On Sinusoidal Structured Light Reconstruction

An Entire Pipeline with Improvements in Accuracy, Stability, Robustness and Speed

Thesis approved by the
Department of Computer Science
University of Kaiserslautern-Landau
for the award of the Doctoral Degree
Doctor of Engineering (Dr.-Ing.)

to

Torben Fetzer

Date of Defense:	November 3, 2023
Dean:	Prof. Dr. Christoph Garth
Reviewer:	Prof. Dr. Didier Stricker
Reviewer:	Prof. Dr. Reinhard Koch
Reviewer:	Dr. Peter Sturm

DE-386

Abstract

The field of 3D reconstruction is one of the most important areas in computer vision. It is not only of theoretical importance, but it is also increasingly used in practical applications, be it in reverse engineering, quality control or robotics. In practical applications, where high precision reconstructions are required for a large variety of different objects, structured light reconstruction is often the method of choice. It allows to achieve accurate and dense point correspondences over the entire scene, regardless of object texture or features. Techniques that project phase-shifted sinusoidals are widely used because, based on the harmonic addition theorem, they theoretically allow surface encoding in full camera resolution invariant to the object's coloring.

In this thesis, a fully-automatic reconstruction pipeline based on the sinusoidal structured light technique is presented. From the projection of the fringe patterns for encoding the object's surface, the robust matching of the point correspondences in sub-pixel accuracy, the auto-calibration of the setup including the active device, up to the fully-automatic alignment of the partial reconstructions, all steps will be described and examined in detail. During that, improvements will be achieved in the area of matching, obtaining highly accurate and topologically consistent correspondences in sub-pixel precision between all the devices used. Furthermore, the auto-calibration from point correspondences, based on the epipolar geometry of the structured light system is improved. Weaknesses of previous methods in the extraction of focal lengths from the fundamental matrices are discovered and addressed. The partial point clouds, reconstructed from the auto-calibrated devices, are finally pre-aligned using a neural network approach, based on light-resistant optical flow estimation and subsequently refined using a global approach.

The weaknesses of the structured light method itself will also be addressed and partially fixed during the course of this work. Since it is an active reconstruction method, certain surface properties can affect the quality of the reconstruction. It will be shown how these problems can be eliminated or at least be reduced using an iterative approach that combines fringe patterns with an inverse texture. Another weakness of the method is its time-consuming acquisition procedure. Typically, a large number of horizontal and vertical fringe patterns are projected onto the scene to achieve high-precision encoding despite the limited dynamic range and resolution of the projector. Therefore, a method will be presented which allows to combine the horizontal and vertical patterns for a simultaneous two dimensional surface encoding.

Contents

Chapter 1

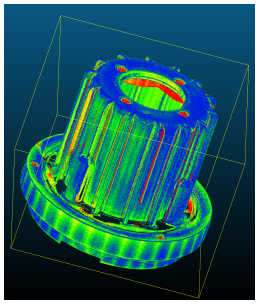
Introduction

Contents

Digital 3D reconstructions of a wide spectrum of real objects are increasingly finding their way into our modern society. In some areas, such as digital shopping, we visibly encounter 3D models and consciously perceive the advantages that reconstruction methods enable. But there are many other areas where they already perform important tasks in the background, often without us knowing or noticing. Whether in industry, cultural heritage, medicine or commerce, the areas of application are impressive and constantly growing, and this trend is by no means expected to slow down.

A particularly important application of 3D reconstruction is the documentation of our cultural heritage. There is a great amount of ephemeral culture that can be preserved for posterity in digital form. In the meantime, museums have begun to digitize an increasingly large part of their exhibits in order to document them, to maintain them, and at the same time to make them accessible to a broad public in digital exhibitions. The weathering and destruction of centuries-old objects, statues and buildings can be controlled, documented and partially prevented with the help of these methods. For example, ancient statues are examined for damage using 3D reconstruction techniques and restored as weathering progresses. In many cases, this prevents irreversible decay. Fig-

(a) 3D reconstruction in cultural heritage [116]



(b) Quality control

(c) Medical tooth prothesis [1]

(d) Object tracking with NNs [127]

Figure 1.1.: *Examples of applications of 3D reconstruction: (a) Different views of a digitized antique marble statue. (b) An example of automatic quality control in industrial production. An overlay of the reconstructed part with the associated CAD model enables highly accurate inspections. (c) Medical dentures can be fitted precisely with the help of 3D reconstructions. (d) For applications in object recognition and tracking, artificial neural networks can be trained using rendered 3D models.*

ure 1.1 (a) shows different views of a reconstruction of a marble statue, which was carried out to meticulously and traceably document damaged and already repaired parts.

Besides that, 3D reconstructions will also be an important component in modern industry. For instance, they enable fully automated quality control in areas of production. Components can be checked for measurement accuracy and quality defects without user interaction. For this purpose, Figure 1.1 (b) shows an overlay of a reconstructed component with the corresponding CAD model. Surface defects and dimensional inaccuracies can be detected with high precision in relation to the ground truth given by the CAD model. Independently of this, the processes also play a leading role in reverse engineering. Components for which no plans, drawings or models exist can be scanned and rebuilt, which allows for example the reproduction of antique machines.

Also medical science is already aware of the advantages of modern reconstruction techniques. The body's own parts, such as joints or teeth, do not have a blueprint and often cannot be replaced with an exact fit if the worst comes to the worst. Through the process of reverse engineering with 3D re-

constructions, highly accurate replicas can be created and thus give people back their sense of life. Also, new dentures, often have to be fitted precisely to remaining tooth parts. Here, precise 3D models can already avoid a lot of effort and pain (see Figure 1.1 (c)).

Especially in retail, we are all particularly often confronted with the many benefits of 3D models. Digital product views have been used in sales for quite some time and are used as a strategy to spotlight the goods. High-quality 3D models allow customers to preview products and assess the effect in interaction with other components. Such digital presentations will continue to be an important element in this increasingly digitizing domain.

Lastly, one of the supreme applications of the digital age should be mentioned. In near future, it is very likely that we will do our weekly shopping in fully automated supermarkets. This will require stable and reliable recognition, classification and tracking of our goods in the supermarket, eliminating the need for time-consuming individual bar-code scanning of goods after shopping. Large companies are already training artificial neural networks using 3D models of their products to recognize them securely in any situation (Figure 1.1 (d)). First prototype-stores are already being tested. With a constantly growing product range, new goods have to be digitized again and again and the networks have to be permanently retrained. For this purpose, 3D scanners are indispensable, enabling fully automated and user-friendly reconstruction of a large number of different product types, in order to provide a large amount of high-quality training data, given by rendered images of the products.

The application areas of 3D reconstruction will certainly continue to grow as digitization progresses and reconstruction methods become more applicable and user-friendly. Also, the increasingly available technologies of 3D printing make reconstruction methods, for quickly obtaining printable models, more and more appealing. In particular, the printability of a wide variety of materials such as plastic, metal, and concrete will make this push noticeable. In this context, another impact on the preservation of cultural heritage through collapsing buildings should be mentioned. For precisely this case, where plans of complex structures, worth to be preserved, do not exist, there is often no practical way to rescue them. Therefore, a combination of modern 3D reconstruction and printing techniques could provide a new low-cost alternative.

1.1. Motivation

There is already a multitude of different 3D scanners available for purchase. These provide 3D models of varying quality based on different reconstruction techniques. These approaches each have advantages and disadvantages that qualify or exclude them for different demanding applications, such as specular surfaces or smooth and un-textured ones. A silver bullet has not been found, yet.

While passive reconstruction approaches are often quite flexible, the quality of the obtained results may not be sufficient for many applications. These methods are usually dependent on identifiable and distinguishable features in

the object texture or geometry that allow to find matches between different camera views. Especially in industrial applications (such as for example the production of industrial parts), often no feature-rich texture can be expected on the objects, which makes finding a large number of correspondences difficult or even completely impossible. Dense reconstructions can thus usually not be guaranteed, which strongly limits the applicability.

Therefore, active reconstruction methods are widely used. They allow to reliably provide accurate and dense reconstructions for a large variety of different object types. In fact, with these techniques, it is not relevant whether the objects are textured or have geometric features or not. Nevertheless, such methods rely on the successful interaction of an active component, that introduces information into the scene, and passive components such as one or more cameras. To ensure this, pre-calibrated fixed setups are usually used for which this interaction can be ensured. Unfortunately, such a setup restricts the usability considerably. The fixed baseline between the devices and the inflexible focal settings of the cameras make the method applicable only for a limited working volume. However, in many applications it would be desirable to be able to adjust the baseline and zoom in on objects as desired. This would allow to flexibly digitize different object sizes in different applications with any desired resolution. Another major disadvantage of pre-calibrated devices is the pre-definition of the devices used. While industrial applications may require particularly high-quality reconstructions, private users may prefer to be satisfied with inexpensive customer devices and the associated loss of quality. In addition, the hardware may have to meet special requirements for different industrial applications, such as being heat resistant or insensitive to moisture. This can only be implemented at great expense in a pre-defined and pre-calibrated setup.

There are alternative approaches that allow calibration of the devices used after each setting change such as positional alignment or focal point adjustment. Thereby, the calibration of the cameras is determined by means of special calibration targets, such as checkerboard patterns, which have to be recorded from different perspectives by all devices. In particular, the calibration of the active component is often particularly complex and in most cases requires special setups. In all cases, such an approach means a significant intervention of the user and a non-negligible amount of time. The operator must know exactly what he is doing and there are many errors that can occur and lead to a failure of the calibration process, which makes the application of such procedures almost impossible for laymen.

1.2. Problem Statement

The goal of this work is to develop a complete pipeline for fully-automatic 3D reconstruction that is as flexible, cost-effective and user-friendly as possible. It should allow to reconstruct a wide range of different object types with high accuracy, density and resolution in a wide variety of applications.

As few requirements as possible should be placed on the objects to be recon-

structed, and the hardware used should be freely available (customer devices) and interchangeable. The method should remain flexible and easily adaptable to a wide range of applications with different requirements without relying on special devices. In this way, resolution and reconstructed detail density can be efficiently controlled by the hardware used. This makes it possible to use, for example, high-quality industrial hardware for applications in production, while at the same time offering an affordable variant with customer devices for private needs.

Unlike most existing systems, the method should be self-calibrating so that it can be set up and adapted to new use cases without complicated and time-consuming procedures and user interaction. The entire reconstruction process, from the acquisition of the data, the calibration of the setup, the generation of the 3D data, the alignment of the partial views, the meshing of the 3D structure and the texturing of the model, should be done fully-automatically. In this way, it should be intuitively applicable and successfully usable even for untrained operators.

1.3. Contributions

During the work on the mentioned task a number of publications resulted which are the basis of this thesis. They are the pillars of almost all the chapters, from matching, over calibration up to automatic registration of point clouds. In addition, new theoretical and application-related results could be obtained and published with respect to the weaknesses of the chosen structured light method itself. The following contributions are the core of this thesis:

[42] **Fast Projector-Driven Structured Light Matching in Sub-Pixel Accuracy using Bilinear Interpolation Assumption**, T. Fetzer, G. Reis and D. Stricker, *Proceedings of the International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2021

A method is introduced that allows to find optimal sub-pixel positions for an arbitrary number of devices in a structured light setup in linear complexity. For this purpose, the quadrilateral regions containing the sub-pixels are extracted. The convexity of these quads and their consistency in terms of topological properties can be guaranteed during runtime. Subsequently, an explicit formulation of the optimal sub-pixel position within each quad is derived, using bilinear interpolation, and the permanent existence of a valid solution is proven. Due to the ensured topological properties, exceptionally smooth, highly precise, uniformly sampled matches with almost no outliers are achieved. The point correspondences obtained do not only have an enormously positive effect on the accuracy of reconstructed point clouds and resulting meshes, but are also extremely valuable for auto-calibrations calculated from them.

[46] **Robust Auto-Calibration for Practical Scanning Setups from Epipolar and Trifocal Relations**, T. Fetzer, G. Reis and D. Stricker, *Proceedings of the International Conference on Machine Vision Applications (MVA)*, 2019

In this paper, we showed how to generate a highly accurate epipolar geometry between more than two views. We investigated the quality of resulting fundamental matrices based on methods that minimize the epipolar error and the trifocal error. We showed that the trifocal error, which tries to reconcile three devices, gives very good results in the case of very accurate point correspondences, but is much more prone to outliers and noise than the well-known epipolar error, which considers only pairwise views. We further showed how significant advantages can be drawn from both approaches by combining both error types with a suitable weighting parameter. For auto-calibration techniques based on the underlying fundamental matrices, this has a particularly large impact on subsequent steps. In addition to the increased probability of successful calibrations, we also showed the increased accuracy of camera matrices whose parameters were extracted from the fundamental matrices calculated in this way. Significantly lower back-projection errors of triangulated points demonstrate the beneficial results.

[48] **Stable Intrinsic Auto-Calibration from Fundamental Matrices of Devices with Uncorrelated Camera Parameters**, T. Fetzer, G. Reis and D. Stricker, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2020

Auto-calibration is an important task in computer vision and is desirable for many visual applications. Unfortunately, methods based on the epipolar geometry of the devices are very sensitive to errors in the fundamental matrices. In addition they need good initialization to converge to the global solution. In practice this leads to difficulties if optical parameters like principal point or focal length are unconstrained. In such situations, even auto-calibration methods tend to diverge and do not yield a valid calibration. In this work, the reasons for this behavior are investigated, in particular for the initialization method of Bougnoux [17] and Lourakis' auto-calibration method [104]. Based on this analysis, a more stable method is proposed. A continuous and smooth energy function is introduced, which offers better convergence properties. Finally, a thorough evaluation was performed and a detailed comparison with the state of the art is presented.

[43] **INV-Flow2PoseNet: Light-Resistant Rigid Object Pose from Optical Flow of RGB-D Images using Images, Normals and Vertices**, T. Fetzer, G. Reis and D. Stricker, *MDPI Sensors 22(22)*, 2022

This paper presents a novel architecture for simultaneous estimation of highly accurate optical flows and rigid scene transformations for difficult scenarios where the brightness constancy assumption is violated by strong illumination changes. In the case of rotating objects or moving light sources, such as those encountered for driving cars in the dark, the scene appearance often changes significantly from one view to the next. The presented method fuses texture and geometry information by combining images, vertices and normal vectors to compute an illumination-invariant optical flow. By using a coarse-to-fine strategy, globally anchored optical flows are learned, reducing the impact of erroneous shading-based pseudo-correspondences. Based on the learned optical flows, a second architecture is proposed that predicts robust rigid transformations from the warped vertex and normal maps. The method has been evaluated on a newly created dataset containing both synthetic and real data with strong rotations and shading effects. This data represents the typical use case in 3D reconstruction, where the object often rotates in large steps between the partial reconstructions. Additionally, we apply the method to the well-known Kitti Odometry dataset.

[45] **Joint Global ICP for Improved Automatic Alignment of Full Turn Object Scans**, T. Fetzer, G. Reis and D. Stricker, *Proceedings of the International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2021

In this paper, we have shown how to stably register closed turns of partial reconstructions, as they typically result from 3D scanners. Thereby, we formulated the global problem as a joint minimization problem and showed how it can be effectively minimized. We showed that there are considerable advantages if ICP iterations are performed jointly instead of the usual pairwise approach. Without the need for increased computational effort, lower alignment errors are achieved, drift is avoided and calibration errors are uniformly distributed over all scans. The joint approach is further extended into a global version, which not only considers one-sided adjacent scans, but updates symmetrically in both directions. The result is an approach that leads to a much smoother and more stable convergence, which moreover enables a stable stopping criterion to be applied. This makes the procedure fully-automatic and therefore superior to most other methods, that often tremble close to the optimum and have to be terminated manually. We presented a complete procedure, which in addition addresses the issue of automatic outlier detection in order to solve the investigated problem data independently, without any user interaction.

[47] **Simultaneous Bi-Directional Structured Light Encoding for Practical Uncalibrated Profilometry**, T. Fetzer, G. Reis and D. Stricker, *Proceedings of the International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2021

Profilometry based on structured light is one of the most popular methods for 3D reconstruction. If it is used to encode the scene in horizontal as well as in vertical direction it allows to compute point correspondences without a pre-calibrated setup. On the contrary, calibration can be estimated directly from the correspondences in this way. Unfortunately, a significant disadvantage is that a large number of images of the scene, with differently illuminated fringe patterns, has to be captured, which yields a considerable amount of acquisition time. This paper presents a new approach that encodes the scene simultaneously in horizontal and vertical directions using combined sinusoidal fringe patterns. This allows to almost halve the number of recorded images, making the approach attractive again for many practical applications with time aspects.

[44] **Iterative Color Equalization for Increased Applicability of Structured Light Reconstruction**, T. Fetzer, G. Reis and D. Stricker, *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, 2020

Due to the accuracy and density of the reconstructions obtained, the structured light approach, whenever applicable, is often the method of choice for industrial applications. Nevertheless, it is an active approach which, depending on material properties or coloration, can lead to problems and fail in certain situations. In this paper, a method based on the standard structured light approach is presented that significantly reduces the influence of the color of a scanned object. It improves the results obtained by iterative application in terms of accuracy and general applicability. Especially in high-precision reconstruction of small structures or high-contrastly colored and specular objects, the technique shows its greatest potential. The advanced method requires neither pre-calibrated cameras or projectors nor information about the equipment. It is easy to implement and can be applied to any existing scanning setup.

1.4. Organization of the Thesis

This is a brief outline of the chapters of the thesis presented. While the first chapters reflect the respective steps of the proposed reconstruction pipeline, the last chapters deal in particular with weaknesses of the chosen active reconstruction approach.

Chapter 2 This chapter will summarize the fundamental theory on which this work is based on. The process of 3D reconstruction will briefly be repeated. It will be explained what happens during calibration, how point correspondences can be generated using the structured light approach based on phase-shifted sinusoidal fringe patterns and how the 3D positions can be estimated from multiple views.

Chapter 3 The various steps are outlined, which the presented 3D reconstruction pipeline goes through in order to obtain a fully-automatic procedure that is user-friendly and applicable. Detailed information about the setup of the 3D scanner that was used for most of the acquired data is provided.

Chapter 4 It will be explicitly shown how to effectively obtain dense point correspondences from a two-dimensional structured light surface encoding between all the cameras used and the projector. It will also be presented how to ensure that the obtained correspondences have excellent consistency properties.

Chapter 5 Detailed information on how the setup, including all cameras and the projector, can be calibrated in a convenient way without user interaction, will be delivered in this chapter. It will be shown how the epipolar geometry can be determined, the intrinsic parameters can be stably extracted, and the relative extrinsics can be calculated. Finally, it will be shown how the setup can be further refined in order to achieve maximal accuracy.

Chapter 6 In this chapter we will show, how the partial point clouds, that are received from the different views can be efficiently pre-aligned using a new approach based on artificial neural networks, that are in addition robust to light and shading changes as they often appear for rotating objects. This comes up for the common case of fully automatic scanners, where turntables are used to ease acquisition.

Chapter 7 A method will be presented, that allows roughly initialized point clouds to be aligned by a global variant of *Iterative Closest Points*. The method especially takes into account, that closed turns of partial object scans are given, where the last scan overlaps the first one, which yields an over-determined registration problem.

Chapter 8 Techniques for representing the reconstructed structure will be briefly mentioned in order to complete the general reconstruction pipeline. In particular, normal vectors and meshes will be discussed, which allow to represent realistic images of the reconstructed models in a memory-efficient way.

Chapter 9 A procedure will be presented that allows to speed up the acquisition by combining the horizontal and vertical fringe images. This is especially of interest, if the correspondences are used for auto-calibrations with slow devices (low frame-rate), such as SLR cameras, which may be extremely time-consuming.

Chapter 10 The penultimate chapter will show how disadvantages of the structured light method itself, due to the active interaction with the object's surface, can be handled. Especially in the case of specular surfaces, high-contrast object coloring and high precision reconstruction of small structures, this can yield large benefits.

Chapter 11 Finally a conclusion of the presented work will be given. It will summarize the tasks of the thesis, its contributions and give suggestions for future work.

Chapter 2

Background

Contents

2.1. Introduction

The task of 3D reconstruction is to create realistic and geometrically identical (up to scale) 3D models of scenes, that have been captured by 2D images. In order to obtain a reliable estimate of the depth of a 3D scene, multiple images from different perspectives have to be acquired. Using underlying camera models that describe the projection process that led to the creation of the scene's images, the 3D position of each projected scene point can be calculated by intersecting light rays from the scene point to the different camera centers. For this purpose, the scene must be viewed from at least two different perspectives in order to make the problem solvable. Additional views can further improve

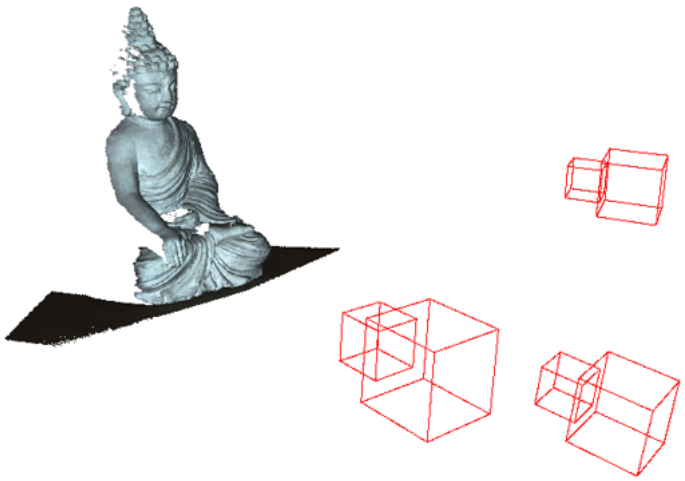


Figure 2.1.: *The task of 3D reconstruction is to compute an accurate 3D model from captured images of a scene. The figure shows a reconstructed point cloud with visualizations of the three calibrated devices, that have been used for the reconstruction process.*

the quality of the depth estimate. In order to get multiple views of an object, a single camera can be used to move around a static scene. Conversely, the object may be moving in front of a single camera, to provide different views. Or, as it is often the case, multiple cameras in a stereo or multi-view setup can be used at the same time. Figure 2.1 visualizes a reconstructed model from three views. The camera views, that were used for reconstruction are additionally sketched in the scene.

There are also first methods that estimate depth from single RGB images ([144], [97], [59]) using focus cues to guess relative depth. These can indeed be used to estimate which objects in a scene are closer and which are further away, but they are far from achieving high-quality 3D models of an object within the depth of field of a camera. Therefore, at this point there seams to be no way around the classical approach with at least two different camera views.

The basis for the calculation of a 3D point in the scene is the knowledge of its projected location in the respective captured images. Corresponding points between the images of the different views must be found so that the 3D position can be triangulated using the camera models. Basically, the problem of 3D reconstruction can be reduced to the following steps, that are addressed in this chapter:

- Calibration of the cameras
- Matching of corresponding points in the camera images
- Triangulation of the 3D scene points

With a multi-view approach, that follows these steps, only the parts of a scene can be reconstructed that are seen by all involved cameras. In the context of

3D scanners, which are usually supposed to generate complete reconstructions of an object, it is therefore necessary to perform several partial reconstructions and to align them afterwards. This way, a complete point cloud can be generated. More detailed information about the procedure of point cloud alignment will be given in chapter 6 and 7. In order to visualize the generated point clouds, additional information such as normal vectors and surface reflectance properties are often estimated and modeled. In addition, surfaces are often expressed by textured meshes in order to save memory during representation. Further steps regarding the object representation will be briefly addressed in Chapter 8.

2.2. Calibration

The process of calibration specifies the calculation of all parameters of a virtually modeled pinhole camera, which lead to the generation of 2D images of a 3D scene. A distinction is made between intrinsic and extrinsic calibration. While the intrinsic camera parameters model the projection process of a 3D point in the camera's coordinate system onto the image plane, extrinsic parameters specify the camera's position and orientation in a fixed world coordinate system.

2.2.1. Intrinsic Camera Parameters

The intrinsic calibration matrix of a camera is defined in the following way:

$$\mathbf{K} = \begin{pmatrix} f_x & s & x_p \\ 0 & f_y & y_p \\ 0 & 0 & 1 \end{pmatrix} \tag{2.1}$$

It contains the camera's parameters that determine the projection process:

- Focal lengths in pixels f_x and f_y, that define the projection cone. For modern devices that have been manufactured in high-quality, square pixels can be assumed, whereby $f := f_x = f_y$ can be considered in the following.
- Skew parameter s that models non-rectangular pixel properties. For modern devices, this also no longer occurs, which means that this value can be set equal to 0 throughout.
- Principal point located at coordinates (x_p, y_p) that specifies the position where the optical ray hits the image plane. It is often assumed to be in the image center, but this does not hold true for arbitrary devices such as for example video projectors. Also for cameras that are particularly small or cheaply manufactured, such as those in some light field systems, this cannot be guaranteed in general. The location of the principal point also has a not insignificant influence on focal length calculations in calibration procedures. Therefore, it should not be considered to be fixed at an uncertain estimate in general.

Applying calibration matrix $\mathbf{K}$ to a 3D point $\mathbf{X}_C = (X_C, Y_C, Z_C)^\mathsf{T} \in \mathbb{R}^3$ in the camera coordinate system, performs the projection process onto an image point $\tilde{\mathbf{x}} = (u_I, v_I, w_I)^\mathsf{T} \in \mathbb{R}^3$ in homogeneous image space:

$$\mathbf{K}\mathbf{X}_C = \begin{pmatrix} f & 0 & x_p \\ 0 & f & y_p \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_C \\ Y_C \\ Z_C \end{pmatrix} = \begin{pmatrix} u_I \\ v_I \\ w_I \end{pmatrix} = \tilde{\mathbf{x}} \qquad \Rightarrow \quad \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{w_I} \begin{pmatrix} u_I \\ v_I \end{pmatrix} \tag{2.2}$$

As visualized in Figure 2.2, the application of $\mathbf{K}$ maps the 3D point to the image space and then shifts the optical ray to the principal point of the image. The resulting ray $\tilde{\mathbf{x}}$ in homogeneous coordinates is finally mapped to image point $\mathbf{x} = (x, y)^\mathsf{T} \in \mathbb{R}^2$ on the 2D image plane by dividing and subsequently dropping the last entry w_I.

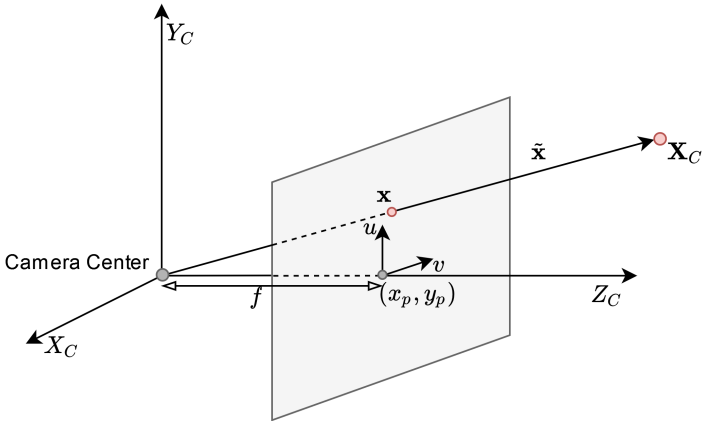


Figure 2.2.: *Sketch of the projection process in a pinhole camera model. The internal camera parameters define the projection operation.*

2.2.2. Extrinsic Camera Parameters

Independently of the internal projection procedure, the extrinsic camera parameters describe the location and orientation of the camera in a fixed world coordinate system and vice versa. The so-called pose of a camera is given by an orthogonal rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ with $\det(\mathbf{R}) = 1$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$. For imagination, Figure 2.3 shows the transformation of a point in world coordinates towards camera coordinates. While the rotation matrix $\mathbf{R}$ rotates the axes of the world coordinate system towards the axes of the camera coordinate system, the translation vector $\mathbf{t}$ is given by the world origin with respect to the camera center.
If these extrinsic parameters are applied to a 3D point $\mathbf{X} = (X, Y, Z)^\mathsf{T} \in \mathbb{R}^3$ in the world coordinate system, its coordinates are transferred to the camera coordinate system:

$$\mathbf{R}\mathbf{X} + \mathbf{t} = \mathbf{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{t} = \begin{pmatrix} X_C \\ Y_C \\ Z_C \end{pmatrix} = \mathbf{X}_C \tag{2.3}$$

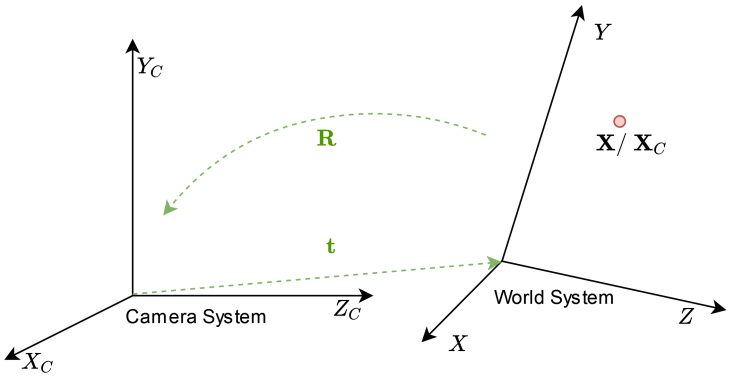


Figure 2.3.: *Sketch of the coordinate transform of a point in world coordinates to camera coordinates.*

In order to apply both rotation and translation in a single joint operation, the pose matrix $[\mathbf{R}|\mathbf{t}]$ is usually applied to a homogeneous version of the 3D world point $\tilde{\mathbf{X}} \in \mathbb{R}^4$, which is obtained by adding an additional dimension:

$$\mathbf{X}_C = \mathbf{R}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{t} = [\mathbf{R}|\mathbf{t}]\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = [\mathbf{R}|\mathbf{t}]\tilde{\mathbf{X}} \tag{2.4}$$

Conversely, these parameters can also express the position of the camera in the world coordinate system by $-\mathbf{R}^\mathsf{T}\mathbf{t}$.

2.2.3. Camera Model

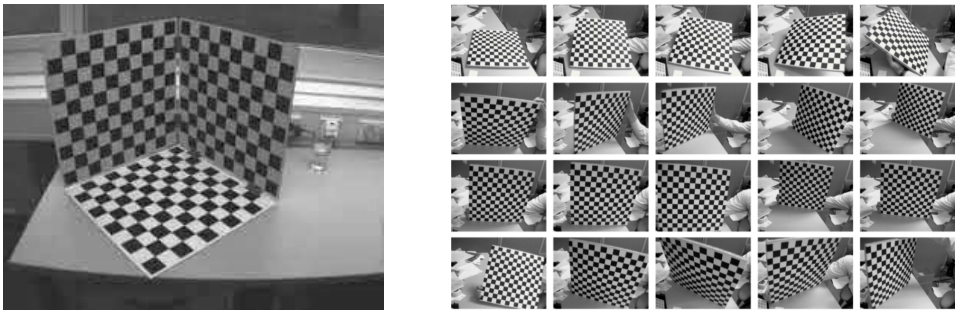
When working with multiple cameras it is important to anchor them in a common coordinate system. In the following, it will be assumed that all points are given in the world coordinate system and all extrinsic camera parameters are defined in relation to this reference system. A complete transformation of a 3D point in world coordinates to camera image coordinates can be achieved by applying a camera matrix $\mathbf{P} := \mathbf{K}[\mathbf{R}|\mathbf{t}]$ in homogeneous space:

$$\tilde{\mathbf{x}} = \mathbf{P}\tilde{\mathbf{X}} = \mathbf{K}[\mathbf{R}|\mathbf{t}]\tilde{\mathbf{X}} \tag{2.5}$$

The application of the camera matrix is a subsequent execution of the coordinate transformation (introduced in Section 2.2.2) that transforms the world point into the camera coordinate system, followed by the projection process (from Section 2.2.1) that maps the point to homogeneous image space.

2.2.4. Calibration from Calibration Targets

In order to calibrate a camera, common procedures use either known 3D structures given by special 3D calibration targets, or 2D calibration targets such as checkerboards, as shown in Figure 2.4.



(a) 3D calibration target [7]

(b) Multiple views of 2D calibration target [147]

Figure 2.4.: *Example of 3D and 2D calibration targets, that are commonly used for camera calibration.*

3D Calibration Target A 3D calibration object as depicted in Figure 2.4 (a) delivers a set of 3D-2D point correspondences $\{\mathbf{X}_1 \leftrightarrow \mathbf{x}_1, ..., \mathbf{X}_N \leftrightarrow \mathbf{x}_N\}$. Thereby the structure of the 3D points in space is known. Corresponding 2D points in the image can be easily detected from the checkerboard patterns (see [161], [119]). If the 3D points are not all in one plane, a camera matrix $\mathbf{P}$ can be directly fitted into these point correspondences. Therefore, the correspondences are transformed into homogeneous coordinates and the cross product is minimized, allowing for scale invariant minimization. Explicitly, the following minimization problem is solved using singular value decomposition:

$$\underset{\mathbf{P}\in\mathbb{R}^{3\times4}}{\operatorname{argmin}} \sum_{n=1}^{N} \|\mathbf{P}\tilde{\mathbf{X}}_n \times \tilde{\mathbf{x}}_n\|_2^2 \,, \qquad \text{s.t. } \|\mathbf{P}\|_{\mathcal{F}} = 1, \tag{2.6}$$

where $\|\cdot\|_{\mathcal{F}}$ denotes the Frobenius matrix norm, that is equivalent to the Euclidean norm of a vectorized version of the matrix.

In a subsequent step, calibration matrix $\mathbf{K}$ and rotation matrix $\mathbf{R}$ can be extracted from $\mathbf{P}$ using *QR-decomposition* on a sub-matrix of $\mathbf{P}$, that results from dropping the last column. Finally, knowing $\mathbf{K}$, the translation can be directly extracted from the last column of $\mathbf{P}$. Further detailed information about such standard methods in computer vision can be found in [62].

2D Calibration Target A much more widely used method that is the standard in target-based camera calibration is a homography-based technique as introduced by Zhang in [188]. Given an image of a 2D checkerboard in 3D space (see Figure 2.4 (b)) corners of the pattern can be reliable detected in the image using strategies that involve the arrangement of the corners of the checkerboards (see [170], [110]). Thus, assuming that the X-Y

plane of the world coordinate system coincides with the plane of the checkerboard, we obtain 2D-2D correspondences $\{\mathbf{X}_1 \leftrightarrow \mathbf{x}_1, ..., \mathbf{X}_N \leftrightarrow \mathbf{x}_N\}$, where $\mathbf{X}_n = (X_n, Y_n)^\mathsf{T} \in \mathbb{R}^2$, since $Z_n = 0$ is assumed. Under the assumption of

$$\tilde{\mathbf{x}}_n = \mathbf{K}[\mathbf{r}_1|\mathbf{r}_2|\mathbf{t}] = \begin{pmatrix} X_n \\ Y_n \\ 1 \end{pmatrix} = \mathbf{H}\tilde{\mathbf{X}}_n \tag{2.7}$$

with $\mathbf{r}_1$ and $\mathbf{r}_2$ being orthonormal rows of an unknown rotation matrix $\mathbf{R}$, we can fit a homography $\mathbf{H}$, that models the 2D-2D mapping by solving a similar problem to 2.6:

$$\underset{\mathbf{H}\in\mathbb{R}^{3\times3}}{\operatorname{argmin}} \sum_{n=1}^{N} \|\mathbf{H}\tilde{\mathbf{X}}_n \times \tilde{\mathbf{x}}_n\|_2^2 , \qquad \text{s.t. } \|\mathbf{H}\|_{\mathcal{F}} = 1 \tag{2.8}$$

If at least $M \geq 2$ checkerboards have been captured, that are not in a common plane, M independent homographies $\mathbf{H}_1, ..., \mathbf{H}_M$ can been computed and their common component $\mathbf{K}$ can be extracted. While each homography contains its own extrinsic parameters, the calibration matrix always remains the same. Thus, a simple further minimization problem can be set up and solved, yielding an estimate of the calibration matrix $\mathbf{K}$.

Assuming that one of the checkerboards is located in the desired world coordinate system, the vectors $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{t}$ can be obtained by inverse application of the calibration matrix to the respective homography matrix. The extrinsic camera parameters are finally given by $\mathbf{t}$ and $\mathbf{R} = [\mathbf{r}_1|\mathbf{r}_2|\mathbf{r}_1 \times \mathbf{r}_2]$.

The disadvantage of methods based on the utility of calibration targets is that they rely on special hardware. A particular drawback of the 3D calibration object is that the working volume is already limited by the object size. The volume of the space from which 3D points affect the calibration accuracy is already defined by the object. For different work ranges, therefore, different standardized calibration targets are required. In contrast, 2D checkerboards can theoretically cover an arbitrarily large space by placing the checkerboard at different locations in the scene in order to provide points from all regions for calibration. A disadvantage of this method, however, is the considerable user effort involved in placing the checkerboard in different parts of the room. A more convenient solution to this problem can be provided by *auto-calibration* methods, which estimate the calibration from unstructured point correspondences in different image views only, without knowledge about the 3D reference in space. More information on this and a procedure to apply it in a stable way for structured light setups is provided in detail later on in Chapter 5.

2.3. Point Matching

Matching is probably the main component of any 3D reconstruction procedure. The task is to identify which pixels in different views of an object represent the projection of the same 3D scene point. Once the locations of corresponding points between two calibrated views are determined, the 3D position of the 3D scene point can be estimated by out-projecting respective rays into the scene. Figure 4.2 shows two camera views of a 3D object and the rays that project a 3D point to respective image points.

Figure 2.5.: *Matching is the task of identifying corresponding image points between different views, that result from projecting the same 3D scene point to the images.*

A well known and popular approach to find matching points between different views of a scene is so called feature matching. This involves searching for meaningful points in the images and assigning properties to them using suitable descriptors. Based on these properties, corresponding features between the views are matched to each other. Particularly famous are, for example, *SIFT-Features* or *KAZE-Features* as described in [107], [106] and [5]. They are advantageous since they provide a set of properties that make them invariant under several transformations (such as scaling and translation), that typically occur from one image to the next. There is a number of other descriptors with their own advantages and disadvantages, as recently listed and compared by Tareen and Saleemin in [157]. The problem with feature-based methods is that they do not provide dense correspondences. They only yield matches in meaningful locations of the scene, such as edges or corners. This may work well for texture-rich objects with many unambiguous and non-repetitive features and also in many outdoor scenes, but fails for uniformly colored objects that have rather smooth geometries, such as in many industrial and medical applications. Dense and detailed reconstruction of such surfaces is hardly possible with feature-based methods. Therefore, this approach is not suitable for a general reconstruction method to be used in a wide variety of fields.

2.3.1. Epipolar Constraint

A typical approach to create dense correspondences over the whole scene is to use the calibration information of the different camera views. For each point in the first image, the respective projection ray can be out-projected into the scene. This ray theoretically hits at some point the corresponding 3D point in the scene. When this entire ray is projected onto the image of the second view, it provides a line, the so-called *epipolar line*, on which the corresponding point, as a projection of the same 3D point, must theoretically lie. Based on this *epipolar constraint*, the search for suitable correspondences is reduced from a two-dimensional search problem to a one-dimensional one. Figure 2.6 gives a sketch of this epipolar relation.

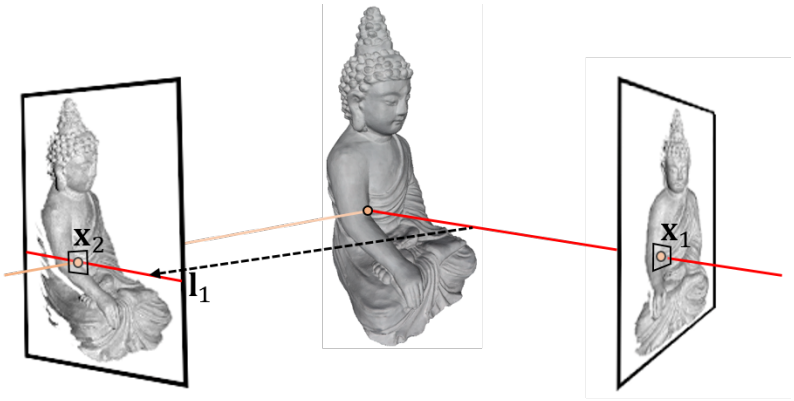


Figure 2.6.: *The epipolar geometry between two camera views constraints possible locations of correspondences between the views. If the calibration of the cameras is known, this reduces the search for corresponding points from a two-dimensional problem to a one-dimensional one.*

This can be modelled by a fundamental matrix $\mathbf{F} \in \mathbb{R}^{3\times3}$ (see [108] or [62]) that maps any point of the first image $\tilde{\mathbf{x}}_1$ in homogeneous representation to its corresponding epipolar line $\mathbf{l}_1$ in the second view:

$$\mathbf{l}_1 = \mathbf{F}\tilde{\mathbf{x}} = \mathbf{K}_2[\mathbf{t}]_\times \mathbf{R}\mathbf{K}_1^{-1}\tilde{\mathbf{x}}_1$$

Thereby, $\mathbf{K}_1$ and $\mathbf{K}_2$ are the instrinsic calibration matrices of the two cameras, $\mathbf{t}$ and $\mathbf{R}$ denote the relative extrinsics between the views and $[\cdot]_\times$ is the skew-symmetric cross-product matrix.

A typical approach to obtain rather dense matches between two calibrated views starts with determining the fundamental matrix from the known camera intrinsics and extrinsics. Possible candidates along or near the line are evaluated using regional descriptors such as *Sum of Absolute Distances*, *Normalized Cross Correlation* or many others ([15], [180], [66]) and the most coinciding pixels are selected for matching. This leads to significantly denser matches than sole feature approaches, but only works if points can be sufficiently identified and assigned by their neighborhood. For smoother surfaces or repetitive patterns, this approach is also not effective and only a limited improvement of the extremely sparse correspondences from feature matching methods.

2.3.2. Structured Light Encoding

A much denser encoding of the scene, independent of the surface texture or geometry, is made possible by so-called structured light approaches. Thereby, one of the cameras is replaced by a video projector that illuminates the scene by specific patterns, such as stripes, pseudo-random-dots, etc. and thus actively adds information to the 3D scene. The object itself is not required to have any features on its own; instead, the visible projected data serves to encode the surface. This makes the approach exceptionally well suited to ensure highly accurate and dense reconstructions of a variety of different object types. The calculated depth measurements are in no way inferior to those of other hardware-sensitive methods such as time-of-flight [54]. Particularly interesting are fringe projection strategies, where multiple shifted sinusoidal fringe patterns are projected onto the scene, in order to encode it in the shifted direction. In this way, texture invariant encoding of the surface at full camera resolution can be enabled. The method is based on practical application of the harmonic addition theorem as introduced in the next section. Figure 2.7 shows the setup in which one of the cameras has been replaced by a projector, illuminating the scene by an exemplary fringe pattern.

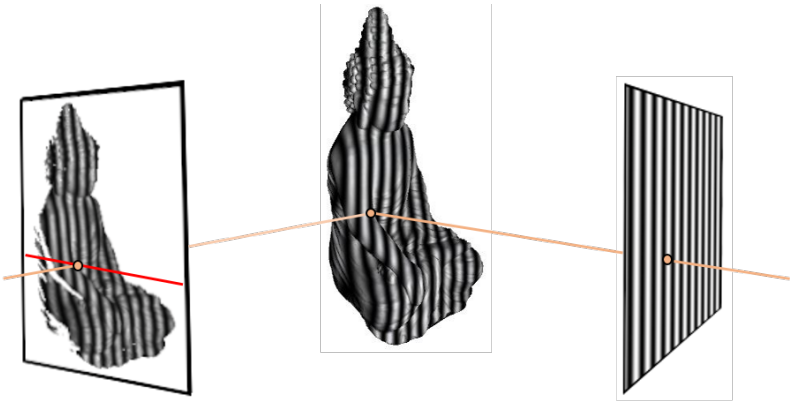


Figure 2.7.: *Sinusoidal structured light techniques use a video projector, that actively projects multiple phase-shifted fringe patterns onto the scene in order to encode the surface densely along the shifted direction.*

Phase Shift Method The projected patterns are modulated in one direction (horizontally in the example of Figure 2.7) by a sine/cosine and continued constantly in the other direction (vertically in the example of Figure 2.7). Let be given a set of N horizontal sinusoidal fringe patterns $\{P_1^H, ..., P_N^H\}$ with frequency F_H that are shifted $n = 1, ..., N$ times. Then the elements $P_n^H(i,j)$ of the patterns can be explicitly generated by

$$P_n^H(i,j) = \cos\left(\frac{2\pi j}{width} F_H + \frac{2\pi(n-1)}{N}\right), \qquad n = 1, ..., N\ . \tag{2.9}$$

Projecting these patterns onto the scene and capturing respective illuminated scene image $\{I_1^H, ..., I_N^H\}$ from the camera view, allows to compute an encoding

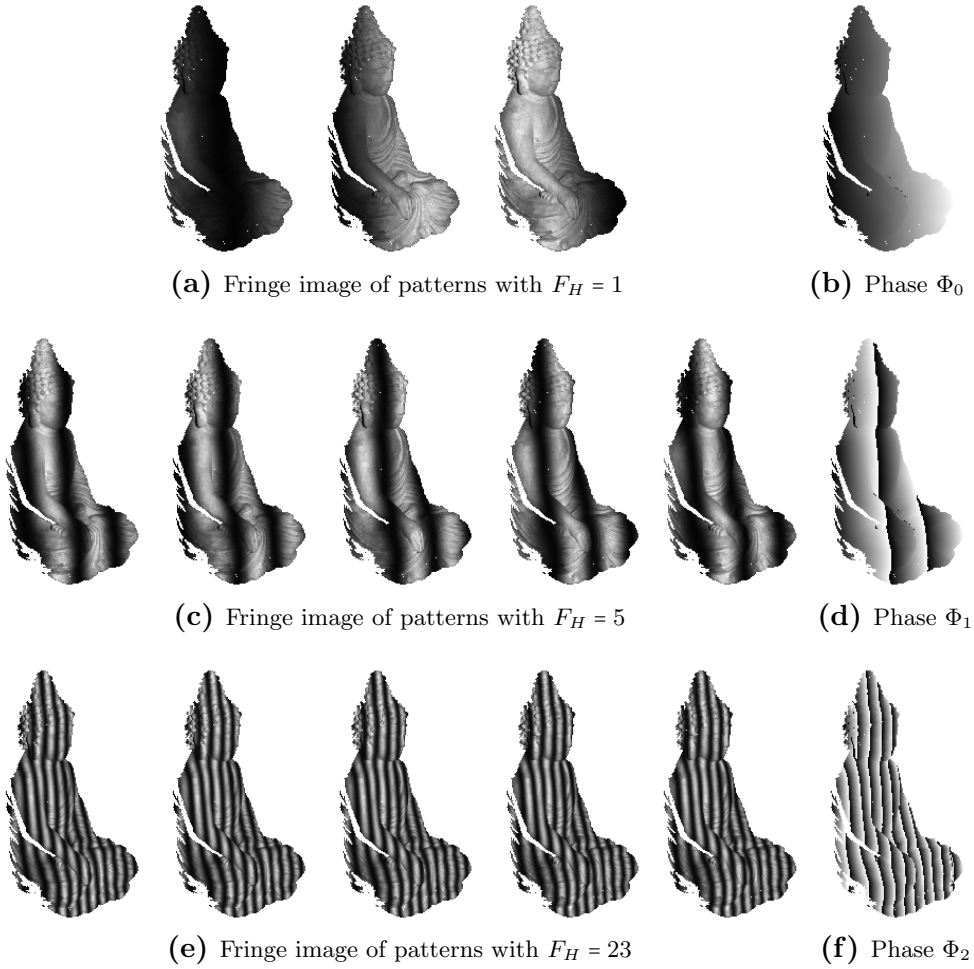
of the scene by application of the *harmonic addition theorem* (see [126]). It states that any superposition of cosines with same phase is again a cosine with the same phase:

$$\sum_{n=1}^{N} I_n^H \cos(\delta_n) = A\cos(\Phi) \tag{2.10}$$

$$\text{with} \quad \Phi = \text{atan2}\left(\sum I_n^H \sin(\delta_n),\ \sum I_n^H \cos(\delta_n)\right) \tag{2.11}$$

$$\text{and} \quad A^2 = \sum_{n=1}^{N}\sum_{m=1}^{N} I_n^H I_m^H \cos(\delta_n - \delta_m)\ , \tag{2.12}$$

where δ_n, δ_m denote the equidistant phase shifts, Φ the phase to be recovered and A the amplitude of the superposition. atan2 denotes the two-dimensional arcustangens function taking into account the quadrants of the input.



(a) Fringe image of patterns with $F_H = 1$ **(b)** Phase Φ_0

(c) Fringe image of patterns with $F_H = 5$ **(d)** Phase Φ_1

(e) Fringe image of patterns with $F_H = 23$ **(f)** Phase Φ_2

Figure 2.8.: *Several horizontally shifted fringe images of different frequencies (a,c,d) and the computed horizontal phase images of the respective levels that encode the scene from left to right (b,d,f).*

Phase image Φ of Equation (9.3), which is computed from the illuminated scene images I_n^H, encodes the object's surface from left to right by values in the range $[-\pi, \pi]$. The basic phase, computed from the first level of patterns with frequency $F_H = 1$ usually does not provide sufficiently precise encoding, due to limited dynamic range of the digital devices. Therefore, successively patterns with higher frequencies are projected. Unfortunately, the phase images calculated for patterns with higher frequencies contain several segments from $-\pi$ to π over the encoding range. Due to the limited range of values of atan2, these are each interrupted by wraps (jumps from π to $-\pi$). Figure 2.8 shows the captured fringe images of three levels with increasing frequencies. Thereby Φ_0 is the basic phase with frequency $F_H = 1$ that encodes the surface from left to right without any wraps. Higher phases Φ_1 and Φ_2 need to be unwrapped in order to get a continuous encoding. Once the wraps are removed, the accuracy of the encoding multiplies in each level. A simple unwrapping method that allows a stable and pixel-wise procedure will be shown later on in Chapter 9.

