than with flat images. It would also allow performing absolute 3D measurements such as the area and volume of a pathological structure. Moreover, it might simplify solving advanced tasks such as coverage analysis (i.e. checking if 100% of a surface has been seen in the course of an inspection) or registration of endoscopic images with pre-operative data generated in a CT or MR scan.

One solution for generating endoscopic 3D data is Structure-from-Motion [Thor 02, O 11, Wang 08, Zhou 10b, Hu 10]. The distribution and quantity of trackable features in the scene determines the density of the resulting point cloud. There are currently two major limitations for SfM besides the dependency on features: First of all, implementations often have difficulties providing live feedback as typically a whole image sequence has to be processed before 3D data can be computed. The lag described in the literature varies widely; Hu et al. [Hu 10] report a processing time of several minutes while Grasa et al. [Gras 09] demonstrate an SfM system running at 25 Hz. More importantly, existing approaches assume a rigid scene, yet the scene tends to be non-rigid in medical applications. This may prevent a reconstruction or - in the

worst case - cause artifacts. It is important to note that SfM generates 3D data only up to scale; it cannot be used for absolute measurement, but it is suitable for stereo view synthesis. Hu et al. report an RMS reprojection error of their tracked features of around 1 pixel. This corresponds to a mean residual error of 1.68mm between their reconstruction and a ground truth surface; however it is unclear how the scale of the metric reconstruction was determined.

Stereoendoscopes provide two different perspectives on the scene. They can be used for direct human viewing or to reconstruct 3D data with computer vision algorithms. A stereo set-up is typically realized using two imaging sensors with two distinct lenses [Durr 95], but there are also alternative set-ups such as a single lens behind two pupil openings combined with a lenticular array on a single sensor chip. This design permits a small endoscope diameter [Taba 09, D 10] at the cost of a smaller triangulation base, which results in a 'weak' 3D effect. As always, with passive stereo algorithms the quality of the 3D data depends on the structure of the scene; featureless areas or viewpoint-dependent glares tend to cause problems.

There are also 3D endoscopes based on Structured Light. Armbruster [Armb 98] describe a rather large endoscope (targeted at industrial applications) based on the well-known phase shifting approach for the illumination pattern. In [Kole 03], the authors present a conceptually similar miniaturized holographic interferometer that can acquire data at a rate of 5Hz. The capsule has a diameter of 10mm and three protruding arms to provide three different illumination directions. They employ temporal phase shifting to get rid of the disturbing zero order and the complex-conjugate image arising in digital holography. The reported quality is impressive, but endocopes using phase shifting are not very suitable for moving scenes. To summarize, for many Structured Light systems described in the literature the overall diameter of the endoscope is considerably greater than 10 mm and consequently too large for many medical applications. An exception is Clancy et al. [Clan 11], who present a fiber-optic addon for the instrument channel of a rigid endoscope. They project dots of different wavelengths onto the scene and try to identify the wavelength in the camera image. However, given the color filters in typical cameras, this is a difficult task and the resulting density of 3D points is very low.

Time-of-Flight (ToF) methods have also been considered for endoscopic imaging [Penn 09]. The advantage of ToF is that it does not suffer from occlusion, unlike triangulation-based methods such as Stereo or Structured Light. At the same time, it is very challenging to build a small endoscope with an integrated ToF sensor. Penne et al. [Penn 09] did not miniaturize the hardware to the required level, but rather used a rigid endoscope with fiber optics for illumination and observation. The authors report an average error of 0.89mm for measurements of a plastic cube with a side length of 15mm at a standoff distance of 30mm. Surface texture and volume scattering in biological tissue (a significant effect for the infrared illumination typically used in ToF sensors) pose problems.

Shape-from-Focus has also been proposed for endoscopic measurements [Take 09], but it assumes a textured surface. Furthermore, the scene must be stationary while multiple images are acquired. As previously mentioned, this assumption tends to be invalid for medical scenarios. Conoscopic holography [Mugn 95] has also been applied for endoscopic measurements. It is a scanning method that can reconstruct the depth

of a single point at a time. Prasciolu [Pras 07] used a micromirror device on the tip of a rigid endoscope to build an in-ear scanner. The patient has to be fixated during the scanning process, which takes about two minutes. The scanner is guided using precise mechanics; in combination with the rigid scene this allows for the combination of the many single point measurements into a complete surface.

Emission-Reabsorption Laser Induced Fluoroscopy (ERLIF) is a method to determine the thickness of fluid films [Hidr 01], but it has also been proposed for 3D cavity measurements [Hern 10]. The cavities need to be filled with a suitable medium containing fluorescent dyes, which can be achieved with inflatable balloons. Clearly this can be problematic in many medical scenarios. Also it is no longer possible to record the surface color of the scene, therefore the approach cannot be used for realistic stereo image synthesis.

Detailed surveys about 3D reconstruction techniques for endoscopic applications can be found in [Moun 10] and [Miro 11]. In sections 4.3.3 and 4.3.4, we introduce a new endoscopic 3D sensor, based on Single-Shot Structured Light. It has a diameter of only 3.6mm and can be used to reconstruct cavities with a mean absolute error of below 100µm. These measurement results are presented in section 6.1.3.

Chapter 4

Design and Calibration of Single-Shot Structured Light Systems

In this chapter we first outline the goals and considerations for designing a Single-Shot Structured Light 3D scanning system. In single-shot systems, the design of the single pattern to be used naturally deserves much attention and is presented in greater detail. We also propose a new approach for the calibration of projector-camera systems. It is based on *active calibration targets,* which do not exhibit a fixed dot or chessboard pattern but rather encode their surface with the help of typical Structured Light schemes. This calibration approach is evaluated with simulated images as well as with different combinations of displays, cameras and lenses. It compares favorably to traditional feature-based calibration - in a stereo triangulation test, the reconstruction error could be reduced to a fifth.

4.1 General Design Goals

There are several common considerations and requirements for the design of a real-time 3D acquisition system.

- *Low cost.* Always a concern for any type of sensor. Single-Shot Structured Light (S3L) has advantages here as only off-the-shelf hardware is required. It needs only one camera and one projector. The latter only needs to project one single pattern, so in the simplest setup a static slide projector and an off-the-shelf camera can be used.

- *Safety.* To be usable in everyday environments, the sensor should pose no danger to humans. The critical point is mainly the eye safety of high-power lasers, even more so in non-visible wavelength ranges where there is no blinking reflex. For S3L systems this is generally not a problem.

- *Compactness and mobility.* S3L is based on triangulation and as such needs a certain baseline to be effective. A typical baseline length is about a fifth of the object distance. Hence, the compactness depends on the working range. For close-range measurements the sensor can be very small. In fact, the miniaturization potential of S3L systems is very good. Cameras with sizes in the mm

range are commercially available, and a static slide projector can also be very small. An S3L sensor can be light and easily transportable. Since only one image of the scene has to be acquired to calculate 3D data, the sensor can be hand-held. With a sufficiently low exposure time there are no motion artifacts.

- *Simple setup without high-precision adjustments. Easy, reliable and stable calibration.* The geometry of an S3L system does not have to fulfill strict precision requirements. It only needs to be stable, which may be a mechanical challenge. For a given setup, a few images of a calibration target (e.g. 6 each for camera and projector) suffice to determine all system parameters with high precision.

- *Large measurement volume.* This can be realized with appropriate wide-angle optics, but may increase the effort needed for accurate calibration. Depth-of-field is also a concern and may necessitate stopping down the system, which has to be compensated by increasing the exposure times or the illumination brightness to keep image noise at an acceptable level.

- *High absolute accuracy. High lateral resolution.* The lateral resolution is determined by the camera. One factor is the raw sensor resolution, which can reach into the megapixel range, but of course there also physical limits like diffraction and the general imaging quality that need to be considered. The accuracy of a S3L sensor depends on the quality of the calibration as well as the imaging noise and surface properties. Relative accuracies of better than $\frac{1}{1000}$ of the working distance are possible.

- *Suitable for all object and scene types.* Naturally, transparent and specular surfaces pose difficulties for an S3L sensor. Traditionally, S3L has been applied to uniformly colored, smooth surfaces. Part of this work is to improve the performance on highly textured or discontinuous objects. Another aspect are moving objects. In an S3L system dynamic scenes pose no problems.

- *Robust against environment influences.* An S3L sensor can be built without any moving parts and can therefore be very robust against vibrations. With LED lighting it can also be passively cooled, so accretion of dust is no problem. Temperature changes can influence the calibration qualitity and must be considered in the mechanical design. The remaining challenge is environment light, especially sunlight. Assuming environment light 10 times as bright as the scene illumination, the Structured Light decoding must make do with no more than $\frac{1}{11}th$ of the camera's dynamic range.

- *Reliability.* The sensor should not generate false depth data. In triangulation-based systems, a false correspondence can cause arbitrarily large depth errors. The pattern and the decoding algorithm must be designed accordingly to avoid such errors.

- *Fast measurement.* Real-time is understood here as video frame rate, that is 25 frames per second. In parallel processing one has to discern between throughput and lag. We therefore specify a lag of no longer than 0.1 seconds. In general, S3L systems trade hardware complexity for algorithmic complexity in the decoding.

In a single-shot system it is easy to acquire images at the required rate. The bottleneck is in the decoding.

Single-Shot Structured Light can fulfill most of these points. Time-of-Flight is a contender but has a number of drawbacks. The results are noisy and thus often have to be averaged over many frames to get a usable result. Most importantly, the sensors are challenging to calibrate because of systematic errors. The measured depth depends on the sensor temperature, the object reflectivity and even the air humidity [Lind 08]. Passive Stereo has major problems on unstructured surfaces. Active Stereo solves this, but the active illumination can be intrusive if it is in the visible wavelength range. Although this is also true of S3L, Active Stereo needs a second camera in addition to the projector and so is less compact and more expensive than an S3L system. Other monocular approaches like Structure from Motion do not yield dense depth data. Therefore, we consider S3L as the overall best-suited method. A key advantage of an S3L system is that it can be miniaturized for endoscopic applications with relatively little effort. Its main drawback is that it is not inherently robust on all surface types. When designing an actual S3L system, the aspects mentioned above have to be taken under consideration. Some are intrinsic to the S3L principle, other can be influenced by the geometric setup or hardware changes. Many points also depend on the choice of Structured Light pattern.

4.2 Pattern Design

Color stripe patterns offer a good compromise between resolution, robustness and fast decoding. A color-stripe pattern has to fulfill certain criteria to be useful. It should be as long as possible to encode a large volume with high density. At the same time the individual code words should be as short and distinct as possible to facilitate robust decoding. The required length of the pattern depends on the application. To be reliably detected in the camera image, a stripe should be at least 4 pixels wide. Thus, we need a pattern with about 200 stripes for a typical camera with a horizontal resolution of 780 pixels. The various design choices and their influence are described below.

- *Alphabet.* In our case the alphabet is the set of colors to be used in the pattern. Everything else being equal, a larger alphabet makes it possible to assemble longer patterns, but the invidual colors are harder to distinguish, so robustness suffers. Aiming for maximum robustness we choose to use colors with maximum distance, i.e. the eight corners of the RGB color cube: red (R), green (G), blue (B), cyan (C), magenta (M), yellow (Y), black (K) and white (W). In many cases it is even advisable to forego black and white as they can be easily confused with shadows (respectively highlights) on the object. The patterns are represented as strings, for example GBYMCRBGMYCYBRCMGR. Since a given color channel can only be either on or off in our alphabet, another possible notation for the colors are 3-digit binary numbers. The lowest bit is the blue channel, the middle bit the green channel and the high bit the red channel. For example, R corresponds to 4 in decimal notation, and C corresponds to 3.

- *Code word size.* For a dense encoding, the pattern must satisfy the *window uniqueness* property. That is, subsequences (code words) of a given minimal length must only occur once. It is important to note that a code word can be defined as a sequence of colors or as a sequence of color changes. Different colors can map to the same color change, for example in RG and MC the red channel falls while the green channel rises. A typical code word length is four, corresponding to unique subsequences of three color changes. The color changes are either denoted as strings of the form R-G+, or as six-digit binary numbers. The lowest bit indicates if the blue channel is falling, the second if blue is rising, the third if green is falling and so on. Thus R-G+ can also be represented as the single decimal number 24, while R+B- is 33.

- *Compatibility rules.* Obviously, neighboring stripes cannot have the same color, otherwise no edge can be detected between them. However, there are also combinations of non-identical colors that should be avoided. In particular, the contrast of red edges is often reduced by volume scattering as the penetration depth depends strongly on the wavelength. Since skin and other biological tissues are volume scatterers, this effect should not be neglected. Green edges have the advantage that green has a higher resolution in the Bayer Pattern of the camera (see figure 2.2). Blue is also good as it exhibits the least amount of volume scattering. Hence, a common rule is that at least two color channels must change at each edge. Other variants are that at least green or at least blue must change. All three rules reduce the number of colors that can lie next to a given color from 5 to 3 (in a six-color alphabet). We typically also require the pattern to be *normalizable*. That is, every stripe must have at least one neigbor where a given color channel changes. This helps to judge whether a color change is significant or not in the decoding stage. As an example, the sequence RGR is illegal, because the green stripe has no neighbor with a different blue value.

- *Minimum Hamming Distance.* To maintain the uniqueness property, the minimum Hamming Distance between two code words is one. We can require a Hamming distance higher than one, that is, we avoid code words that are too similar. However, with suitably chosen compatibility rules, it is often possible to use a Hamming distance of one.

- *Circularity* (also called wrapping). It is not always possible to generate a pattern with the exact number of stripes required by the application (if an exact number is even known). Instead of using a pattern that is "too long" and must thus be truncated, it is preferable to use a pattern with a smaller alphabet or smaller window size. This pattern may be "too short", so we partially repeat it. To make this possible the pattern has to be circular, that is, the last stripes together with the first stripes must form valid and unique codewords as well. The repetition of codewords of course introduces ambiguity for the depth values. But typically a pattern is repeated not more than twice. Given the expected object distance, it is therefore easy to determine the correct depth out of the two possibilities.

How long are patterns that can be assembled under these rules? Some theoretical considerations give an upper limit. We also implemented a pattern generator to find realizations of the various patterns. An exhaustive search is only practical for the shortest patterns. The longer patterns must be found in a pseudorandom search. Fortunately there is not only one optimal pattern for a given set of rules, but many. They can be created by circular shifts, inversion, color channel exchanges and other transformations.

In the first set of examples we use a Hamming distance $h = 1$, an alphabet size $k = 6$, and the rule that at least two color channels must change at an edge. That means a given color has $n = 3$ different valid neighbor colors. All patterns are normalizable and circular. We vary the code word length w.

- $w = 2$. There are $kn = 18$ code words. The maximum pattern length is 18. Optimal patterns have been found using exhaustive search.

- $w = 3$. There are $kn^2 = 54$ code words. Of these, six pairs are not unique in the sense that they map to the same color changes, for example RGR and MCM. Incidentally, these 12 code words are also not normalizable. The resulting maximum pattern length is therefore 54-12=42. Optimal patterns have been found using pseudorandom search.

- $w = 4$. Building on the previous result for $w = 3$, there are 21 sequences of length 3 that can be used to continue a sequence starting with a given color. Of these, 4 result in invalid sequences. There are therefore $17 \cdot 6 = 102$ valid code words. The best patterns that have been found using pseudorandom search have a length of 90.

- $w = 5$. Again building on the previous result for $w = 4$, there are 51 sequences of length 4 that can be used to continue a sequence starting with a given color. 10 of these turn out to be invalid. Hence the maximum pattern length is $41 \cdot 6 = 246$. The best patterns found by pseudorandom search have a length of 202.

It is of course also possible to use higher Hamming Distances, a larger alphabet and different compatibility rules. Consider, for instance, the following representative cases.

- $w = 3$, $k = 6$, $h = 2$. At least one channel must change. The effect of the larger Hamming distance is similar to reducing the code word length by one - for example, in RGx only one choice of x is allowed, all other choices then result in a code word that is too close to the existing one. The given rules allow 30 codewords with regular spacing. However, these regular codewords cannot be combined in longer patterns because they form "loops", for example BGR, GRB and RBG. The maximum pattern length found by exhaustive search is 18.

- $w = 5$, $k = 6$, $h = 2$. The maximum pattern length found by pseudorandom search is 90, just as in the case w=4, k=6, h=1.

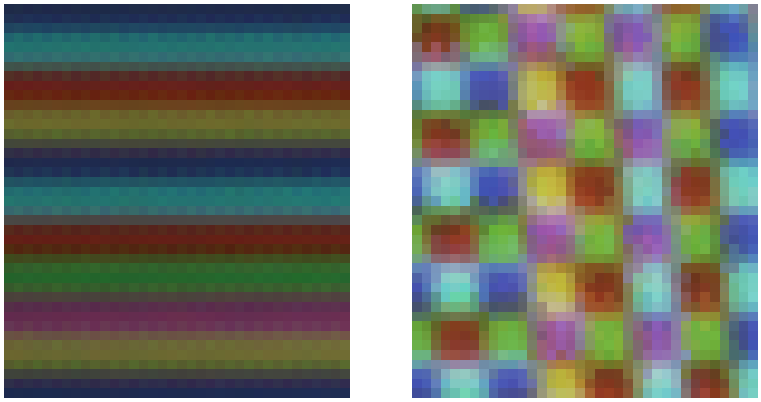
- $w = 4$, $k = 8$, $h = 2$. The maximum pattern length found by pseudorandom search is 104. The larger alphabet allows a reduction of the window size by one, while still forming a longer pattern.

- $w = 3$, $k = 8$, $h = 1$. Lowered Hamming distance and shorter window size again result in a maximum length of 104 stripes. This is the optimal length, as there are 128 possible codewords, of which 24 are not normalizable.

We have seen that patterns with up to 202 stripes can be generated with a window size of 5 and six colors. Although not optimal in the sense that all possible code words are incorporated, the pattern is long enough for practical use while the window size is still modest. For other applications, as for example the endoscopic system, the projection device supports only a much lower number of stripes. In that case it is possible to use window sizes as short as two stripes. Examples for the various classes are listed in the appendix.

Some authors like Caspi et al. [Casp 98] or Koninckx [Koni 05b] propose on-line adaptions of the projected pattern to the scene. This requires a projector capable of displaying different patterns, which negates one of the key advantages of Single-Shot Structured Light. We commit ourselves to maximum robustness and in advance design an application-specific pattern with at most 8 colors, which can then be realized as a projection slide. Furthermore, we choose colors in the visible range in order to be able to use commodity cameras with a standard Bayer filter. In principle it is also possible to use other colors, for example in the infrared range. This has the benefit of being less obtrusive for observers, but it also results in lower accuracy on some surface types because of increased volume scattering.

It is tempting to extend the stripe encoding with a perpendicular set of stripes to obtain a color checkerboard, which nominally should yield twice as much depth data per frame. However, there are problems with this approach. Firstly, as mentioned in section 3.4.2, the possible codewords of a given size are exhausted more quickly in a 2D encoding. This can be worked around by tiling the secondary direction with many repetitions of the same pattern, as long as the primary direction gives sufficient uniqueness. For example an $n \times m$ checkerboard can be created by repeating an $n \times 2$ pattern. Figure 4.1 shows details of a color stripe pattern and a color checkerboard pattern. Still, other problems remain. Because of blurring, the color contrast is much lower in the 2D encoding, leading to less reliable decoding. Furthermore, the edges cannot be precisely localized at the crossings. The 2D encoding thus results in depth data of lower accuracy, and the amount of depth data emperically is only about 30% larger than with 1D encoding. A smaller issue to keep in mind is that the system needs both a vertical and a horizontal triangulation base to calculate depth values for both sets of edges.



(a) Color stripe pattern

(b) Color checkerboard pattern (created by tiling a $n \times 2$ base pattern)

Figure 4.1: 1D vs 2D encoded pattern observed on a cooperative surface. The edges can be localized more accurately in the stripe encoding. In the 2D pattern the crossings result in ill-defined edge positions.

4.3 System Calibration

The calibration of cameras is a standard task in Computer Vision. The typical procedure is outlined in section 2.2. A calibration target with well-known fiducial marks is placed in different poses and imaged by the camera. From the correspondences between the world coordinates of the marks and their image coordinates the camera parameters can be determined. Instead of the traditional checkerboard or dot array calibration targets, we propose to use digitial displays as "active" targets. This approach is presented in detail in the next section 4.3.1. The key insight for the calibration of Structured Light systems incorporating projection hardware is that a projector can be treated as a reverse camera. Where for a typical camera calibration the world coordinates are known and the image coordinates are measured, for a projector the image coordinates are known and the world coordinates of the marks have to be found. This can be achieved with the help of a calibrated camera. Details are presented in section 4.3.2. We do not consider model-free calibration approaches based on lookup tables.

In sections 4.3.3 and 4.3.2 methods for calibrating an endoscopic Structured Light sensor are presented. This poses special challenges because of the unusual design of the camera and the projector used in the miniature sensor [Schi 11].

4.3.1 Camera Calibration with Active Targets

In contrast to the classic calibration approach that makes use of fixed features on dedicated calibration bodies, we generate virtual calibration marks on active targets using a Structured Light coding scheme. On the one hand this has simple practical advantages. There is no need to manufacture and validate a special target. The marks on the target do not have to be laboriously identified in an error-prone manual process, as is often the case. The coding scheme we use is tolerant against defocusing,

so the target does not have to be in focus for the calibration. On the other hand, we show that the achievable accuracy (as measured in the RMS reprojection error) is comparable to the best published calibration results and much better than the typically obtained values.

The idea of using a Structured Light coding scheme for camera calibration has also been proposed by [Saga 05], where it was used to undistort the images of a wide-angle camera in a model-free manner. In contrast, we perform a full camera calibration and recover the camera parameters, as they are needed for many tasks, for example 3D reconstruction. A similar active calibration approach is also briefly mentioned in [Rama 05, Gros 01, Gros 05] in the context of calibrating a catadioptric wide-angle camera. These works focus on calibration for wide-angle imaging and do not include thorough quantitative performance comparisons with other calibration methods. We are of the opinion that active camera calibration has advantages for any camera, not only extreme wide-angle cameras where the traditional pinhole model breaks down. We apply the active calibration approach to narrow-angle imaging with the pinhole model and show that the achievable calibration accuracy is higher than that of conventional passive targets.

Digital displays are suitable for calibration tasks as they are manufactured to very high precision using lithographic techniques. The pixel pitch is well-known, therefore pixel coordinates can be converted to metric 2D coordinates. One could simply show a checkerboard on the display and use that for calibration. However, such a method would still be subject to the noise-prone corner localization step. Instead, we propose the use of a series of coded patterns which can uniquely identify each individual pixel. Many coding schemes are possible [Salv 10]. Phase shifting is a dense encoding so that every single pixel can be identified. It also offers high accuracy because it does not involve any differentiation or binarization steps but works directly with the measured image intensities in each pixel. We use two four-bucket phase shift sequences, one horizontal and one vertical, to determine the x and y components of the pixel coordinates. The recovered phase is ambiguous, however. To obtain a unique phase value we have to "unwrap" it. There are various ways to achieve this. In our case it is known that the target is flat, so naive unwrapping would work. But since calibration is not a time-critical task, we used additional Gray Code sequences. Details of this standard Structured Light coding scheme can be found for example in [Scha 03]. All in all, a full pattern sequence consists of 4 images for the phase shift and 8 for the Gray Code (depending on the display resolution). Some of the resulting camera images are shown in figure 4.2.

Examples of the final unwrapped phase maps can be seen in figure 4.3. Using the phase maps φ_x and φ_y we can find correspondences of world coordinates with image coordinates. These can then be used as input for the camera calibration just as before. The actual lookup of the pixel coordinates (x_i, y_i) for given phase coordinates (x_p, y_p) is done by the "reverse" bilinear interpolation described in Algorithm 1 (see also figure 4.4).

With Algorithm 1 we can generate "virtual" marks from the phase maps with arbitrary density. An example image showing the distribution of the virtual marks can be seen in figure 4.5. The accuracy of the phase map depends on the local dynamic range. We typically discard marks with a dynamic range lower than 20 digits. As the

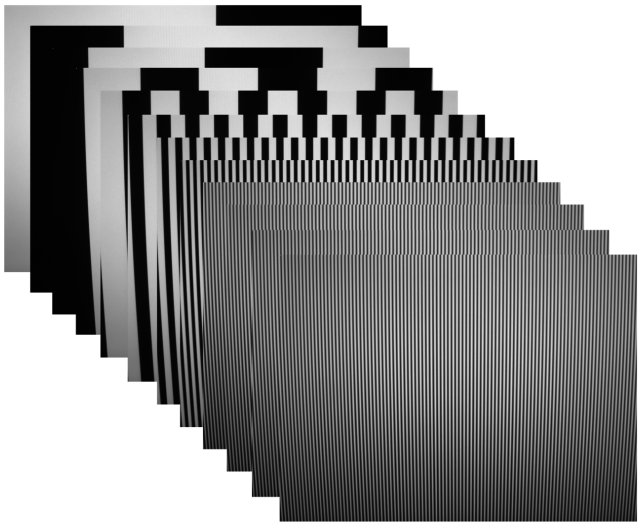


Figure 4.2: A pattern sequence to uniquely identify all pixels of the display. Only the vertical component is shown. The four images in the front are used to compute ambiguous phase values. The images in the back form the Gray Code used to unwrap the phase.

Algorithm 1 Subpixel Phase Lookup

1. Find a block of four neighboring pixels $\{p_k\}$ in the phase maps where both $min(\varphi_x(p_k)) \leq x_p < max(\varphi_x(p_k))$ and $min(\varphi_y(p_k)) \leq y_p < max(\varphi_y(p_k))$.
2. Perform least-squares fits to obtain a plane P_x from the values of φ_x in $\{p_k\}$ and a plane P_y from the values of φ_y in $\{p_k\}$. The set $\{p_k\}$ can be augmented by additional neighbors.
3. Intersect P_x with the plane $\varphi_x = x_p$ and P_y with the plane $\varphi_y = y_p$. This gives two lines.
4. Set the phase-component of the lines to zero and calculate the intersection point (x_i, y_i)

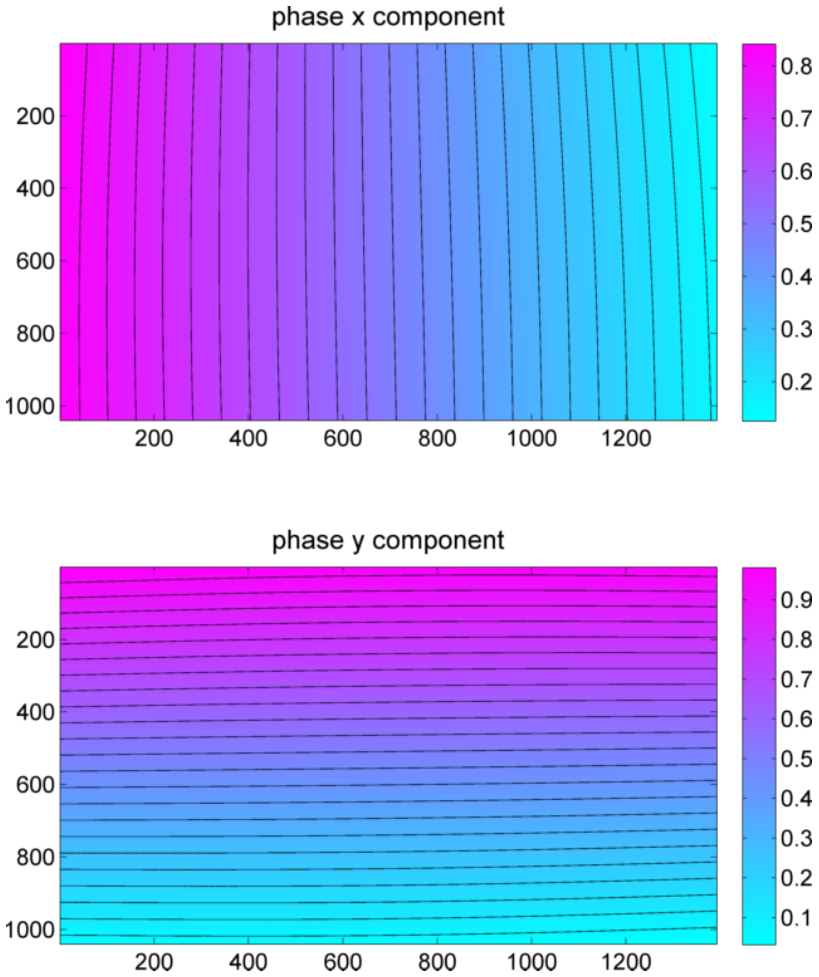


Figure 4.3: Phase coordinate components φ_x and φ_y with contour lines as seen by the camera. The values are normalized to $[0; 1]$. In this particular view the camera was rotated by approximately 180 degrees relative to the display.

apparent display brightness changes with the viewing angle, it can happen that some areas are too dark even when other parts of the image have optimal brightness. The plane fitting in step 2 of Algorithm 1 also provides us with the standard deviation of the measured phase values from the fitted plane. Good phase maps are very smooth, so typical values of the standard deviation are around 10^{-6}. If the phase map is noisy, the standard deviation is higher and we discard those points as well.

Ray offsets

A further improvement can be achieved by modelling the refraction caused by glass plate that covers the pixels of the display. The protective glass plate refracts the emitted light and causes a shift in the pixels' apparent position. To correct for this effect we use a three-step algorithm. We first calibrate with the point correspondences we found as if there were no glass plate. We obtain approximations of the camera poses relative to the display. In the second step we compute the (small) offsets

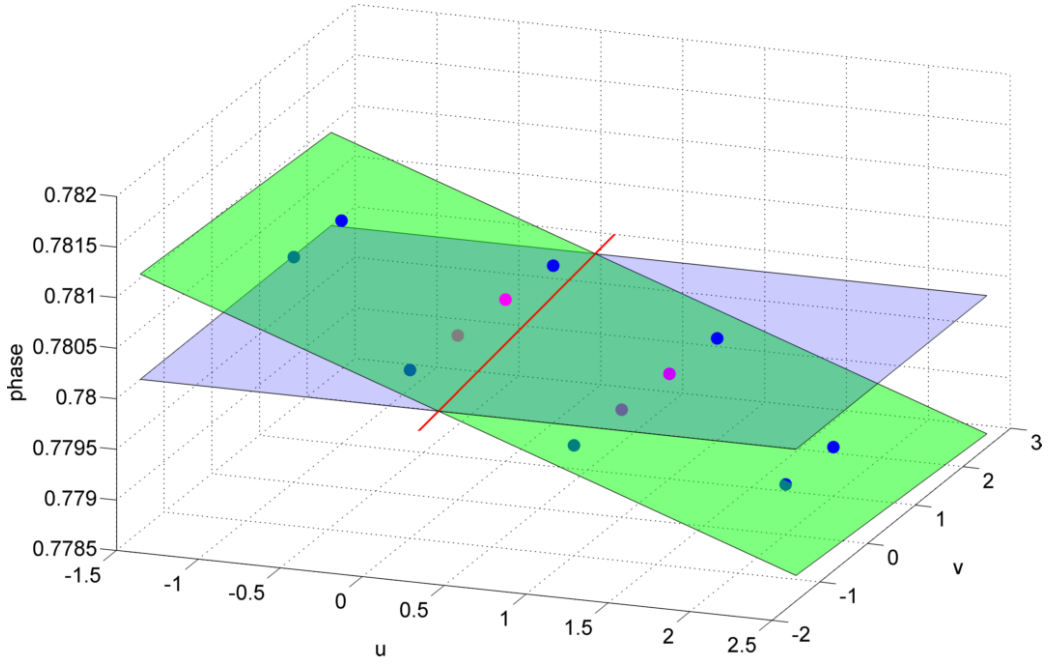


Figure 4.4: Phase coordinate lookup for one component. The dots are the measured phase values. Magenta indicates the original block of four pixels. The blue dots are additional neighbors used in the plane fit. The green plane is the linear local approximation of the phase (P_x). The blue plane ($\varphi = x$) represents the sought-after phase value. The intersection of the two planes is marked by the red line. The second phase component yields another line (not shown here). The intersection of both lines gives the pixel location of the phase coordinate of interest.

introduced by oblique viewing angles through the glass. Finally, in the third step we calibrate again with the corrected coordinates.

The height offset is

$$h = d \cdot \left(1 - \frac{\tan \alpha_2}{\tan \alpha_1}\right) \tag{4.1}$$

where $\alpha_1$and α_2 are related by Snell's law (figure 4.6). Note that h is undefined in the case $\alpha = 0$, but the limit for $\alpha \to 0$ is $h_\perp = d\left(1 - \frac{n_1}{n_2}\right)$. In the case of glass and air we have $n_1 = 1$ and $n_2 = 1.5$, so for perpendicular view $h_\perp = \frac{d}{3}$.

The thickness of the glass layer and its index of refraction are only approximately known. For our experiments we assumed a refractive index $n = 1.5$, which is typical for glass and glass-like substances. We estimated the thickness of the coating as $d = 1mm$. In the example plot of the height offsets shown in figure 4.7 the difference in the height offset between a perpendicular view in the center and an oblique view at the edges is only 0.04mm. Assuming a maximum viewing angle of 45°, this corresponds to lateral offsets below 0.028mm. The pixel size is 0.272mm. As a rule of thumb, the lateral offsets introduced by the glass plate are thus below 0.1 pixels.

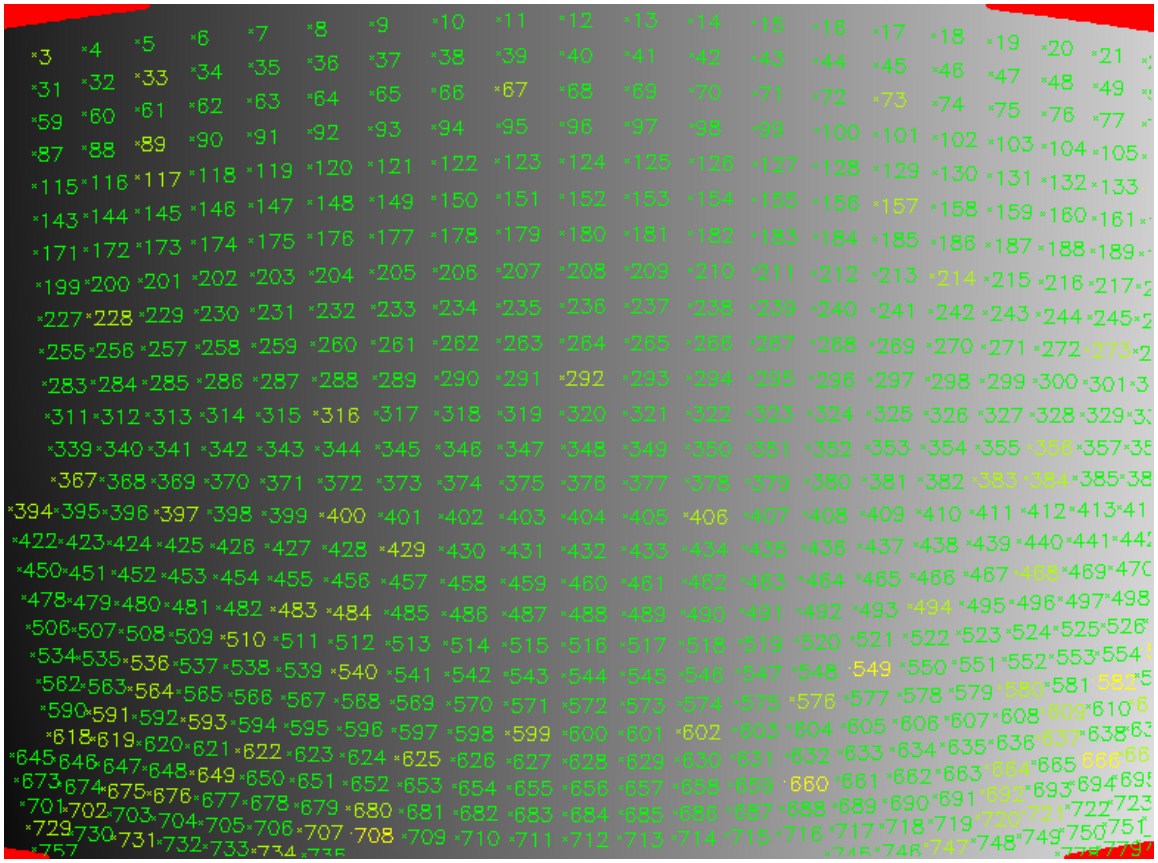


Figure 4.5: Marks selected from the phase map. Red areas are invalid. The more yellow a mark, the lower its quality.

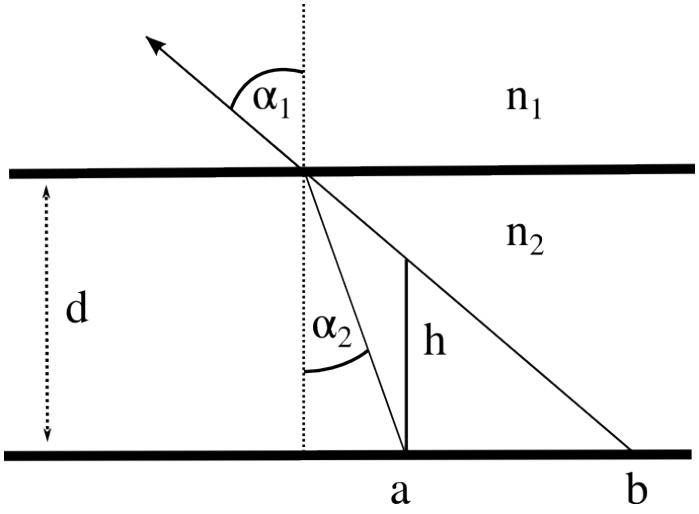


Figure 4.6: The glass plate refracts the ray coming from pixel (a) so that its apparent position is (b). Adding the offset h corrects the error.

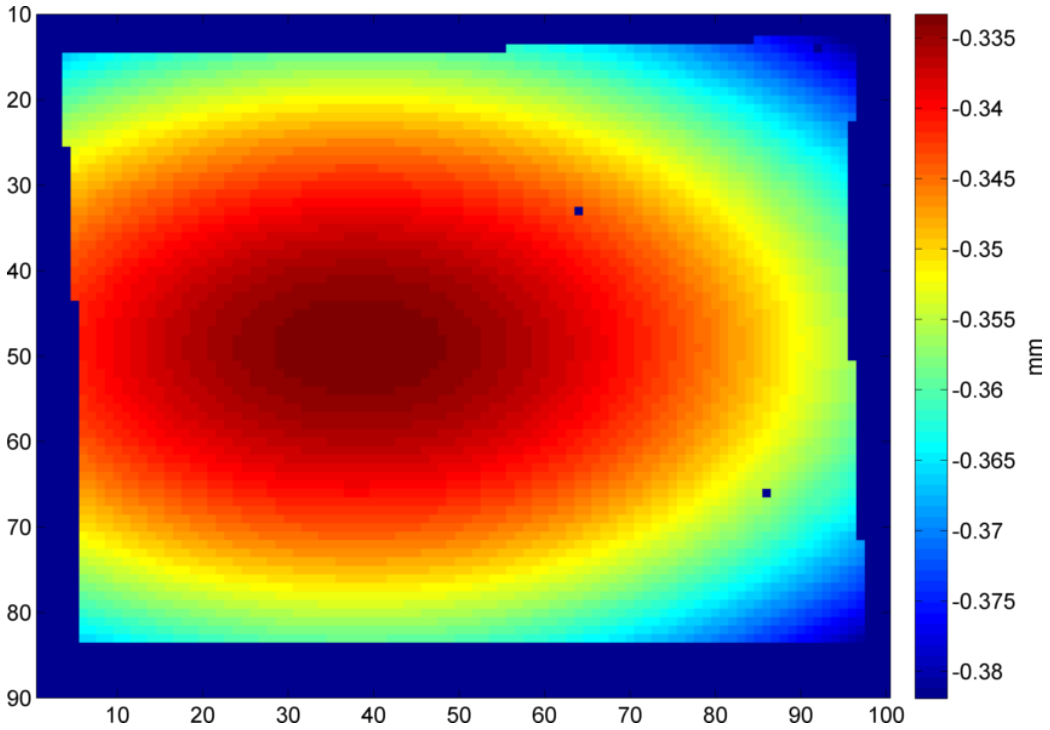


Figure 4.7: Offsets introduced by the glass plate covering the display. One sample per virtual mark.

Display Gamma

In our experiments we found that the gamma of the display is typically not constant over the entire area. We performed a local gamma calibration by displaying a series of progressively brighter images and tracking the observed brightness in each pixel. The brightness of some pixels rises earlier than others (figure 4.8). This effect seems to be due to the backlight arrangements used in the displays. This is a concern for a high-quality phase shift, as the sinusoidal intensity pattern is distorted. After the pixel-wise gamma calibration the individual display pixels observed by the camera can be identified using the coding scheme outlined above. The gamma can then be pre-corrected when drawing the patterns. If the effect of a wrong gamma setting is large, this calibration could even be done iteratively. However, the four-bucket phaseshift is robust against such errors [Liu 10], so a precise gamma calibration is not necessary. The effect is also plotted in figure 4.9.

Evaluation

The active calibration method was evaluated in several ways. We used simulated images where the ground truth camera parameters are known. We also tested various real-world setups with different combinations of cameras, lenses and displays. In each test, the calibration with an active target is compared to a calibration using a checkerboard pattern.

The standard targets in our lab are checkerboard targets with isolated squares (figure 2.6). Their advantage is that the unoccupied space in between the markers

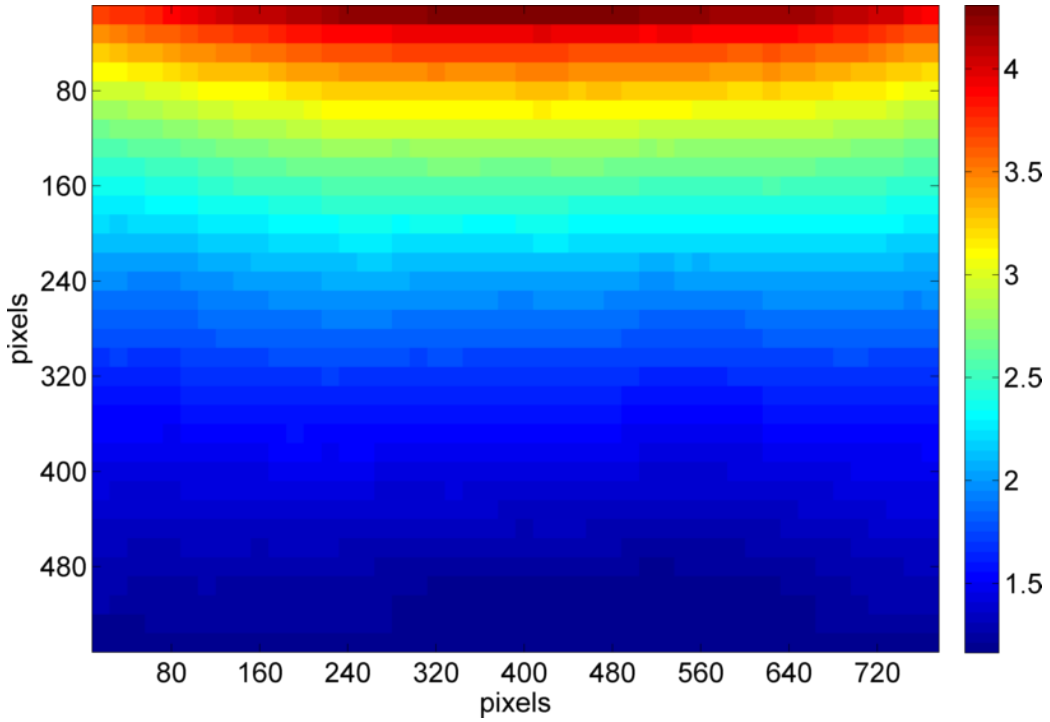


Figure 4.8: Gamma variations over the area of a Samsung Synchmaster 2433 display, as observed by a camera. The display uses edge-lighting. The brightness of the pixels near the top rises later, but steeper.

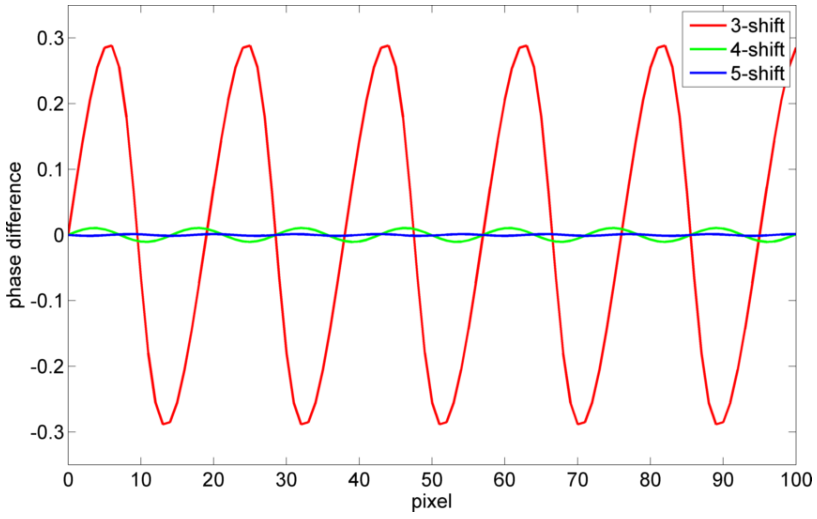


Figure 4.9: Phase differences between simulated patterns with gamma 1 and patterns with gamma 2.2. The three-bucket algorithm should not be used. For the four-bucket variant the maximum difference is 0.01, for the five-bucket algorithm 0.001. In this example a wavelength of 57 pixels was used.

can be used to perform projector calibrations. On a regular dense checkerboard the projected marks are much harder to detect. Our targets also have been examined with a coordinate-measuring machine, so the mark locations are known with very high precision. Standard checkerboards offer no means of determining the orientation, so the mark spacing must be assumed to be constant. The corners of the checkers are localized in the camera image either with the Saddle Point method (SP) [Lucc 02] or with the Line Intersection technique (LI) [Stoc 02]. We use the standard SP implementation provided by OpenCV and a self-implemented LI variant. Since we use a Phase Shift coding for the active target, our proposed method is abbreviated as PS in the subsequent sections.

All calibrations use the camera model and optimization algorithm proposed by Zhang, as implemented in the OpenCV library. The error metric used for the comparisons is the RMS reprojection error between the observed and undistorted mark coordinates and the projected mark coordinates in the images. This is a standard metric and is comparable with that presented in different publications. The projected mark locations $[\widetilde{u}_i, \widetilde{v}_i]$ are computed from the known world coordinates of the mark with help of equation 2.8. The tilde indicates that these coordinates are calculated by pure perspective projection without any image distortion. Another way to obtain these "ideal" coordinates is to correct the distortion in the observed coordinates. The undistorted mark coordinates $[\hat{u}_i, \hat{v}_i]$ are denoted with a hat. They are computed from the observed distorted mark positions by inverting equations 2.9 and 2.10. The reprojection error is then

$$e = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{u}_i - \widetilde{u}_i)^2 + (\hat{v}_i - \widetilde{v}_i)^2} \tag{4.2}$$

Here N is the total number of points used for the calibration. An individual calibration mark can occur several times in the images from different camera poses.

Simulated images

Calibration images in five camera poses were rendered with a simulated resolution of 800x600 pixels. No noise was added to the images. We minimized aliasing artifacts by avoiding poses aligned with the camera axes. The five poses we used can be seen in figure 4.10. Table 4.1 shows the resulting internal camera parameters and reprojection error for all three calibration methods. The parameters obtained with the proposed method are at least one order of magnitude closer to the real values than with the two other methods. The remaining error is probably due to quantization noise.

Real images

In our real-world experiments we used combinations of different displays, cameras and lenses. The details of the displays and cameras are collected in table 4.2. The lenses had focal lengths of 12.5mm, 8.5mm, 6.0mm and 4.8mm. We used an f-number of 8 and an object distance of 0.5m in our image acquisition.

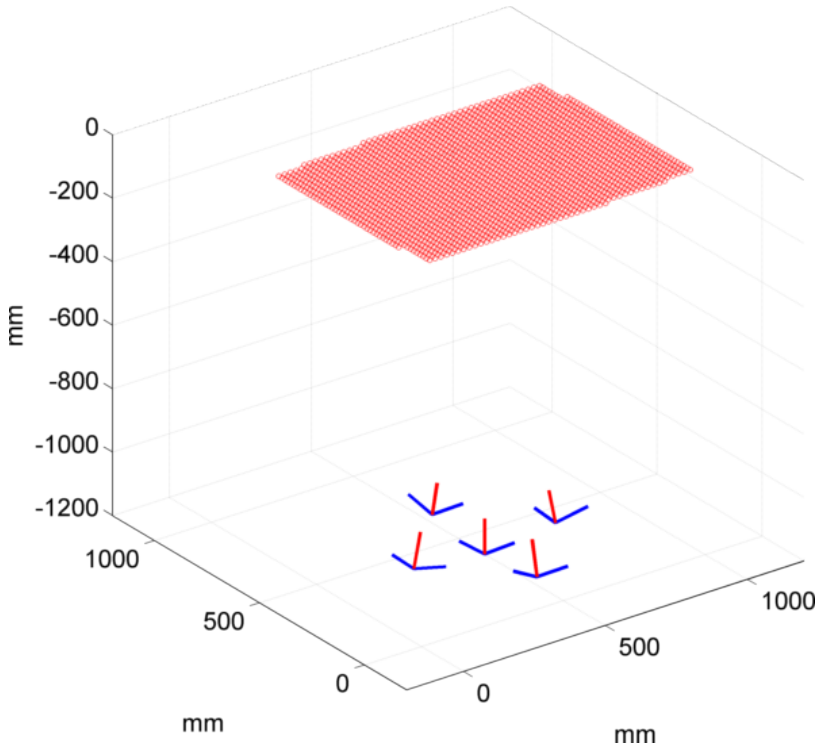


Figure 4.10: Simulated Camera Poses. The red circles are marks, the red lines the camera z-axes. The blue line pairs indicate the camera image planes.

	ideal	PS	LI	SP
$f\ [mm]$	12	12.0004	12.0050	12.0042
$k_2\ [\frac{10^{-4}}{mm^2}]$	0	-1.03	15.6	18.0
$k_4\ [\frac{10^{-4}}{mm^4}]$	0	4.74	-160	-2.71
$k_6\ [\frac{10^{-4}}{mm^6}]$	0	-6.87	285	613
$p_1\ [\frac{10^{-6}}{mm^2}]$	0	0.442	-349	-49.2
$p_2\ [\frac{10^{-6}}{mm^2}]$	0	-6.54	-23.7	-35.9
$u_0\ [pix]$	399.5	399.491	399.428	399.427
$v_0\ [pix]$	299.5	299.508	299.447	299.282
$RMSE\ [pix]$	**0**	**0.01424**	**0.11375**	**0.21395**

Table 4.1: Calibration results for simulated images. The proposed PS method is roughly an order of magnitude closer to the ground truth.

Name	Type	Resolution	Pixel Size
D1	ScenicView A24W	1920x1200	0.270mm
D2	SyncMaster 2433LW	1920x1080	0.272mm

(a) Displays

Name	Type	Resolution	Pixel Size
HR	Basler Scout 1390m	1392x1040	4.65µm
LR	Basler A312fc	780x580	8.3µm

(b) Cameras

Table 4.2: Hardware used for experiments

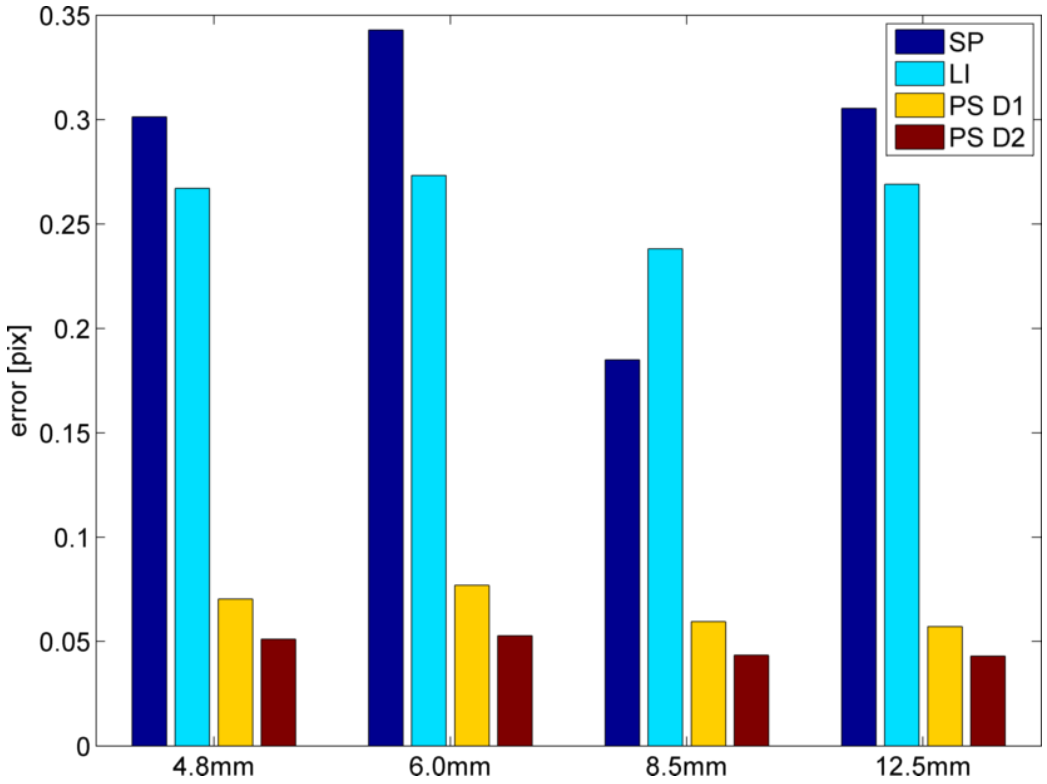


Figure 4.11: Results for the low-resolution camera with different lenses. The LI method is slightly better than SP. The errors of the proposed method are much lower.

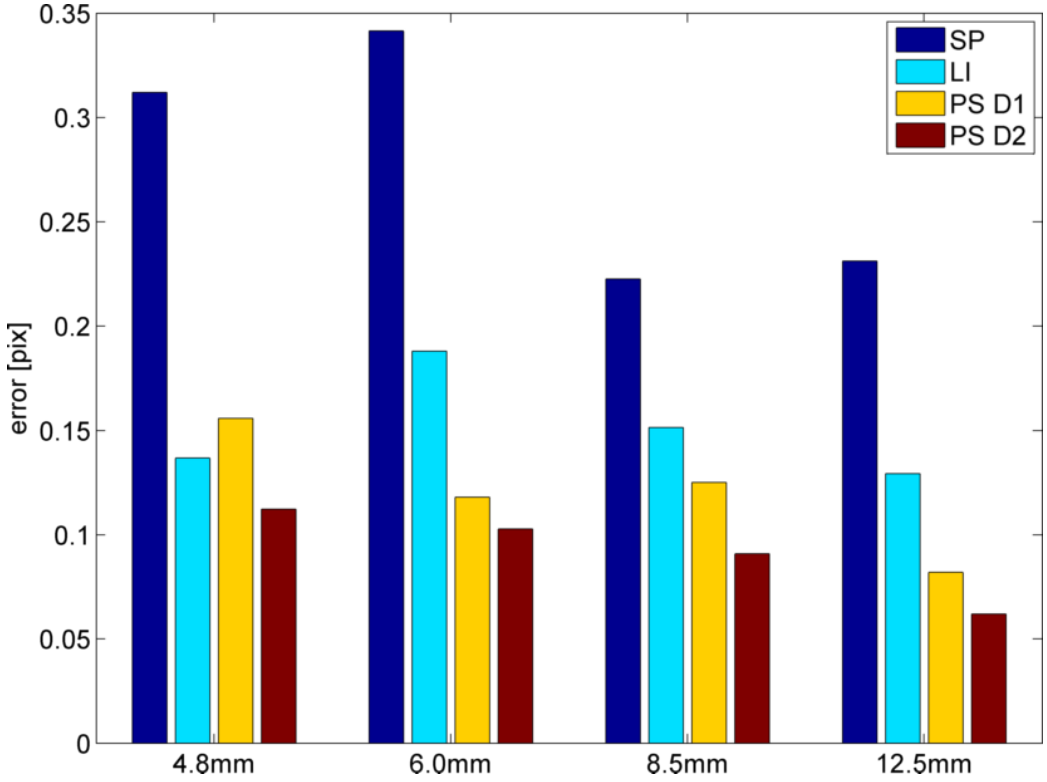


Figure 4.12: Results for the high-resolution camera with different lenses. The errors of the proposed method are slightly lower than the LI method. The SP method performs worst.

The results are shown in figures 4.11 and 4.12. The main conclusion is that PS is the best method for the low-resolution camera by a large margin and for the high-resolution camera by a smaller margin. Compared to the LI method, the reprojection error is between a factor four and five better in the low-resolution case and up to a factor two better in the high-resolution case. Please note that the poses used for the calibrations were not perfectly identical in the different experiments. Differences of a hundredth of a pixel are therefore not significant. However, display 2 consistently gave better results than display 1. The existence of display-specific systematic errors is a topic for further study.

The SP method performs worst. This comes as a surprise, since it is the standard method for users of OpenCV. However, the residuals of the SP calibration show a systematic error. The corner positions are mostly shifted towards the center of the calibration squares, compared to the LI positions (figure 4.13). Fiala and Shu [Fial 10] have identified this effect as related to lighting; it seems to arise from defocusing as well. However, in a "real" checkerboard the shift should cancel out between the two touching corners. The high-quality targets at our lab have isolated squares, so we use only the LI method in the following experiments. In table 4.1, LI was compared to SP on a dense checkerboard pattern. It performed comparably with respect to the reprojection error and better with respect to the ground truth camera parameters, so LI can be used as a reference calibration method.

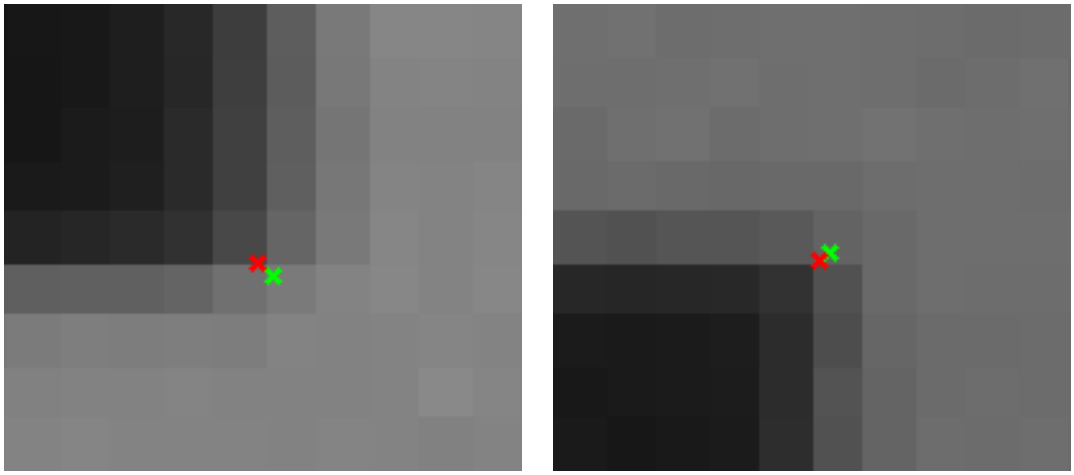


Figure 4.13: Typical corner detection images, enlarged by a factor 20. The green cross marks the LI corner position, the red cross the SP corner position. Left: High resolution image. Right: Low resolution image.

D1	4.8mm	6mm	8.5mm	12.5mm
LR RMS [px]	0.0704	0.0770	0.0596	0.0572
HR RMS [px]	0.1557	0.1180	0.1250	0.0819
LR RMS [mm]	0.5845	0.6392	0.4948	0.4754
HR RMS [mm]	0.7242	0.5488	0.5815	0.3812

Table 4.3: The reprojection error for different lenses with display 1. Expressed in micrometers the values are similar between the low resolution and the high resolution cameras.

The difference in the PS residual error between the HR and LR cameras is approximately in line with the difference in pixel size (table 4.3 and 4.4). This is consistent with a constant size defocus spot on the sensor that depends on the employed lens.

Poses

The choice of camera poses is of course a major factor in the quality of a calibration. We tried to use comparable poses, that is with similar angles to the target. They are shown in figure 4.14. The number of poses does not seem to have much influence.

D2	4.8mm	6mm	8.5mm	12.5mm
LR RMS [px]	0.0512	0.0528	0.0435	0.0432
HR RMS [px]	0.1123	0.1027	0.0908	0.0620
LR RMS [mm]	0.4250	0.4388	0.3611	0.3585
HR RMS [mm]	0.5223	0.4777	0.4224	0.2886

Table 4.4: The reprojection error for different lenses with display 2. Expressed in micrometers the values are similar between the low resolution and the high resolution cameras.

poses	RMSE [px]	$cx\ [px]$	$cy\ [px]$	$f\ [mm]$
7	0.1481	702.9860	526.0807	6.1911
5	0.1474	701.9275	526.4855	6.1961
3	0.1469	701.3100	526.6469	6.1979

Table 4.5: RMS errors for different number of poses and some of the resulting internal parameters. For these calibrations the high-resolution camera with a 6.0 mm lens was used.

HR	8.5mm	D2	LR	4.8mm	D1
number of marks	poses	RMSE	number of marks	poses	RMSE
23238	4	0.09105	10876	4	0.07173
2583	4	0.09085	1739	4	0.07042
466	4	0.09449	435	4	0.06890
224	4	0.08456	106	4	0.06187

Table 4.6: Influence of mark density. For comparison, a typical view of a checkerboard yields around 100 marks.

Table 4.5 shows that the reprojection error barely changes, whether 3, 5 or all 7 poses are used. However, the internal parameters do change. There is no ground truth to compare against, but it seems reasonable to have higher confidence in a calibration result if it is based on more poses.

Mark Density

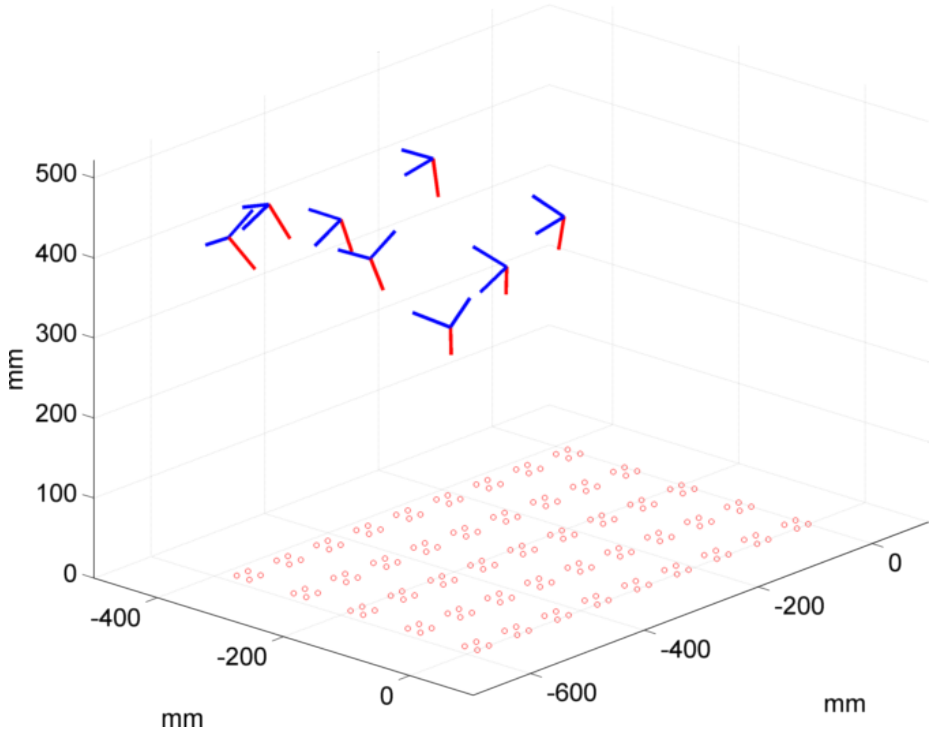
The density of marks generated with the PS approach has little influence (table 4.6). However, as already stated in the previous section, when in doubt there is no reason to avoid using as many marks as possible. Also, for the calibration of a fully generic non-parametric camera model, dense correspondences are important, and can be easily generated with PS.

Defocusing

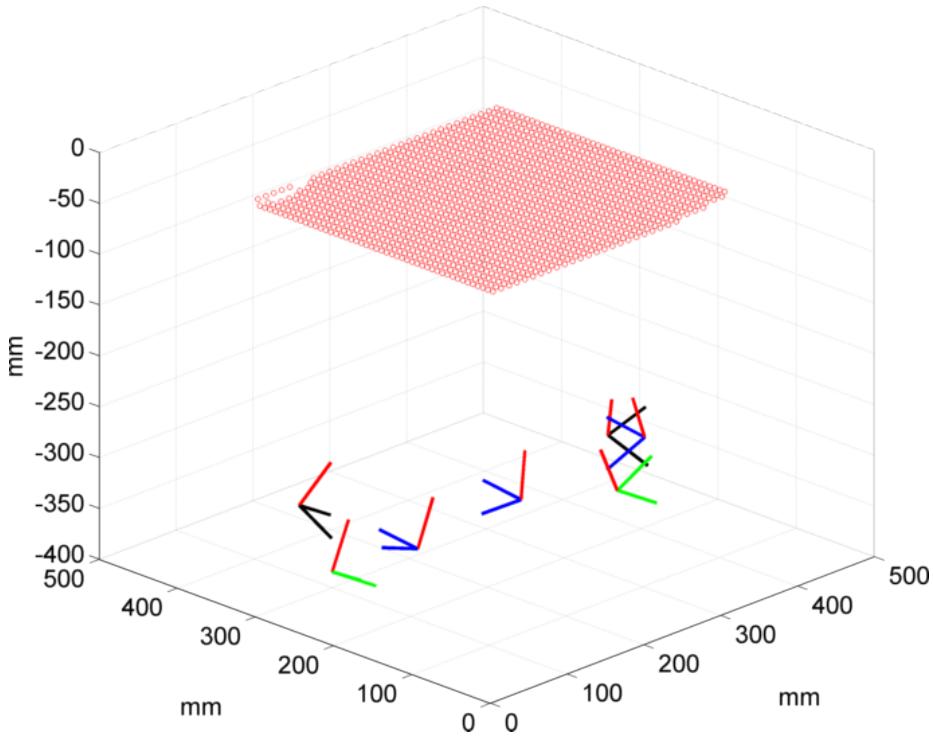
High feature localization accuracy depends on a sharp image. This can be a problem, for example when depth-of-field is limited. PS results are robust against defocusing (table 4.7). The measured phase at a given pixel does not change when the image is blurred, only the contrast is reduced. In fact, PS even profits from a moderate amount of defocusing as aliasing between the display pixel grid and the camera pixel grid is reduced. Therefore it is possible to move the camera close to the display during calibration so that the entire field of view is covered.

Glass Plate Offsets

As mentioned in section 4.3.1, the protective glass plate in front of the display pixels introduces a shift in the apparent mark coordinates. As can be seen in table 4.8, modelling this refraction does indeed result in an improvement of the reprojection error. However, the effect is relatively minor. It is on the order of a few thousands of



(a) Poses in front of the checkerboard target 2.6



(b) Poses in front of the active target. Blue: 1, 2, 3. Green: 4, 5. Black: 6, 7.

Figure 4.14: Example camera poses.

f-stop	RMSE LI [px]	RMSE PS [px]
5.6	0.1834	0.1312
11	0.1367	0.1400

Table 4.7: Robustness against defocusing. These results stem from the high-resolution camera with a 4.8mm lens.

RMSE [px]	4.8mm	6.0mm	8.5mm	12.5mm
D1+LR	0.0755	0.0825	0.0692	0.0598
D1+LR+glass	0.0704	0.0770	0.0596	0.0572
D1+HR	0.1592	0.1168	0.1286	0.0846
D1+HR+glass	0.1557	0.1180	0.1250	0.0819
D2+LR	0.0523	0.0548	0.0449	0.0435
D2+LR+glass	0.0512	0.0528	0.0435	0.0432
D2+HR	0.1161	0.1048	0.0925	0.0625
D2+HR	0.1123	0.1027	0.0908	0.0620

Table 4.8: Improvements in the RMS reprojection error by modelling the ray offsets introduced by the glass cover of the display. They were evaluated for different lenses and displays.

a pixel only, while the mark offsets are up to 0.1 pixels in lateral direction. This is because the shifts can be partially compensated by the camera distortion parameters.

Stereo calibration

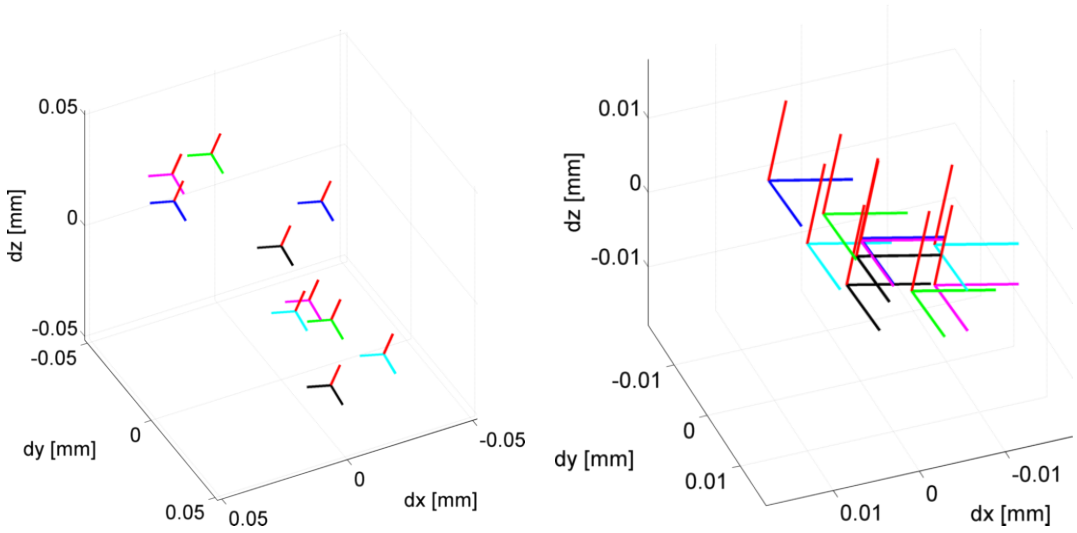
As also noted by Albarelli et al. [Alba 09], a lower reprojection error does not automatically imply a more correct calibration. Therefore we tested the proposed method further. We performed a stereo calibration and subsequently triangulated the positions of the calibration marks. We then compared the known positions of the marks to the triangulation results. Since the display was positioned closer to the camera than the checkerboard target, we also normalized the errors. The stereo rig used consisted of two Basler A312fc cameras (the low resolution model in the experiments above) with 8.5mm lenses and a baseline of approximately 150mm. Four poses were used. The checkerboard target yielded 169 marks visible in both cameras, the active target yielded 1219 marks to triangulate. As can be seen in table 4.9, the deviations are much lower for the proposed Phase Shift calibration. The accuracy is improved approximately by a factor of five, which is consistent with the results of the monocular calibration.

	error [mm]		normalized error [mm/m]	
	mean	sigma	mean	sigma
PS	0.0299	0.0175	0.1153	0.0643
LI	0.3028	0.2286	0.5528	0.3947

Table 4.9: Stereo Triangulation Results. The error for the proposed PS technique is approximately one fifth of the error resulting from the LI method.

	σ_x [mm]	σ_y [mm]	σ_z [mm]
classic	0.0146	0.0173	0.0138
active	0.0014	0.0023	0.0060

Table 4.10: Standard deviations of the translation parameters for a classic and an active target for 10 repeated external calibrations.



(a) Poses recovered from the classic target (b) Poses recovered from the active target

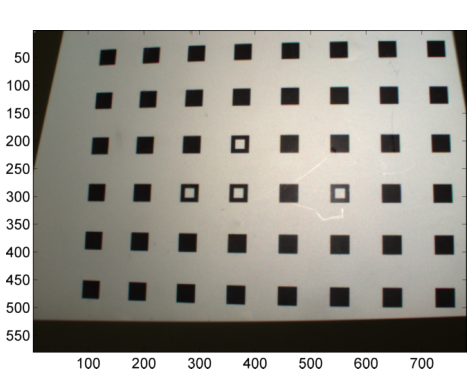
Figure 4.15: Repeatability of external calibration. The camera indicators have the same size, but the scales are different. The red lines indicate the camera z-axis, the different colors for the x and y-axes are for better visual differentiation.

Repeatability

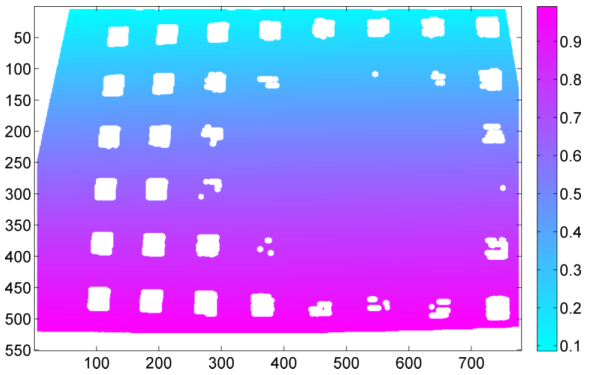
Another test for the proposed calibration method is repeatability. We performed ten external calibrations of a pre-calibrated Basler A312fc camera with a 12.5mm lens. Purely external calibration has the advantage that a single view of the calibration target suffices, so no parts of the setup had to be moved. A classic feature-based target and an active target were used. The resulting poses are plotted in figure 4.15. The standard deviations of the translational parameters are shown in table 4.10 (the noise in the rotation parameters was very close to zero). The mean offsets from the mean position were 5.3µm for the active calibration target and 22.1µm for the classic target. The absolute distances to the respective calibration targets were practically equal with 258mm and 256mm.

Summary

There are many variables that influence the quality of a calibration, from the choice of camera poses to the tuning of algorithm parameters. Additionally, errors are often compounded, so the source of problems is not always obvious. The calibration method with active targets has several advantages: It is fully automatic, no user interaction to identify marks is necessary. Digital displays are highly accurate targets, so there is no



(a) Reference plane with marks for external camera calibration.



(b) Projector phase map φ_y, as seen by the camera. The hotspot effect makes some of the reference marks bright enough to be usable while others are below the threshold.

Figure 4.16: Projector phase shift calibration. The target with reference marks defines a plane in space. Virtual marks with known projector image coordinates can be defined by intersecting camera rays with the target plane.

need for costly target validation. The only input parameters are the display resolution and the pixel size, both of which are exactly known. The method is robust against defocusing. Lastly and most importantly, the achievable accuracy is very high. One possible disadvantage is that the active calibration requires multiple images per pose and thus cannot be performed with a hand-held camera. However, we are of the opinion that the additional accuracy over a classic feature-based calibration is worth the effort for tasks like precise 3D reconstruction. The final accuracy of a complete 3D scanning system calibrated with the proposed approach is evaluated and compared to a 'classic' calibration in section 6.1.2.

4.3.2 Projector Calibration

As mentioned before, projector calibration can be treated as a reverse camera calibration. We can use the same pinhole model and calibration algorithms as for cameras. We project a known pattern, so the image coordinates of the marks are known. To recover the world coordinates of a mark, we need a calibrated camera and a suitable calibration target. The target defines a plane in space. The intersection of the camera ray to a projected feature with the target plane gives the 3D camera coordinates of the feature. Unfortunately, right now it is still necessary to use the traditional checkerboard patterns for projector calibration. Digital displays, as proposed in the previous section, currently do not have the diffuse surface required to observe a pattern projected onto the display. Therefore a hybrid approach must be used: Calibrate the camera with an active target, use a checkerboard to define a plane in space, and use phase shifting to create projector marks. Once e-paper technology matures, the checkerboard will be no longer necessary and a purely active calibration can be realized.

Note that a static projector cannot display the required phase-shift sequence. In that case it is necessary to embed calibration marks in the measurement pattern. The

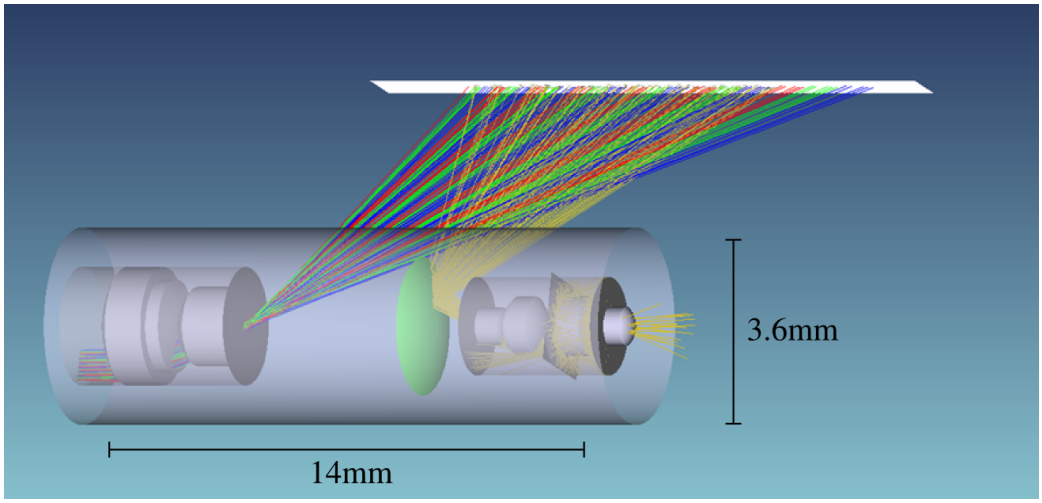


Figure 4.17: Schematic cross-section of the endoscopic camera. The light for the projector is supplied through a glass fiber (not shown). A ring-shaped slide projects colored cones to the side (not all rays are shown). They are observed by the camera via a curved mirror (shown in green). A second camera provides a front view (yellow rays to the right). The refraction of the rays in the glass tube is ignored in this diagram.

number of marks that can be included without perceptible performance degradation is limited. There is, however, a less general calibration approach that does not require embedded calibration marks in the measurement pattern. In a standard setup the edges of the stripe pattern form a pencil of "light planes" in space. More correctly, when lens distortion is taken into account, each edge corresponds to a ruled surface. For each view of the measurement pattern on the calibration body, we can simply decode the pattern and generate marks on the known target plane in space. The data from from several views can then be used to fit the parameters of the light planes. The difference to the "full" calibration is that for dedicated calibration marks both the u and the v component of the image coordinates are known, while for a stripe pattern only one component can be recovered. Therefore we can not formulate an explicit image distortion model, but that is unneccessary if we know the light planes. It must be noted, though, that the light planes are not actually planes in the presence of distortion. If the distortion is small, the planarity assumption can be applied iteratively (see section 6.1). A different approach will be used in section 4.3.4 for the calibration of the endoscopic projection system, where the projected surfaces are cones.

4.3.3 Endoscopic Camera Calibration

The endoscopic 3D scanner described in [Schi 11] uses a miniature catadioptric camera system. This wide-angle system consists of a pinhole camera combined with a spherical mirror, mounted inside a protective glass tube. The setup is shown schematically in figure 4.17.

parameter type	number
camera pose	6 per pose
focal length	1
principal point	2
radial distortion	3
tangential distortion	2
mirror position	3
mirror radius	1
tube origin	2
tube rotation	2

Table 4.11: Camera calibration parameters. The first five types are used in the pinhole model with radial and tangential distortion. The augmented model with mirror and glass tube has additional parameters, listed in the last four rows. Because of spherical symmetry, the mirror rotation is irrelevant. Also, the rotation of the tube around its axis is degenerate. The origin of the tube is defined as the intersection point of its center line with the arbitrary plane z=0 and thus needs only two parameters. The inner and outer tube radius, as well as its index of refraction, are known and therefore not part of the optimization at all.

The combined catadioptic system with mirror and glass tube is a non-single-viewpoint camera, so the pinhole model does not strictly apply. There are two choices to be made for the calibration. They concern the camera model and the error metric for the parameter optimization. Two different camera models were used. One is the well known Zhang model [Zhan 00] for a pinhole camera with radial and tangential distortion. However, since the camera is actually a wide-angle catadioptric system that does not have a single viewpoint, using Zhang's model is an approximation at best. Therefore we created a more elaborate model that combines a pinhole camera with a mirror and an encasing glass tube. With this so-called pinhole+mirror+tube model the reflection and refraction processes that each ray undergoes can be explicitly calculated. The necessary mathematical relations can be found in the appendix. Zhang's pinhole model has 8 intrinsic parameters and 6 extrinsic parameters per pose. The pinhole+mirror+tube model has 8 additional intrinsic parameters. See Table 4.11 for a complete list of the calibration parameters for each camera model.

The model parameters are determined with a Levenberg-Marquardt optimization [More 78]. There are three different error metrics that may be used.

- The distorted reprojection error is the distance between the image coordinates of the observed marks and the projected and distorted image coordinates generated from the mark world coordinates by the model. It is measured in pixels. The distorted error is not a good metric in our case, as the strong fisheye distortion "compresses" the image border. Thus points near the border have relatively small residuals in any case and the points towards the image center get a relatively higher weight in the optimization.

- The undistorted reprojection error is the distance between the observed and undistorted image coordinates and the image coordinates obtained by project-

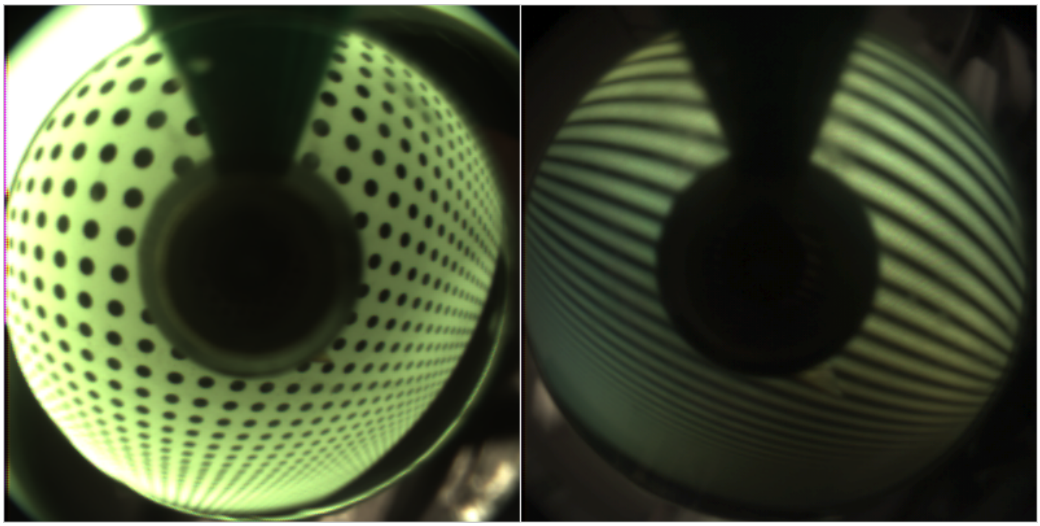


Figure 4.18: Views of the 'classic' dot grid target (left) and one pattern of the proposed active approach. The dark area in the center is caused by the sensor chip looking at itself in the mirror. The dark area at the top is the shadow of the camera's data cable.

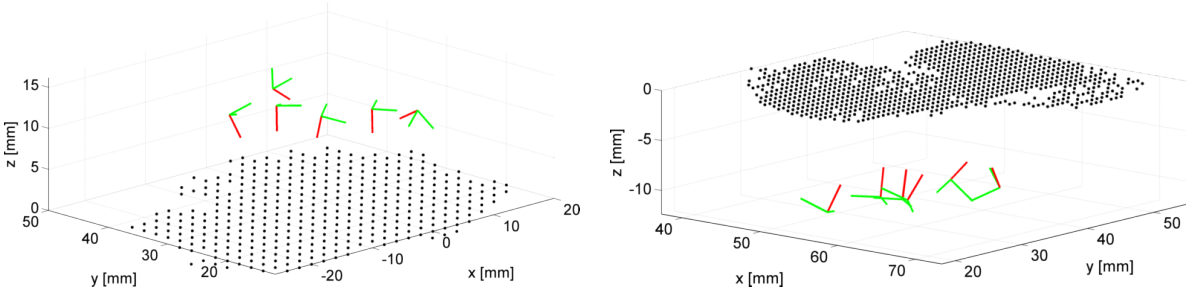
ing the 3D mark world coordinates onto the image plane. It is also measured in pixels. The undistorted error treats all points equally.

- The object space error is the distance between the ray of view for a given image coordinates and the corresponding mark in camera coordinates. It is measured in *mm*. The object space error is a good metric only if all calibration points are at approximately the same distance from the camera. Points that are farther away have larger residuals and therefore a higher weight in the optimization than points close to the camera.