Such an encoded surface can be used together with the epipolar constraint from Section 2.3.1 to generate densely and highly accurate correspondences on the scene's surface. Thereby, along the epipolar line, the corresponding point is determined on the basis of the absolute phase value. This phase value theoretically equals the value in the other image where the projector holds the theoretically perfect phase.

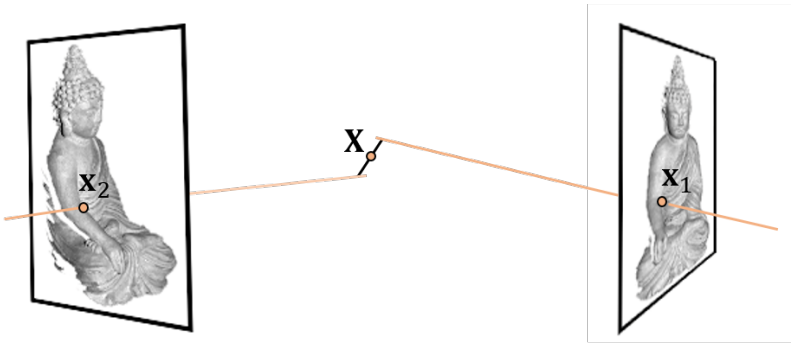


Figure 2.9.: *From calibrated cameras, the 3D position of a scene point can be estimated by triangulating the corresponding image points. The optimal 3D location has minimal distance to the out-projected rays.*

2.4. Triangulation

If matching pixels in two or more views are given and the calibrations of the cameras are known, the 3D position of the associated scene point can be determined. Out-projecting the rays of corresponding points with known camera matrices of the views, the 3D position with smallest distance to the rays is assumed to be the optimal estimate. Figure 2.9 visualizes the process of triangulation for depth estimation. In order to formulate this algebraically, let

two corresponding image points $\mathbf{x}_1$ and $\mathbf{x}_2$ be given. Furthermore the camera matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ of the two views are known. These are assumed to model the projection of the searched 3D scene point $\mathbf{X}$ in the world coordinate system onto the corresponding image points $\mathbf{x}_1$ and $\mathbf{x}_2$, as described in Section 2.2.3.

Let us set up an optimization problem that minimizes the projection error scaling invariant in homogeneous space, similar to the one in (2.6):

$$\underset{\tilde{\mathbf{X}}\in\mathbb{R}^4}{\operatorname{argmin}} \sum_{i=1}^{2} \|\mathbf{P}_i\tilde{\mathbf{X}} \times \tilde{\mathbf{x}}_i\|_2^2 , \qquad \text{s.t. } \|\tilde{\mathbf{X}}\|_2 = 1 \tag{2.13}$$

This can be reformulated into a simple homogeneous minimization problem using the cross product matrix $[\cdot]_\times$:

$$\underset{\tilde{\mathbf{X}}\in\mathbb{R}^4}{\operatorname{argmin}} \sum_{i=1}^{2} \|\mathbf{P}_i\tilde{\mathbf{X}} \times \tilde{\mathbf{x}}_i\|_2^2 , \qquad \text{s.t. } \|\tilde{\mathbf{X}}\|_2 = 1 \tag{2.14}$$

$$= \underset{\tilde{\mathbf{X}}\in\mathbb{R}^4}{\operatorname{argmin}} \sum_{i=1}^{2} \|[\tilde{\mathbf{x}}_i]_\times\mathbf{P}_i\tilde{\mathbf{X}}\|_2^2 , \qquad \text{s.t. } \|\tilde{\mathbf{X}}\|_2 = 1 \tag{2.15}$$

$$= \underset{\tilde{\mathbf{X}}\in\mathbb{R}^4}{\operatorname{argmin}} \Bigg\| \underbrace{\begin{pmatrix} [\tilde{\mathbf{x}}_1]_\times\mathbf{P}_1 \\ [\tilde{\mathbf{x}}_2]_\times\mathbf{P}_2 \end{pmatrix}}_{=:\mathbf{A}} \tilde{\mathbf{X}} \Bigg\|_2^2 , \qquad \text{s.t. } \|\tilde{\mathbf{X}}\|_2 = 1 \tag{2.16}$$

This system can be solved again by *singular value decomposition (SVD)*. Let $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}$ be the SVD of $\mathbf{A}$. Then the minimizer of problem (2.16) is given by the eigenvector (given by columns of $\mathbf{V}$), with respect to the smallest eigenvalue that is larger than zero (i.e. not in the nullspace of $\mathbf{A}$).

In a last step, the reconstructed 3D point in homogeneous coordinates is transformed back into spatial coordinates. Therefore, $\mathbf{X}$ is obtained by dividing and subsequently omitting the homogeneous coordinate of vector $\tilde{\mathbf{X}}$.

Note that this procedure can be extended to any number of devices by adding more rows with the measurements of the additional cameras in matrix $\mathbf{A}$. Additional views can improve the depth estimation. However, it should be noted that searching for correspondences across many devices often results in fewer correspondences, since each view introduces its own occlusions. The choice of the used number of devices should therefore be considered carefully.

Chapter 3

Pipeline and Setup

Contents

This chapter outlines the basic steps that the presented 3D reconstruction pipeline will go through. These steps have been chosen so that the procedure is widely applicable due to its active nature, but at the same time remains flexible and unlimited due to an auto-calibration of the complete setup. By using a structured light approach that includes a customer projector, we rely on a method that allows to auto-calibrate both the passive devices (cameras) and the active device (projector), which distinguishes it from other active methods using highly engineered pre-calibrated devices.

Although the application of auto-calibration from point correspondences works without complicated user interactions or calibration targets, there are nevertheless certain circumstances that can facilitate success of the procedure.

In the case of a pre-defined fully-automatic 3D scanner, optimal conditions can be created from which the process can benefit. It will therefore be shown how such an environment can be chosen to speed up the acquisition of a complete object from all sides with a turntable and to be able to perform an automatic separation of the object to be reconstructed from the undesired background. For completeness, the setup and hardware used for most of the data acquired in this work will also be presented here.

3.1. General Reconstruction Pipeline

Basically, the implementation of a complete reconstruction of an object is composed of two main independent steps.

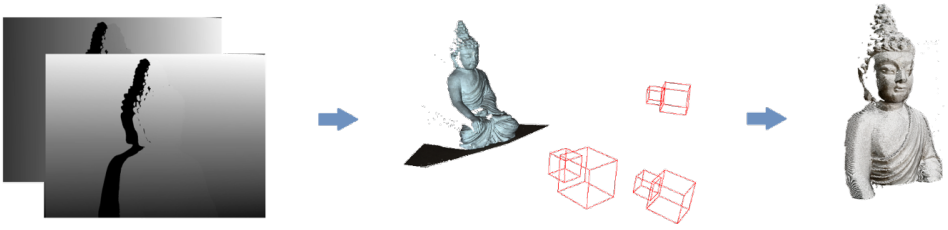
1. The acquisition, calibration and reconstruction of each partial view (Figure 3.1 (a)). This independence between the views allows different scene-adjusted camera settings for each view.

2. The alignment of the partial views to a complete point cloud of the object, the calculation of the normal vectors, the meshing of the surface and its texturing for an appropriate realistic representation of the generated 3D model (Figure 3.1 (b)).

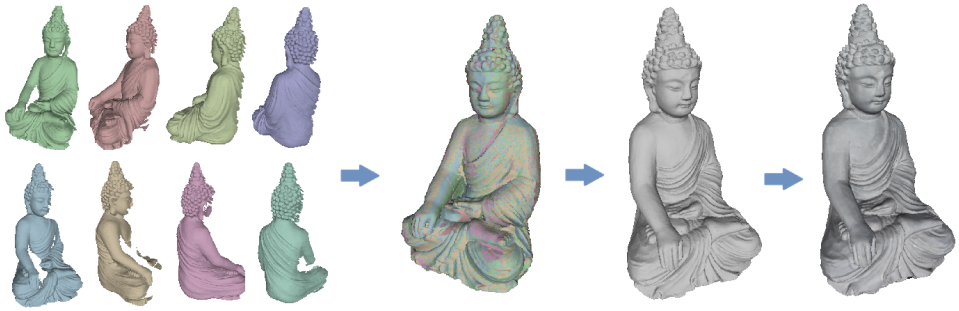
Acquisition, Calibration and Partial Reconstruction For flexibility and to avoid being dependent on object features, the method uses an approach based on the sinusoidal structured light technique described in Section 2.3.2. Thereby, the encoding strategy, using the phase-shift algorithm, is performed in both horizontal and vertical directions. The given two-dimensional encoding of each scene point, given by a horizontal and a vertical phase value, allows for establishing matches between the views without prior knowledge on the calibration of the used devices. Unlike the standard approach, the epipolar constraint is not needed and replaced by the second phase in independent direction. In Chapter 4, it will be shown how phase images generated in this way can be used to create highly accurate and dense point correspondences with sub-pixel accuracy between all included devices.

For the structured light encoding, good experiences have been achieved using pattern sets consisting of four levels each with $\{N_0 = 3, N_1 = 3, N_2 = 5, N_3 = 11\}$ shifts for both the horizontal and vertical pattern sets.
Good choices for the frequencies of the levels that also take into account the resolution and aspect ratio of a standard high-definition projector image are $\{F_H^{(0)} = 1, F_H^{(1)} = 5, F_H^{(2)} = 23, F_H^{(3)} = 91\}$ for horizontal encoding patterns and $\{F_V^{(0)} = 1, F_V^{(1)} = 2, F_V^{(2)} = 13, F_V^{(3)} = 52\}$ for the vertical ones. This choice is just a recommendation that has always served well in many investigations that have been carried out. More than satisfactory results have been achieved with it for a wide variety of scene types and object sizes. Besides that, however, there is also research, as recently by Mirdehghan *et al.* [117], which aims to optimize patterns depending on the scene.

With the help of this two-dimensional encoding approach, point correspondences between the devices used, i.e. cameras and video projectors, can be obtained without knowledge about their calibration. Conversely, however, the calibration of the devices with auto-calibration procedures can theoretically be calculated from these. A stable way, to do this for all devices will be presented in Chapter 5. In particular, it will be shown how the projector can be calibrated in addition to the cameras used. It has specific properties that cause standard auto-calibration procedures to fail in many cases. It will be shown how the projector can nevertheless make a valuable contribution to the overall calibration, especially with respect to determining the focal length of the cameras. Note that this approach, which puts matching before calibration, makes the procedure flexible and thus allows to arbitrarily adjust settings between different views, thus enabling better scene-specific reconstructions.



(a) Data acquisition, auto-calibration and reconstruction of one view.



(b) Alignment of partial views, meshing and texturing of full reconstruction.

Figure 3.1.: *Steps of the presented 3D reconstruction pipeline. While the substeps in (a) are performed individually for each partial view, the substeps in (b) merge the partial reconstructions to a complete model and compute an effective realistic visualization.*

From the calibration and dense point correspondences generated in this way, a 3D point cloud of the scene can finally be triangulated, as described earlier in Section 2.4. This step of generating partial reconstructions of different views will be performed independently multiple times in order to capture all sides of the object. Experience has shown that it is usually sufficient to record 8 positions in which either the object is rotated by 45° or conversely the recording setup is moved around the object. Figure 3.1 (a) visualizes the steps for an example scene.

Alignment and Representation of Full Model As outlined in Figure 3.1, the partial reconstructions must subsequently be merged into a complete model. How this can be done successfully and how the typically closed turn of partial reconstructions (where neighboring views usually overlap and the last scan catches up with the first one) can be exploited to enable stable convergence, will be shown in chapters 6 and 7. Subsequent post-processing steps such as calculating high-quality, crispy normal vectors for the point cloud, meshing the 3D structure in order to visualize the surface in a memory-efficient way, and high-resolution texturing of the point cloud will finally be addressed in Chapter 8.

(a) Scan-head

(b) Scanning environment

Figure 3.2.: *Scan-head (a) consisting of two industrial cameras for geometry estimation, a texture camera and a video projector for scene illumination. Advantageous scanning environment (b) with turntable for comfortable rotation of the object, dark diffuse material for automatic background masking and additional texture banderole for alignment.*

3.2. Specific Setup for Fully Automatic 3D Scanner

If the aim is to build a fully-automatic 3D scanner, some benefits can be generated from the creation of an optimal scanning environment. Although, it is important to mention that the approach outlined above can be applied to any scenario, a specific setup is explained, that enables to generate high-resolution models without any interaction, in greater detail.

As shown in Figure 3.2 (a), the scanner used to capture most of the data in this thesis, consists of a scan-head on which all included optical systems are fixed. Two monochrome industrial cameras (FLIR Blackfly S BFS-U3-200S6M) are mounted, which are used exclusively for geometry calculation. These have the advantage that they have a high frame rate, thus the projected patterns can be recorded very quickly. Moreover, experience has shown that monochrome cameras provide better reconstructions than RGB cameras whose color images pass through a Bayer filter. The scene is illuminated with a small consumer projector (Optoma ML 750e), that has a resolution of 1280×800 pixels, which has been shown to be sufficient. Later, correspondences with a higher resolution can be achieved by using an imaginary projector resolution, as will be shown in Chapter 4. Discrepancies in projector and camera resolutions can be compensated for by projector blurring, as shown by Taylor in [158]. This effect was observed to be sufficient in the applied system even without strong additional blurring of the projected patterns. In this chapter, it will also be explained, why having more than one camera in conjunction with the projector is beneficial for depth estimation and not redundant. To enable high quality textures on the created 3D models, an SLR camera (CANON EOS 5DS R) is additionally attached to the scan head, which finally allows to map crispy textures onto the created mesh. The acquisition time with this camera is significantly higher than with the geometry cameras. Therefore, we usually only take texture images with this camera, which we then

map to the geometry that was created with the geometry cameras. However, if the camera settings are adjusted to the scene, re-calibration is easily possible as described before.

How the calibration process can be further accelerated by reducing the number of acquisitions with such low frame-rate devices will be shown later in Chapter 9.

To quickly capture data of the object from all sides, it is placed on a turntable in an advantageously designed room/box as shown in Figure 3.2 (b). This allows the object to be rotated comfortably while the scan head remains static. This eliminates the need for re-calibration in each view and only requires it after camera setting adjustments. The background of the box is designed in a particularly dark and diffuse black. This reflects particularly little light and makes the patterns difficult to be recognized in corresponding camera views. Thus, only inconsistent phase values are determined here, which makes them easy to detect and to mask afterwards. In this way masking of the unwanted background and the object to be reconstructed is implicitly achieved. Furthermore, a texture banderole is attached to the edge of the turntable. This can be used to roughly align the various partial reconstructions. This is merely an aid to make the alignment more convenient.

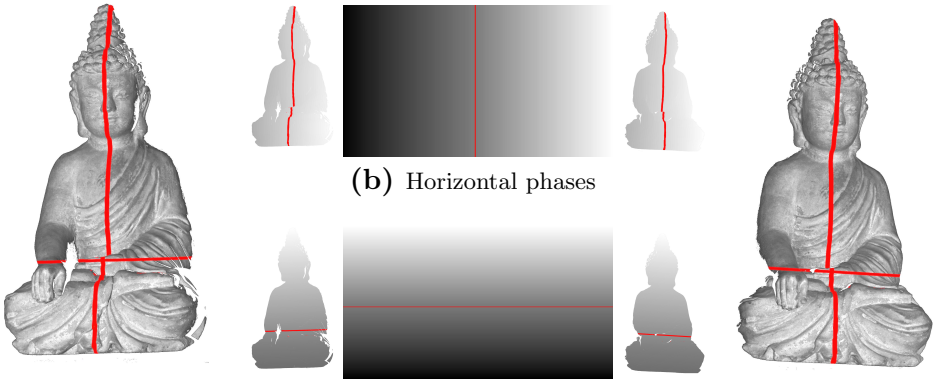
Chapter 4

Consistent Sub-Pixel Matching

Contents

4.1. Introduction

The structured light approach enables the determination of precise and dense point correspondences between a camera and a projector view. For general calibration-independent surface encoding, as introduced in Chapter 3, multiple sinusoidal patterns are projected to encode the scene in two directions. With the help of the deformed patterns, horizontal and vertical phase images are calculated for each camera view, that theoretically lead to a direct correspondence between each projector pixel and its position in the camera image. From these point correspondences, cameras and projector can be calibrated (see Chapter 5) and a dense point cloud can be triangulated using the obtained camera matrices.



(a) Camera 1 **(b)** Horizontal phases **(c)** Vertical phases **(d)** Camera 2

Figure 4.1.: *Illustration of projector-driven matching of two cameras (left and right) and a projector (middle). Red lines visualize the encoding of a point by its horizontal and vertical phase values. The optimal match is given by the intersection of these lines.*

The optimal matches between images with respect to an encoded surface point are usually not on pixel but on sub-pixel level. Common matching techniques that look for pixel-to-pixel correspondences between camera and projector often lead to noisy results that must be subsequently smoothed. The method presented here allows to find optimal sub-pixel positions for each projector pixel in a single pass and thus requires minimal computational effort. For this purpose, the quadrilateral regions containing the sub-pixels are extracted. The convexity of these quads and their consistency in terms of topological properties can be guaranteed during runtime. Subsequently, an explicit formulation of the optimal sub-pixel position within each quad is derived, using bilinear interpolation, and the permanent existence of a valid solution is proven. In this way, an easy-to-use procedure arises that matches any number of cameras in a structured light setup with high accuracy and low complexity. Due to the ensured topological properties, exceptionally smooth, highly precise, uniformly sampled matches with almost no outliers are achieved. The point correspondences obtained do not only have an enormously positive effect on the accuracy of reconstructed point clouds and resulting meshes, but are also extremely valuable for auto-calibrations calculated from them.

In theory, a setup consisting of a projector as active device, holding the perfect phase, and a camera is sufficient for depth estimation. However, in many practical arrangements, several cameras, at least two, are used in addition to the projector. This is due to a much cleaner projective behavior of high quality cameras compared to most projectors:

- Cameras usually cause less distortion (radial and especially tangential distortions) than most projectors, since higher quality lenses are available.
- Auto-calibration directly from point correspondences is more stable to achieve with high-quality cameras due to constrained principal points.

- Most industrial cameras allow gamma correction to be disabled, which has a significant impact on assumptions in computer vision applications. Since this is not possible with affordable projectors, it is of considerable advantage to triangulate the final point cloud with the camera information only.

In order to cover the general case of any number of cameras, in this chapter the situation with two cameras and one projector is considered. Thus, the procedure can be trivially extended to an arbitrary number of cameras. Also the simplest case with one camera and a projector requires no special treatment and can be matched by the presented approach.

The idea of projector-driven matching is to find suitable correspondences in the camera images for each projector pixel. In this way, all the camera views are transitively matched via the projector pixels. Figure 4.1 illustrates this procedure. (a) and (d) show the texture images of the two camera views. (b) and (c) show the corresponding horizontal and vertical phase images of the cameras and in the center of the projector. The red lines illustrate the unique encoding process of a pixel through the two phases. This method appears to be simple and to create dense point correspondences in a trivial way. However, a number of difficulties arise during the exact execution, which can often lead to problems and are comprehensibly solved in the following:

- Phase images are discrete samples of continuous camera and projector phases. Therefore, there is usually no exact pixel-to-pixel mapping. Instead, it is very likely that the match of a pixel in the projector image lies between certain camera pixels.

- The topology of the pixels remains locally preserved during the projection process. In simple terms, this means that a point to the left of another point on the object surface is also to the left of this point in the projected camera image. Thus, certain conditions can be defined which must be fulfilled by the phases and met during the matching process in order to avoid noisy results.

- Matching is only a sub-step in 3D reconstruction and auto-calibration and should therefore be fast. The trivial procedure of searching for the optimal match for each pixel of each image in each of the other images is not practical at all. The procedure would be of quadratic complexity in terms of resolution, and with increasing camera resolutions this is very poor.

In the following, a procedure is developed that is extremely fast and can match any number of devices stably and consistently with sub-pixel accuracy. Each image pixel must be passed through exactly once resulting in a linear complexity. The consistency conditions are enforced during run-time, thus avoiding mismatches already during execution.

4.2. Related Work

Matching is one of the main components in the field of 3D reconstruction. The goal is to find point correspondences as dense and precise as possible across the entire scene. Standard approaches search for suitable candidates along the epipolar lines and evaluate them according to their neighbors using suitable region descriptors ([15], [180], [66], [16]). This is a common approach, but requires a calibrated setup and can fail in many cases, as in uniform areas of the scene. If it is, in contrast, possible to create very precise matches without prior calibration information auto-calibration methods allow to perform an exact calibration of the system directly from these matches, which makes such a computer vision system much more user-friendly and flexible. It also makes it suitable for a variety of other applications where pre-calibration is not possible, extremely tedious or problematic, since the setup may de-calibrate over time.

Common matching procedures without pre-calibration are based on transformation invariant features, such as introduced by Lowe [106] and further applied by Hu [70], which provide robust matches if sufficient object texture is available. Also, there are methods that do not only include appearance but also object geometry into the search as shown by Isack and Boykov in [77]. However, all of them most likely fail in the case of very smooth uniform objects, which limits the applicability. In order to enable the reconstruction of un-textured objects, the structured light approach is a common tool. In [109] Ma *et al.* use structured light information to handle large disparities in binocular matching. Similarly, in [134] Pribanic *et al.* use the wrapped phase to refine the stereo matches. In [98] Liu *et al.* use binary patterns to reduce the search range and thus speed up the matching. Scharstein and Szeliski [145] show how to get accurate dense matches using only the reconstructed phase. In [34] Asmi and Roy introduce a sub-pixel matching for un-synchronized structured light, while for each match an energy is minimized by gradient descent. Matching based on peak calculation as introduced by Donate *et al.* [28] and Xie *et al.* [173] also achieve sub-pixel accuracy but require higher computational effort than the method presented here.

In [32] Du *et al.* proposed a deep learning approach for structured light matching recently. It uses a Siamese network trained on a synthetic data set that expects rectified images, which is not suitable for arbitrary un-calibrated systems and auto-calibration. Also in [95] Li *et al.* combine structured light and deep learning to achieve good and exact matches. In [141] Ryan Fanello *et al.* presents a method that even skips matching and directly calculates depth using deep learning.

4.3. Fast Projector Driven Matching (FPDM)

The task of fast projector-driven matching is to find corresponding positions in the camera phases for each integer projector pixel. Since this is usually not again an integer position, it must be estimated with sub-pixel accuracy. Figure 4.2 (a) illustrates this for the projector in the middle and camera images left and right. Everything at the sub-pixel level can only be described by the pixels in its environment, since no finer information is available in an image. In order to compute the sub-pixel matches, it is therefore necessary to find integer camera pixels that span a quadrilateral (Figure 4.2, green pixels) that encloses the optimal sub-pixel match as closely as possible. From this quad, the sub-pixel match can then be interpolated in a subsequent step. The quadrilateral does not necessarily have to be square or rectangular, but should at least be convex. This constraint is fulfilled in the general case and only violated at regions with depth discontinuities. It ensures that the enclosed area can be described smoothly through its corners. In addition, there are certain consistency characteristics that should be met.

4.3.1. Matching Integer Pixel Quads

In a first step, best possible convex quads, enclosing the sub-pixel match for each projector pixel, are found in each camera image. Each camera pixel should only be processed once in order to maintain linear complexity. The chosen vertices should fulfill several properties, which will be implemented step by step. Therefore, for each projector pixel, four corner points will be stored whose quadrilateral contains the optimal camera correspondence as shown in Figure 4.2. An array four times the size of the projector resolution is needed as a buffer. Note that the projector image can be selected in any resolution as it is completely imaginary. The resulting density of the point cloud can be precisely controlled in this way. Practice has shown that the projector resolution should be selected in approximately the same order as the camera resolutions, since usually both cover about the same area of the scene.

In the following, horizontal and vertical phase images $\mathbf{\Phi}_H$ and $\mathbf{\Phi}_V$ of a camera with values in the interval $[0, 1]$ are assumed. The phases run from left to right and from bottom to top according to the common coordinate axes. Similarly, the optimal projector phases run from 0 to 1 at a selected resolution

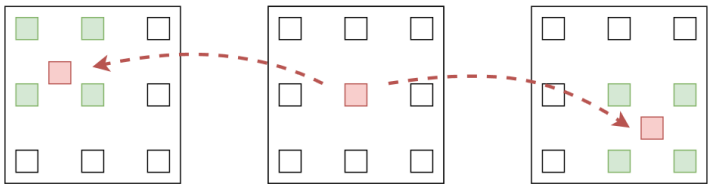


Figure 4.2.: *Visualization of optimal sub-pixel matches (red) between projector (center) and two cameras (left and right). The surrounding integer pixels of the sub-pixel matches, are marked in green.*

(w_P, h_P). This is depicted in Figure 4.1.

For each camera pixel (x, y), the theoretical corresponding position in the projector image is uniquely given by the vertical and horizontal phase values $\mathbf{\Phi}_H(x, y)$ and $\mathbf{\Phi}_V(x, y)$. Therefore, a camera pixel (x, y) would theoretically match projector pixel

$$(\hat{x}_P, \hat{y}_P) = \big(\mathbf{\Phi}_H(x, y)w_P, \mathbf{\Phi}_V(x, y)h_\mathrm{P}\big), \tag{4.1}$$

which is not an integer value, as sought. Nevertheless, it is an approximate position and likely a lower and upper corner of the next integer projector pixels, which is the basic idea of the presented fast (linear) method.

For each integer projector pixel (x_P, y_P) the vertices of the spanned matching quad in the camera image are noted as indicated in Figure 4.6 (right). So (x_{00}, y_{00}), (x_{10}, y_{10}), (x_{01}, y_{01}) and (x_{11}, y_{11}) denote the bottom left, the bottom right, top left and top right corner pixels of the quad around sub-pixel match $(\hat{x}, \hat{y})$ in the camera image with respect to the integer projector pixel (x_P, y_P). Using the notations $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ for *floor* and *ceil* integer rounding, a camera pixel (x, y) would be a feasible corner point of four adjacent quadrilaterals containing sub-pixel camera matches with respect to four projector pixels. Thereby, it would be exactly one bottom left, one bottom right, one top left and one top right corner of the four corresponding quadrilaterals. The buffers for the projector pixels are filled by traversing the image and assigning each image pixel as:

$$(x, y) \longrightarrow \begin{matrix} \big(\lceil\hat{x}_P\rceil_{00}, \lceil\hat{y}_P\rceil_{00}\big) \\ \big(\lfloor\hat{x}_P\rfloor_{10}, \lceil\hat{y}_P\rceil_{10}\big) \\ \big(\lceil\hat{x}_P\rceil_{01}, \lfloor\hat{y}_P\rfloor_{01}\big) \\ \big(\lfloor\hat{x}_P\rfloor_{11}, \lfloor\hat{y}_P\rfloor_{11}\big) \end{matrix} \tag{4.2}$$

Since phases in arbitrary real scenes are usually sampled non-uniformly, it may be possible that several camera pixels are feasible corner points of a specific quadrilateral. Using the example of a lower left corner point, the quality of the corner point can be calculated by its distance to the optimal sub-pixel value:

$$E = \left|\hat{x}_P - \lceil\hat{x}_P\rceil_{00}\right| + \left|\hat{y}_P - \lceil\hat{y}_P\rceil_{00}\right| \tag{4.3}$$

If a corner point is already occupied when running through the image, it is only replaced if this error is less than that of the previously stored pixel. This ensures that the enclosing quadrilateral becomes minimal.

4.3.2. Topological Consistency Check (TCC)

An important property of a projection is that the topology of the projected points is locally preserved. Therefore, also surface points that have been encoded using structured light must remain consistent in the corresponding phase images. Theoretically, there are a few situations at borders of objects and very different viewing angles where this property may be violated. However, by considering local neighborhoods, these exceptional cases can be excluded. Some

test strategies are introduced in the following, that enforce the topology preservation property. Most importantly, they remain valid for non-minimal quads, allowing their application on non-final temporal stores of corner points. This way, incorrect and noisy phase values are excluded from matching, resulting in smoother and more accurate matches with way less outliers. Advantages can be seen both in applied auto-calibration and in triangulated point clouds.

Before saving any image pixel to a corner point (x_{00}, y_{00}), (x_{10}, y_{10}), (x_{01}, y_{01}) or (x_{11}, y_{11}) with respect to a projector pixel (x_P, y_P), it is ensured that a lower left pixel in the camera phase is also a lower left pixel in the projector phase and so on. In this way, many faulty matches can be detected and avoided. Moreover, it ensures that the resulting quads are convex. The following simple checks have to be fulfilled:

$$\begin{array}{ccc} (x_{01}, y_{01}) & \xrightarrow{x_{01} \leq x_{11}} & (x_{11}, y_{11}) \\ \uparrow{\scriptstyle y_{00} \leq y_{01}} & & \uparrow{\scriptstyle y_{10} \leq y_{11}} \\ (x_{00}, y_{00}) & \xrightarrow{x_{00} \leq x_{10}} & (x_{10}, y_{10}) \end{array} \tag{4.4}$$

The tests are applied to the pixels during the storing process while looping through the images. Naturally, therefore, during the storing process, one vertex is checked for consistency with respect to other vertices that are not final and that may be part of non-minimal representations of a quad around a sub-pixel match. As already mentioned these tests are also valid for non-minimal quads as long as they do not represent severe outliers, which moreover would simply lead to finding no match for the projector rather than an outlier. Figure 4.3 (a) illustrates an update of a corner point to a closer representation. It is easy to see that the convexity properties are fulfilled throughout by all points, while converging to the minimal representation.

Diagonal Check for Weak Quads (DCW) Theoretically, the quadrilaterals can take a wide variety of shapes and still satisfy the desired topology and convexity. But the more unusual the shape, the worse its content is determined by bilinear interpolation. Practice has shown that it is optimal if the vertices span a square, but this is not absolutely necessary. An additional optional test avoids unnatural quads by checking the diagonal values:

$$|x_{00} - x_{11}| + |y_{00} - y_{11}| < \tau, \qquad |x_{10} - x_{01}| + |y_{10} - y_{01}| < \tau \tag{4.5}$$

The quads should not be of arbitrary size just because they might theoretically be feasible. Usually they will not provide an accurate measurement if the corners are above a certain distance, which can be generously set to $\tau = 5$ pixels for most applications.

For illustration, Figure 4.3 (b) shows an example of an unfavorable quad that would be removed. Note that this check should only be done after the entire quad matching, otherwise some quads may be removed due to non-minimal representations that may have improved over time.

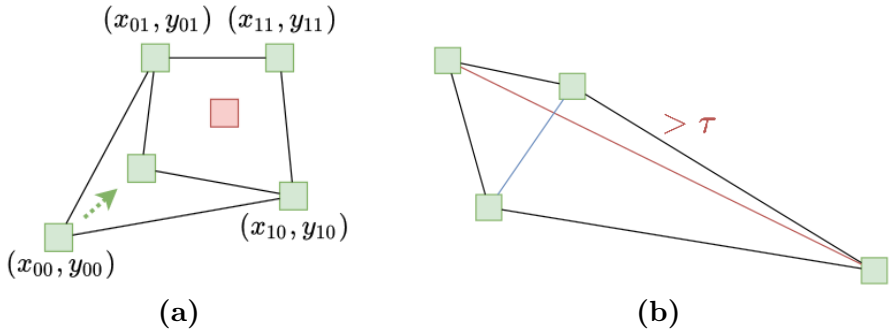


Figure 4.3.: *(a) Example of a corner point update, where the consistency properties stay fulfilled. (b) Example of a unfavorable quad, that would be removed by the diagonal check (4.5).*

Epipolar Consistency Check (ECC) In many practical scenarios a rough calibration of the setup is already available. This can be extremely advantageous and easily involved into the scheme. In this case a camera point (x, y) should only be mapped to a corner point of projector pixel (x_P, y_P) if the symmetric epipolar error is below a certain threshold σ, which can also be set generously, like $\sigma = 10$ pixels to fit almost all scenarios. Using the homogeneous point representations $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}_P$, the epipolar lines in the respective other image can be calculated in homogeneous representation by applying the fundamental matrix $\mathbf{F}$ and its transpose. The symmetric Euclidean epipolar error, can then be easily checked using the following formula:

$$\frac{\tilde{\mathbf{x}}\ \mathbf{F}\ \tilde{\mathbf{x}}_P^\mathsf{T}}{\sqrt{(\mathbf{l})_1^2 + (\mathbf{l})_2^2}} + \frac{\tilde{\mathbf{x}}\ \mathbf{F}^\mathsf{T}\ \tilde{\mathbf{x}}_P^\mathsf{T}}{\sqrt{(\mathbf{l}_P)_1^2 + (\mathbf{l}_P)_2^2}} < \sigma \tag{4.6}$$

Thereby $(\mathbf{l})_1$, $(\mathbf{l}_P)_1$ and $(\mathbf{l})_2$, $(\mathbf{l}_P)_2$ denote the first and second entries of the respective epipolar lines $\mathbf{l} = \mathbf{F}\ \tilde{\mathbf{x}}_P$ and $\mathbf{l}_P = \mathbf{F}^\mathsf{T}\tilde{\mathbf{x}}$.

In order to illustrate the effect of the checks on calculated matches, Figure 4.4 shows the resulting point cloud of *FPDM* without (a) and with *TCC* (b). There are significantly fewer outliers, resulting in less flying points. (c) shows how the matches can be further improved by *ECC* by avoiding faulty assignments, especially in discontinuities of the scene. False matches can also occur due to incorrect but permissible phase regions, caused by (inter-)reflections.

4.4. Bilinear Sub-Pixel Matching

After the quad matching, for each permissible projector pixel a consistent convex quadrilateral is given per camera image. Under certain assumptions it is possible to determine the sub-pixel position of the optimal match from the corners of the quad and their phase values. The optimal sub-pixel position is calculated in the unit patch using bilinear interpolation assumption and then mapped to the convex region as shown in Figure 4.6.

(a) FPDM **(b)** FPDM+TCC+DCW **(c)** FPDM+TCC+DCW +ECC

Figure 4.4.: *Resulting point clouds of FPDM applied to the exemplary scene with and without Topological Consistency Check (TCC) (4.4), Diagonal Check for Weak Quads (DCW) (4.5) and Epipolar Consistency Check (ECC) (4.6).*

4.4.1. Sub-Pixel Position in Unit Patch

Given a unitary patch with horizontal phase values ϕ_{H00}, ϕ_{H10}, ϕ_{H01} and ϕ_{H11} of the corner points as depicted in Figure 4.6, the bilinear interpolated value $\phi_H(\tilde{x}, \tilde{y})$ for any position $(\tilde{x}, \tilde{y}) \in [0,1]^2$ is given by

$$\phi_H(\tilde{x}, \tilde{y}) = a_0 + a_1\tilde{x} + a_2\tilde{y} + a_3\tilde{x}\tilde{y} \tag{4.7}$$

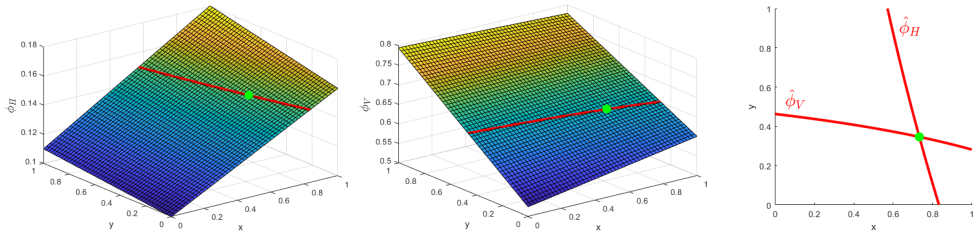
$$\text{with coefficients} \quad \begin{aligned} a_0 &= \phi_{H00} \\ a_1 &= \phi_{H10} - \phi_{H00} \\ a_2 &= \phi_{H01} - \phi_{H00} \\ a_3 &= \phi_{H11} + \phi_{H00} - \phi_{H10} - \phi_{H01} \end{aligned}$$

and analogously for the vertical phase by

$$\phi_V(\tilde{x}, \tilde{y}) = b_0 + b_1\tilde{x} + b_2\tilde{y} + b_3\tilde{x}\tilde{y} \tag{4.8}$$

$$\text{with coefficients} \quad \begin{aligned} b_0 &= \phi_{V00} \\ b_1 &= \phi_{V10} - \phi_{V00} \\ b_2 &= \phi_{V01} - \phi_{V00} \\ b_3 &= \phi_{V11} + \phi_{V00} - \phi_{V10} - \phi_{V01} \end{aligned}$$

Figure 4.5 (a) illustrates how two bilinearly interpolated phases on the unit patch can look like. The task is to find the optimal sub-pixel position inside the patch meeting the phase values $(\hat{\phi}_H, \hat{\phi}_V)$. The patch that interpolates the horizontal phase values defines a two-dimensional curve on which the value $\hat{\phi}_H$ is assumed. The same applies to the patch of the vertical phase values,



(a) Horizontal (left) and vertical (right) phases in unit patch **(b)** Top view

Figure 4.5.: *Unit patch with interpolated phases (a). Red curves are possible solutions, where the interpolated patch meets the sought phase value. The intersection of the curves (green) solves the problem.*

which describes a curve for $\hat{\phi}_V$. Such curves are visualized by red lines in Figure 4.5 (a). The point where the curves intersect within the patch is the optimal position of the sought sub-pixel match. Figure 4.5 (b) shows the top view of the unit patch and the two curves defined by the phase values. The intersection is marked by a green dot, which is also plotted in (a).

In order to find optimal positions $\tilde{x} \in [0,1]$ and $\tilde{y} \in [0,1]$ at which the bilinear interpolated patches meet the sought values $\hat{\phi}_H$ and $\hat{\phi}_V$ the following system of equations is solved:

$$\begin{aligned} \hat{\phi}_H &= a_0 + a_1\tilde{x} + a_2\tilde{y} + a_3\tilde{x}\tilde{y} \\ \hat{\phi}_V &= b_0 + b_1\tilde{x} + b_2\tilde{y} + b_3\tilde{x}\tilde{y} \end{aligned} \tag{4.9}$$

Solving for $\tilde{x}$ results in the simple quadratic equation

$$u\tilde{x}^2 + v\tilde{x} + w = 0 \tag{4.10}$$

$$\text{with coefficients} \quad \begin{aligned} u &= b_1a_3 - b_3a_1 \\ v &= b_1a_2 + (b_0 - \hat{\phi}_V)a_3 - b_2a_1 - b_3(a_0 - \hat{\phi}_H) \\ w &= (b_0 - \hat{\phi}_V)a_2 - b_2(a_0 - \hat{\phi}_H) \end{aligned}$$

which can be explicitly solved by

$$\tilde{x} = \begin{cases} -\frac{v}{2u} \pm \sqrt{\left(\frac{v}{2u}\right)^2 - \frac{w}{u}} & , \quad u \neq 0 \\ -\frac{w}{v} & , \quad u = 0 \end{cases}. \tag{4.11}$$

The vertical position $\tilde{y}$ is directly computed from one of the equations in (4.9). Note that the properties of the quads received in Section 4.3 ensure the existence of intersection within each patch.

Existence of Solution The interpolated value $\hat{\phi}_H$ is by construction achieved inside the patch and moreover the following holds true due to the consistency checks:

$$\phi_{H00}, \phi_{H01} \leq \hat{\phi}_H \leq \phi_{H10}, \phi_{H11} \tag{4.12}$$

Of course for any convex combination with $\tilde{y} \in [0,1]$, we also have:

$$(1-\tilde{y})\phi_{H00} + \tilde{y}\phi_{H01} \leq \hat{\phi}_H \leq (1-\tilde{y})\phi_{H10} + \tilde{y}\phi_{H11} \tag{4.13}$$

Therefore the curve, defined by the first equation of (4.9), that maps feasible positions $\tilde{x}$ to any value $\tilde{y} \in [0,1]$ has the following property:

$$\tilde{x} = \frac{\hat{\phi}_H - a_0 - a_2\tilde{y}}{a_1 + a_3\tilde{y}} = \underbrace{\frac{\overbrace{\hat{\phi}_H - (1-\tilde{y})\phi_{H00} - \tilde{y}\phi_{H01}}^{=(i)\geq 0 \ (4.13),}}{(1-\tilde{y})\underbrace{(\phi_{H10} - \phi_{H00})}_{>0 \ (4.12)} + \tilde{y}\underbrace{(\phi_{H11} - \phi_{H01})}_{>0 \ (4.12)}}}_{=(ii)>0 \ (4.13)} \geq 0 \tag{4.14}$$

Thereby, the situation in which all corner points carry the same value is neglected. In this case division by zero would not be defined. Nevertheless, in this case an optimal integer pixel match exists and interpolation is not necessary.

Additionally, the denominator (ii) in this fraction is greater or equal than the numerator (i), which limits the fraction to 1:

$$(ii) - (i) = (1-\tilde{y})\phi_{H10} + \tilde{y}\phi_{H11} - \hat{\phi}_H \overset{(4.13)}{\geq} 0 \tag{4.15}$$

Proceeding similar for the vertical phase, the following properties are obtained for the curves of equations (4.9):

$$\tilde{x} = \frac{\hat{\phi}_H - a_0 - a_2\tilde{y}}{a_1 + a_3\tilde{y}} \in [0,1] \quad \text{for} \quad \tilde{y} \in [0,1] \tag{4.16}$$

$$\tilde{y} = \frac{\hat{\phi}_V - b_0 - b_1\tilde{x}}{b_2 + b_3\tilde{x}} \in [0,1] \quad \text{for} \quad \tilde{x} \in [0,1] \tag{4.17}$$

Therefore, the curves are defined for all $\tilde{y}, \tilde{x} \in [0,1]$ and map to valid values $\tilde{x}, \tilde{y} \in [0,1]$. This proves that the continuous curve (4.16) describes a continuous connection between the top and bottom of the patch that runs inside the patch ($\tilde{x} \in [0,1]$). Similarly, curve (4.17) is a continuous connection within the patch ($\tilde{y} \in [0,1]$) from the left side to the right side. These curves must therefore intersect at least once within the patch. This guarantees a solution, which can be explicitly computed by solving the resulting quadratic equation (4.9) and choosing the feasible solution inside the patch. Theoretically, it can happen that both $\phi_{H00} = \hat{\phi}_H = \phi_{H11}$ and $\phi_{V00} = \hat{\phi}_V = \phi_{V11}$ are satisfied. This case has practically no meaning but for completeness it is briefly mentioned. In this situation, both curves run from point $(0,0)$ to point $(1,1)$, so that two solutions exist. If in addition the other vertices take exactly such values, that both curves are exactly diagonal, there can even be an infinite number of solutions. These cases will usually not occur, but if they do, it is evidence of badly chosen quads, which often carry noisy data and do not contribute much information anyway. In this case, the middle of the patch $(\tilde{x}, \tilde{y}) = (0.5, 0.5)$ can simply be chosen as solution, or the patch can just be discarded. ■

4.4.2. Mapping to Convex Quad

In general, the corner points around a sub-pixel match will not span a square region. However, for convex quads, the method can be applied by assuming an additional bilinear interpolation scheme. With corresponding corner points in the image given by

$$\begin{aligned}(x_{01}, y_{01}) &\leftrightarrow (0,1) \qquad (x_{11}, y_{11}) \leftrightarrow (1,1)\\ (x_{00}, y_{00}) &\leftrightarrow (0,0) \qquad (x_{10}, y_{10}) \leftrightarrow (1,0)\end{aligned} \tag{4.18}$$

a point $(\tilde{x}, \tilde{y}) \in [0,1]^2$ in the unit square can be mapped to the convex quadrilateral by:

$$\begin{pmatrix}\hat{x}\\ \hat{y}\end{pmatrix} = \begin{pmatrix}x_{00} & x_{10} & x_{01} & x_{11}\\ y_{00} & y_{10} & y_{01} & y_{11}\end{pmatrix}\begin{pmatrix}1 & -1 & -1 & 1\\ 0 & 1 & 0 & -1\\ 0 & 0 & 1 & -1\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix}1\\ \tilde{x}\\ \tilde{y}\\ \tilde{x}\tilde{y}\end{pmatrix} \tag{4.19}$$

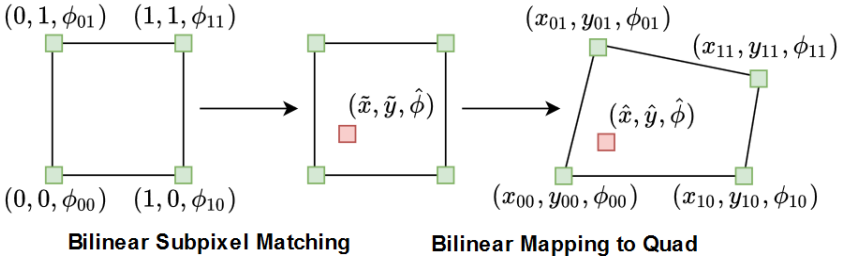
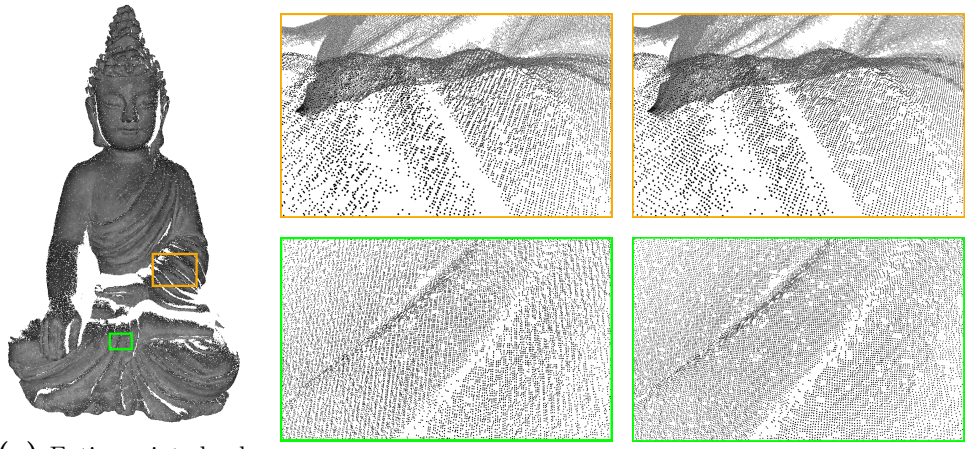


Figure 4.6.: *Procedure of bilinear sub-pixel matching. The position is computed in the unit patch and afterwards mapped to the convex quad.*

4.5. Results

Figure 4.8 shows the reconstructed point clouds of different scenes as a qualitative illustration of the reconstructions obtained. For each scene, the left reconstruction shows the result of the matches obtained with best-pixel correspondences. The right point cloud shows the result of the *Fast Projector Driven Consistent Sub-Pixel Matching (FPCSM)* presented in this chapter. For (a-d) the images on the far right show in addition the back-projections of the points onto the projector image, with in- and outliers marked in green and red. The reconstructions are significantly smoother and contain almost no outliers. Of particular note is the *Monkey* data, which was taken from a highly specular metallic brushed monkey statue, which clearly shows the influence and improvements of the consistency checks. Of course there are methods to smooth out noisy results and to remove flying points in a post processing, but the method presented here removes outliers during the matching process without any additional computational effort. Also, in contrast to smoothing, erroneous measures are removed and not smeared over the entire point cloud.



(a) Entire point cloud **(b)** Best-pixel matching **(c)** FPDM

Figure 4.7.: *Reconstructed point clouds using best-pixel matches (b) and FPCSM (c). The position of the enlarged areas in the global point cloud is depicted in (a). Due to the optimal sub-pixel matching, the surface is much more uniformly sampled and less noisy.*

Especially, if the correspondences are used for auto-calibration procedures this can be a huge advantage.

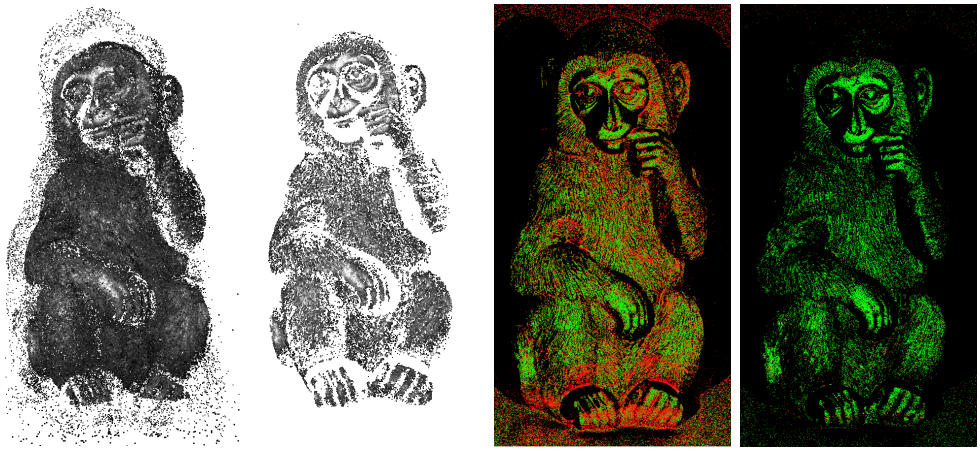
Figure 4.7 (b) and (c) shows the enlargement of two regions in the reconstructed *Buddha* statue (a). Due to the optimal sub-pixel matching, the surface is much more uniformly sampled and less noisy. Especially for subsequent meshing and precise depth measurement this may have a significant influence.

Finally, Table 4.1 shows the reduction of the median back-projection errors on the camera images from which they were triangulated. The median error was chosen to avoid over-weighting extreme outliers of the standard approach without consistency checks. Throughout, the error is improved in all data sets.

Data Set	**Best-Pixel Matches**		**FPCSM Matches**	
	Camera 1	Camera 2	Camera 1	Camera 2
Buddha	0.354478	0.354868	0.253260	0.254019
Bird	0.371168	0.372484	0.261994	0.261302
Totem	0.329969	0.338314	0.258966	0.261391
Monkey	0.378873	0.375521	0.277572	0.278189
Scene	0.267318	0.279251	0.170884	0.178666

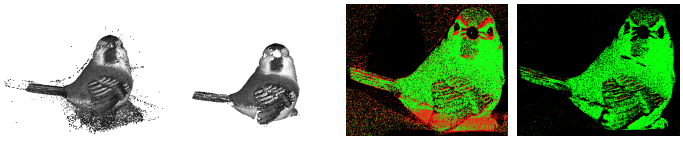
Table 4.1.: *Median back-projection errors of the evaluated data sets for best-pixel matching (left) and the proposed method (right).*

(a) Point clouds and projector views of *Buddha* dataset.

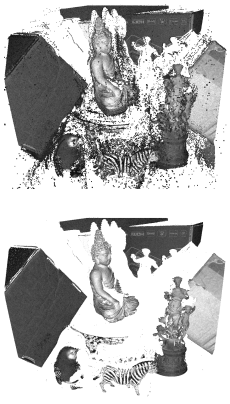


(b) Point clouds and projector views of *Monkey* dataset.

(c) Point clouds and projector views of *Totem* dataset.



(d) Point clouds and projector views of *Bird* dataset.



(f) Point clouds of *Scene* dataset.

Figure 4.8.: *Results of Fast Projector Driven Consistent Sub-Pixel Matching (FPCSM) (right) applied to exemplary scenes in comparison to point clouds obtained by standard best-pixel matching (left). Each row shows the point clouds and the back-projected points to the projector image with labeled in- and outliers.*

Algorithm 1: Fast Projector Driven Consistent Sub-Pixel Matching (FPCSM)

Input: Phase images $\boldsymbol{\Phi_H}$ and $\boldsymbol{\Phi_V}$ for each camera.

Create buffers B_{00}, B_{10}, B_{01} and B_{11} of size of the projector resolution (w_P, h_P).

for *each camera* **do**

  for *each camera pixel* (x, y) **do**

    Compute projector position $\hat{x}_P = \boldsymbol{\Phi}_H(x,y)w_P,\ \hat{y}_P = \boldsymbol{\Phi}_V(x,y)h_\mathrm{P}$.

    Where TCC (+ECC) is fulfilled and buffer is free or E decreases store:

$$
\begin{aligned}
(x,y) &\longrightarrow B_{00}(\lceil \hat{x}_P \rceil + k, \lceil \hat{y}_P \rceil + l) \\
(x,y) &\longrightarrow B_{10}(\lfloor \hat{x}_P \rfloor - k, \lceil \hat{y}_P \rceil + l) \\
(x,y) &\longrightarrow B_{01}(\lceil \hat{x}_P \rceil + k, \lfloor \hat{y}_P \rfloor - l) \\
(x,y) &\longrightarrow B_{11}(\lfloor \hat{x}_P \rfloor - k, \lfloor \hat{y}_P \rfloor - l) \\
&\qquad\qquad\qquad\qquad\qquad\qquad k,l = 0, ..., \lfloor \tau \rfloor
\end{aligned}
$$

  end

  Perform *Diagonal Check* (DCW) to remove weak quads.

  for *each projector pixel* (x_P, y_P) **do**

    Compute interpolation coefficients (4.7) and (4.8) from phase values:

$$
\begin{aligned}
\phi_{Hij} &= \boldsymbol{\Phi}_H\big(B_{ij}(x_P, y_P)\big), \\
\phi_{Vij} &= \boldsymbol{\Phi}_V\big(B_{ij}(x_P, y_P)\big), \quad (i,j) \in \{0,1\}^2
\end{aligned}
$$

    Using (9.21) with $\hat{\phi}_H = \frac{x_P}{w_P}$ and $\hat{\phi}_V = \frac{y_P}{h_P}$ delivers the optimal sub-pixel match $(\tilde{x}, \tilde{y})$ in the unit patch.

    Apply convex mapping (Sec. 4.2) to transform $(\tilde{x}, \tilde{y})$ to its position in the real quad in order to receive the final sub-pixel match $(\hat{x}, \hat{y})$.

  end

end

Remove all matches for a projector pixel if not for every camera a match was found.

Output: Optimal sub-pixel matches between all cameras, stored for every pixel of the projector.

4.6. Conclusions

In this chapter a matching strategy has been presented which generates high-precision correspondences for two-dimensional encodings of structured light systems with any number of cameras. The matches are estimated in sub-pixel accuracy. Therefore, an explicit formula has been derived, which provides matches under the assumption of bilinearly interpolated patches. The existence of such matches has been mathematically investigated and proven. An important contribution is that this is achieved with linear complexity, while simultaneously ensuring topological consistency over the views. This results in high quality matches with nearly no outliers, that are uniformly sampled over the scene. Overall, a method has been developed which reaches extremely high accuracy with extremely low (linear) computational effort. In order to achieve maximal reproducibility of the procedure, the individual steps are given in Algorithm 16 as pseudo-code.

Chapter 5

Auto-Calibration

Contents

5.1. Introduction

The task of calibration is to calculate the intrinsic and extrinsic camera parameters, that model the projection process of a scene to a captured image, directly from given point correspondences. As introduced in Chapter 2, basically, there are two types of calibration approaches: On the one hand, methods based on 2D or 3D calibration targets, which exploit additional scene information such as planar structures, parallel lines or calibration tools. On the other hand, modern techniques on auto-calibration that aim to calibrate multiple devices from general scenes without any form of user interaction or additional assumptions. Although, the consideration of additional information works well, its use in practice is cumbersome and time-consuming. An auto-calibration procedure without these requirements is therefore preferable. The presented procedure uses images from at least three different views. Assuming static scenes, it is basically irrelevant whether these images were taken with several cameras at the same time or one after the other with only one camera. In particular, active lighting elements such as a projector treated as an "inverse" camera can also be considered. The following investigations will therefore be focused on static scenes and use the term *view* to describe an image content including its pose, completely independent of a point in time or whether an image is captured or projected. The camera settings used for the acquisition, will not be limited at all. Even if a single camera is used, the camera settings will not be assumed to remain constant for all images. Therefore, shooting is supported with automatic image settings such as autofocus, as well as with different camera models. The only limitation made for the observations listed here is that all images reproduce the same scene and the visual content sufficiently overlaps. This is especially necessary in order to find point correspondences between the views, which are the only required input for the calibration.

Since no calibration objects are used, planar structures can not be assumed to be available in the scene in general. Planar-based methods such as the ones of Sturm and Maybank [155], Malis *et al.* [112] or Chen *et al.* [22]) may work in many in-house scenarios but not in arbitrary scenes. The general approach on auto-calibration, that is treated here, is only based on the computation of epipolar relations between multiple devices, which are represented as fundamental matrices. Well-known procedures as described by Zhang in [189], and Hartley and Zisserman in [62] already address this tasks. Nevertheless, in the presented approach all subsequent steps on intrinsic and extrinsic calibration depend on the accuracy of epipolar relations estimated by such methods. Moreover, these subsequent steps appeared to be very sensitive to small errors in the epipolar geometry. Therefore, it is important to achieve the highest possible accuracy to avoid a failure of further calibration steps.

Having computed the fundamental matrices, intrinsic parameters can theoretically directly be extracted. Since the entire calibration process, except the estimation of the fundamental matrix, depends further on the intrinsic parameters, their accurate estimation is crucial and likely the most difficult

and error-prone part. Consequently, extensive research has been carried out in this respect in recent decades and methods have been discovered that enable intrinsic calibration directly on the basis on the fundamental matrix. Although state-of-the-art methods are theoretically sound and valuable, their practical application is often not stable and fails in many cases. Especially setups consisting of different camera models and projectors, as in the case of active scanning, can cause problems. This chapter combines and extends our works [46] and [48] on auto-calibration in order to provide an entire pipeline based on recent research results. The presented procedure provides accurate calibration without any user interaction.

Method [46] focuses on the robust computation of accurate and compatible epipolar geometry between all used devices. It heavily increases the chance of convergence of further calibration steps based on the fundamental matrices. It is the first approach to combine epipolar and trifocal relations, as well as distortions of arbitrary order. The correction of errors caused by the optical system of the devices is an essential aspect. Distortions are corrected while the fundamental matrices are estimated. Based on this, further auto-calibration also apply to cameras with a large field of view.

Method [48] moreover increases the region of convergence in order to provide stable focal length estimates of the used devices directly from the previously computed fundamental matrices. In order to compute Euclidean reconstructions, precise estimates are essential. As mentioned in Chapter 2, other intrinsics like skew and aspect ratio are of less importance nowadays, as modern devices are equipped with square pixels. The minimized energy term is smooth and of superior properties, which allows a stable estimation of the principal points, even if they are far off the image centers. This leads in particular to an advantage over the widely used method of Pollefeys *et al.* [133]. Their approach is not based on the fundamental matrices, but uses directly knowledge on the absolute conic and is widely used for camera calibration. It estimates the camera matrices up to a projective transformation and upgrades them to generate Euclidean reconstructions in a second step. However, in the presence of video projectors, it performed poorly in all experiments due to the strongly shifted principal point. This motivated the investigations carried out based on Kruppa's equations, which led to a method that, in particular, no longer requires particularly good initializations for convergence and is therefore suitable for structured light setups.

In order to extract the extrinsic parameters of the devices (rotation and translation) from the fundamental matrices, as described by Hartley and Zisserman in [62], a simple formula is provided to find the correct decomposition of extrinsic parameters. It allows to choose the correct configuration between four algebraic solutions only by comparing two scalar values. Famous bundle-adjustment as it was treated in a multitude of publications e.g. by Triggs *et al.* [163], Engels *et al.* [37], Chen *et al.* [23], Zach [178], Aravkin *et al.* [9], Kanatani and Sugaya [84]) or methods like the one from Gherardi and Fusiello [53] are methods that require already appropriate calibration as initialization. They may be finally applied in order to further refine the parameters.

5.2. Related Work

Extensive research has been done in the field of auto-calibration. Estimating the epipolar geometry between two views has been reviewed by Zhang in [189]. In a nutshell, Euclidean epipolar errors between point correspondences are minimized in a nonlinear energy functional. This approach represents the standard method when distortion-free input images and precise point correspondences are available. Torr and Zisserman [160] examined the trifocal tensor to derive relationships between three images. Their approach depends on very precise point correspondences and is therefore practically hardly usable. Brito et al. [18] and Stein [150] proposed methods to estimate a single radial distortion parameter. Fitzgibbon [49] combined single parameter estimation with the calculation of the epipolar geometry. Unfortunately, this approach is limited to constant intrinsic parameters for all views.

The most popular distortion model has already been introduced in 1966 by Brown [19] and is still widely used. Romera and Gomez [139] proposed a method for approximating Brown's distortion parameters using calibration boards for homography estimation. In the context of auto-calibration, Li *et al.* [94] presented an alternating method for simultaneously estimating distortions and fundamental matrices, combining the work of Zhang and Brown. Unfortunately, the resulting fundamental matrices are often not sufficiently accurate to apply further intrinsic calibration from the fundamental matrices. Gherardi and Fusiello [53] extended this approach, but require very good initialization and the choice of a large number of weighting parameters, which are often not available in practice.

Research on intrinsic calibration from epipolar geometry achieved a quantum leap by the theory of the absolute conic used by Faugeras *et al.* in order to introduce *Kruppa's Equations* into computer vision in [38]. These equations will represent the basis of the proposed auto-calibration method as they describe a direct connection between fundamental matrices and the respective intrinsic calibration matrices. Bougnoux [17] and Hartley [61] gave formulations for computing the focal lengths of two uncorrelated views given their fundamental matrix. Both approaches depend on known principal points and epipoles. Since the epipoles are usually estimated as the null-space of the fundamental matrix, they are sensitive to small inaccuracies in the fundamental matrix. This can lead to instabilities of the methods, even if correct principal points are given. To avoid this problem, Hartley reformulated Kruppa's equations in terms of the singular value decomposition of the fundamental matrix to introduce epipole-invariant Kruppa Equations in [64]. Based on Hartleys work, Sturm [154] presented a method for two views with constant focal lengths and fixed principal points. Whitehead and Roth [172] gave a more general approach for multiple devices with varying focal lengths, but still restricted to given principle points. Although high quality cameras can be assumed to have the principal point close to the image center, this is generally not the case. For optical components with interfaces that are not orthogonal to the principal ray, e.g. in the case of projectors, the principal point can even be outside

the image. Therefore, in a number of practical applications, said methods are not suitable for auto-calibration.

In order to estimate both focal lengths and principal points, Pollefeys *et al.* [133] presented a least-squares method based on the absolute dual quadric supporting an arbitrary number of devices. The approach is widely used for camera compositions, but can fail in order to calibrate setups containing projectors whose principal points are usually far away from the image center. Gherardi and Fusiello [53] built on this approach in upgrading given camera matrices and introduced a more specific energy functional with several regularizations. This method is based on a given initial calibration and is essentially a post-processing. In order to converge to the global minimum, both approaches require good initialization and suitable regularizers. Finally, Lourakis and Deriche [104] presented a method that minimizes pairwise differences of the epipole-invariant Kruppa equations from [64]. These differences are weighted by covariance matrices from the numerical optimization of the fundamental matrix. This method is build on solving Kruppa's equations and appeared to be able to handle even coarser initializations. Nevertheless, this method has some weaknesses, which are addressed later on in Section 5.4.1. All three energy-based methods are likely to fail if the principal points are far off the image center or in presence of significantly differing focal lengths. The method proposed here is based on an energy, derived from epipole-invariant Kruppa equations. It will converge to the global minimum under reasonable initial conditions and thus significantly extends the practical applicability.

5.3. Determining the Epipolar Geometry

The presented auto-calibration method that does not require additional calibration tools, starts with the estimation of the epipolar geometry from point correspondences. Therefore, it is a basic component of the procedure since all subsequent calibration steps are based on the fundamental matrices. The accuracy of them is therefore decisive for an exact calibration. Even with small inaccuracies, it is likely that the intrinsic calibration, which would be the next step, and thus the entire calibration process fails.

Finding suitable fundamental matrices that meet all expected requirements, depends strongly on the quality of the correspondences. Furthermore, since the pinhole camera model is assumed for theoretical consideration, it may be crucial to correct distortions of practical systems jointly.

Standard methods minimize the epipolar error to approximate fundamental matrices. If more than two views are given, the trifocal error can theoretically also be used, but it is sensitive to noise and therefore less practical. In this chapter, a combination of both error types is proposed, that leads to consistently improved fundamental matrices. The proposed method has been evaluated on both synthetic and real data sets. Besides the increased probability that intrinsic calibration methods converge, the resulting intrinsic and extrinsic parameters are of superior accuracy. The method is quasi parameter-free, easy to implement, and requires only a slightly increased computational

effort in comparison to the standard methods.

5.3.1. Background

First, both epipolar and trifocal errors are introduced, which are used to evaluate epipolar geometry between different camera views. These error measures are used to estimate as well the fundamental matrices between all the views and to determine the distortions caused by the lenses of the used devices. Relations between two different views C_i and C_j are given by fundamental matrices $\mathbf{F}_{ij}$ which model the epipolar geometry between image I_i and image I_j. Thereby, fundamental matrix $\mathbf{F}_{ij}$ is defined as a rank two matrix that satisfies the epipolar constraint

$$\tilde{\mathbf{x}}_j^\mathsf{T} \mathbf{F}_{ij} \tilde{\mathbf{x}}_i = 0, \quad \forall\ \tilde{\mathbf{x}}_i \in I_i,\ \tilde{\mathbf{x}}_j \in I_j \tag{5.1}$$

for corresponding image points $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$ in homogeneous coordinates.

Epipolar Error A standard approach to approximate (5.1) is to minimize the *epipolar error* for all correspondences and all view pairs. For each point pair $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$ this error is defined by the Euclidean distance of the computed epipolar line $\mathbf{l}_{ij} = \mathbf{F}_{ij}\tilde{\mathbf{x}}_i$ to its respective point $\tilde{\mathbf{x}}_j$ in the other image:

$$d(\tilde{\mathbf{x}}_j, \mathbf{F}_{ij}\tilde{\mathbf{x}}_i) = \frac{\tilde{\mathbf{x}}_j^\mathsf{T} \mathbf{F}_{ij} \tilde{\mathbf{x}}_i}{\sqrt{(\mathbf{l}_{ij})_1^2 + (\mathbf{l}_{ij})_2^2}}, \tag{5.2}$$

where $(\mathbf{l}_{ij})_1$ and $(\mathbf{l}_{ij})_2$ denote the first and second entry of epipolar line $\mathbf{l}_{ij}$. Since this also applies to the mapping $\mathbf{F}_{ij}^\mathsf{T}$ from $\tilde{\mathbf{x}}_j$ to $\tilde{\mathbf{x}}_i$, both errors can be combined to a least squares error by adding the squared distances

$$E^{ij}_{\text{epipolar}} = d(\tilde{\mathbf{x}}_j, \mathbf{F}_{ij}\tilde{\mathbf{x}}_i)^2 + d(\tilde{\mathbf{x}}_i, \mathbf{F}_{ij}^\mathsf{T}\tilde{\mathbf{x}}_j)^2. \tag{5.3}$$

This error measure is symmetrical between two views. Considering all epipolar relations between C views, this results in $\frac{1}{2}C(C-1)$ pairwise epipolar errors.

Trifocal Error Unlike the epipolar error, which describes only relationships between two views, the *trifocal error* connects relations between three views that are not collinear. For a triple of point correspondences $(\tilde{\mathbf{x}}_{i_1}, \tilde{\mathbf{x}}_{i_2}, \tilde{\mathbf{x}}_j) \in I_{i_1} \times I_{i_2} \times I_j$, theoretically epipolar lines $\mathbf{l}_{i_1 j} = \mathbf{F}_{i_1 j}\tilde{\mathbf{x}}_{i_1}$ and $\mathbf{l}_{i_2 j} = \mathbf{F}_{i_2 j}\tilde{\mathbf{x}}_{i_2}$ should intersect in image point $\tilde{\mathbf{x}}_j$. The squared Euclidean distance between intersection $s_j(\mathbf{l}_{i_1 j} \times \mathbf{l}_{i_2 j})$ (with scaling factor s_j) and measured point $\tilde{\mathbf{x}}_j$ is defined by:

$$E^{j}_{\text{trifocal}} = \left\| \tilde{\mathbf{x}}_j - s_j(\mathbf{F}_{i_1 j}\tilde{\mathbf{x}}_{i_1} \times \mathbf{F}_{i_2 j}\tilde{\mathbf{x}}_{i_2}) \right\|_2^2 \tag{5.4}$$

$$\text{with } s_j = \left((\mathbf{F}_{i_1 j}\tilde{\mathbf{x}}_{i_1})_1 (\mathbf{F}_{i_2 j}\tilde{\mathbf{x}}_{i_2})_2 - (\mathbf{F}_{i_1 j}\tilde{\mathbf{x}}_{i_1})_2 (\mathbf{F}_{i_2 j}\tilde{\mathbf{x}}_{i_2})_1\right)^{-1}, \tag{5.5}$$

$$\text{for } (i_1, j), (i_2, j) \in D_\mathcal{F},\ (i_1, j) \neq (i_2, j)$$

$$\text{and } \mathbf{F}_{kl} = \mathbf{F}_{lk}^\mathsf{T} \text{ for } (k, l) \in D_\mathcal{F},\ k > l$$

The set of indices $D_\mathcal{F}$ is given by all combinations of views. For each image we can compute $\frac{1}{2}(C-1)(C-2)$ trifocal errors, which leads to $\frac{1}{2}C(C-1)(C-2)$ contributions to the total trifocal error.

Extensions to Quadrifocal Error In theory error measures for relations between four or more images exist as well, like the ones using the quadrifocal tensor. In practice the number of combinations to be considered would be way too large. Nevertheless, the trifocal relations are sufficient to correlate even more than three views transitively, which makes the use of higher error relations dispensable.

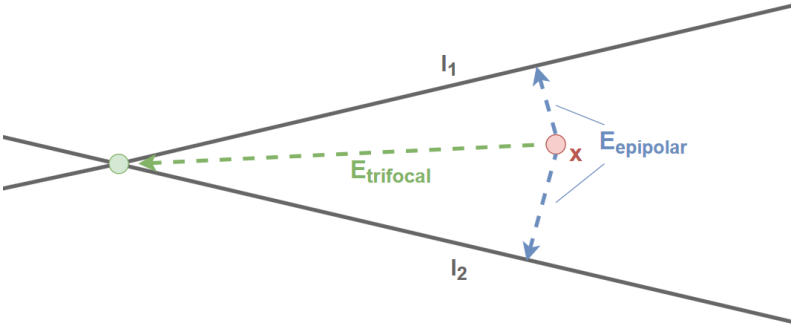


Figure 5.1.: *Visualization of $E_{epipolar}$ and $E_{trifocal}$ for an image point $\mathbf{x}$ and associated epipolar lines $\mathbf{l}_1$ and $\mathbf{l}_2$ with respect to corresponding image points $\mathbf{x}_1$ and $\mathbf{x}_2$ from two other views.*

5.3.2. Fundamental Matrices

In general, minimizing the epipolar error is well suited to estimate fundamental matrices between two views. The trifocal error, however, includes a more global arrangement between all views and ensures that all fundamental matrices are coherent. Nevertheless, minimizing the trifocal error usually does not lead to good results as it requires disproportionately good initialization to converge to the global minimum and is sensitive to noisy data. In order to investigate the advantages of the different errors to different situations, a combination of both errors, linked by a scalar regularization parameter $\tau \in \mathbb{R}_0^+$, is introduced to exploit the advantages of both approaches.

For a given set of N correspondences in C views, optimal fundamental matrices are sought as the rank two minimizers of functional (5.6).

$$\underset{\substack{\mathbf{F}_{ij} \in \mathbb{R}^{3\times 3}, \\ (i,j) \in D_{\mathcal{F}}}}{\operatorname{argmin}} \sum_{n=1}^{N} \sum_{(i,j)\in D_{\mathcal{F}}} E_{\text{epipolar}}^{ij,(n)} + \tau \sum_{j=1}^{C} E_{\text{trifocal}}^{j,(n)} \tag{5.6}$$

$$\text{s.t.} \quad \operatorname{rank}(\mathbf{F}_{ij}) = 2, \ \forall (i,j) \in D_{\mathcal{F}} \tag{5.7}$$

$E_{\text{epipolar}}^{ij,(n)}$ and $E_{\text{trifocal}}^{j,(n)}$ denote the epipolar and trifocal errors with respect to the n-th point correspondence.

Parameterization Since a fundamental matrix has rank two and is up to scale, it has seven degrees of freedom. Following Csurka et al. [27] a fundamental matrix can be parameterized according to (5.8). This parameterization has an optimal condition number, which is important for the convergence rate of the numerical optimization.

$$\mathbf{F}(f_1,...,f_7) = \begin{pmatrix} f_6(f_1f_4+f_2f_5) \\ +f_7(f_3f_4+f_5) & f_1f_6+f_3f_7 & f_2f_6+f_7 \\ f_1f_4+f_2f_5 & f_1 & f_2 \\ f_3f_4+f_5 & f_3 & 1 \end{pmatrix} \tag{5.8}$$

Minimizing functional (5.6) under this parameterization enforces property (5.7). Therefore, it is a well-conditioned optimization problem in $t \cdot C$ unknowns, that can be efficiently solved by truncated *Levenberg-Marquardt* algorithm.

5.3.3. Distortion Correction

At the same time, distortion parameters are estimated to account for the underlying pinhole camera model. Due to the least-squares formulation, the quality of a fundamental matrix, computed by minimizing (5.6), strongly depends on highly accurate point correspondences. Therefore, a distortion correction is essential. According to *Brown's Model* [19], every un-distorted image point $\hat{\mathbf{x}}$ is related to the observed distorted image point $\mathbf{x} = (x,y)^\mathsf{T}$ by

$$\hat{\mathbf{x}}(x_p,y_p,k_1,...,k_L) = \begin{pmatrix} x_p + (x-x_p)(1+\sum_{l=1}^{L} k_l(\frac{r}{d})^{2l}) \\ y_p + (y-y_p)(1+\sum_{l=1}^{L} k_l(\frac{r}{d})^{2l}) \end{pmatrix}, \tag{5.9}$$

(x_p,y_p) denotes the center of distortion and $\frac{r}{d} \in [0,1] \subset \mathbb{R}$ the Euclidean distance of the normalized distorted image point $\mathbf{x}$ to the center of distortion. Taking into account that points $\mathbf{x}_s$ are distorted in this way by parameters $x_{p_s}, y_{p_s}, k_{1_s}, ..., k_{L_s}$ for cameras $s = 1,...,C$, the distortions are corrected by minimizing functional (5.10) assuming fixed fundamental matrices.

$$\underset{\substack{x_{p_s},y_{p_s},k_{l_s} \in \mathbb{R}, \\ l \in \{1,...,L\}, \\ s \in \{1,...,C\}}}{\operatorname{argmin}} \sum_{n=1}^{N} \sum_{(i,j)\in D_{\mathcal{F}}} E^{ij,(n)}_{\text{epipolar}} + \tau \sum_{j=1}^{C} E^{j,(n)}_{\text{trifocal}} \tag{5.10}$$

5.3.4. Minimization

In order to obtain the desired epipolar geometry, while correcting the distortions jointly, problems (5.6) and (5.10) are alternatingly minimized. For both

sub-problems solutions can be found separately using truncated *Levenberg-Marquardt* ([120], [137], [102]), while keeping the variables of the other sub-problem constant. A global solution can finally be found by alternating between both sub-problems. Therefore, let $J^{ij,(n)}_{\text{epipolar}}$ and $J^{j,(n)}_{\text{trifocal}}$ denote the Jacobian matrices of the epipolar and trifocal errors either with respect to the fundamental matrices or the parameters of the distortion model. An infinitesimal update δ of the parameter vector can be obtained solving

$$(A + \alpha I)\delta = b \tag{5.11}$$

with a numerical step size α of the algorithm. System matrix A, containing the approximated Hessians and right hand side b are explicitly given by

$$A = \sum_{n=1}^{N} \sum_{(i,j)\in D_{\mathcal{F}}} J^{ij,(n)\mathsf{T}}_{\text{epipolar}} J^{ij,(n)}_{\text{epipolar}} + \tau \sum_{j=1}^{C} J^{j,(n)\mathsf{T}}_{\text{trifocal}} J^{j,(n)}_{\text{trifocal}}. \tag{5.12}$$

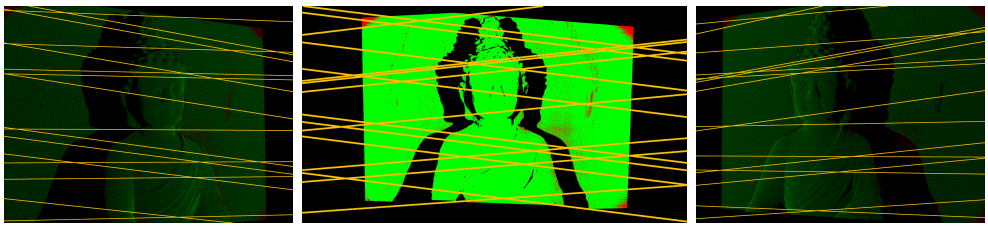
$$b = -\sum_{n=1}^{N} \sum_{(i,j)\in D_{\mathcal{F}}} J^{ij,(n)\mathsf{T}}_{\text{epipolar}} \sqrt{E^{ij,(n)}_{\text{epipolar}}} - \tau \sum_{j=1}^{C} J^{j,(n)\mathsf{T}}_{\text{trifocal}} \sqrt{E^{j,(n)}_{\text{trifocal}}}.$$

To achieve good convergence rates, suitable dumping or preconditioning strategies as reported by Kwak *et al.* in [91] or recently by Bellavia *et al.* in [12] can be used. For the centers of distortion, the image centers have appeared to be a good initialization. Although the principal point of projectors is generally not in the image center, no special treatment is needed. The radial distortion parameters k_{l_s} are initialized with zeros. Since the center of distortion is often inside the image, in most cases scaling factor d can be chosen as half of the image diagonal.

5.3.5. Robust Initialization

A robust initialization for matrices $\mathbf{F}_{ij}$ can be achieved by the *Normalized-Eight-Point* algorithm, combined with a *RANSAC* procedure in order to account for outliers. The well-known approach minimizes Equation (5.1) for at least 8 measurements including a normalization step that maximizes numerical accuracy as proposed by Hartly in [63]. The minimization is performed after reformulation to matrix-vector form similar to the procedure shown in Section 2.4. Finally, after reordering the destination vector into a 3×3 fundamental matrix $\mathbf{F}$, the rank 2 condition is enforced by setting the smallest singular value to zero.

This approach is of low computational cost and allows for application in a robust *RANSAC* approach, that is well-established. For visualization Figure 5.2 shows the plotted epipolar lines of an exemplary setup consisting of two cameras and a projector. Point correspondences are generated using phase shifted sinusoidal structured light. Valid matches are plotted in green, while outliers are plotted in red. The removal of outliers by this robust initialization step is of high importance for subsequent steps, that are variants of least squares energies, which are known to be sensitive to outliers.



(a) First camera view **(b)** Projector view **(c)** Second camera view

Figure 5.2.: *Results of RANSAC initialization of the fundamental matrices in a structured light setup with two cameras and a projector. Green points are valid correspondences, while red ones are outliers, detected by the robust method. Respective epipolar lines are plotted in orange.*

5.4. Intrinsic Calibration

Pollefeys *et al.* [133] as well as Lourakis et al. [105] developed methods for auto-calibration of camera setups with uncorrelated intrinsic camera parameters. While the former represents the common way for camera compositions, it did not work satisfactorily in a large number of investigations with setups containing video projectors. A probable reason is the principal point, which is far away from the image center for such devices. Here, the second approach, which is in contrast based on the available epipolar geometry, provided more stable results, which is why it serves as the basis for the presented method for calibrating structured light setups.

Lourakis [105] introduced a method to estimate intrinsic parameters directly from fundamental matrices. Therefore, at least three views are necessary, while more views increase the accuracy as well as the number of calibration parameters that can be estimated. The covariance of the numerical minimization algorithm is used to weight the uncertainties to enhance the stability of the method, as described by Csurka *et al.* in [27]. Unfortunately, it requires epipolar relations of high accuracy as it is very error-prone and sensitive to small errors in the fundamental matrices. This work assesses reasons for this behavior, in particular for [104] and the method of Bougnoux [17], which treats especially the two-view case. Based on the analysis, a more stable method is proposed. A continuous and smooth energy functional is introduced, providing superior convergence properties. I.e. it converges faster and has a significantly enlarged convergence region with respect to the global minimum.

Finally, a detailed evaluation has been conducted and a comparison with the method of Lourakis is presented.

5.4.1. Background

The basis of intrinsic auto-calibration was the development of the theory of the absolute conic. The main idea is that any quadric, captured by an optical device, is projected as a conic onto the image plane and the respective epipolar lines are tangential to this conic. Figure 5.3 visualizes this basic relation.

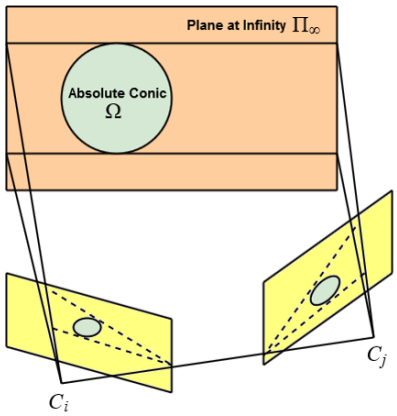


Figure 5.3.: *Visualization of the absolute conic at the plane at infinity and its projections. Kruppa's equations give a direct relation of the cameras' intrinsics and their fundamental matrix.*

Furthermore, the dual of the image of the absolute conic is independent of the camera pose. Its computation is equivalent to the calculation of the intrinsic calibration of the device.

Given the epipolar geometry between image planes I_i and I_j, represented by a fundamental matrix $\mathbf{F}_{ij}$, *Kruppa's equations* use this knowledge to describe a direct connection between $\mathbf{F}_{ij}$ and the intrinsic calibration matrices $\mathbf{K}_i$ and $\mathbf{K}_j$ of the respective cameras C_i and C_j.

Kruppa Equations Let $\mathbf{e}_i$ and $\mathbf{e}_j$ be the left and right epipoles computed from the left and right null-space of $\mathbf{F}_{ij}$ and $\mathbf{w}_i^* = \mathbf{K}_i\mathbf{K}_i^\mathsf{T}$ and $\mathbf{w}_j^* = \mathbf{K}_j\mathbf{K}_j^\mathsf{T}$ the duals of the absolute conic. Then *Kruppa's equations* read:

$$\begin{aligned}[\mathbf{e}_j]_\times \mathbf{w}_j^* [\mathbf{e}_j]_\times &= \mathbf{F}_{ij}\mathbf{w}_i^*\mathbf{F}_{ij}^\mathsf{T} \\ [\mathbf{e}_i]_\times \mathbf{w}_i^* [\mathbf{e}_i]_\times &= \mathbf{F}_{ij}^\mathsf{T}\mathbf{w}_j^*\mathbf{F}_{ij}\end{aligned} \tag{5.13}$$

where $[\cdot]_\times$ denotes the cross-product matrix. However, solving these equations is not practicable due to the strong dependence on the epipole estimates, which are in general very error-prone.

Epipole-Invariant Kruppa Equations Hartley [64] expressed the equations by avoiding dependencies on the epipoles. Equations (5.13) are equivalent to:

$$\underbrace{\frac{\sigma_1^2\mathbf{v}_1^\mathsf{T}\mathbf{w}_i^*\mathbf{v}_1}{\mathbf{u}_2^\mathsf{T}\mathbf{w}_j^*\mathbf{u}_2}}_{=:\rho_1} = \underbrace{\frac{\sigma_1\sigma_2\mathbf{v}_1^\mathsf{T}\mathbf{w}_i^*\mathbf{v}_2}{-\mathbf{u}_2^\mathsf{T}\mathbf{w}_j^*\mathbf{u}_1}}_{=:\rho_2} = \underbrace{\frac{\sigma_2^2\mathbf{v}_2^\mathsf{T}\mathbf{w}_i^*\mathbf{v}_2}{\mathbf{u}_1^\mathsf{T}\mathbf{w}_j^*\mathbf{u}_1}}_{=:\rho_3} \tag{5.14}$$

$$\text{with} \qquad \mathbf{F}_{ij} = \mathbf{U}\mathbf{S}\mathbf{V}^\mathsf{T} = (\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)\begin{pmatrix}\sigma_1 & 0 & 0\\ 0 & \sigma_2 & 0\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}\mathbf{v}_1\\ \mathbf{v}_2\\ \mathbf{v}_3\end{pmatrix} \tag{5.15}$$

Numerators and denominators of the terms in Equations (5.14) describe the tangent lines of the image of the absolute conic in the different views. These must be identical up to scale and are therefore considered relatively. These equations are the basis of Lourakis' method [104]. Since the method derived in this chapter addresses weaknesses of the method, its main idea will be briefly introduced.

Basic Approach: Method of Lourakis Lourakis *et al.* [104] proposed a nonlinear approach for approximating Equations (5.14). The least-squares energy to be minimized is defined by

$$\underset{\mathbf{K}_l,\ l\in\{1,\ldots,C\}}{\operatorname{argmin}} \sum_{\substack{(i,j)\in D_{\mathcal{F}}\\(u,v)\in D_{\mathcal{K}}}} \frac{(\rho_u^{ij}-\rho_v^{ij})^2}{{\sigma_{uv}^{ij}}^2}, \tag{5.16}$$

where $\mathbf{K}_l$ denotes the intrinsic calibration matrices, $D_{\mathcal{F}}$ the set of device pairings and $D_{\mathcal{K}}$ the set of combinations of Kruppa terms. σ_{uv} are confidence measures calculated during the estimation of the fundamental matrices.

This method extends the two-view case from Equation 5.14 to any number of C devices by considering $\frac{1}{2}C(C-1)$ pairwise fundamental matrices. Each of them provides two independent constraints, which limits the number of computable camera parameters to $C(C-1)$. The number of determinable parameters per device is sufficient for most applications and increases with the number of devices used, as explained in more detail in Section 5.7.2.

Avoiding degenerate cases as reported by Sturm in [153] and [152] by ensuring sufficient camera parameter variance (as it is usually given, by using multiple devices with different intrinsic camera settings), the method is known to work well for three or more devices, assuming high quality epipolar relations and good initialization of the principal points and focal lengths. Nevertheless, the method may fail in many practical situations for the following reasons:

- Weak initialization of focal lengths or principal points.
- Significantly differing focal lengths.
- Bias towards larger focal lengths.
- Significantly off-center principal points.
- Singularities of the energy.

Assumptions As mentioned in Chapter 2, for modern devices, zero skew and square pixels can be assumed. Thus the dual of the absolute conic can be written as

$$\mathbf{w}_s^* = \mathbf{K}_s\mathbf{K}_s^{\mathsf{T}} = \begin{pmatrix} f_s^2+x_{p_s}^2 & x_{p_s}y_{p_s} & x_{p_s} \\ x_{p_s}y_{p_s} & f_s^2+y_{p_s}^2 & y_{p_s} \\ x_{p_s} & y_{p_s} & 1 \end{pmatrix}, \tag{5.17}$$

where f_s denotes the focal length and $\mathbf{c}_{p_s} = (x_{p_s}, y_{p_s}, 1)^\mathsf{T}$ the principal point of any device C_s.

5.4.2. Stable Energy Minimization

In this chapter, a new robust energy functional is proposed. Compared to Lourakis' it has the following beneficial properties:

- Focal lengths of different scales are treated homogeneously and unbiasedly.
- The multidimensional energy field is smooth and has no discontinuities or singularities in the range of possible solutions.
- A significantly larger region of convergence to the global minimum, which is finite and uniquely defined.
- The energy function is quasi-symmetric with respect to the *Kruppa curves* (5.24), and seams convex with respect to the principal point.

This greatly increases the stability of the numerical optimization as well as the likelihood of convergence.

Kruppa Curves of Focal Lengths

Using the notation and assumptions of Section 5.4.1, the terms of (5.14) can be written as

$$\rho_1 = \frac{f_i^2\sigma_1^2(v_{11}^2 + v_{12}^2) + \sigma_1^2(\mathbf{c}_{p_i}^\mathsf{T}\mathbf{v}_1)^2}{f_j^2(u_{21}^2 + u_{22}^2) + (\mathbf{c}_{p_j}^\mathsf{T}\mathbf{u}_2)^2} \tag{5.18}$$

$$\rho_2 = \frac{f_i^2\sigma_1\sigma_2(v_{11}v_{21} + v_{12}v_{22}) + \sigma_1\sigma_2(\mathbf{c}_{p_i}^\mathsf{T}\mathbf{v}_1)(\mathbf{c}_{p_i}^\mathsf{T}\mathbf{v}_2)}{-f_j^2(u_{11}u_{21} + u_{12}u_{22}) - (\mathbf{c}_{p_j}^\mathsf{T}\mathbf{u}_1)(\mathbf{c}_{p_j}^\mathsf{T}\mathbf{u}_2)} \tag{5.19}$$

$$\rho_3 = \frac{f_i^2\sigma_2^2(v_{21}^2 + v_{22}^2) + \sigma_2^2(\mathbf{c}_{p_i}^\mathsf{T}\mathbf{v}_2)^2}{f_j^2(u_{11}^2 + u_{12}^2) + (\mathbf{c}_{p_j}^\mathsf{T}\mathbf{u}_1)^2}, \tag{5.20}$$

where u_{kl} and v_{kl} denote the l-th entries of vectors $\mathbf{u}_k$ and $\mathbf{v}_k$. With the explicit formulations of (5.18), (5.19) and (5.20), Equations (5.14) of any fundamental matrix $\mathbf{F}_{ij}$ can be moreover written in the form

$$\frac{f_i^2 a_{i1} + b_{i1}}{f_j^2 a_{j1} + b_{j1}} = \frac{f_i^2 a_{i2} + b_{i2}}{f_j^2 a_{j2} + b_{j2}} = \frac{f_i^2 a_{i3} + b_{i3}}{f_j^2 a_{j3} + b_{j3}} \tag{5.21}$$

with coefficients

$$\begin{pmatrix} a_{i1} \\ a_{i2} \\ a_{i3} \end{pmatrix} = \begin{pmatrix} \sigma_1^2(v_{11}^2 + v_{12}^2) \\ \sigma_1\sigma_2(v_{11}v_{21} + v_{12}v_{22}) \\ \sigma_2^2(v_{21}^2 + v_{22}^2) \end{pmatrix}, \quad \begin{pmatrix} b_{i1} \\ b_{i2} \\ b_{i3} \end{pmatrix} = \begin{pmatrix} \sigma_1^2(\mathbf{c}_{p_i}^\mathsf{T}\mathbf{v}_1)^2 \\ \sigma_1\sigma_2(\mathbf{c}_{p_i}^\mathsf{T}\mathbf{v}_1)(\mathbf{c}_{p_i}^\mathsf{T}\mathbf{v}_2) \\ \sigma_2^2(\mathbf{c}_{p_i}^\mathsf{T}\mathbf{v}_2)^2 \end{pmatrix}$$

$$\begin{pmatrix} a_{j1} \\ a_{j2} \\ a_{j3} \end{pmatrix} = \begin{pmatrix} u_{21}^2 + u_{22}^2 \\ u_{11}u_{21} + u_{12}u_{22} \\ u_{11}^2 + u_{12}^2 \end{pmatrix}, \quad \begin{pmatrix} b_{j1} \\ b_{j2} \\ b_{j3} \end{pmatrix} = \begin{pmatrix} (\mathbf{c}_{p_j}^\mathsf{T}\mathbf{u}_2)^2 \\ (\mathbf{c}_{p_j}^\mathsf{T}\mathbf{u}_1)(\mathbf{c}_{p_j}^\mathsf{T}\mathbf{u}_2) \\ (\mathbf{c}_{p_j}^\mathsf{T}\mathbf{u}_1)^2 \end{pmatrix}. \tag{5.22}$$

For fixed principal points, the equations define curves, named *Kruppa curves*, which describe direct relations between the focal lengths.
For each fundamental matrix F_{ij} coefficient vectors are defined as:

$$d_{uv}^{ij} \coloneqq \begin{pmatrix} a_{iu}a_{jv} - a_{iv}a_{ju} \\ a_{iu}b_{jv} - a_{iv}b_{ju} \\ b_{iu}a_{jv} - b_{iv}a_{ju} \\ b_{iu}b_{jv} - b_{iv}b_{ju} \end{pmatrix} \quad \text{for} \quad (u,v) \in D_{\mathcal{K}} \tag{5.23}$$

Each equation from (5.22) defines a two-dimensional parametric curve that can be represented by the one-dimensional functions $\mathcal{K}_{1,uv}^{ij}$ and $\mathcal{K}_{2,uv}^{ij}$:

$$\mathcal{K}_{1,uv}^{ij}(f_j) \coloneqq -\frac{f_j^2 d_{uv,3}^{ij} + d_{uv,4}^{ij}}{f_j^2 d_{uv,1}^{ij} + d_{uv,2}^{ij}}, \qquad \mathcal{K}_{2,uv}^{ij}(f_i) \coloneqq -\frac{f_i^2 d_{uv,2}^{ij} + d_{uv,4}^{ij}}{f_i^2 d_{uv,1}^{ij} + d_{uv,3}^{ij}} \tag{5.24}$$

The curves $\mathcal{K}_{1,uv}^{ij}(f_j)$ and $\mathcal{K}_{2,uv}^{ij}(f_i)$ and the coefficients d_{uv}^{ij} are obtained by resolving Equations (5.22) with respect to f_i and f_j. Figure 5.4 shows the Kruppa curves for three independent fundamental matrices (from left to right), plotted as green lines. Famous two-view techniques such as Bougnoux [17] determine the intersections of the curves to estimate the focal lengths. Having said that, Bougnoux and similar methods fail in the many cases where the Kruppa curves nearly coincide. Moreover, the curves are plotted into visualizations of the top views of the energies of Lourakis (top) and the proposed method (bottom) to illustrate relationship of the methods. The color coding indicates a rather high energy (yellow) up to a low energy (blue). This may give an idea of how the methods behave during minimization.

Energy as Relative Distances to Kruppa Curves

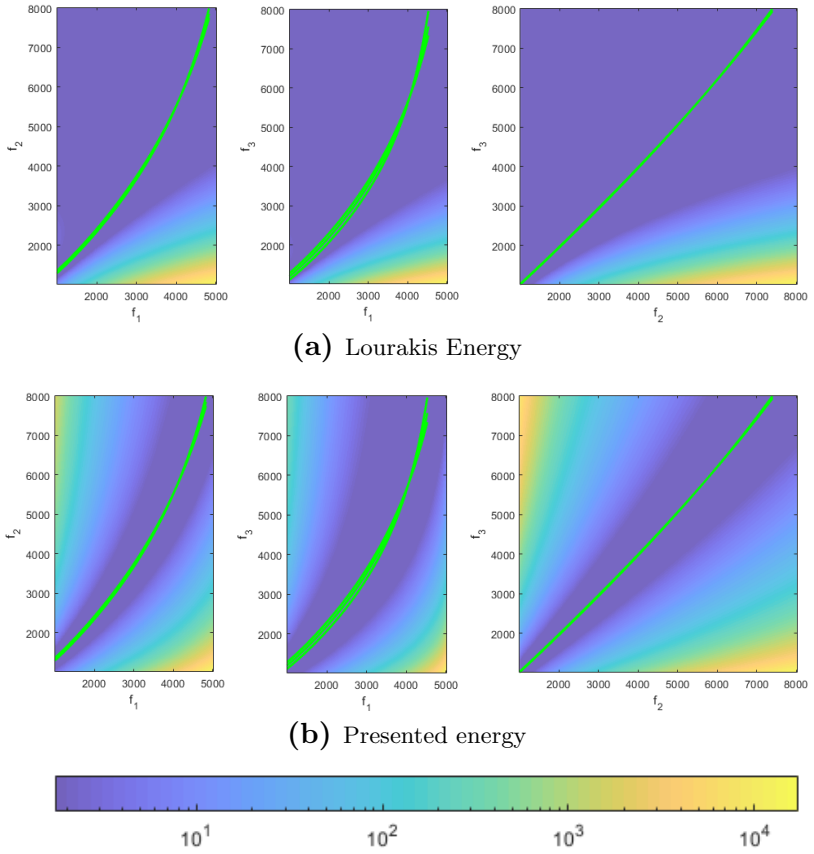
In order to establish a suitable energy term, relative Euclidean distances between focal length estimates and Kruppa curves are used, which provide a scale invariance with respect to largely different focal lengths. The new energy term reads:

$$\underset{\substack{\mathbf{c}_{p_j}, f_j \\ j \in \{1,\dots,C\}}}{\operatorname{argmin}} \sum_{\substack{(i,j) \in D_{\mathcal{F}} \\ (u,v) \in D_{\mathcal{K}}}} \left(\frac{f_i^2 - \mathcal{K}_{1,uv}^{ij}(f_j)}{f_i^2}\right)^2 + \left(\frac{f_j^2 - \mathcal{K}_{2,uv}^{ij}(f_i)}{f_j^2}\right)^2 \tag{5.25}$$

By setting up the Jacobians $J_{uv,1}^{ij}$ and $J_{uv,2}^{ij}$ for each pair of energies, (5.25) can be solved by applying truncated Levenberg-Marquardt, with system matrix A and inhomogenity b:

$$A = \sum_{\substack{(i,j) \in D_{\mathcal{F}} \\ (u,v) \in D_{\mathcal{K}}}} J_{uv,1}^{ij^\mathsf{T}} J_{uv,1}^{ij} + J_{uv,2}^{ij^\mathsf{T}} J_{uv,2}^{ij} \tag{5.26}$$

$$b = -\sum_{\substack{(i,j) \in D_{\mathcal{F}} \\ (u,v) \in D_{\mathcal{K}}}} J_{uv,1}^{ij\top}\left(1 + \frac{f_j^2 d_{uv,3}^{ij} + d_{uv,4}^{ij}}{f_i^2 f_j^2 d_{uv,1}^{ij} + f_i^2 d_{uv,2}^{ij}}\right) + J_{uv,2}^{ij\top}\left(1 + \frac{f_i^2 d_{uv,2}^{ij} + d_{uv,4}^{ij}}{f_i^2 f_j^2 d_{uv,1}^{ij} + f_j^2 d_{uv,3}^{ij}}\right).$$



(a) Lourakis Energy

(b) Presented energy

Figure 5.4.: *Top view of Lourakis' (5.16) (top) and the presented energy function from (5.25) (bottom) for several fundamental matrices. The color coding indicates a rather high energy (yellow) up to a low energy (blue) in logarithmic scale. For each fundamental matrix, the three nearly coinciding Kruppa curves are plotted in green.*

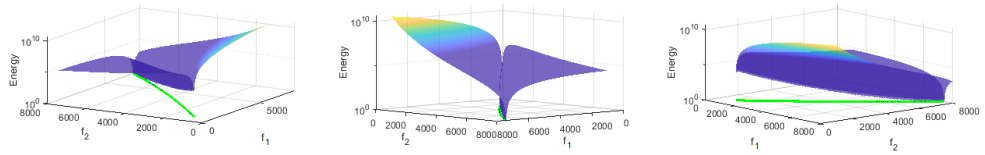
Computational effort Since both, Lourakis' method and the proposed one are based on the singular value decomposition of $\frac{1}{2}C(C-1)$ fundamental matrices, the energy functions can be set up with the same computational effort. The minimization of the energies with Levenberg-Marquardt consistently led to a faster convergence of the proposed method compared to [104], which may be explained by better conditioning of the system matrices of the new method. Appropriate preconditioning may improve the convergence rate in both cases. Since the running time in both cases is short and negligible compared to other calibration steps, no further investigations were performed.