To generate the necessary calibration marks, both an active and a classic calibration target were used. As the system is designed for object distances of only a few millimeters, an active target must have very dense pixels so that they will not be resolved individually. We used the 5.6 inch 1280x800 display of a Fuijitsu UH900 mini-notebook with a pixel size of 94µm for the active calibration. The classic target was a dot grid from Edmund Optics (no checkerboard target in the required size was available). The marks were localized using ellipse fitting, but especially in the peripheral area this is difficult. An example input image can be seen in figure 4.18. For each target type, a set of six different poses were used (figures 4.19). We took care to ensure that the poses were comparable.

The calibration results can be seen in tables 4.12 and 4.13. Several observations can be made:

- The active target gives lower errors than the classic target across the board. Besides that, the target type does not influence the relative performance of the different calibration methods.



(a) Poses for the classic calibration

(b) Poses for the active calibration

Figure 4.19: Poses used in the calibration of the endoscopic camera. For each method six different poses with a comparable range of rotations and translations were used. The black dots symbolize the calibration marks. Not all marks were observed in all poses.

- The more elaborate model with mirror and tube gives an improvement over the simpler pinhole model. This is true even when default parameters are used for the mirror and the tube and only the pinhole parameters are optimized.
- Full optimization of all parameters of the pinhole+mirror+tube model yields a small additional improvement of the errors. The recovered parameters are within manufacturing tolerances of their design values.
- In the pinhole model, optimizing the distorted error leads to unacceptably large undistorted errors. This is due to the extremely high radial distortion. The polynomial model tends to diverge at the image borders and can hardly be inverted. Small errors in the distorted error thus give rise to large undistorted errors. The effect is less severe in the pinhole+mirror+tube model, as a large part of the distortion in this case is explained by the reflection on the sphere.
- The optimization of the image plane errors fails with the full parameter set of the pinhole+mirror+tube model. The final parameters are implausible, and the errors are high. There is also a considerable dependence on the initial parameter values, leading to the conclusion that the minimization algorithm gets stuck in local minima. The use of alternative optimization algorithms is a topic for future work. In contrast, minimization of the object space error works, so it seems to be the best metric in this case.
- Optimization of the object space error in the pinhole+mirror+tube model gives even lower distorted and undistorted errors than directly optimizing them.

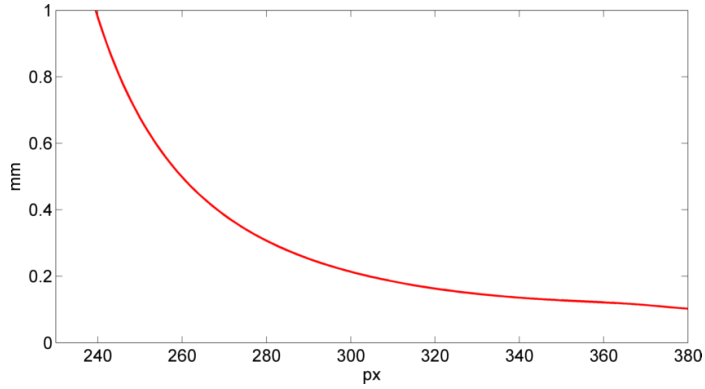
All in all, the optimization of the object space error in the pinhole+mirror+tube model seems to be the best choice. The absolute values of the object space errors are very much in line with the general resolution of the camera system. To estimate the object space resolution, two virtual planes were placed at a distance of 10mm from the camera. For each pixel on a horizontal line through the camera image the viewing ray and its intersection with the test planes was calculated. The distances between neighboring intersection points are a measure of the achievable spatial resolution. The results are shown in figure 4.20. The resolution is in the given planes

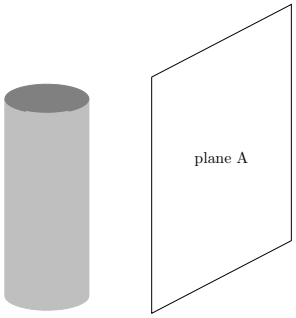
model	error metric	resulting RMS error		
		distorted [px]	undistorted [px]	object space [mm]
pinhole	distorted	0.7855	2.3796	0.0732
pinhole	undistorted	0.9974	1.3728	0.0820
pinhole	object space	0.8133	1.4882	0.0674
pmt default	distorted	0.7739	0.9929	0.0661
pmt default	undistorted	0.7633	0.9361	0.0635
pmt default	object space	0.6867	0.8612	0.0546
pmt optimized	distorted	n/a	n/a	n/a
pmt optimized	undistorted	n/a	n/a	n/a
pmt optimized	object space	0.6454	0.8196	0.0501

Table 4.12: Endoscopic camera calibration results for the active target. The full model with pinhole camera, mirror and glass tube (abbreviated pmt) is better than the simple pinhole model, even if default parameters for mirror and tube are used. The optimization of the image plane errors failed in the full model. Using the object space error metric gives the most balanced results.

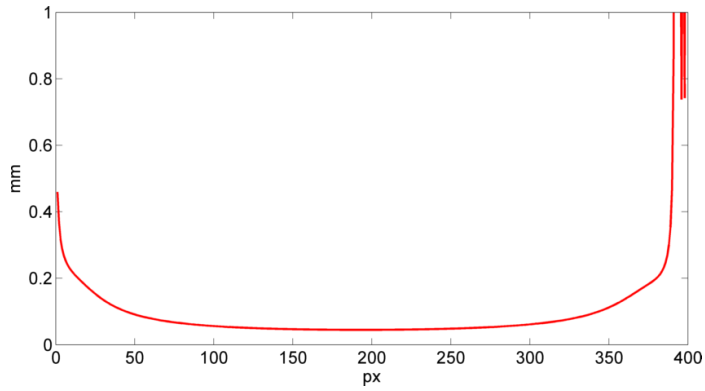
model	error metric	resulting RMS error		
		distorted [px]	undistorted [px]	object space [mm]
pinhole	distorted	0.8216	2.7762	0.0978
pinhole	undistorted	0.9506	1.5127	0.0996
pinhole	object space	0.8559	1.6570	0.0864
pmt default	distorted	0.8402	1.2069	0.0899
pmt default	undistorted	0.8219	1.1355	0.0883
pmt default	object space	0.7884	1.0472	0.0802
pmt optimized	distorted	n/a	n/a	n/a
pmt optimized	undistorted	n/a	n/a	n/a
pmt optimized	object space	0.7166	0.9266	0.0707

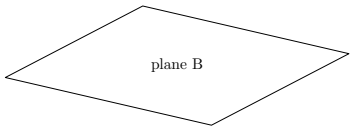
Table 4.13: Endoscopic camera calibration results for the classic target. The relative performance of the different methods is similar to Table 4.12, except that all errors are slightly higher. Again the full model with the pinhole camera, mirror and glass tube (abbreviated pmt) is better than the simple pinhole model, even if default parameters for mirror and tube are used. Again the optimization of the image plane errors failed in the full model. Again, using the object space error metric gives the most balanced results.





(a) Resolution in plane A. Since only the rays in the right half of the image intersect the plane, the x-axis does not start at zero.





(b) Resolution in plane B. The outliers at the very right are caused by the breakdown of the distortion model. There were no calibration marks in this area and the fitted polynomial cannot be inverted at this radius. The principal point was at approximately 190px, so this problem is not symmetrical.

Figure 4.20: Endoscopic object space resolution in two test planes at a distance of 10mm from the camera.

is typically between 100µm and 200µm, but much worse at the image borders, where the distortion is very high. An RMS object space error of 50µm therefore can be considered a good result.

4.3.4 Endoscopic projector calibration

The projector in the endoscopic sensor is of the static type. It can only project a single pattern. Furthermore, the pattern does not contain any explicit calibration marks (figure 4.21). Therefore the projector calibration method outlined in the previous paragraph was used. The ring-shaped pattern slide produces nested cones which can be explicitly parameterized. We used the Levenberg-Marquardt algorithm [More 78] to perform the cone fit. Again there are different error metrics that can be used. One is the distance between the calibration point and the intersection of the ray of view for the calibration point with the cone. The other is the orthogonal distance of the

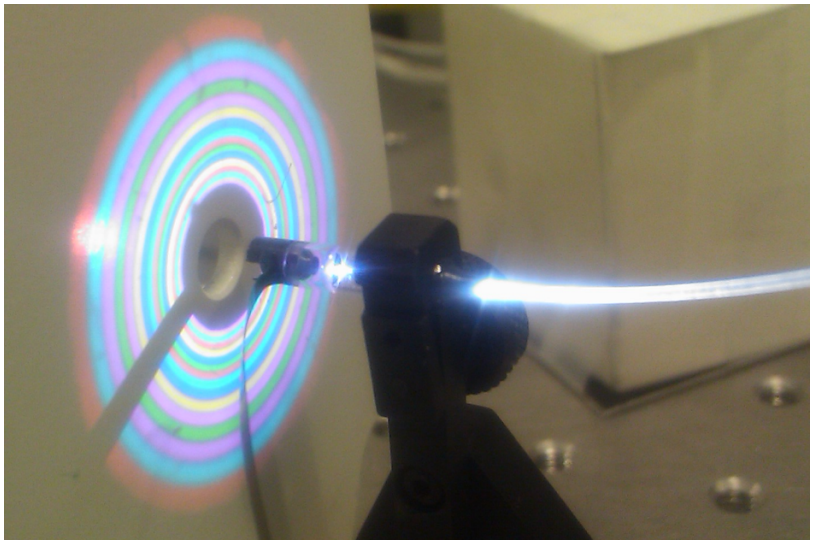


Figure 4.21: The endoscopic structured light system in operation, seen from the outside. The outer rings are wider to compensate the image distortion in the camera.

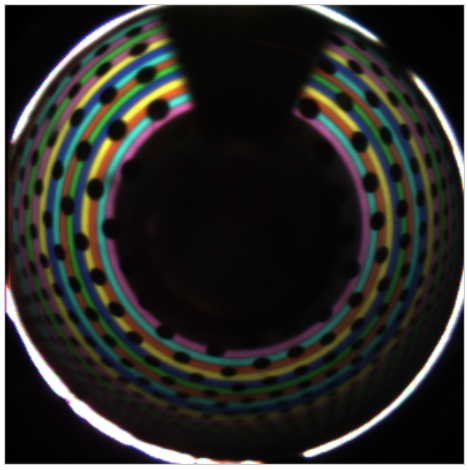


Figure 4.22: Endoscopic image of calibration target with projected rings. The dark area in the center is caused by the sensor chip looking at itself in the mirror. The dark area at the top is the shadow of the camera's data cable.

calibration points from the cones. We tested both, and found that the former gives better results, as it mirrors the way the final depth data is computed.

Figure 4.22 shows an example input images with calibration marks (to determine the pose relative to the camera) and projected rings (to generate sample points from each ring for the cone fit). In figure 4.23 the accumulated data for a given cone from four different viewpoints is plotted. Figure 4.24 shows some of the resulting nested cones with different opening angles.

Numerical instabilities may occur during the fit. Instead of optimizing each cone separately we can therefore constrain all cones to a common axis and common vertex. The x and y components of the vertex position are also initially fixed to zero, that is, on the optical axis of the camera. The only free parameters for each single cone is the opening angle. In a second iteration we allow the common vertex position to

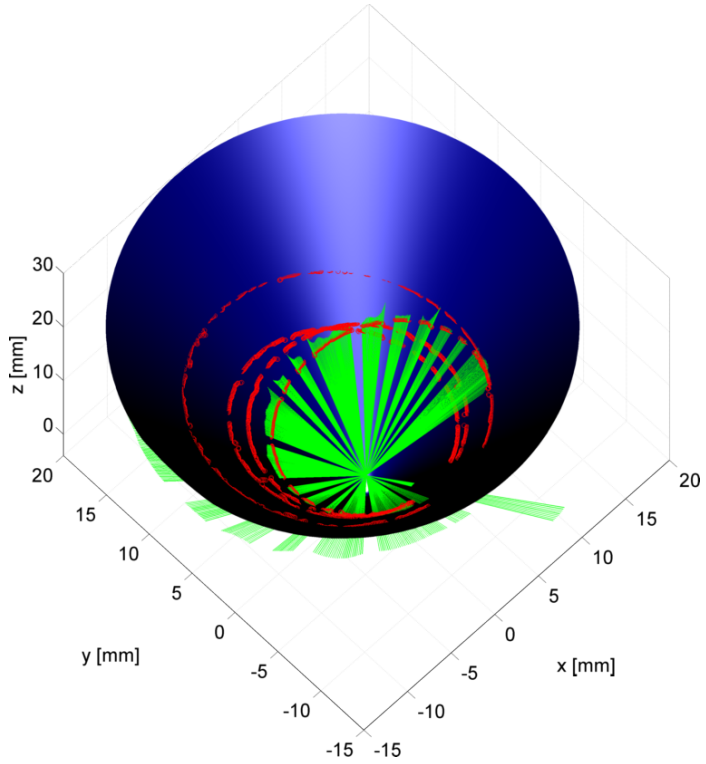


Figure 4.23: Data points (red) from four poses of the calibration target and fitted cone (blue). Example camera rays for one pose (green) indicate the fitting errors. The viewing rays originate from the approximate virtual single viewpoint of the camera.

deviate from the optical axis. This is necessary because of manufacturing tolerances. Additionally, we introduce an individual z-position for each cone. As can be seen in figure 4.25, refraction in the glass housing of the scanner leads to a z-offset that depends on the opening angle of the cones. An estimation with the help of equation 4.1 assuming a glass wall with a thickness of 0.4mm shows that an offset of 0.02mm is to be expected between cones with opening angles of 34 and 42 degrees. While not negligible, this is relatively small compared to the errors from other sources, for example imperfect camera calibration. Still, the effect can be detected in the optimized parameters (figure 4.26).

The results of the optimization procedure can be seen in figures 4.28 and 4.29. The quality measure is the RMS object space error. For the outermost cones (0 to 3) and innermost cones (15 and 16) there is not enough data for a reliable fit (figure 4.27). Nevertheless, the single cone optimization recovered parameters for cones 2 and 3, but the parameters are completely implausible. The results of the simultaneous optimization of all cones with a common vertex and axis are more reliable, even if the errors are slightly higher. The projector calibration for a camera calibrated with the classic method performs comparatively well, considering the results of section 4.3.3. However, this may be an artifact of the experimental setup. The classic projector calibration used the exact same poses of the classic target that were used for the camera calibration. The 'active' projector calibration consists of a camera calibration with an active target and a projector calibration with a classic target. Therefore two different target types had to be used, and the camera calibration poses

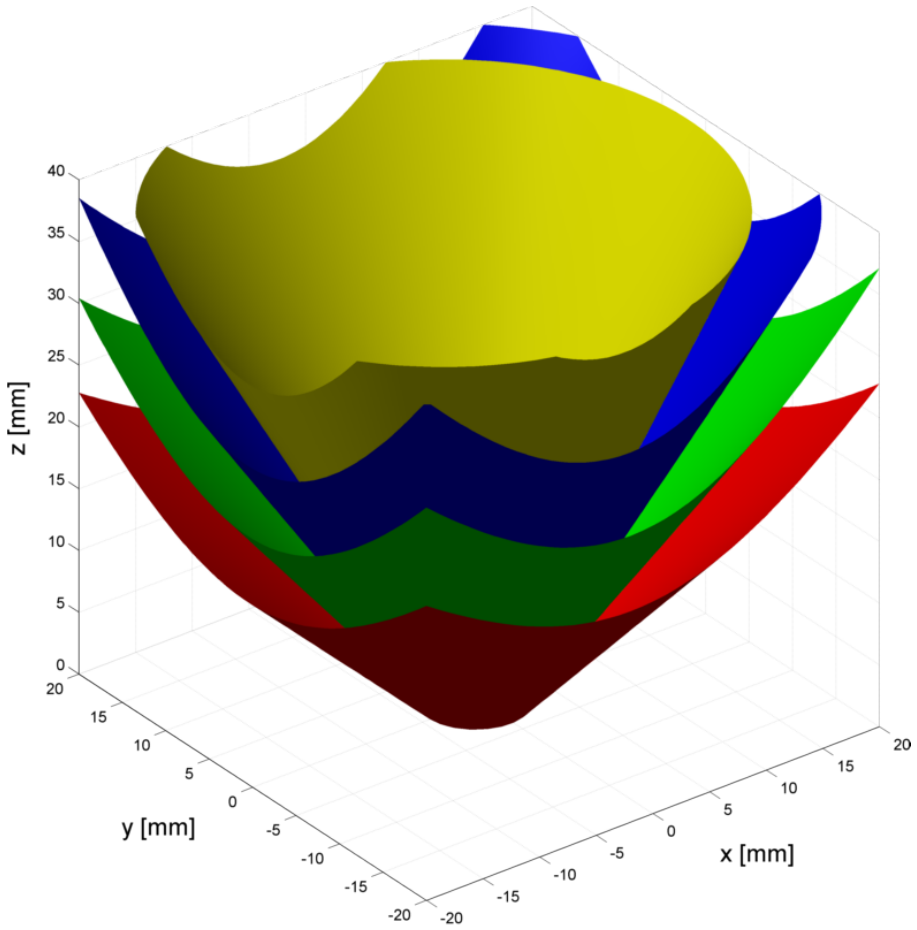


Figure 4.24: Some light cones of the endocam system

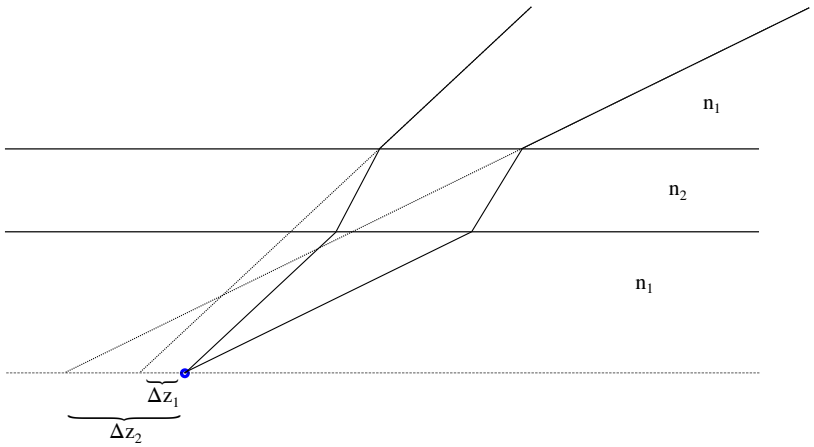


Figure 4.25: Schematic cross section of the endoscopic pattern projector. Refraction causes a shift in the apparent z-position of the cone vertices. The magnitude of the offset depends on the cone's opening angle.

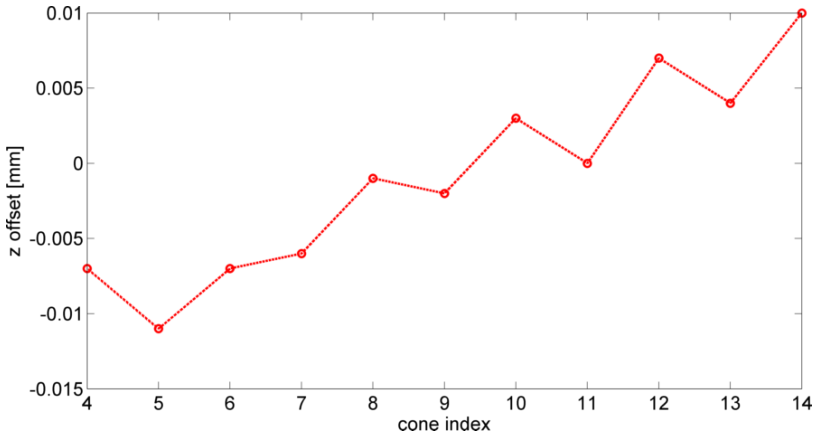


Figure 4.26: Offsets in the z-direction for different cones after numerical optimization of the model parameters (relative to the average position). The rising trend and magnitude correspond to the expected values. .

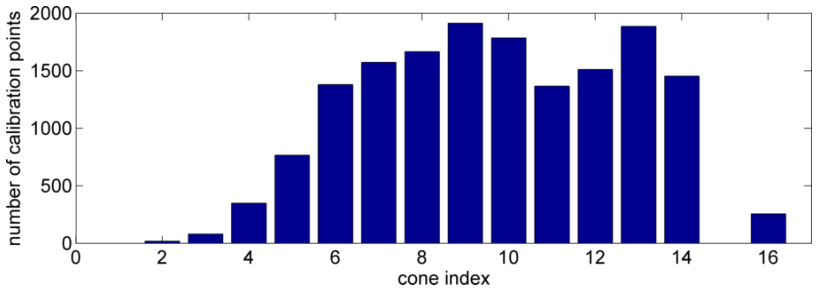


Figure 4.27: Number of calibration points per cone. The inner and outer rings have very few points, if any, and can not be expected to yield a reliable calibration result.

were not the same as the poses used in the projector calibration. This introduces additional errors, so the active calibration could not realize its full potential. Fully active projector calibration will only be possible with an e-paper calibration target.

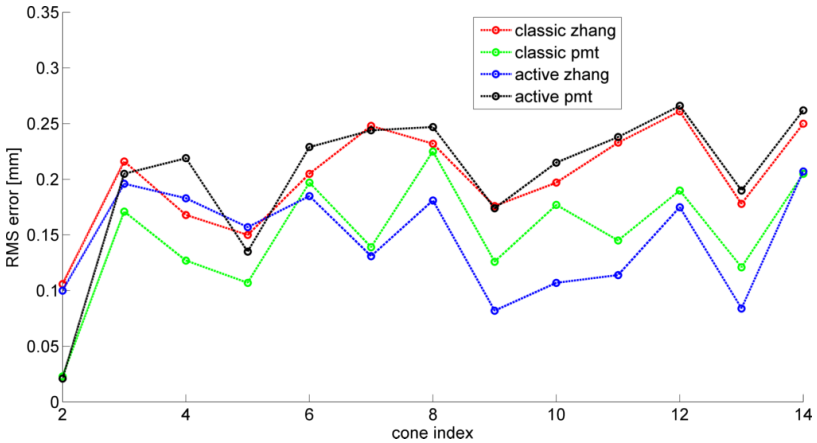


Figure 4.28: Results for single cone optimization with four different camera calibrations (see section 4.3.3).

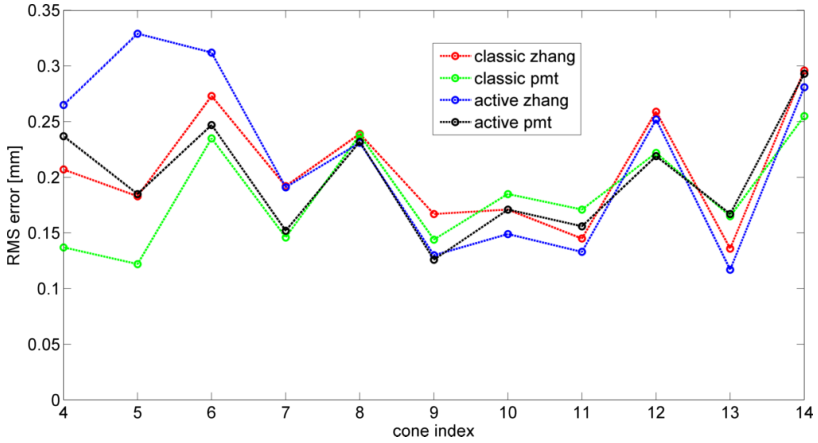
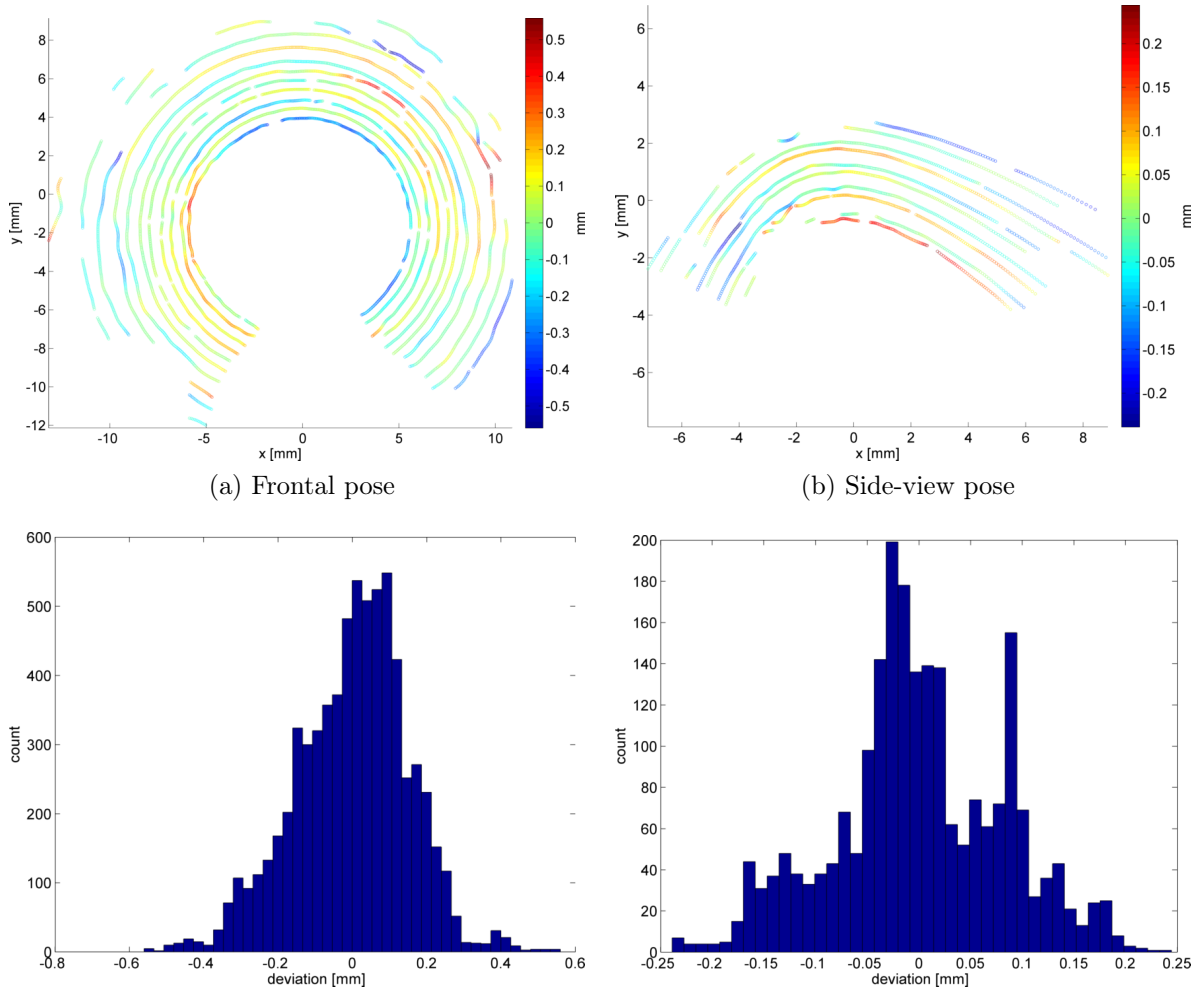


Figure 4.29: Results for simultaneous optimization of all cones with common vertex and axis for four different camera calibrations (see section 4.3.3)

To judge the quality of the resulting calibration a planar object was measured in two poses similar to the virtual planes shown in figure 4.20. The calibration was based on the 'pinhole+mirror+tube' model with a classic dot grid target, optimizing the object space error for the camera and the distance along the ray of view for all cones of the projector.

The resulting depth errors are plotted in figure 4.30. The standard deviation of the measured depth values from the plane was 153µm for the frontal pose respectively 88µm for the side-view pose. Due to the axial setup of the projector and the camera, the triangulation angle in the forward direction is small, between 2 and 10 degrees. Therefore the accuracy is lower in the frontal pose. Compared to the cone fit errors seen in figure 4.29 the errors on the planar target are relatively low. However, there is a systematic component to the errors that depends on the ring number. Correcting this error is a topic for future work.



(a) Frontal pose

(b) Side-view pose

(c) Error distribution in the frontal pose

(d) Error distribution in the side-view pose

Figure 4.30: Evaluation of the endoscopic calibration. The colors indicate the deviation from the plane. In the frontal pose on the left the RMS error is 153µm, in the side-view pose on the right it is 88µm.

Chapter 5

Graph-Based Pattern Decoding

Next to the choice of the pattern, the decoding algorithm is the most important part of an S3L system. We propose a graph-based decoding scheme built on a superpixel segmentation of the input image. The overview of the chain of processing steps for each image can be seen in figure 5.1. Next to the robustness and the accuracy of the recovered depth data, processing speed is also a concern. The graph traversal steps are not very suitable for parallelization but the image processing part can be implemented on a GPU. The individual processing steps are described in detail in the next sections.

The influence of various parameters and processing steps is evaluated with the help of various test datasets, consisting of video sequences acquired with a scanner submersed in a liquid-filled pig stomach. They were acquired in the course of a feasibility test for endoscopic measurements on biological tissue, when an endoscopic sensor was not yet available. The liquid was necessary to keep the stomach from collapsing, but it also provided an interesting test case of severe image artifacts in the form of bubbles. The representative image sequences used for the evaluation are numbered 2, 3 and 6. They have different image quality with respect to contrast, bubbles and similar disturbances. Sequence 2 is best, 3 is worst and 6 is intermediate. All sequences use the same illumination pattern with 8 colors and a length of 104 stripes. This pattern was not ideal as it contains some non-normalizable code words. Unfortunately no other pattern slide was available at the time. See section 4.2 for more details on the pattern design. Additionally we use a video sequence acquired in the ear canal of a test person with a prototype endoscopic 3D scanner. The pattern in this case consists of 15 concentric rings in 6 colors.

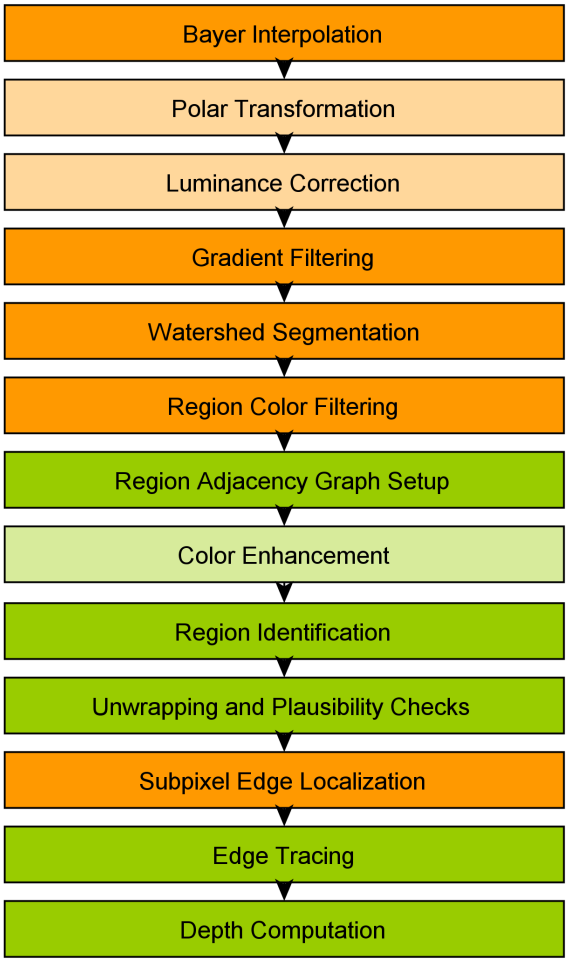


Figure 5.1: Processing steps applied to each input image, starting from the raw camera output. The orange steps can be easily parallelized and are implemented on the GPU as well as on the CPU. Green steps are performed on the CPU only. The desaturated colors indicate optional steps.

5.1 Superpixel Representation

Our aim is to identify the color stripes in the camera image. We observe that the pixel representation is redundant: many adjacent pixels have approximately the same color. We can therefore reduce the complexity of the image representation by using superpixels instead. The Watershed Transform offers a computationally efficient way to achieve this. The advantages of the watershed transform are that it is unsupervised, parameter-free and fast. It is, however, a low-level method that produces severe oversegmentation. For our system this is immaterial: The goal is to represent the image with superpixels of approximately uniform color and thus markedly reduce the image complexity. This, in turn, allows us to use graph-based decoding algorithms in real-time. An additional advantage of superpixel-based representations is that properties like color are defined as statistics over an ensemble of 'simple' pixels, thus reducing the effects of defocus blur and noise.

Preprocessing

The input for the watershed transform is the magnitude of the gradient of the input image. Depending on the quality of the input images, several preprocessing steps may be necessary to obtain a good basis for the segmentation.

1. Bayer Interpolation. Industrial digital cameras usually provide the raw image data, that is, without demosaicing and other potentially interfering quality improvements performed by consumer cameras. There are many highly advanced algorithms for Bayer Interpolation [Li 08b, McGu 08, Dubo 05, Mure 05]. As the accuracy of our 3D measurements crucially depends on the exact location of edges in the image, we prefer to perform as little manipulation of the raw image as possible. Therefore we use a simple linear interpolation scheme to recover the color image. This has the additional advantage of being the fastest method. Tests with more advanced algorithms did not improve the decoding performance.

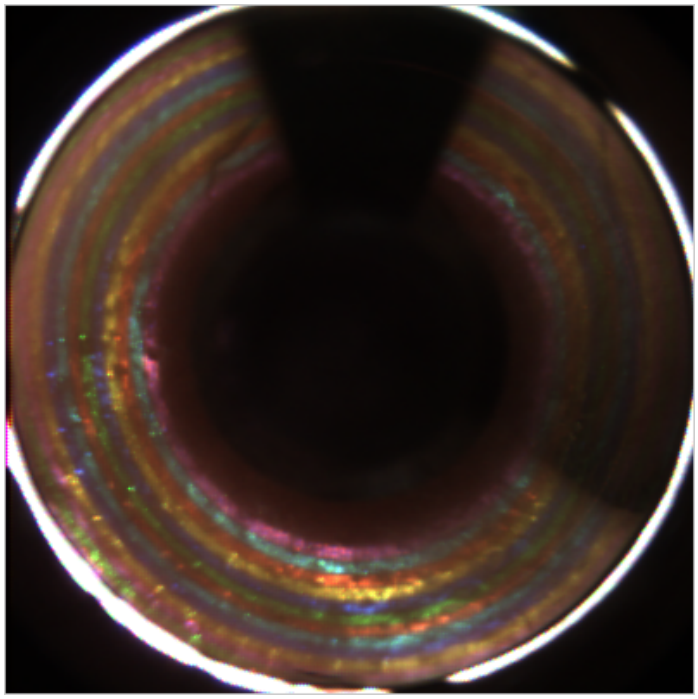
2. Polar Transformation. In the endoscopic sensor, the pattern consists not of color stripes but rings. To facilitate the gradient calculation and all subsequent processing steps, we transform the image to polar coordinates and obtain stripes which are approximately linear. This makes it possible to use the same fast gradient filters in the x (respectively r) and y (respectively ϕ) direction as in the case of natively linear stripes. Otherwise the gradient would have to be computed in directions which are not aligned with the pixel grid and moreover are not constant across the image.

3. Luminance correction. Some types of input images exhibit small specularities, which cause spurious edges in the image. To suppress these brightness changes, the image is first transformed to the HSL color space. The L component is then fixed at a certain value, typically 0.5. Finally, the image is transformed back to RGB color space. The resulting image does not look very realistic, but it is suitable for our purposes. Highlight elimination is of course a topic of research in its own right [Lin 08, Arno 10]. However, we found the simple luminance correction to be quite effective for our limited problem scope. The technique can not be applied to patterns containing white and black stripes, as those differ only in luminance.

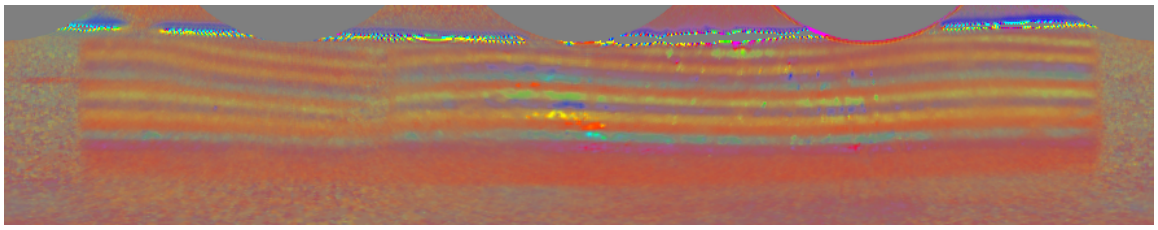
4. Smoothing. As differentiation emphasizes the noise in an image, applying a smoothing filter before the gradient filter is almost always advisable.

Example images illustrating the polar transformation and luminance correction steps are show in figure 5.2.

As a side note, we also evaluated a "leveling" preprocessing step as proposed in [Meye 04]. A morphological closing filter can be applied to the image before the watershed transform. This fills in shallow potential basins, which may be spurious. The remaining basins are thus, ideally, more meaningful. While the reduced number of basins led to a minor speedup of the graph traversal part of the decoding, the number of recovered depth values did in fact decrease. Leveling is therefore not used.



(a) Original image



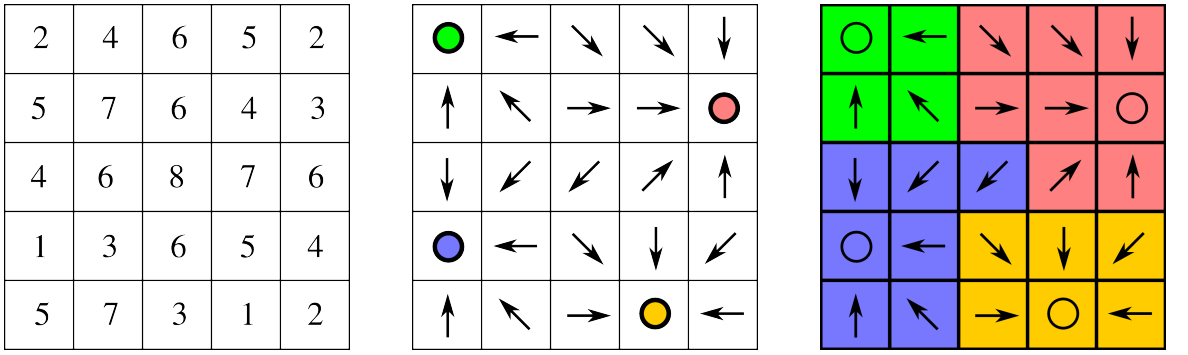
(b) Result after polar transformation and luminance fixing. The color fringes at the top are artifacts caused by saturated pixels whose hue could not be recovered.

Figure 5.2: The input image with a color ring pattern is tranformed to polar coordinates. The luminance is set to a constant value to eliminate small specular highlights.

The last preprocessing step is of course the gradient computation. There are different algorithms for extracting gradient information from an image. The most popular is probably the 3×3 Sobel filter. We also tested Savitzky-Golay filtering (with kernel size 5 and order 3) and the Shen-Castan (ISEF) filter. All in all, the Sobel filter performed best because of its small kernel size. It works reliably even for tightly spaced edges. We compute the gradients in the x and y directions in all three color channels. The L_2 norm of this six-dimensional vector in each pixel forms the gradient magnitude image which is used as the basis for the watershed transform. All the preprocessing steps are well-suited for a GPU implementation and require only a few milliseconds to execute.

5.1.1 Watershed Segmentation

The watershed transform is fast compared to other segmentation methods like Normalized Cuts. There are two basic variants of the algorithm: immersion and rain-



(a) Input image. The values typically are gradients of the image to be segmented.

(b) Arrowing. Each pixel is assigned an arrow pointing to its lowest neighbor. If no lower neighbor exists, the pixel is marked as a seed.

(c) Chaining. For each pixel, we follow the arrows until a seed is reached. The set of pixels ending up at the same seed forms one superpixel.

Figure 5.3: Basic steps of the fast watershed segmentation on a toy image.

falling. The latter does not need a sorting step of all the pixels in the image and thus has a speed advantage. Furthermore, rainfalling type algorithms are more amenable to parallelization on a GPU. While rainfalling is conceptually simple (see figure 5.3), plateaus and non-unique "drain" pixels generally make implementations complicated. Fortunately all our computations are performed on floating point images, so these two problems are extremely improbable. Saturated areas of the image are an exception, but they are of no interest anyway. We can therefore apply a very fast watershed transform variant without regard for plateaus. It consists of two steps.

1. Arrowing. Each pixel checks its neighbors and marks the one with the lowest value. If no lower neighbor is found, the pixel is marked as a seed pixel for a new segmentation region. As remarked above, the case with two or more lower neighbors having the same value is highly unlikely in floating point images. Plateau pixels will be marked as seeds at this stage. Each seed gets a unique number, called the region ID. We also store the coordinates of the seed for future reference.

2. Chaining. For each pixel, walk along the arrows pointing to neighbor pixels until a seed pixel is found. The starting pixel is assigned the region ID of the terminating seed in the chain. As an optimization, it is possible to terminate the chain early if a non-seed pixel is hit which has already been assigned an ID. Plateau pixels will result in single-pixel regions which can easily be marked invalid for further processing.

Our serial implementation needs about 15ms to segment an image with a resolution of 780x580 pixels on a 2.2GHz CPU. The algorithm is not perfectly suitable for a massively parallel GPU implementation. However, with some minor modifications it is possible to obtain good performance. The basic premise is that on the GPU there is one thread per pixel, as opposed to the single-threaded CPU version.

1. The generation of a unique region ID is a bottleneck as it requires global atomic operations to synchronize the region counter between the many threads. There-

fore we use a two-tier scheme with frequent updates to a local count (for each work-group in OpenCL terminology) and only infrequent updates to the global count. The rest of the arrowing step does not need changes.

2. The chaining step causes divergence among threads because of different chain lengths. Because of the way GPUs work, this divergence is bad for performance. To avoid it, the chaining step is reversed and implemented as "pumping". One iteration consists of a pixel taking over the ID of its lowest neighbor, which may be invalid. Per iteration, the seeds thus grow one pixel in all possible directions. This is repeated until all pixels have a valid ID. For degenerate images this can take very long, but in the typical case no more than 10 iterations are necessary.

On an NVidia GForce GTX285 GPU the watershed transform typically takes 2ms, excluding transfer time. This corresponds to a speedup factor of 7.5 compared to the CPU version. Figures 5.4 and 5.5 show example inputs and output of the watershed transform. Section 6.4 contains a comparison of the runtimes of the complete decoding algorithm with and without GPU support.