5.4.3. Discussion

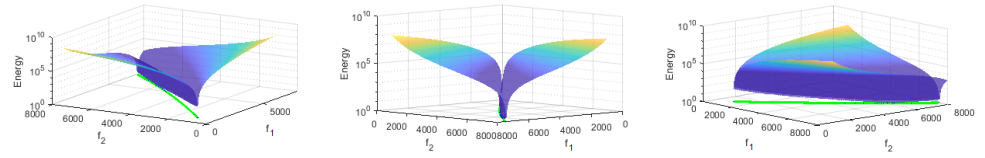
In this section the advantages of the proposed approach are presented by means of visualizations of the minimized energy functional and comparisons to the state of the art are shown.

Case: Individual focal lengths per device In real scenarios, the focal lengths of the devices often differ significantly. If these differences become too large, methods based on Kruppa's equations are likely to fail if initialization is not close to the true values. Another disadvantage is the uneven slope of the gradient of Lourakis' energy in vicinity of the Kruppa curve: For small focal lengths, the slope is significantly smaller than for large ones. Therefore, a Levenberg-Marquard update prefers the gradient direction of the larger to the smaller focal lengths when optimizing such a system. Due to the gradient slope, the method generally tends to overestimate the focal lengths. Figure 5.5 compares Lourakis' energy functional (5.16) (top row) with the proposed one (5.25) (bottom row) for several combinations of focal lengths $f_1, f_2 \in [1000, 8000]$ for different views. In particular, in the right sub-image, the increase of the slope can be observed when increasing the values of the focal lengths. Due to the relative Euclidean distances used in (5.25), the new energy functional is much more homogeneous.

Moreover, it is quasi-symmetric with respect to the Kruppa curves, which avoids the preference of a particular direction over others.



(a) Lourakis energy with respect to focal length

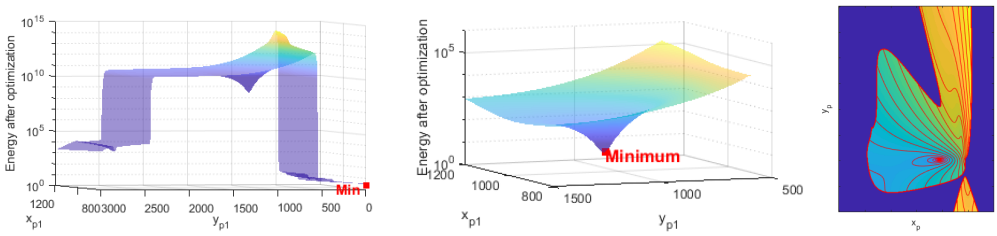


(b) Proposed energy with respect to focal length

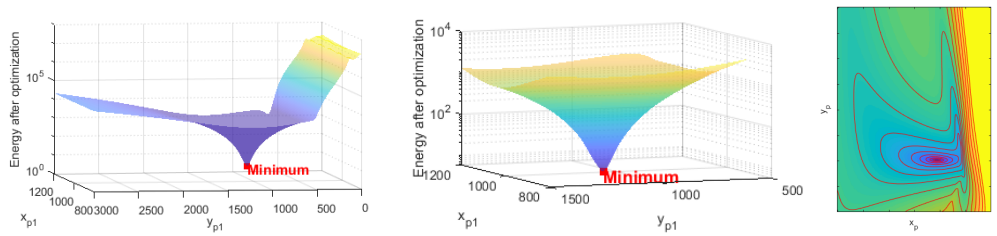
Figure 5.5.: *Kruppa curve distances computed from the fundamental matrix for several combinations of focal lengths $f_1, f_2 \in [1000, 8000]$ of devices with fixed principal points. The upper row corresponds to Lourakis' and the lower row to the proposed method. Note that the new energy is quasi-symmetric with respect to the Kruppa curve, while the state of the art is sloped unfavorably. Plots are given in logarithmic scale.*

Case: Principal point far off the image center If the principal point of a device is not close to the image center, all known methods are likely to fail. In case of Lourakis' functional, the energy surface corresponding to the

principal point positions has been analyzed and two main issues have been identified, depicted in Figure 5.6 (left): The sought minimum of the energy is almost completely surrounded by a discontinuity, so that initialization beyond the discontinuity cannot converge (top left); Even for initialization on the plateau, convergence cannot be guaranteed because the entire plateau is inclined (top left). In contrast, the proposed energy functional appears to be globally continuous, smooth and convex (Figure 5.6 top right). Comparing the second row of Figure 5.6 demonstrates that the sought minimum of both methods coincide. Although it can be assumed that the principal point of a modern camera is close to the image center, optical systems in industrial setups often have a displaced principal point. Reasons for this include additional lens assemblies, obstacles such as glass plates or liquids, and periscopic systems. Also for projection systems, e.g. used in active scanning solutions, it is not unusual for the principal point to be completely outside the image. Initialization of the principal point with the image center will often be outside the convergence plateau of Lourakis' method. The third row of Figure 5.6 shows slightly enlarged top views of Lourakis' and the proposed energy functional of the first row. The contour plots give a good indication of the improved convergence properties of the newly proposed functional.



(a) Lourakis energy with respect to principal point



(b) Proposed energy with respect to principal point

Figure 5.6.: *Energies of Lourakis (left) and the proposed Kruppa curve distance energies (right) with respect to the principal point position. While the top row shows an overview, the second row is a close-up of the area around the sought solution. Note that the location of the solution coincides for both energies. In the third row a top view of the energies of the first row is given. The region beyond the discontinuity is colored in dark blue. Please observe that the contour lines indicate significantly improved convergence properties. Plots are given in logarithmic scale for visualization.*

5.5. Extrinsic Calibration

With precise fundamental matrices and known intrinsic calibration, the essential matrix can be estimated and in turn the relative extrinsic parameters of each pair of views can be extracted, as described by Hartley and Zisserman in [62]. To this end, in the following $\mathbf{E}_{ij}$ denotes the essential matrix, which can be seen as a calibrated version of the fundamental matrix $\mathbf{F}_{ij}$. The essential matrix is computed by

$$\mathbf{E}_{ij} = \mathbf{K}_j^\mathsf{T} \mathbf{F}_{ij} \mathbf{K}_i. \tag{5.27}$$

with intrinsic calibration matrices $\mathbf{K}_i$ and $\mathbf{K}_j$ of the respective views. The essential matrix is moreover composed by the relative rotation matrix $\mathbf{R}_{ij}$ and the skew-symmetric cross-product matrix $[\mathbf{t}_{ij}]_\times$ of the translation vector $\mathbf{t}_{ij}$:

$$\mathbf{E}_{ij} = [\mathbf{t}_{ij}]_\times \mathbf{R}_{ij} \tag{5.28}$$

Following [62], the basic idea for extracting the extrinsic parameters is to use *QR-decomposition*, which however is not unique. For convenience, the indices i and j are omitted in the next steps. Given the singular value decomposition of any essential matrix $\mathbf{E} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}$, possible *QR-decomposition* are given by:

$$\begin{aligned} \mathbf{R}_1 &= \mathbf{U}\mathbf{W}_1^\mathsf{T}\mathbf{V}^\mathsf{T} \\ [\mathbf{t}_1]_\times &= \mathbf{U}\mathbf{W}_1\mathbf{\Sigma}\mathbf{U}^\mathsf{T} \end{aligned} \quad \text{and} \quad \begin{aligned} \mathbf{R}_2 &= \mathbf{U}\mathbf{W}_2^\mathsf{T}\mathbf{V}^\mathsf{T} \\ [\mathbf{t}_2]_\times &= \mathbf{U}\mathbf{W}_2\mathbf{\Sigma}\mathbf{U}^\mathsf{T} \end{aligned} \tag{5.29}$$

$$\text{with} \quad \mathbf{W}_1 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{W}_2 = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.30}$$

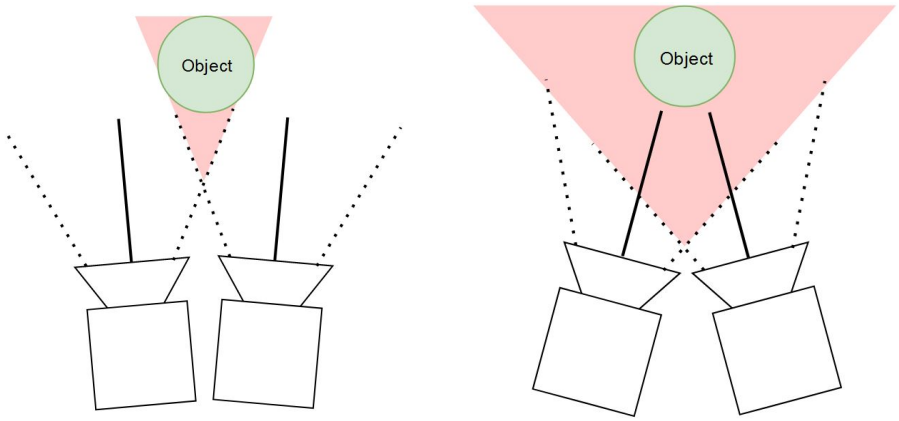
Moreover, an essential matrix $\mathbf{E}$ is up to scale, which leads to the fact, that the extracted translation vector $\mathbf{t}$ will be up-to-scale, too. The absolute length of $\mathbf{t}$ can not be determined from the essential matrix and is not of interest at this point, in contrast to the orientation. Consequently, the following four algebraic decompositions of $\mathbf{E}$, respective **-E** exist:

$$(\mathbf{R}_1, \mathbf{t}_1), \quad (\mathbf{R}_1, -\mathbf{t}_1), \quad (\mathbf{R}_2, \mathbf{t}_2), \quad (\mathbf{R}_2, -\mathbf{t}_2) \tag{5.31}$$

Nevertheless, there is exactly one feasible configuration that generates two projection matrices oriented in a way that reconstructed points lie in front of both cameras.

5.5.1. Feasible Decomposition of Essential Matrices

In order to find the right configuration, usually randomly chosen points are triangulated from the different configurations and the reconstructed positions relative to the cameras are tested. In the case of structured light reconstructions, the devices used are usually oriented towards the same object and the overlapping areas are chosen to be as large as possible. Otherwise, a large part



(a) Theoretical case with cameras pointing to different directions

(b) Practical case with adjusted cameras to have a large overlap

Figure 5.7.: *Theoretically, the cameras to be calibrated can point in any directions as long as the projection cones have a partial overlap from which point correspondences can be obtained. When reconstructing with structured light, it can be assumed that the devices are adjusted so that the cones cover a common volume that is as large as possible.*

of the acquired images would not be reconstructable. Figure 5.7 visualizes the influence of the orientation on the overlapping regions of the camera images. In the investigated case, it can be assumed that the optical axes of the devices point in the direction of the scene. Here, the possible combination of rotations and translations can be determined independently of arbitrarily chosen points.

To find the possible combination of the relative extrinsics they are applied to the optical axis of the first camera whose position w.l.o.g. is assumed to be in the origin of the world coordinate system. This results in the optical axis of the second camera. Computing the improper intersection of both axes, given by the point of shortest distance to both, allows to decide if this point is in front of both cameras or not.

Let $z_1(\lambda)$ and $z_2(\mu)$ denote parameterizations of rays passing though respective camera centers in view direction:

$$z_1(\lambda) = \begin{pmatrix}0\\0\\0\end{pmatrix} + \lambda \begin{pmatrix}0\\0\\1\end{pmatrix}, \quad z_2(\mu) = -\mathbf{R}_{ij}^{\mathsf{T}}\mathbf{t}_{ij} + \mu \mathbf{R}_{ij}^{\mathsf{T}} \begin{pmatrix}0\\0\\1\end{pmatrix} \tag{5.32}$$

Computing the point of smallest distance on every ray by solving the minimization problem

$$\underset{\lambda,\mu}{\operatorname{argmin}} \; \|z_1(\lambda) - z_2(\mu)\|_2^2 \tag{5.33}$$

results in the parameters λ and μ that minimize the distance of the rays:

$$\lambda = \frac{\mathbf{t}_1\mathbf{r}_1 + \mathbf{t}_2\mathbf{r}_2}{\mathbf{r}_3^2 - 1}, \qquad \mu = \mathbf{t}_3 + \mathbf{r}_3\lambda \tag{5.34}$$

Thereby, $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ denote the elements of the third column of rotation matrix $\mathbf{R}_{ij}$ and $\mathbf{t}_1$, $\mathbf{t}_2$, $\mathbf{t}_3$ the entries of the rotation vector $\mathbf{t}_{ij}$. Only in the feasible configuration both values λ and μ will be positive. Checking this for all four configurations leads to the feasible choice. Despite the exceptional simplicity of this approach, to our knowledge there is no other method deciding for a feasible decomposition of essential matrices by simply checking two scalar values independent from any reconstructed points.

5.5.2. Scaling Translations

From extrinsic and intrinsic camera parameters, camera matrices can be composed (see Chapter 2), ready for triangulating a point cloud from given correspondences. Since the fundamental matrix is up to scale, the extracted relative translation is also up to scale. In case of two views usually scaling is done by simply setting the translation to unit length. If there are more than two views, they must be reconciled, with respect to one chosen references. Without loss of generality, an arbitrary camera C_i is set as reference to the origin and translation $\mathbf{t}_{ij}$ between cameras C_i and C_j is defined as the reference translation. Following [151], camera matrices $\mathbf{P}_i$, $\mathbf{P}_j$ and all following cameras $\mathbf{P}_k$ are composed by

$$\begin{aligned} \mathbf{P}_i &= \mathbf{K}_i[\mathbf{I}|\mathbf{0}] \\ \mathbf{P}_j &= \mathbf{K}_j[\mathbf{R}_{ij}|\mathbf{t}_{ij}] \\ \mathbf{P}_k &= \mathbf{K}_k[\mathbf{R}_{ik}|s_k\mathbf{t}_{ik}], \quad s_k = \frac{(\mathbf{t}_{jk}\times\mathbf{t}_{ik})^\mathsf{T}(\mathbf{t}_{jk}\times\mathbf{R}_{jk}\mathbf{t}_{ij}}{\|\mathbf{t}_{jk}\times\mathbf{t}_{ik}\|_2^2} \end{aligned} \tag{5.35}$$

with factors s_k, that scale all cameras consistently with respect to the reference.

A metric calibration can finally be inferred if e.g. one of the camera baselines or the metric size of any object in the reconstructed scene is known. Subsequently, the translation vectors are simply scaled to the metric value.

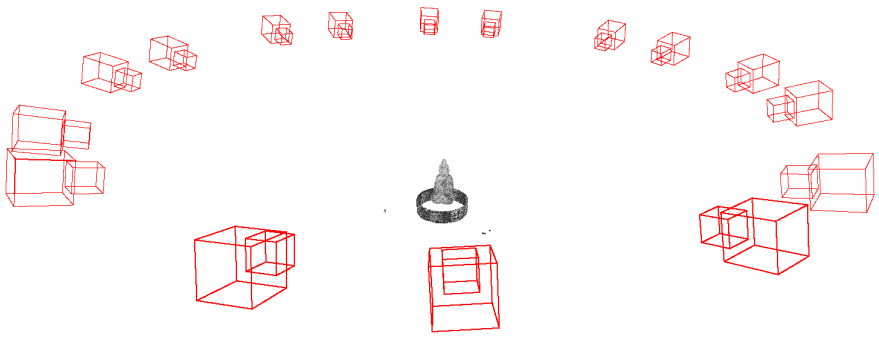


Figure 5.8.: *Visualization of a static scene and the 16 camera views used for the reconstruction. The result was obtained from 8 partial reconstructions, each with two camera views, which were subsequently aligned.*

5.6. Bundle Adjustment

In almost every auto-calibrated computer vision application that requires high-precision calibration, bundle adjustment ([103], [101], [163], [3], [37], [89], [178], [90]) is performed as a final step in order to refine the estimated camera matrices. This involves refining both intrinsic and extrinsic parameters of all devices used and correspondingly mapped 3D points in a joint optimization. Minimizing the re-projection error while taking all parameters into account is the usual choice that has been proven to be very efficient for a long time:

$$\underset{\mathbf{P}_i,\mathbf{X}_n}{\operatorname{argmin}} \sum_i \sum_n \|\mathbf{P}_i\mathbf{X}_n - \mathbf{x}_{i,n}\|_2^2 \tag{5.36}$$

To make the method more robust, an energy that is less sensitive to outliers can be used instead. One approach, described by Engels *et al.* in [37], is to assume that the re-projection errors satisfy a *Cauchy distribution* with a chosen variance σ. Therefore, the minimization problem simply changes to

$$\underset{\mathbf{P}_i,\mathbf{X}_n}{\operatorname{argmin}} \sum_i \sum_n \ln\left(1 + \frac{\|\mathbf{P}_i\mathbf{X}_n - \mathbf{x}_{i,n}\|_2^2}{\sigma^2}\right). \tag{5.37}$$

In many cases, it is necessary, to also correct projected points $\mathbf{x}_{i,n}$ for distortions, in order to achieve further improvements. During this whole optimization process, a wide range of possible values is searched for a large number of parameters. This is an extensive calculation and time consuming procedure. Even though the state of the art has generally been given for some time by Lourakis and Agyros [103], [101], to this day there is still research carried out on pre-conditioning methods. Approaches, such as Katayan's *et al.* recently published work [85], still aim to speed up this famous optimization task.

A statement that is generally true, is that the better the initial calibration information, the faster the convergence. After applying the calibration method presented in this chapter, the parameters are already of very high accuracy, so that a few steps of bundle adjustment are sufficient to reduce the calculated re-projection error below 0.1 pixels in all our tests, which is sufficient for almost all conceivable applications.

To illustrate the result of the presented calibration procedure, Figure 5.8 shows the result for a static scene reconstructed and calibrated from 16 camera perspectives.

5.7. Evaluation

In order to demonstrate the advantages and improvements of the presented methods, the critical sub-steps of the calibration are independently evaluated.

First, the accuracy of the calculated epipolar geometry using the combination of the epipolar error and the trifocal error, presented in Section 5.3, is investigated. The new method is compared to current approaches that minimize either the epipolar error or the trifocal error. The advantage of the combined approach becomes apparent in terms of both epipolar and trifocal

errors achieved. When applied to a large data set, a pattern becomes visible which suggests a generally advantageous value for the weighting parameter τ. Improvements can also be seen with respect to the accuracy of the calculated relative rotations and the behavior of the back-projection error when noise occurs.

In the next section, the intrinsic calibration from epipolar geometry is considered (Section 5.4) and compared to the state of the art.

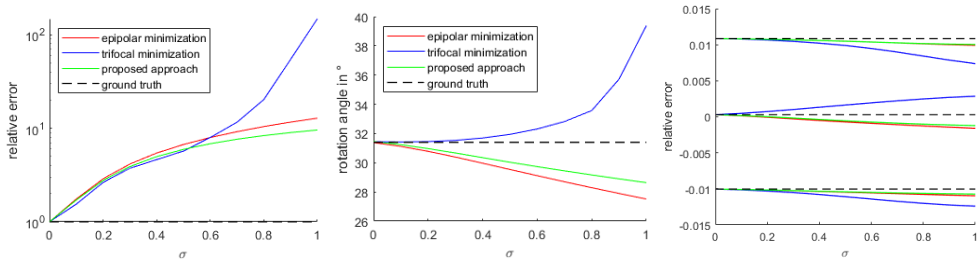
The comparisons start with the minimum case of two cameras whose focal lengths are determined. In contrast to the established method, no problems occur with the presented approach even in particularly demanding situations with very different camera models. In the case of three devices, which is an important scenario also in the field of reconstruction, the greatest gain in stability of the method becomes visible. With the presented method it is possible to stably calibrate a projector together with two cameras, which makes the application particularly powerful for common structured light setups. Even in extreme situations where extremely poor fundamental matrices are used as input, the method works more stably. Finally, for the sake of completeness, the case with 4 or more cameras is also covered.

5.7.1. Epipolar Geometry

In this section, the proposed method from Section 5.3 for determining the epipolar geometry is evaluated to assess its benefits. The effect of regularization parameter τ is examined on real and synthetic data sets. Note that the method minimizes the epipolar error for $\tau = 0$, while $\tau \to \infty$ yields a pure trifocal minimization. Further note, that Li *et al.* mention in [94] that the minimization of the epipolar energy term in several ways, like for example in the gold-standard approach [62], is still seen as the common approach and the trifocal error is often not practicable, which may be due to its noise susceptibility.

Test Data Both synthetic and real datasets were acquired to evaluate the proposed method. For all datasets, up to 300 correspondences were carefully selected and validated, in order to guarantee absence of strong outliers. In both cases the same setup, comprised from two cameras and a projection device, has been used. Such a setup is quasi standard for most active scanning solutions and is suitable for evaluation in the contexts of active as well as passive methods.

For the real setup, wDSLR cameras with a resolution of 6M pixels and a full-HD projector were used. The synthetic setup reproduces the real setup. It was modelled with *Unity* [2] and has different principal points, various degrees of distortion, and different focal lengths. Apart from that, the synthetic model is a perfect pinhole camera. In order to assess the robustness of the proposed calibration procedure, the synthetic datasets were artificially degraded. To this end, multiple datasets with different levels of positional noise added to the correspondences (Gaussian with $\sigma \in [0, 2]$ and $\mu = 0$) were derived. Based



(a) Back-projection error **(b)** Rotational error **(c)** Distortions

Figure 5.9.: *Evaluation on synthetic data for an increasing level of Gaussian noise. Back-projection errors (left), angular errors (middle) and distortion coefficients (right) for epipolar, trifocal and the proposed minimization method.*

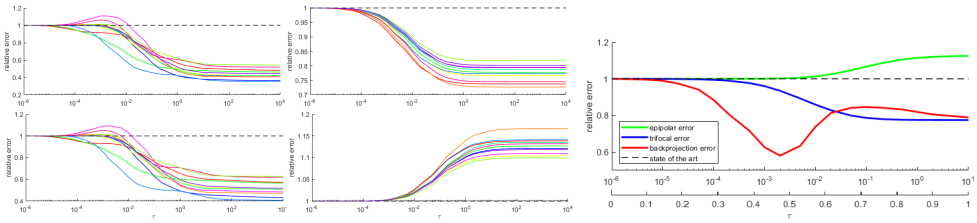
on this data, the back-projection error as well as the angular error with respect to the cameras were computed (see Figure 5.9).

For evaluation of the real data, a calibration with several values for the regularization parameter $\tau \in [10^{-6}, 10^{4}]$ was calculated. After computing the calibration, the remaining epipolar as well as trifocal error were estimated (see Figure 5.10).

Results Using the synthetic data we observe that:

- For noise-free data the selection of τ is irrelevant, since both error terms converge robustly to the global minimum (see Figure 5.9 for $\sigma = 0$). Also the proposed combination provides the expected result.
- For slightly noisy data ($\sigma < 0.5$) the trifocal error clearly outperforms epipolar optimization. Although the proposed method is not better than the trifocal minimization, it consistently outperforms the more robust epipolar error (see Figure 5.9).
- For very noisy data ($\sigma > 0.5$), minimizing the trifocal error does not provide useful results. In this case, minimizing the epipolar error is much more robust and yields stable results. The same is true for the proposed method, which consistently provides significantly better results than the state of the art, given by pure epipolar minimization.

The proposed combination of both errors improves the state of the art (i.e. epipolar optimization) for all cases with $\sigma > 0$. Figure 5.9 gives an impression of the relation between noise and errors. In terms of the back-projection error (left), the improvement is up to 30% with respect to the state of the art. In terms of angular error (middle), we observe an improvement of 1-2 degrees. This is equivalent to a 3D point position error of not less than 1.7cm assuming a baseline of 1m.



(a) Mean epipolar and trifocal errors for increasing τ **(b)** Average errors

Figure 5.10.: *Mean epipolar errors (a, top left, top right) and mean trifocal errors (a, bottom left, bottom right) computed from high quality (a, left) and noisy (a, right) real datasets. Average errors from real data sets with different noise levels (b).*

To evaluate the accuracy of the distortion parameters computed by the proposed method, distorted synthetic data has been generated using *Unity*. An example of distortion parameters $(-1 \cdot 10^{-2}, 5 \cdot 10^{-4}, 1.1 \cdot 10^{-2})$ that have been estimated by the method is given in Figure 5.9 (right). Unfortunately, the parameter space under investigation is too large to allow a comprehensive evaluation. Therefore, the focus was on evaluating the first coefficient of the distortion model, since it dominates the others and reflects most of the distortion caused by lenses. Gaussian noise of $\sigma \in [0, 1]$ was applied to investigate the influence of measurement errors. Figure 5.9 (c) shows the estimated radial distortion coefficients for increasing values of σ. It can be observed that all methods provide useful distortion parameters. The proposed method provides consistently superior parameters, although the improvement may be marginal.

Choice of the Regularization Parameter

In Figure 5.10 all errors are given relatively to the state of the art ($\tau = 0$), i.e. pure epipolar error optimization. On the left side (a), the behavior of the mean epipolar error (top) and the mean trifocal error (bottom) is visualized. In total 100 uncorrelated datasets have been investigated, using 300 high quality correspondences each. By emphasizing the trifocal term, both the trifocal error as well as the epipolar error are improved in the case of low noisy data.

Figure 5.10 (a, right) visualizes the behavior of the errors after adding weak Gaussian noise ($\sigma = 0.3$). It can be observed, that the minimization of the trifocal error in the presence of noise does not lead to lower epipolar errors, but the epipolar error increases significantly ($> 10\%$). Hence, adding a small amount of noise dramatically reduces the dependency (see Figure 5.1) between the epipolar and the trifocal error.

Since neither of the two errors is to be preferred in principle, a combination of both errors is well reasoned. A suitable measure for the calibration quality is the back-projection error. Applying the proposed method, similarly to the previous section, to 100 uncorrelated datasets with different noise levels, results in average errors visualized in Figure 5.10 (b). Minimal back-projection errors were achieved with a selection of $\tau = 10^{-3}$ over a large number

of datasets. Therefore, it can be assumed that τ is a constant, rendering the proposed method quasi parameter-free. Note that trifocal errors are usually much larger than epipolar errors, a value of $\tau = 10^{-3}$ leads to nearly equal influence of both errors to the minimization. In Figure 5.10 (b) the original value of τ according to (5.6) is shown in the upper ordinate. The lower ordinate shows τ after a transformation into the interval $[0, 1]$ using normalized energies.

5.7.2. Intrinsic Calibration

Both the method [105] and the proposed approach, are based on Kruppa's equations. These equations provide two independent constraints for each fundamental matrix. Therefore, the number of computable parameters is determined by the number of devices (see Table 5.1). For two devices, only two parameters can be estimated based on the single fundamental matrix between the views. This case is the most basic and most frequently examined system setup. With four or more devices, the problem of intrinsic calibration is well defined and theoretically all parameters can be estimated. Nevertheless, even the calibration of four devices can be a challenge in practice. A particularly interesting case, which motivated this course, is the use of three devices, such as two cameras and a projector, as found in most active scanning setups. For all three devices, the focal lengths can be calculated. With the remaining constraints, the principal point of the projector can be estimated, which is usually far off the image center.

For the evaluation, three cases are considered, i.e. two, three and four devices. In order to investigate the stability of the methods, probability maps are calculated that visualize the convergence chances for different initializations and thus represent the convergence regions of the methods. To calculate these probability maps fixed setups with two, three and four devices and fixed extrinsic and intrinsic parameters are used. A total of 16 different scenes were recorded with these setups. The scenarios were selected in such a way that they cover a multitude of different practical application scenarios. From the different scenes, fundamental matrices are computed using the technique in Section 5.3. The matches used for the computations were previously validated to avoid falsification by strong outliers. Consequently, the resulting

# Devices	# Basic Equations	Computable Parameters
2	2	f_1, f_2 or x_{p_l}, y_{p_l}
3	6	f_1, f_2, f_3, x_{p_1}, y_{p_1}
4	12	f_l,x_{p_l}, y_{p_l}, $l = 1, ..., 4$
⋮	⋮	⋮
C	$C(C-1)$	f_l,x_{p_l}, y_{p_l}, $l = 1, ..., C$

Table 5.1.: *Overview of degrees of freedom in terms of the number of devices and useful calibration parameters that can be computed.*

fundamental matrices of each setup approximate exactly the same epipolar relations with uncorrelated numerical errors because they are computed from different matches received from totally different scenes. Applying the methods under investigation on a fundamental matrix for all combinations of initial focal lengths $f_1, f_2 \in [1, 10000]$ leads to a binary map, which indicates whether the method converged or not. The binary maps of all the fundamental matrices have been combined into probability maps depicted in Figures 5.11, 5.12 and 5.13. Therefore, the percentage at which convergence has been achieved color-codes the maps. Green color indicates a very high probability of convergence, while red indicates either divergence or convergence to an incorrect value. Yellow depicts regions with approximately 50% chance to converge to the correct value. Further interpolated values between green and red indicate corresponding probabilities.

Since the probability maps are not dependent on individual scenes, correspondences, or fundamental matrices, they are meaningful indicators for the convergence behavior of the procedures.

In the following, the focal lengths are given in terms of sensor pixel size. For typical devices, a plausible range would be in $[500, 15000]$. Depending on the sensor size, this would correspond to approximately $[18mm, 50mm]$ for a standard camera. The principal points are given in terms of image pixel size, depending on the resolution.

Two-View Focal Length Estimation

In the case of two cameras, the principal points are usually assumed to be in the image centers. Therefore, in most cases only the focal lengths are computed. Bougnoux [17] gave a famous formula to calculate the focal lengths directly:

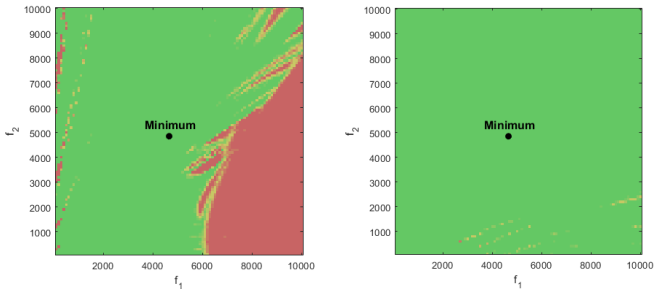
$$f_1 = \sqrt{-\frac{\mathbf{c}_{p_2}^\mathsf{T}[\mathbf{e}_2]_\times \tilde{\mathbf{I}}\mathbf{F}_{12}\mathbf{c}_{p_1}\mathbf{c}_{p_1}^\mathsf{T}\mathbf{F}_{12}^\mathsf{T}\mathbf{c}_{p_2}}{\mathbf{c}_{p_2}^\mathsf{T}[\mathbf{e}_2]_\times \tilde{\mathbf{I}}\mathbf{F}_{12}\tilde{\mathbf{I}}\mathbf{F}_{12}^\mathsf{T}\mathbf{c}_{p_2}}}, \quad f_2 = \sqrt{-\frac{\mathbf{c}_{p_1}^\mathsf{T}[\mathbf{e}_1]_\times \tilde{\mathbf{I}}\mathbf{F}_{12}^\mathsf{T}\mathbf{c}_{p_2}\mathbf{c}_{p_2}^\mathsf{T}\mathbf{F}_{12}\mathbf{c}_{p_1}}{\mathbf{c}_{p_1}^\mathsf{T}[\mathbf{e}_1]_\times \tilde{\mathbf{I}}\mathbf{F}_{12}^\mathsf{T}\tilde{\mathbf{I}}\mathbf{F}_{12}\mathbf{c}_{p_1}}}, \tag{5.38}$$

where $\mathbf{c}_{p_s}$ and $\mathbf{e}_s$ denote the principal points and epipoles of camera $C_s \in \{1, 2\}$ in homogeneous coordinates. Moreover, $[.]_\times$ denotes again the cross-product matrix and $\tilde{\mathbf{I}} = \mathrm{diag}(1, 1, 0)$ is the embedding of the two-dimensional identity matrix. Unfortunately, this formula fails in many practical situations. As already mentioned in Section 5.4.2, it often tries to intersect curves that are almost coinciding in many situations. This is likely related to situations close to the degenerated cases reported by Sturm in [153] and [152]. Although it is not well suited for auto-calibration in general, it can still be used as initialization for iterative methods in case it is not degenerated.

For the case with two cameras, Bougnoux's, Lourakis' and the proposed method are compared. Bougnoux's method is a direct one and therefore does not depend on initialization. Therefore, no region of convergence can be determined and visualized in the following. Having said this, Bougnoux's method failed in most cases during the tests (likely because of dependency on error-

prone epipole estimates), while the iterative methods could still converge when initialized accordingly.

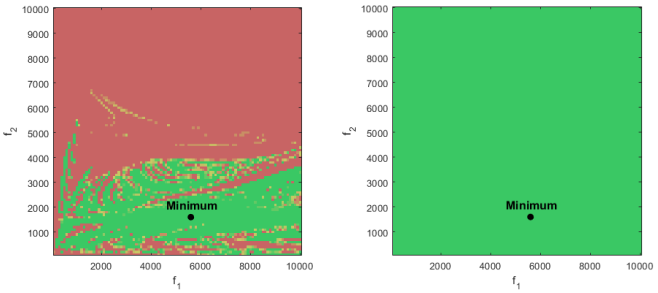
For the current investigation, the focal lengths of the devices were chosen to be approximately equal in order to resemble the practical case of manually adjusting the cameras. Despite that, exactly the same focal lengths would lead to a degeneration that would be perfectly captured by Sturm's method [154]. However, this special case rarely occurs in practice for setups including multiple devices. Inspecting Figure 5.11 for the two camera case with similar focal lengths, it can be observed that the proposed method converges for nearly all initializations, while Lourakis' method only converges in a region of radius of approximately 1500 pixels relative to the true solution.



(a) Lourakis method **(b)** Proposed method

Figure 5.11.: *Comparison of the convergence in the two-view case with similar focal lengths. Colors visualize the probability of successful convergence to the correct solution for different combinations of initial focal lengths. Left: Lourakis' method, right: the proposed method.*

Strongly Differing Focal Lengths In the case of strongly varying focal lengths, even more benefits can be achieved. Figure 5.12 shows the convergence probability map of a similar configuration as in Figure 5.11. While the method of Lourakis converges in a region with a radius of only 500 pixels, the proposed method converges in almost all cases.



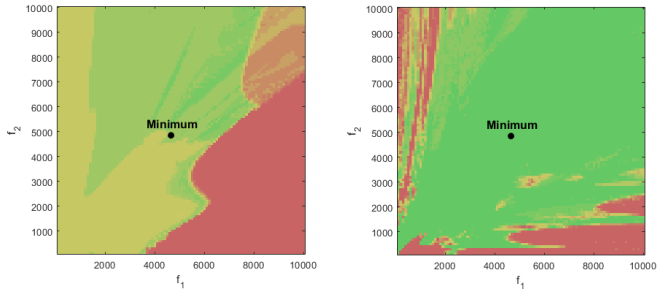
(a) Lourakis method **(b)** Proposed method

Figure 5.12.: *Comparison of the convergence behavior in the two-view case for dissimilar focal lengths. The axes represent the respective focal lengths. Left: Lourakis' method, right: the proposed method.*

Three-View Intrinsic Calibration

Perhaps, the most interesting case for the presented application is a three device setup. According to Table 5.1 the focal values plus the position of the principal point of one device can be estimated. This allows the calibration of setups consisting of two cameras and one projector, which is of practical importance, as it is common for modern structured light setups. In this case, it is assumed that the principal points of the cameras are in the image center, while their focus values can be very different. The projector is assumed to have a completely independent focal length and an extreme position of the principal point, usually near the image border.

Again a system setup with fixed extrinsics and intrinsics is assumed. Fundamental matrices are computed from 16 scenes similar to the previous test. Now that three devices are given, the respective probabilistic convergence maps with respect to focal length intializations would be three-dimensional. In order to achieve an expressive visualization in two dimensions, the focal length of the projector was initialized by $f_3 \in \{1, 10, 100, 1000, 10000\}$ and the resulting maps averaged. Figure 5.13 depicts the convergence regions for Lourakis' method on the left and the proposed method on the right. As can be clearly observed, Lourakis' method does not provide a secure convergence region, i.e. a region of focal length selections that converges for an arbitrary principal point.



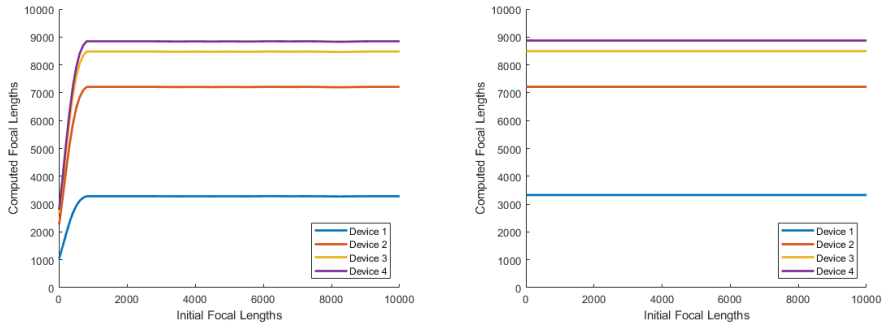
(a) Lourakis method **(b)** Proposed method

Figure 5.13.: *Comparison of the convergence behavior for focus optimization in the three-view case. The axes represent two of the three focal lengths, the third one is visualized as a mean projection along the third coordinate axis. Left: Lourakis' method, right: the proposed method.*

Multi-View Intrinsic Calibration

In the multiview case of four or more devices, the problem is much easier to solve. Theoretically, it is possible to fully calibrate all devices, including focal lengths and principal points. However, practice shows that Lourakis' method does not converge if the focal lengths are initialized far too small, while the proposed method converges in all situations.
However, for failure of Lourakis' method the focal lengths have to be so small that this case can be neglected (see Figure 5.14 for visualization).

In the case of five or more devices, the stability of the convergence of each procedure further increases. Due to the inherent difficulty of visualizing multi-dimensional data and the fact that both methods perform well in practice, respective visualization are omitted.



(a) 4 view case Lourakis method **(b)** 4 view case proposed method

Figure 5.14.: *Comparison of the convergence behavior in the four-view case (third Louraki's, fourth the proposed method) with extreme initializations of the principal points and the focal lengths outside the convergence regions of the methods. In the four-view case the newly proposed method always converges to the correct solution, while Lourakis' method may still fail.*

5.8. Conclusions

A stable and robust method for computation of accurate calibration of multiple views (i.e. cameras and projectors) based on their epipolar geometry has been proposed and evaluated. The procedure combines two improvements on auto-calibration and extends them to a complete calibration method. The procedure eliminates weaknesses of existing methods, especially in the presence of noisy data. A suitable regularization parameter τ has been estimated and fixed, so that the whole procedure can be assumed parameter-free. The optimal result is observed to be achieved when epipolar and trifocal errors contribute about the same amount to the calibration.

Contrary to the former approach, the presented method for intrinsic calibration from fundamental matrices converges to the global solution for nearly all reasonable initializations and enables the calibration of projectors and low quality devices. It therefore has a major impact on active scanning techniques that can thus be calibrated from scratch, including the active element. It has been shown that and why the applicability of the standard method on intrinsic calibration from fundamental matrices is subject to systematic limitations.

Chapter 6

Light-Resistant Pre-Alignment from Optical Flow

Contents

6.1. Introduction

3D reconstructions of objects as well as depth information of scenes play an increasingly important role in industry. Whether it is quality control in production or the recognition of the environment in autonomous driving, the number of applications is continuously increasing. Due to the simplicity of applicability, depth cameras are more and more used in parallel to flexible 3D scanners, and the availability of depth data for a wide variety of applications is steadily increasing. At the same time, the demand for scene understanding methods, represented by optical flow estimation, is constantly increasing, especially in the field of automation. Since in addition to images alone, more and more information is available, also the demand for higher quality scene understanding increases.

For the vast majority of applications, rigid scenes can be assumed and taken into account. And even for dynamic scenes, the optical flow can be approximated by rigid models if not too large motions of the camera or the environment are expected. This rigidity assumption can even guide the estimation of optical flow, whose accuracy can benefit from it. The simultaneous extraction of the rigid transformation between two subsequent frames is then also desirable. In this way, the method can be used for automatic alignment of point clouds in difficult scenarios, including large motion (fast driving cars) and large rotation (3D reconstruction, where often approx. 45° rotation between partial views occur), that yield strong shading changes.

The presented method will use an optical flow approach based on *PWC-Net* [156], that has been adapted in order to use data from texture images, normal maps and vertex maps simultaneously. This procedure is moreover combined with the extraction of rigid transformations, that are computed from the normal and vertex maps, that are warped by the predicted optical flow. The so predicted pose can mutually benefit from the coarse to fine strategy of the optical flow, which can find dense correspondences over the whole scene using a pyramidal approach, even in the presence of large motion. Textural, geometric and shading features are included, which partly compensate for each other's weaknesses (sparsity of normal and vertex maps, illumination susceptibility of the texture images). From the warped 3D information of the scene, the rigid transformation can be determined stably in a second step.

6.1.1. Motivation: Flow-Based Alignment

In order to compute the alignment of two subsequently reconstructed frames, usually robust and transformation invariant features (SIFT, KAZE, ...) are detected and matched between the frames. Robust and outlier resistant methods like RANSAC-based *PnP-solvers* are used to compute the rigid transformation between the views [40]. It is commonly known, that this approach, applied with some few good features only, results in way better alignments than using many worse features jointly. Modern deep learning approaches adopt this scheme and deliver competitive results on a wide range of data in real time.

The basis of all common feature-based methods is the *brightness constancy*

(a) SIFT Matches **(b)** Illuminated Scene **(c)** Overlap of Views **(d)** Flow Matches

Figure 6.1.: *(a) and (d) show matches based on SIFT features and optical flow. (b) shows the scene, which has been illuminated by a strong spot light, in a different color space that is more visual to human perception. This more clearly visualizes the different shadings of the object, which is the reason for the failure of the common method based on SIFT features. (c) shows overlapping regions of subsequent scans. Even a rotation of approx. 45° yields a large overlap of more than 80%.*

assumption, which expects that the appearance of the object does not change significantly from one frame to another. This assumption is fulfilled for a large number of applications, especially in scenarios, in which the camera moves smoothly through a scene or an object undergoes slow motion. If, on the contrary, the direction of the light incidence changes, the shading of the scene also differs dramatically and the brightness constancy assumption gets violated strongly. This leads to a very probable failure of the standard methods based on this requirement, especially in the following situations:

- Outdoor scenes where lighting conditions can change suddenly. This can occur from direct sun light, as well as indirect light reflections from other objects.
- Moving objects, especially rotating ones, inevitably change the relative direction of light incidence. This leads in particular to considerable difficulties in the application area of 3D reconstruction, where the object is often rotated in order to capture it from all sides.
- Driving cars in the dark may cause strong shading differences in the captured images of the environment. Visible elements in the scene are illuminated by the car's headlights. These light sources move together with the car through the scene, which may yield strong variation of the direction of light incidence.

In order to illustrate this problem and investigate it, a setup with a static light source, a static camera and a rotated object is considered. Figure 6.1 (a) shows how the standard approach based on SIFT matches fails, due to different light incidence. Figure 6.1 (b) shows the scene in a different color coding, that maps the grayscale values to a color scale that is more visual to human perception, which makes the different shading become obvious. While the features in the scene change in appearance, it can still be assumed that a significant portion of the scene overlaps in the different views. In the important case of object

rotation in 3D reconstruction, our research shows that in the vast majority of cases a typical rotation of 45° still yields more than 80% overlap of the scene's content. Figure 6.1 (c) visualizes the overlapping areas of the two views. Optical flow methods can benefit from this in turn, as they view and match the motion as a whole, using pyramidal approaches. Finally, Figure 6.1 (d) shows correspondences determined using an optical flow method, as introduced in the following. The correspondences do contain noise and smaller errors, especially in feature-poor regions. They are nevertheless capable of predicting stable orientations of the object, significantly more stable ones than feature-based methods.

6.2. Related Work

Optical flow estimation is a well-known problem in applied machine vision and has wide spread use cases in industrial applications such as robotics, autonomous driving and quality control. The task is to determine dense motion at pixel level between image pairs as accurately as possible. Starting with the method of Horn and Schunck [69], variational methods were the state of the art for a long time. Since the problem itself is an ill-posed problem, further assumptions have to be made on the flow field, which led to a multitude of different methods that use the most diverse regularization procedures to make the problem solvable according to the specific application. In recent years the problem of *optical flow* estimation increasingly expanded to the problem of *scene flow* estimation, which deals with the 3D motion of scene points in space, whereas optical flow was limited to 2D point motion on the image plane. Based on the variational approaches for optical flow, a number of variational scene flow methods have been developed. Most of them use rectified stereo image pairs as input and thus estimate scene flow with different regularization methods or partial rigidity assumptions ([21], [71], [78], [96], [11], [128], [187], [41]). At the same time, methods that determine the scene flow directly from RGB-D data have been developed. With an increasingly number of depth sensors that became available, this approach is quite justified. Several variants of methods handle this case ([93], [55], [65], [136]).

The appearance of *FlowNet* [31] revolutionized the field of optical flow estimation. It became possible to treat the problem in real time with the help of *convolutional neural networks (CNNs)*. In contrast, the variational methods were extremely time consuming and computationally expensive. A higher accuracy at the expense of a much larger network was subsequently achieved with *FlowNet2* [76]. This was followed by the release of *PWC-Net* [156], which uses warping layers at different levels of an image pyramid, representing the current state of the art that is in addition much smaller than the previously released *FlowNet2*. Based on *PWC-Net*, Saxena *et al.* have presented a method for estimating scene flow from rectified stereo image pairs.

In addition, they handle occlusions within the forward pass. Previous methods required at least one forward and one backward warping to stably detect occlusions ([75], [115], [168]). Other approaches even tackle the task by itera-

tive approaches such as [74]. Besides that, a large amount of research currently focuses on either making networks lighter ([73], [72]) or on training networks without ground truth through un- or self-supervision ([99], [82], [194], [79]). A survey on variational as well as CNN-based optical flow methods can be found in [164].

Similar to earlier approaches in the variational path, methods that extract the scene flow directly from RGB-D data also evolved over time. Qiao *et al.* showed how scene flow based on *FlowNet* can be improved by fusion with features of depth data extracted in an extra network pass. Based on *PWC-Net*, Rishav *et al.* [138] use depth data from a Lidar sensor to determine the scene flow. In doing so, they account for the lower resolution of Lidar data using appropriate reliability weights from [35]. In general, scene flow networks based on RGB-D data show poor performance for outdoor scenes, due to range limitations of the sensors. A number of approaches attempt to address this issue ([168], [175], [177], [196]). Since the omission of active components removes the range limitations, but is accompanied by a loss of quality of the depth information, we will nevertheless restrict ourselves to this limited case. We are content with the scene flow within the sensor limits, since it is sufficient for an overwhelming number of practical applications, where the limits of the sensor can be planned accordingly.

In order to predict the pose of an object, a long time RANSAC approaches using explicit pose estimates based on the singular value decomposition were used. In recent years, first deep learning approaches predicted the pose directly using neural networks. Kendall *et al.* [88] use in their *PoseNet* several convolutional layers, followed by linear layers to directly predict rotation and translation from RGB images. This way, they were the first to solve the problem of camera re-localization in static scenes by a deep learning approach. A few years later Vijayanarasimhan *et al.* [166] extend this principal in *SfM-Net* in order to predict simultaneously the rigid transformations and the depth of the scene. They basically adopt the principals of the famous *Structure-from-Motion* pipeline to a deep learning framework. In parallel Zhou *et al.* [193] developed a related model and showed how to train it in an un-supervised manner.

Finally, there has been a row of methods for direct point cloud registration with deep learning. Some of them replace parts of the standard strategies by deep learning methods and some try to replace the full pipeline. A large number of different approaches, correspondence-based and correspondence-free, are reviewed in [191] and [167].

Related to the presented work, [50] and recently [131] introduced variational and CNN-based methods for flow-aided pose estimation, based on fulfilled brightness constancy assumption. Nevertheless, an automatic and light resistant flow-based pose-estimation method, that works correspondence-free, and takes geometrical, textural and coherent scene motion into account has never been addressed before.

6.3. Light-Resistant Optical Flow

The *optical flow* between two images is understood as the displacements of the individual pixels from one to the other image. Determining the optical flow between images of a scene often serves the purpose of scene understanding, as it directly allows the analysis of a large amount of scene information:

- The optical flow between calibrated camera images from different perspectives of the same static scene allows theoretically to compute dense point correspondences and accompanying depth data.
- The optical flow between static camera images of a moving scene theoretically allows the analysis of scene motion and object tracking. If depth data is additionally available, the *scene flow*, i.e. the spatial movement of the points in the scene, can be calculated.