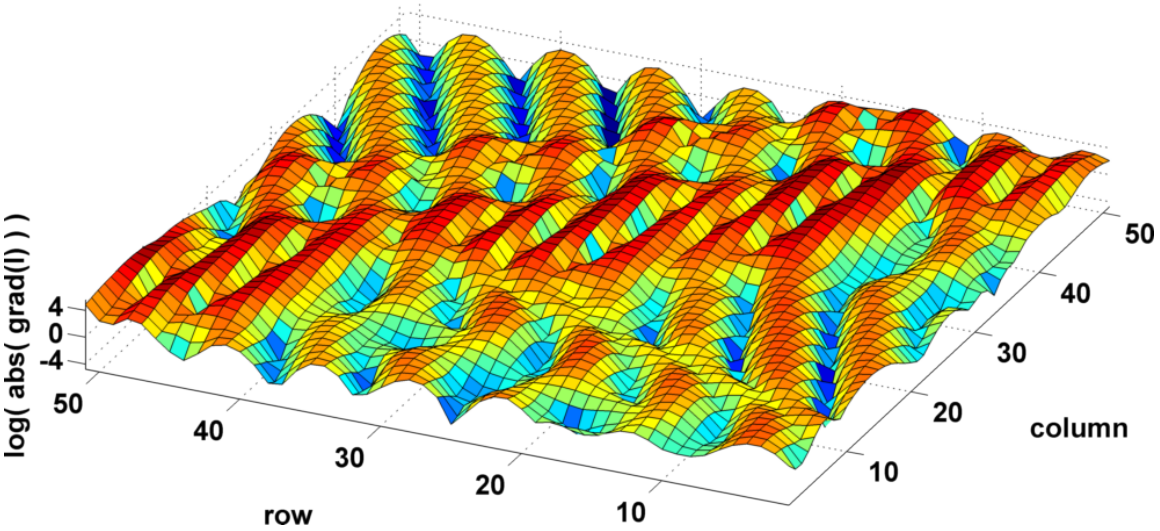
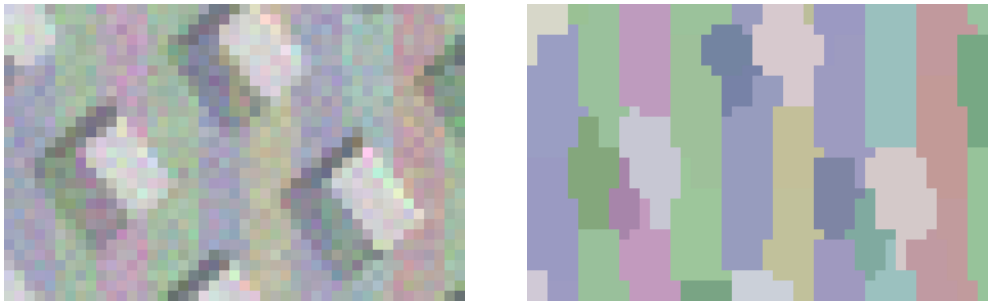


Figure 5.4: Example plot of the gradient magnitude of a stripe pattern. It is the input to the watershed transform.



(a) Input image (b) Resulting superpixels

Figure 5.5: Example of a watershed segmentation performed on a scene illuminated by a stripe pattern.

5.2 Region Adjacency Graph Setup

Once the image is segmented into superpixels, we can perform the core pattern decoding algorithm. It consists of the following series of steps:

1. Build the region adjacency graph. Calculate vertex colors.
2. Calculate the color change over each edge. Assign edge symbols and probability estimates.
3. Find a unique path of edges.
4. Recursively visit all neighbors in a best-first-search, while the edge symbol probability is sufficiently high.

The oversegemented input image can be represented in the form of a region adjacency graph. This graph has one vertex for every superpixel and edges connecting neighboring superpixels. In this context it is necessary to make a distinction between image edges on the one hand and graph edges on the other. The former separate neighboring superpixels in the image, the latter connect neighboring superpixels in the graph. There may be graph edges that do not correspond to discernible image edges. Conversely, an image edge will typically give rise to several graph edges because of the oversegmentation. A vertex of the region adjacency graph corresponds to at least one pixel in the input image. An edge in the region adjacency graph corresponds to pairs of border pixels (see figure 5.5). Unless explicitly noted, in the following text an edge refers to a graph edge.

5.2.1 Vertices

A one megapixel image of a scene illuminated by our color stripe pattern is typically represented by a graph with 50000 vertices. The key property of each vertex is its color K. It is determined by a robust nonlinear rank filter over all the original image pixels that belong to the corresponding superpixel (figure 5.5). Since color is a vector we use marginal ordering [Pita 91]. Another vertex property is the size s of the corresponding superpixel, that is the number of pixels it covers.

The vertex color should additionally be corrected for the color crosstalk that occurs in the camera. We use a technique similar to [Casp 98]. The crosstalk is modeled as a color mixing matrix M_{CM}. Its entries are determined in a precalibration step. The actual color can then be recovered from the observed color by multiplication with M_{CM}^{-1}.

In general, the observed superpixel color is not the original projected color, but rather a color distorted by the object's reflectivity. Therefore, we cannot use the observed color directly, but only color changes between two neighboring superpixels. If the surface color is constant across the two superpixels, its influence will be cancelled out. If it is not, spurious color changes may be detected. However, our decoding algorithm is explicitly designed to handle them.

A second effect is that the surface color influences the relative response of the different color channels. For example the blue channel will appear very weak compared to green and red on a yellow surface. In that case, a 10-digits change in blue may be more significant than a 20-digits change in red. Our patterns are designed to be normalizable, that is, each color channel changes after at least every second stripe. We know, therefore, that each valid superpixel must have at least one neighbor where a given channel changes. Thus, we can define the color range per channel as the maximum absolute change over all neighbors and use it to normalize the color change.

$$d_i = \max_k |c_i^k| \tag{5.1}$$

where c_i denotes the color change in the individual channels and k iterates over all neighbors of a superpixel. The key assumption for this equalization is that each superpixel is as wide as a stripe in the camera image and thus directly borders with both neighboring stripes. This is a reasonable conjecture since in our pattern the stripes are very densely packed in order to produce a high resolution depth map. Empirical evidence shows that the condition is nearly always met.

The next step is to set up the edges and their properties appropriately, so that the specific position of a stripe in the pattern can be recovered via graph traversal.

5.2.2 Edges

The edges of the graph describe how the color changes between two adjacent superpixels. They have several important properties. These are the color change, the edge weight, match probabilities for different color change symbols, the gradient ratio and the direction and length of the edge. We introduce them in this order.

The raw color change $\hat{C}$ is the most basic edge property. The color change between vertex a and vertex b is defined as

$$\hat{C} = K_b - K_a = [\hat{c}_r\, \hat{c}_g\, \hat{c}_b]^T \in \mathbb{R}^3 \tag{5.2}$$

The scalar edge weight w is defined to be its L_∞ norm. This means that edges with only one channel changing can have the same weight as edges with multiple channels changing. A high edge weight indicates a high reliability of the edge. This will be needed later in the traversal step.

$$w = ||\hat{C}||_\infty = max(\hat{c}_r, \hat{c}_g, \hat{c}_b) \tag{5.3}$$

We would like to assign each element of $\hat{C}$ to one of three categories: channel rising, constant or falling. Since we use an alphabet with two intensity levels per channel, there are only three possible labels. We denote triples of labels by symbols, e.g. the symbol for red rising, green falling, blue constant is $S = \begin{bmatrix} +1 & -1 & 0 \end{bmatrix}^T$. The alternative string representation is R+G-. The actual assignment of these symbols involves some preparation.

To equalize the response across the three channels we first multiply each component of $\hat{C}$ by its corresponding inverse range d_i^{-1} from eq. 5.1.

$$C = \hat{C} \otimes D^{-1} \tag{5.4}$$

Here $\otimes$ denotes the elementwise multiplication of two vectors. In fact, since the color dynamic is defined per vertex and the edge connects two vertices, the mean color dynamic of the two vertices is used in eq. 5.4. This normalization procedure makes the decoding algorithm independend of the local stripe contrast. The effect is illustrated in figure 5.6. The normalization can also be used for patterns that are not strictly normalizable, since typically only a few codewords are affected.

We define the symbol match error E associated with assigning symbol S to C as

$$E(C, S) = \sqrt{\sum_i e_t(c_i, s_i)^2} \tag{5.5}$$

with the single channel error e_t defined as

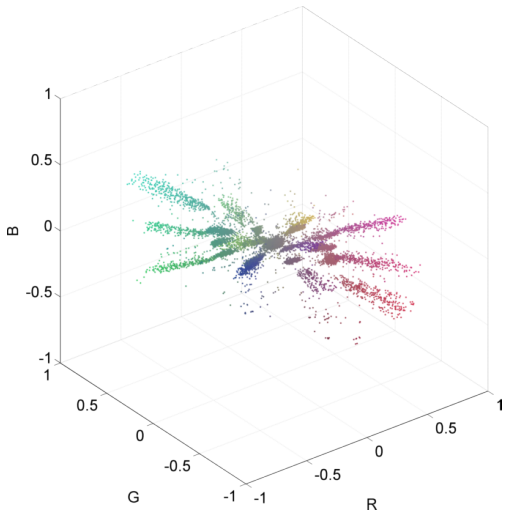
$$e_t(c_i, s_i) = \begin{cases} \frac{1+c_i}{t} & s_i = -1 \\ \frac{|c_i|}{t} & s_i = 0 \\ \frac{1-c_i}{t} & s_i = +1 \end{cases} \tag{5.6}$$

where t is a threshold value below which color changes are considered insignificant. The typical choice is $t = \frac{2}{3}$ for an even partitioning of the interval $[-1; +1]$. Building on the match error we define the match probability

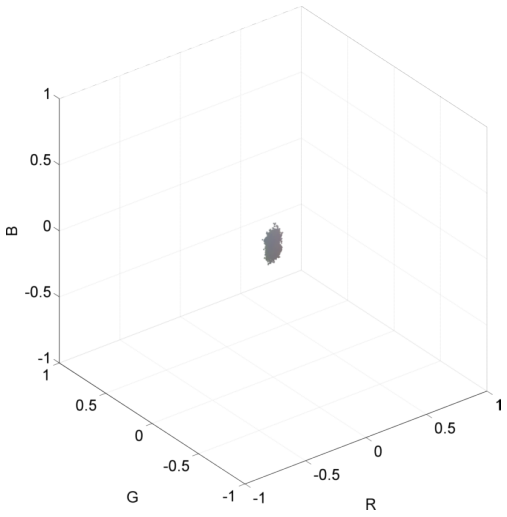
$$P(C, S) = \begin{cases} 1 - E(C, S) & E(C, S) \leq 1 \\ 0 & E(C, S) > 1 \end{cases} \tag{5.7}$$

Note that there can be several symbols that fit almost equally well. Given context information, the algorithm can also assign a suboptimal symbol with lower match probability than the optimal symbol if necessary. We therefore refer to the optimal symbol also as the naive symbol. P can be visualized with the help of spherical isosurfaces in the space of color change vectors. Examples are shown in figure 5.7. The challenge is that the shells for different symbols can partially overlap.

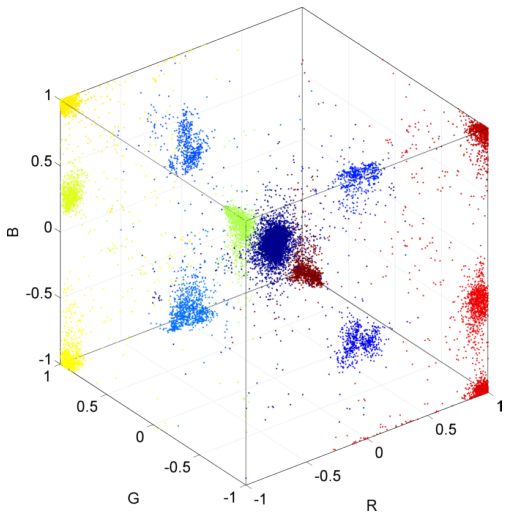
Color transitions that occur at occlusion boundaries are a particular case. Consider for example the case shown in figure 5.8. The blue, yellow, magenta and green stripes appear to be vertically continuous, but are not. Even though the colors of the superpixels above and below the boundary are very similar, we find a high gradient



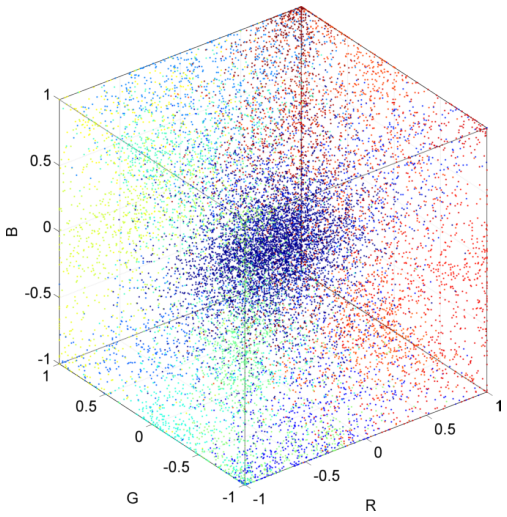
(a) Raw color change vectors of high quality input image.



(b) Raw color change vectors of low quality input image.



(c) Normalized color change vectors of high quality input image.



(d) Normalized color change vectors of low quality input image.

Figure 5.6: The effect of normalization on the color change vectors. The point colors of the normalized vectors depend on the edge symbol that was finally assigned.

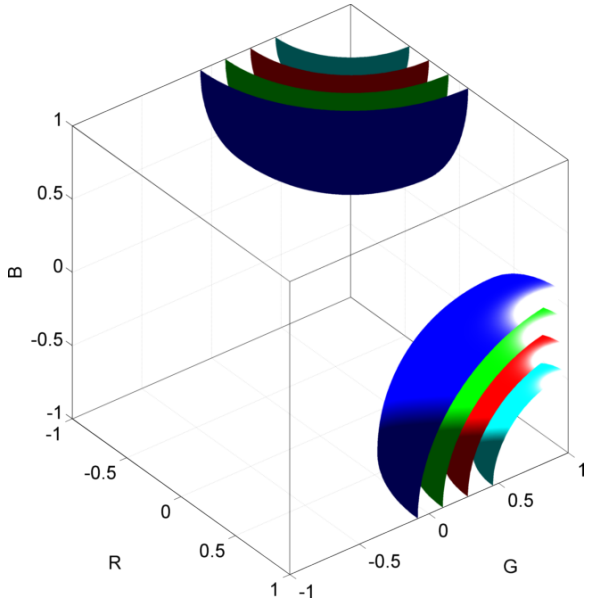


Figure 5.7: Probability isosurfaces for the symbols R+G+B- and R-G+B+. The cyan shells are $P(C,S) = 0.7$, the red shells $P(C,S) = 0.5$, the green shells $P(C,S) = 0.6$ and the blue shells $P(C,S) = 0.4$.

in the direction along the stripes at the border of the two objects. To capture this information, we define the gradient ratio

$$R = \sum_n \sqrt{\frac{\left(\frac{d}{ds}I\right)^2}{\left(\frac{d}{dp}I\right)^2 + \epsilon}} \tag{5.8}$$

where $\frac{d}{ds}I$ and $\frac{d}{dp}I$ denote the components of the image gradient in secondary and primary direction as shown in figure 5.8. The index n iterates over all edge pixels in the image and ϵ is a small regularization constant, typically 10^{-2}. The actual image edge orientation may deviate from the ideal orientation on sloped surfaces. By design, this causes a rising gradient ratio, as oblique image edges are more likely to be artifacts. The gradient ratio will be put to use later in graph traversal step to judge the trustworthiness of position assignments.

Furthermore, each graph edge has to be classified according to its direction in relation to the primary direction of the pattern. The edge direction can be forward, backward or parallel to the pattern. We perform a line fitting over all the pixels associated with the image edge. The position of the superpixel centroids relative to the line allows us to label the directions. The image edges consist of only a few pixels each, so the line approximation works well. The process is illustrated in figure 5.9. Each graph edge also has a length l, defined as the number of neighboring pixel pairs that make up the edge in the image. This property will be used later as a quality criterion along with the edge weight and the gradient ratio.

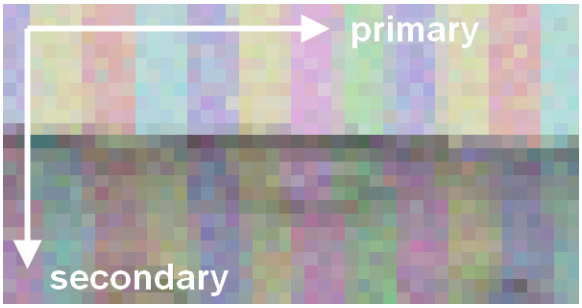


Figure 5.8: Importance of the secondary gradient. The upper and the lower half of the image belong to different objects. Some stripes appear to be continuous in the secondary direction when in fact they are not. The window size of the code is 5.

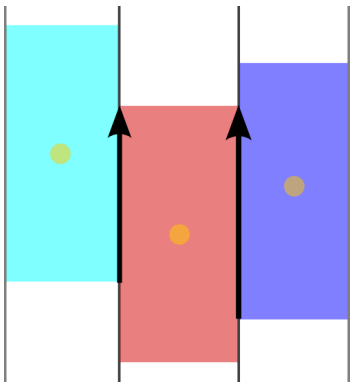


Figure 5.9: Illustration of edge direction assignment. The location of the region centroids relative to the fitted lines give the edge directions. In this case, from the viewpoint of the red region, the edge to the cyan region is "backward" and the edge to the blue region is "forward"..

5.3 Graph Traversal

In our methodology, decoding the pattern is equivalent to finding the correspondence between the vertices of the region adjacency graph and the pattern primitives in the projected image. The window uniqueness property of the pattern makes this association possible. Identifying the position of a subsequence within a suitably designed longer code can be done analytically [Mitc 96]. However, it is easier, faster and more flexible to simply use pre-computed lookup tables which store all the locations where a given primitive occurs. In our case the primitives are colored stripes and the windows used for identification are represented as sequences of edge symbols.

To find the correspondences between graph paths and stripes in the pattern, we first sort all the edges in the adjacency graph by their optimal symbol match probability P (eq. 5.7). The edge with the highest probability is selected as the starting point of a new graph path. The set of its possible positions in the pattern is determined by its optimal symbol S. These positions, in turn, determine the next edge symbols that could be used to extend the path. If one of the two end-vertices of the path has a qualified edge we add it to the path. To qualify for extending the path an edge must: a) have the correct direction and b) its match probability P

for the desired symbol must be higher than a certain user-defined threshold α which controls the 'aggressiveness' of the decoding algorithm. The best value of α depends on the slope t from eq. 5.6, but is typically 0.5. A number of possible positions that would have needed different neighboring edge symbols to continue the path are invalidated by adding a given edge. This process is repeated until only one possible position remains. This happens, at the latest, when the path length equals the unique window length of the pattern. When there is more than one edge that can be added, the one with the lowest error is selected. If there are no valid edges to add, we start again with a new edge.

Once a unique path has been found, the associated pattern positions are uniquely determined as well. We pick an arbitrary seed vertex on the path and propagate the pattern position to its neighbors. The neighbors are visited in a best-first search as follows. If the direction of the edge between the two vertices is correct and the edge error is smaller than α, we add the neighboring vertex to an 'open' heap. The edge symbol used to calculate the edge error may be different from the optimal symbol, as long as the error is below the threshold. Additionally, we maintain a token bucket that can be used to temporarily go below the probability threshold. If the bucket is not full, tokens are collected when an edge with a probability higher than α is encountered. Available tokens are added to the match probability error when it is lower than α. This makes it possible to tolerate isolated bad edges. When all neighbors of a vertex have been visited, we continue the identification process with the best neighbor on the heap, i.e. the one with the highest match probability. When the heap is empty, the propagation stops. If there are unused edges left, we begin a new pass and try to assemble a new unique path starting with the best unused edge.

The quality of a position assignment is captured in the position score Q. It is defined as:

$$Q = \max_k \{\beta P_k - \gamma R_k\} \tag{5.9}$$

with the trust factor β defined as

$$\beta = \sqrt{\frac{w_k}{\overline{w}} \cdot \frac{l_k}{\bar{l}}} \tag{5.10}$$

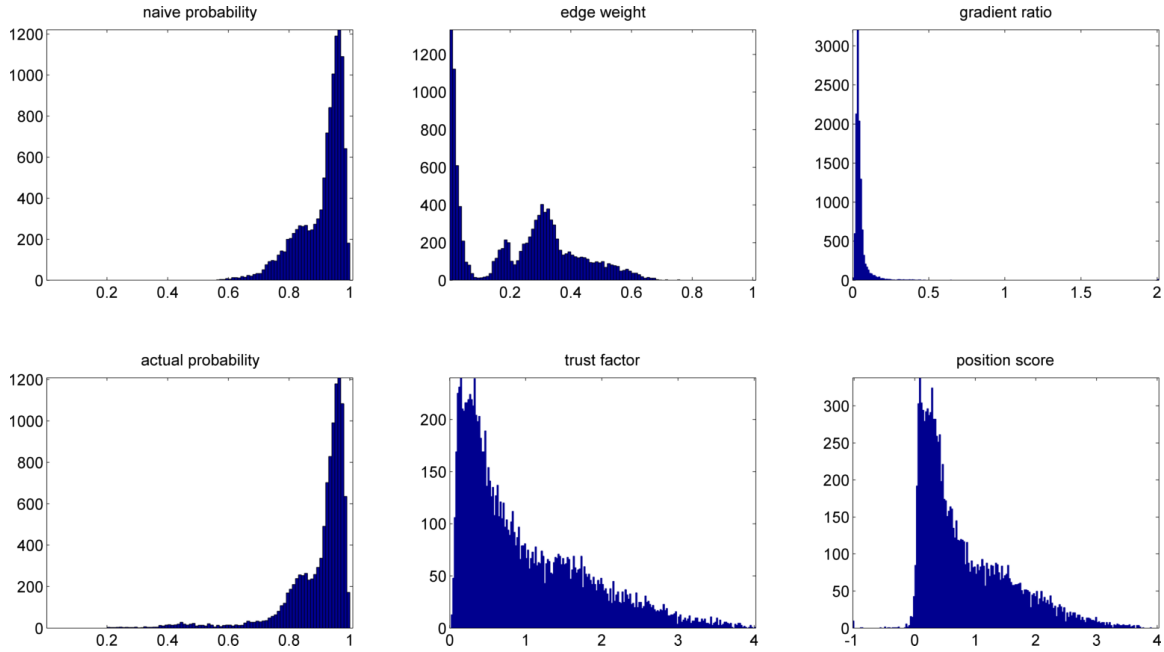
where w is the edge weight, $\overline{w}$ is the average edge weight, l is the edge length, $\bar{l}$ is the average edge length. The parameter γ is a user-defined scale factor (typically 0.5) for the gradient ratio R_k from eq. 5.8. The index k is the neighbor index as before. The inclusion of the edge weight w reflects the fact that high-weight edges are less likely to be disturbed by noise. Longer edges are also more trustworthy than shorter edges. At occlusion boundaries two conflicting pattern positions may occur. In that case, the one with the higher position score Q is chosen. Figure 5.10 illustrates the edge properties and their distributions.

Note that in a stripe pattern there are many so-called *null edges* connecting vertices of equal color. Because of the channel equalization it is well possible that an edge is assigned the symbol 0. However, to prevent undetected cumulative color changes, we do not allow chains of such null edges. Furthermore we perform an additional indirect check before using a prospective null edge. This is illustrated in figure 5.11. We have entered vertex b coming from vertex a. To test if the edge between b and c

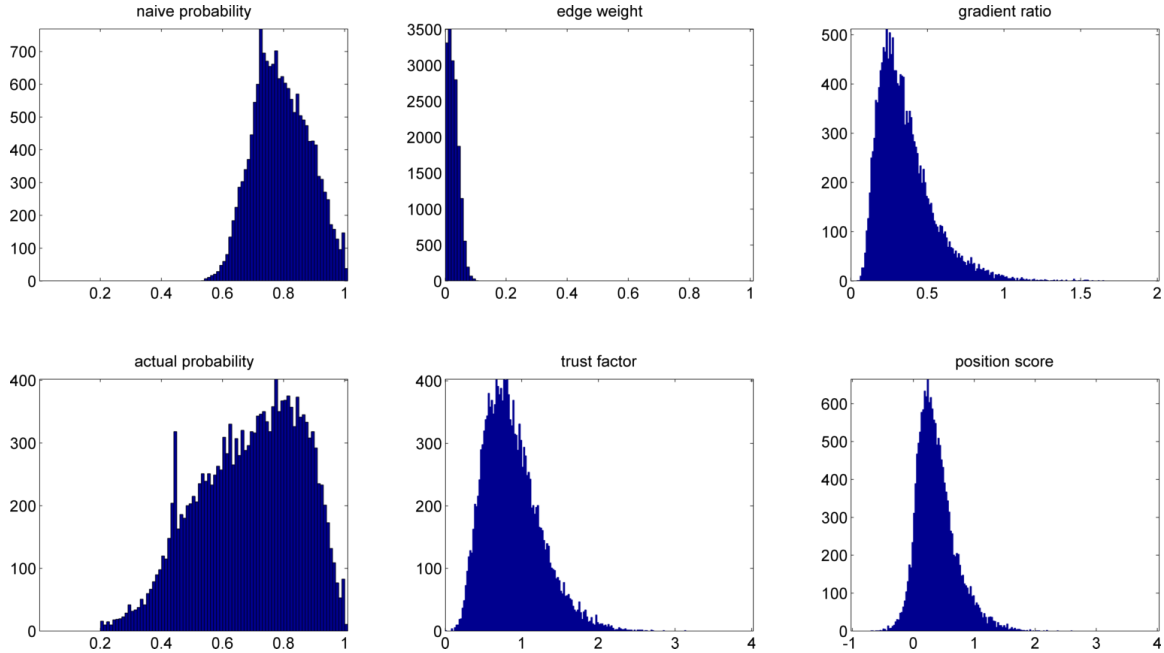
is really a null edge, we calculate the optimal symbol of the (possibly virtual) edge between a and c. If it is identical to the edge symbol assigned between a and b, vertex c has the same color and the same pattern position as b.

An example subgraph with assigned edge symbols is shown in figure 5.11. The bold edges could actually be used, the dashed ones were ignored. There may be shadow areas in the image where there is no pattern to decode. It is statistically possible that a valid edge sequence can still be detected, but it is extremely unlikely that the growth process will lead far. We can, therefore, easily suppress these false positives if an insufficient number of valid neighbors is found.

Our decoding algorithm is local. It starts at a certain edge and recursively visits the neighboring vertices in a best-first-search. We also experimented with an MRF-based graphcut optimization algorithm for finding the globally optimal labeling of the vertices. However the results were less accurate because of complexities of reliably modeling long-range interactions. Furthermore, the runtime of the MRF method was much higher due to the large number of possible labels, which is typically more than 100.



(a) High image quality. The naive and actual symbol probabilites are very similar, indicating that almost exclusively optimal symbols were used. The edge weight histogram shows three peaks, one for a bright object, one for a less bright object and one for null edges. The gradient ratio is very low as the stripe contrast and thus also the primary gradients are very high.



(b) Low image quality. Many suboptimal symbols were used. Null edges and non-null edges have very similar weight. As the stripe constast is low, the gradient ratio is relatively high. This reduces the position scores.

Figure 5.10: Histograms of the various edge properties for two example images of different quality. The histograms include only data from edges that were actually used in the decoding process.

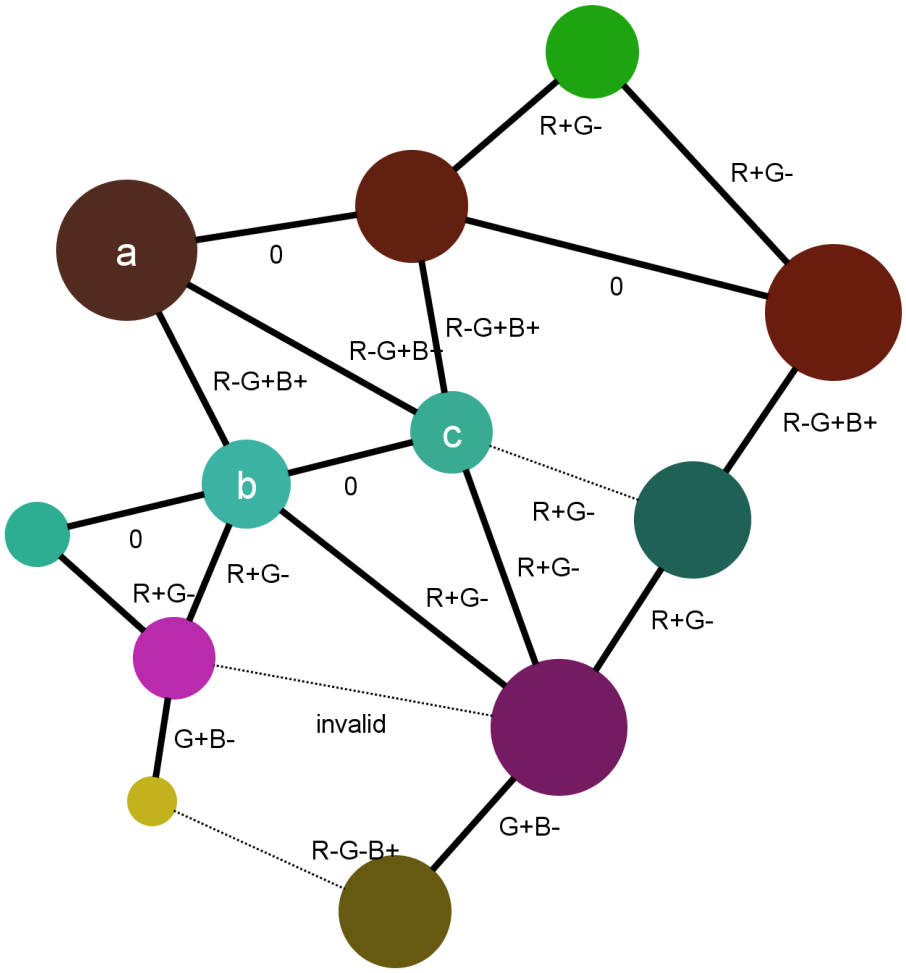


Figure 5.11: An example subgraph of the region adjacency graph. The bold edges could actually be used in the decoding process, the light dashed edges were ignored.

5.4 Color Enhancement with Belief Propagation

We use Belief Propagation [Yedi 03, Felz 04a] to implement a color enhancement step that tries to isolate the projected colors from possible object texture. BP is an iterative message passing algorithm. Each node of the graph receives messages from its neighbors containing their current beliefs about their state. This incoming information is combined with local evidence and passed on. Assuming pairwise cliques, the update equation for the message from node i to node j is

$$m_{ij}^{t+1}(x_j) = \sum_i f_{ij}(x_i, x_j) g_i(x_i) \prod_{k \in N_i \setminus j} m_{ki}^t(x_i) \tag{5.11}$$

Here $f_{ij}(x_i, x_j)$ is the pairwise smoothness term for assigning labels x_i and x_j to nodes i and j, $g_i(x_i)$ is the data term for assigning x_i to node i and $N_i \setminus j$ is the neighborhood of node i excluding node j. The messages m_{ij} are simply vectors of probabilities for all possible labels. They can be initialized randomly or to a uniform distribution. After convergence, the final belief b_i is

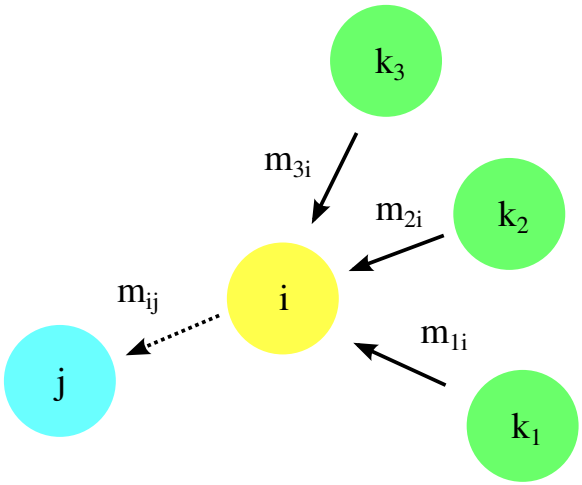


Figure 5.12: Belief Propagation by message passing between nodes in a graph. The message from node i to node j is the product of all incoming messages from nodes other than j, weighted by the local evidence g and the compatibility f.

$$b_i \propto g_i(x_i) \prod_{k \in N_i} m_{ki}^t(x_i) \tag{5.12}$$

It has to be noted that in this notation the messages are multiplied element-wise. The message passing equation is also visualized in figure 5.12. How can it be applied to the region adjacency graph? We propose to re-estimate the colors of the graph vertices from the color changes C across the edges. Of course, if the vertex colors are unreliable, the color changes are also unreliable. However, by using Belief Propagation we can enforce consistency between the different pieces of information. A consensus between the color changes relative to all neighbors is formed and outliers can be corrected.

Furthermore, we observe that the patterns we use consist of no more than eight different projected colors. This means the number of labels is rather low and the inference can be performed in real time. Since the three color channels are independent, we can even split the problem and perform per-channel inference with binary labels: At a given vertex, a given color channel can only be either on or off. The smoothness term $f_{ij}(x_i, x_j)$ can therefore be written as a 2×2 compatibility matrix $\mathbf{F}$.

$$\mathbf{F} = \begin{bmatrix} p_{constant} & p_{rising} \\ p_{falling} & p_{constant} \end{bmatrix} \tag{5.13}$$

On the diagonal we have the probability of the channel state being constant, as both nodes have the same label. If node i is in "off" state (index 0) and j in "on" state (index 1) the channel must rise, or fall in the opposite case. In our case there is no data cost, that is $g_i(x_i) = 1$, as we do not judge the absolute color values but only the color changes. We initialize the BP messages as $m_{ij}^0(x_j) = 1$, i.e. we make no assumption whether a given channel is on or off in the beginning.

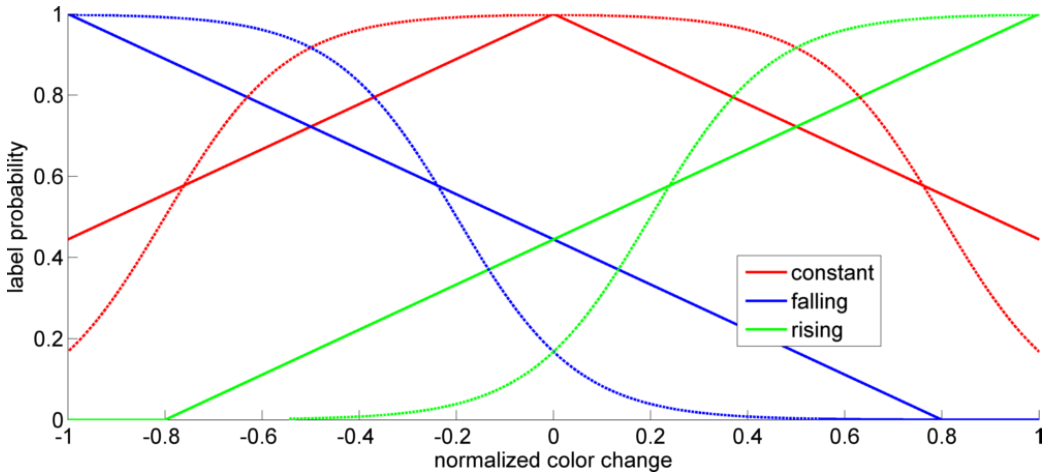


Figure 5.13: Linear (solid lines) and sigmoid (dotted lines) probability functions used for Belief Propagation message update. The parameters used were $h_l = 1.8$, $h_s = 0.8$ and $s = 8$.

The actual values of f_{ij} can be computed with different models. One possibility is a truncated linear model:

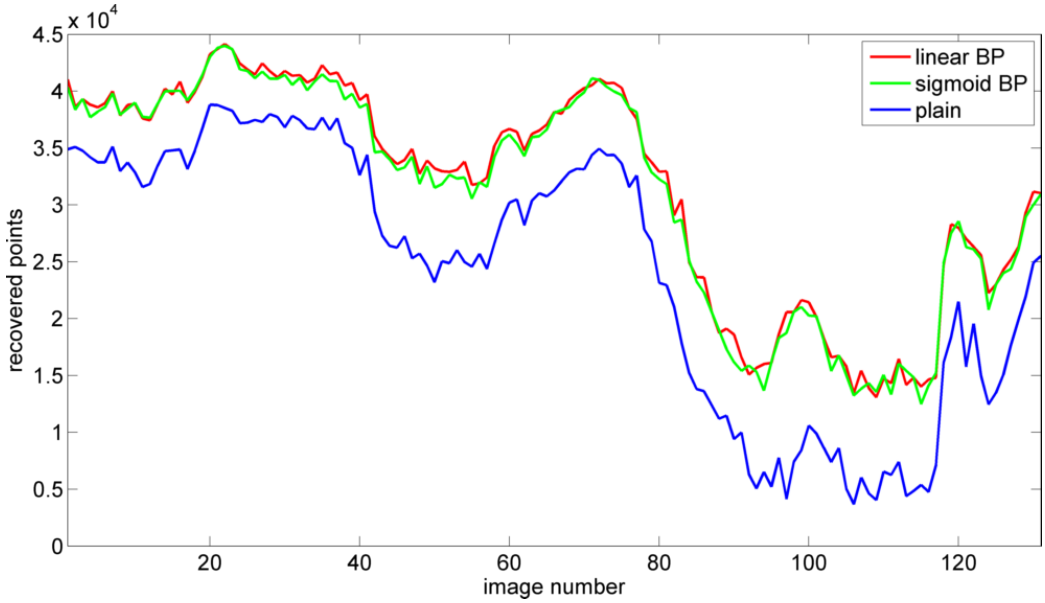
$$p^{linear} = \begin{cases} max(0, 1 - \frac{1-c_i}{h_l}) & \text{falling} \\ max(0, 1 - \frac{|c_i|}{h_l}) & \text{constant} \\ max(0, 1 - \frac{1+c_i}{h_l}) & \text{rising} \end{cases} \tag{5.14}$$

where h_l is the inverse slope of the probability function, typically set to values between 1.5 and 2. An alternative is a sigmoid-type model

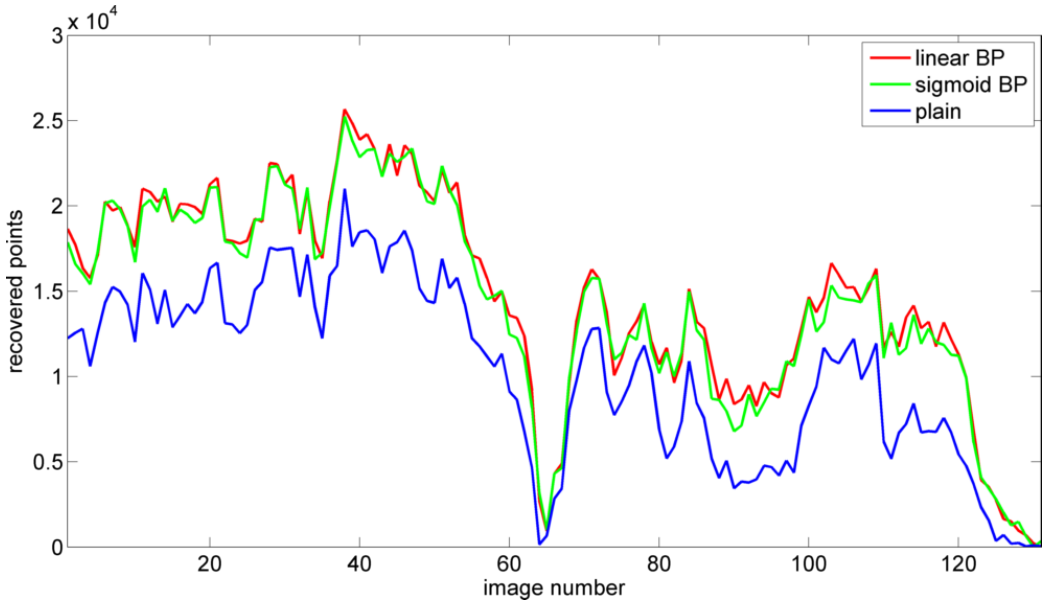
$$p^{sigmoid} = \begin{cases} 1 - (1 + exp\left((h_s - c_i - 1)\, s\right)^{-1} & \text{falling} \\ (1 + exp\left((|c_i| - h_s)\, s\right)^{-1} & \text{constant} \\ 1 - (1 + exp\left((h_s + c_i - 1)\, s\right)^{-1} & \text{rising} \end{cases} \tag{5.15}$$

where h_s is the "width" of the peak and s is the steepness of the peak. Typical values for h_s and s are around 0.8 and 8 respectively. The c_i are the indiviual normalized channels of the color changes from eq. 5.4 and so the probabilities are in the range of $[1; 0]$ as they should. Both models are plotted in figure 5.13.

In practice the linear and the sigmoid model deliver similar performance. This is illustrated in figure 5.14 for two different test sequences. The increase in the number of recovered points is between 30% and 40%. However, there are some caveats. The first is that there is no ground truth for these sequences. More data is not automatically better data. In fact, the additional data points might be false positives. Additionally, there is a ceiling effect - if the parameters are set too loose, even plain decoding without color enhancement can identify almost all stripes that can reasonably be expected. We therefore set the algorithm parameters to the conservative default values given in table A.2 to make sure only valid depth data is generated. In particular the minimum symbol probability was 0.6. If it is increased to 0.7, the improvements that can be achieved with color enhancement are even more pronounced: 146% for sequence 3 and 76% for sequence 6 (see also figure 5.15).

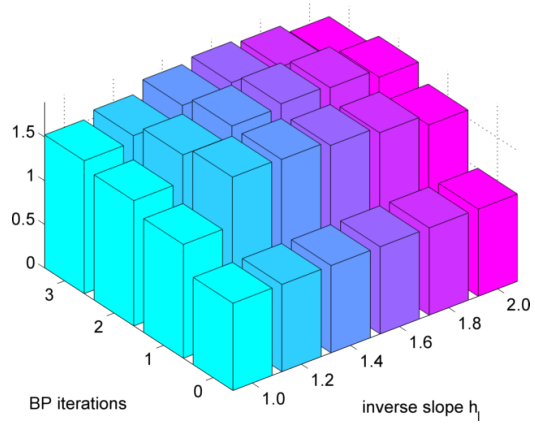


(a) Test sequence 6 with a low amount of bubbles degrading the image quality. The improvement for linear BP is 30%, for sigmoid BP it is 28%.

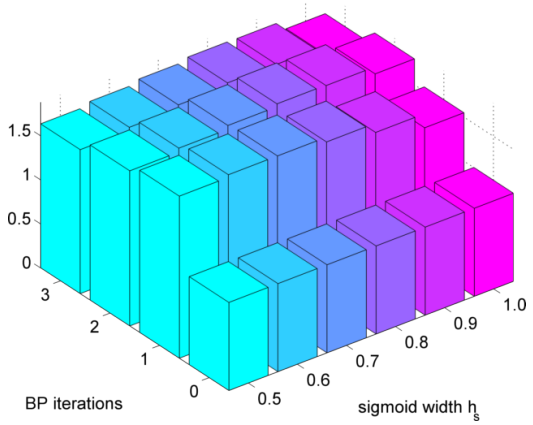


(b) Test sequence 3 with a large amount of bubbles degrading the image quality. The improvement for linear BP is 43%, for sigmoid BP it is 40%.

Figure 5.14: Decoding performance of color enhancement with the linear model and the sigmoid model after two iterations. Plain decoding without color enhancement is used as a reference. Both models markedly increase the number of points recovered. The minimum symbol probability during decoding was 0.6 in all cases.



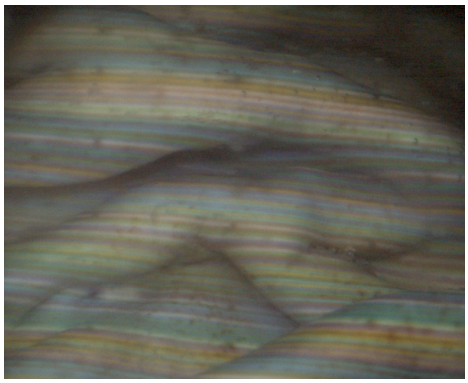
(a) Sequence 6, linear model. The maximum is at $h_l = 1.6$ with two iterations.



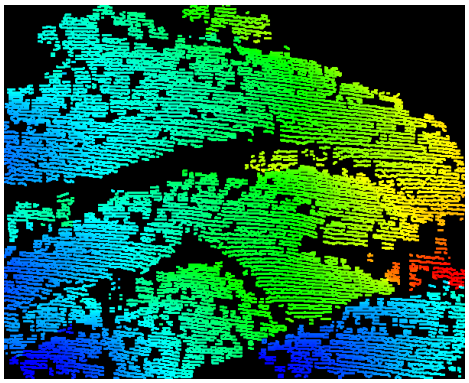
(b) Sequence 6, sigmoid model. The maximum is at $h_s = 0.6$ with one iteration. The steepness parameter s was fixed at a value of 4.

Figure 5.15: Decoding performance for different number of iterations and different model parameters. The z-values are the average number of decoded points over all image of sequence 6. They are normalized to the number of data points generated without color enhancement, that is with zero iterations. A minimum symbol probability of 0.7 was used.

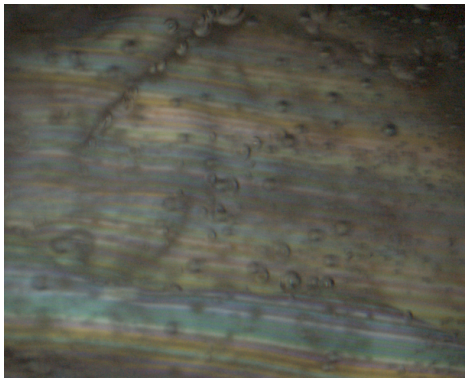
The best performance is achieved after one color enhancement iteration. For softer parameter choices in the probability model (higher h_l or h_s) a second iteration can improve the result. The color enhancement step is especially beneficial in low-contrast situations, where single edges may be unreliable. In that case integrating the information from all neighbors before making a decision is especially helpful. This is a crucial improvement for medical purposes, to counter the sub-surface scattering in skin. Figure 5.16 shows the input image and recovered range data for some example images. Figure 5.17 shows the evolution of the superpixel colors for another example image.



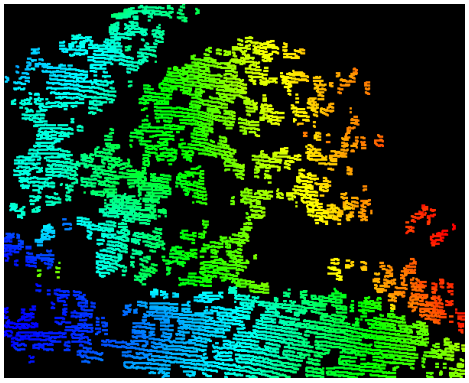
(a) Example image 64 from sequence 2



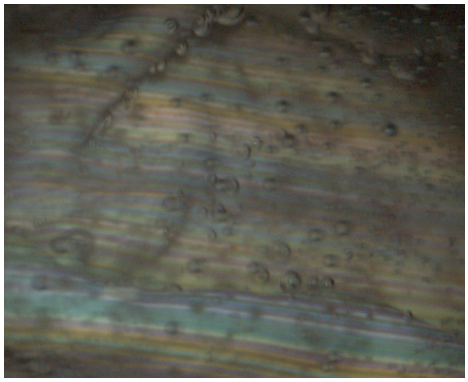
(b) Color-coded depth values recovered from image 2_64. Range is 140mm to 160mm.



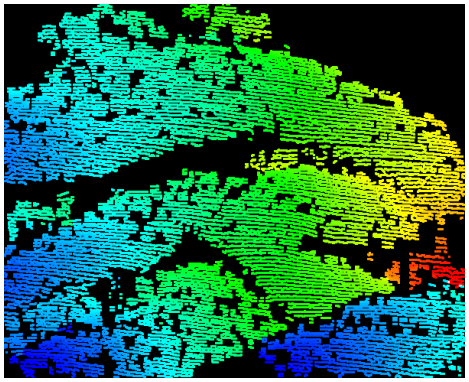
(c) Example image 46 from sequence 3



(d) Color-coded depth values recovered from image 3_46. Range is 140mm to 165mm.



(e) Example image 70 from sequence 6



(f) Color-coded depth values recovered from image 6_70. Range is 142mm to 158mm.

Figure 5.16: Example images from the different sequences with recovered depth values. For better visibility, the input images were gamma-adjusted and results dilation filtered.

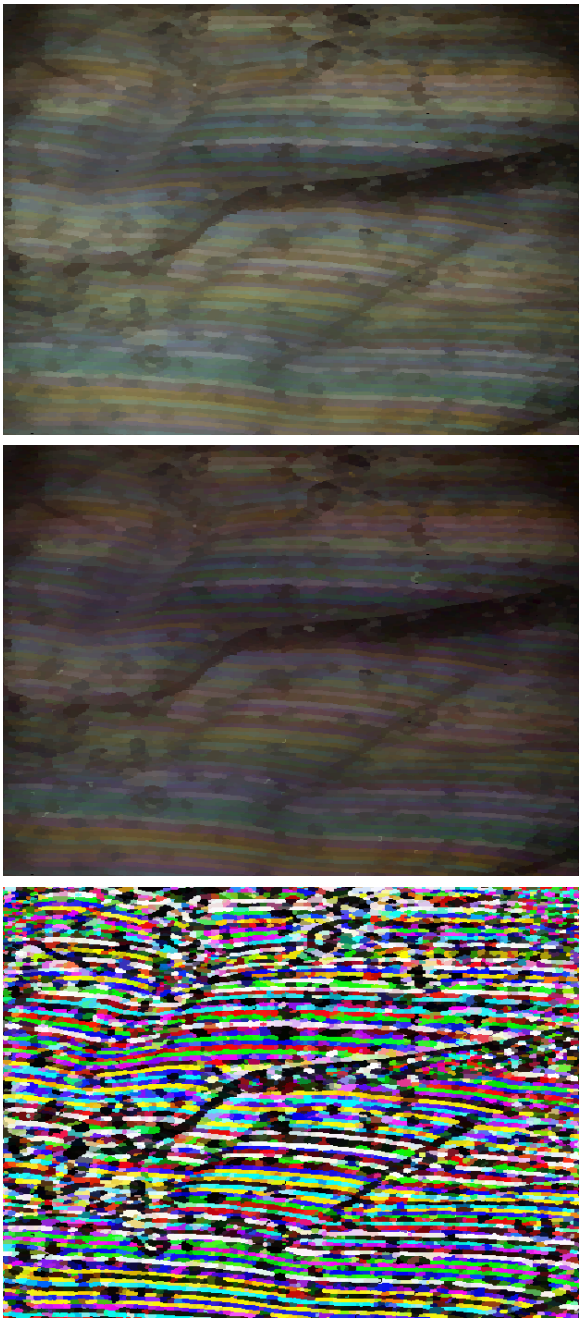


Figure 5.17: Influence of the color enhancement step. On the top an example input image with the raw superpixel colors (gamma adjusted for better visibility). In the center, the colors have been corrected for crosstalk (see section 5.2.1). On the bottom, the superpixel colors after two iterations of Belief Propagation.

5.5 Edge Localization and Tracing

Once the region adjacency graph has been traversed and all possible vertices have been identified we have to localize the corresponding image edges with subpixel accuracy. We iterate over the segmented input image. At the border of two superpixels we check if the corresponding graph vertices could be identified and have successive pattern positions. If that is the case, we look for the nearest local extremum of the image gradient and interpolate its subpixel position. There are three possible ways in which the gradients can be used.

- Use the magnitude of the gradient in all three channels. We know that the gradient magnitude must have a local maximum where the two superpixels meet, otherwise the watershed transformation would not have resulted in two distinct superpixels at this point. The gradient magnitude may contain influences of color channels which are irrelevant at the particular location.

- Use all color channels separately and compute the average position, possibly weighted by the absolute value of the extremum. The edge symbol between the two vertices specifies which individual color channels are supposed to change. However, under noisy conditions it is not granted that the corresponding extrema can be detected reliably and accurately. Single dropouts can be tolerated and the edge position can be determined from the remaining channels. The different edge positions in the different channels also do not always agree perfectly. One component of the mismatch is of course due to image noise, but there are also systematic components like the manufacturing tolerances of the projection slide and chromatic aberrations in projector and camera. This necessitates elaborate edge position calibration if the highest accuracy is to be reached.

- Use only the green channel gradient. The green channel has the highest resolution in the camera Bayer pattern and therefore promises to yield the most accurate edge position. To put it differently, the red and blue channels are more prone to aliasing because of their lower sampling frequency. More importantly, there are no edge position shifts through chromatic aberration in the projector and camera lenses. The pattern has to be designed in a way that the green channel changes at every edge. If the system is calibrated with a feature-based method, the features should also be localized in the green channel only so that everything fits together.

The method of choice for the subpixel localization is parabolic interpolation. It is fast and needs only a small support of three pixels. For comparison, the Blais-Rioux detector needs six samples. Interpolation of a Gauss function delivers a minimal increase in accuracy but is much slower (see section 2.3). The principle is illustrated in figure 5.18. A parabola is fitted to a local extremum and its left and right neighbors. The interpolated position of the extremum is then

$$p_{subpix} = p_c + \frac{v_l - v_r}{(v_l + v_r - 2v_c)} \tag{5.16}$$

where p_c is the coordinate of the center pixel and v_l, v_c and v_r are the values of the left, center and right pixel.

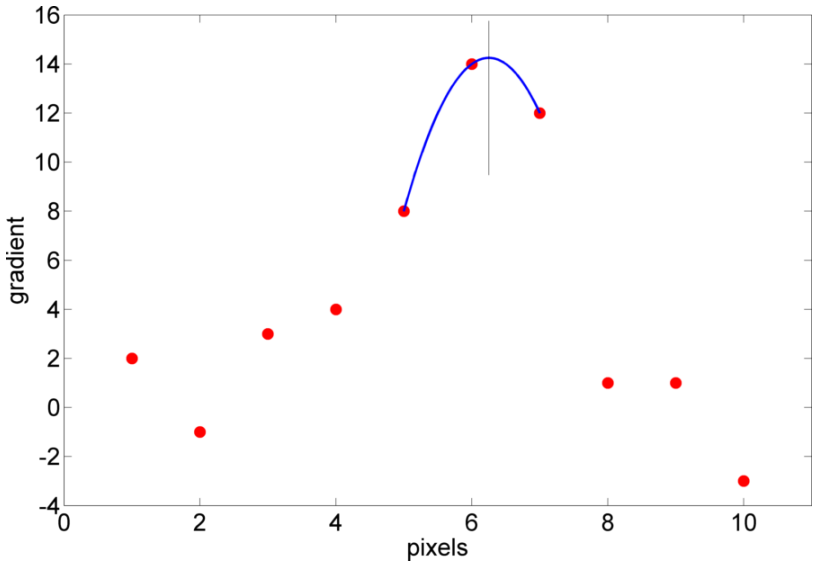
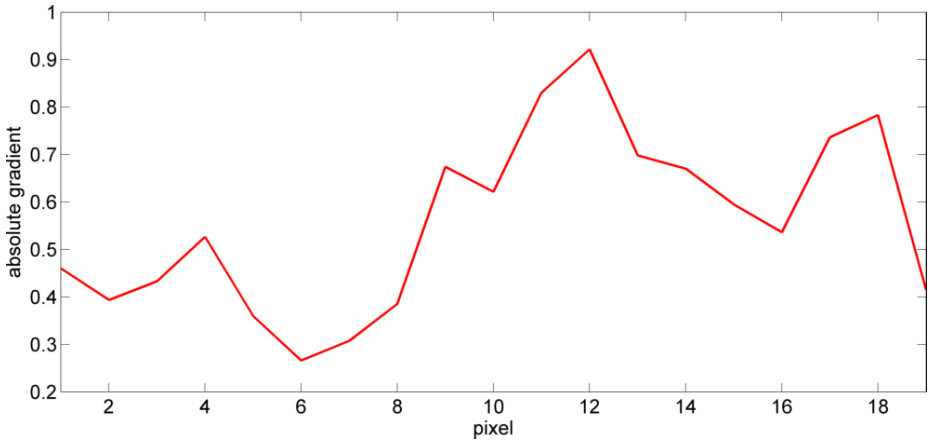
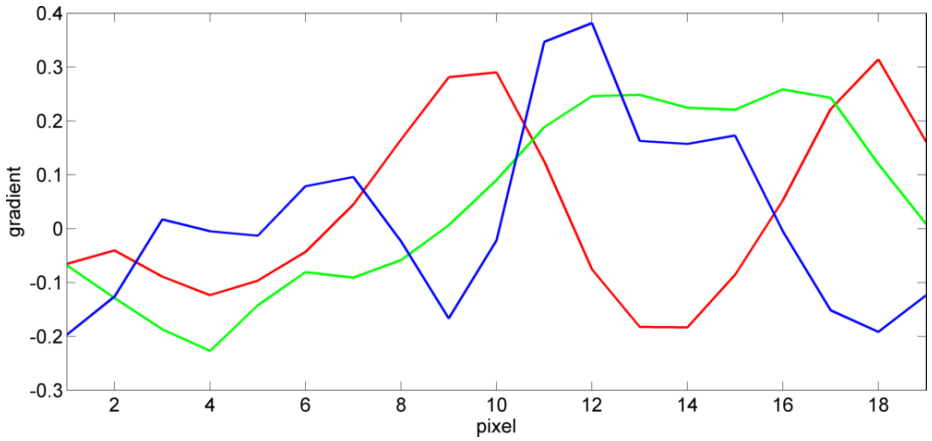


Figure 5.18: Subpixel edge localization by parabola fitting.



(a) Example profile of the gradient magnitude for all three color channels.



(b) Example profile of the single-channel gradients. There are many peaks at different positions. Some belong together, some are false positives. It is difficult to establish a definitive edge position based this data.

Figure 5.19: Edge localization example in a low-quality image.

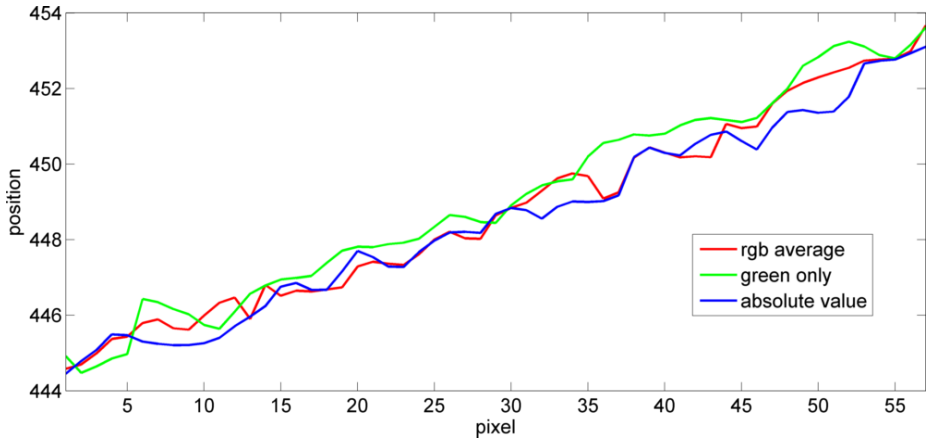


Figure 5.20: Example results for different edge localization methods. The differences are up to one pixel.

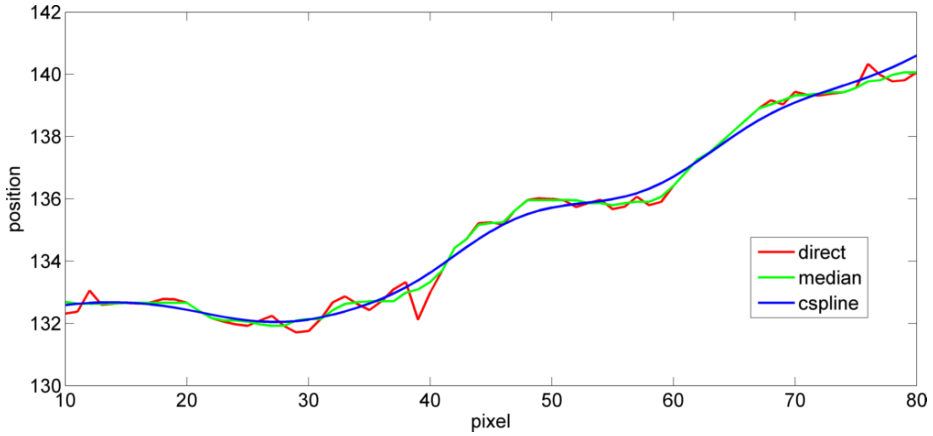


Figure 5.21: Edge filtering. Median of seven eliminates spikes but retains kinks. Cubic spline approximation results in a smooth edge.

We implemented all three methods. If the pattern supports it, we use the green channel only method as default to sidestep possible errors caused by chromatic aberration. However, there can still be minor edge position shifts caused by crosstalk from the red and blue channels. This can be neglected in comparison to the influence of image noise on the detected edge positions. There are several filters that can be applied to reduce the noise. Of course, the image can be pre-smoothed before computing the gradient. However, weak edges may be erased if the smoothing is too extreme. For post-processing we chose a median-of-seven filter and a cubic spline approximation [Wein 09]. The results are shown in figure 5.21. The filters are applied to each edge fragment separately, so edges will not be smoothed across discontinuities. In a simulated noisy example image, the standard deviation from the plane was 1.127mm. With median filtering this reduced to 0.944mm. The cubic spline approximation resulted in an error of 0.820mm.

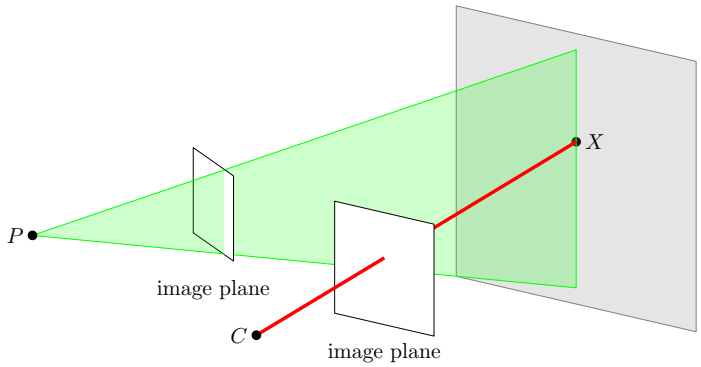


Figure 5.22: Ray-plane intersection for depth computation

5.6 Depth Computation

The final step after decoding of the region adjacency graph and subpixel localization of the stripe edges is the computation of depth data. It differs between the "standard" 3D scanning setups using standard linear color stripe patterns and the endoscopic sensor using a color ring pattern. In the former case, the actual depth values are computed by ray-plane intersections. Note that ray-ray intersection cannot be used, because in a stripe projection system only one coordinate in the projected image is known, the other is initially unconstrained. However, this is not necessarily a disadvantage. In a stereo ray-ray intersection both rays must be assumed to be slighly erroneous. In contrast, the light plane defined by the edge between two projected stripes is theoretically known exactly.

In practice manufacturing tolerances and optical effects like chromatic aberration, finite depth-of-field and image distortion can make it difficult to define an exact light plane. With a DMD-based projector, the manufacturing tolerances are practically zero. With our slide-based pattern projection, the stripes are created by multilayer interference filters whose width is often not exactly as specified. By measuring a flat surface these systematic offsets can be detected and corrected with a simple lookup table. If chromatic aberrations are present, the best way to manage them is to use only one color channel for calibration and for edge localization. If there is substantial chromatic aberration and multiple color channels have to be used, projector and camera can be calibrated for each color channel separately.

Real projectors with image distortion do not project perfect planes of light but ruled surfaces. It is possible to model these more general geometric shapes explicitly as has been done in the calibration of the endoscopic sensor. For small deviations from the planar shape the problem can also be solved iteratively, as for example proposed by [Fors 05]. We first ignore the projector image distortion and calculate a preliminary depth value. The missing second projector image coordinate can then be reconstructed approximately by reprojection of the depth value to the pattern image plane. Using the two projector image coordinates the usual distortion model can be applied and a new corrected light plane can be calculated. We use this new light plane to compute a better depth value and repeat until convergence. In practice one iteration is enough for projection systems with limited distortion.

In the endoscopic sensor the depth data is computed by ray-cone intersection. The cones are co-axis with the optical axis of the projector, so radial distortion only changes the apparent opening angles of the cones. Each projected light cone has been calibrated explicitly, so projector distortion can be ignored. However, one has to keep in mind that the triangulation angle varies considerably over the working space. In combination with the low image contrast, which hampers precise stripe edge localization, a low triangulation angle can lead to large depth errors. The accuracy of the depth data will be evaluated in the next chapter.

Chapter 6

Evaluation

In this chapter the performance of the proposed 3D scanning system is evaluated with respect to accuracy and robustness. For the accuracy tests, synthetic and real images are used. Comparing our 3D sensor to other methods is not trivial. The measurement uncertainty must be normalized to the field of view, lateral and longitudinal resolution have to be taken into account, and even the time and bandwidth required to record the data may be a factor in determining the 'efficiency' of a sensor. Of cource, each sensor type has its physical limitations [Haus 11] which cannot be surpassed. To test the performance of the pattern decoding algorithm, we compare our results to those of other approaches. For the endoscopic scanning system comparisons to other sensors are even more difficult, so we present reconstruction results for four different test objects. In the last section some notes on the implementation and the runtime of the algorithm are presented.

6.1 Accuracy

The accuracy of a measurement system can be defined as the closeness of agreement between a measured quantity and the true quantity. The depth accuracy of our 3D scanning approach is first tested on various synthetic scenes. For these scenes ground truth is available and the calibration of the simulated scanner is perfectly known. The test scenes feature geometry and textures of various complexity and come in different controlled noise levels. This test isolates errors caused by imprecise edge localization and false correspondences. The calibration accuracy of a real system was evaluated for a desktop scanner with a calibrated reference object as well as for an endoscopic scanner in a known artificial cavity.

6.1.1 Simulated images

To test the accuracy of the depth calculation, we rendered test images with Povray [Pers 04]. A simulated planar test object was illuminated by a color stripe pattern and ambient light of various intensities. Additionally white gaussian noise of different standard deviations was added to the images. Details of the images can be seen in figure 6.2. The resulting standard deviations of the depth values from the ground truth are shown in figure 6.1. The simulated Structured Light setup has a baseline

	mean [mm]	sigma [mm]
Sphere Fit Residuals	0.0283	0.0123
Sphere Diameter Errors	0.0342	0.1043
Sphere Distance Errors	0.0416	0.0744

Table 6.1: Barbell measurements summary for the sensor calibrated with the Active Target method.

of 80mm and a working distance of 600mm. The simulated camera has a resolution of 780x580 pixels with a pixel size of 8.5µm and a focal length of 8.5mm. In this setup, an edge localization error of 1 pixel causes a depth error of approximately 5mm, depending on the exact position in the working space. The depth errors in the high noise cases therefore correspond to edge position errors of approximately $\frac{1}{2}$ pixel.

We also simulated a non-planar target in the form of the Stanford Dragon with a moderate contrast of 0.5 and moderate noise with $\sigma = 0.067$. In this case the system had a baseline of 250mm. One pixel shift in the edge location therefore causes an error of approximately 3.5mm at a working distance of 1000mm. Figure 6.3 illustrates the resulting depth errors.

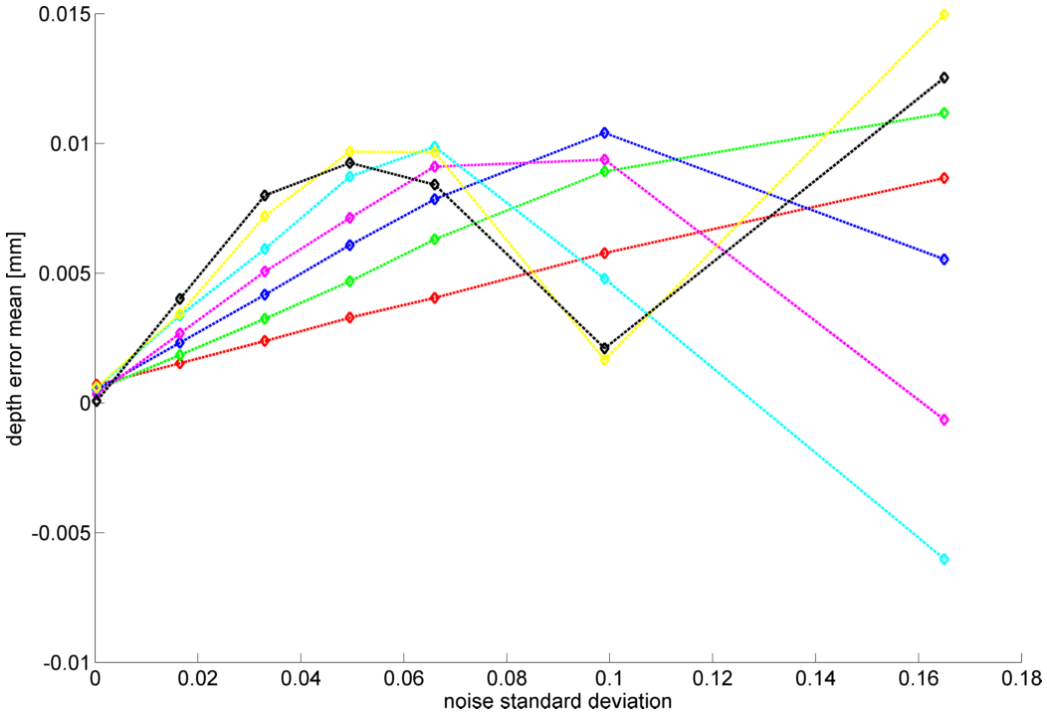
The influence of an edge localization error in the camera image depends on the system geometry and on the exact position of the true point in the working space. For a given scene, it can be visualized in a sensitivity map. Figure 6.4 shows the depth errors caused by a localization error of 1 pixel for the Stanford Dragon.

6.1.2 Desktop reference object

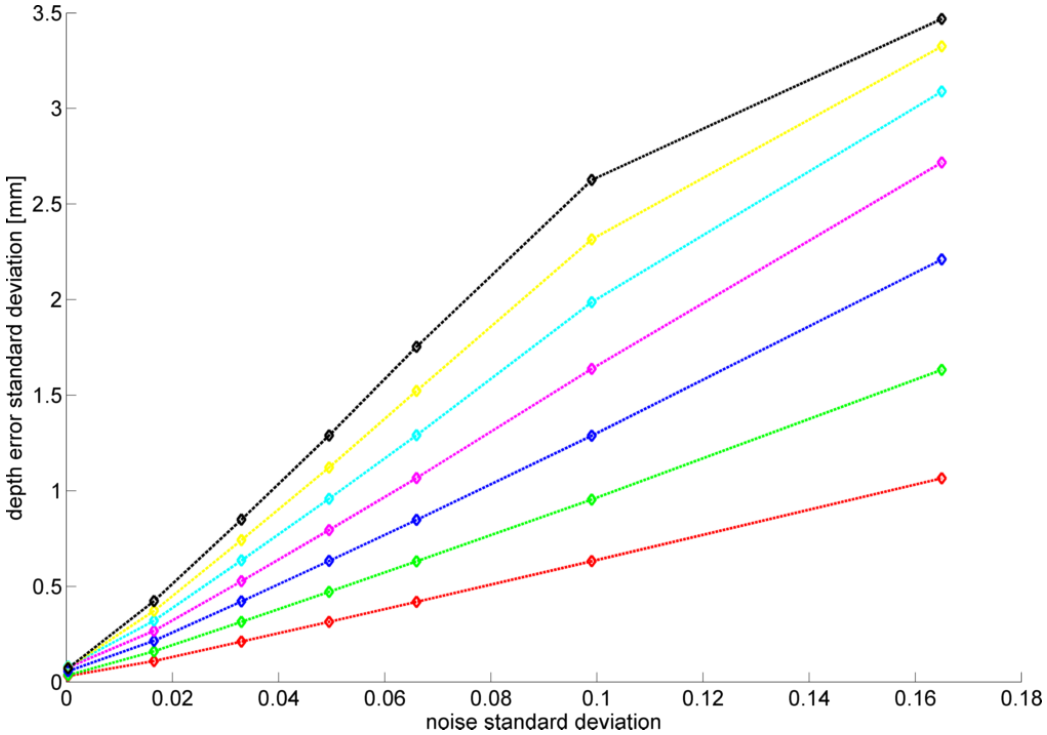
A calibrated reference body in the form of two rigidly connected, diffusely reflecting spheres was used for evaluation of the absolute measurement errors (figure 6.5). These errors are a combination of edge localization error and calibration error. The results are summed up in table 6.1. Most errors are below 100µm, which is a good result.

This accuracy test was also used to verify the performance of the active calibration method proposed in 4.3.1. We calibrated a desktop 3D scanning system with a working volume of approximately $100 \times 100 \times 100 mm^3$ with both the 'classic' feature-based method and with an active target. However, it has to be kept in mind that the active calibration still needs a classic target for the projector calibration. The resulting system parameters are given in tables 6.2 and 6.3. There are marked differences in the internal as well as in the external parameters.

The barbell was positioned in six different poses. A depthmap was acquired in each pose and two spheres were fitted to the measured data. The ground truth diameters of the spheres are 29.827mm and their center distance is 80.006mm. Both values are known with a tolerance of 2µm. For both calibration methods the sphere fit residuals, the computed sphere diameters and the computed sphere distances were used in the evaluation. The results are given in tables 6.4, 6.5 and 6.6 as well as in graphical form in figures 6.6, 6.7 and 6.8. The poses used for both calibrations were unfortunately not perfectly identical as the sensor had to be moved in between for the recalibration. However, in both cases the whole measurement volume was covered.

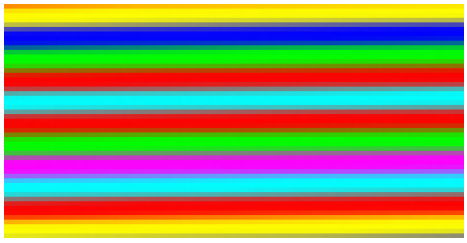


(a) Mean depth error. The working distance is 600mm, so the relative error is around 10^{-5}.

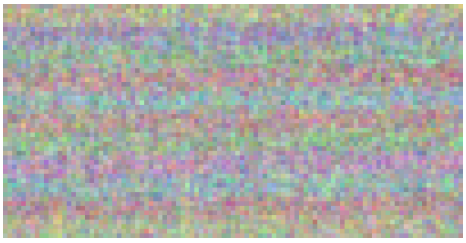


(b) Depth error standard deviation. For the better image qualities the relative error is around 10^{-3}.

Figure 6.1: Accuracy on a synthetic planar test scene. Mean and standard deviation of the depth error for different noise levels and different pattern contrasts. The contrast ranges from 1 (red) to 0.25 (black). In the simulated system geometry, an edge localization error of 1 pixel gives rise to a depth error of roughly 5mm.



(a) Maximum contrast of 1, minimum noise with $\sigma = 0$.



(b) Minimum contrast of 0.25, maximum noise with $\sigma = 0.165$.

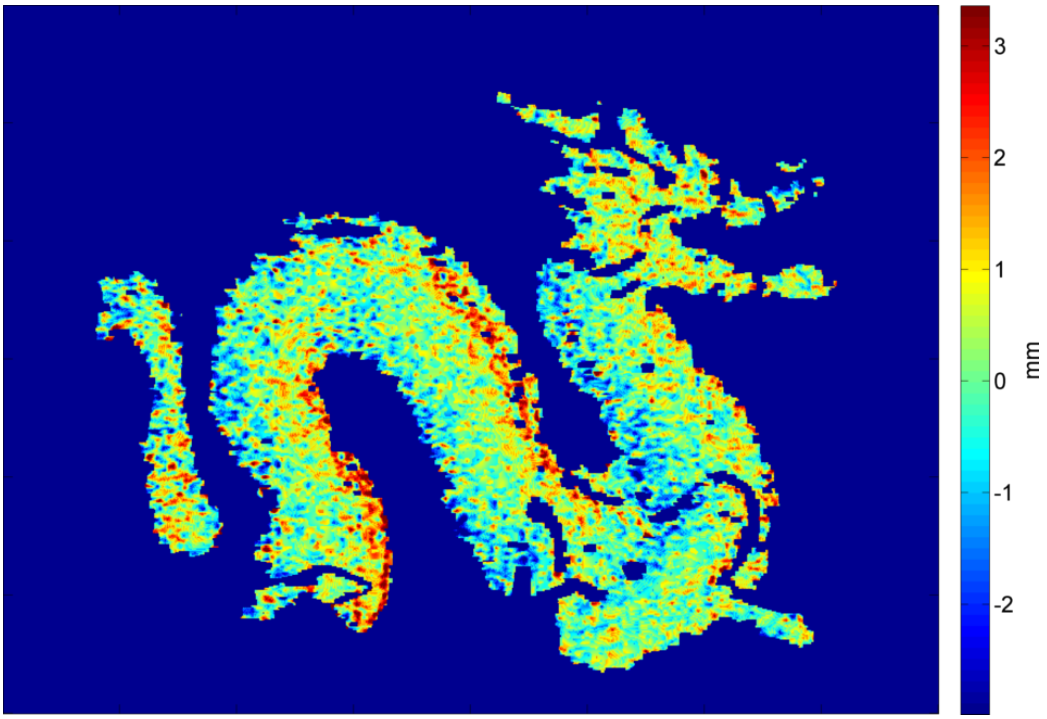
Figure 6.2: Example details of two extreme input images used in figure 6.1. The projector is rotated slightly to reduce aliasing.

	camera		projector	
	classic	active	classic	active
$f\ [mm]$	13.039	13.045	15.618	15.608
$u_0\ [pix]$	409.317	410.007	379.175	382.424
$v_0\ [pix]$	296.066	296.145	597.424	598.293
$k_2\ [\frac{10^{-3}}{mm^2}]$	-120.969	-123.932	-6.743	-18.055
$k_4\ [\frac{10^{-3}}{mm^4}]$	-39.286	180.245	-68.860	25.378
$k_6\ [\frac{10^{-3}}{mm^6}]$	1705.174	-253.735	63.154	-160.817
$p_1\ [\frac{10^{-3}}{mm^2}]$	0.288	0.595	-0.329	0.574
$p_2\ [\frac{10^{-3}}{mm^2}]$	-0.259	-0.580	-3.410	-2.276
$RMSE\ [pix]$	**0.16502**	**0.09294**	**0.21702**	**0.08541**

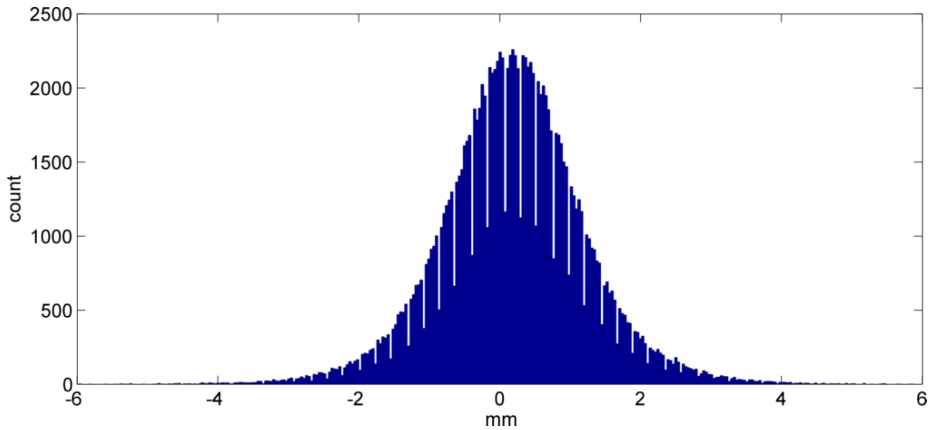
Table 6.2: Comparison of system calibration with classic method and active method. The differences in the parameters as well as in the reprojection error are considerable.

	classic	active
$t_x\ [mm]$	6.88730	6.69478
$t_y\ [mm]$	-81.88410	-81.93482
$t_z\ [mm]$	18.45150	17.80056
$R_x\ [']$	20.053	20.837
$R_y\ [']$	14.145	1.930
$R_z\ [']$	-38.826	-42.811

Table 6.3: Projector pose relative to camera for classic and active calibration. The differences are most apparent for the z-component of the translation and the y-component of the rotation.



(a) Deviation from reference



(b) Distribution of depth errors. Mean is 0.19mm, sigma is 1.06mm. The mean is non-zero because of the asymmetric perspective on the object. The projector was located to the right of the camera, and the positive errors on the sloping sides of the dragon predominate slightly (compare figure 6.3a and 6.4).

Figure 6.3: Depth error distribution on a non-planar target. As the noise added to the image has a gaussian distribution, the depth error distribution is similar.

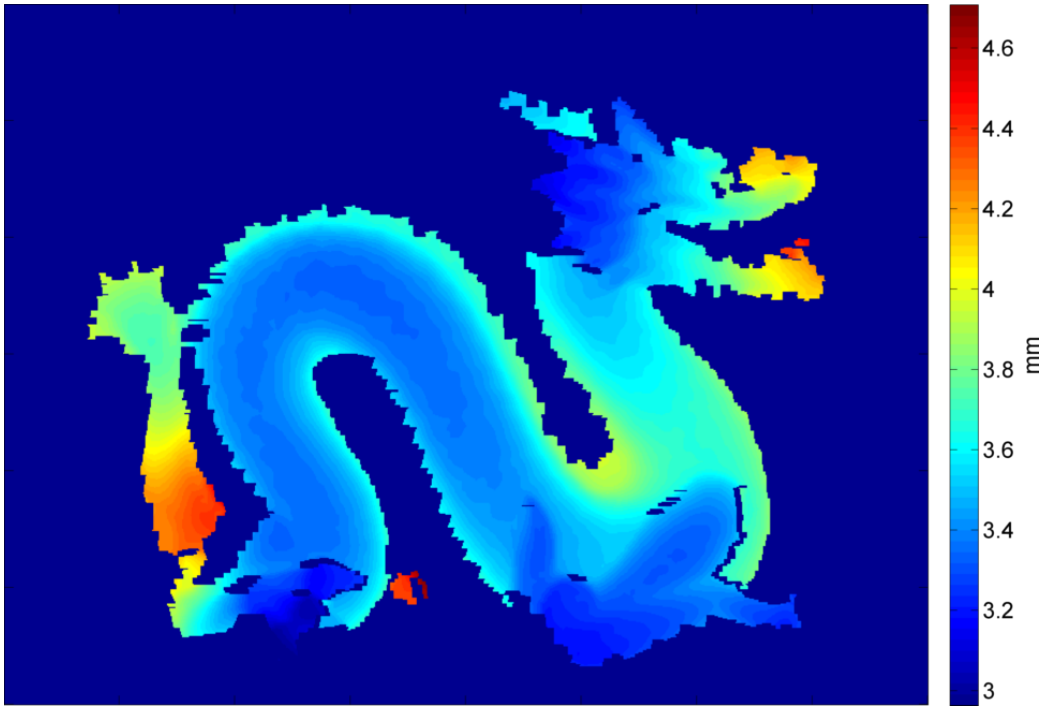


Figure 6.4: Dragon scene sensitivity map. An edge position that is shifted by 1 pixel causes different depth changes depending on the image coordinates and the light plane in question.

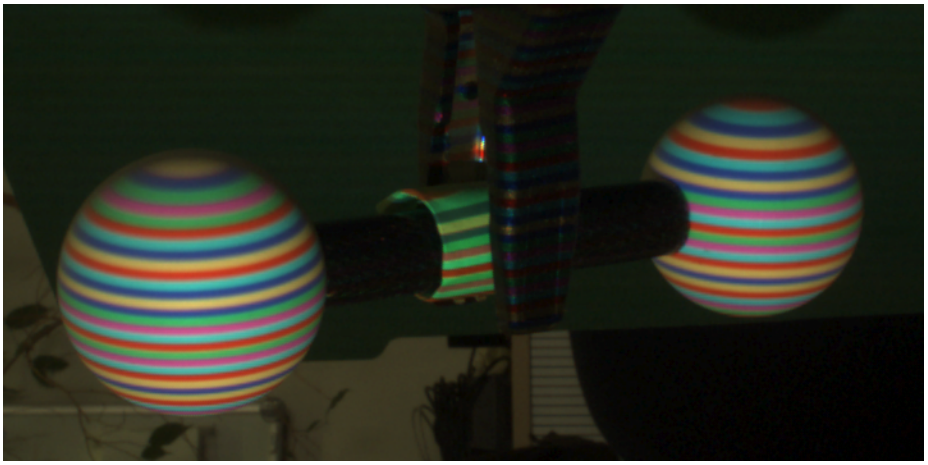


Figure 6.5: Example barbell input image. For this test body the sphere diameters and their distance are known with high accuracy.