In the estimation of the optical flow between two consecutive images I_0 and I_1, a horizontal and a vertical displacement field (F_x^{01}, F_y^{01}) are calculated, mapping each pixel in image I_0 to its corresponding pixel in image I_1. The usual basis of the estimation is the brightness constancy assumption, which assumes that corresponding pixels have the same appearance in the different images:

$$I_0(x, y) \approx I_1(x + F_x^{01}, y + F_y^{01}) \tag{6.1}$$

Figure 6.2 shows image I_0 and besides image I_1, which has been warped by the optical flow F^{01}. Since the used optical flow has been computed from real data, the flow field is semi-dense and contains some masked pixels. Such errors will be addressed later on, where we will also show how to adopt filters to sparse, semi-sparse and mixed data. Instead of looking for exactly the same values between I_0 and I_1, filtered values are considered in a regional context in order to robustify the matching. Deep neural networks have proven to be extremely effective for this purpose. The current state of the art is given by *PWC-Net*, which will be briefly introduced in the following to serve as a basis for the subsequently presented light-resistant method.

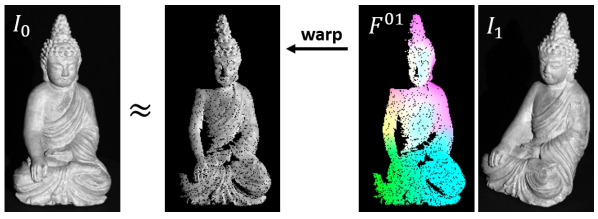


Figure 6.2.: *Image I_0 in comparison to image I_1 that has been warped by optical flow F^{01}. Assuming consistent brightness, these should be identical (ignoring masked pixels due to the semi-dense optical flow from real data). In case of strong rotations of the object the shading changes dramatically, which violates this assumption.*

6.3.1. PWC-Net

PWC-Net combines classical techniques such as a pyramidal approach, warping and correlation to create a highly effective network for optical flow estimation. The input images are passed through a pyramid of convolutions which extract rotation- and translation-invariant features at different levels of the receptive field. The number of hierarchies should be adapted appropriate to the image resolution. By successively halving the resolution in each step, the procedure should cover almost the entire scene in the filter of the last stage. From the lowest level, cost volumes based on extracted features are established from which the optical flow is effectively predicted. These flows are refined upwards with each level, incorporating new features of the current level and the flows and more global features from previous levels. By warping the data using the previous flow, the search space is significantly reduced and even large displacements can be treated and predicted with this comparatively small network. Figure 6.3 depicts the architecture of the network. Each prediction block consists of a cost volume for flow prediction and is fed with the corresponding layer in a U-Net structure, in order to predict a flow field in full resolution. Note that the standard network presented by Sun *et al.* in [156] predicts the optical flow up to the second last level and afterwards refines the resulting flow by a context-network as a post-processing. This results in a final optical flow whose resolution is only $\frac{1}{16}$ of the input images' resolution. Instead of up-sampling by variational methods, we go for two additional texture-guided up-sampling steps within the network, in order to provide full resolution optical flows within a single training routine.

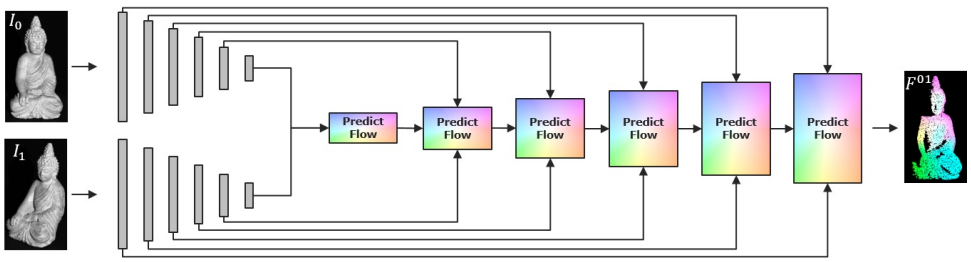


Figure 6.3.: *Sketch of the PWC-Net architecture. The input is convolved by multiple layers and the optical flow is predicted starting from the lowest level upwards in a U-Net structure. In each level, the layers of I_1 are warped towards the layers of I_0 in order to provide initial flows from previous lower levels. With this pyramidal approach also large flows are predictable with quite small filter kernels.*

6.3.2. INV-Net using Images, Normals and Vertices

Classical *PWC-Net* uses texture images only. Unfortunately, for the investigated use case these texture images may be disturbed due to shading changes, resulting from rotations of the object or light position changes, which would make the network likely to fail due to a violated brightness constancy assump-

tion (see Figure 6.1). In many situations, where depth data is available, a lot of additional information can be provided to the network, that is invariant under the shading effects related to light changes or object rotations:

- Texture images I_0 and I_1 that underlay shading effects. Nevertheless, they provide full and dense data, which can deliver local context.
- Depth maps D_0 and D_1 that store the relative geometrical information of the scene, light- and shading-invariant, with respect to the camera center. Due to measuring errors there may be outliers or data-less pixels, resulting in semi-dense depth maps.
- Vertex maps V_0 and V_1 that store the spatial information of the scene, light- and shading-invariant in three channels of a map in image resolution. They are computed from the depth maps and the available camera calibration in order to store the geometrical information calibration independent. Therefore, they are similar to the depth maps semi-dense maps with masked pixels. Moreover, they are structured representations of point clouds, that allow to perform neighboring operations on 3D data in 2D space, which yields large advantages in the following approach.
- Normal maps N_0 and N_1 that store spatial information of the surfaces in the scene. They are related to partial derivatives of the 3D vertices and do not underlay scaling and translation bias. They are in a specific range and responsible for a large amount of shading features of a scene (where standard methods based on fulfilled brightness constancy assumption get a large amount of information from), without being disturbed by the light changes. They can be directly computed from the vertex maps, using the topological information given by the image grid (see Chapter 8). Unfortunately, they thus also inherit the semi-density from underlying vertex maps.

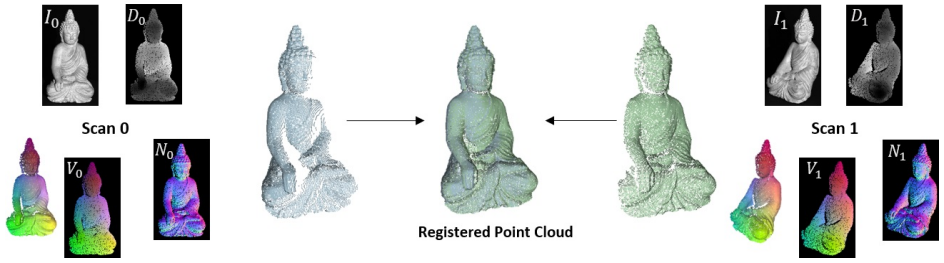


Figure 6.4.: *Possible input that is available to the task of light resistant optical flow estimation and subsequent pose prediction. In addition to texture images, there are depth maps, vertex maps, point clouds and normal maps available. The depth maps as well as the vertex maps contain geometrical information. Since the vertex maps are independent of the calibration it is the preferable choice for the presented method.*

Figure 6.4 sketches the basic problem of finding a light resistant pose estimation from all the available input. The first task is to find a light resistant

high quality optical flow from this large amount of input data. Depth maps as well as vertex maps store the spatial information of reconstructed surface points. Since they are somehow interchangeable we use the vertex maps only. This way the method becomes independent of the intrinsic calibration at the cost of a higher amount of data that needs to be processed. Figure 6.7 (left part) sketches the basic network that takes features from *images* (textural features), *normal maps* (shading features) and *vertex maps* (geometrical features). Thereby we follow the basic principal of *PWC-Net* but run the different input through separate feature pipelines and set up independent cost volumes, that contribute to the flow prediction. All features are processed as in [156] and fed to the pose prediction in each layer. This way the network learns to treat the feature appropriate and to achieve advantages from all. Figure 6.5 depicts the prediction procedure in each layer, except the first one, where only the cost volumes are used for initialization of the flow.

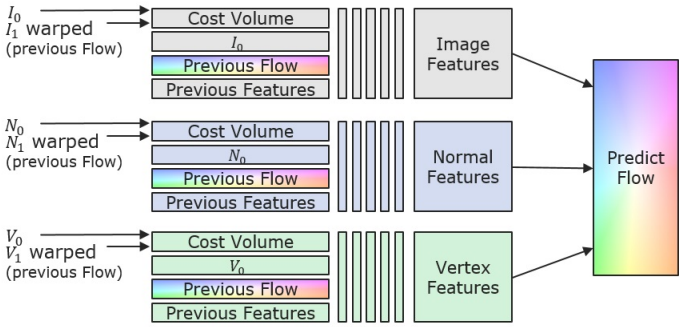


Figure 6.5.: *Flow prediction architecture in each layer (except first one). Features of images (texture), normals (shading) and vertices (geometry) are extracted separately and jointly fed to the prediction module.*

Normalized Convolutions In order to take into account the semi-density of the vertex maps and the normal maps, the convolutions, leading to the first layer are replaced by *normalized convolutions* as introduced by Eldesokey *et al.* in [36]. Using the following slightly changed convolution procedure, the known masks can be used to ensure that data-less pixels do not contribute to the convolution with respect to neighbored pixels. Suppose, we are given a signal $\mathbf{A}$ to be convolved with a filter kernel $\mathbf{K}$. Further assume that the measurements of the signal $\mathbf{A}$ are of varying quality with a confidence measure $\mathbf{W}$ of the same size as $\mathbf{A}$ having values between 0 and 1 to describe these uncertainties. It is desired to use the confidence measure as a weighting of the entries of $\mathbf{A}$ during convolution to ensure that reliable measurements have a higher influence on the convolution signal than inferior measurements or missing data for certain points. For this purpose, each summand within the convolution is weighted accordingly and divided by the sum of the weights to ensure the normalized character of the convolution. In detail, the normalized convolution of signal $\mathbf{A}$, convolved with kernel $\mathbf{K}$ and weighted by confidence

$\mathbf{W}$ around data point $[n]$ is given by

$$\big(\mathbf{K} * \mathbf{A}\big)_{\mathbf{W}}[n] = \frac{\sum_m \mathbf{K}[m] \cdot \mathbf{W}[n-m] \cdot \mathbf{A}[n-m]}{\sum_m \mathbf{K}[m] \cdot \mathbf{W}[n-m]} = \frac{\big(\mathbf{K} * (\mathbf{W} \odot \mathbf{A})\big)[n]}{\big(\mathbf{K} * \mathbf{W}\big)[n]}\,, \tag{6.2}$$

where $\odot$ denotes the element-wise *Hadamard-Product*. In order to avoid influence of missing pixels, a binary mask, that contains zeros in case of missing data and ones otherwise, can be fed to the convolutions as confidence $\mathbf{W}$.

Consistency Assumptions Similar to the brightness constancy assumption given in Equation (6.1) the following consistency assumptions hold true for normal vectors and vertices of rigid scenes:

$$N_0(x,y) \approx \mathbf{R} N_1(x + F_x^{01}, y + F_y^{01}) \tag{6.3}$$

$$V_0(x,y) \approx \mathbf{R} V_1(x + F_x^{01}, y + F_y^{01}) + \mathbf{t} \tag{6.4}$$

Figure 6.6 visualizes the consistency relations for normal vectors and vertices. While the pixels of the warped normal map coincide with the reference normals up to a rotation matrix $\mathbf{R}$, the vertices coincide up to rotation $\mathbf{R}$ and a translation vector $\mathbf{t}$. These relations will be essential later on, in order to extract the rigid pose from the given optical flow. A very important result of our research is, that features, computed from filtered normal and vertex maps allow for computation of accurate optical flows. This means, the standard approach for feature extraction from images (as used in *PWC-Net*), is suitable to compute rotation- and transformation-invariant features from normal and vertex maps, as well.

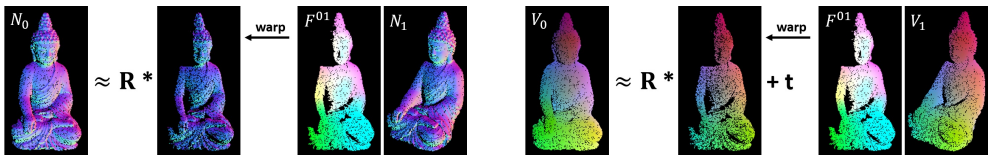


Figure 6.6.: *Normal maps and vertex maps that have been warped by optical flow F^{01}. Assuming rigid scenes, normal vectors should be identical up to a rotation, vertices up to a rotation and a translation.*

6.4. Pose from Warped Normals and Vertices

Several research papers have already shown that it is possible to predict the relative pose of two views of a scene using neural networks. Usually, features are detected, matched, outliers are rejected and then passed through a series of layers in order to get representative feature vectors. Finally, as introduced in [88], the parameters of a relative transform (consisting of a translation and a rotation) are predicted jointly using at least two fully connected layers.

In the previous section, a light-resistant optical flow has been computed by *INV-Net*. Based on this, it is not necessary to search for matches in the

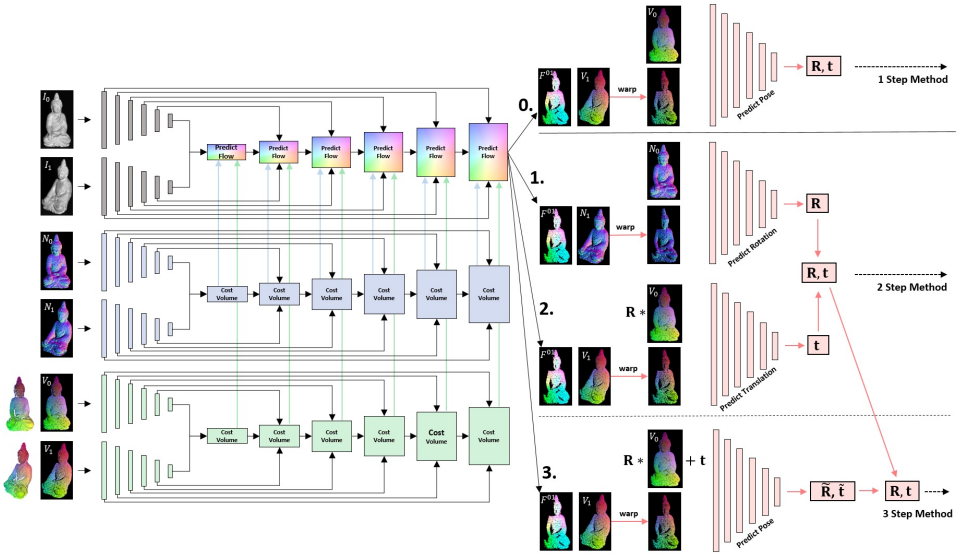


Figure 6.7.: *Architecture of Flow2PoseNet. The left part of the network aims to predict accurate optical flow from images, normal- and vertex-maps, using textural features from images, shading features from normal vectors and geometrical features from vertices in order to predict accurate and light resistant flow fields. The pose of the rigid scene is computed in three steps from the warped normal- and vertex-maps. The first step predicts the normal vectors from the warped normal-maps. The second step predicts the translation from the warped and rotated vertex-maps. The third step predicts a correction transformation to refine the predicted rotation and translation incrementally.*

entire image. Considering images, normal maps, and vertex maps from two views, that have been warped towards one reference frame using the computed optical flow, the data at each pixel-position theoretically matches densely. It should be mentioned, that there still might be many erroneous and inaccurate regions in the flow field, especially in feature-poor regions, where the flow is mainly interpolated. Nevertheless, previous work has shown that in general more accurate poses are estimated if only a few good features are used for the calculation, instead of many less good ones. This idea is implemented by an additional feature extraction from the warped normal and vertex maps for the pose prediction sub-network. It should be noted that in areas where good features for the pose estimation are found, usually also a good optical flow is available. In a way, both the optical flow and the subsequently calculated pose are based on the identical good features. Nevertheless, in the case of low quality features, as is the case with texture-poor and smooth surfaces, or even many false features due to light changes, we benefit from the more general information of the dense flow field.

In order to obtain best poses from the warped vertex and normal maps, we investigated two different approaches (*1 Step Method* and *2 Step Method*), that have led to the conclusion that a *3 Step Method* as combination of both performs best, as it compensates for the weaknesses of each method.

1 Step Method This approach uses the concatenated warped vertex maps to extract jointly rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ that align the vertex maps rigidly. The relation is based on consistency assumption (6.4). Note that after warping, the matching vertices are theoretically placed at the same location in the concatenated input. Due to convolutional layers the network is able to extract reliable locations, where a more accurate optical flow has been provided. The basic structure is shown in Figure 6.7 at branch **0.** on the right.

2 Step Method This approach uses two steps to predict rotation and translation individually by two separate networks. Following the consistency property of Equation 6.3, the warped normal map N_1 and the reference normal map N_0 are related by a rotation matrix $\mathbf{R}$ only. In a first step, this relative rotation $\mathbf{R}$ is predicted by stacking N_0 and the warped N_1 to processing them through several convolutional layers, followed by two fully connected layers in order to predict an optimal rotation with respect to the normals vectors.

Based on the third consistency property of rigid transformations, given in Equation 6.4, the translation $\mathbf{t}$ is predicted from the warped vertex map V_1, that has been rotated by matrix $\mathbf{R}$ and the reference vertex map V_0. Rotation matrix $\mathbf{R}$, from the previous step, has been applied in order to get dependency on the translation vector $\mathbf{t}$ for this inference step only. The structure is again shown in Figure 6.7 at branches **1.** and **2.** on the right.

3 Step Method Rotation and translation are two fundamentally different operations that have a strong influence on each other. The smaller a rotation, the better it can be approximated linearly. Unfortunately, the joint extraction as in the *1 Step Method* may yield inaccuracies in case of large rotations. In these situations, it may be beneficial to extract them separately like in the *2 Step Method*. Nevertheless, small rotational errors, from the first step of this approach influence the predicted translation from the second step.

The idea of the *3 Step Method* is to first apply the *2 Step Method* to pre-align the vertex maps. In a third step a correctional rotation matrix $\tilde{\mathbf{R}}$ and a correctional translation vector $\tilde{\mathbf{t}}$ are jointly predicted from the warped and pre-transformed vertex map $\mathbf{R}V_1 + \mathbf{t}$ and reference vertex map V_0. The final pose $P = (\hat{\mathbf{R}}, \hat{\mathbf{t}})$, as visualizes in Figure 6.8, is then given by:

$$\hat{\mathbf{R}} = \tilde{\mathbf{R}}\mathbf{R}, \qquad \hat{\mathbf{t}} = \tilde{\mathbf{R}}\mathbf{t} + \tilde{\mathbf{t}} \tag{6.5}$$

For extracting this correctional transformation the *1 Step Method* is used. This is beneficial, since the correctional rotations are usually small, which makes it possible to predict the rotation and the translation jointly in order to avoid weaknesses of successive prediction as in the *2 Step Method*. The structure is again depicted in Figure 6.7 at branches **1.**, **2.** and **3.** on the right.

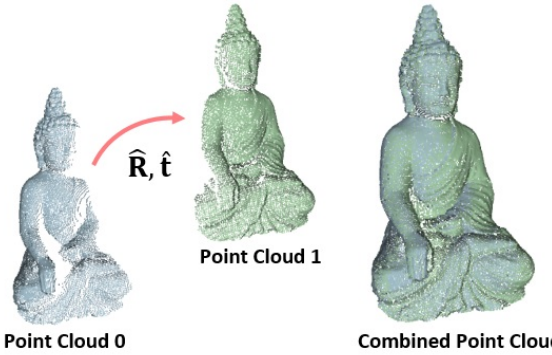


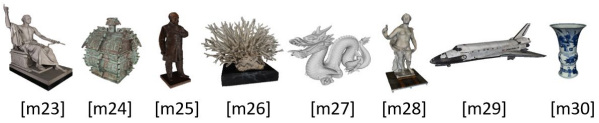
Figure 6.8.: *Point clouds of the two exemplary views. The resulting transformation $P = (\hat{\boldsymbol{R}}, \hat{\boldsymbol{t}})$ aligns the point cloud of the first view to the one of the second view. The registered combined point cloud is shown besides.*

6.5. Datasets and Data-Processing

There is already a number of public datasets for optical flow estimation (*Flying Chairs*, *Sintel*, *Kitti*, *Flying Things3D*) as well as for pose estimation and odometry (*Kitty Odomety*, *3D Match*, *ModelNet14*, *ShapeNet*). Unfortunately, only datasets that provide both images and depth data are suitable for the proposed investigations. Given the depth map and the camera calibration, the required normal maps can be approximated by practical methods, such as introduced later on in Chapter 8 and are thus not prerequisites. Therefore, for the evaluation of the estimated flow fields and the inferred poses, the established *Kitty Odomety* dataset will be used later on. It should be mentioned, that the included scenes do not reflect the main application area for the development of the method, since they involve quite small rotations that barely show shading differences due to movement of the camera instead of the scenes themselves.

Nevertheless, for the task of rotating objects, ground truth data of both optical flow and scene pose are required for training the presented network. In addition, it is advantageous to be able to use absolutely correct normal vectors, depth and calibration data to avoid the influence of errors in the data on the training. To the best of our knowledge, no such dataset exists. In addition, a general dataset for object orientation in the context of 3D reconstruction is not available to our knowledge. Therefore, several datasets have been created together with our investigations that were made publicly available. Among them are two synthetic datasets with rendered images, normals, depth maps and ground truth of camera calibration, optical flow and camera positions. The first one contains data with fixed illumination of the scene (*ConsistentLight*) for both camera views. The other one contains scenes with inconsistent illumination (*InConsistentLight*), where the position of the light source changes significantly between the views. This simulates the difficult case, where, for example, the object rotates, which may dramatically change the angle of incidence of the light (violated brightness constancy assumption).

(a) Models of synthetic training data



(b) Models of synthetic test data

(c) Models of real test data

Figure 6.9.: *3D models that have been used to create the synthetic and real datasets. (a) shows the models on which the synthetic training scenes are based on, (b) shows the models of the synthetic test scenes (c) shows the models, that result from the captured real data.*

The scenes of the synthetic datasets were created and rendered using *Unity* [2]. To avoid dependencies on the background, 75 spherical backgrounds were added to the scenes randomly. The grayscale images, depth maps, normal maps and optical flows have been rendered for random scenes (random objects, random object positions and orientations, random light positions) each from two random camera perspectives. The calibration information, the camera positions and the position of the illuminating point light are also provided in the dataset. For both synthetic datasets, a training subset and a test subset were created. The training sets contain 20,000 random scenes in which objects were randomly placed. The test sets contain 1,000 random scenes in which other objects that have not been used in the training sets were chosen. The 22 models used for the training sets are shown in Figure 6.9 (a) and the eight models used for the test sets are shown in Figure 6.9 (b). Figure 6.10 (a)-(d) shows the rendered data for an exemplary scene.

In a similar format, a real dataset (*BuddhaBirdRealData*) is provided, which consists of captured data from 5 different objects, shown in Figure 6.9 (c). The images were captured by monochrome cameras. The depth data has been reconstructed by the structured light approach based on the presented procedure of the previous chapters. The normal maps were estimated using a geometry-based method that will be presented in the upcoming Chapter 8. After manually aligning the partial scans, the semi-dense flow fields between the views have been directly computed from projecting the aligned point clouds to the calibrated camera views. During the reconstruction process, the scenes have been illuminated by a projector, that has been calibrated jointly with the cameras which thus also provides the light position in the scenes. Each of the five models has been captured from 8 positions with two different cameras each. Flow and pose data is available for each of the camera combinations of adjacent positions, which yields ground truth data for 40 combinations per

object. This results in 200 ground truth scenes of the real data. Thereby, the first 40 pairs represent the scans within one scan head (consistent light) with 8 reconstructions per object. The last 160 pairs represent the inconsistent light case with combinations of camera views between adjacent scans (that use different projectors). Similar to the synthetic case Figure 6.10 (e), (f), (g), (h) shows the captured and estimated data for an exemplary real scene.

6.5.1. Data Sources and Data Formats

The 3D Models that have been used to create the datasets are taken from different sources and free to use. Models [m9, m12, m27] were taken from the Stanford 3D Scanning Repository [149], while models [m2, m7, m8, m11, m20] were taken from [192]. Models [m1, m3, m5, m6, m10, m13, m14, m17, m18, m21, m23, m24, m25, m26, m29, m30] can be found on the Smithsonian 3D Digitization page [148] that collects a large amount of 3D data from several museums and archives, from which a lot is free to use. Finally, models [m4, m15, m16, m19, m22, m28, m31, m32, m33, m34, m35] resulted from own research and are released with this work.
Each scene of the datasets, no matter if real or synthetic, consists of the following data parts:

- `image0` and `image1` contain the 8-bit integer grayscale images of the two camera views.
- `data0` and `data1` are .json files that contain the intrinsic calibration matrices $\mathbf{K}$, camera rotation $\mathbf{R}$ and translation $\mathbf{t}$, the minimal and maximal depth values $minDepth$ and $maxDepth$, the minimal and maximal values of the horizontal and vertical optical flows $minFlowX$, $maxFlowX$, $minFlowY$ and $maxFlowY$ and the coordinates if the light source $lightPos$.
- `depth0` and `depth1` are 16-bit integer grayscale images that need to be scaled after loading using minimal and maximal depth values from the data files:

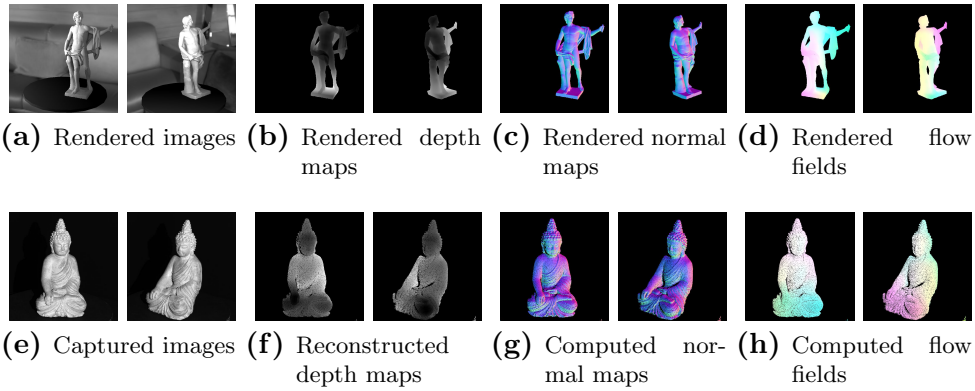
$$D = D \cdot \frac{maxDepth - minDepth}{65535} + minDepth$$

- `normal0` and `normal1` are 24-bit integer RGB images in tangent space that can be re-transformed to spatial space by:

$$\mathbf{n} = \left(\frac{2\mathbf{n}_1}{255} - 1,\ \frac{2\mathbf{n}_2}{255} - 1,\ 1 - \frac{2\mathbf{n}_3}{255}\right)$$

- `flow0` and `flow1` contain the horizontal and vertical displacements of the respective flow fields between the views. The flows are stored as 16-bit integers in three channel images (*flowX, flowY, zeros*) and need to be scaled similar to the depth files.

Note that missing/masked pixels for which no depth information is available contain zeros in the depth, flow and normal files. After rescaling and shifting



(a) Rendered images **(b)** Rendered depth maps **(c)** Rendered normal maps **(d)** Rendered flow fields

(e) Captured images **(f)** Reconstructed depth maps **(g)** Computed normal maps **(h)** Computed flow fields

Figure 6.10.: *Example scene of the synthetic (top row) and real (bottom row) datasets. Each scene contains images, depth maps, normal maps and flow fields of two different camera views. In addition a data file for each camera is stored, that contains calibration information, camera position, light source position and minimal/maximal values of flows and depths in order to allow memory efficient saving of the data.*

these files, the mask should be applied again to keep the masking information with values of zero.

The presented network uses vertex maps instead of depth maps. These can be computed from the depth data and the given calibration information by applying the following operation to each image pixel (x, y):

$$V(x,y) = \frac{\mathbf{K}^{-1}(x\ y\ 1)^\mathsf{T}}{\|\mathbf{K}^{-1}(x\ y\ 1)^\mathsf{T}\|_2} \cdot D(x,y) \tag{6.6}$$

6.5.2. Camera Pose and Scene Pose

The given depth, vertex and normal maps are independent of any camera pose (assuming the camera being placed in the world origin), as these are usually not available beforehand and need to be computed by the procedure. In order to use them with respect to the given camera pose in the world coordinate system, the vertex maps (or point clouds) and normal maps can be transformed in the following way. Given a camera pose $P = (\mathbf{R}, \mathbf{t})$, the 3D point with respect to a complete camera matrix $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$ is given by:

$$V(x,y) = -\mathbf{R}^\mathsf{T}\mathbf{t} + \mathbf{R}^\mathsf{T}V(x,y) \tag{6.7}$$

and the normals of the respective 3D points are given by:

$$N(x,y) = \mathbf{R}^\mathsf{T}N(x,y) \tag{6.8}$$

For completeness, we remind, that the camera itself is located in $\mathbf{R}^\mathsf{T}\mathbf{t}$.

In the usual case of unknown camera poses, only the relative transformation between two vertex maps / point clouds can be estimated from the given

data. In order to train a network, as introduced in the previous sections, it is necessary to convert the provided data to relative transformations between two views. If we are given the camera extrinsics of two views $\mathbf{R}_0$, $\mathbf{t}_0$ and $\mathbf{R}_1$, $\mathbf{t}_1$, the relative pose between vertex map V_0 and vertex map V_1 is given by

$$\mathbf{R}_{01} = \mathbf{R}_1\mathbf{R}_0^\mathsf{T}, \quad \mathbf{t}_{01} = \mathbf{t}_1 - \mathbf{R}_1\mathbf{R}_0^\mathsf{T}\mathbf{t}_0 \tag{6.9}$$

where vertex map V_0 is mapped to vertex map V_1 by applying the transformation as:

$$V_1 = \mathbf{R}_{01}V_0 + \mathbf{t}_{01} \ . \tag{6.10}$$

Example code on how to read, transform and also visualize the given data can be found together with the datasets.

6.5.3. Pre- and Post-Processing of Data

Point clouds that need to be aligned may theoretically be of arbitrary scale. Neural network based approaches, like the presented one, need to extract meaningful features within the given vertex maps to find corresponding points from which the desired transformation can be predicted. For this purpose, the weights of the network that performs this task are determined optimally using a data-based training. Thereby, the learned weights should have the same effective influence on all point clouds. Unfortunately, it is not possible to extract meaningful features from differently scaled vertex maps with always the same weights. Especially, learned thresholds for activations within the network may not be applicable.

A practical way around is to scale and move the point clouds, or equivalently the 3D data in the vertex maps, approximately towards the unit cube, which is located at the world origin. Within this working volume, the neural network can work effectively and perform the alignment. The calculated pose is then combined with the previous transformation towards the unit cube and thus provides the desired alignment operation on the raw data.

In a first step, the point clouds are moved to the origin by subtracting the centroids. In a second step the point clouds are scaled to fit approximately into the unit cube. Note that the method presented assumes each pair of point clouds to be of similar scale, as the underlying depth data usually comes up from the same sensor. Therefore, the scaling factor s towards the unit cube should be chosen similar for each pair of point clouds that are processed.

Let be given the two point clouds $X_0 = \{x_1^{(0)}, ..., x_M^{(0)}\}$ and $X_1 = \{x_1^{(1)}, ..., x_N^{(1)}\}$ that need to be aligned. The centered point clouds at the origin are given by:

$$\begin{aligned} X_0 - \mu_0 &= \{x_n^{(0)} - \mu_0 \mid x_n^{(0)} \in X_0\}, \ \mu_0 = \sum_{m=1}^{M} x_m^{(1)} \\ X_1 - \mu_1 &= \{x_m^{(1)} - \mu_1 \mid x_m^{(1)} \in X_1\}, \ \mu_1 = \sum_{n=1}^{N} x_n^{(1)} \end{aligned} \tag{6.11}$$

In a second step $X_0 - \mu_0$ and $X_1 - \mu_1$ are scaled jointly and robustly in order to ensure that 90% of the point clouds map into the according subspace of the unit cube ($[-0.45, 0.45]^3 \subset \mathbb{R}^3$), that is located at the origin. This robustifies the scaling and reduces the negative effect of outliers dramatically. Note that in general it can be assumed that at least 90% of a point cloud should contain usable data. Let be given the set of values with maximal absolute coordinates of both centered point sets, $Y = \{\max(|x|) \mid x \in (X_0 - \mu_0) \cup (X_1 - \mu_1)\}$. Having sorted the values $y_n \in Y$ in ascending order $y_1 \leq ... \leq y_{M+N}$, the scaling factor, that ensures 90% of both point clouds being mapped into the cube, defined above is given by $s = 1/y_{\lfloor 0.45(M+N) \rfloor}$, where $\lfloor \cdot \rfloor$ denotes floor rounding to integer values. The scaled, centered point clouds, that are ready to be fed to the network, are finally given by:

$$\tilde{X}_0 = s(X_0 - \mu_0), \qquad \tilde{X}_1 = s(X_1 - \mu_1) \tag{6.12}$$

Having computed a pose $\tilde{P} = (\tilde{\mathbf{R}}, \tilde{\mathbf{t}})$ using the neural network, that aligns the scaled point clouds by

$$\tilde{\mathbf{R}}\tilde{X}_0 + \tilde{t} \approx \tilde{X}_1, \tag{6.13}$$

the final transformation $P = (\mathbf{R}, \mathbf{t})$, that aligns the raw point clouds X_0 and X_1 is given by

$$\mathbf{R} = \tilde{\mathbf{R}}, \quad \mathbf{t} = \frac{1}{s}\tilde{\mathbf{t}} + \mu_1 - \tilde{\mathbf{R}}\mu_0 \tag{6.14}$$

6.6. Coherent Learning of INV-Flow2PoseNet

The goal of training the network is to estimate the best possible optical flow that will enable stable extraction of the pose. Therefore, to get an end-to-end trainable network, we define a joint loss function that penalizes both the ground truth flow and the extracted pose under given flow.

The *PWC-Net* structure predicts flows $F^{(l)}$ of different levels $l = 0, ..., L$. *Flow2PoseNet* moreover uses the flow to predict the relative rotation $\mathbf{R}$ and translation $\mathbf{t}$. The upcoming sections will introduce the different error types that are finally combined to provide the overall training loss for the network. Therefore, let according ground truth be given by $F^{(l)}_{\mathrm{GT}}$, $\mathbf{R}_{\mathrm{GT}}$ and $\mathbf{t}_{\mathrm{GT}}$.

6.6.1. Multiscale Endpoint Error

The multiscale endpoint error (EPE) penalizes the different levels of the flow calculation with different hardness, provided by the respective weighting parameters α_l:

$$\mathcal{L}_{\mathrm{EPE}}(F^{(0)}, ..., F^{(L)}) = \sum_{l=0}^{L} \alpha_l \|F^{(l)} - F^{(l)}_{\mathrm{GT}}\|_{\mathcal{F}} \tag{6.15}$$

with suitable level weights α_l, $l = 0, ..., L$ and Frobenius matrix norm $\|\cdot\|_{\mathcal{F}}$. In case of sparse data the differences inside the norm are masked in order to take the sparsity into account.

Note that the higher levels, which describe the rather coarse flow, are more important than the lower levels, which get the higher levels as input. However, since the higher levels have a lower resolution, the flow errors in absolute numbers are smaller than those of the lower levels. As a rule of thumb, due to the pooling between each level, the weighting should be at least halved each time to account for the resolution discrepancy. The weights that have been used for the proposed network are $\{\alpha_0, ..., \alpha_6\} = \{0.001, 0.0025, 0.005, 0.01, 0.02, 0.08, 0.32\}$.

6.6.2. Alignment Error

A measure that treats both rotation and translation jointly is the well-known alignment error. It models the mean Euclidean distance of all point correspondences given by the groundtruth flow:

$$\mathcal{L}_{\mathrm{AE}}(\mathbf{R}, \mathbf{t}) = \sum \|\mathbf{R}V_0(x, y) + \mathbf{t} - V_1(x + F_x^{01}, y + F_y^{01})\|_{\mathcal{F}} \tag{6.16}$$

This measure best describes the problem to be solved. It has the advantage that it weights the impact of rotation against the translation. It is again important to mask errors that contain invalid pixels either of V_0 or of warped V_1, in order to ensure that only locations are taken into account, where matching vertices in both views are available.

Note that this error alone might erroneously interchange rotations and translation effects in order to receive a minimal alignment error. These interchanges can be prevented by adding some direct translational and rotational error terms to the overall loss function. These additional terms act as a regularization to enforce a better decomposition into translation and rotation.

6.6.3. Translational and Rotational Errors

The error of the predicted translation is given by the Eucliden distance towards the ground truth translation:

$$\mathcal{L}_{\mathrm{TRANS}}(\mathbf{t}) = \|\mathbf{t} - \mathbf{t}_{\mathrm{GT}}\|_2 \tag{6.17}$$

Special attention is required for the rotation error. A suitable differentiable error between two rotation matrices $\mathbf{R}$ and $\mathbf{R}_{\mathrm{GT}}$ is given by the angular error, which is defined by the absolute value of the rotation angle θ of the relative rotation $\mathbf{R}_{\mathrm{rel}} = \mathbf{R}\mathbf{R}_{\mathrm{GT}}^{\mathsf{T}}$. Having a look at the conversion towards the axis angle representation there are basically two ways to compute the rotation angle. The first relation is given with the trace of the rotation matrix:

$$\mathrm{Tr}(\mathbf{R}_{\mathrm{rel}}) = 1 + 2\cos(\theta) \tag{6.18}$$

Another way is to calculate the rotation angle from the length of the extracted rotation axis. Having an explicit rotation matrix, the rotation axis $\mathbf{u}$ is given by:

$$\mathbf{R}_{\mathrm{rel}} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \quad \Rightarrow \quad \mathbf{u} = \begin{pmatrix} h - f \\ c - g \\ d - b \end{pmatrix} \tag{6.19}$$

The rotation angle θ is related to the length of $\mathbf{u}$ by:

$$\|\mathbf{u}\|_2 = 2\sin(\theta) \tag{6.20}$$

A direct computation of θ from Equation (6.18) or (6.20) requires the use of an inverse trigonometric function, either *arcus sinus* or *arcus cosinus*. Unfortunately, these yield numeric problems due to singularities in case of angles close to $\pm\frac{\pi}{2}$ or $\pm\pi$, which is unsuitable for a general loss function that is required to be differentiable. To avoid this, we used a more stable way to compute θ based on the two-dimensional *arcus tangens* atan2 with both arguments:

$$\mathcal{L}_{\mathrm{ROT}}(\mathbf{R}) = \left|\mathrm{atan2}\big(\|\mathbf{u}\|_2, 1 - \mathrm{Tr}(\mathbf{R}_{\mathrm{rel}})\big)\right| \tag{6.21}$$

6.6.4. Joint Training Loss

The joint loss function, is subsequently given by:

$$\begin{aligned}\mathcal{L} = \; & \mathcal{L}_{\mathrm{EPE}}(F^{(0)}, ..., F^{(L)}) + \mathcal{L}_{\mathrm{AE}}(\mathbf{R}_{1Step}, \mathbf{t}_{1Step}) \\ & + \mathcal{L}_{\mathrm{AE}}(\mathbf{R}_{2Step}, \mathbf{t}_{2Step}) + \mathcal{L}_{\mathrm{AE}}(\mathbf{R}_{3Step}, \mathbf{t}_{3Step}) \\ & + \mathcal{L}_{\mathrm{TRANS}}(\mathbf{t}_{3Step}) + \mathcal{L}_{\mathrm{ROT}}(\mathbf{R}_{3Step})\end{aligned} \tag{6.22}$$

In order to speed up the training procedure in the beginning, the gradients of the computed optical flow have been detached before backpropagating the alignment errors. Once the sub-network for optical flow prediction has satisfactorily formed its weights we attached the gradients and trained the full network in an end-to-end manner.

6.6.5. Representation of Rotation

In order to ensure the predicted rotation matrix to be a proper rotation, a minimal parameterization by *Euler Angles* is chosen. Therefore, three values (θ, ρ, ϕ) are predicted by the network, defining the rotation angles around the x, y and z axes by the rotation matrices:

$$\mathbf{R}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix}, \; \mathbf{R}_y = \begin{pmatrix} \cos(\rho) & 0 & \sin(\rho) \\ 0 & 1 & 0 \\ -\sin(\rho) & 0 & \cos(\rho) \end{pmatrix}, \; \mathbf{R}_z = \begin{pmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6.23}$$

The total rotation is given by $\mathbf{R} = \mathbf{R}_x\mathbf{R}_y\mathbf{R}_z$ as the consecutive execution of these rotations. Vice versa, the respective *Euler Angles* can be extracted from a given rotation matrix $\mathbf{R}$ by:

$$\theta = \mathrm{atan2}(-\mathbf{R}_{23}, \mathbf{R}_{33}) \tag{6.24}$$

$$\rho = \mathrm{atan2}(\mathbf{R}_{13}, \sqrt{\mathbf{R}_{23}^2 + \mathbf{R}_{33}^2}) \tag{6.25}$$

$$\phi = \mathrm{atan2}(-\mathbf{R}_{12}, \mathbf{R}_{11}) \tag{6.26}$$

This conversion is especially used to compute the *Euler Angles* of the refined rotation matrix $\hat{\mathbf{R}}$ in the *3 Step Method* of Section 6.4.

6.7. Evaluation

For evaluation, we compare the calculated optical flow and the resulting registration qualitatively on different synthetic and real datasets. Highly accurate results visualize a good generalization without finetuning from synthetic training data to the difficult real test scenes. Figure 6.11 and Figure 6.12 show the results for exemplary objects from the training (top 3 rows) and test datasets (bottom 3 rows) for the consistent and inconsistent light (moving light source) case. Thereby the first columns show the input data consisting of images, normal and depth maps (that are converted to vertex maps using the calibration information, as in Equation 6.6). The second column shows the resulting optical flow in comparison to the semi-dense ground truth optical flow in column 3. Columns 4 and 5 finally show the initial and the registered point clouds using the proposed neural networks. Special attention should be given to row 6 of Figure 6.12, which shows the performance of the neural network on a real test scene without finetuning.

Further positions of the real scene are shown in Figure 6.14. It visualizes the performance of the method applied to 8 partial scans of the Buddha scene (from the *BuddhaBirdReal* dataset), as it usually comes up from 3D scanners. Using the alignment given by the neural network, a few iterations of *Iterative Closest Points (ICP)* for refinement yield good results on the overall aligned point cloud of the object.

In addition, we also consider a network trained on the popular training sequences of Kitti Odometry and evaluate it on the test data as shown in Figure 6.13. As the Kitti dataset has less strong rotations and less shading changes, it is not the typical use case for the proposed method. Nevertheless, the proposed method works reliable for this easier kind of situations as well.

6.7.1. Quantitative Evaluation

For quantitative evaluation we compare the different architectures (*1 Step* and *3 Step*) on the datasets published together with this work. Table 6.1 shows the results on the full subsets with *consistent light* and *inconsistent light*. In both cases the *3 Step* method yields superior results in comparison to the standard procedure that directly predicts rotation and translation jointly. Especially the resulting rotation is much more accurate, resulting in an alignment error that is up to 3 times smaller than in the standard prediction method.