The active calbration in general gives better results than the classic method, even if it was not completely active. With a high-quality e-paper display as calibration target, even better results may be expected.

6.1.3 Endoscopic reference object

The accuracy of the prototype endoscopic scanner was evaluated with the help of an artificial cavity in a block of plastic. In this experiment ground truth CAD data is

	classic [mm]	active [mm]
	0.0413	0.0238
	0.0637	0.0159
	0.0329	0.0241
	0.0298	0.0392
	0.0418	0.0245
	0.0235	0.0515
	0.0348	0.0144
	0.0526	0.0172
	0.0429	0.0277
	0.0208	0.0300
	0.0302	0.0225
	0.0273	0.0492
mean	**0.0368**	**0.0283**
sigma	**0.0124**	**0.0123**

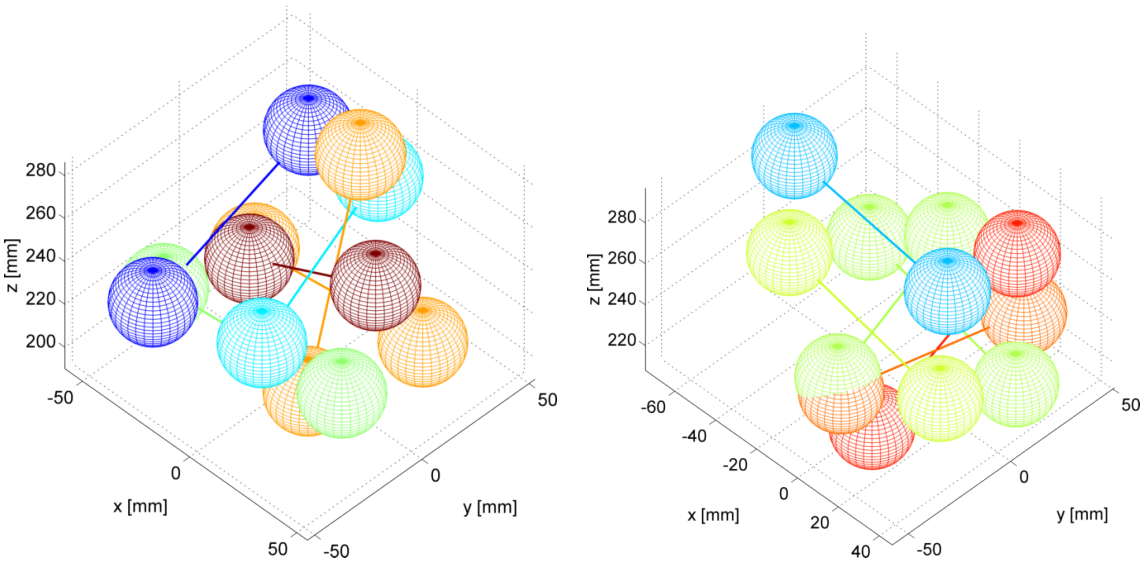
Table 6.4: Barbell test - Sphere fit residuals. The active calibration gives smaller residuals. In both cases a part of the residuals is due to the inherent measurement noise.

	classic [mm]	active [mm]
	0.1774	0.0158
	0.2156	-0.0595
	-0.1024	0.0437
	-0.0819	-0.1126
	-0.2159	0.0296
	-0.1388	0.2015
	-0.1342	0.0197
	-0.0525	-0.0240
	0.0733	-0.0211
	-0.0636	0.1364
	-0.0288	-0.0460
	0.1497	0.2273
mean	**-0.0168**	**0.0342**
sigma	**0.1385**	**0.1043**

Table 6.5: Barbell test - diameter deviations. The active calibration gives a higher absolute mean error but a lower standard deviation.

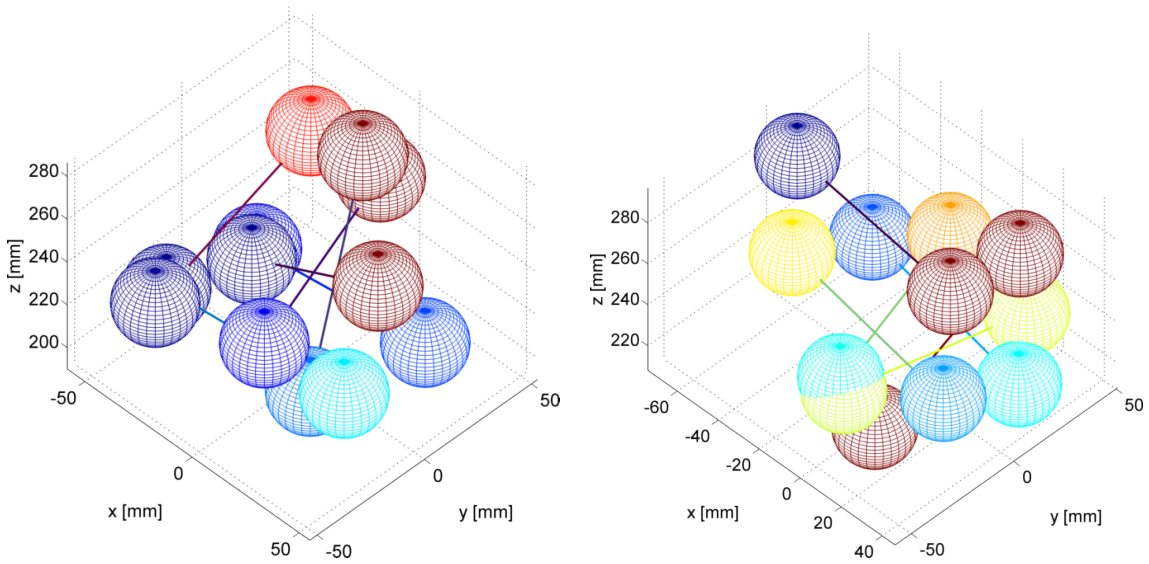
	classic [mm]	active [mm]
	-0.0445	0.1078
	0.0882	0.0195
	-0.1468	0.0209
	0.0915	-0.0707
	0.0066	0.0320
	0.3512	0.1399
mean	**0.0577**	**0.0416**
sigma	**0.1692**	**0.0744**

Table 6.6: Barbell test - distance deviations. The active calibration gives a lower mean and a markedly lower standard deviation. The relatively high standard deviation for the classic calibration method is mainly caused by one outlier.



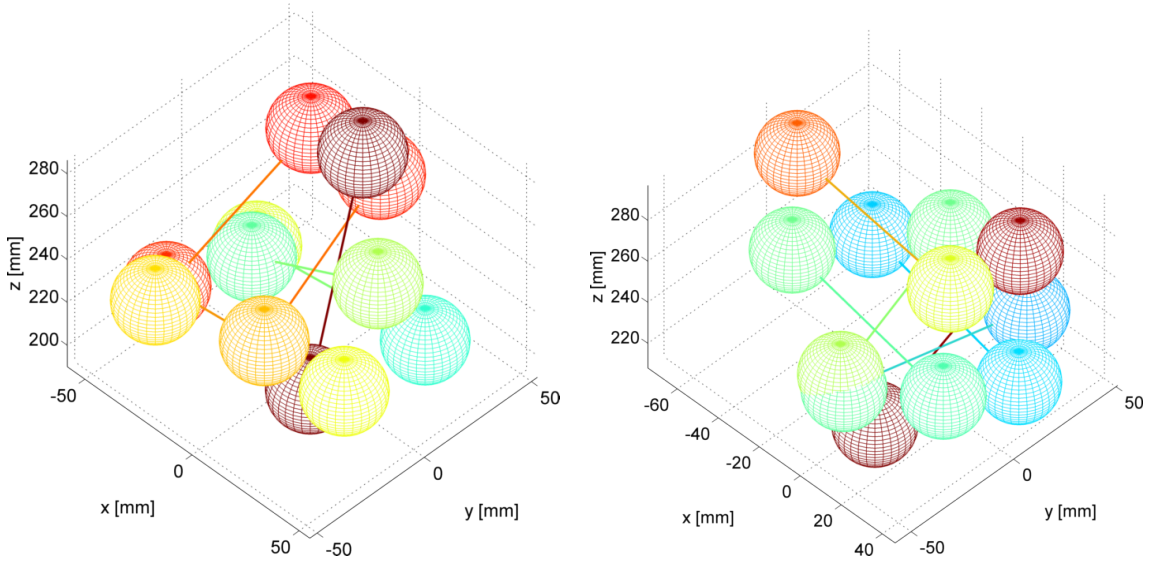
(a) Classic calibration (b) Active calibration

Figure 6.6: Barbell poses. Colors indicate the distance errors. Poses with a high blue component give a distance that is too small, a high red component indicates distances that are too large. The exact numbers can be found in table 6.6.



(a) Classic calibration (b) Active calibration

Figure 6.7: Barbell poses. Color indicates the diameter error. Spheres with a high blue component give a diameter that is too small, a high red component indicates diameters that are too large. The exact numbers can be found in table 6.5.



(a) Classic calibration (b) Active calibration

Figure 6.8: Barbell poses. Color indicates the magnitude of the sphere fit residuals. Spheres with a high blue component have residuals close to zero, the higher the red component, the higher the residuals. The exact numbers can be found in table 6.4.

available for comparison. The cavity surface was reconstructed from a sequence of images acquired while the scanner was moving through the cavity.

For this reconstruction task, the individual point clouds recovered from successive frames have to be registered to each other and merged. The overlap between successive images is very large if the sensor is moving slowly compared to the frame rate of 30Hz. Registration algorithms like ICP [Fitz 03] could be used, but there may be degenerate cases like constant-diameter cylindrical cavities where this algorithm can fail. Therefore we proposed to guide the registration process by motion estimation with help of the second camera (see figure 4.17). The main measurement camera cannot be used for this purpose because the projected pattern moves with the sensor head and masks the underlying scene motion. The auxiliary front camera does not see the pattern and feature tracking or optical flow can be used to derive an initial estimate of the camera translation and rotation between two frames. Unfortunately, because of hardware malfunction, the front camera of the endoscope cannot currently be used. Therefore the scanner was moved in a controlled fashion and the registration of the individual scans was performed with the help of the known fixed offsets between the datasets. The complete workflow for the reconstruction was as follows:

1. Compute a 3D point cloud from every single input frame.
2. Combine the individual point clouds into one by applying the known translation between consecutive frames.
3. Perform thinning by merging points closer than 0.3mm.
4. Smooth the resulting point cloud using the method of Vollmer et al. [Voll 99] with a radius of 1.5mm.

The result for the hollow plastic block is shown in figure 6.9. The input data consisted of a sequence of 41 images. The initial merged point cloud contained 258610 points, after thinning 11323 remained. This final point cloud was registered with the ground truth CAD model using ICP [Fitz 03]. The average error between the reconstructed points and the original CAD data is 92μm. This result is approximately independent of the calibration method used, suggesting that the error is dominated by the edge localization error in the camera image. The cavity has a diameter of around 12mm, so the relative measurement error of the endoscopic scanner is larger than for the desktop system. This can be explained by the difficult imaging geometry and the relatively low resolution and high noise of the camera images. However, for an endoscopic scanner the results are very good. For comparison, [Hu 10] reported an average reconstruction error of 1.68mm at a distance of 20mm to 30mm to the surface for a technique based on Structure-from-Motion.

6.1.4 Comparison with the Kinect scanner

The Kinect 3D sensor system released by Microsoft in the end of 2010 is the only widely available commercial single-shot 3D scanner. However it is geared toward gesture controls for gaming and not for measurement tasks. Nevertheless we tried to compare its output to our results. The test scene consisted of a mostly white calibration plate and a black breadboard. The Kinect has a baseline of about 75mm and a

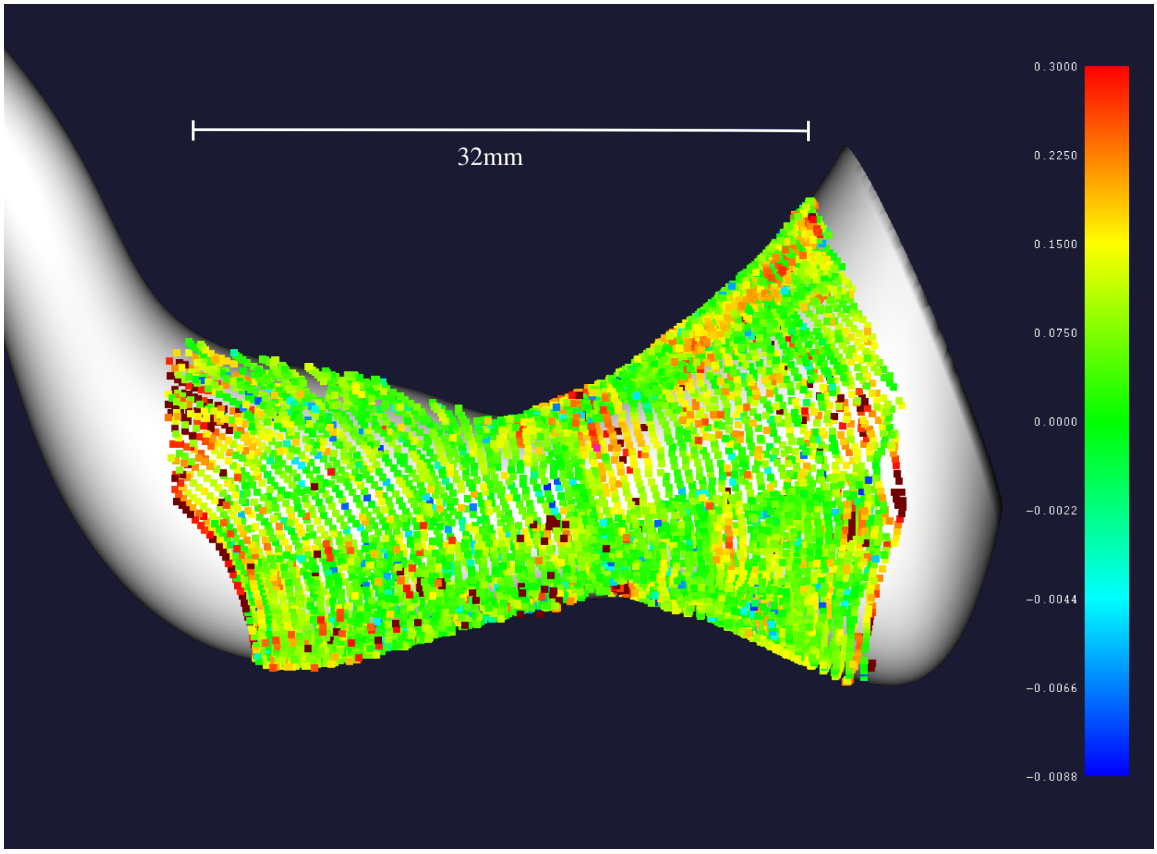
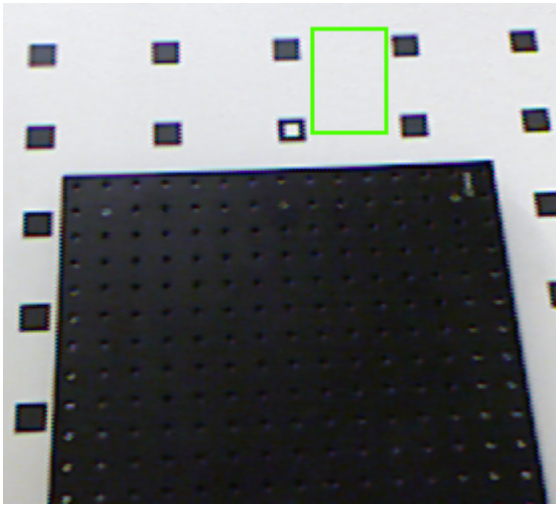
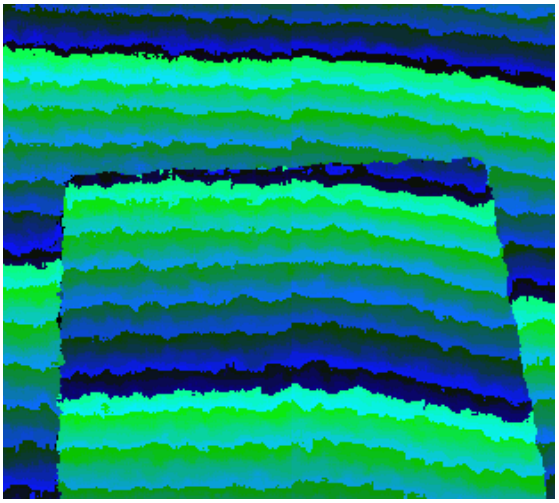


Figure 6.9: Measurement result for an artificial cavity. The colors encode the error relative to the ground truth CAD data (mm). The average error was 0.092mm.

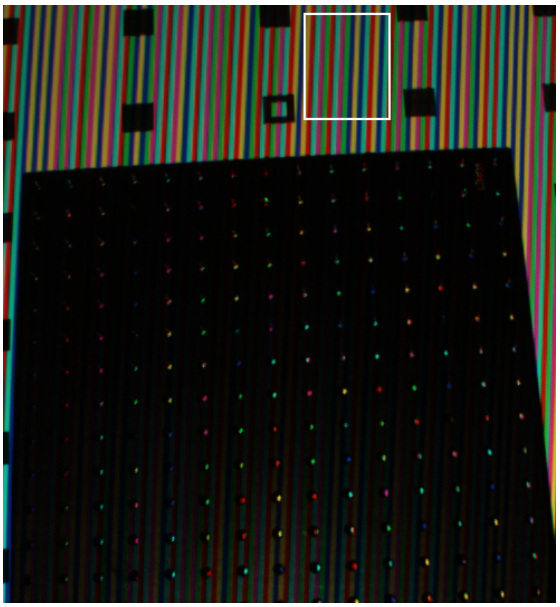
camera resolution of 1280x1024 (although the output is downsampled to 640x480), while our test setup had a comparable baseline length of 80mm and a camera resolution of 780x580. The perspectives on the scene were also close to identical. We evaluated the planarity of the computed depth data on the calibration plate. The plane fits were performed on a small patch only to exclude the influence of calibration errors. The standard deviations of the residuals were 0.167mm for our scanner respectively 0.945mm for the Kinect. One has to keep in mind, however, that the output provided by the Kinect driver software is quantized to 2048 steps. At close range one step corresponds to approximately one millimeter. Furthermore, the depth map generated by the Kinect is heavily smoothed, as can be seen in figure 6.10. The holes in the breadboard have been interpolated over. The Kinect is therefore currently not suitable for high-precision measurements. Still, the high depth of field and the contrast of the projected pattern is remarkable. The same is true for the overall reliability, especially considering that other correlation-based systems [Gock 06, Deve 02] did not perform reliably on textured scenes.



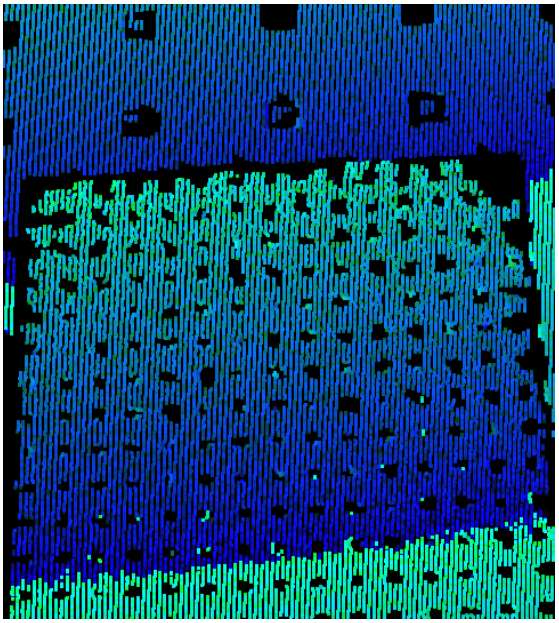
(a) Kinect color image. The green rectangle marks the region for the plane fit.



(b) Kinect depth map. The holes in the breadboard have been interpolated over.



(c) Our color image. The white rectangle marks the region for the plane fit.



(d) Our depthmap without interpolation. The holes in the breadboard are open.

Figure 6.10: Kinect depthmap compared to our result. The plane fit standard deviations are 0.945mm for the Kinect respectively 0.263mm (without interpolation) and 0.167mm (with interpolation) for our scanner.

6.2 Decoding performance

It is not easy to compare the results of the proposed system to other approaches. There are no publicly available reference implementations. Also there is no public database of reference scenes like the Middlesbury Stereo Database. This is understandable since everybody uses different patterns. The reference scenes therefore need to take the form of standardized test objects or virtual scenes that can be rendered with different illumination patterns. We built a few such virtual scenes in Povray and made them publicly available, but met with litte resonance. Therefore we reimplemented the decoding method presented in [Zhan 02] for a comparison (section 6.2.1).

A simple approximate comparison is possible to the system developed by Koninckx [Koni 05a]. He helpfully included a test with a standard object in the form of packing foam in his PhD thesis. The comparison shown in figure 6.11 is favorable for the proposed pattern and decoding algorithm.

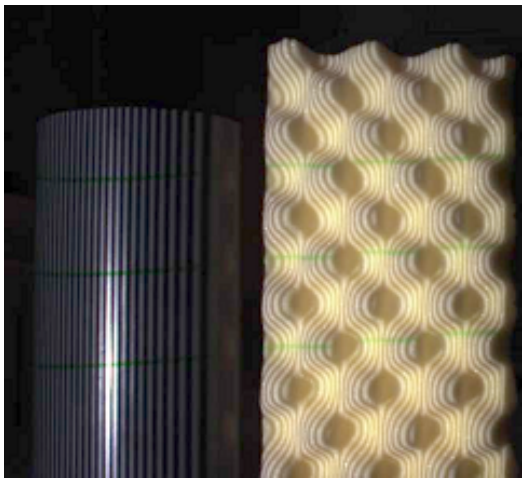
For the endoscopic sensor, comparisons are very difficult, since no other 3D scanning system of similar size was available. We therefore show the reconstructed surfaces of a colon phantom and a lamb's windpipe for qualitative assessment.

6.2.1 Comparison with Dynamic Programming

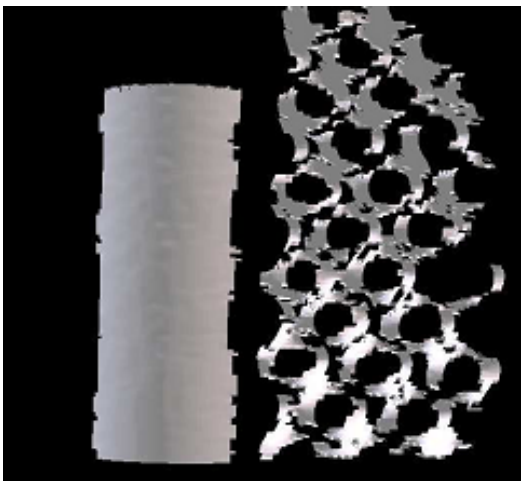
We implemented the decoding method presented in [Zhan 02] for a comparison with our results. This method relies on the ordering constraint, that is, it assumes the observed order of the stripes is the same as the projected order. This is true for simple objects but does not necessarily hold in complex scenes with occlusions. Under the ordering constraint the most plausible correspondence of projected and observed edges is determined recursively with Dynamic Programming. This is performed in each scanline individually. In the following we present decoding results obtained with our algorithm and with the Dynamic Programming-based algorithm.

First we evaluated some synthetic and real test images to see the influence of texture and general image quality. The results are shown in figure 6.12 and figure 6.13. Our method yields better results than Dynamic Programming decoding in all cases.

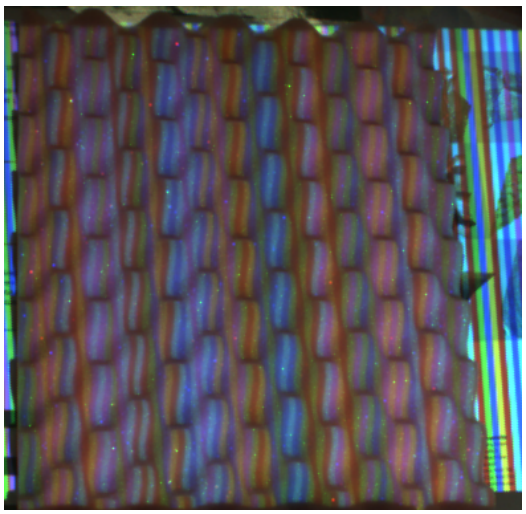
On synthetic scenes it is also possible to perform a more thorough evaluation of the number of inliers and outliers in the recovered depth data. We used three different scenes. The 'grid' features a non-smooth surface with many holes but no texture. The 'dragon' has a smooth surface with some shading variations and a black background where no false positives should be detected. Finally, the 'sun' scene has a dominant texture with non-neutral color. Each scene was rendered with different levels of pattern contrast and additive white gaussian noise. We evaluated the number of correct depth values recovered from each image and the ratio of outliers to inliers. An outlier is defined as having a depth error of greater than 10mm. For the simulated setup this corresponds to a localization error of about 2 pixels. This can result from extreme noise but most likely points to misidentifications. Details of the various test scenes are shown in figure 6.14.



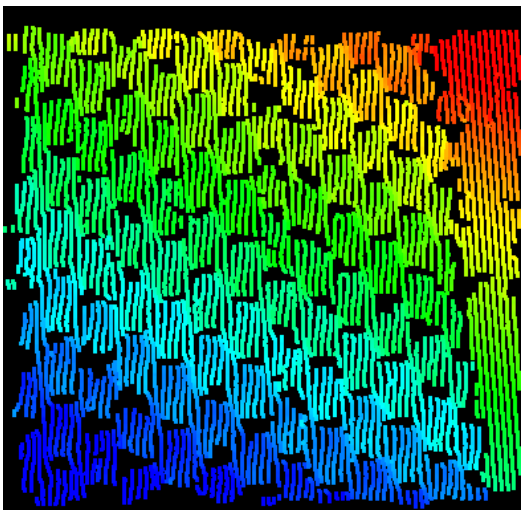
(a) Input image of Koninckx. Only the right part is relevant.



(b) Interpolated and rendered result image of Koninckx' system.

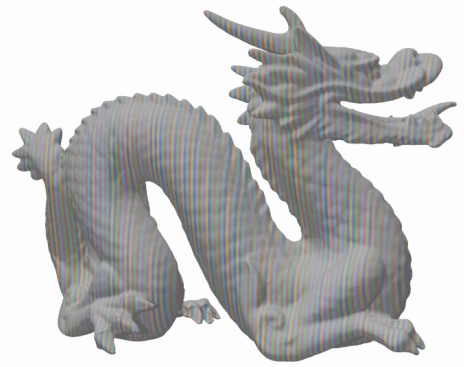


(c) Input image for the proposed algorithm

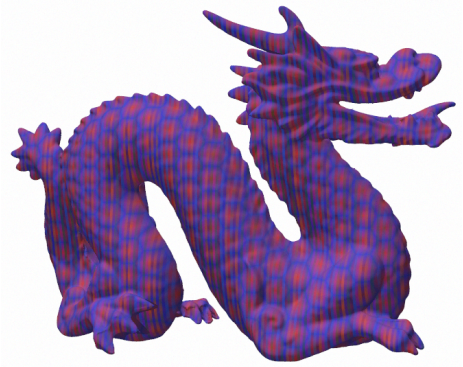


(d) Color coded result image of the proposed algorithm. Uninterpolated, but dilated for better visibility.

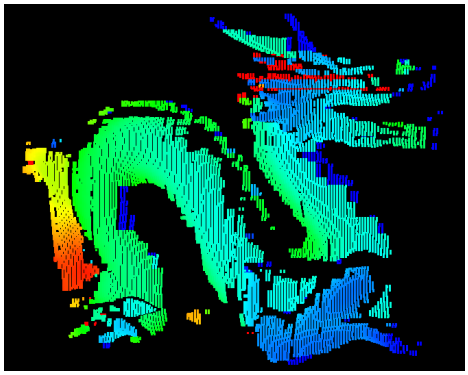
Figure 6.11: Qualitative comparison of Koninckx's results (taken from [Koni 05b]) to ours. While the top left image seems to have been taken in a controlled setting without ambient light, the bottom left input image was taken under daylight indoor conditions. Still, the result on the bottom right side covers the foam almost completely while the top right side shows large unrecovered areas.



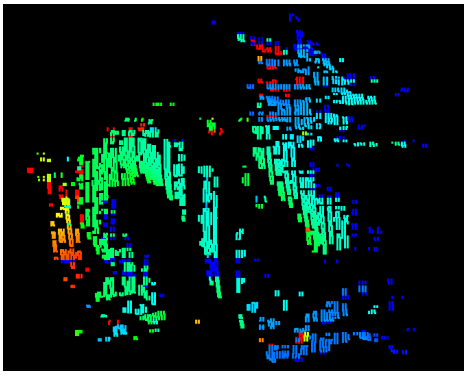
(a) Dragon without texture



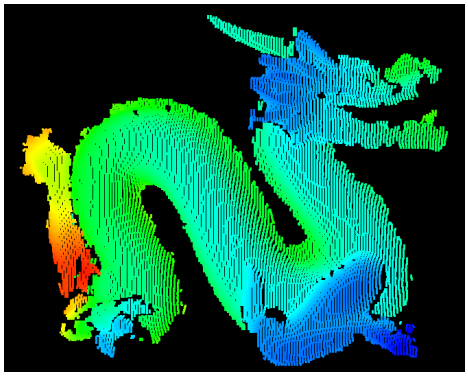
(b) Dragon with texture



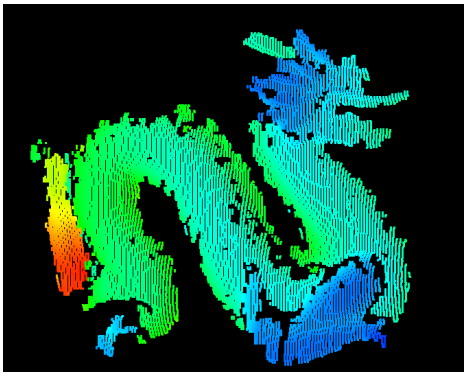
(c) Dynamic Programming result without texture



(d) Dynamic programming result with texture



(e) Result of proposed algorithm without texture

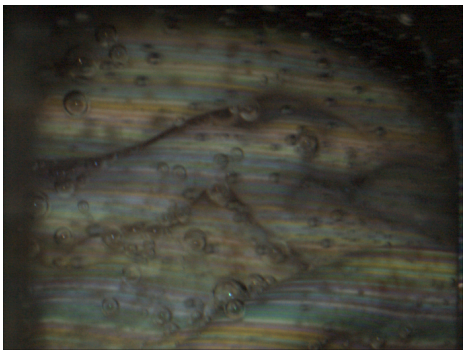


(f) Result of proposed algorithm with texture

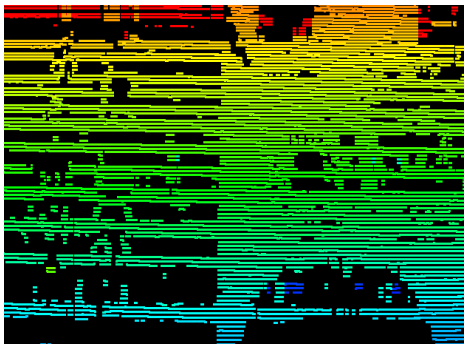
Figure 6.12: Synthetic example scene in neutral color and with texture. Already in the former case Dynamic Programming decoding produces some errors, but in the latter the results are unusable.

As can be seen in figures 6.15, 6.16 and 6.17, our decoding algorithm produces more inliers and less outliers. In situations with high contrast and low noise, Zhang's approach can compete, but it breaks down with low quality images. Furthermore, it is relatively slow. The runtime is given as one minute per frame in the original paper, on current hardware our implementation needs 8 seconds for an image with 800x600 pixels. For our algorithm we used the default parameters given in table A.2.

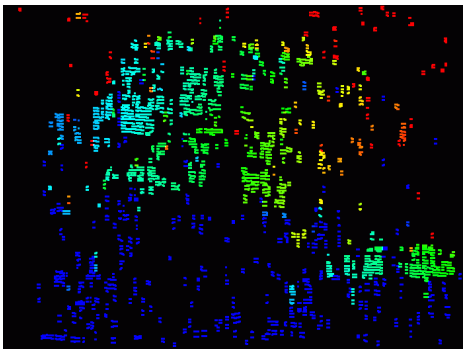
(a) Book cover with texture



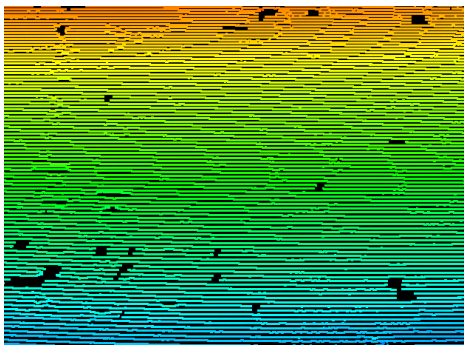
(b) Pig stomach with bubbles



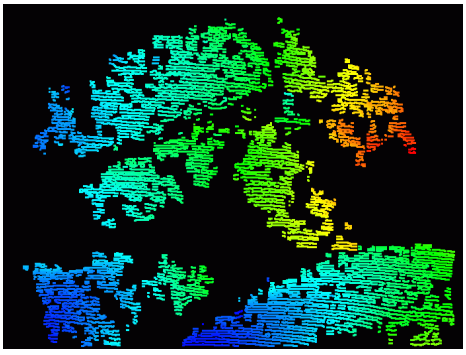
(c) Book result with Dynamic Programming



(d) Stomach result with Dynamic Programming decoding

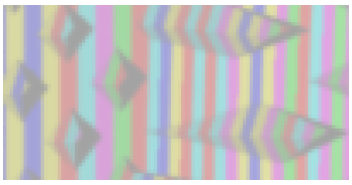


(e) Book result with proposed algorithm

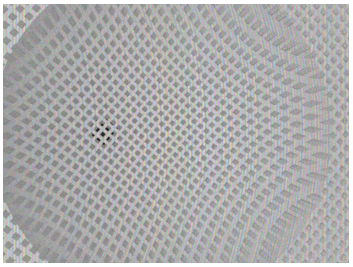


(f) Stomach result with proposed algorithm

Figure 6.13: Real example scenes and decoding results for the proposed algorithm and Dynamic Programming. Our results contain more correct depth data and less outliers.

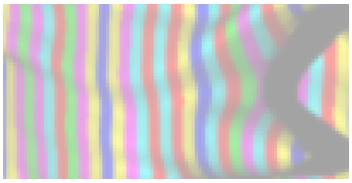


(a) 'Grid' detail with highest contrast pattern and lowest noise.



(b) 'Grid' with medium contrast pattern and medium noise.

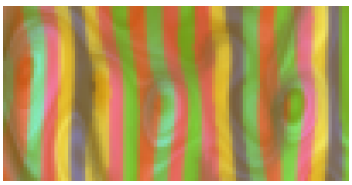
(c) 'Grid' detail with lowest pattern contrast and highest noise level.



(d) 'Dragon' detail with highest contrast pattern and lowest noise.

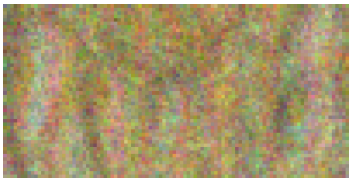
(e) 'Dragon' with medium contrast pattern and medium noise.

(f) 'Dragon' detail with lowest pattern contrast and highest noise level.



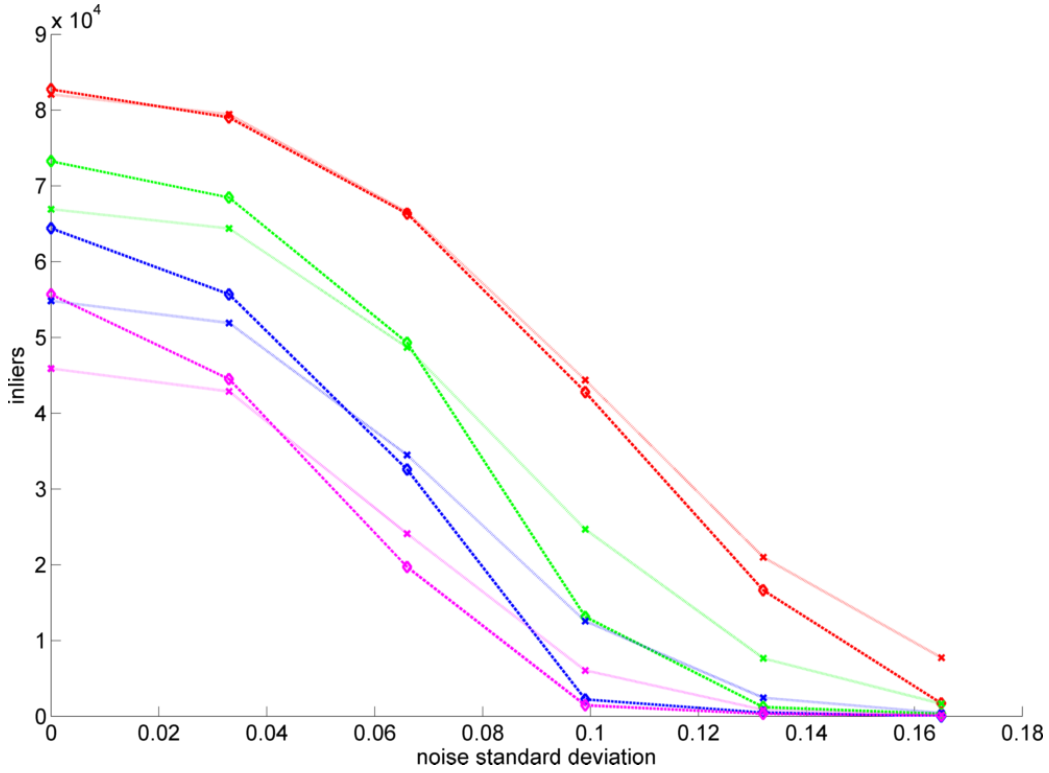
(g) 'Sun' detail with highest contrast pattern and lowest noise.

(h) 'Sun' with medium contrast pattern and medium noise.

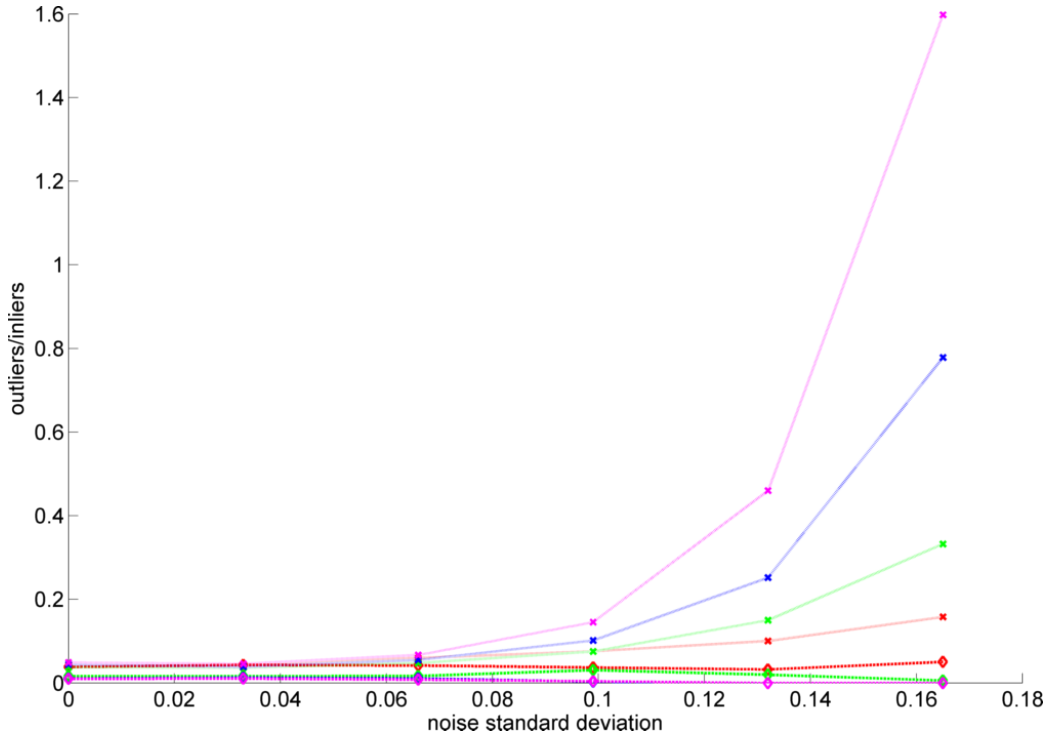


(i) 'Sun' detail with lowest pattern contrast and highest noise level.

Figure 6.14: Overviews and details of the different test scenes for different levels of noise and contrast.

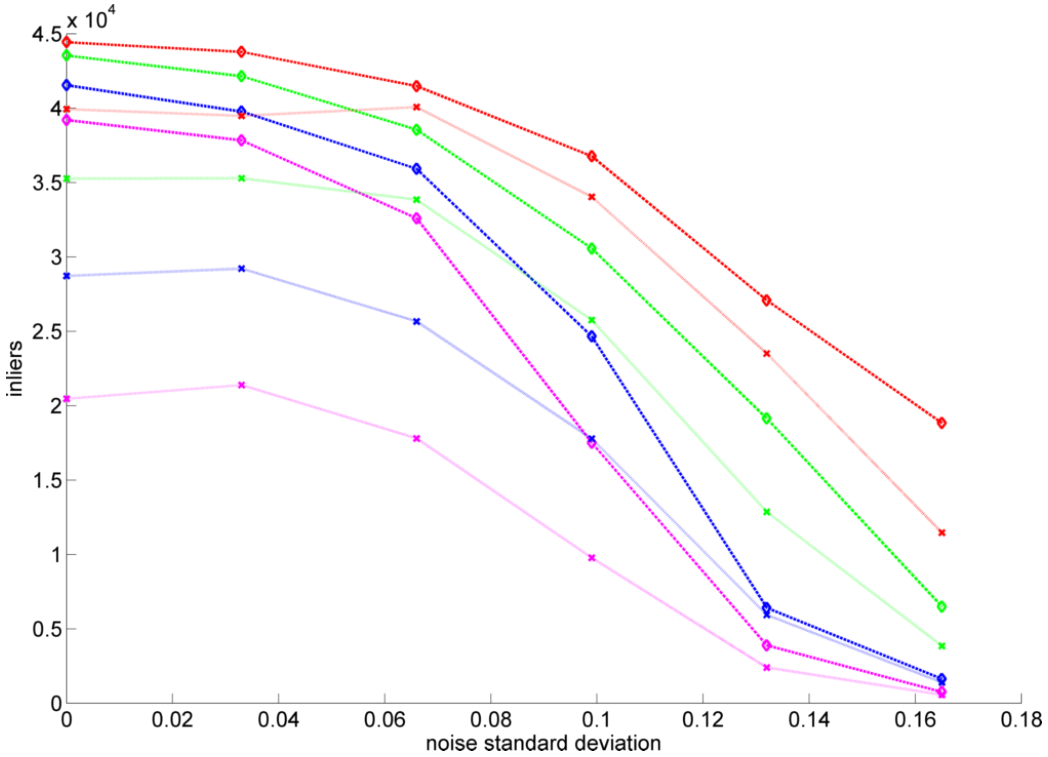


(a) Number of inliers for different levels of contrast and noise. At the highest contrast level (red) and with low noise, both methods are equal. At lower contrast levels with low noise our algorithm recovers more points. Notably, at high noise levels our method detects fewer points, but also generates very few outliers (compare figure 6.15b).

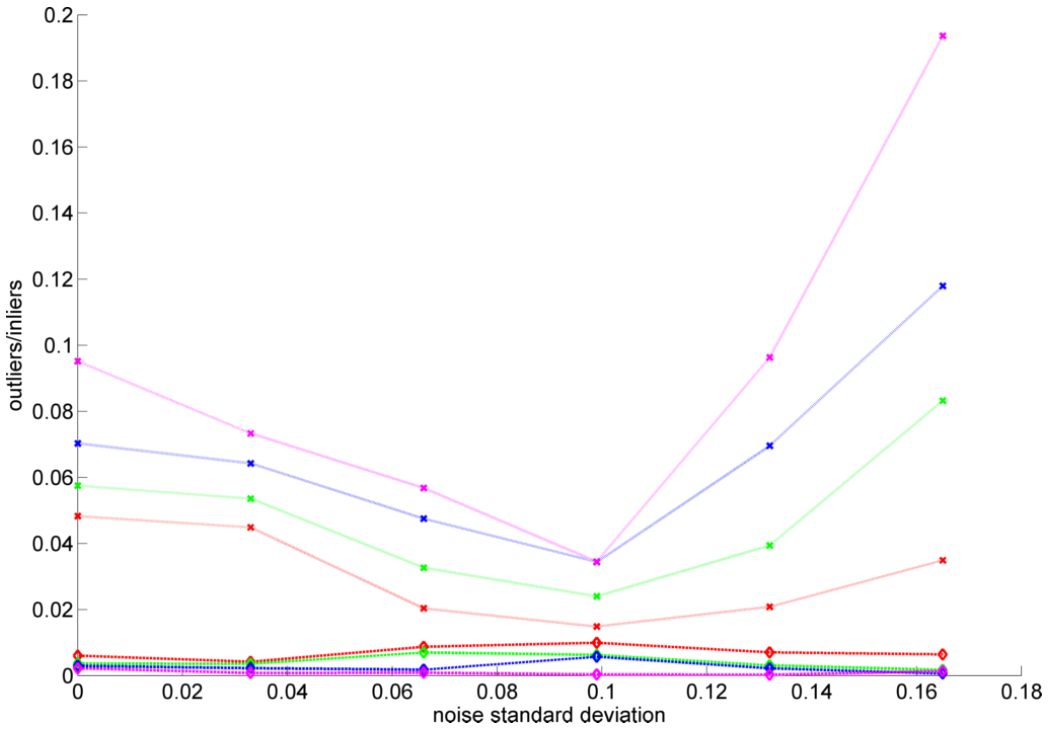


(b) Ratio of outliers to inliers for different levels of contrast and noise. Our method results in very few outliers at all contrast and noise levels while with DP decoding a considerable fraction of the generated depth data is in fact wrong.

Figure 6.15: Comparison for the 'grid' scene. The contrast levels are 1 (red), 0.63 (green), 0.45 (blue) and 0.35 (magenta). The diamonds with dashed lines represent the results of the proposed algorithm. The crosses with dotted lines are the Dynamic Programming results. Outliers are defined as points deviating more than 10mm from the ground truth.

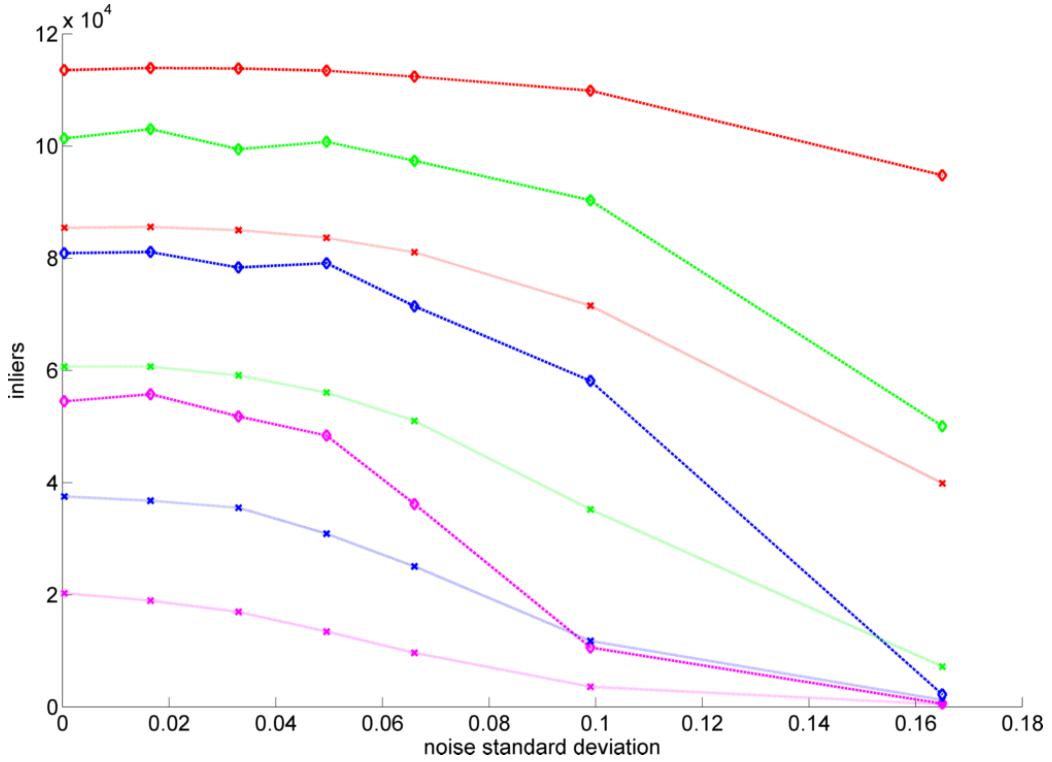


(a) Number of inliers for different levels of contrast and noise. Our algorithm recovers more points at all levels of noise and contrast.

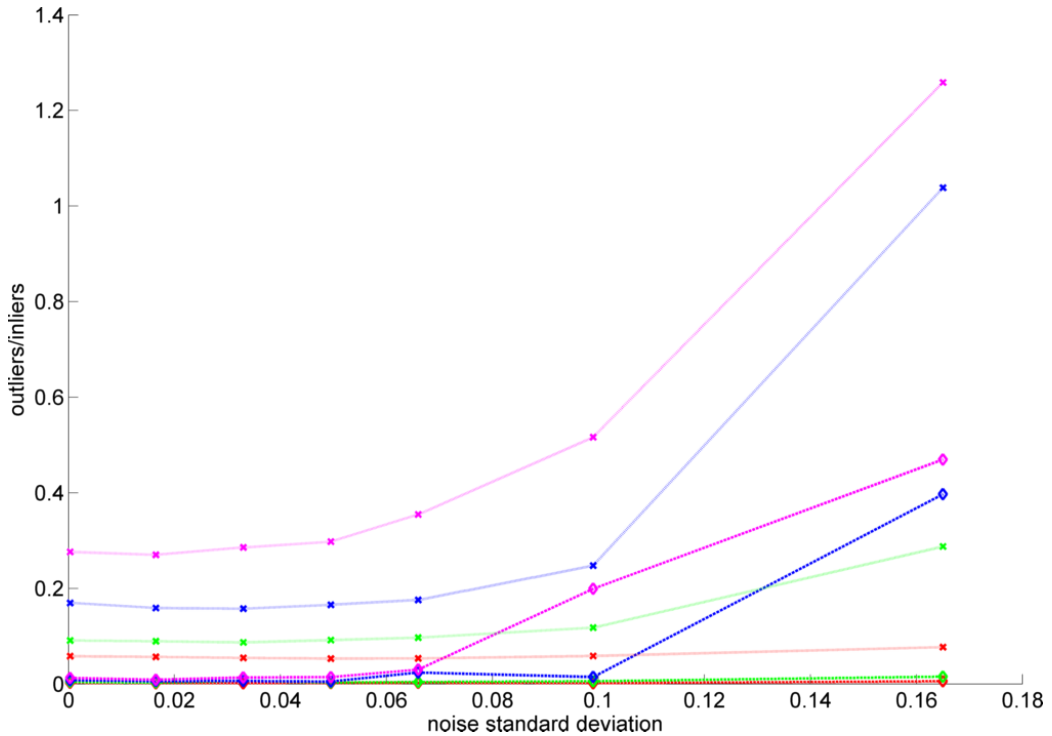


(b) Ratio of outliers to inliers for different levels of contrast and noise. Our algorithm produces fewer outliers at all levels of contrast and noise. Interestingly fraction of outliers is actually the highest for the lowest noise level. This is due to points at the very border of the object where the surface gradient is high (see 6.14). A small edge localization error thus can cause large deviations from the ground truth. With lower image qualities these border regions are skipped.

Figure 6.16: Comparison for the 'dragon' scene. The contrast levels are 1 (red), 0.63 (green), 0.45 (blue) and 0.35 (magenta). The diamonds with dashed lines represent the results of the proposed algorithm. The crosses with dotted lines are the Dynamic Programming results. Outliers are defined as points deviating more than 10mm from the ground truth.



(a) Number of inliers for different levels of contrast and noise. Our algorithm recovers more points at all levels of contrast and noise.



(b) Ratio of outliers to inliers for different levels of contrast and noise. Although some decoding errors occur at low contrast and high noise levels our algorithm again produces a lower fraction of outliers at all levels of contrast and noise.

Figure 6.17: Comparison for the 'sun' scene. The contrast levels are 1 (red), 0.63 (green), 0.45 (blue) and 0.35 (magenta). The diamonds with dashed lines represent the results of the proposed algorithm. The crosses with dotted lines are the Dynamic Programming results. Outliers are defined as points deviating more than 10mm from the ground truth.

6.3 Endoscopic measurements

For the endoscopic scanner there was no access to alternative measurement devices, therefore a direct comparison is difficult. However, a visual inspection of some reconstructed surfaces shows their satisfactory quality. The workflow for the reconstruction was identical to section 6.1.3, except for one additional step: After the smoothing we performed a Poisson surface reconstruction [Kazh 06]. The decoding results of a single image of the windpipe sequence are shown in figure 6.18. Because of the special hardware arrangement (see figure 4.17) not the complete area of the image can be used for the measurement. The dark area in the center is caused by the sensor chip looking at itself in the mirror. The dark area at the top is the shadow of the camera's data cable. In the corners of the image, the rays of view bypass the mirror and no pattern can be observed.

6.3.1 Colon Phantom

A rubber replica of a human colon was measured with the endoscopic sensor. The colon diameter was approximately 40mm and therefore at the upper limit of the current sensor prototype. Nevertheless, good reconstruction results could be obtained. A sequence of 50 frames was recorded, from which 267260 points could be recovered. After registration the average point distance was 0.108mm. A thinning step reduced the number of points to 93587. Next, a Poisson surface reconstruction was performed, which resulted in a watertight mesh without any holes. From this, the large artificial faces closing the holes were removed, giving a final surface consisting of 39288 vertices. The reconstructed shape clearly shows the folds of the colon (figure 6.19). Unfortunately each fold also causes a shadow, leading to some holes in the data.

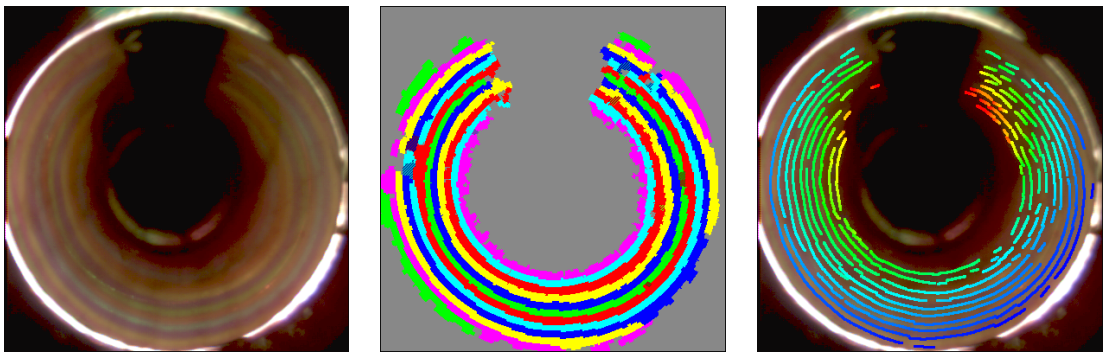


Figure 6.18: Decoding of a single frame of the windpipe sequence. The input image on the left has been gamma adjusted for better visibility. The center image shows the projected ring colors as detected by the algorithm. On the right side the recovered range data is color coded and overlaid on the input image. The color scale ranges from 4mm to 18mm.

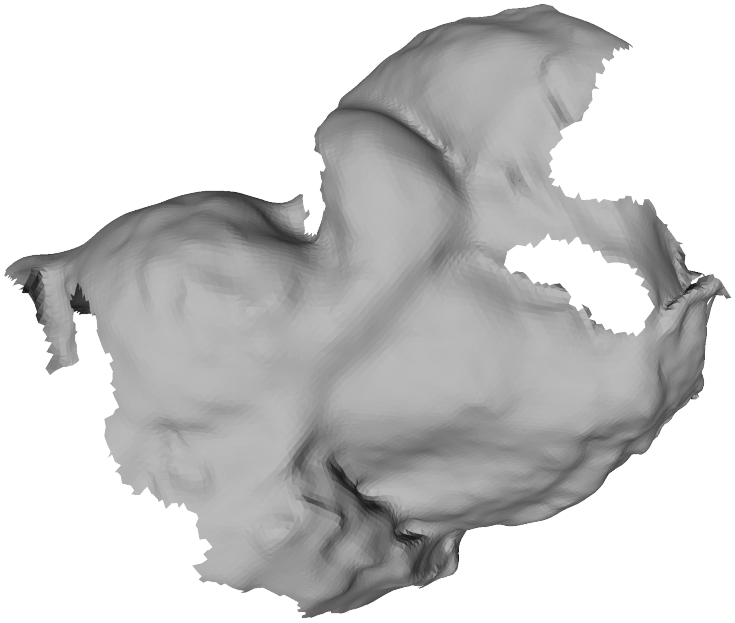


Figure 6.19: Colon surface reconstructed from 50 frames. The folds are clearly visible, but also cause shadows which result in holes in the recovered surface.

6.3.2 Windpipe

Figure 6.20 shows the measured surface of the windpipe, which has a diameter of approximately 14mm. The input data consisted of a sequence of 26 frames. These images yielded 131146 points with an average distance of 0.057mm. The point density is markedly higher here because of the smaller diameter compared to the colon phantom. After thinning, 7417 points remained and were again used for a Poisson surface reconstruction. Overly large faces were removed from the mesh. The result shows that the sensor works even on biological surfaces, which can be difficult because of volume scattering and highlights. The data quality is very promising. Even the ripples at the 'bottom' side could be recovered.

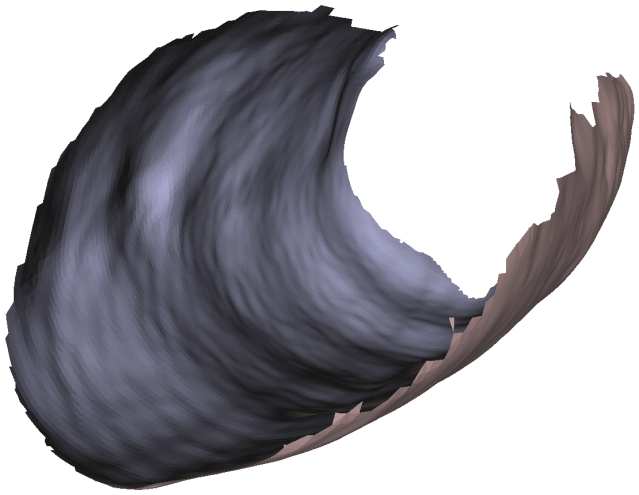


Figure 6.20: Inner surface of a lamb's windpipe reconstructed from 26 images. No additional smoothing was applied. Note the recovered longitudinal ripples at the bottom. The missing area at the top is due to the camera connection cable.

6.4 Runtime

There are a number of details relevant for an efficient and fast implementation of the algorithm described above. The first concerns the superpixel colors. As described above, we compute the median of the colors of all image pixels belonging to the corresponding segmentation region. This is however relatively slow. Using a simple median-of-five or median-of-nine filter around the region seed pixel is faster, needs less memory, and gives very similar results.

The next issue is a trade-off of computation versus memory. The edge symbol match probabilities, color enhancement edge probabilities and the ray-plane intersections can be precomputed and stored in lookup tables. The former two are very small, but the latter requires a large amount of memory. We need to cache $w \cdot h \cdot n$ depth values, where w is the camera image width, h the camera image height and n the number of projected stripes. As the depth varies smoothly with the image coordinates, it is possible to reduce the memory consumption with little quality loss by subsampling the camera image and using interpolation.

Another speedup can be achieved by subsampling the gradient magnitude before the watershed segmentation. Typically, the image changes only slowly in the direction along the stripes. Averaging every two pixels in this direction yields a segmentation results with fewer superpixels. The decoding results on this reduced graph are comparable to the full results. In fact, in some cases they may be better. The runtime of the decoding stage is linear in the number of superpixels. Table 6.7 gives example timings for the first image of test sequence 6. Figure 6.21 shows the decoding results for the whole sequence.

Table 6.8 sums up some approximate timing results for the different stages of the current implementation of our algorithm. While the image processing part can be

	LR [ms]	HR [ms]
build graph	149	244
color enhancement	87	142
identify	161	271
number of superpixels	12619	21553
µs / superpixel	31	30

Table 6.7: Decoding stage runtime dependency on the number of superpixels on a 2GHz Intel Core2.

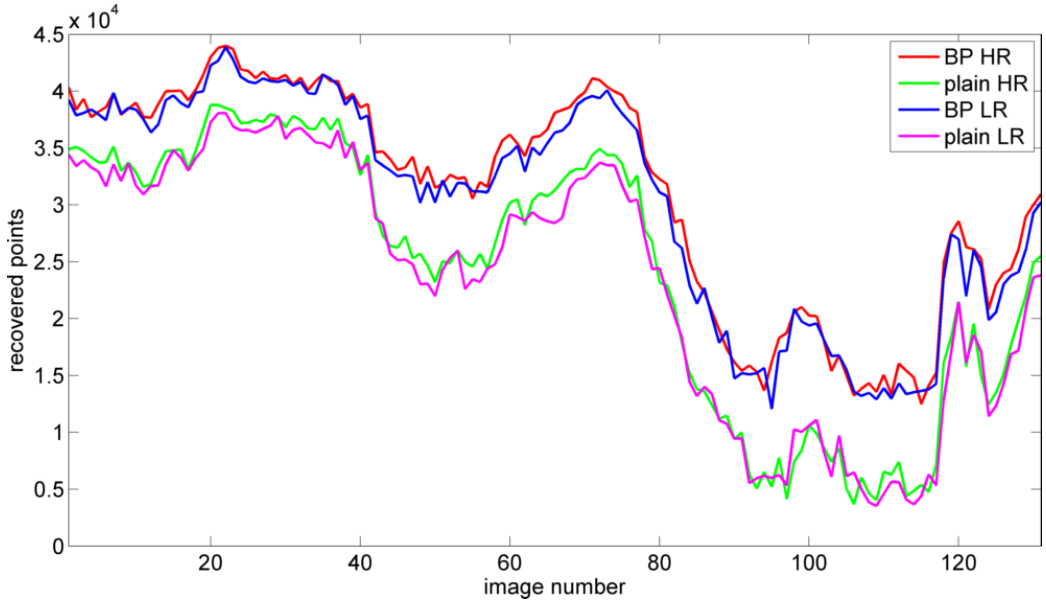


Figure 6.21: Results for high-resolution and low-resolution segmentation. Overall, the former is marginally better, but not on each individual image.

performed faster on a GPU, the total speedup is limited by the non-parallelized parts of the algorithm. Table 6.9 gives the final framerates reached on current hardware. We can conclude that the goal of real-time operation has been reached.

	CPU [ms]	GPU [ms]
gradients	15	4
watershed	15	2
copy	-	6
build graph	34	
color enhancement	64	
identify	28	

Table 6.8: Approximate timings of some major processing steps for a 780x580 image with approximately 13000 watershed regions. The initial steps can be executed on the CPU (Intel Q9650) or the GPU (NVidia GTX 285). The final steps are only implemented on the CPU. If the GPU is used for image processing, the intermediate results have to be copied back to the host. This step takes just as long as the actual computation in this example. The total time is 156ms on the CPU and 138ms on GPU+CPU.

	CPU only [fps]	CPU+GPU [fps]
plain decoding	25	31
color enhanced	18	21

Table 6.9: Performance of the proposed system on a 3Ghz quad-core Intel Q9650 CPU and with an NVidia GTX 285 GPU. Multithreading allows to process several images in parallel.

Chapter 7

Conclusion

The goal of this thesis was to develop a simple, accurate, robust and fast 3D scanning methodology that can operate in dynamic and uncontrolled environments. In the first part of the work we presented a literature review and concluded that Single-Shot Structured Light is a suitable measurement principle, provided that the pattern design and the decoding algorithm are sufficiently sophisticated. In the next chapters we introduced the calibration methods we applied to provide high-accuracy 3D data, and a robust algorithm for the decoding of Single-Shot Structured Light patterns that provides reliable range data even for low-quality input images. The basic challenge of Single-Shot Structured Light methods is to detect the projected pattern in the scene despite possible distortions and disturbances that have to be expected. The proposed method works on textured objects as well as on non-smooth objects and can cope with external light, low pattern contrast, high camera noise and other artifacts. One compromise that has to be made is that the resolution in lateral and in z-direction is lower than that which can be achieved by multi-shot methods. For our intended applications, however, the demonstrated resolution is sufficient and an acceptable trade-off for the simplified hardware setup and the immunity to sensor and scene motion.

The proposed decoding algorithm performs a superpixel segmentation of the input image and builds a region adjacency graph from the result. The color estimates for the superpixels are based on ensembles of image pixels, which enhances the robustness against noise. Once the correspondence problem has been solved for one seed region, the information is propagated through the graph in a best-first-search. If there are artifacts in the image, it is possible to find paths around them. The algorithm does not use any fixed thresholds for color changes but adapts automatically to the local pattern contrast. Furthermore, inference algorithms like Belief Propagation can be used to estimate the projected colors from the observed colors and recover more 3D data from the input images. The performance of the decoding algorithm was tested in a variety of situations. It consistently gave better results than a reference implementation of an alternative decoding algorithm. With a camera resolution of 780×580 pixels, we can typically recover around 50000 3D points per input frame.

Of course, the graph-based decoding approach still has its limitations. While it gives much better results that the Dynamic Programming-based method, there still comes a point where the image quality is so low that no depth data can be recovered.

The basic assumptions of the decoding algorithm are that each superpixel covers only image pixels from one projected stripe, and that it borders directly on superpixels from adjacent stripes. When the stripe contrast is too low and the noise level too high, these assumptions no longer hold. However, under such extreme conditions, other approaches suffer even worse. One advantage of the proposed algorithm is that it degrades gracefully. False depth values are very rare and in the worst case no data is recovered. Unfortunately, low image quality not only reduces the number of 3D points that can be computed from the image, but it also reduces the accuracy of the final 3D data. Thus all efforts to increase the robustness further are in vain if the noise in the resulting depth data is too high for the given application.

Next to the stripe contrast, the major factor determining the accuracy of the recovered range data is the quality of the sensor calibration. We implemented and tested a calibration method based on active targets, which gives superior results compared to calibration with classic targets. For example, the length of a barbell reference object could be measured with a relative mean error of 0.052%. One open issue here is the projector calibration with active targets, which will need e-paper displays to work, but promises even more accurate sensor calibrations.

A showcase application requiring both robust decoding and elaborate calibration is endoscopic 3D scanning. It is especially challenging because of the mandatory small size of the sensor. To the best of our knowledge the presented endoscope is the first 3D endoscope based on Single-Shot Structured Light as well as the smallest Structured Light set-up so far. The endoscope does not contain moving parts and can be realized cost-efficiently. It acquires 3D data at 30 Hz with minimal lag and is not affected by movement. The reconstruction accuracy of about 0.1 mm is very competitive.

All in all, the proposed Single-Shot Structured Light approach offers a solution for 3D acquisition of a vast variety of close-range scenes. The sensor principle is scalable from millimeter to meter range. Typical measurement errors are in the range of $\frac{1}{1000}$ of the working distance. A promising direction for future developments are versatile, lightweight, handheld 3D sensors. The challenges here include the data transmission and reliable real-time registration of the individual point clouds from each frame. Hardware improvements are also an option. Narrowband, high-power LED illumination together with matched, custom filter arrays in the camera would be a big step for a further improvement of the input image quality, with corresponding gains in the amount and accuracy of 3D data that can be recovered. On the decoding side, currently each image is decoded separately and no assumptions on the scene are made. It should be possible to increase the performance by making assumption on maximum depth gradient in the scene and integrating evidence collected over several frames, so that even partial codewords can be decoded. Going even further, scene understanding algorithms that can recognize the objects in the scenes and segment them into piecewise continuous surfaces could give another boost to the decoding.

Appendix A

Appendix

A.1 List of Publications

Christoph Schmalz:
Decoding Color Structured Light Patterns with a Region Adjacency Graph
(DAGM 2009)

Christoph Schmalz, Elli Angelopoulou:
Robust Single-Shot Structured Light
(PROCAMS Workshop at CVPR 2010)

Christoph Schmalz, Elli Angelopoulou:
Belief Propagation for Improved Color Assessment in Structured Light
(DAGM 2010)

Christoph Schmalz, Frank Forster, Elli Angelopoulou:
Camera Calibration: Active versus Passive Targets
(Optical Engineering, Volume 50, Issue 11, 2011)

Christoph Schmalz, Frank Forster, Anton Schick, Elli Angelopoulou:
An Endoscopic 3D Scanner based on Structured Light
(Medical Image Analysis, to appear)

A.2 Example Patterns

Our decoding algorithm does not require a specific pattern. In fact, depending on the application, different patterns should be used. The only restriction is that the pattern must consist of stripes. The following table lists some example patterns with different properties. They are the length (L), the code word size in stripes (w), the minimum Hamming Distance between two code words (h), the alphabet size (c) and the compatibility rules (which color channels must change between neighboring stripes). All patterns except the last are circular, all patterns except the first fulfill the normalization condition. For some patterns an application is given, the others are for reference.

purpose	L	w	h	c	rules	pattern
stomach	106	4	2	8	any two, no normalization	12434170350712521270 56063061650343524707 21605253070607471617 27430534217412142506 53563536347427147241 636036
–	104	4	2	8	any two	16524365070716160560 34163430534274170360 72174356127124125307 42142506536172563061 434724714705270635035252
simulation	112	6	2	6	green	16534216124343425243 46161643524316434356 16524352561342561256 16134316521652525216 43161612534253461346 125253435216
–	52	5	2	6	green	13435253421616534316 12561243425216525243 561643434616
–	90	4	2	6	any one	13421432452412612435 62523431641615216516 21561361423542536532 51256352652543643461 6314634534
–	90	5	2	6	any two	12434161421634125214 34361436524165343525 24356163521616525342 52563563434214256125 3653616124
–	90	4	1	6	any two	16356352163416536534 25214256343614342143 56125243652416165252 53616142161243434124 1253435256
–	42	4	2	6	any two	14253652416534125256 161434216356343612435 2
–	42	3	1	6	any two	12416356142536125243 52161652563421436534 34
endoscopic	15	3	2	6	any two	435256134216435

Table A.1: Example patterns with different properties.

A.3 Endoscopic Calibration

We present a short derivation of the most important formulas used for calibration and measurements with the endoscopic Structured Light 3D scanning system. A more complete treatment of these raytracing fundamentals can be found in [Glas 89]. All rays are expressed in camera coordinates. This coordinate system has the x and y axes in the image plane, the z axis along the optical axis and the origin in the optical center of the camera.

Rays of view

The ray of view for image coordinates (x_i, y_i) in a pinhole camera is the set of all points X with

$$X = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} (x_i - c_x)d_x \\ (y_i - c_y)d_y \\ f \end{pmatrix} = O_p + \lambda T_p \tag{A.1}$$

with the pixel pitch (d_x, d_y), the principal point (c_x, c_y), the focal length f and the free parameter $\lambda > 0$. If the camera exhibits image distortion, it has to be corrected first, for example using Zhang's model [Zhan 00]. In the simple pinhole model this suffices to get the final ray of view. The distance d of a point P to the ray is

$$d = \|(O - P) \times T\| \tag{A.2}$$

The sum of these distances over all calibration points is minimized in the calibration of the camera.

Reflection and refraction

A sphere with the center C and radius r is the set of points X with

$$(X - C_m)^2 = r_m^2 \tag{A.3}$$

Plugging eq. A.1 into eq. A.3 and simplifying, the intersection point of a ray with the sphere must fulfill

$$a\lambda^2 + b\lambda + c = 0 \tag{A.4}$$

with $a = T_p^2$, $b = 2(O_p - C_m) \cdot T_p$ and $c = (O_p - C_m)^2 - r_m^2$. Ignoring degenerate cases, the desired intersection point with the mirror is

$$O_m = O_p + \frac{-b - \sqrt{b^2 - 4ac}}{a} T_p \tag{A.5}$$

The surface normal in this point is

$$N_m = \frac{O_m - C_m}{\|O_m - C_m\|} \tag{A.6}$$

And the reflected ray

$$X = O_m + \lambda \left(T - 2N_m \left(N_m \cdot T\right)\right) = O_m + \lambda T_m \tag{A.7}$$

The glass housing is modeled as a pair of co-axial cylinders. A cylinder with the point C_c on its centerline, the axis direction A_c and the radius r_c is the set of points X with

$$\left((C_c - X) \times A_c\right)^2 = r_c^2 \tag{A.8}$$

Plugging eq. A.1 into eq. A.8 and simplifying, we obtain another quadratic equation

$$a\lambda^2 + b\lambda + c = 0 \tag{A.9}$$

with $a = \left(T_m \times A_c\right)^2$, $b = 2\left(\left(O_m - C_c\right) \times A_c\right) \cdot \left(T_m \times A_c\right)$ and $c = \left(\left(O_m - C_c\right) \times A_c\right)^2 - r_c^2 A_c^2$. Again ignoring degenerate cases, the intersection point of the ray with the cylinder is

$$O_r = O_m + \frac{-b + \sqrt{b^2 - 4ac}}{a} T_m \tag{A.10}$$

The normal N_r in the intersection point is

$$N_r = \frac{O_r - C_r}{\|O_r - C_r\|} \tag{A.11}$$

where C_r is the closest point to O_r on the cylinder center line, $C_r = C_c + \frac{(O_r - C_c)A_c}{A_c^2} A_c$.

Applying Snell's law with the refraction indices n_1 and n_2 we get the refracted ray direction T_r from the incident ray direction T_m as

$$T_r = \frac{n_1}{n_2} T_m - \left(\frac{n_1}{n_2} cos\gamma_i + \sqrt{1 - sin^2\gamma_t}\right) N_r \tag{A.12}$$

where $sin^2\gamma_t = \frac{n_1}{n_2}\left(1 - cos^2\gamma_i\right)$ and γ_i is the angle of the incident ray with the surface normal. A second refraction is computed with the outer cylinder of the glass tube to obtain the final ray of view in the augmented camera model.

The reverse problem of finding the image coordinates for a given point P in space, taking into account refraction and reflection, is more complex. It can be solved by optimizing the image coordinates with respect to the object space error.

Light cones

A acute cone with the axis A, the vertex V and the angle $\theta < \frac{\pi}{2}$ is the set of all points X with

$$A \cdot \left(\frac{X - V}{\|X - V\|}\right) = cos\theta \tag{A.13}$$

The distance of a point P to the cone is

$$d = \|P - V\| \cdot sin\left(min(\delta, \frac{\pi}{2})\right)$$

with $\delta = acos\left(\frac{P-V}{\|P-V\|} \cdot A\right) - \theta$. Eq. A.13 can also be written as

$$(X-V)^T \mathbf{M} (X-V) = 0 \tag{A.14}$$

with

$$\mathbf{M} = AA^T - \mathbf{1} \cdot cos^2\theta \tag{A.15}$$

and

$$A \cdot (X-V) > 0 \tag{A.16}$$

Plugging eq. A.1 into eq. A.14 and simplifying, we obtain a quadratic equation

$$a\lambda^2 + b\lambda + c = 0 \tag{A.17}$$

where $a = T^T\mathbf{M}T$, $b = T^T\mathbf{M}(O-V)$ and $c = (O-V)^T\mathbf{M}(O-V)$. Excluding all degenerate cases, the sought after intersection point is

$$X_c = O + \frac{-b - \sqrt{b^2 - ac}}{a}T \tag{A.18}$$

This relation is used in the calibration of the light cones as well as in the calculation of 3D data from the calibrated sensor.

A.4 Algorithm Parameters

The proposed decoding algorithm has a number of parameters. In the interest of usability it is of course desirable to have as few as possible. The following tables list the parameters. Typically only the values in the first table need to be changed. Out of them, the single most important parameter is the minimum edge symbol probability. All others can be treated as constant for a given application.

Parameter	Default Value	Useful Range	Comment
Color Enhancement Iterations	1	0 to 4	Number of iterations for Belief Propagation
Edge Symbol Error Function Slope	1.5	0.5 to 4	Controls the error function used for edge symbol assignment
Minimum Edge Symbol Probability	0.6	0.1 to 1	Controls how carefully the decoding is performed
Edge Symbol Probability Credit Limit	0.2	0 to 0.5	Single Edges may fall below the minimum probability by this amount
Gradient Ratio Weighting Factor	0.5	0.1 to 1	Weighting of gradient ratio relative to symbol probability in position score computation

Table A.2: Variable parameters of the decoding algorithm.

Parameter	Default Value	Useful Range	Comment
Low Resolution Segmentation	1	0; 1	Speedup for decoding as the number of regions is reduced
Local Channel Equalization	1	0; 1	Improvements on colored surfaces
Fixed Luminance	0	0; 1	Improvements for small-scale specularities
Crosstalk Correction	1	0; 1	Apply inverse color-mixing matrix to measured region colors. Needs calibration.
Use Mask	0	0;1	Evaluate input image only where mask is non-zero
Use Green Channel Gradient Only	0	0;1	Workaround for chromatic aberration
Polar Transform Center	(0.5,0.5)	0 to 1	Center point of the cartesian image as fractions of width and height
Polar Transform Range Radial	(0.1,0.4)	0 to 0.5	Minimum and maximum radius as a fraction of the cartesian diagonal length
Polar Transform Phi Offset	π	0 to 2π	Controls the cut location for the polar transform

Table A.3: Quasi-constant parameters for image processing

Parameter	Default Value	Useful Range	Comment
Color Enhancement Sigmoid Width	0.6	0.5 to 1.5	Controls the probability function used for color enhancement
Color Enhancement Sigmoid Steepness	6	4 to 10	Controls the probability function used for color enhancement
Color Enhancement Linear Slope	0.5	0.25 to 1	Controls the probability function used for color enhancement
Maximum Angle Deviation	60	30 to 90	Edges should have approximately the expected direction
Maximum Gradient Ratio	1	0.1 to 10	Secondary Gradient should small relative to Primary Gradient
Maximum Number of Identified Sequences	15	1 to 100	Speedup for decoding. Stop after the first few good components
Minimum Edge Sequence Length	5	1 to 10	Additional safety. Basic uniqueness is determined from the pattern.
Ballast Fading Factor	0.9	0 to 1	Exponential fading of ballast picked up during identification
Minimum Component Size	23	1 to 1000	Ignore small connected components as they are unreliable
Edge Fit Interval Half Size	3	2 to 5	Search interval for subpixel edge localization
Tracing Minimum Edge Length	3	1 to 100	Minimum number of consecutive edge pixels required
Tracing Maximum Gap Size	2	1 to 10	Maximum number of missing edge pixels allowed
Tracing Max Lateral Displacement	2	1 to 3	Larger lateral displacement is considered as a gap
Edge Smoothing Method	0	0;1;2	0 is none, 1 is median of seven, 2 is c-spline
C-Spline Tau	0.2	0 to 1	Lower means more smoothing

Table A.4: Quasi-constant parameters of the decoding algorithm.

Bibliography

[Alba 09] A. Albarelli, E. Rodolà, and A. Torsello. "Robust Camera Calibration using Inaccurate Targets". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 2, pp. 376–383, 2009.

[Anne 97] F. Annexstein. "Generating De Bruijn Sequences: An Efficient Implementation". *Computers, IEEE Transactions on*, Vol. 46, No. 2, pp. 198–200, 1997.

[Armb 98] K. Armbruster and M. Scheffler. "Messendes 3D-Endoskop". *Horizonte*, Vol. 12, p. 15–16, 1998.

[Arno 10] M. Arnold, A. Ghosh, S. Ameling, and G. Lacey. "Automatic Segmentation and Inpainting of Specular Highlights for Endoscopic Imaging". *EURASIP Journal on Image and Video Processing*, Vol. 2010, pp. 1–12, 2010.

[Atki 06] G. A. Atkinson and E. R. Hancock. "Recovery of surface orientation from diffuse polarization". *IEEE Transactions on Image Processing*, Vol. 15, No. 6, pp. 1653–1664, 2006.

[Bars 03] S. Barsky and M. Petrou. "The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1239–1252, 2003.

[Basr 07] R. Basri, D. Jacobs, and I. Kemelmacher. "Photometric stereo with general, unknown lighting". *International Journal of Computer Vision*, Vol. 72, No. 3, pp. 239–257, 2007.

[Bay 06] H. Bay, T. Tuytelaars, and L. V. Gool. "Surf: Speeded up robust features". *European Conference on Computer Vision, IEEE Conference on*, pp. 404–417, 2006.

[Belh 99] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille. "The bas-relief ambiguity". *International Journal of Computer Vision*, Vol. 35, No. 1, pp. 33–44, 1999.

[Besl 88] P. J. Besl. "Active, optical range imaging sensors". *Machine vision and applications*, Vol. 1, No. 2, pp. 127–152, 1988.

[Biga 06] M. Bigas, E. Cabruja, J. Forest, and J. Salvi. "Review of CMOS image sensors". *Microelectronics Journal*, Vol. 37, No. 5, pp. 433–451, 2006.

[Biou 07] J. M. Bioucas-Dias and G. Valadao. "Phase unwrapping via graph cuts". *IEEE Transactions on Image Processing*, Vol. 16, No. 3, pp. 698–709, 2007.

[Blai 04] F. Blais. "Review of 20 years of range sensor development". *Journal of Electronic Imaging*, Vol. 13, No. 1, pp. 231–243, 2004.

[Blai 86] F. Blais and M. Rioux. "Real-time numerical peak detector". *Signal Processing*, Vol. 11, No. 2, pp. 145–155, Sep. 1986.

[Blea 00] A. Bleau and L. J. Leon. "Watershed-based segmentation and region merging". *Computer Vision and Image Understanding*, Vol. 77, No. 3, pp. 317–370, 2000.

[Blos 87] S. D. Blostein and T. S. Huang. "Error analysis in stereo determination of 3-D point positions". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, No. 6, p. 752–765, 1987.

[Boug 08] J. Y. Bouguet. "Camera calibration toolbox for Matlab". 2008.

[Boye 87] K. L. Boyer and A. C. Kak. "Color-encoded structured light for rapid active ranging". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, No. 1, 1987.

[Boyk 04] Y. Boykov and V. Kolmogorov. "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 9, pp. 1124–1137, Sep. 2004.

[Brad 08] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library.* O'Reilly Media, 1st Ed., Sep. 2008.

[Brin 08] W. Brink, A. Robinson, and M. Rodrigues. "Indexing uncoded stripe patterns in structured light systems by maximum spanning trees". In: *British Machine Vision Conference (BMVC), Proceedings of*, p. 575–584, 2008.

[Cann 87] J. Canny. "A computational approach to edge detection". *Readings in computer vision: issues, problems, principles, and paradigms*, Vol. 184, 1987.

[Carr 85] B. Carrihill and R. A. Hummel. "Experiments with the Intensity Ratio Data Sensor". *Computer Vision, Graphics, and Image Processing*, Vol. 32, No. 3, pp. 337–358, Dec. 1985.

[Case 97] V. Caselles, R. Kimmel, and G. Sapiro. "Geodesic active contours". *International Journal of Computer Vision*, Vol. 22, No. 1, pp. 61–79, 1997.

[Casp 98] D. Caspi, N. Kiryati, and J. Shamir. "Range imaging with adaptive color structured light". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 5, pp. 470–480, May 1998.

[Chen 05] D. Chen and G. Zhang. "A new sub-pixel detector for X-corners in camera calibration targets". In: *Computer Graphics, Visualization and Computer Vision, International Conference in Central Europe on*, pp. 97–100, 2005.

[Chen 08] S. Y. Chen, Y. F. Li, and J. Zhang. "Vision processing for realtime 3-D data acquisition based on coded structured light". *IEEE Transactions on Image Processing*, Vol. 17, No. 2, pp. 167–176, 2008.

[Clan 11] N. T. Clancy, D. Stoyanov, L. Maier-Hein, A. Groch, G.-Z. Yang, and D. S. Elson. "Spectrally encoded fiber-based structured lighting probe for intraoperative 3D imaging". *Biomed. Opt. Express*, Vol. 2, No. 11, pp. 3119–3128, Nov 2011.

[Clau 05] D. Claus and A. W. Fitzgibbon. "A rational function lens distortion model for general cameras". 2005.

[Coma 02] D. Comaniciu and P. Meer. "Mean shift: A robust approach toward feature space analysis". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, pp. 603–619, 2002.

[Coup 10] C. Couprie, L. Grady, L. Najman, and H. Talbot. "Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest". In: *International Conference on Computer Vision, IEEE Conference on*, pp. 731–738, 2010.

[Crea 86] K. Creath. "Comparison of phase-measurement algorithms". In: *Proceedings of the SPIE*, p. 19, 1986.