Light		Consistent		Inconsistent	
Data Type	Method	EPE	AE	EPE	AE
Train Data	1 Step	1.83	0.035	2.33	0.035
Train Data	3 Step	1.83	**0.012**	2.33	**0.013**
Test Data	1 Step	4.09	0.037	8.08	0.048
Test Data	3 Step	4.09	**0.023**	8.08	**0.035**

Table 6.1.: *Quantitative comparison of the 1 Step and the proposed 3 Step methods to predict the pose from given warped vertex and normal maps.*

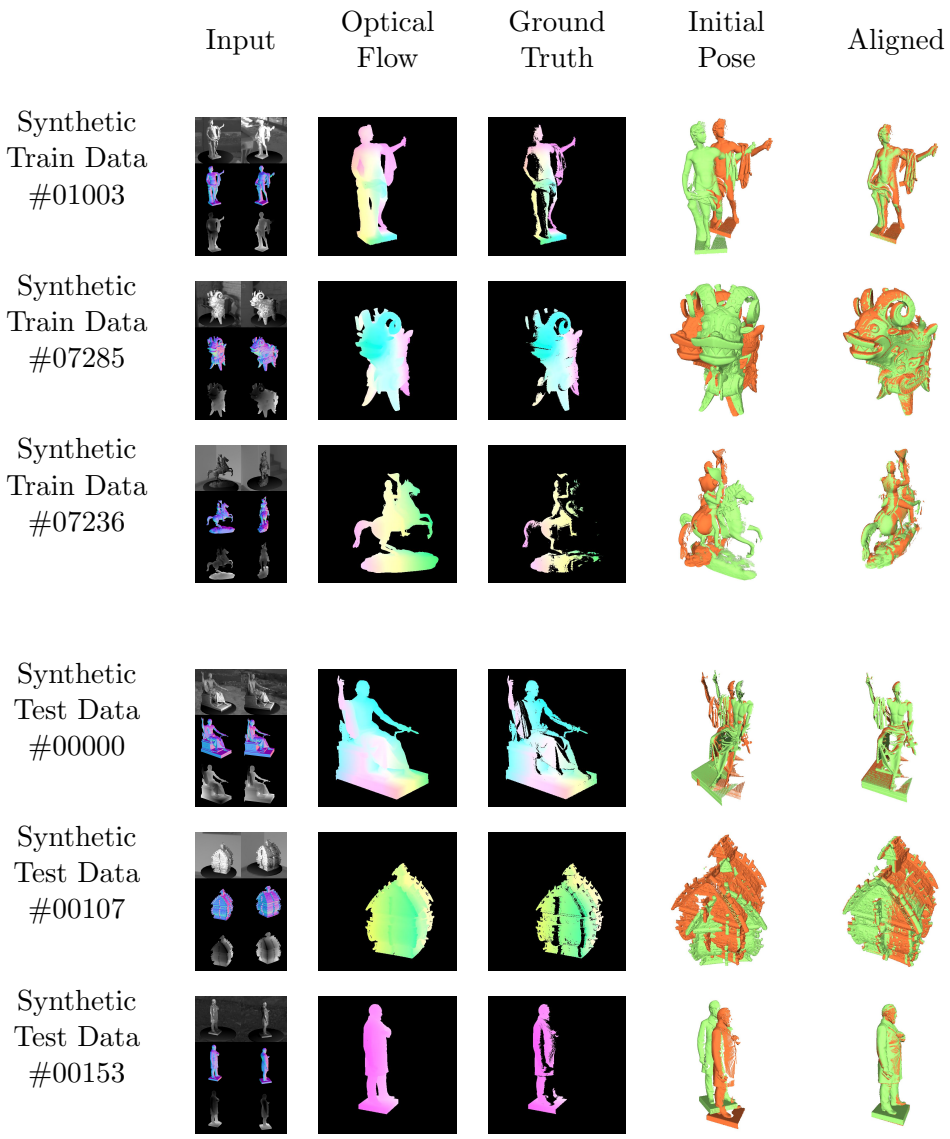


Figure 6.11.: *Qualitative results of the proposed method on training (top 3 rows) and test (bottom 3 rows) data of the synthetic consistent light dataset. The situation of consistent light represents the standard case, where for example the camera moves through a static scene with static light sources. The brightness constancy assumption is usually not violated. The network generalizes well from known training to unknown test data.*

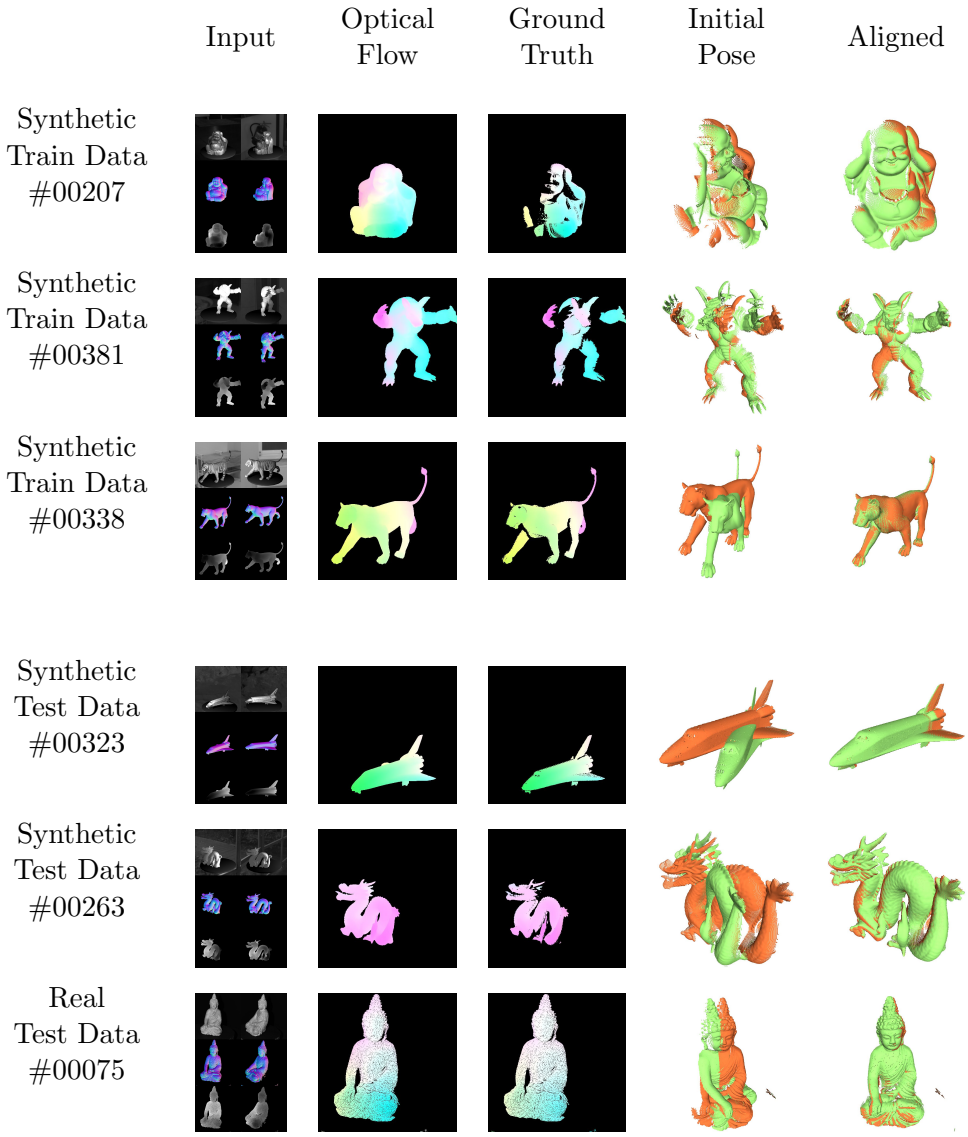


Figure 6.12.: *Qualitative results of the proposed method on training (top 3 rows) and test (bottom 3 rows) data of the synthetic inconsistent light dataset as well as real test data. The situation of inconsistent light represents the situation under investigation, motivating this paper, where the light sources or the objects in the scene move or rotate, yielding strong shading changes. The brightness constancy assumption is dramatically violated. The network still generalizes well from known training to unknown test data. Even for real data without additional finetuning the results are satisfying.*

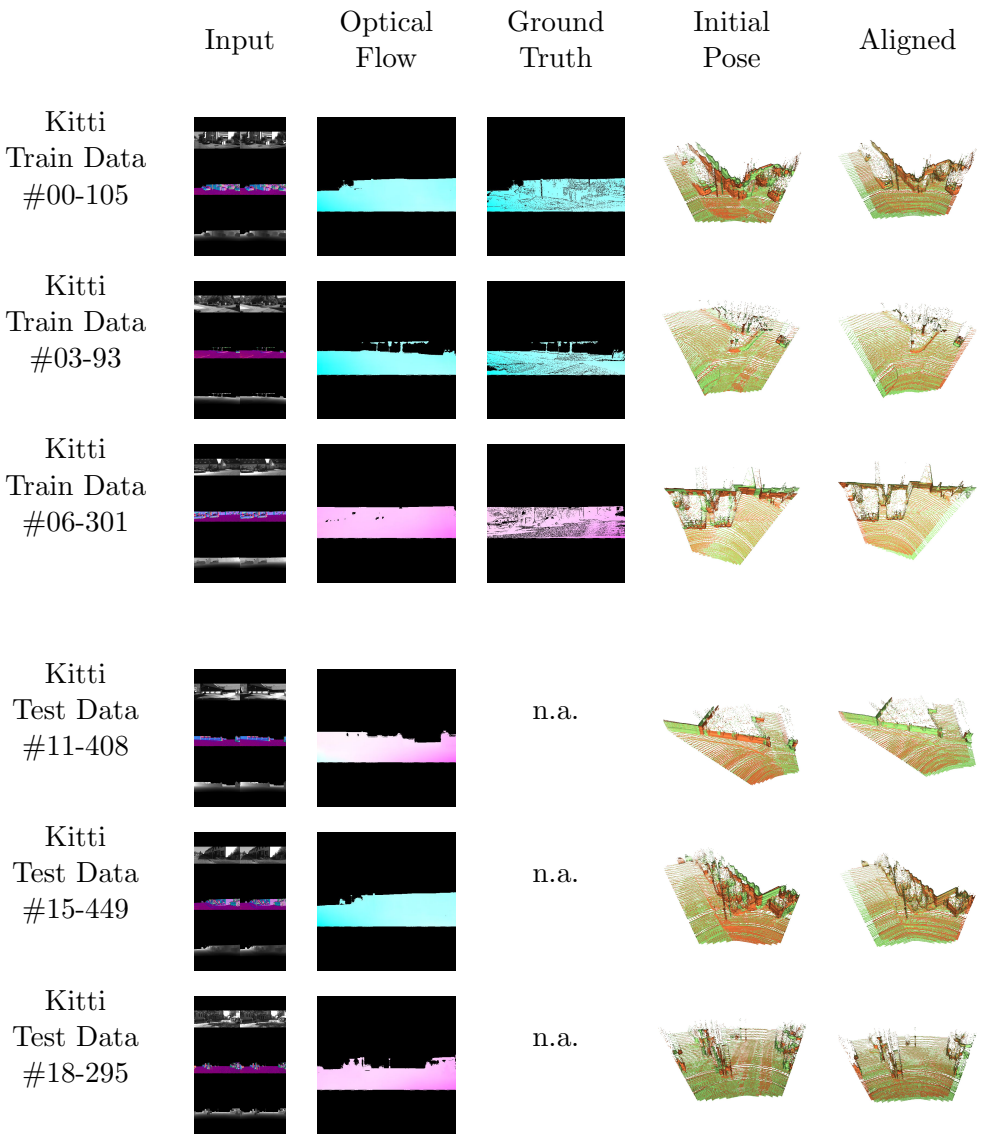


Figure 6.13.: *Qualitative results of the proposed method on training and test data of the Kitti Odometry dataset. The method also works on this kind of scenario with less rotations and less shading changes than in the mainly investigated case, but also handles noise resulting from the lidar depth mesurement in the Kitti data. The network generalizes well from known training to unknown test data.*

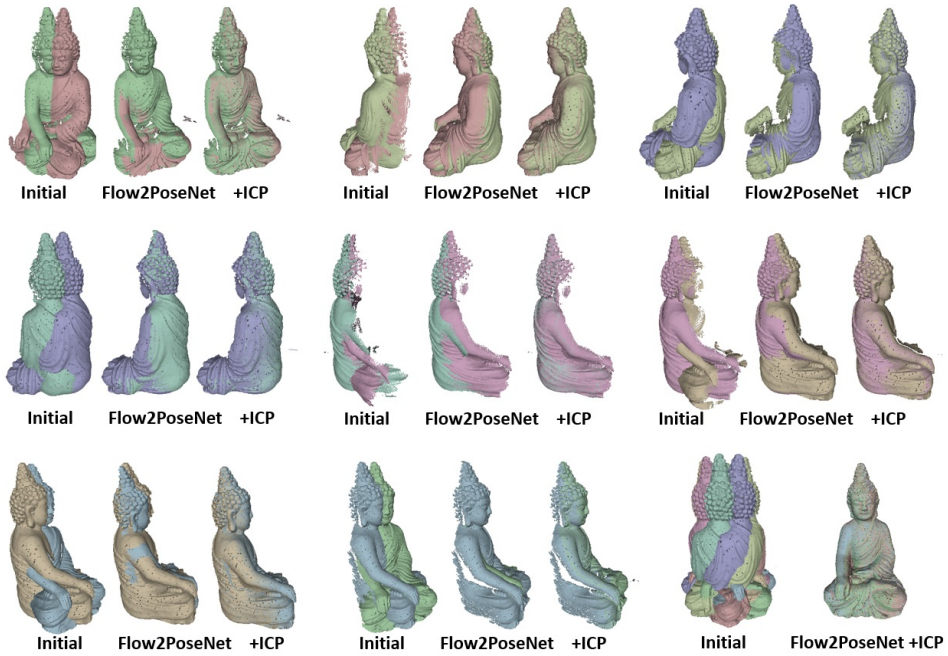


Figure 6.14.: *Application of the method to a full sequence of partial reconstructions of a the real Buddha object from the BuddhaBirdReal dataset. Such sequences usually result from 3D scanners (as here from a structured light scanner). Since usually a turntable is used, strong rotations (≈ 45°) and shading changes disturb the data. After the prealignment a few iterations of the ICP algorithm are applied to refine the alignment of the point clouds. The image on the bottom right shows the result on the overall aligned full point cloud of the statue.*

6.7.2. Predicted Dense Optical Flow

A special feature of the proposed method is its coarse to fine pyramidal optical flow base, combined with the rigid pose extraction. Therefore one can assume that the optical flow predicting sub-network learns rigidity relations from the extractability of the rigid pose from the dense optical flow. As shown in Figure 6.15, the ground truth optical flow (column 2) that has been used for training and evaluating the networks, is sparse, as it only contains the flow of points that are visible in both views. As the data is created synthetically, it is possible to also render dense ground truth optical flows (column 4) that contain the flow of points which are occluded in one of the views and therefore may not be computable at all by the network. As can be seen, the predicted optical flow (column 3) is dense. It also predicts flow values for points that are not visible in both views. These values result from context of other points, where the flow can be estimated stably. The network learns how the flow behaves for rigid objects and transfers the knowledge to interpolated pixels. This works for objects that are known from the training set (rows 1 and 3) as well as for test objects, that have never been used for training (rows 2 and 4). This applies to the *ConsistentLight* case (rows 1 and 2) and to the *InconsistentLight* case

(rows 3 and 4), as well. Table 6.2 moreover shows that the resulting *Endpoint Errors (EPE)* do not dramatically increase for the invisible points, which indicates, that the network learns to predict flows for the invisible points from context, according to the behavior of rigid objects.

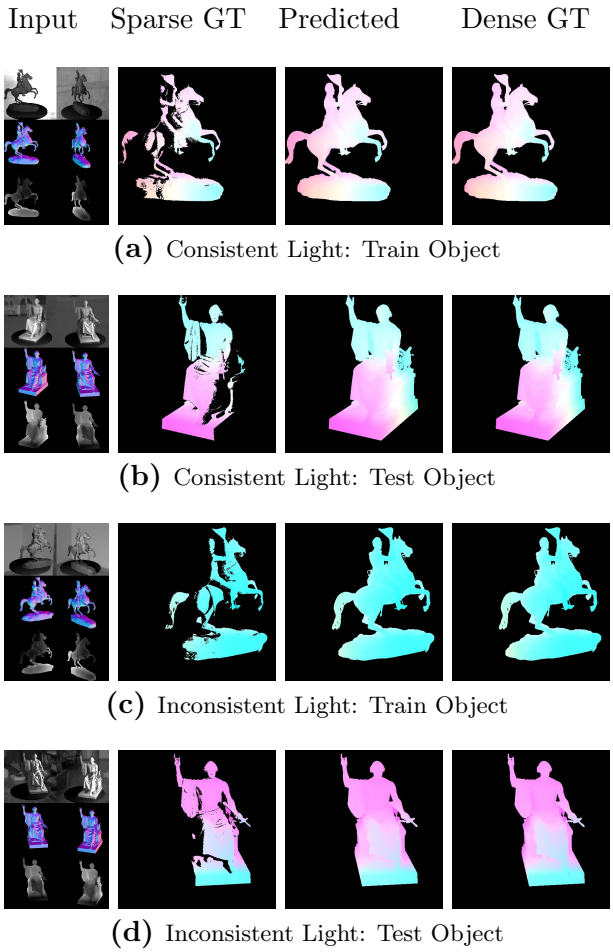
Consistent Light		
Data Type	Visible Points EPE	Invisible Points EPE
Train Data	2.7446	3.4978
Test Data	3.6411	4.9284

Inconsistent Light		
Data Type	Visible Points EPE	Invisible Points EPE
Train Data	3.6974	5.4024
Test Data	4.7996	4.7703

Table 6.2.: *Quantitative results for the visible and invisible points in the evaluated scenes. The resulting Endpoint Errors (EPE) do not heavily increase. The network is still able to predict accurate flows from context of visible points and to generalize to the test data for the consistent as well as for the inconsistent data.*

6.8. Conclusion

In this chapter, a method has been presented that combines optical flow estimation of rigid scenes with a posterior pose estimation. In this way, a method has been developed that allows scenes with difficult lighting conditions to be registered in a stable way.
Optical flow is thereby estimated accurately using geometric, shading and texture features. The variety of different feature types allows the system to be trained to be illumination resistant (using geometric and normal features) without having to completely sacrifice potentially important texture features. The pose is then stably estimated from the warped normals and vertex maps using a new 3-step procedure. This has, compared to typical approaches that directly infer the pose, significant advantages especially in cases with strong rotations that often cause the considered shading changes.
The combination of optical flow and rigid pose estimation allows the pose to benefit from the features of different levels of the underlying coarse-to-fine flow approach, which means that the method is not dependent on highly accurate features and can also align smooth scenes with weak features. In turn, the optical flow sub-network learns a typical flow behavior of rigid scenes from the posterior estimability of the pose. This allows accurate dense estimates to be achieved, even for occluded areas based on context and overall learned behavior.



(a) Consistent Light: Train Object

(b) Consistent Light: Test Object

(c) Inconsistent Light: Train Object

(d) Inconsistent Light: Test Object

Figure 6.15.: *Qualitative results of the predicted (dense) optical flow. The network allows to compute accurate flows for invisible pixels from context of visible parts for the consistent as well as for the inconsistent data.*

Chapter 7

Automatic Alignment of Full Turn Object Scans

Contents

7.1. Introduction

Point cloud registration is an important task in computer vision, computer graphics, robotics, odometry and many other disciplines. The problem has been studied for a long time and many different approaches have been established. In the case of existing rough initializations, the *Iterative Closest Point* (ICP) method is widely used. Often only the pairwise problem is treated. In case of many applications, especially in 3D reconstruction, closed rotations of sequences of partial reconstructions have to be registered. In this chapter, it will be shown that there are considerable advantages if ICP iterations are per-

formed jointly instead of the usual pairwise approach (Pulli's approach [135]). Without the need for increased computational effort, lower alignment errors are achieved, drift is avoided and calibration errors are uniformly distributed over all scans. Based on the underlying global energy functional, the joint approach is further extended into a global version, which not only considers one-sided adjacent scans, but updates symmetrically in both directions. The result is an approach that leads to a much smoother and more stable convergence, which moreover enables a stable stopping criterion to be applied. This makes the procedure fully automatic and therefore superior to most other methods, that often tremble close to the optimum and have to be terminated manually. A complete procedure is presented, which in addition addresses the issue of automatic outlier detection in order to solve the investigated problem data independently without any user interaction.
The task of point cloud registration is to align two point sets so that they resemble each other as closely as possible in as many regions as possible. In order to make this problem well-defined, it is assumed that the point clouds represent the same scene or at least that sufficiently large parts of the point clouds represent overlapping parts of the scene. Otherwise, no matching areas can be identified and the problem cannot be solved.
A distinction is made between rigid and non-rigid registration. For the rigid case, two point clouds are aligned only by rotation and translation (in some cases also scaling). The appearance and proportions are fully preserved. In contrast, for non-rigid registration, deformable objects are aligned by non-linear transformations.
In classical computer vision and robotics, rigid alignment is by far the most common case and has been extensively studied. For this purpose, methods have been established, which simultaneously detect point correspondences and align them iteratively. In particular the ICP approach ([13], [195], [25], [24]) has to be named, which has been successfully applied for decades and that will also form the basis of the presented procedure.
A special case, which occurs in many practical applications, is given by a sequence of point clouds, which partially overlap pairwise and whose last point cloud closes up with the first one. In this case, it is no longer a matter of several pairwise registration problems but a global over-determined registration problem. This is because each point cloud has two neighbors (last and next one) with which it must be aligned. In the case of real data, such as the partial reconstructions of a 3D scanner, pairwise sequential alignment would usually lead to a drift, i.e. a large gap or too much overlap between the last and first position. This drift occurs when the partial alignment errors and possible calibration errors in the partial reconstructions add up to a large error. To avoid such drift, it is common to apply *Pulli's procedure* [135], which involves aligning and merging opposite pairs of adjacent point clouds. The resulting merged larger point clouds are then further treated together. In this way, the error is not concentrated between two scans and the drift is distributed over a larger number of scans.
In this chapter will be shown that there are nevertheless better ways with

much better properties to solve the global alignment problem in a stable way and to actually distribute the drift evenly without higher computational effort. It will be shown that a joint iteration of the pairwise registrations distributes the drift uniformly and achieves lower alignment errors. Furthermore, it will be presented how the standard procedure of pairwise minimization can be extended into a global procedure by symmetrically registering each scan with the next and previous scan in the sequence. This results in a global approach that leads to a much smoother convergence, which allows the reliable use of automatic stopping criteria. In contrast, standard procedures usually begin to tremble near the minimum, which often requires a manual termination of the iterations. Finally, a practical approach to the automatic detection of outliers is presented. This is to provide a complete and stable solution to the problem without any user interaction. To allow maximum reproducibility, the entire procedure is attached as pseudo-code at the end of the chapter.

Figure 7.1.: *Partial reconstructions of a full turn with 8 separate scans and the complete point cloud after alignment (middle).*

7.2. Related Work

The problem of point cloud registration has been well studied for several decades. Explicit methods for rigid alignment of given point correspondences from two datasets have already been developed in the last century by Arun *et al.* in [10] and Umeyama in [165]. They are based on the singular value decomposition and due to their simplicity they are still the basis of the modern state of the art. These methods can further be robustified by additional weights, based on the certainty of the correspondences as shown by Arun in [4]. In [33] these approaches are extensively evaluated and compared with other approaches.
For many applications, there are more than two views to be aligned. In order

to treat multiple point clouds jointly, an extension of the *Orthogonal Procrustes Problem* has been introduced by Berge in [159]. In [162] Trendafilov and Lippert use a relaxation of the orthogonal constraints. Jointly obtained solutions are projected to the space of the orthogonal matrices afterwards. Piztarro and Bartoli [132] transferred the problem into a simple semi-definite programming in order to ease solving. These methods are no longer explicit and require higher computational effort. In the context of point cloud alignment performed in the upcoming task, the given correspondences are erroneous approximations and change from iteration to iteration. Therefore, a higher accuracy at the costs of additional internal iterations is not reasonable.
Usually, no exact point correspondences are available. A famous principle proven in practice is *Iterative Closest Point* [13], [135], [195], which iteratively selects the closest points of the data sets as correspondences and calculates infinitesimal updates accordingly. There are also variants that take the normal vectors of the point clouds into account, as the one of Masuda *et al.* [113] or Gelfand *et al.* [52], and thus improve the alignment for badly sampled and very smooth objects.
In order to accelerate convergence of the methods a possibility is to adeptly sample the point clouds like proposed by Rusinkiewicz and Levoy in [140] or by Gelfand *et al.* in [52]. Outliers are often efficiently detected and rejected like introduced by Zhang [190], Dorai *et al.* [30] or Rusinkiewicz and Levoy in [140]. Another way to speed up convergence is to extrapolate iteration updates like in [176]. There are also methods that accelerate by a multi-resolution approach like proposed by Jost and Hügli [83] or recently by Anderson acceleration as shown in [129]. In order to register a closed sequence of scans, as it is the case in a large number of applications, and to distribute alignment and calibration errors to all views, Pulli's method [135], followed by joint iterations, was considered to be the state of the art for a long time. In this chapter we will show that there is a better way to improve global alignment without increasing complexity, resulting from a joint iteration in a global approach that updates symmetrically towards all neighbors.

7.3. Background: Rigid Point Cloud Alignment

The most common algorithm for rigidly aligning point clouds is *Iterative Closest Point*. Thereby, the closest points of two point sets are chosen as correspondences and optimally aligned with each other. Afterwards, new correspondences are chosen, based on the improved alignment. Iteratively, the alignment of the point clouds is improved. For given point correspondences there is a closed form of the optimal rotation matrix and the translation vector for the pairwise case (*Procrustes Analysis*). Since this is also the basis of the method presented in the following and in order to make the chapter independently, the procedure for the pairwise case will be briefly presented. Afterwards, it will be shown how the method can be applied to a full turn according to the current state of the art.

7.3.1. Orthogonal Procrustes Problem

Assume two sets of point clouds $P = \{\mathbf{x}_0, ..., \mathbf{x}_{N-1}\}$ and $P' = \{\mathbf{x}'_0, ..., \mathbf{x}'_{N-1}\}$ consisting of matching point pairs $\mathbf{x}_n \leftrightarrow \mathbf{x}'_n$, for $n = 0, ..., N-1$ are given. The task is to find an optimal rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ in order to align points $\mathbf{x}_n$ from P by $\mathbf{R}\mathbf{x}_n + \mathbf{t}$ to points $\mathbf{x}'_n$ from P'. Therefore, the sum of Euclidean distances between all point pairs is minimized:

$$\underset{\mathbf{R},\,\mathbf{t}}{\operatorname{argmin}} \sum_{n=0}^{N-1} \|\mathbf{x}'_n - \mathbf{R}\mathbf{x}_n - \mathbf{t}\|_2^2 \tag{7.1}$$

Setting the derivative of (7.1) with respect to translation vector $\mathbf{t}$ equal to zero leads to the minimizer $\mathbf{t}$ of the energy:

$$\mathbf{t} = \frac{1}{N}\sum_{n=0}^{N-1} \mathbf{x}'_n - \mathbf{R}\frac{1}{N}\sum_{n=0}^{N-1} \mathbf{x}_n = \mu_{P'} - \mathbf{R}\mu_P \tag{7.2}$$

Thereby μ_P and $\mu_{P'}$ denote the centroids of the point clouds computed by the mean of the point sets. Inserting Equation (7.2) into the problem formulation (7.1) decouples the problem. It is equivalent to aligning point clouds with zero centroids by optimal rotation only:

$$\underset{\mathbf{R}}{\operatorname{argmin}} \sum_{n=0}^{N-1} \|\mathbf{q}'_n - \mathbf{R}\mathbf{q}_n\|_2^2, \quad \begin{matrix} \text{with} & \mathbf{q}_n = \mathbf{x}_n - \mu_P \\ \text{and} & \mathbf{q}'_n = \mathbf{x}'_n - \mu_{P'} \end{matrix} \tag{7.3}$$

Calculating the norm explicitly and replacing the remaining scalar product by the trace formulation leads to the following formulation of the problem that can be solved in terms of the singular value decomposition of matrix $\mathbf{H}$. The validity of this optimizer can be shown by application of *Cauchy-Schwartz Inequality*.

$$\underset{\mathbf{R}}{\operatorname{argmax}} \operatorname{Tr}\Big(\mathbf{R}\sum_{n=0}^{N-1} \mathbf{q}_n \mathbf{q}'^{\mathsf{T}}_n\Big) = \underset{\mathbf{R}}{\operatorname{argmax}} \operatorname{Tr}(\mathbf{R}\mathbf{H}) \quad \rightarrow \mathbf{R} = \mathbf{V}\mathbf{U}^{\mathsf{T}}, \text{ with } \mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{\mathsf{T}} \tag{7.4}$$

Weighted Case When working with real data it is usual to apply certainty weights $w_n \geq 0$ with $\sum_{n=0}^{N-1} w_n = 1$ with respect to the point pairs to the alignment error (7.1) in order to robustify the approach.

$$\underset{\mathbf{R},\,\mathbf{t}}{\operatorname{argmin}} \sum_{n=0}^{N-1} w_n \|\mathbf{x}'_n - \mathbf{R}\mathbf{x}_n - \mathbf{t}\|_2^2 \tag{7.5}$$

The problem is solved similarly to the unweighted case using weighted versions of the centroids and matrix $\mathbf{H}$:

$$\mu_P = \sum_{n=0}^{N-1} w_n \mathbf{x}_n, \quad \mathbf{H} = \sum_{n=0}^{N-1} w_n \mathbf{q}_n \mathbf{q}'^{\mathsf{T}}_n \tag{7.6}$$

7.3.2. Iterative Closest Point (ICP)

Usually, no point correspondences are available between two point clouds. In the procedure of *ICP*, these are approximately chosen in each iteration as the nearest points between the data sets and infinitesimal updates are calculated by *Orthogonal Procrustes Analysis*. Since it is a least-squares formulation, it is important to assess the quality of the correspondence. For this purpose, outliers, i.e. points that obviously have no matches in the other data set, are rejected. All other points are weighted according to their quality, which is often done by the point-to-point distance. Sampling rates and methods can also have a strong influence on performance and should not be disregarded.

Initialization For this procedure to work, an initial alignment is urgently required. This prevents the procedure from getting stuck in a local minimum. Based on feature points in the object, from texture as well as from geometry an initial alignment of the point clouds can be achieved as shown in Chapter 6. Based on a good initial registration, the ICP algorithm has proven over a long period of time to be a good choice for refining the alignment.

7.3.3. Full Turn Registration: Pulli's Approach

In a variety of practical applications, full turns of overlapping partial reconstructions are captured as depicted in Figure 7.1. Usually, the last scan overlaps with the first one and therefore completes the reconstruction process. In sequential pairwise registrations of the scans, a drift error between the last and the first position often occurs. To avoid or at least reduce this drift error, Pulli's approach [135] has always been the undisputed state of the art. One after the other, scans are registered and merged with their neighbors. These merged point clouds are then registered again until the whole object is composed. In fact, the error is distributed more evenly than in the naive approach and is not added up to a single gap, but it is far from uniform. While the first registration procedures only contain the local alignment errors, the last step combines the alignment errors of several sub-alignments and possible calibration errors. Therefore, in a further step, often pairwise iterations of all scans are performed jointly, which can lead to strong alternations in the global convergence. In the following, the problem will be formulated completely as a global registration problem, and the connection of the joint pairwise iterations to the minimization of the problem will be shown. In particular, the problem of alternating convergence in this case becomes clear. This is, since the pairwise approach considers in each iteration only an incomplete sub-problem with respect to the current minimization objective.

7.4. Joint Rigid Point Cloud Alignment

In the following, the alignment problem of a full rotation of scans is formulated as common optimization problem. It is assumed that two successive scans have

at least some overlap and that the last scan closes up to the first one, thus well-defining the problem.

Joint Minimization Problem Given a full turn of partial reconstructions consisting of S scans $\{\mathcal{S}_0, ..., \mathcal{S}_{S-1}\}$, where the last position $\mathcal{S}_{S-1}$ is assumed to be overlapping with the first one $\mathcal{S}_0$. Between two subsequent scans, say scan s and scan $s+1$, N point matches are assumed each, given by $\mathbf{x}_n^{(s,s+1)} \leftrightarrow \mathbf{x}_n^{(s+1,s)}$, for $n = 0, ..., N-1$. The objective error function that has to be minimized is then given by

$$\underset{\mathbf{R}^{(s)},\ \mathbf{t}^{(s)}}{\operatorname{argmin}} \sum_{s=0}^{S-1} \sum_{n=0}^{N-1} \|\mathbf{R}^{(s+1)}\mathbf{x}_n^{(s+1,s)} + \mathbf{t}^{(s+1)} - \mathbf{R}^{(s)}\mathbf{x}_n^{(s,s+1)} - \mathbf{t}^{(s)}\|_2^2. \tag{7.7}$$

Note that a periodic arrangement is assumed, so that the scans' indices are treated modulo S, which means $S \equiv 0$. Setting the partial derivative with respect to any translation vector $\mathbf{t}^{(s)}$ equal to zero yields:

$$2\mathbf{t}^{(s)} - \mathbf{t}^{(s-1)} - \mathbf{t}^{(s+1)} = \mathbf{R}^{(s-1)}\mu_{s-1,s} + \mathbf{R}^{(s+1)}\mu_{s+1,s} - \mathbf{R}^{(s)}(\mu_{s,s-1} + \mu_{s,s+1}) \tag{7.8}$$

which is sufficiently fulfilled for

$$\mathbf{t}^{(s+1)} - \mathbf{t}^{(s)} = \mathbf{R}^{(s)}\mu_{s,s+1} - \mathbf{R}^{(s+1)}\mu_{s+1,s}\ . \tag{7.9}$$

Therefore, the objective function (7.7) can be decoupled into:

$$\underset{\mathbf{R}^{(s)}}{\operatorname{argmin}} \sum_{s=0}^{S-1} \sum_{n=0}^{N-1} \|\mathbf{R}^{(s+1)}(\mathbf{x}_n^{(s+1,s)} - \mu_{s+1,s}) - \mathbf{R}^{(s)}(\mathbf{x}_n^{(s,s+1)} - \mu_{s,s+1})\|_2^2 \tag{7.10}$$

$$= \underset{\mathbf{R}^{(s)}}{\operatorname{argmin}} \sum_{s=0}^{S-1} \sum_{n=0}^{N-1} \|\mathbf{R}^{(s+1)}\mathbf{q}_n^{(s+1,s)} - \mathbf{R}^{(s)}\mathbf{q}_n^{(s,s+1)}\|_2^2 \tag{7.11}$$

Joint Sequential ICP Solving the terms of the joint minimization problem (7.11) sequentially for one s after the other by simply applying the standard strategy (7.4) leads to a pairwise approach with joint iterations. Iteratively, closest points between each neighboring pair are chosen and alignment updates by *Orthogonal Procrustes Problem* (7.4) are applied to each pair. If it does not get stuck in a local minimum, this procedure already avoids drift and the errors are uniformly distributed without additional computational effort.

Joint Global ICP In order to derive a global formulation that does not only take pairwise point clouds into account, but also treats the global arrangement information, functional (7.11) is minimized with respect to each $\mathbf{R}^{(s)}$ while fixing the others. This is equivalent to solving the following optimization problem:

$$\underset{\mathbf{R}^{(s)}}{\operatorname{argmax}} \sum_{s=0}^{S-1} \operatorname{Tr}(\mathbf{R}^{(s)}\mathbf{H}_{s,s+1}\mathbf{R}^{(s+1)^\mathsf{T}}) \tag{7.12}$$

$$= \underset{\mathbf{R}^{(s)}}{\operatorname{argmax}} \operatorname{Tr}\Big(\mathbf{R}^{(s)}\underbrace{(\mathbf{H}_{s,s+1}\mathbf{R}^{(s+1)^\mathsf{T}} + \mathbf{H}_{s,s-1}^\mathsf{T}\mathbf{R}^{(s-1)^\mathsf{T}})}_{\mathbf{H}_s}\Big)$$

This is a form of a symmetric alignment update of scan $\mathcal{S}_S$ towards previous and next adjacent scans $\mathcal{S}_{S-1}$ and $\mathcal{S}_{S+1}$. The problem can be solved similar to (7.4) using singular value decomposition and without special treatment.

Efficient Point Matching To efficiently find the nearest points between two point clouds, the use of space partitioning techniques such as *k-d-trees* ([56], [123], [20]) has been established for a long time. Building them means a not inconsiderable effort, but once they are created, the nearest points can be found in logarithmic time. A special feature is, that for each point cloud of a scan only one tree has to be set up, which can be further used after transformation by applying the inverse transformation to the input points, as shown in Algorithm 23. Especially in the iterative application to large point sets, this means an enormous time saving.

7.5. Outlier Rejection

In order to achieve an automatic procedure that can be applied to a possibly large number of configurations, outliers must be reliably detected in every set of correspondences. Standard procedures, such as rejecting the 10% of correspondence with largest point-to-point distances in each iteration, are widely used, but rely on a well-chosen value. In order to be independent of a fixed value, investigations on a large number of datasets have been carried out.
The task is to separate a set of N point correspondences into two subsets. The separation should divide the outliers as well as possible from the eligible correspondences.
Let $D = \{d_0, ..., d_{N-1}\}$ be the set of point-to-point distances of respective correspondences, sorted in a descending order ($d_0 \geq d_1 \geq ... \geq d_{N-1}$). Tests on approximately 25000 different point sets and configurations have shown that a good partition

$$D = D_{\text{outliers}} \cup D_{\text{inliers}} = \{d_0, ..., d_t\} \cup \{d_{t+1}, ..., d_{N-1}\} \tag{7.13}$$

is achieved at a split point $t \in \{0, ..., N-1\}$ if the *coefficients of variation* of both subsets is equal or as close as possible. Therefore, t can be successively increased until the following equation holds approximately true:

$$\frac{\frac{1}{t+1}\sum_{n=0}^{t} d_n^2}{\left(\frac{1}{t+1}\sum_{n=0}^{t} d_n\right)^2} = \frac{\frac{1}{N-t}\sum_{n=t+1}^{N-1-1} d_n^2}{\left(\frac{1}{N-t-1}\sum_{n=t+1}^{N-1} d_n\right)^2} \tag{7.14}$$

Figure 7.2 shows the behavior of the rejection strategy for the point sets that are evaluated in the upcoming section (see Figure 7.3, bottom right). Each reconstruction consists of 8 partial scans as it usually comes up from 3D scanners. Between each adjacent pair of scans, matches are computed and outliers are detected by the proposed strategy. Each of the subplots in Figure 7.2 shows the 8 curves which result from sorting 1000 matches between each of the 8 point pairs. The red segments visualize the detected set of outliers. For better visualization the plots are normalized and given in a logarithmic scale.

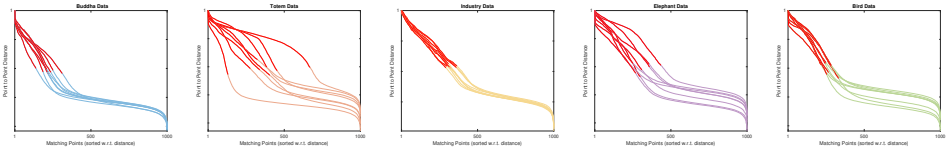


Figure 7.2.: *Outlier rejection strategy (7.14) applied to the point clouds shown in Figure 7.3 bottom right. The lines result from point-to-point distances of matches sorted in descending order. The red segments visualize the detected subset of outliers.*

7.6. Evaluation

For perfect artificial data or uniformly added noise, all alignment strategies work satisfactorily. The situation is different for the real use case of recorded data. In the following the considered ICP methods for registration of full turns are evaluated on a number of sample datasets as they appear from typical 3D scanners. For five independent objects (*Buddha*, *Totem*, *Industry*, *Elephant*, *Bird*), full rotations of eight partial reconstructions each were created. In order to fully align them, the registration methods must be able to deal with both, local alignment errors of the partial point clouds and calibration errors that can have an impact on the overall fit. The standard procedures were compared to the presented joint ICP variants. Figure 7.3 bottom right shows the resulting aligned point clouds of the *Joint Global ICP* approach to represent the objects under investigation.

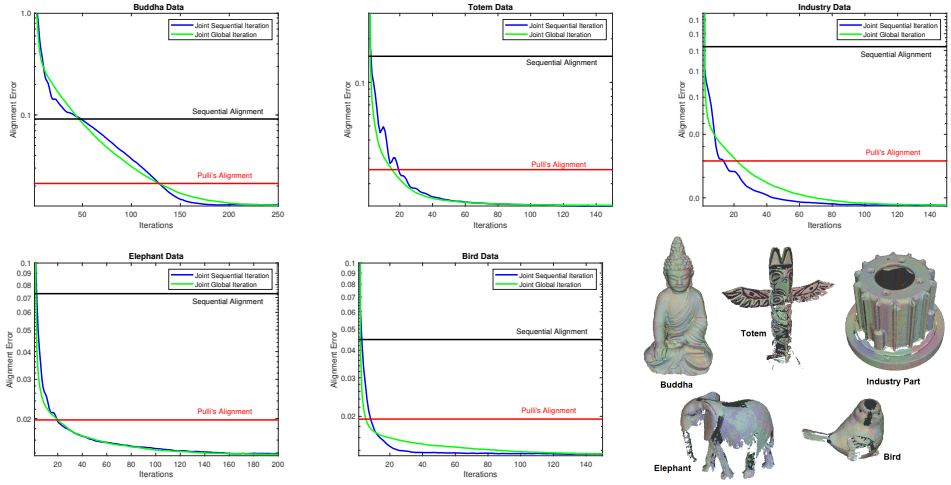


Figure 7.3.: *Convergence behavior for five independent data sets. Alignment error of the naive pairwise approach is given by the black line (contains drift error). State of the art is given by Pulli's approach (red line). Jointly iterating approaches converge to much lower errors. While the sequential procedure (blue) may alternate depending on the data, the global approach (green) converges smoothly.*

The plots in Figure 7.3 show the convergence behaviour of the alignment strategies for increasing numbers of iterations. Both methods that were proposed converge to significantly lower errors than the trivial sequential pairwise alignment (black line) and Pulli's drift preventing procedure (red line). Although both, the *Joint Sequential ICP* and the *Joint Global ICP* converge to the same optimum, the alignment error of the sequential variant occasionally alternates depending on the data (see *Totem*). In contrast, the global approach converges completely smoothly and evenly, which leads to a more stable convergence in general.

7.6.1. Stopping Criterion

The smooth convergence behaviour of the presented global ICP variant provides a considerable advantage over all previous ICP methods. Most of them alternate during the procedure, due to iteratively updated point correspondences, which increases the chance of getting stuck in local minima.
Moreover, the error often starts to alternate around the minimum. Most papers write "we iterate until the alignment error does not reasonably improve any more" without further information. Standard stopping criteria for convergence do not hold in most situations, since the differences between two subsequent iterations may not fall under a given threshold. This is the reason why in many practical implementations the alignment does continue and start to tremble until it is manually stopped.
Figure 7.4 shows the behavior of the weighted alignment errors for the investigated datasets. Left plot shows the behavior for the *Joint Sequential ICP* and right plot for *Joint Global ICP*. A simple smooth stopping strategy like checking for improvements of the alignment within the last few iterations enables the stable automatic termination of the procedure after a reasonable number of iterations.
The stopping points are visualized by the stars in Figure 7.4.

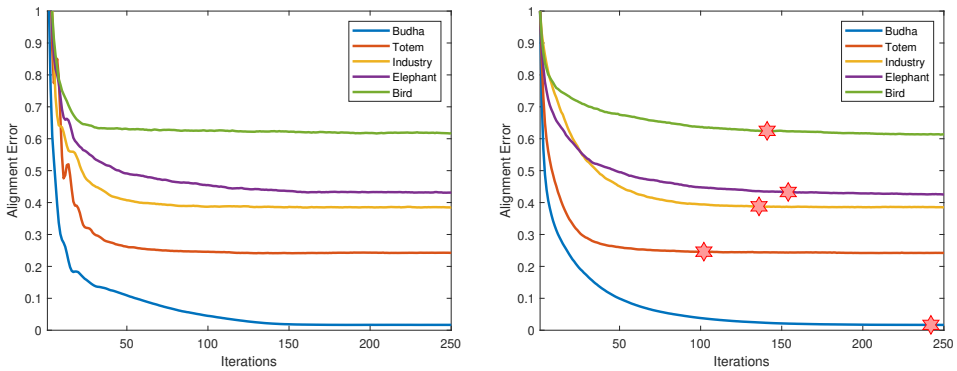


Figure 7.4.: *Alignment errors of the proposed joint sequential (left) and global (right) ICP variants. Red stars mark automatic stopping points of the global method.*

7.7. Conclusion

In this chapter a procedure has been presented that aligns complete closed turns of partial point clouds jointly in a global manner. Not only pairwise adjacent point clouds are considered but also corrected symmetrically to all neighbors. The usual, widely spread ICP procedure can be applied in a slightly adapted way. By iterating the sub-problems jointly, alignment and calibration errors are evenly distributed over all scans and drift is prevented. The global approach leads to a smooth convergence behaviour, which enables the credible application of automatic stopping criteria. Together with an introduced outlier rejection strategy, this results in an extremely stable automatic procedure. The results are moreover achieved without any user interaction or additional computational effort. To ease reproduction, the procedure is finally attached as pseudo-code.

Algorithm 2: Joint Global ICP for Full Turn Alignment

Input: Initially aligned partial point clouds $P_0, ..., P_{S-1}$ of scans $\mathcal{S}, ..., \mathcal{S}_{S-1}$.

Initialize parameters $\mathbf{R}^{(s)} = \mathbf{I}$, $\mathbf{t}^{(s)} = \mathbf{0}$ for all partial scans $\mathcal{S}_s$, $s = 0, ..., S-1$.

Setup a k-d-tree $\mathcal{T}_s$ for each point cloud P_s.

Sample point clouds P_s to a size of N elements.

for $i = 0, 1, 2, ...$ **do**

  for $s = 0, ..., S-1$ **do**

    Search in k-d-trees of adjacent scans for correspondences:

$$\mathcal{T}_{s-1}\Big(\mathbf{R}^{(s-1)^\mathsf{T}}\big(\mathbf{R}^{(s)}P_s + \mathbf{t}^{(s)} - \mathbf{t}^{(s-1)}\big)\Big) \rightarrow P^{(s,s-1)}$$

$$\mathcal{T}_{s+1}\Big(\mathbf{R}^{(s+1)^\mathsf{T}}\big(\mathbf{R}^{(s)}P_s + \mathbf{t}^{(s)} - \mathbf{t}^{(s+1)}\big)\Big) \rightarrow P^{(s,s+1)}$$

    Reject outliers as introduced in Sec. 7.5.

    Weight correspondences with respect to point distances.

    Subtract centroids from point sets:

$$\mu_{s,s-1} = \sum_{n=0}^{N-1} w_n \mathbf{x}_n^{s,s-1} \quad \rightarrow Q^{(s,s-1)} = P^{(s,s-1)} - \mu_{s,s-1}$$

$$\mu_{s,s+1} = \sum_{n=0}^{N-1} w_n \mathbf{x}_n^{s,s+1} \quad \rightarrow Q^{(s,s+1)} = P^{(s,s+1)} - \mu_{s,s+1}$$

  end

  for $s = 0, ..., S-1$ **do**

    Set up symmetric system matrices:

$$\mathbf{H}_{s,s-1} = \sum_{n=0}^{N-1} w_n \mathbf{q}_n^{(s,s-1)} \mathbf{q}_n^{(s-1,s)^\mathsf{T}}, \quad \mathbf{H}_{s,s+1} = \sum_{n=0}^{N-1} w_n \mathbf{q}_n^{(s,s+1)} \mathbf{q}_n^{(s+1,s)^\mathsf{T}}$$

$$\rightarrow \mathbf{H}_s = \mathbf{H}_{s,s+1}\mathbf{R}^{(s+1)^\mathsf{T}} + \mathbf{H}_{s,s-1}^\mathsf{T}\mathbf{R}^{(s-1)^\mathsf{T}}$$

    Compute SVD $\mathbf{H}_s = \mathbf{U}_s\mathbf{\Lambda}_s\mathbf{V}_s^\mathsf{T}$ and compose updated rotation $\mathbf{R}^{(s)} = \mathbf{V}_s\mathbf{U}_s^\mathsf{T}$.

  end

  for $s = 0, ..., S-1$ **do**

    Update translation vectors $\mathbf{t}^{(s)}$ by Eq. 7.8.

  end

  Compute weighted alignment error and check for improvement withing the last say

  10 iterations. If no improvement break.

end

Output: Optimal transformations $\mathbf{R}^{(s)}$, $\mathbf{t}^{(s)}$.

Chapter 8

Object Representation

Contents

With the methods presented so far, highly accurate 3D reconstructions in the form of point clouds (unstructured set of 3D points) can be generated. This is already suitable for a variety of applications in which only geometric measurements are required, such as for example automatic quality control or object inspection methods [6]. However, it is often desirable to generate 3D models that can be rendered realistically or at least as close to as possible. In this way, they also become suitable for impressions from human observers.
This chapter will briefly address how to compute normal vectors for point clouds generated with the presented procedure. A simple method will be shown, that allows efficient computation of normal vectors for semi-dense point clouds reconstructed from images. The introduced method does not require a fixed neighborhood, so that it is suitable for scenes reconstructed with different densities. Although the method has not been published due to its simplicity, it comes from own research and, to our knowledge, has not been used in this way anywhere else.
Subsequently, to complete the general reconstruction pipeline, it will be briefly mentioned how meshes can be generated from the calculated points and normal vectors. These allow complex geometries to be represented with desired accuracy in a memory-saving manner. Finally, high-resolution textures can be mapped onto these geometries to show color details in the model.

8.1. Normal Vector Estimation

Standard approaches, such as the established and widely used approach of Hoppe *et al.* [68], compute the k-nearest neighbors for each point in the set and subsequently fit planes to the chosen environments. Mitra and Nguyen [118] extended this approach to deal with noisy data as well. These methods give good results, but depend on a well-chosen neighborhood size, which can cause problems when working with inhomogeneous data.
Moreover, the extraction of 3D neighborhoods in full-resolution point clouds is particularly costly, even when k-d-trees are used. In 3D reconstruction, where the underlying data comes from 2D images, it is possible to embed the spatial data again into 2D images. This allows further steps to benefit from the embedded topology in 2D. An approximation of neighborhoods in this way was presented by Holzer *et al.* in [67].
Since the method presented in the following is independent of explicitly chosen neighborhoods, it is ideally suited for inhomogeneous data that strongly varies in terms of density. By using a Gaussian weighting that assigns closer pixels always a higher influence than more distant points, consistent influences of surrounding points can be obtained even for very isolated pixels. Masking of uncoded pixels is considered throughout. In the following, will be shown how neighborhood information can be extracted over the entire image by 2D convolutions, taking into account weighting and masking of all points in the image. Since these operations can be performed in Fourier space, kernels of arbitrary size can be applied without expensive computational overhead. This makes it possible to consistently ensure the influence of even distant neighbors on isolated pixels.

Vertex Maps for Topology Approximation in 2D Assume we are given a *depth map* $D \in \mathbb{R}^{H\times W}$ of a scene and a binary mask $\Lambda \in \mathbb{R}^{H\times W}$ that specifies if a pixel $D(x,y)$ carries information or not. Given the camera's intrinsic calibration matrix $\mathbf{K}$, a 3D point $V(x,y) \in \mathbb{R}^3$ can be computed for every encoded pixel (x,y) in D by scaling the out-projected ray to the desired depth:

$$V(x,y) = \frac{\mathbf{K}^{-1}(x,y,1)^\mathsf{T}}{\|\mathbf{K}^{-1}(x,y,1)^\mathsf{T}\|_2} \cdot D(x,y) \tag{8.1}$$

The three channel image $V \in \mathbb{R}^{H\times W\times 3}$, which stores the 3D position for each pixel is called *vertex map*. Approximating 3D neighborhoods by the image neighborhood yields a great computational advantage. Further, all calculations can then be performed by 2D convolutions as will be shown in the following.