[D 10] S. D., S. M., P. P., and Y. G.Z. "Real-Time Stereo Reconstruction in Robotically Assisted Minimally Invasive Surgery". In: *MICCAI*, pp. 275–282, 2010.

[Dain 84] J. C. Dainty. *Laser Speckle and Related Phenomena.* Springer, 2 enl sub Ed., Oct. 1984.

[Davi 05] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz. "Spacetime stereo: a unifying framework for depth from triangulation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 2, pp. 296–302, Feb. 2005.

[Dene 90] J. Denes and A. D. Keedwell. "A new construction of two-dimensional arrays with the window property". *Information Theory, IEEE Transactions on*, Vol. 36, No. 4, pp. 873–876, July 1990.

[Deve 01] F. Devernay and O. Faugeras. "Straight lines have to be straight". *Machine vision and applications*, Vol. 13, No. 1, p. 14–24, 2001.

[Deve 02] F. Devernay, O. Bantiche, and È. Coste Manière. "Structured light on dynamic scenes using standard stereoscopy algorithms". 0 RR-4477, INRIA, 06 2002.

[Doux 08] D. Douxchamps and K. Chihara. "High-accuracy and robust localization of large control markers for geometric camera calibration". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 2, pp. 376–383, 2008.

[Dres 92] T. Dresel, G. Häusler, and H. Venzke. "Three-dimensional sensing of rough surfaces by coherence radar". *Applied Optics*, Vol. 31, No. 7, pp. 919–925, 1992.

[Drew 00] M. S. Drew and M. H. Brill. "Color from shape from color: a simple formalism with known light sources". *Journal of the Optical Society of America A*, Vol. 17, No. 8, pp. 1371–1381, 2000.

[Duar 08] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk. "Single-pixel imaging via compressive sampling". *Signal Processing Magazine, IEEE*, Vol. 25, No. 2, pp. 83–91, 2008.

[Dubo 05] E. Dubois. "Frequency-domain methods for demosaicking of Bayer-sampled color images". *Signal Processing Letters, IEEE*, Vol. 12, No. 12, pp. 847–850, 2005.

[Dunn 07] A. Dunne, J. Mallon, and P. Whelan. "A comparison of new generic camera calibration with the standard parametric approach". In: *Proceedings of the IAPR Conference on Machine Vision Applications, Tokyo, Japan*, p. 114–117, 2007.

[Durr 95] A. F. Durrani and G. M. Preminger. "Three-dimensional video imaging for endoscopic surgery". *Computers in biology and medicine*, Vol. 25, No. 2, p. 237–247, 1995.

[e2v 03] e2v technologies. "UV conversion coatings". 2003.

[Etzi 88] T. Etzion. "Constructions for perfect maps and pseudorandom arrays". *Information Theory, IEEE Transactions on*, Vol. 34, No. 5, pp. 1308–1316, Sep. 1988.

[Faug 01] O. Faugeras, Q. Luong, and T. Papadopoulo. *The geometry of multiple images*. Vol. 2, MIT press, 2001.

[Fava 05] P. Favaro and S. Soatto. "A geometric approach to shape from defocus". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 406–417, 2005.

[Felz 04a] P. F. Felzenszwalb and D. R. Huttenlocher. "Efficient belief propagation for early vision". In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. I–261–I–268, July 2004.

[Felz 04b] P. Felzenszwalb and D. Huttenlocher. "Efficient Graph-Based Image Segmentation". *International Journal of Computer Vision*, Vol. 59, No. 2, pp. 167–181, Sep. 2004.

[Fial 10] M. Fiala and C. Shu. "Fully automatic camera calibration using self-identifying calibration targets". Tech. Rep., NRC Institute for Information Technology; National Research Council Canada, 2010.

[Figu 02] M. A. F. Figueiredo and A. K. Jain. "Unsupervised learning of finite mixture models". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 3, pp. 381–396, 2002.

[Fish 96] R. B. Fisher and D. K. Naidu. "A comparison of algorithms for subpixel peak detection". *Image Technology, Advances in Image Processing, Multimedia and Machine Vision*, pp. 385–404, 1996.

[Fitz 03] A. W. Fitzgibbon. "Robust registration of 2D and 3D point sets". *Image and Vision Computing*, Vol. 21, No. 13-14, p. 1145–1153, 2003.

[Fors 05] F. Forster. *Real-Time Range Imaging for Human-Machine Interfaces*. PhD thesis, Technische Universität München, 2005.

[Fors 06] F. Forster. "A High-Resolution and High Accuracy Real-Time 3D Sensor Based on Structured Light". In: *3D Data Processing Visualization and Transmission, International Symposium on*, pp. 208–215, IEEE Computer Society, Los Alamitos, CA, USA, 2006.

[Garc 08] J. García, Z. Zalevsky, P. García-Martínez, C. Ferreira, M. Teicher, and Y. Beiderman. "Three-dimensional mapping and range measurement by means of projected speckle patterns". *Applied Optics*, Vol. 47, No. 16, pp. 3032–3040, 2008.

[Gass 03] J. Gass, A. Dakoff, and M. K. Kim. "Phase imaging without 2π ambiguity by multiwavelength digital holography". *Optics letters*, Vol. 28, No. 13, pp. 1141–1143, 2003.

[Geng 96] Z. J. Geng. "Rainbow three-dimensional camera: new concept of high-speed three-dimensional vision systems". *Optical Engineering*, Vol. 35, No. 2, pp. 376–383, Feb. 1996.

[Geor 03] A. S. Georghiades. "Incorporating the Torrance and Sparrow Model of Reflectance in Uncalibrated Photometric Stereo". In: *International Conference on Computer Vision, IEEE Conference on*, pp. 816–, IEEE Computer Society, Washington, DC, USA, 2003.

[Glas 89] A. S. Glassner. *An Introduction to Ray Tracing*. Academic Press Inc., June 1989.

[Gluc 01] J. Gluckman and S. K. Nayar. "Rectifying transformations that minimize resampling effects". In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2001.

[Gock 06] T. Gockel. *Interaktive 3D-Modellerfassung mittels One-Shot-Musterprojektion und Schneller Registrierung*. Universitäts-Verlag Karlsruhe, 2006.

[Gokt 05] S. B. Gokturk, H. Yalcin, and C. Bamji. "A time-of-flight depth sensor-system description, issues and solutions". In: *Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Conference on*, p. 35, 2005.

[Gold 88] R. M. Goldstein, H. A. Zebker, and C. L. Werner. "Satellite radar interferometry: two-dimensional phase unwrapping". *Radio Science*, Vol. 23, No. 4, pp. 713–720, 1988.

[Gong 07] M. Gong, R. Yang, L. Wang, and M. Gong. "A performance study on different cost aggregation approaches used in real-time stereo matching". *International Journal of Computer Vision*, Vol. 75, No. 2, pp. 283–296, 2007.

[Gorr 90] P. A. Gorry. "General least-squares smoothing and differentiation by the convolution (Savitzky-Golay) method". *Analytical Chemistry*, Vol. 62, No. 6, pp. 570–573, 1990.

[Gort 10] S. S. Gorthi and P. Rastogi. "Fringe projection techniques: Whither we are?". *Optics and Lasers in Engineering*, Vol. 48, No. 2, pp. 133–140, 2010.

[Grad 06] L. Grady. "Random walks for image segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1768–1783, 2006.

[Gras 09] O. G. Grasa, J. Civera, A. Guemes, V. Muoz, and J. M. M. Montiel. "Ekf monocular slam 3d modeling, measuring and augmented reality from endoscope image sequences". In: *5th Workshop on Augmented Environments for Medical Imaging including Augmented Reality in Computer-Aided Surgery, held in conjunction with MICCAI2009*, 2009.

[Grif 92] P. M. Griffin, L. S. Narasimhan, and S. R. Yee. "Generation of uniquely encoded light patterns for range data acquisition". *Pattern Recognition*, Vol. 25, No. 6, pp. 609–616, June 1992.

[Gros 01] M. D. Grossberg and S. K. Nayar. "A General Imaging Model and a Method for Finding its Parameters". *International Conference on Computer Vision, IEEE Conference on*, Vol. 2, p. 108, 2001.

[Gros 05] M. D. Grossberg and S. K. Nayar. "The raxel imaging model and ray-based calibration". *International Journal of Computer Vision*, Vol. 61, No. 2, p. 119–137, 2005.

[Guan 03] C. Guan, L. Hassebrook, and D. Lau. "Composite structured light pattern for three-dimensional video". *Optics Express*, Vol. 11, No. 5, pp. 406–417, 2003.

[Guan 09] S. Guan, R. Klette, and Y. Woo. "Belief propagation for stereo analysis of night-vision sequences". *Advances in Image and Video Technology*, pp. 932–943, 2009.

[Hall 01] O. Hall-Holt and S. Rusinkiewicz. "Stripe Boundary Codes for Real-Time Structured-Light Range Scanning of Moving Objects". *International Conference on Computer Vision, IEEE Conference on*, Vol. 2, p. 359, 2001.

[Hao 10] P. Hao, Y. Li, Z. Lin, and E. Dubois. "A geometric method for optimal design of color filter arrays". *IEEE Transactions on Image Processing*, 2010.

[Hara 84] R. M. Haralick. "Digital step edges from zero crossing of second directional derivatives". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, No. 1, pp. 58–68, 1984.

[Hari 06] P. Hariharan. *Basics of Interferometry, Second Edition.* Academic Press, 2 Ed., Oct. 2006.

[Hart 03] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision.* Cambridge University Press, 2003.

[Hart 07] R. Hartley and S. B. Kang. "Parameter-free radial distortion correction with center of distortion estimation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1309–1321, 2007.

[Hasi 09] S. Hasinoff and K. Kutulakos. "Confocal Stereo". *International Journal of Computer Vision*, Vol. 81, pp. 82–104, 2009. 10.1007/s11263-008-0164-2.

[Haus 11] G. Häusler and S. Ettl. "Limitations of Optical 3D Sensors". In: R. Leach, Ed., *Optical Measurement of Surface Topography*, pp. 23–48, 2011.

[Haus 88] G. Häusler and W. Heckel. "Light sectioning with large depth and high resolution". *Applied Optics*, Vol. 27, No. 24, pp. 5165–5169, Dec. 1988.

[Haus 93] G. Häusler and D. Ritter. "Parallel three-dimensional sensing by color-coded triangulation". *Applied Optics*, Vol. 32, No. 35, pp. 7164–7169, 1993.

[Haus 97] G. Häusler and G. Leuchs. "Physikalische Grenzen der optischen Formerfassung mit Licht". *Physikalische Blätter*, Vol. 53, No. 5, pp. 417–422, 1997.

[Hebe 92] M. Hebert and E. Krotkov. "3D measurements from imaging laser radars: how good are they?". *Image and Vision Computing*, Vol. 10, No. 3, pp. 170–178, 1992.

[Heik 00] J. Heikkilä. "Geometric Camera Calibration Using Circular Control Points". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, pp. 1066–1077, Oct. 2000.

[Heik 97] J. Heikkilä and O. Silven. "A Four-step Camera Calibration Procedure with Implicit Image Correction". In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 1106–, IEEE Computer Society, Washington, DC, USA, 1997.

[Hema 03] E. E. Hemayed. "A Survey of Camera Self-Calibration". *Advanced Video and Signal Based Surveillance, IEEE Conference on*, Vol. 0, p. 351, 2003.

[Hern 10] D. Hernandez-Stewart. *Digital ear scanner: measuring the compliance of the ear*. PhD thesis, Massachusetts Institute of Technology, 2010.

[Hibi 95] K. Hibino, B. F. Oreb, D. I. Farrant, and K. G. Larkin. "Phase shifting for nonsinusoidal waveforms with phase-shift errors". *Journal of the Optical Society of America A*, Vol. 12, No. 4, pp. 761–768, Apr. 1995.

[Hidr 01] C. H. Hidrovo and D. P. Hart. "Emission reabsorption laser induced fluorescence (ERLIF) film thickness measurement". *Measurement Science and Technology*, Vol. 12, p. 467, 2001.

[Hirs 07] H. Hirschmüller and D. Scharstein. "Evaluation of cost functions for stereo matching". In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 1–8, 2007.

[Hoff 08] G. Hoffmann. "Cie color space". Tech. Rep., FHO Emden, 2008.

[Horn 97] E. Horn and N. Kiryati. "Toward optimal structured light patterns". In: *Recent Advances in 3-D Digital Imaging and Modeling, International Conference on*, pp. 28–35, May 1997.

[Hu 10] M. Hu, G. Penney, M. Figl, P. Edwards, F. Bello, R. Casula, D. Rueckert, and D. Hawkes. "Reconstruction of a 3D surface from video that is robust to missing data and outliers: Application to minimally invasive surgery using stereo and mono endoscopes". *Medical Image Analysis*, Dec. 2010. PMID: 21195656.

[Hu 89] G. Hu and G. Stockman. "3-D Surface Solution Using Structured Light and Constraint Propagation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 4, pp. 390–402, 1989.

[Huan 03] P. S. Huang, C. Zhang, and F. Chiang. "High-speed 3-D shape measurement based on digital fringe projection". *Optical Engineering*, Vol. 42, No. 1, pp. 163–168, Jan. 2003.

[Hube 04] P. M. Hubel, J. Liu, and R. J. Guttosch. "Spatial frequency response of color image sensors: Bayer color filters and Foveon X3". In: M. M. Blouke, N. Sampat, & R. J. Motta, Ed., *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pp. 402–407, June 2004.

[Hugl 89] H. Hugli and G. Maitre. "Generation and use of color pseudorandom sequences for coding structured light in active ranging". In: *Proceedings of the SPIE*, p. 75, 1989.

[Ihrk 10] I. Ihrke, G. Wetzstein, and W. Heidrich. "A theory of plenoptic multiplexing". In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 483–490, 2010.

[Jarv 83] R. Jarvis. "A perspective on range finding techniques for computer vision". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 5, pp. 122–139, 1983.

[Kaka 92] R. Kakarala and A. O. Hero. "On achievable accuracy in edge localization". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 7, pp. 777–781, 1992.

[Kana 08] K. Kanatani, Y. Sugaya, and H. Niitsuma. "Triangulation from two views revisited: Hartley-Sturm vs. optimal correction". In: *British Machine Vision Conference (BMVC), Proceedings of*, p. 173–182, 2008.

[Kann 06] J. Kannala and S. S. Brandt. "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1335–1340, 2006.

[Kanu 02] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. "An efficient k-means clustering algorithm: Analysis and implementation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, pp. 881–892, 2002.

[Kawa 08] H. Kawasaki, R. Furukawa, R. Sagawa, and Y. Yagi. "Dynamic scene shape reconstruction using a single structured light pattern". In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 1–8, 2008.

[Kazh 06] M. Kazhdan, M. Bolitho, and H. Hoppe. "Poisson surface reconstruction". In: *Proceedings of the fourth Eurographics symposium on Geometry processing*, p. 61–70, 2006.

[Knau 04] M. C. Knauer, J. Kaminski, and G. Häusler. "Phase measuring deflectometry: a new approach to measure specular free- form surfaces". In: *Proceedings of the SPIE*, pp. 366–376, 2004.

[Kolb 08] A. Kolb, E. Barth, and R. Koch. "ToF-sensors: New dimensions for realism and interactivity". In: *Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Conference on*, pp. 1–6, 2008.

[Kole 03] E. Kolenovic, W. Osten, R. Klattenhoff, S. Lai, C. von Kopylow, and W. Jüptner. "Miniaturized digital holography sensor for distal three-dimensional endoscopy". *Applied Optics*, Vol. 42, No. 25, p. 5167–5172, 2003.

[Koni 05a] T. P. Koninckx, P. Peers, P. Dutre, and L. V. Gool. "Scene-adapted structured light". In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 611–618, June 2005.

[Koni 05b] T. Koninckx. *Adaptive Structured Light*. PhD thesis, KU Leuven, 2005.

[Kons 09] G. Konstantatos. *Sensitive Solution-processed Quantum Dot Photodetectors*. PhD thesis, University of Toronto, 2009.

[Kopl 94] J. Koplowitz and V. Greco. "On the edge location error for local maximum and zero-crossing edge detectors". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1207–1212, 1994.

[Kutu 00] K. N. Kutulakos and S. M. Seitz. "A theory of shape by space carving". *International Journal of Computer Vision*, Vol. 38, No. 3, pp. 199–218, 2000.

[Labo 01] X. Laboureux and G. Häusler. "Localization and Registration of Three-Dimensional Objects in Space-Where are the Limits?". *Appl. Opt.*, Vol. 40, No. 29, pp. 5206–5216, Oct 2001.

[Lee 05] H. C. Lee. *Introduction to color imaging science*. Cambridge University Press, 2005.

[Lei 06] C. Lei, J. Selzer, and Y. H. Yang. "Region-tree based stereo using dynamic programming optimization". In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 2378–2385, 2006.

[Lei 10] S. Lei and S. Zhang. "Digital sinusoidal fringe pattern generation: Defocusing binary patterns VS focusing sinusoidal patterns". *Optics and Lasers in Engineering*, Vol. 48, No. 5, pp. 561–569, 2010.

[Levi 07] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. "Image and depth from a conventional camera with a coded aperture". *ACM Trans. Graph.*, Vol. 26, July 2007.

[Levi 09] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi. "Turbopixels: Fast superpixels using geometric flows". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 12, pp. 2290–2297, 2009.

[Li 08a] X. Li and X. G. Xia. "A fast robust Chinese remainder theorem based phase unwrapping algorithm". *Signal Processing Letters, IEEE*, Vol. 15, pp. 665–668, 2008.

[Li 08b] X. Li, B. Gunturk, and L. Zhang. "Image demosaicing: a systematic survey". In: *Proceedings of the SPIE*, pp. 68221J–68221J–15, San Jose, CA, USA, 2008.

[Lin 06] Y. Lin, Y. Tsai, Y. Hung, and Z. Shih. "Comparison between immersion-based and toboggan-based watershed image segmentation". *IEEE Transactions on Image Processing*, Vol. 15, No. 3, pp. 632–640, March 2006.

[Lin 08] S. Lin, L. Quan, and H. Y. Shum. "Highlight removal by illumination-constrained inpainting". In: *International Conference on Computer Vision, IEEE Conference on*, pp. 164–169, 2008.

[Lind 08] M. Lindner, A. Kolb, and T. Ringbeck. "New insights into the calibration of ToF-sensors". In: *Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Conference on*, pp. 1–5, 2008.

[Lind 90] T. Lindeberg. "Scale-Space for Discrete Signals". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 3, pp. 234–254, 1990.

[Liu 10] K. Liu, Y. Wang, D. L. Lau, Q. Hao, and L. G. Hassebrook. "Gamma model and its analysis for phase measuring profilometry". *Journal of the Optical Society of America A*, Vol. 27, No. 3, pp. 553–562, 2010.

[Lowe 99] D. G. Lowe. "Object recognition from local scale-invariant features". In: *International Conference on Computer Vision, IEEE Conference on*, p. 1150, 1999.

[Lucc 01] L. Lucchese and S. K. Mitra. "Color Image Segmentation: A State-of-the-Art Survey". In: *Proceedings of the Indian National Science Academy (INSA-A)*, pp. 207–221, March 2001.

[Lucc 02] L. Lucchese and S. K. Mitra. "Using saddle points for subpixel feature detection in camera calibration targets". In: *Circuits and Systems (APCCAS), Asia-Pacific Conference on*, p. 191–195, 2002.

[MacK 03] D. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003.

[Mall 07] J. Mallon and P. F. Whelan. "Which pattern? Biasing aspects of planar calibration patterns and detection methods". *Pattern Recognition Letters*, Vol. 28, pp. 921–930, June 2007.

[Marr 80] D. Marr and E. Hildreth. "Theory of edge detection". *Proceedings of the Royal Society of London. Series B, Biological Sciences*, pp. 187–217, 1980.

[Maru 93] M. Maruyama and S. Abe. "Range sensing by projecting multiple slits with random cuts". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 6, pp. 647–651, June 1993.

[Mata 04] J. Matas, O. Chum, M. Urban, and T. Pajdla. "Robust wide-baseline stereo from maximally stable extremal regions". *Image and Vision Computing*, Vol. 22, No. 10, pp. 761–767, 2004.

[McGu 08] M. McGuire. "Efficient, high-quality bayer demosaic filtering on gpus". *Journal of Graphics, GPU, & Game Tools*, Vol. 13, No. 4, pp. 1–16, 2008.

[Mehn 97] A. Mehnert and P. Jackway. "An improved seeded region growing algorithm". *Pattern Recognition Letters*, Vol. 18, No. 10, pp. 1065–1071, Oct. 1997.

[Meng 02] P. Mengel, G. Doemens, and L. Listl. "Fast range imaging by CMOS sensor array through multiple double short time integration (MDSI)". In: *Image Processing, Proceedings of the International Conference on*, pp. 169–172, 2002.

[Meye 04] F. Meyer. "Levelings, image simplification filters for segmentation". *Journal of Mathematical Imaging and Vision*, Vol. 20, pp. 59–72, 2004.

[Miku 08] P. Mikulastik, R. Höver, and O. Urfalioglu. "Error analysis of subpixel edge localisation". *Signal Processing for Image Enhancement and Multimedia Processing*, pp. 103–113, 2008.

[Miro 11] D. J. Mirota, M. Ishii, and G. D. Hager. "Vision-Based Navigation in Image-Guided Interventions". *Annual Review of Biomedical Engineering*, Vol. 13, No. 1, pp. 297–319, 2011.

[Mitc 95] C. J. Mitchell. "Aperiodic and semi-periodic perfect maps". *Information Theory, IEEE Transactions on*, Vol. 41, No. 1, pp. 88–95, Jan. 1995.

[Mitc 96] C. J. Mitchell, T. Etzion, and K. G. Paterson. "A method for constructing decodable de Bruijn sequences". *Information Theory, IEEE Transactions on*, Vol. 42, No. 5, pp. 1472–1478, Sep. 1996.

[Mohr 94] P. Mohr and P. Brand. "Accuracy in Image Measure". *Accuracy in image measure, Proceedings of the SPIE*, Vol. 2350, pp. 218–228, 1994.

[Monk 93] T. Monks and J. Carter. "Improved stripe matching for colour encoded structured light". In: *Computer Analysis of Images and Patterns*, pp. 476–485, 1993.

[Mora 98] R. A. Morano, C. Ozturk, R. Conn, S. Dubin, S. Zietz, and J. Nissano. "Structured light using pseudorandom codes". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 3, pp. 322–327, March 1998.

[More 78] J. More. "The Levenberg-Marquardt algorithm: implementation and theory". *Numerical Analysis*, pp. 105–116, 1978.

[Mori 05] G. Mori. "Guiding model search using segmentation". In: *International Conference on Computer Vision, IEEE Conference on*, pp. 1417–1423, 2005.

[Mori 88] H. Morita, K. Yajima, and S. Sakata. "Reconstruction of Surfaces of 3-D Objects by M-array Pattern Projection Method". In: *International Conference on Computer Vision, IEEE Conference on*, p. 468, 1988.

[Moun 10] P. Mountney, D. Stoyanov, and Y. G.Z. "Three-Dimensional Tissue Deformation Recovery and Tracking". *Signal Processing Magazine, IEEE*, Vol. 27, No. 4, pp. 14 –24, july 2010.

[Mugn 95] L. M. Mugnier. "Conoscopic holography: toward three-dimensional reconstructions of opaque objects". *Applied Optics*, Vol. 34, No. 8, p. 1363–1371, 1995.

[Mure 05] D. Muresan and T. Parks. "Demosaicing Using Optimal Recovery". *IEEE Transactions on Image Processing*, Vol. 14, No. 2, pp. 267–278, Feb. 2005.

[Naya 96] S. K. Nayar, M. Watanabe, and M. Noguchi. "Real-Time Focus Range Sensor". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 12, pp. 1186–1198, 1996.

[Neha 05] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi. "Efficiently Combining Positions and Normals for Precise 3D Geometry". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, Vol. 24, No. 3, Aug. 2005.

[Newc 10] R. A. Newcombe and A. J. Davison. "Live dense reconstruction with a single moving camera". In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 1498–1505, 2010.

[Nico 00] G. Nico, G. Palubinskas, and M. Datcu. "Bayesian approaches to phase unwrapping: theoretical study". *Signal Processing, IEEE Transactions on*, Vol. 48, No. 9, pp. 2545–2556, 2000.

[Nist 05] D. Nistér. "Preemptive RANSAC for live structure and motion estimation". *Machine vision and applications*, Vol. 16, No. 5, pp. 321–329, 2005.

[Nobl 06] J. A. Noble and D. Boukerroui. "Ultrasound image segmentation: A survey". *Medical Imaging, IEEE Transactions on*, Vol. 25, No. 8, pp. 987–1010, 2006.

[O 11] G. O., C. J., and M. J. "EKF Monocular SLAM with Relocalization for Laparoscopic Sequences". In: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 4816 – 4821, 2011.

[Osko 10] M. A. Oskoei and H. Hu. "A Survey on Edge Detection Methods". Tech. Rep., University of Essex, 2010.

[Otsu 75] N. Otsu. "A threshold selection method from gray-level histograms". *Automatica*, Vol. 11, pp. 285–296, 1975.

[Pal 93] N. R. Pal and S. K. Pal. "A review on image segmentation techniques". *Pattern Recognition*, Vol. 26, No. 9, pp. 1277–1294, Sep. 1993.

[Pan 09] B. Pan, Q. Kemao, L. Huang, and A. Asundi. "Phase error analysis and compensation for nonsinusoidal waveforms in phase-shifting digital fringe projection profilometry". *Optics Letters*, Vol. 34, No. 4, pp. 416–418, Feb. 2009.

[Park 11] I. K. Park, N. Singhal, M. H. Lee, S. Cho, and C. Kim. "Design and Performance Evaluation of Image Processing Algorithms on GPUs". *Parallel and Distributed Systems, IEEE Transactions on*, Vol. 22, No. 1, pp. 91–104, 2011.

[Pell 00] D. Pelleg and A. Moore. "X-means: Extending K-means with Effcient Estimation of the Number of Clusters". In: *Machine Learning, International Conference on*, p. 727, 2000.

[Penn 09] J. Penne, K. Höller, M. Stürmer, T. Schrauder, A. Schneider, R. Engelbrecht, H. Feussner, B. Schmauss, and J. Hornegger. "Time-of-flight 3-D endoscopy". *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, p. 467–474, 2009.

[Pere 98] P. Perez. "Markov random fields and images". *CWI Quarterly*, Vol. 11, No. 4, pp. 413–437, 1998.

[Pers 04] Persistence of Vision Pty. Ltd. "Persistence of Vision Raytracer (Version 3.6)". 2004.

[Pinh 97] A. J. Pinho and L. B. Almeida. "A review on edge detection based on filtering and differentiation". *Revista do Detua*, Vol. 2, No. 1, pp. 113–126, 1997.

[Pita 91] I. Pitas and P. Tsakalides. "Multivariate ordering in color image filtering". *Circuits and Systems for Video Technology, IEEE Transactions on*, Vol. 1, No. 3, pp. 247–259,295–6, Sep. 1991.

[Posd 82] J. L. Posdamer and M. D. Altschuler. "Surface measurement by space-encoded projected beam systems". *Computer graphics and image processing*, Vol. 18, No. 1, pp. 1–17, 1982.

[Pras 07] M. Prasciolu. *3D laser scanner based on surface silicon micromachining techniques for shape and size reconstruction of the human ear canal.* PhD thesis, Università degli studi di Trieste, 2007.

[Proe 96] M. Proesmans, L. J. V. Gool, and A. J. Oosterlinck. "Active acquisition of 3D shape for moving objects". In: *Image Processing, International Conference on*, pp. 647–650, 1996.

[Quan 10] C. Quan, W. Chen, and C. Tay. "Phase-retrieval techniques in fringe-projection profilometry". *Optics and Lasers in Engineering*, Vol. 48, No. 2, pp. 235–243, Feb. 2010.

[Radh 10] A. Radhakrishna, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. "SLIC Superpixels". Tech. Rep., Technical Report 149300, EPFL, 2010.

[Rama 05] S. Ramalingam, P. Sturm, and S. K. Lodha. "Theory and Experiments towards Complete Generic Calibration". Tech. Rep., INRIA - CNRS, 2005.

[Remo 06] F. Remondino and C. Fraser. "Digital camera calibration methods: considerations and comparisons". In: Isprs, Ed., *Photogrammetry, Remote Sensing and Spatial Information Sciences, International Archives of*, Dresden, Germany, 2006.

[Rock 99] P. Rockett. "The accuracy of sub-pixel localisation in the canny edge detector". In: *British Machine Vision Conference (BMVC), Proceedings of*, 1999.

[Roer 00] J. B. T. M. Roerdink and A. Meijster. "The watershed transform: definitions, algorithms and parallelization strategies". *Fundam. Inf.*, Vol. 41, No. 1-2, pp. 187–228, 2000.

[Ryoo 08] S. Ryoo, C. I. Rodrigues, S. S. Baghsorkhi, S. S. Stone, D. B. Kirk, and W. W. Hwu. "Optimization principles and application performance evaluation of a multithreaded GPU using CUDA". In: *Principles and practice of parallel programming (SIGPLAN), ACM Symposium on*, pp. 73–82, 2008.

[Saga 05] R. Sagawa, M. Takatsuji, T. Echigo, and Y. Yagi. "Calibration of lens distortion by structured-light scanning". In: *Intelligent Robots and Systems, IEEE International Conference on*, p. 832–837, 2005.

[Salv 10] J. Salvi, S. Fernandez, T. Pribanic, and X. Llado. "A State of the Art in Structured Light Patterns for Surface Profilometry". *Pattern Recognition*, Vol. 43, pp. 2666–2680, 2010.

[Sans 99] G. Sansoni, M. Carocci, and R. Rodella. "Three-Dimensional Vision Based on a Combination of Gray-Code and Phase-Shift Light Projection: Analysis and Compensation of the Systematic Errors". *Applied Optics*, Vol. 38, No. 31, pp. 6565–6573, 1999.

[Sato 86] K. Sato, H. Yamamoto, and S. Inokuchi. "Tuned range finder for high precision 3D data". In: *Pattern Recognition, International Conference on*, pp. 1168–1171, 1986.

[Sava 97] C. Savage. "A survey of combinatorial Gray codes". *SIAM Review*, Vol. 39, No. 4, p. 605=629, 1997.

[Savi 64] A. Savitzky and M. J. Golay. "Smoothing and differentiation of data by simplified least squares procedures.". *Analytical Chemistry*, Vol. 36, No. 8, pp. 1627–1639, 1964.

[Scha 00] H. Scharr. *Optimale Operatoren in der Digitalen Bildverarbeitung*. PhD thesis, Universität Heidelberg, 2000.

[Scha 02] D. Scharstein and R. Szeliski. "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms". *International Journal of Computer Vision*, Vol. 47, No. 1, pp. 7–42, 2002.

[Scha 03] D. Scharstein and R. Szeliski. "High-accuracy stereo depth maps using structured light". In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 195–202, 2003.

[Sche 00] Y. Y. Schechner and N. Kiryati. "Depth from defocus vs. stereo: How different really are they?". *International Journal of Computer Vision*, Vol. 39, No. 2, pp. 141–162, 2000.

[Schi 11] A. Schick, F. Forster, and M. Stockmann. "3D measuring in the field of endoscopy". In: *Proceedings of the SPIE*, p. 808216, 2011.

[Seth 99] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 2 Ed., June 1999. Published: Paperback.

[Shen 92] J. Shen and S. Castan. "An optimal linear operator for step edge detection". *Graphical Models and Image Processing (CVGIP)*, Vol. 54, No. 2, pp. 112–133, March 1992.

[Shi 97] J. Shi and J. Malik. "Normalized cuts and image segmentation". In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 731–737, June 1997.

[Shor 94] M. R. Shortis, T. A. Clarke, and T. Short. "Comparison of some techniques for the subpixel location of discrete target images". In: S. F. El-Hakim, Ed., *Proceedings of the SPIE*, pp. 239–250, Oct. 1994.

[Shpu 07] A. Shpunt and Z. Zalevsky. "Three-dimensional sensing using speckle patterns". March 2007. US Patent App. 12/282,517.

[Stoc 02] C. Stock, U. Mühlmann, M. K. Chandraker, and A. Pinz. "Subpixel corner detection for tracking applications using cmos camera technology". In: *26th Workshop of the AAPR*, p. 191–199, 2002.

[Stoe 00] S. L. Stoev. "Rafsi - a Fast Watershed Algorithm Based on Rainfalling Simulation". In: *Computer Graphics, Visualization, and Interactive Digital Media (WSCG), International Conference on*, pp. 100–107, 2000.

[Swam 02] R. Swaminathan, M. D. Grossberg, and S. K. Nayar. "Caustics of catadioptric cameras". In: *International Conference on Computer Vision, IEEE Conference on*, pp. 2–9, 2002.

[Szel 07] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. "A comparative study of energy minimization methods for markov random fields with smoothness-based priors". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1068–1080, 2007.

[Szel 08] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. "A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 6, pp. 1068–1080, June 2008.

[Taba 09] A. Tabaee, V. K. Anand, J. F. Fraser, S. M. Brown, A. Singh, and T. H. Schwartz. "Three-dimensional endoscopic pituitary surgery". *Neurosurgery*, Vol. 64, No. 5, p. 288–295, 2009.

[Taba 84] A. J. Tabatabai and O. R. Mitchell. "Edge location to subpixel values in digital imagery". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, No. 2, pp. 188–201, 1984.

[Taji 90] J. Tajima and M. Iwakawa. "3-D data acquisition by Rainbow Range Finder". In: *Pattern Recognition, International Conference on*, pp. 309–313, 1990.

[Take 07] J. Takei, S. Kagami, and K. Hashimoto. "3,000-fps 3-D shape measurement using a high-speed camera-projector system". In: *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 3211–3216, 2007.

[Take 09] T. Takeshita, Y. Nakajima, M. K. Kim, S. Onogi, M. Mitsuishi, and Y. Matsumoto. "3D Shape Reconstruction Endoscope using Shape from Focus". In: *International Conference on Computer Vision Theory and Applications*, p. 411–416, 2009.

[Take 83] M. Takeda and K. Mutoh. "Fourier transform profilometry for the automatic measurement of 3-D object shape". *Applied Optics*, Vol. 22, No. 24, pp. 3977–3982, 1983.

[Take 96] M. Takeda and T. Abe. "Phase unwrapping by a maximum cross-amplitude spanning tree algorithm: a comparative study". *Optical Engineering*, Vol. 35, No. 8, pp. 2345–2351, 1996.

[Tard 09] J. Tardif, P. Sturm, M. Trudeau, and S. Roy. "Calibration of cameras with radially symmetric distortion". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 9, pp. 1552–1566, 2009.

[Thor 02] T. Thormahlen, H. Broszio, and P. Meier. "Three-dimensional endoscopy". In: *Medical Imaging in Gastroenterology and Hepatology*, p. 199–212, 2002.

[Tomb 09] F. Tombari and K. Konolige. "A practical stereo system based on regularization and texture projection". In: *Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2009.

[Torr 84] V. Torre and T. A. Poggio. "On edge detection". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, No. 2, pp. 147–163, 1984.

[Trig 00] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. "Bundle adjustment—a modern synthesis". *Vision algorithms: theory and practice*, pp. 153–177, 2000.

[Trob 95] M. Trobina. "Error model of a coded-light range sensor". Tech. Rep., Communication Technology Laboratory, ETH Zurich, 1995.

[Truc 98] E. Trucco and A. Verri. *Introductory techniques for 3-D computer vision.* Vol. 93, Prentice Hall New Jersey, 1998.

[Tsai 92] R. Y. Tsai. "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses". In: L. B. Wolff, S. A. Shafer, and G. Healey, Eds., *Radiometry*, pp. 221–244, Jones and Bartlett Publishers, Inc., , USA, 1992.

[Tsuc 10] R. Tsuchiyama, T. Nakamura, T. Iizuka, A. Asahara, S. Miki, S. Tagawa, and S. Tagawa. *The OpenCL Programming Book.* Fixstars Corporation, 1 Ed., Apr. 2010.

[Vinc 91] L. Vincent and P. Soille. "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 6, pp. 583–598, June 1991.

[Voll 99] J. Vollmer, R. Mencl, and H. Mueller. "Improved Laplacian smoothing of noisy surface meshes". In: *Computer Graphics Forum*, p. 131–138, 1999.

[Vuyl 90] P. Vuylsteke and A. Oosterlinck. "Range Image Acquisition with a Single Binary-Encoded Light Pattern". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 2, pp. 148–164, 1990.

[Wang 08] H. Wang, D. Mirota, G. Hager, and M. Ishii. "Anatomical reconstruction from endoscopic images: Toward quantitative endoscopy". *American journal of rhinology*, Vol. 22, No. 1, p. 47, 2008.

[Wang 10] Y. Wang, K. Liu, D. Lau, Q. Hao, and L. Hassebrook. "Maximum SNR pattern strategy for phase shifting methods in structured light illumination". *Journal of the Optical Society of America A*, Vol. 27, No. 9, pp. 1962–1971, 2010.

[Wata 97] M. Watanabe and S. K. Nayar. "Telecentric optics for focus analysis". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 12, p. 1360–1365, 1997.

[Wein 09] H. L. Weinert. "A fast compact algorithm for cubic spline smoothing". *Computational Statistics & Data Analysis*, Vol. 53, No. 4, pp. 932–940, 2009.

[Weis 07] T. Weise, B. Leibe, and L. V. Gool. "Fast 3d scanning with automatic motion compensation". In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 1–8, 2007.

[Whit 94] R. G. White and R. A. Schowengerdt. "Effect of point-spread functions on precision edge measurement". *Journal of the Optical Society of America A*, Vol. 11, No. 10, pp. 2593–2603, 1994.

[Will 93] R. G. Willson and S. A. Shafer. "What is the Center of the Image?". Tech. Rep., Carnegie Mellon University, Pittsburgh, PA, USA, 1993.

[Wink 02] S. Winkelbach and F. Wahl. "Shape from single stripe pattern illumination". *Pattern Recognition*, pp. 240–247, 2002.

[Wior 01] G. Wiora. *Optische 3D-Messtechnik: Präzise Gestaltvermessung mit einem erweiterten Streifenprojektionsverfahren.* PhD thesis, Universität Heidelberg, 2001.

[Wiss 11] P. Wissmann, R. Schmitt, and F. Forster. "Fast and Accurate 3D Scanning Using Coded Phase Shifting and High Speed Pattern Projection". *3D Imaging, Modeling, Processing, Visualization and Transmission, International Conference on*, Vol. 0, pp. 108–115, 2011.

[Witk 84] A. Witkin. "Scale-space filtering: A new approach to multi-scale description". In: *Acoustic Speech Signal Processing, Proceedings of IEEE International Conference on*, pp. 150–153, 1984.

[Wood 80] R. J. Woodham. "Photometric method for determining surface orientation from multiple images". *Optical Engineering*, Vol. 19, No. 1, pp. 139–144, 1980.

[Wu 93] X. Wu. "Adaptive Split-and-Merge Segmentation Based on Piecewise Least-Square Approximation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 8, pp. 808–815, 1993.

[Wust 91] C. Wust and D. Capson. "Surface profile measurement using color fringe projection". *Machine vision and applications*, Vol. 4, pp. 193–203, 1991.

[Wyan 02] J. C. Wyant. "White light interferometry". In: *Proceedings of the SPIE*, pp. 98–107, 2002.

[Xion 02] Y. Xiong and L. Matthies. "Error analysis of a real-time stereo system". In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 1087–1093, 2002.

[Yang 96] Z. Yang and Y. F. Wang. "Error analysis of 3D shape construction from structured lighting". *Pattern Recognition*, Vol. 29, No. 2, pp. 189–206, 1996.

[Yedi 03] J. S. Yedidia, W. T. Freeman, and Y. Weiss. *Understanding belief propagation and its generalizations*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[Ying 05] Q. Ying-Dong, C. Cheng-Song, C. San-Ben, and L. Jin-Quan. "A fast subpixel edge detection method using Sobel-Zernike moments operator". *Image and Vision Computing*, Vol. 23, No. 1, pp. 11–17, 2005.

[Youn 07] M. Young, E. Beeson, J. Davis, S. Rusinkiewicz, and R. Ramamoorthi. "Viewpoint-Coded Structured Light". In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 1–8, June 2007.

[Zale 06] Z. Zalevsky, A. Shpunt, A. Maizels, and J. Garcia. "Method and System for Object Reconstruction". March 2006. US Patent App. 11/991,994.

[Zhan 00] Z. Zhang. "A Flexible New Technique for Camera Calibration". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 11, 2000.

[Zhan 02] L. Zhang, B. Curless, and S. M. Seitz. "Rapid shape acquisition using color structured light and multi-pass dynamic programming". In: *3D Data Processing Visualization and Transmission, International Symposium on*, pp. 24–36, June 2002.

[Zhan 04] S. Zhang and P. Huang. "High-Resolution, Real-time 3D Shape Acquisition". In: *Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Conference on*, p. 28, 2004.

[Zhan 07] S. Zhang and S. Yau. "Generic nonsinusoidal phase error correction for three-dimensional shape measurement using a digital video projector". *Applied Optics*, Vol. 46, No. 1, pp. 36–43, 2007.

[Zhan 99] R. Zhang, P. S. Tsai, J. E. Cryer, and M. Shah. "Shape-from-shading: a survey". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 8, pp. 690–706, 1999.

[Zhao 96] W. Zhao and N. Nandhakumar. "Effects of camera alignment errors on stereoscopic depth estiamtes". *Pattern Recognition*, Vol. 29, No. 12, pp. 2115–2126, 1996.

[Zhao 97] B. Zhao and Y. Surrel. "Effect of quantization error on the computed phase of phase-shifting measurements". *Applied Optics*, Vol. 36, No. 10, pp. 2070–2075, 1997.

[Zhou 10a] C. Zhou, O. Cossairt, and S. Nayar. "Depth from Diffusion". In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 1110–1117, 2010.

[Zhou 10b] J. Zhou, Q. Zhang, B. Li, and A. Das. "Synthesis of stereoscopic views from monocular endoscopic videos". In: *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, p. 55–62, 2010.

[Zick 02] T. E. Zickler, P. N. Belhumeur, and D. J. Kriegman. "Helmholtz stereopsis: Exploiting reciprocity for surface reconstruction". *International Journal of Computer Vision*, Vol. 49, No. 2, pp. 215–227, 2002.

[Zitn 07] C. L. Zitnick and S. B. Kang. "Stereo for image-based rendering using image over-segmentation". *International Journal of Computer Vision*, Vol. 75, No. 1, pp. 49–65, 2007.