Normals from Masked Vertex Maps The idea is to find a normal vector $\mathbf{n}$ for each reconstructed point $V(x,y)$ that fits as well as possible into the sampled object surface. Thereby, $\mathbf{n}$ is supposed to minimize the following functional (8.2) describing the angular error of $\mathbf{n}$ to be orthogonal to all lines between the point $V(x,y)$ and the points $V(\tilde{x},\tilde{y})$ in a given region weighted

by a Gaussian G_σ.
To ensure that un-coded points do not adversely affect the normal calculation, only values in the particular neighborhood for which the binary mask specifies $\Lambda(\tilde{x},\tilde{y}) = 1$ should be used. So for each point in the vertex map $V(x,y)$ with $\Lambda(x,y) = 1$ the following optimization problem is solved:

$$\underset{\mathbf{n}\in\mathbb{R}^3}{\operatorname{argmin}} \sum_{\substack{V(\tilde{x},\tilde{y}) \\ \text{with } \Lambda(\tilde{x},\tilde{y})=1}} G_\sigma\big(V(x,y) - V(\tilde{x},\tilde{y})\big) \cdot \big\|\big(V(x,y) - V(\tilde{x},\tilde{y})\big)^\mathsf{T}\mathbf{n}\big\|_2^2 \tag{8.2}$$

The large advantage of vertex maps is the possibility to efficiently approximate the Gaussian in image space:

$$G_\sigma\big(V(x,y) - V(\tilde{x},\tilde{y}\big) \approx G_\sigma\big((x,y) - (\tilde{x},\tilde{y})\big) \tag{8.3}$$

In this way, the Gaussian is reduced from a 3D to a 2D version with still great expressiveness. Therefore, also the problem reduces to:

$$\underset{\mathbf{n}\in\mathbb{R}^3}{\operatorname{argmin}} \sum_{\substack{(\tilde{x},\tilde{y}) \\ \text{with } \Lambda(\tilde{x},\tilde{y})=1}} G_\sigma\big(V(x,y) - V(\tilde{x},\tilde{y})\big) \cdot \big\|\big(V(x,y) - V(\tilde{x},\tilde{y})\big)^\mathsf{T}\mathbf{n}\big\|_2^2 \tag{8.4}$$

$$\approx \underset{\mathbf{n}\in\mathbb{R}^3}{\operatorname{argmin}} \sum_{\substack{(\tilde{x},\tilde{y}) \\ \text{with } \Lambda(\tilde{x},\tilde{y})=1}} G_\sigma(x-\tilde{x}, y-\tilde{y}) \cdot \big\|\big(V(x,y) - V(\tilde{x},\tilde{y})\big)^\mathsf{T}\mathbf{n}\big\|_2^2 \tag{8.5}$$

$$= \underset{\mathbf{n}\in\mathbb{R}^3}{\operatorname{argmin}} \sum_{(\tilde{x},\tilde{y})} \Lambda(\tilde{x},\tilde{y}) \cdot G_\sigma(x-\tilde{x}, y-\tilde{y}) \cdot \big\|\big(V(x,y) - V(\tilde{x},\tilde{y})\big)^\mathsf{T}\mathbf{n}\big\|_2^2 \tag{8.6}$$

Computing the gradient with respect to $\mathbf{n}$ and setting it equal to zero leads to

$$\sum_{(\tilde{x},\tilde{y})} \Lambda(\tilde{x},\tilde{y}) G_\sigma(x-\tilde{x}, y-\tilde{y}) \cdot \big(V(x,y) - V(\tilde{x},\tilde{y})\big)\big(V(x,y) - V(\tilde{x},\tilde{y})\big)^\mathsf{T}\mathbf{n} \overset{!}{=} \mathbf{0} \tag{8.7}$$

which is moreover equivalent to solving a problem of the form

$$\mathbf{A}\mathbf{n} \overset{!}{=} \mathbf{0} \tag{8.8}$$

with matrix $\mathbf{A}$ set up as

$$\begin{aligned}\mathbf{A} = V(x,y)V(x,y)^\mathsf{T}(\Lambda * G_\sigma)(x,y) &+ \begin{pmatrix} \widetilde{V}_{11}(x,y) & \widetilde{V}_{12}(x,y) & \widetilde{V}_{13}(x,y) \\ \widetilde{V}_{12}(x,y) & \widetilde{V}_{22}(x,y) & \widetilde{V}_{23}(x,y) \\ \widetilde{V}_{13}(x,y) & \widetilde{V}_{23}(x,y) & \widetilde{V}_{33}(x,y) \end{pmatrix} \\ &- \mathbf{p}(x,y)\begin{pmatrix} \widetilde{V}_1(x,y) \\ \widetilde{V}_2(x,y) \\ \widetilde{V}_3(x,y) \end{pmatrix}^\mathsf{T} - \begin{pmatrix} \widetilde{V}_1(x,y) \\ \widetilde{V}_2(x,y) \\ \widetilde{V}_3(x,y) \end{pmatrix}\mathbf{p}(x,y)^\mathsf{T}\end{aligned} \tag{8.9}$$

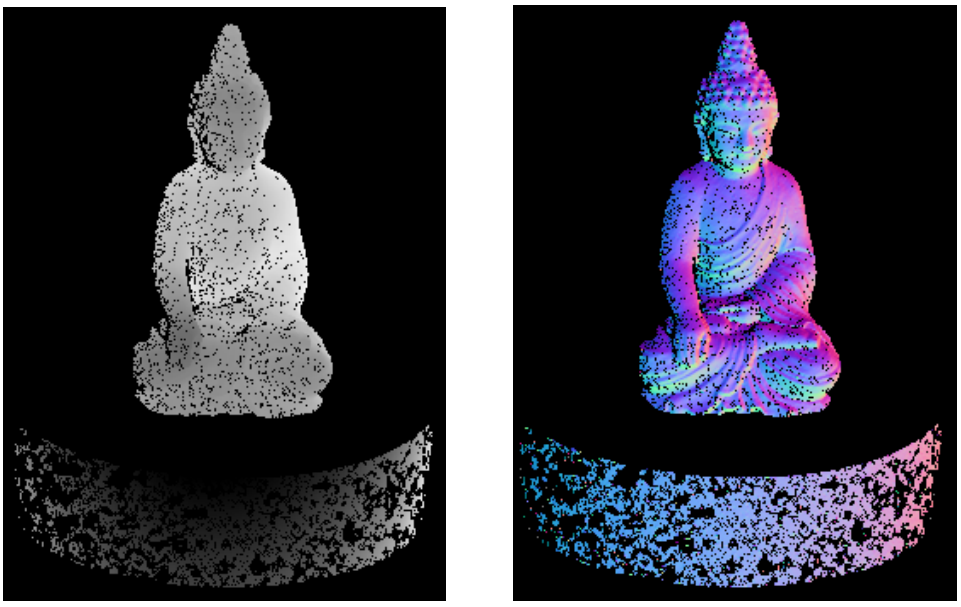
using the following convolved vertex maps, that only need to be calculated once:

$$\widetilde{V}_i = (\Lambda \odot V_i) * G_\sigma, \qquad \widetilde{V}_{ij} = (\Lambda \odot V_i \odot V_j) * G_\sigma, \qquad i,j = 1,...,3, j \geq i \quad (8.10)$$

Thereby, $\odot$ denotes the point-wise multiplication and $*$ the 2D convolution that can be efficiently performed in Fourier space independently of the Kernel size G_σ which leads to an overall complexity of $\mathcal{O}\big(HW \log(HW)\big)$ only. Finally, $\mathbf{n}(x,y)$ is computed by applying *Singular Value Decomposition* on system matrix $\mathbf{A}$, similar to the problem in 2.4. Due to usually full rank of system matrix $\mathbf{A}$, the optimal normal vector is given by the singular vector with respect to the smallest singular value of $\mathbf{A}$:

$$\text{SVD}(\mathbf{A}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T} \qquad \Rightarrow \qquad \mathbf{n}(\tilde{x},\tilde{y}) = \mathbf{V}_3$$

Figure 8.1 shows the resulting normals, applied to the investigated exemplary scene.



(a) Depth map **(b)** Normal map

Figure 8.1.: *Example of a normal map (b) computed for a semi-dense depth map (a) with a Gaussian kernel with standard derivation $\sigma = 0.5$. Due to the integration of the mask into the optimization process missing points do not negatively affect the method.*

8.2. Outlook: Meshing and Texturing

After performing the methods from Chapters 4, 5 , 7 and Sections 2.4 and 8.1, a fairly dense point cloud of a full object and associated normal vectors are given. This allows already to create a representation of the object with light interactions, which enables a visual perception of the geometry by humans. However, this representation has some disadvantages that make it unsuitable for most practical applications:

- Storing all points and associated normal vectors is extremely memory intensive and especially too expensive for regions of the object surface where little details are given.
- No matter how densely a point cloud samples the surface, there will always be sparsely scanned areas and holes due to different surface properties of an object. At the same time there will always be overlapping areas where the partial scans have been aligned. This leads to inconsistencies and redundancies in the representation of the object surface.
- Some applications, such as many 3D printers, require watertight representations of an object, which can not be achieved with point clouds.

A long-established approach for memory-efficient representation of 3D geometry is to compute a mesh, comprised from connected faces that represents the surface, which is sampled by the point cloud as closely as possible. Only the vertices of each face and their connections to each other need to be stored. This usually results in significant cost savings in data storage, since only few vertices need to be considered for smooth surfaces. Remaining capacity can be used for finer resolution of more detailed areas. Thereby, the mesh should be designed in a way that the orientation of the individual faces corresponds as closely as possible to the given normal vectors of the point cloud (up to multiplication by -1).
A widely used method is *Poisson Surface Reconstruction* [86], which was introduced by Kazhdan *et al.*, that can be followed by well established meshing procedures such as *Marching Cubes* [100]. These procedures provide an approximation of the surface from given point clouds and respective normal vectors. In *Poisson Surface Reconstruction* the gradient field of local environments is approximated, leading to a reduction of the problem to a Poisson equation. These types of equations have been studied extensively in the past and can be approximated using finite elements or finite differences methods ([122], [81], [26]). By using an octree structure to efficiently search for neighborhoods within the data, the computation time is significantly reduced. The fineness of the obtained mesh can be controlled by the depth of the octree used. Nevertheless, a considerable computational effort remains in the execution of the method.
Some time later, Kazhdan and Hoppe extended their approach of *Poisson Surface Reconstruction* to *Screened Poisson Surface Reconstruction* [87]. In this process, they explicitly added the given points as constraints and obtained sharper details in the mesh in areas where guiding points are present.

(a) Plain and textured mesh of *Buddha*

(b) Plain and textured mesh of *Totem*

Figure 8.2.: *Meshes of the reconstructed point clouds of the Buddha and the Totem data. Plain meshes (a, b, left) and respective textured ones (a, b, right).*

By adding this constraint, the Poisson equation has been transferred into a screened Poisson equation, which can be solved with comparable computational effort as the previous problem. However, a screening parameter needs to be chosen, in order to weight the additional term. Thus, the detail of the resulting mesh is weighed against its susceptibility to noise. Figure 8.2 shows two meshes reconstructed using the *Screened Poisson Meshing*, which has been applied to the point clouds resulting from Section 7.
Even though these methods work well and are widely used, they are time consuming. Recent methods such as [60] and [39] address meshing from point clouds with approaches based on artificial neural networks. These could significantly reduce the time aspect of meshing in the post-processing of any reconstruction pipeline.
In order to add color to the fitted mesh, it is necessary to identify, for all the faces of the mesh, the cameras that are best suited to map their captured texture. The quality of a camera texture to a given face can be estimated using the face normals and the associated camera position and its angle of view. For surfaces that are seen by multiple cameras, it is a common strategy to blend the textures by weighting with the angular errors.
As a reference Fu *et al.* [51] recently presented an approach for parameterizing and texturing meshes, that can be applied to the presented method, where all the devices, including the texture camera are carefully calibrated. Finally, Figure 8.2 (a, b, right) shows the textured meshes resulting from the presented pipeline so far.

Chapter 9

Speeding Up The Acquisition

Contents

9.1. Introduction

The self-calibrating approach, presented in Chapters 4 and 5, is extremely flexible, but requires a large number of image acquisitions of the scene with multiple patterns projected. In this chapter, a new approach will be presented that encodes the scene simultaneously in horizontal and vertical directions using sinusoidal fringe patterns. This allows to almost halve the number of recorded images, making the approach attractive for many practical applications with time aspects. As explained in Chapter 2, the frequency of the projected fringes is increased several times, depending on the required accuracy, in order to successively improve the quality of the encoding. The high number of camera shots required, leads to a considerable expenditure of time in data acquisition, which is one of the main weaknesses of the method.
According to the state of the art, the phase shifting method is the basis of

sinusoidal structured light methods, due to its texture invariance and the surface encoding in full camera resolution. Thereby, sinusoidal fringe patterns with equidistantly shifted phases are projected onto the scene. A superposed phase, which can be calculated from at least three shifted patterns, encodes the scene pixel by pixel in the direction of the phase shifts. Doing this, both in horizontal and vertical direction, results in a minimum of 6 captured images. Even more acquisitions are necessary if further refinement steps with higher frequencies are performed. The patterns are sinusoidally modulated in the direction to be encoded and constantly continued in the decoded direction. Therefore, one dimension of the patterns (the direction orthogonal to the encoded one) does not carry any information, which leaves room for new options. In the following, more detailed investigations on the phase shifting algorithm and the harmonic addition theorem are carried out. Especially, findings with regard to the resulting amplitude of the phase superposition will encourage to combine the horizontal and vertical patterns in order to encode both directions simultaneously. The horizontal and vertical phases are then extracted from the combined patterns by a per pixel strategy, making the whole procedure scene-independent.

9.2. Related Work

Extensive research has been carried out in the field of structured light reconstruction. Various strategies, assuming pre-calibrated setups, have been reviewed and compared by Salvi *et al.* [142] and more recently by Zanuttigh *et al.* [179]. Popular methods, based on the phase shifting algorithm, have been reviewed by Servin *et al.* in [146]. In the mean time, new approaches, like the Fourier-based regularized method of Legarda-Saenz and Espinosa-Romero [92] came up which, however, could not compete with the state of the art.
Based on the phase shifting method, Mirdehghan *et al.* [117] recently presented a procedure to generate optimal scene-dependent projection patterns and thus to control the quality of the resulting encoding. Zhang and Yau presented in [186] a system with two cameras that offers many quality advantages in contrast to standard setups with one camera and one projector.
Unfortunately, the recording time is the great weakness of the structured light method. One way to shorten the required acquisition time is to distribute several patterns among the different color channels of the cameras and projectors used ([80], [181]). These approaches work in theory but suffer from color cross-talk between cameras and projector and a very accurate color calibration is required. In particular, the object color influences the type and strength of the cross-talk. This leads to difficulties in implementing the procedures in practice and even then, one has to expect large quality losses.
To reduce the number of acquisitions required, Guan *et al.* [57] and Sansoni and Redaelli [143] combine patterns of different frequencies into individual patterns. Several fringes are encoded by carrier waves and additively combined. Afterwards they are separated by filter methods. These methods also work in theory, but have a poor applicability in practice. The frequencies of the

carrier waves must be stable in the image to enable an appropriate extraction, which cannot be achieved for arbitrary 3D scenes. Nevertheless, they made it possible for the first time to provide information in the un-coded direction of the patterns. Based on this idea, Yang *et al.* [174] further improved this approach, and created special patterns based on co-prime frequent sine waves that can be separated more robustly by a Garbor filter. Recent advances in real-time measurement with structured light have been detailed and analyzed by Zhang in [183]. Finally, Wang and Yang [169] recently introduced a one-shot approach based on binary stripe patterns, from which the phase can be directly approximated and interpolated. However, the approach assumes the stripes to be continuously visible in the scene, which cannot be guaranteed for general scenarios.
All in all, the task of combining multiple patterns has already been regarded, but has not been solved satisfactorily, yet. In particular, the combination of horizontal and vertical encoding patterns has not been addressed, before.

9.3. Mathematical Investigation

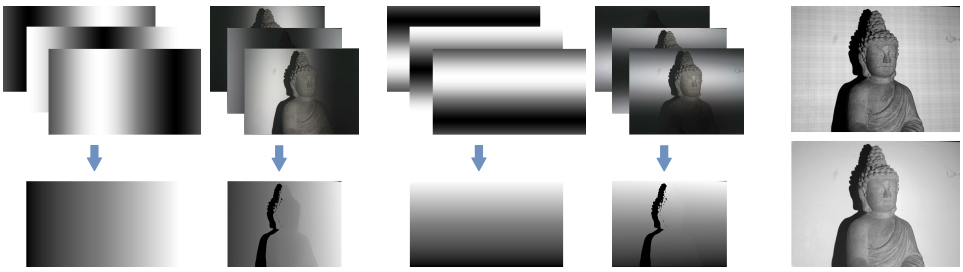
In order to develop a new projection pattern, that allows to simultaneously recover the horizontal and vertical phases, the standard case is investigated more closely. This will provide new insights into the amplitude of the superposition of the illuminated scene. These findings will finally enable a subsequent separation of the phase directions from the combined patterns.

9.3.1. Background: Sinusoidal Phase Shifting Method

Basis of the presented work is the *sinusoidal phase shifting* method. Thereby, patterns are modulated by sine or cosine (convention-dependent) signals and equidistantly shifted at least three times over the periodic domain. The patterns are projected onto the scene and captured by a camera. Superposition of the resulting images allows to encode the scene texture-invariantly in full camera resolution. Horizontal and vertical directions are treated by separate sets of patterns, each modulated by a sine/cosine in the respective direction and continued constantly in the other direction. Horizontal and vertical sets of patterns P_n^H and P_n^V with frequencies F_H and F_V that are shifted $n = 1, ..., N$ times, can be explicitly generated as:

$$\begin{aligned} P_n^H(i,j) &\coloneqq \cos\left(\frac{2\pi j}{width}F_H + \frac{2\pi(n-1)}{N}\right), \\ P_n^V(i,j) &\coloneqq \cos\left(\frac{2\pi i}{height}F_V + \frac{2\pi(n-1)}{N}\right) \end{aligned} \tag{9.1}$$

Thereby, $i = 1, ..., height$ and $j = 1, ..., width$ denote the image pixels. The first row of Figure 9.1 shows an exemplary set of horizontal (a) and vertical (b) patterns with $N = 3$ and $F_H = F_V = 1$. In both cases, the patterns are shown to the left, and the projection of the patterns onto an exemplary scene is shown to the right.



(a) Horizontal phase recovery **(b)** Vertical phase recovery **(c)** Amplitudes

Figure 9.1.: *Example of the phase shifting algorithm for encoding a scene by phase recovery via harmonic addition theorem. The top rows of (a) and (b) show sets of horizontal and vertical fringe patterns and the resulting scene after projection. The bottom rows show the phase images computed by Equation (9.3). (c) shows the amplitude of the superposition (Equation (9.4)) (top) and the one given by the scaled texture as introduced in Lemma 9.2 (bottom).*

Harmonic Addition Theorem

Structured light approaches with sinusoidal patterns are based on practical application of the *harmonic addition theorem* as introduced in Section 2.3.2:

$$\sum_{n=1}^{N} I_n \cos(\delta_n) = A\cos(\Phi) \tag{9.2}$$

$$\text{with} \quad \Phi = \operatorname{atan2}\left(\sum I_n \sin(\delta_n),\ \sum I_n \cos(\delta_n)\right) \tag{9.3}$$

$$\text{and} \quad A^2 = \sum_{n=1}^{N}\sum_{m=1}^{N} I_n I_m \cos(\delta_n - \delta_m)\ , \tag{9.4}$$

where δ_n, δ_m denote the equidistant phase shifts, Φ the phase to be recovered and A the amplitude of the superposition. atan2 denotes the two-dimensional *arcus tangens* function taking into account the quadrants of the input.

Recovering the Phases in the Scene

Projecting patterns from Equation (9.1) to a scene I results in images I_n^H and I_n^V for the different phase shifts $n = 1, ..., N$. Applying Equation (9.3), the horizontal and vertical phases Φ_H and Φ_V can then be computed by:

$$\begin{aligned} \Phi_H &= \operatorname{atan2}\left(\sum I_n^H \sin(\delta_n),\ \sum I_n^H \cos(\delta_n)\right) \\ \Phi_V &= \operatorname{atan2}\left(\sum I_n^V \sin(\delta_n),\ \sum I_n^V \cos(\delta_n)\right) \end{aligned} \tag{9.5}$$

The second rows of Figure 9.1 (a) and (b) show the recovered phases computed for the patterns and the scene. Using this information, the scene points are uniquely encoded by their horizontal and vertical phase values. This allows robust and dense matches between the different camera views and the projector to be achieved. Amplitude A is not needed here and usually not treated further. For the method that is developed in this chapter, A plays an important role and is therefore further investigated.

9.3.2. Amplitude of Superposition

From illuminated images I_n, amplitude A is given by Equation (9.4), which can be directly expressed in terms of a captured texture image I of the scene. This will be formally proven and therefore assumed in the following. These findings will make subsequent procedures possible. Firstly, the following lemma is needed. It is well known, that equidistantly shifted cosines sum up to zero. This is also the case for integer multiples of the shifts. Formally the following holds true:

Lemma 9.1 *Cosines with equidistantly shifted phases sum up to zero, even for integer multiples of the shifts:*

$$\sum_{n=1}^{N} \cos(x + k \cdot \delta_n) = 0 \ , \quad \textit{with } \delta_n = \frac{2\pi n}{N} \ , \ k \in \mathbb{Z} \tag{9.6}$$

Proof The proof is straight forward using the *Euler Formula* $(*_1)$ and the *Formula of Geometric Series* $(*_2)$:

$$\sum_{n=1}^{N} \cos(x + k \cdot \delta_n) = \sum_{n=0}^{N-1} \cos(x + k \cdot \delta_n) \tag{9.7}$$

$$\overset{(*_1)}{=} \sum_{n=0}^{N-1} \frac{1}{2}\left(e^{ix+ik\delta_n} + e^{-ix-ik\delta_n}\right) \overset{(*_2)}{=} \frac{e^{ix}}{2}\frac{\overbrace{e^{i2k\pi}}^{=1}-1}{e^{i\frac{2k\pi}{N}}-1} + \frac{e^{-ix}}{2}\frac{\overbrace{e^{-i2k\pi}}^{=1}-1}{e^{-i\frac{2k\pi}{N}}-1} = 0 \tag{9.8}$$

■

Using this knowledge it is possible to proof the following fundamental information about A in Equation (9.2):

Lemma 9.2 *A captured scene I_n, illuminated by patterns $P_n = sin(x + \delta_n)$, with an arbitrary number of equidistant shifts $n = 1, ..., N \geq 2$, can be superposed to*

$$\sum_{n=1}^{N} I_n \cos(\delta_n) = \frac{N}{4} I \cos(\Phi) \ , \tag{9.9}$$

where Φ denotes the phase angle and I the fully illuminated scene. Therefore, the amplitude of the superposition is given by a scaled version of the scene image I.

Proof Using the *Harmonic Addition Theorem* the lemma is fully proven if $A = \frac{N}{4} I$ is shown to be true:

$$A^2 = \sum_{n=1}^{N}\sum_{m=1}^{N}\left(P_n \odot \frac{I}{2}\right)\left(P_m \odot \frac{I}{2}\right)\cos(\delta_n - \delta_m) \tag{9.10}$$

$$= \frac{I^2}{4}\sum_{n,m} P_n P_m \cos(\delta_n - \delta_m) \tag{9.11}$$

$$= \frac{I^2}{4}\sum_{n,m} \cos(x + \delta_n)\cos(x + \delta_m)\cos(\delta_n - \delta_m) \tag{9.12}$$

$$= \frac{I^2}{16}\sum_{n,m} \cos(0) + \cos(2x + 2\delta_n) + \cos(2x + 2\delta_m) + \cos(2\delta_n - 2\delta_m) \tag{9.13}$$

$$\overset{\text{Lemma 9.1}}{=} \frac{N^2}{16} I^2 \ , \tag{9.14}$$

where I denotes the fully projected scene and $\odot$ denotes element-wise multiplication of corresponding camera and projector points. x denotes any position in the projector image. Taking the square root proves the lemma. ■

Figure 9.1 (c) shows the amplitudes of the example scene computed by Equation (9.4) (top) and from scaled texture as in Lemma 9.2 (bottom). Apart from artifacts caused by clipping, due to limited dynamic range of the cameras and gamma corrections of the devices, these are identical. The lemma can be proven straight forwardly using properties of trigonometric functions and the harmonic addition theorem.

9.3.3. Combined Patterns

In the following, additively combined patterns are going to be introduced and a mathematical problem with the newly gathered information about the amplitude is set up. Solving this problem enables the simultaneous recovery of horizontal and vertical phase values. Let the combined patterns P_n^C be defined as

$$P_n^C := \frac{1}{2}(P_n^H + P_n^V) \quad \text{for } n = 1, ..., N. \tag{9.15}$$

Thereby, two-dimensional sinusoidal patterns result as visualized in Figure 9.2 (a, left) and projected onto the scene (b, left). The shifting direction naturally becomes the diagonal. The task in the following is to extract the horizontal as well as the vertical phase simultaneously from images of the scene, illuminated by these patterns. Assuming the optimal case, where cameras and projector respond linearly and do not perform any gamma correction or internal post-processing, a captured scene of a combined pattern I^C is proportional to the sum of the separately illuminated scenes I^H and I^V:

$$I^C = P^C \odot \frac{I}{2} = \frac{1}{2}(P^H + P^V) \odot \frac{I}{2} = \frac{1}{2}(I^H + I^V), \tag{9.16}$$

where $\odot$ denotes pixel-wise multiplication of the patterns and the scene appearance I.

(a) Ideal phases computed from patterns **(b)** Scene phases from camera images

Figure 9.2.: *Combined sinusoidal patterns, computed by Equation (9.15) and projected onto the scene. The horizontal and vertical phases, to be recovered, are shown to the right.*

Problem Formulation

Lemma 9.2 and Equation (9.16) directly deliver the basic properties to set up the problem to be solved, in order to extract the phase information:

$$\begin{aligned} 2\sum_{n=1}^{N} I_n^C \cos(\delta_n) &= \frac{N}{4} I \cos(\Phi_H) + \frac{N}{4} I \cos(\Phi_V) \\ 2\sum_{n=1}^{N} I_n^C \sin(\delta_n) &= \frac{N}{4} I \sin(\Phi_H) + \frac{N}{4} I \sin(\Phi_V) \end{aligned} \tag{9.17}$$

This gives two equations with respect to two phase values, that have to be recovered from the superpositions per pixel. In the following section the problem is treated strictly mathematically, before it is again applied to the real world.

9.3.4. Mathematical Solution to the Problem

Given the following system of equations:

$$\begin{aligned} a\cos(x) + a\cos(y) &= b \\ a\sin(x) + a\sin(y) &= c \end{aligned} \tag{9.18}$$

with measured data a, b, c, the task is to compute optimal values x, y that solve both equations. Using addition theorems of trigonometric functions and dividing the equation leads to:

$$\begin{aligned} 2a\cos(\frac{x}{2}+\frac{y}{2})\cos(\frac{x}{2}-\frac{y}{2}) &= b \\ 2a\sin(\frac{x}{2}+\frac{y}{2})\cos(\frac{x}{2}-\frac{y}{2}) &= c \end{aligned} \quad \Rightarrow \quad x + y = 2\arctan\left(\frac{c}{b}\right) \tag{9.19}$$

In this way the equations of (9.19) can be decoupled. Thereby, both lead to the same equation:

$$2ab\cos(z) + 2ac\sin(z) = b^2 + c^2 \quad z \in \{x, y\} \tag{9.20}$$

Using harmonic addition theorem, four explicit solutions for this equation can be derived, where the two feasible ones are given by

$$x/y = 2\arctan\left(\frac{2ac \pm \sqrt{(b^2+c^2)(4a^2-b^2-c^2)}}{b^2+2ab+c^2}\right). \tag{9.21}$$

9.4. Application to Real World

The left column of Figure 9.4 shows the results of Equation (9.21) applied to the system (9.17) that models the real process. If there is a significant influence of ambient light, it may be necessary to subtract an ambient image from the captured images.
With the proposed procedure, phase values Φ_H and Φ_V can be recovered robustly. Unfortunately, due to the symmetric additive superposition of the phases in the whole procedure there is no information about which of the two phase values corresponds to the horizontal and which to the vertical phase. These interchanges can occur at pixel level due to the pixel-wise approach. However, due to the natural continuity of the phases, these interchanges usually occur fragmentary. This can be seen in the first columns of the different examples of Figure 9.4 for different frequencies, for both the synthetic and the real case. Note that application directly to the patterns is meaningful for any synthetic scene, because of the scene-independent pixel-wise approach.
The first errors can be corrected by simple comparison (*comparison step*), which sorts the values to fragments, lifting the swaps from pixel to region level:

$$\Phi_H = \max\{\Phi_H, \Phi_V\} \ , \quad \Phi_V = \min\{\Phi_H, \Phi_V\} \tag{9.22}$$

The second columns of Figure 9.4 (a,b,c,d) show the effect to the respective scenes. This step is also pixel-wise and therefore scene independent.

9.4.1. Swapping Step

After this step, still many values are swapped (see Figure 9.4). A gradient based strategy could be used to assign them, which would not be per-pixel and therefore scene dependent.
Nevertheless, a common procedure, to obtain reconstructions of high accuracy, is the projection of several levels of fringe images with increasing frequencies. It is assumed that separate horizontal and vertical fringe images recorded at frequency 1 were projected in the first level, so that basic phases are available. In order to get a more applicable swapping procedure, that is per-pixel and stable in difficult situations (e.g. discontinuities in the phase) this can be done during the phase unwrapping of the higher level phases. In the following, a simple pixel-wise unwrapping strategy is presented, that can be used to perform the swapping step simultaneously by checking for consistency with the recorded images.

Per Pixel Unwrapping using Predicted Phase

Assume a wrapped phase $\hat{\Phi}$ has been computed from fringes with some frequency F and a predicted phase Φ_0 of a previous level is given, that is not wrapped. A refined phase Φ can be computed by unwrapping $\hat{\Phi}$ using information from Φ_0.

As depicted in Figure 9.3 in a perfect world the following statement would hold true:

$$F \cdot \Phi = \hat{\Phi} + k \cdot 2\pi, \qquad k \in \mathbb{Z}_{\geq 0} \tag{9.23}$$

In a scenario with carefully increased frequencies, one can at least assume that there are no jumps larger than π from one level to the next one, which means

$$\hat{\Phi} + k \cdot 2\pi - F \cdot \Phi_0 \leq \pi \quad \Rightarrow \quad k = \left\lfloor \frac{F \cdot \Phi_0 - \hat{\Phi} + \pi}{2\pi} \right\rfloor \tag{9.24}$$

with floor rounding $\lfloor \cdot \rfloor$. Therefore, $\hat{\Phi}$ can be explicitly unwrapped to Φ by:

$$\Phi = \frac{\hat{\Phi}}{F} + \frac{2\pi}{F} \left\lfloor \frac{F \cdot \Phi_0 - \hat{\Phi} + \pi}{2\pi} \right\rfloor \tag{9.25}$$

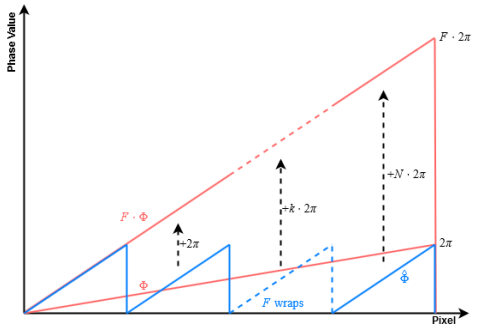


Figure 9.3.: *Illustration of the unwrapping process for an ideally wrapped linear phase, that was computed from fringes with frequency $F > 1$, resulting in a F times wrapped phase.*

Per Pixel Swapping

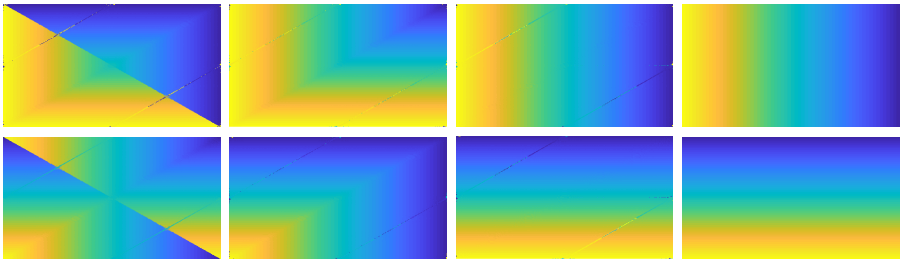
Using the per pixel unwrapping, the step can be performed as well on the phase combination (Φ_H, Φ_V) as on the swapped one (Φ_V, Φ_H). The consistency towards captured fringe images can be described by a suitable error like:

$$E_n(\Phi_H, \Phi_V) = \left| \big(\cos(F^H \Phi_H + \delta_n) + \cos(F^V \Phi_V + \delta_n) + 2 \big) I - 4 I_n^C \right| \tag{9.26}$$

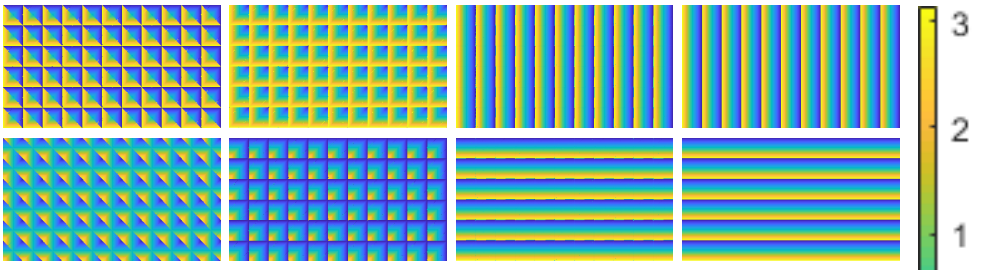
Since one can assume the refined phases to improve after every unwrapping step, the accumulated consistency error of all captured images

$$E(\Phi_H, \Phi_V) = \sum \frac{1}{l} E_n(\Phi_H, \Phi_V) \tag{9.27}$$

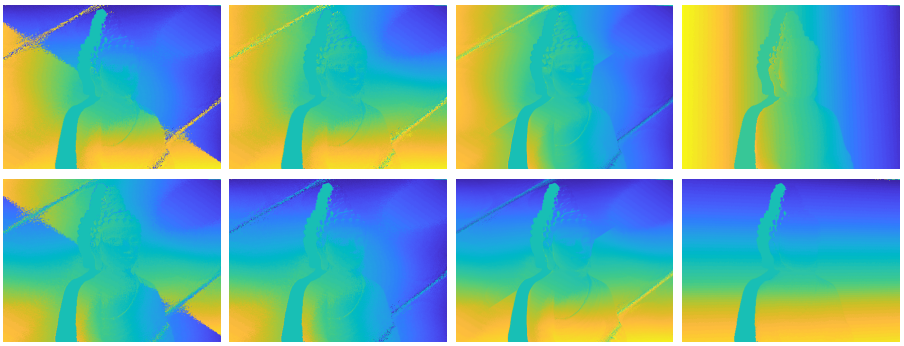
should decrease. Therefore, the combination with lower consistency error can be chosen to complete the swapping. Results of this tactic for the sample scenes are shown in the third columns of Figure 9.4 in comparison to the ground truth phases in the last columns.



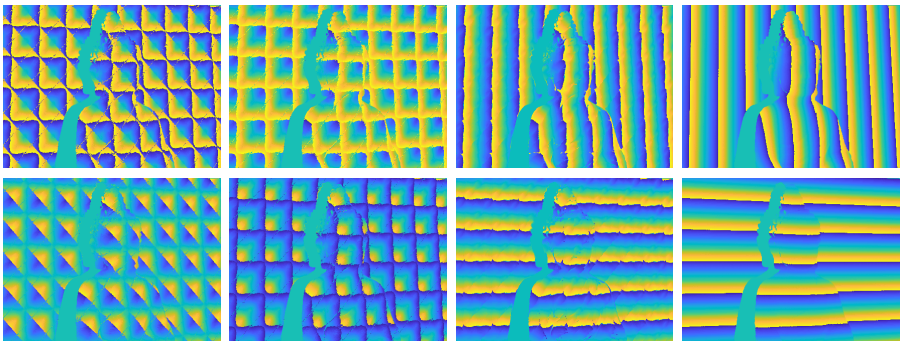
(a) Basic level of synthetic data set ($F_H = 1$, $F_V = 1$)



(b) Higher level of synthetic data set ($F_H = 11$, $F_V = 6$)



(c) Basic level of real captured data set ($F_H = 1$, $F_V = 1$)



(d) Higher level of real data set ($F_H = 11$, $F_V = 6$)

Figure 9.4.: *Results of the algorithm applied to synthetic and real data for different frequencies. For each set the two rows show the horizontal and the vertical phase. The left two columns show the results of Formula (9.21) before and after the comparison step. The third column shows the phase after the swapping step. The ground truth is depicted in the right column. The colorbar indicates the coding in the range of* $[-\pi, \pi]$.

9.4.2. Handling One-Dimensional Artifact

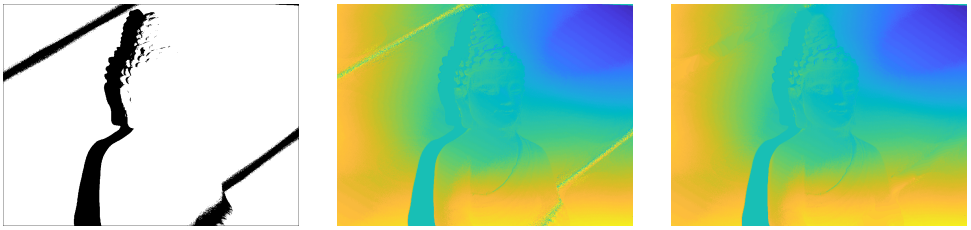
There are one-dimensional lines in every image, where both superpositions from (9.17) are equal to zero or at least close to. Close to these lines, there is no information to recover the phase. Since it is only a one dimensional region in a two dimensional encoded scene, the missing data is smoothly filled by neighboring information. Very likely, the erroneous regions of the next higher level do not coincide and therefore correct them. A sparse matrix Λ localizes and stores the erroneous stripes:

$$\Lambda(i,j) = \begin{cases} 0, & \text{if} \quad |\sum I_n^C(i,j)\cos(\delta_n)| + |\sum I_n^C(i,j)\sin(\delta_n)| < \varepsilon \\ 1, & \text{else} \end{cases} \tag{9.28}$$

The artifacts are removed by choosing pixels in the near neighborhood, which is done using a Gaussian filter G_σ specified by the variance σ:

$$\tilde{\Phi} = \Lambda \odot \Phi + (1 - \Lambda) \odot \frac{(\Lambda \odot \Phi) * G_\sigma}{\Lambda * G_\sigma} \tag{9.29}$$

Figure 9.5 shows the result on the first level of the exemplary scene.



(a) Λ after thresholding **(b)** Artifacts **(c)** Corrected

Figure 9.5.: *(a) shows the resulting Λ with a threshold of $\varepsilon = 0.05$, (b) and (c) show a computed phase with artifacts and after the proposed correction.*

9.5. Evaluation

In order to evaluate the behavior of the procedure, the median pixel error of the calculated phases to the ground truth after several levels are given in Table 9.1, for the proposed approach with combined patterns as well as for the separate approach. As expected, the error of the combined phases in each level is slightly higher than the procedure with separately computed horizontal and vertical phases. However, less recordings were necessary. Considering the accuracy in relation to the image captures used, even with a two-stage procedure and the 12 shots usually required for this, the combined procedure can take another level and double the accuracy of the calculated phases (see Figure 9.6). The plots next to Table 9.6 visualize this. Figure 9.7 shows the final phases, computed by the proposed approach, in comparison to the ground truth. It should be noted that the gamma correction of the devices used (especially the projector) strongly influences the quality of later reconstructions,

Level	1	2	3	4
Combined Patterns				
Median Error of Φ_H	0.0514	0.0149	0.0066	0.0028
Median Error of Φ_V	0.0587	0.0390	0.0124	0.0055
Captured Images	6	9	12	15
Separate Patterns				
Median Error of Φ_H	0.0514	0.0110	0.0052	0.0019
Median Error of Φ_V	0.0587	0.0278	0.0093	0.0034
Captured Images	6	12	18	24

Table 9.1.: *Median errors for different levels of the proposed method with combined patterns and separate patterns, applied to the example data.*

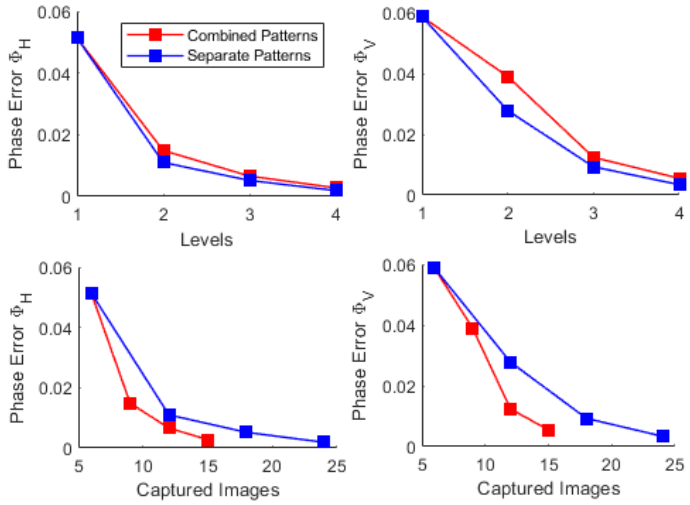
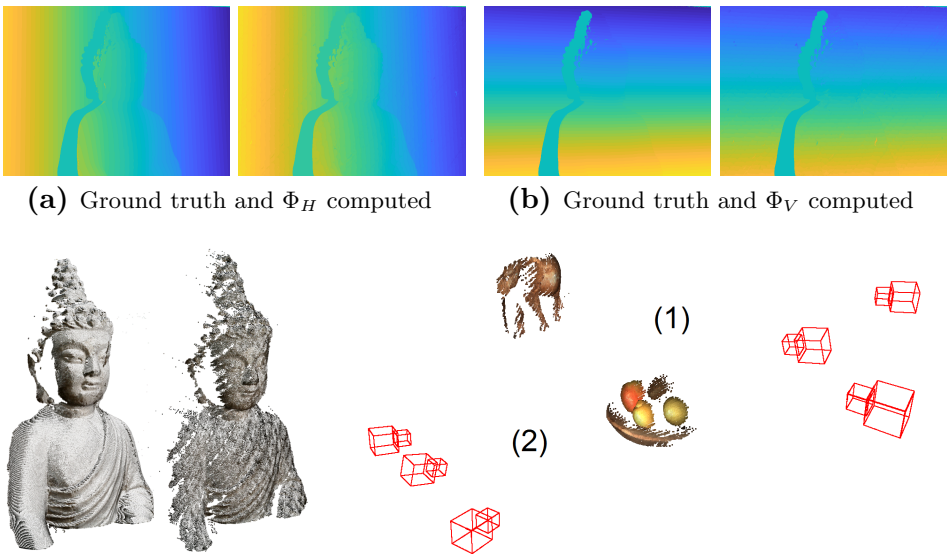


Figure 9.6.: *Behavior of the median phase error with respect to the images that have to be captured.*

since it violates the assumed linearity condition from Equation (9.16). In the shown tests, inverse gamma correction has been applied to the projected patterns with a roughly determined gamma value in order to compensate for this. Therefore real results can still be demonstrated. Nevertheless, the use of industrial projectors without gamma correction or precise gamma calibration of the consumer device used, would significantly improve the quality of the reconstructions. Finally, the calibration results of two other setups and scenes, directly computed from the received point correspondences are visualized in Figure 9.7 (d).

9.6. Conclusions

A new method has been introduced, which allows to perform sinusoidal structured light encoding in horizontal and vertical directions, at the same time. Thereby, the recording time is effectively halved. This procedure especially



(a) Ground truth and Φ_H computed **(b)** Ground truth and Φ_V computed

(c) Reconstructed point clouds **(d)** Auto-calibrations received from the correspondences

Figure 9.7.: *Ground truth phases of the exemplary scene (a, b, left) in comparison to the recovered phases (a, b, right). The triangulated point clouds from ground truth (left) and simultaneously recovered phase (right) are shown in (c). Auto-calibrations from point correspondences of two additional scenes are visualized in (d).*

allows to auto-calibrate arbitrary setups directly from the achieved point correspondences. Extensive mathematical investigations were carried out, which yield new findings in the field of applied harmonic addition theorem. Overall, a method was developed, which can determine the horizontal and vertical phase values from the combined captured patterns pixel-wise, making it scene-independent and therefore applicable to a wide variety of scenarios. The applicability to real scenes besides artificial ones was demonstrated as well.

Chapter 10

Inverse Texturing for Challenging Surfaces

Contents

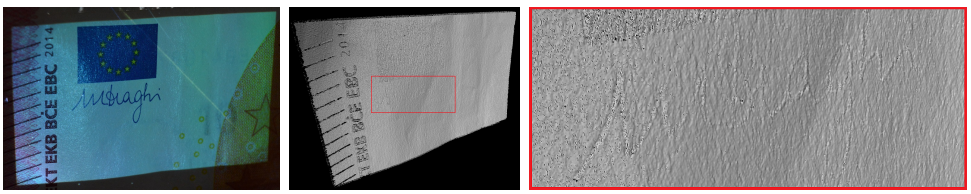
10.1. Introduction

Due to the accuracy and density of the reconstructions obtained, the structured light approach, whenever applicable, is often the method of choice for industrial applications. Nevertheless, it is an active approach which, depending on material properties or coloration, can lead to problems and fail in certain situations. There is a number of disadvantages that should by no means be neglected. The basis of the process is the visibility of the projections on the object's surface. In this context, transparent, mirroring and specular scenes should be mentioned above all. Even objects whose textures contain both highly absorbent and highly reflective areas can cause problems. In many cases such scenes lead to inaccuracies or even to a complete failure of the method.
This chapter, will attempt to eliminate or at least reduce some of these prob-

lems in a way that can be easily adopted to existing scanning setups. A method based on the sinusoidal structured light approach is presented, that significantly reduces the influence of the color of a scanned object. It improves the results obtained by repeated application in terms of accuracy, robustness and general applicability. Especially in high-precision reconstruction of small structures or high-contrast colored and specular objects, the technique shows its greatest potential. The advanced method requires neither pre-calibrated cameras or projectors nor information about the equipment. It is easy to implement and can be applied to any existing scanning setup. Therewith, the application area of structured light reconstruction is increased without additional hardware requirements.
As a basis, phase shifted sinusoidal fringe patterns are used which sum up to zero. Therefore, the encoding method is in theory invariant to the object's texturing. The following reflection properties are unfortunately neglected in this idealized model:

- Different materials cause different light reflection.
- Different colors reflect the light in different ways.

Therefore, in some cases the appearance of the projected fringe patterns can change into the imperceptible. Most reconstructions of objects made of standard material are not very strongly affected and time-consuming procedures, to always treat this behavior, would certainly be overdone. Nevertheless, this effect is clearly noticeable in high-precision reconstructions and it is worth taking a closer look. To illustrate this, Figure 10.1 shows a captured area of a 10 Euro note and a point cloud thereof, reconstructed with structured light. The enlargement of the point cloud clearly shows the erroneous depth, caused exclusively by the texturing of the object. As will be shown, a combination of the projection patterns with an inverse texture significantly reduces this effect.



(a) Texture Image **(b)** Reconstructed point cloud (left) and enlarged red region (right)

Figure 10.1.: *Recorded section of a 10 Euro note (left) and corresponding point cloud, reconstructed by structured light (middle). The enlarged area (right) shows errors in the estimated depth, solely caused by the object's coloration.*

10.2. Related Work

Modern phase-shifted structured light systems with digital devices were introduced by Zhang and Huang in [184]. In order to shorten the acquisition time, Zhang *et al.* [181] and Zhang and Huang [185] introduced methods that use color-coded patterns. Sansoni and Redaelli [143] and later Yang *et al.* [174] introduced single shot structured light techniques, where the multiple shifted fringe images were coded by carrier waves and combined to one pattern. Also Donlic *et al.* [29] and Petkovic *et al.* [130] presented single-shot structured light approaches, based on de Bruijn color sequences. All methods to shorten the acquisition time reduced the quality of reconstructions significantly. In contrast Zhang and Yau [186] presented a setup with two cameras to significantly increase the scan quality. When using several cameras (ideally the same camera model) there are many advantages with regard to calibration, gamma correction of the recorded scenes and the resulting quality of the reconstructions. Although, the approach based on this technique has become popular in many areas, there are several scenarios, where this approach is not applicable. Extensive research has been carried out to improve the applicability to general situations. In order to cope with strong ambient lighting such as sunlight, Gupta *et al.* introduced in [58] a possibility of compensation by sequentially allocating a given energy budget to several sections. Nayar *et al.* [121] and later O'Toole *et al.* [124], [125] presented ways to split direct and indirect light paths, enabling the reconstruction of mirroring, reflecting and light emitting objects and even scanning through dust. Unfortunately, this requires expensive hardware, the process is error-prone and requires high-precision calibration of camera pixels to a DLP panel, making it difficult to use for practical applications. Also Weinmann *et al.* [171] worked on increasing the range of application of structured light but using additional devices.

10.3. Inverse Texture

First of all, the influence of a projection to the captured image of a scene is investigated. After that, a procedure is presented that modulates the projection in a way that as many points as possible in the scene have an equivalent influence to the captured image. This procedure equalizes the appearance of an object iteratively and converges after only a few steps. The projector-camera correspondences are generated using the sinusoidal structured light approach and are reliably cleaned of erroneous phases with the help of an introduced simple masking method.
Let I be a captured image of a scene that was illuminated by a projection T. Schreiber and Bruning [111] described the physical influence of T on image I. Accordingly, the captured image can be approximated as the composition

$$I = I' + I'' \odot T, \tag{10.1}$$

of the ambient intensity I' and the scene intensity I'', modulated by the projected texture T. Thereby, $\odot$ denotes the element-wise multiplication operator.

In order to minimize the influence of the coloring of an object to its image, projection texture T has to be estimated so that it balances the object colors as much as possible. Therefore, the following minimization problem has to be solved:

$$\underset{T,\bar{I}}{\operatorname{argmin}} \sum_{ij} (I_{ij} - \bar{I})^2 = \underset{T,\bar{I}}{\operatorname{argmin}} \sum_{ij} (I'_{ij} + I''_{ij} T_{ij} - \bar{I})^2 \tag{10.2}$$

While I_{ij} denotes the pixels of image I and $\bar{I}$ the optimal common color value of the equalized pixels. Projecting T on the scene approximates an uncolored grayish scene as visualized in Figure 10.2.
Theoretically, an optimal texture T as the solution of problem (10.2) can be calculated explicitly from at least two captured images. However, this is not recommended from a practical point of view. This is due to the following phenomena, which cannot be considered in the underlying model (10.1):

- Color cross-talk between the different color channels of projector and camera.
- Scattered light from one pixel to others are not considered, since it only takes into account direct pixel-to-pixel correspondences.
- Specular or absorbing materials, leading to clipped values in the captured image, due to a limited dynamic range.

10.3.1. Iterative Color Equalization

To solve Problem (10.2), an iterative method is proposed that is robust against the irregularities listed above. Alternated updates of the projected texture T and the equalized target value $\bar{I}$ are combined with a logarithmic search that uses the limited range of projector pixel values (8-bit in [0,255]). In this way, a stable convergence of the process is achieved after a few iterations.

Logarithmic Search In the following, it is assumed that direct correspondences between camera and projector pixels are given by a mapping $\mathcal{P} : (x_I, y_I) \to (x_P, y_P)$ that assigns a corresponding projector pixel (x_P, y_P) to each image pixel (x_I, y_I). Therewith, any projection pixel T_{ij} in the captured scene is known to be located at $\mathcal{P}(T_{ij})$ in the projector input image. Since these projector colors are usually limited to 8-bit values in three color channels, a discrete search range for texture values of T is given, that can be effectively used by a logarithmic search. Moreover, projecting light is a monotonous procedure, therefore increasing values of T_{ij} lead to increasing values of I_{ij}. In order to implement and exploit this knowledge, the values of the projection $\mathcal{P}(T)$ are adjusted via a logarithmic search until the error (10.2) of the resulting image $I = I' + I'' \odot T$ to the equalization value $\bar{I}$ for all pixels is minimal.

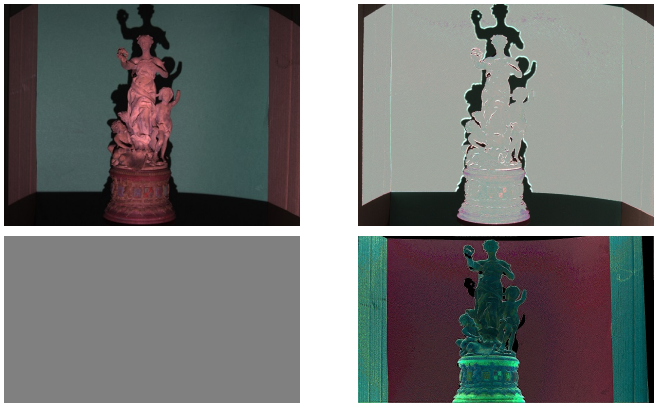


Figure 10.2.: *Textures T projected onto an exemplary scene: Gray projection (bottom, left) and inverse texture calculated with Algorithm 9 (bottom, right). Correspondingly captured scenes without and with color correction (top row).*

Equalization Value Minimizing energy (10.2) with a fixed texture T leads to an optimal equalization value

$$\bar{I} = \frac{1}{MN} \sum_{ij} I_{ij} \tag{10.3}$$

given by the mean of image $I \in \mathbb{R}^{M \times N}$. Alternating updates of the inverse texture T and the equalization value $\bar{I}$ with adjusted increments lead to Algorithm 9, which already converges after 7 iterations in case of standard 8-bit projective devices. Since the texture update is pixel-wise independent, the individual iterations can be implemented efficiently. Figure 10.2 (right) shows a scene that has been equalized in this way. Each iteration was applied separately to the different color channels to intercept the color cross-talk.

Algorithm 3: Iterative Color Equalization

Input: Camera-projector correspondences $\mathcal{P}$.

Initialize projection texture $\mathcal{P}(T_{ij}^{(0)}) = 128 \ \forall ij$.

Project pattern $T_{ij}^{(0)}$ and capture lit scene $I^{(0)}$.

for $z = 1, ..., 7$ **do**

  $\bar{I}^{(z-1)} = \frac{1}{MN} \sum_{ij} I_{ij}^{(z-1)}$

  $$\mathcal{P}(T_{ij}^{(z)}) = \begin{cases} \mathcal{P}(T_{ij}^{(z-1)}) + 2^{7-z}, & I_{ij}^{(z-1)} < \bar{I}^{(z-1)} \\ \mathcal{P}(T_{ij}^{(z-1)}) - 2^{7-z}, & I_{ij}^{(z-1)} > \bar{I}^{(z-1)} \\ \mathcal{P}(T_{ij}), & \text{else} \end{cases}$$

  Project pattern $T^{(z)}$ and capture resulting scene $I^{(z)}$.

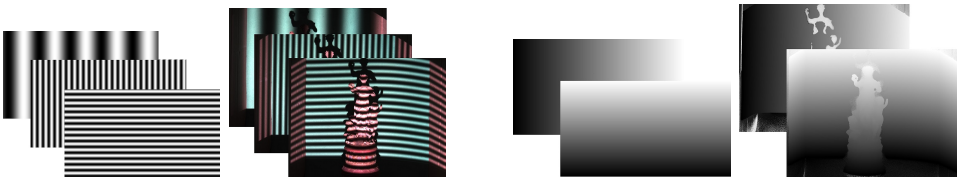
end

Output: Inverse texture T.

10.3.2. Camera-Projector Correspondences

In order to apply Algorithm 9, a reliable mapping $\mathcal{P}(\cdot)$ as described in Section 10.3.1 is required. A recommended approach for determining close point correspondences between projector image and camera image is the structured light approach introduced in [184]. The projection of phase-shifted sine waves (Figure 10.3 (a), allows the calculation of phase images encoding the scene through the projection (Figure 10.3 (b)). This usually requires phase unwrapping methods like [14] or [8] or the pixel-wise method presented in Chapter 9 to use generated wrapped phases to improve the surface encoding. The phase information can be used to determine point correspondences of projector and camera pixels as described in Chapter 4. Errors in the underlying phase information can be caused by

- Overexposed or underexposed areas in the scene where the projected fringes are not visible.
- Regions in the scene that are visible to the camera but not to the projector.
- Shadows caused by the illuminated object.



(a) Fringe patterns and images **(b)** Horizontal and vertical phase images

Figure 10.3.: *Examples of sinusoidal fringe patterns (a, left) and thus illuminated scenes captured by a camera (a, right). Horizontal and vertical phases of the projector (a, left) and the camera (a, right) calculated from the fringe images.*

Masking Erroneous Phase Values For the success of the proposed color equalization method it is a prerequisite to have an accurate mapping $\mathcal{P}$ available. No false correspondences should be used that would significantly falsify the result. For this purpose, incorrect phase information should be masked out beforehand. Defective phase regions, which are calculated using standard phase shift approaches [111], are usually much more noisy than correctly coded ones. Therefore, gradient based filters are typically used to mask out erroneous regions. However, these approaches are not sufficiently accurate for the presented application. Due to the gradient dependency, edges are falsely masked out, which runs counter to the later goal of reconstructing highly accurate small structures. In order to create an appropriate masking, a simple method is presented that provides much more accurate results.

Given is the basic property of phase-shifted sinusoidal patterns:

$$\frac{1}{N}\sum_{n=1}^{N} I_n^H = \frac{1}{M}\sum_{m=1}^{M} I_m^V = I' + 0.5I''. \tag{10.4}$$

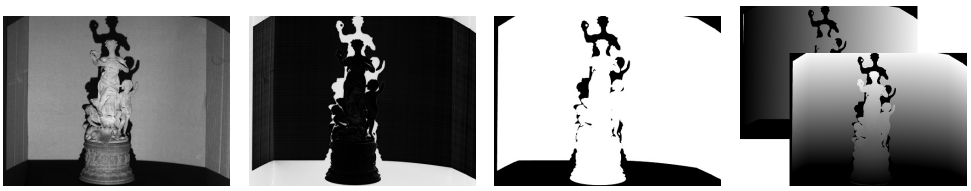
The sum of the phase-shifted sine waves results in zero, which neglects their influence. This means that the applied illumination of the scene in sum is equivalent to a uniform grey projection. Let I_n^H and I_n^V denote the captured scenes, illuminated by respective sine patterns P_n^H and P_m^V as defined in Formula 9.1. If the scene was captured without illumination, I' is already given, so I'' can be estimated by

$$I'' = \frac{1}{N}\sum_{n=1}^{N} I_n^H + \frac{1}{M}\sum_{m=1}^{M} I_m^V - 2I'. \tag{10.5}$$

Finally, an error E of the horizontal and vertical phase values Φ_H and Φ_V to the captured images I_n^H and I_m^V can be calculated by

$$\begin{aligned} E = &\sum_{n=1}^{N}\left(\cos\left(\Phi_H F_H + \frac{2\pi(n-1)}{N}\right) - \frac{I_n^H - I'}{I''}\right)^2 \\ &+ \sum_{m=1}^{M}\left(\cos\left(\Phi_V F_V + \frac{2\pi(m-1)}{M}\right) - \frac{I_m^V - I'}{I''}\right)^2 . \end{aligned} \tag{10.6}$$

Figure 10.4 (a) shows an example of a texture computed from image means (10.4) and (b) the respective error image E from (10.6). This error reliably indicates the quality of the phase values in relation to all captured images of the scene. Since erroneous phase values produce much higher errors than correct ones, a high-quality mask can be generated by applying *k-Means Clustering* to error image E. Note that bi-clustering can be efficiently implemented in $\mathcal{O}(MNlog(MN))$. Moreover, bi-clustering on scalar values can be solved with a guarantee of a global solution. Figure 10.4 (c) shows the final mask as a result of *k-Means Clustering* applied with two clusters. Figure 10.4 (d) shows the final masked phases.



(a) Superposed fringe images **(b)** Error image **(c)** Computed mask **(d)** Masked phase images

Figure 10.4.: *Texture image calculated from averaging captures (10.4) (a) and error image calculated from (10.6) (b). Projection mask clustered from error image (c) and correspondingly masked phase images (d).*

10.4. Inverse Texturing Structured Light (ITSL)

In order to neglect color influences on the geometry estimation by the structured light approach, the inverse texture calculated by Algorithm 9 is combined with the fringe images P_n^H and P_m^V. Instead of the normal patterns the *Inverse Texturing Structured Light Patterns (ITSLP)* are projected:

$$T_n^H = P_n^H \odot \mathcal{P}(\tilde{T}), \quad n = 1, ..., N, \quad T_m^V = P_m^V \odot \mathcal{P}(\tilde{T}), \quad m = 1, ..., M \tag{10.7}$$

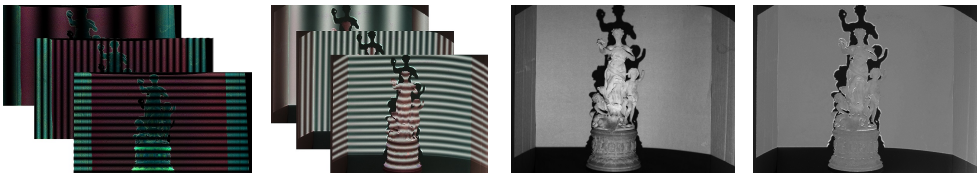
Values of the inverse texture close to zero are lifted to avoid that no fringes are projected in these regions after multiplication:

$$\tilde{T}_{ij} = \begin{cases} T_{ij} & , \text{ if } T_{ij} > 0.05 \cdot \max(T) \\ 0.05 \cdot \max(T) & , \text{ else} \end{cases} \tag{10.8}$$

In the process, masked areas are also coded after several iterations of the approach.
An important feature of ITSLP is, that they fulfill the basic property for fringes of a sinusoidal structured light system, as mentioned in (10.4). In the new case for every scene the following holds true:

$$\frac{1}{N}\sum_{n=1}^{N} I_n^H = I' + \frac{1}{N}\sum_{n=1}^{N} \mathcal{P}^{-1}(P_n^H) \odot \tilde{T} \odot I'' = I' + 0.5(\tilde{T} \odot I'') \approx I' + 0.5(T \odot I''). \tag{10.9}$$

This is equivalent to usual sinusoidal structured light patterns being projected onto a greyish scene with little color influence. Figure 10.5 (a, right) shows ITSLP, computed by (10.8) and the patterns projected onto the scene (a, left). For further visualization Figure 10.5 (right) shows the scenes after averaging (10.4) the standard structured light patterns (left) and ITSLP (right).
Each iteration of ITSLP increases the quality of the reconstructions. Regions with incorrect phase information of the first iteration can be corrected by multiple iterations of *ITSL*.



(a) ITSL Patterns (left) projected to scene (right)

(b) Mean images computed from normal SL patterns (left) and ITSL patterns (right)

Figure 10.5.: *Examples of inverse texturing fringe images projected by the projector (top left) and illuminated scene (top right). Textures calculated by (10.5) from the fringe images (bottom).*

10.5. Evaluation

In order to evaluate the usefulness of the method presented, some quantitative and qualitative tests are carried out. First the performance of the color equalization from Algorithm 9 is examined. In particular, the behavior after several iterations of ITSL is investigated. Subsequently, the advantages and the important practical benefits of ITSLP to structured light reconstruction is shown. In several scenarios, in which the standard structured light approach usually fails, the benefits, which arise from the new, improved approach, become clear.

10.5.1. Inverse Projection Texture

Figure 10.6 (left) shows a captured image of a standardized *X-Rite ColorChecker* commonly used for color calibration. It consists of 24 calibrated colors that well cover the entire visual color spectrum. Figure 10.6 (middle) shows the captured checkerboard after applying Algorithm 9. The method applied to this object demonstrates its behavior in case of very large color differences in an object's texture. It is clearly visible that some color patches (dark red, yellow) cannot be equalized completely. The reason for this is that light can only be projected but not removed. If the red, green or blue component is already in an area above the mean value of the equalization, it cannot be reduced by any projection. To be exact, the required value to be projected does not lie in the gamut generated by the projector. Nevertheless, equalization results of this quality lead to a significant improvement in the reflective properties of an object. Moreover, since grayscale images are sufficient for reconstruction, monochrome cameras can also be used. Errors that may occur during color equalization due to inconsistent gamuts are negligible in this case. Multiple iterations of ITSL, further improve the quality of equalization. Figure 10.6 (right) shows the behavior of the Root-Mean-Squared-Error (RMSE), referred to the mean value (10.3), for an iterative application. While after the first iteration the RMSE decreases by more than 90%, the following iterations reduce this error only slightly in this case.
However, these minor changes can lead to a dramatic improvement of reconstructions. In particular, the reflection behavior at edges of strongly contrasting areas can be significantly improved after a few iterations, due to lower

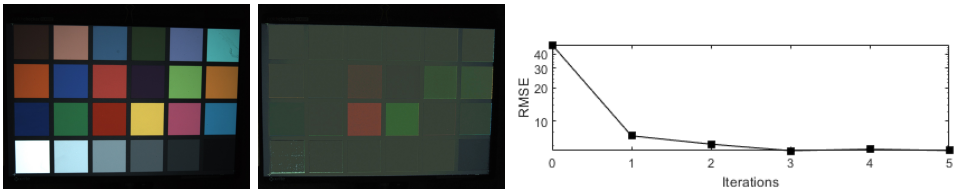


Figure 10.6.: *Captured image of a X-Rite ColorChecker before (top left) and after one iteration of color equalization with Algorithm 9 (top right). RMSE of color equalization for several iterations of ITSLP (bottom).*

radiation. For visual evaluation, Figure 10.7 shows several iterations of the method applied to a figurine of a zebra. The coloration of this object contains maximally strong edges in the transitions from black to white areas. Due to better approximated phase values, every iteration improves the equalization quality. Note that limited projector brightness and stray light between pixels are the reasons why it is impossible to achieve complete equalization in an extreme scenario like the one shown in Figure 10.7. But within the scope of the possibilities, the result obtained here is by far sufficient to yield significant improvements in reconstruction, as will be demonstrated later on.

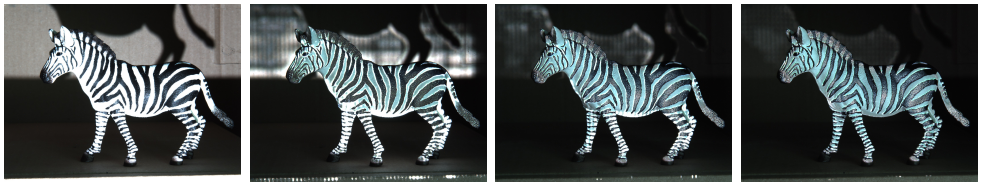


Figure 10.7.: *Captured scene of a zebra to visualize the improvements of several iterations of ITSL to color equalization. Normal image (left) and equalized captures after one, two and three iterations (second to right).*

10.5.2. Inverse Texturing Structured Light

In order to demonstrate the advantages of the presented method in the context of 3D reconstruction, it will be applied to exemplary scenarios in the following. Quantitative and qualitative improvements in the important case of high-precision reconstruction of very small structures will be shown. After that it will be applied to objects with high-contrast staining, such as the one in Figure 10.7, and the behavior after several iterations of the method will be qualitatively investigated. Finally, the chances arising from the method in context of specular and reflecting objects are shown by an exemplary reconstruction of a shiny metal sphere.

High-Precision Reconstruction

Different colors of an object's texture reflect different wavelengths of light. These specific properties, depending on the object coloration, cause projected patterns in structured light applications to be reflected in slightly different ways. Therefore, depending on the coloration of an object, slightly different depth values are estimated. Usually, this effect is very small, compared to the geometry of an object, and can therefore be neglected. However, in high-precision 3D reconstructions, as for example encountered in quality control setups, this problem has considerable effects and significantly affects the results.
In order to evaluate the usefulness of the procedure in relation to this problem, it is applied to a flat checkerboard with patches of different colors, as shown in Figure 10.8. The checkerboard is absolutely flat and any differences in the

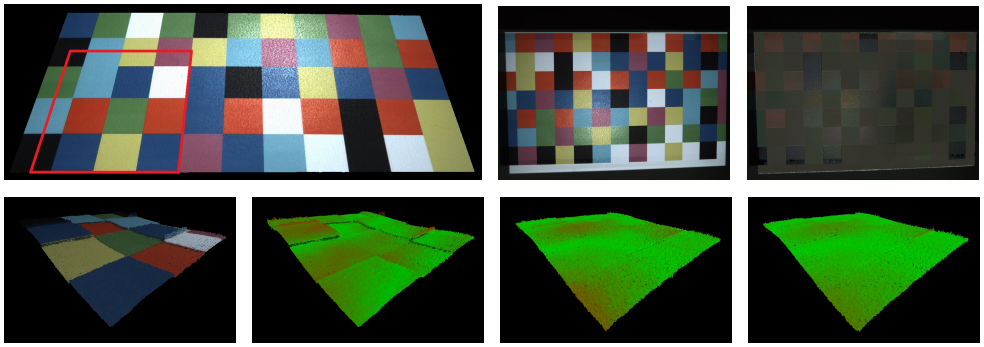


Figure 10.8.: *Flat colored checkerboard: Initial reconstructed point cloud (top left), normal texture (top middle) and equalized texture after one iteration of ITSL (top right). Enlarged marked area of point cloud (bottom left) and point clouds before and after two iterations of the proposed approach (bottom second left to right).*

depth of the reconstruction are errors due to the different colors of the patches. Figure 10.8 (top left) shows the reconstructed point cloud resulting from the standard structured light approach. For further visualization, Figure 10.8 (top center, right) shows the captured scene before and after color equalization. To demonstrate the problem more clearly, Figure 10.8 (bottom, left) shows an enlarged version of the marked area. Figure 10.8 (bottom, second left) shows the point cloud of the enlarged region without texture information, but colored by the Euclidean error with respect to the flat ground truth. Finally, Figure 10.8 (bottom, second right and right) shows the reconstructed regions after one and two iterations of ITSL. To make a qualitative evaluation possible, the depth value of the checkerboards is enhanced by a factor of 3 for visualization. This clearly shows the improvements of the method presented.
Besides the qualitative evaluation, the benefits are demonstrated by a quantitative error measurement. Figure 10.9 shows the behavior of the average depth error of the reconstructed points with respect to the ground truth. Multiple iterations improve the quantitative error continuously. Nevertheless, since the improvements are in the range of μm, one should decide whether the improvement of accuracy of the reconstruction justifies higher additional expenses in the specific case.

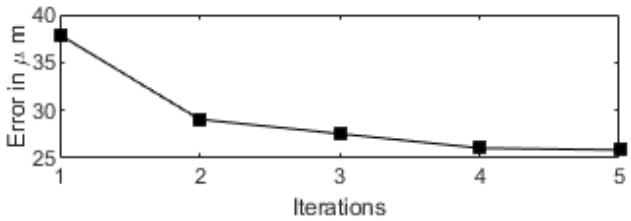


Figure 10.9.: *Average Euclidean depth error in μm of reconstructed point cloud for several iterations of ITSL.*

High-Contrast Colored Objects

Another important field of application of the method is the reconstruction of objects with extremely unfavourable coloring. The statue of a zebra from Figure 10.7 is treated as an example. Due to the very bright and very dark areas there is no camera setting that allows a complete encoding of the surface with structured light. Figure 10.10 (left) shows the result of standard structured light with a rather short exposure time. The white areas of the zebra are well reconstructed, while the black areas are underexposed and not encoded by the patterns. Conversely, Figure 10.10 (middle) shows the reconstructions at a higher exposure time, which allows the reconstruction of the black areas, but overexposes the white regions. Finally, Figure 10.10 (right) shows the result of ITSL. Already one iteration can solve the problem caused by the color contrasts and allows the reconstruction of the entire surface. Further iterations improve the quality slightly, but they should again be weighed according to the benefit and the recording time that needs to be spent.

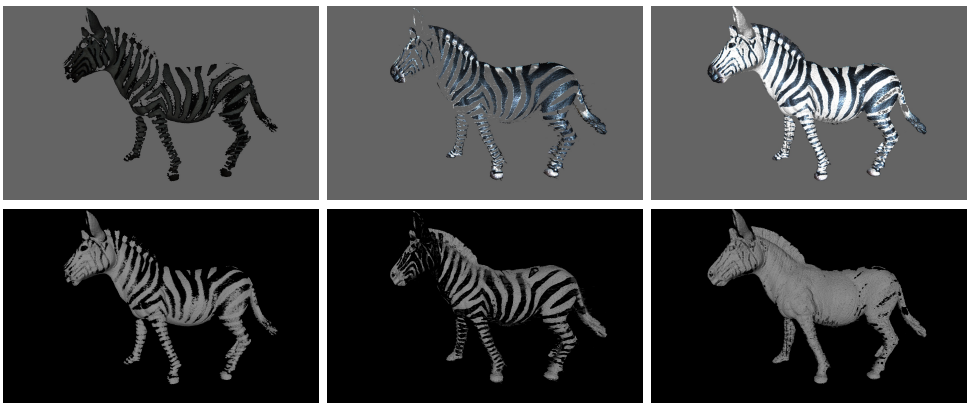


Figure 10.10.: *Reconstructed point clouds of a zebra statue. Top row and bottom row show the same point clouds with and without color information. Left and middle columns show results of standard structured light in case of over- and underexposed images. Right shows ITSL handling the proposed problems.*

Specular Objects

As a last field of application of the method, its benefits in structured light reconstruction of specular objects is demonstrated. The method is applied to a specular metal sphere that strongly reflects the light emitted by the projector. The resulting highlighted areas are overexposed and cannot be encoded by the projected patterns. This effect cannot be completely avoided, but it can be noticeable reduced. Figure 10.11 shows the reconstructed point clouds with standard structured light (left) and after one and two iterations of ITSL (center and right).
The highlighted points do not only depend on the projector's position relative to the object, but also on the camera perspective. To illustrate that

the overexposed point is different for different camera positions. The point clouds in Figure 10.11 are triangulated from a pair of cameras instead of a camera-projector pair. Therefore, there are two independent faulty holes in the reconstructed point clouds from Figure 10.11. Finally, it should be pointed out that this property means, that the defective area of the phase of one camera is most likely correctly encoded in the other camera. This could be used to improve reconstructions and to make them invariant to reflective objects in multiple camera structured light setups, as they are typical used for practical applications.

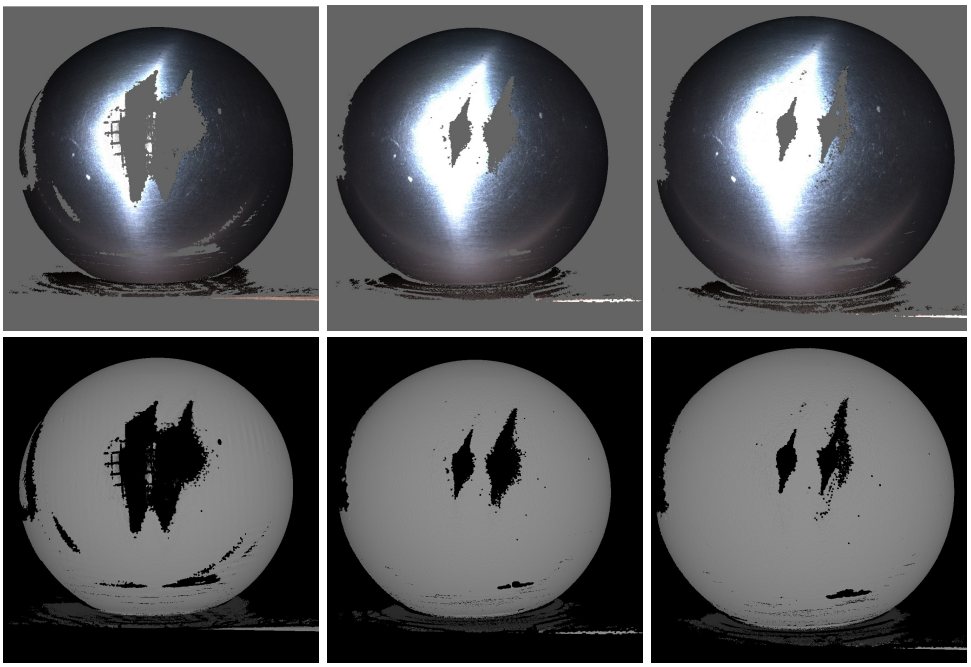


Figure 10.11.: *Triangulated point clouds of a specular sphere, captured from two views. Standard structured light (left) and results of ITSL after one and two iterations (middle and right). While the top row shows the textured point clouds, the bottom row visualizes the sole geometry.*

10.6. Conclusion

In this chapter, a method has been presented that expands the practical scope of structured light reconstruction. Typical scenarios, in which the standard approach usually fails, are now treatable. Unfavorably colored and reflective objects cause no or significantly fewer problems. In the area of high-precision reconstruction, a significant leap in accuracy is achieved. However, the iterative character of the method also increases the recording time. Several iterations increase the accuracy, but should be weighed against the additional time required. Therefore, the approach is designed so that it can be easily built on existing setup and applied or omitted as needed.

Chapter 11

Conclusions

Contents

11.1. Goals of the Thesis

The goal of this work was to develop a complete, fully automatic 3D reconstruction pipeline that works for a large number of different objects. This was delivered using an active structured light approach based on sinusoidal patterns that enables highly accurate reconstructions without dependence on object features. The pipeline should also be suitable and applicable for different applications. For this purpose, a flexible auto-calibration procedure was developed that works independently of the selected devices and enables stable calibration of all devices, including the projector used. It does not depend on pre-calibration (especially of the active component) and thus allows any adaptation of the equipment to the targets to be reconstructed. This approach is thus cost-efficient, flexible, and at the same time user-friendly, i.e. fully automatic.

11.2. Summary of Thesis Achievements

In terms of the resulting quality and applicability of structured light reconstruction, contributions have been made in several ways.

Accuracy Using the consistent sub-pixel matching for two-dimensional structured light encodings, introduced in Chapter 4, a reduction of the back-projection error by about 28% was achieved, while the matching cost still remained linear. In addition, outliers were excluded without additional filters. In the calibration of the used devices, the combination of the methods presented in Chapter 5 allows to reduce the resulting back-projection errors even further by up to 30%. In the case of high precision reconstructions, in Chapter 10 a method has been presented that iteratively reduces erroneous reconstructions of very small structures caused by color influences. The examined errors decreased by 32% from 38μm to 26μm in the examined object after three iterations. When fully automatically aligning full turns of 3D reconstructions from Chapter 7, a reduction of the alignment errors by an average of 15% was achieved without more iterations than the standard method.

Stability and Robustness In terms of stability, a number of results was obtained, especially in the area of calibration in Chapter 5. The presented method allows minimizing the trifocal error for estimating the epipolar geometry even in the presence of strong noise and outliers. Intrinsic auto-calibration, which estimates camera parameters directly from fundamental matrices, converges under minimization with the proposed energy with an extraordinarily increased convergence region, allowing the approach to be used for applications where success was previously critical. In particular, the approach enables stable calibration of a projector from point correspondences and to cleanly estimate its principal point. The pre-alignment of Chapter 6 allows to stably register ancient partial scans, that are usually differently shaded, as they are illuminated by projectors from different relative positions. In the alignment of Chapter 7, it was also possible to experience much smoother convergence behavior by introducing symmetric updates of the partial reconstructions. It proceeds much more smoothly and allows the use of automatic stopping criteria, which otherwise often fail due to alternations of the alignment error in other ICP approaches.

Speed With respect to the extremely time-consuming acquisition procedure of two-dimensional scene encoding with sinusoidal structured light, on which the presented approach is based, is was possible to present a method in Chapter 9 that almost halves the required acquisition images by combining horizontal and vertical patterns. Unfortunately, the obtained correspondences suffer especially from influences of gamma correction of the used devices. The obtained point correspondences are significantly more noisy, but still, considering the median errors, significantly lower than encodings with separate patterns, which could have made fewer refinements in the same recording time. The correspondences obtained in this way are of particular interest for calibration where outliers are not significant, due to subsequently used stable RANSAC approaches. Especially for the calibration of devices like SLR cameras, which have a slow recording time, this approach to calibration is of interest.

Usability and Flexibility Last but not least, the overall method has gained some applicability especially in terms of flexibility and usability without user interaction. This could be achieved in particular through the auto-calibration, based on self-generated correspondences in the scene, and the opportunity to use arbitrary cameras and projectors. In addition, the application area of the active structured light approach has been extended to some problematic surfaces, as shown in Chapter 10. In this way, even extremely unfavorable high-contrast textures and specular surfaces can be examined and their negative influence on the resulting reconstructions can at least be reduced.

11.3. Future Work

Possible future work to expand the pipeline would be on applications that combine and benefit from joint normal vector calculations and texturing. In situations such as a structured light scanner, where turntables are used to rotate the object, the illumination and thus the shading of the object often changes due to the rotation. The different shadings of the object for different views can lead to seams in the model's texture afterwards. At the same time the direction of illumination by the projector is known, since the active device has also been calibrated with high precision. A combination of geometric approaches to normal vector estimation and a Lambertian assumption of the resulting shading, as exploited in *Shape-from-Shading* methods [182] may allow to estimate better normal vectors and at the same time a base-color of the texture without disturbing influence of the active illumination.
Another future improvement of the pipeline would be to co-estimate material properties, such as *Bidirectional Reflectance Distribution Functions* [114] for the reconstructed surfaces within the 3D scanner. These allow to render material properties photorealistically. The renderings obtained from the 3D reconstruction pipeline would thus be even more realistically representable.

Bibliography

[1] URL: `https://www.implantate.com` (visited on 09/15/2021).

[2] URL: `https://unity.com` (visited on 12/02/2021).

[3] Sameer Agarwal et al. "Bundle adjustment in the large". In: *European conference on computer vision*. Springer. 2010, pp. 29–42.

[4] Devrim Akca. *Generalized procrustes analysis and its applications in photogrammetry*. Tech. rep. ETH Zurich, 2003.

[5] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. "KAZE features". In: *European conference on computer vision*. Springer. 2012, pp. 214–227.

[6] Abd Albasset Almamou et al. "Quality control of constructed models using 3d point cloud". In: (2015).

[7] Abdulrahman S Alturki and John S Loomis. "X-corner detection for camera calibration using saddle points". In: *International Journal of Computer and Information Engineering* 10.4 (2016), pp. 676–681.

[8] Yatong An, Jae-Sang Hyun, and Song Zhang. "Pixel-wise absolute phase unwrapping using geometric constraints of structured light system". In: *Optics express* 24.16 (2016), pp. 18445–18459.

[9] Aleksandr Aravkin et al. "Student's t robust bundle adjustment algorithm". In: *2012 19th IEEE International Conference on Image Processing*. IEEE. 2012, pp. 1757–1760.

[10] K Somani Arun, Thomas S Huang, and Steven D Blostein. "Least-squares fitting of two 3-D point sets". In: *IEEE Transactions on pattern analysis and machine intelligence* 5 (1987), pp. 698–700.

[11] Tali Basha, Yael Moses, and Nahum Kiryati. "Multi-view scene flow estimation: A view centered variational approach". In: *International journal of computer vision* 101.1 (2013), pp. 6–21.

[12] Stefania Bellavia, Serge Gratton, and Elisa Riccietti. "A Levenberg–Marquardt method for large nonlinear least-squares problems with dynamic accuracy in functions and gradients". In: *Numerische Mathematik* 140.3 (2018), pp. 791–825.

[13] Paul J Besl and Neil D McKay. "Method for registration of 3-D shapes". In: *Sensor fusion IV: control paradigms and data structures*. Vol. 1611. International Society for Optics and Photonics. 1992, pp. 586–606.

[14] Jos M Bioucas-Dias and Gonalo Valadao. "Phase unwrapping via graph cuts". In: *IEEE Transactions on Image processing* 16.3 (2007), pp. 698–709.

[15] Stan Birchfield and Carlo Tomasi. "A pixel dissimilarity measure that is insensitive to image sampling". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.4 (1998), pp. 401–406.

[16] Michael Bleyer, Christoph Rhemann, and Carsten Rother. "PatchMatch Stereo-Stereo Matching with Slanted Support Windows." In: *Bmvc*. Vol. 11. 2011, pp. 1–11.

[17] Sylvain Bougnoux. "From projective to euclidean space under any practical situation, a criticism of self-calibration". In: *Computer Vision, 1998. Sixth International Conference on*. IEEE. 1998, pp. 790–796.

[18] José Henrique Brito et al. "Radial distortion self-calibration". In: *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE. 2013, pp. 1368–1375.

[19] Duane C Brown. "Decentering distortion of lenses". In: *Photogrammetric Engineering and Remote Sensing* (1966).

[20] Russell A Brown. "Building a balanced kd tree in o (kn log n) time". In: *arXiv preprint arXiv:1410.5420* (2014).

[21] Jan Čech, Jordi Sanchez-Riera, and Radu Horaud. "Scene flow estimation by growing correspondence seeds". In: *CVPR 2011*. IEEE. 2011, pp. 3129–3136.

[22] Yisong Chen et al. "Full camera calibration from a single view of planar scene". In: *International Symposium on Visual Computing*. Springer. 2008, pp. 815–824.

[23] Yu Chen, Yisong Chen, and Guoping Wang. "Bundle Adjustment Revisited". In: *arXiv preprint arXiv:1912.03858* (2019).

[24] Dmitry Chetverikov, Dmitry Stepanov, and Pavel Krsek. "Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm". In: *Image and vision computing* 23.3 (2005), pp. 299–309.

[25] Dmitry Chetverikov et al. "The trimmed iterative closest point algorithm". In: *Object recognition supported by user interaction for service robots*. Vol. 3. IEEE. 2002, pp. 545–548.

[26] Ion Aurel Cristescu. "Approximate solution of nonlinear Poisson equation by finite differences method". In: *Rom. Rep. Phys* 68.2 (2016), pp. 473–485.

[27] Gabriella Csurka et al. "Characterizing the uncertainty of the fundamental matrix". In: *Computer vision and image understanding* 68.1 (1997), pp. 18–36.

[28] Arturo Donate, Xiuwen Liu, and Emmanuel G Collins. "Efficient path-based stereo matching with subpixel accuracy". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41.1 (2010), pp. 183–195.

[29] Matea Donlic, Tomislav Petkovic, and Tomislav Pribanic. "3D surface profilometry using phase shifting of De Bruijn pattern". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 963–971.

[30] Chitra Dorai et al. "Registration and integration of multiple object views for 3D model construction". In: *IEEE Transactions on pattern analysis and machine intelligence* 20.1 (1998), pp. 83–89.

[31] Alexey Dosovitskiy et al. "Flownet: Learning optical flow with convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2758–2766.

[32] Qiuchen Du et al. "Stereo-Matching Network for Structured Light". In: *IEEE Signal Processing Letters* 26.1 (2018), pp. 164–168.

[33] David W Eggert, Adele Lorusso, and Robert B Fisher. "Estimating 3-D rigid body transformations: a comparison of four major algorithms". In: *Machine vision and applications* (1997).

[34] Chaima El Asmi and Sébastien Roy. "Subpixel unsynchronized unstructured light". In: *Manuscript submitted for publication* (2019).

[35] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. "Confidence propagation through cnns for guided sparse depth regression". In: *IEEE transactions on pattern analysis and machine intelligence* 42.10 (2019), pp. 2423–2436.

[36] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. "Propagating confidences through cnns for sparse data regression". In: *arXiv preprint arXiv:1805.11913* (2018).

[37] Chris Engels, Henrik Stewénius, and David Nistér. "Bundle adjustment rules". In: *Photogrammetric computer vision* 2.2006 (2006).

[38] Olivier D Faugeras, Q-T Luong, and Stephen J Maybank. "Camera self-calibration: Theory and experiments". In: *European conference on computer vision*. Springer. 1992, pp. 321–334.

[39] Yutong Feng et al. "Meshnet: Mesh neural network for 3d shape representation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 8279–8286.

[40] Luis Ferraz, Xavier Binefa, and Francesc Moreno-Noguer. "Very fast solution to the PnP problem with algebraic outlier rejection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 501–508.

[41] David Ferstl et al. "aTGV-SF: Dense variational scene flow through projective warping and higher order regularization". In: *2014 2nd International Conference on 3D Vision*. Vol. 1. IEEE. 2014, pp. 285–292.

[42] Torben Fetzer, Gerd Reis, and Didier Stricker. "Fast Projector-Driven Structured Light Matching in Sub-Pixel Accuracy using Bilinear Interpolation Assumption". In: *International Conference on Computer Analysis of Images and Patterns*. Springer. 2021.

[43] Torben Fetzer, Gerd Reis, and Didier Stricker. "INV-Flow2PoseNet: Light-Resistant Rigid Object Pose from Optical Flow of RGB-D Images Using Images, Normals and Vertices". In: *Sensors* 22.22 (2022), p. 8798.

[44] Torben Fetzer, Gerd Reis, and Didier Stricker. "Iterative Color Equalization for Increased Applicability of Structured Light Reconstruction". In: *15th International Conference on Computer Vision Theory and Applications*. 2020.

[45] Torben Fetzer, Gerd Reis, and Didier Stricker. "Joint Global ICP for Improved Automatic Alignment of Full Turn Object Scans". In: *International Conference on Computer Analysis of Images and Patterns*. Springer. 2021.

[46] Torben Fetzer, Gerd Reis, and Didier Stricker. "Robust Auto-Calibration for Practical Scanning Setups from Epipolar and Trifocal Relations". In: *2019 16th International Conference on Machine Vision Applications (MVA)*. IEEE. 2019, pp. 1–6.

[47] Torben Fetzer, Gerd Reis, and Didier Stricker. "Simultaneous Bi-Direct-
ional Structured Light Encoding for Practical Uncalibrated Profilometry". In: *International Conference on Computer Analysis of Images and Patterns*. Springer. 2021.

[48] Torben Fetzer, Gerd Reis, and Didier Stricker. "Stable Intrinsic Auto-Calibration from Fundamental Matrices of Devices with Uncorrelated Camera Parameters". In: *The IEEE Winter Conference on Applications of Computer Vision*. 2020, pp. 221–230.

[49] Andrew W Fitzgibbon. "Simultaneous linear estimation of multiple view geometry and lens distortion". In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Vol. 1. IEEE. 2001, pp. I–I.

[50] Katerina Fragkiadaki, Han Hu, and Jianbo Shi. "Pose from flow and flow from pose". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 2059–2066.

[51] Yanping Fu et al. "Texture mapping for 3d reconstruction with rgb-d sensor". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4645–4653.

[52] Natasha Gelfand et al. "Geometrically stable sampling for the ICP algorithm". In: *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003.* IEEE. 2003, pp. 260–267.

[53] Riccardo Gherardi and Andrea Fusiello. "Practical autocalibration". In: *European Conference on Computer Vision*. Springer. 2010, pp. 790–801.

[54] Silvio Giancola, Matteo Valenti, and Remo Sala. *A survey on 3D cameras: Metrological comparison of time-of-flight, structured-light and active stereoscopy technologies*. Springer, 2018.

[55] Jens-Malte Gottfried, Janis Fehr, and Christoph S Garbe. "Computing range flow from multi-modal kinect data". In: *International symposium on visual computing*. Springer. 2011, pp. 758–767.

[56] Michael Greenspan and Mike Yurick. "Approximate kd tree search for efficient ICP". In: *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings*. IEEE. 2003, pp. 442–448.

[57] C Guan, LG Hassebrook, and DL Lau. "Composite structured light pattern for three-dimensional video". In: *Optics Express* 11.5 (2003), pp. 406–417.

[58] Mohit Gupta, Qi Yin, and Shree K Nayar. "Structured light in sunlight". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 545–552.

[59] Shir Gur and Lior Wolf. "Single image depth estimation trained via depth from defocus cues". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7683–7692.

[60] Rana Hanocka et al. "Meshcnn: a network with an edge". In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), pp. 1–12.

[61] Richard Hartley. "Extraction of focal lengths from the fundamental matrix". In: *Unpublished manuscript* (1993).

[62] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[63] Richard I Hartley. "In defense of the eight-point algorithm". In: *IEEE Transactions on pattern analysis and machine intelligence* 19.6 (1997), pp. 580–593.

[64] Richard I. Hartley. "Kruppa's equations derived from the fundamental matrix". In: *IEEE Transactions on pattern analysis and machine intelligence* 19.2 (1997), pp. 133–135.

[65] Evan Herbst, Xiaofeng Ren, and Dieter Fox. "Rgb-d flow: Dense 3-d motion estimation using color and depth". In: *2013 IEEE international Conference on robotics and automation*. IEEE. 2013, pp. 2276–2282.

[66] Heiko Hirschmuller and Daniel Scharstein. "Evaluation of cost functions for stereo matching". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2007, pp. 1–8.

[67] Stefan Holzer et al. "Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 2684–2689.

[68] Hugues Hoppe et al. "Surface reconstruction from unorganized points". In: *Proceedings of the 19th annual conference on computer graphics and interactive techniques*. 1992, pp. 71–78.

[69] Berthold KP Horn and Brian G Schunck. "Determining optical flow". In: *Artificial intelligence* 17.1-3 (1981), pp. 185–203.

[70] Yingfeng Hu. "Research on a three-dimensional reconstruction method based on the feature matching algorithm of a scale-invariant feature transform". In: *Mathematical and computer modelling* 54.3-4 (2011), pp. 919–923.

[71] Frédéric Huguet and Frédéric Devernay. "A variational method for scene flow estimation from stereo sequences". In: *2007 IEEE 11th International Conference on Computer Vision*. IEEE. 2007, pp. 1–7.

[72] Tak-Wai Hui and Chen Change Loy. "Liteflownet3: Resolving correspondence ambiguity for more accurate optical flow estimation". In: *European Conference on Computer Vision*. Springer. 2020, pp. 169–184.

[73] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. "Liteflownet: A lightweight convolutional neural network for optical flow estimation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8981–8989.

[74] Junhwa Hur and Stefan Roth. "Iterative residual refinement for joint optical flow and occlusion estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5754–5763.

[75] Junhwa Hur and Stefan Roth. "MirrorFlow: Exploiting symmetries in joint optical flow and occlusion estimation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 312–321.

[76] Eddy Ilg et al. "Flownet 2.0: Evolution of optical flow estimation with deep networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2462–2470.

[77] Hossam Isack and Yuri Boykov. "Energy based multi-model fitting & matching for 3D reconstruction". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1146–1153.

[78] Michael Isard and John MacCormick. "Dense motion and disparity estimation via loopy belief propagation". In: *Asian conference on computer vision*. Springer. 2006, pp. 32–41.

[79] Joel Janai et al. "Unsupervised learning of multi-frame optical flow with occlusions". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 690–706.

[80] Changsoo Je, Sang Wook Lee, and Rae-Hong Park. "Color-phase analysis for sinusoidal structured light in rapid range imaging". In: *arXiv preprint arXiv:1509.04115* (2015).

[81] Ziad Jomaa and Charlie Macaskill. "The embedded finite difference method for the Poisson equation in a domain with an irregular boundary and Dirichlet boundary conditions". In: *Journal of Computational Physics* 202.2 (2005), pp. 488–506.

[82] Rico Jonschkowski et al. “What matters in unsupervised optical flow”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 557–572.

[83] Timothée Jost and Heinz Hügli. “Fast ICP algorithms for shape registration”. In: *Joint Pattern Recognition Symposium*. Springer. 2002, pp. 91–99.

[84] Kenichi Kanatani and Yasuyuki Sugaya. “Bundle adjustment for 3-d reconstruction: Implementation and evaluation”. In: *Memoirs of the Faculty of Engineering, Okayama University* 45 (2011), pp. 27–35.

[85] Siddhant Katyan, Shrutimoy Das, and Pawan Kumar. “Two-Grid Preconditioned Solver for Bundle Adjustment”. In: *The IEEE Winter Conference on Applications of Computer Vision*. 2020, pp. 3599–3606.

[86] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. “Poisson surface reconstruction”. In: *Proceedings of the fourth Eurographics symposium on Geometry processing*. Vol. 7. 2006.

[87] Michael Kazhdan and Hugues Hoppe. “Screened poisson surface reconstruction”. In: *ACM Transactions on Graphics (ToG)* 32.3 (2013), pp. 1–13.

[88] Alex Kendall, Matthew Grimes, and Roberto Cipolla. “Posenet: A convolutional network for real-time 6-dof camera relocalization”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2938–2946.

[89] Kurt Konolige and Motilal Agrawal. “FrameSLAM: From bundle adjustment to real-time visual mapping”. In: *IEEE Transactions on Robotics* 24.5 (2008), pp. 1066–1077.

[90] Kurt Konolige and Willow Garage. “Sparse Sparse Bundle Adjustment.” In: *BMVC*. Vol. 10. Citeseer. 2010, pp. 102–1.

[91] Young-tae Kwak, Ji-won Hwang, and Cheol-jung Yoo. “A new damping strategy of Levenberg-Marquardt algorithm for multilayer perceptrons”. In: *Neural Network World* 21.4 (2011), p. 327.

[92] Ricardo Legarda-Saenz and Arturo Espinosa-Romero. “Wavefront reconstruction using multiple directional derivatives and Fourier transform”. In: *Optical Engineering* 50.4 (2011), p. 040501.

[93] Antoine Letouzey, Benjamin Petit, and Edmond Boyer. “Scene flow from depth and color images”. In: *BMVC 2011-British Machine Vision Conference*. BMVA Press. 2011, pp. 46–1.

[94] Francis Li et al. “Simultaneous projector-camera self-calibration for three-dimensional reconstruction and projection mapping”. In: *IEEE Transactions on Computational Imaging* 3.1 (2017), pp. 74–83.

[95] Fu Li et al. “Depth acquisition with the combination of structured light and deep learning stereo matching”. In: *Signal Processing: Image Communication* 75 (2019), pp. 111–117.

[96] Rui Li and Stan Sclaroff. "Multi-scale 3D scene flow from binocular stereo sequences". In: *Computer Vision and Image Understanding* 110.1 (2008), pp. 75–90.

[97] Fayao Liu et al. "Learning depth from single monocular images using deep convolutional neural fields". In: *IEEE transactions on pattern analysis and machine intelligence* 38.10 (2015), pp. 2024–2039.

[98] Kun Liu et al. "Optimized stereo matching in binocular three-dimensional measurement system using structured light". In: *Applied optics* 53.26 (2014), pp. 6083–6090.

[99] Pengpeng Liu et al. "Selflow: Self-supervised learning of optical flow". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4571–4580.

[100] William E Lorensen and Harvey E Cline. "Marching cubes: A high resolution 3D surface construction algorithm". In: *ACM siggraph computer graphics* 21.4 (1987), pp. 163–169.

[101] Manolis Lourakis and Antonis Argyros. *The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm*. Tech. rep. Technical Report 340, Institute of Computer Science-FORTH, Heraklion, Crete ..., 2004.

[102] Manolis IA Lourakis et al. "A brief description of the Levenberg-Marquardt algorithm implemented by levmar". In: *Foundation of Research and Technology* 4.1 (2005), pp. 1–6.

[103] Manolis IA Lourakis and Antonis A Argyros. "SBA: A software package for generic sparse bundle adjustment". In: *ACM Transactions on Mathematical Software (TOMS)* 36.1 (2009), pp. 1–30.

[104] Manolis IA Lourakis and Rachid Deriche. "Camera self-calibration using the Kruppa equations and the SVD of the fundamental matrix: The case of varying intrinsic parameters". In: (2000).

[105] Manolis IA Lourakis and Rachid Deriche. "Camera self-calibration using the singular value decomposition of the fundamental matrix: From point correspondences to 3D measurements". PhD thesis. INRIA, 1999.

[106] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[107] David G Lowe. "Object recognition from local scale-invariant features". In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[108] Quan-Tuan Luong and Olivier D Faugeras. "The fundamental matrix: Theory, algorithms, and stability analysis". In: *International journal of computer vision* 17.1 (1996), pp. 43–75.

[109] Shiwei Ma et al. "Binocular structured light stereo matching approach for dense facial disparity map". In: *Australasian Joint Conference on Artificial Intelligence*. Springer. 2011, pp. 550–559.

[110] Elmar Mair et al. “Adaptive and generic corner detection based on the accelerated segment test”. In: *European conference on Computer vision*. Springer. 2010, pp. 183–196.

[111] Daniel Malacara. *Optical shop testing*. John Wiley & Sons, 2007.

[112] Ezio Malis and Roberto Cipolla. “Camera self-calibration from unknown planar structures enforcing the multiview constraints between collineations”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.9 (2002), pp. 1268–1272.

[113] Takeshi Masuda, Katsuhiko Sakaue, and Naokazu Yokoya. “Registration and integration of multiple range images for 3-D model construction”. In: *Proceedings of 13th international conference on pattern recognition*. Vol. 1. IEEE. 1996, pp. 879–883.

[114] Wojciech Matusik et al. “Efficient isotropic BRDF measurement”. In: (2003).

[115] Simon Meister, Junhwa Hur, and Stefan Roth. “Unflow: Unsupervised learning of optical flow with a bidirectional census loss”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 2018.

[116] Johanna Menningen and Torben Fetzer et al. “Ultrasound tomography and 3D scanning technologies as a tool to constrain the weathering state of objects made of marble”. In: *Monument Future: Decay and Conservation of Stone*. Mitteldeutscher Verlag, 2020.

[117] Parsa Mirdehghan, Wenzheng Chen, and Kiriakos N Kutulakos. “Optimal structured light à la carte”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[118] Niloy J Mitra and An Nguyen. “Estimating surface normals in noisy point cloud data”. In: *Proceedings of the nineteenth annual symposium on Computational geometry*. 2003, pp. 322–328.

[119] Farzin Mokhtarian and Riku Suomela. “Robust image corner detection through curvature scale space”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.12 (1998), pp. 1376–1381.

[120] Jorge J Moré. “The Levenberg-Marquardt algorithm: implementation and theory”. In: *Numerical analysis*. Springer, 1978, pp. 105–116.

[121] Shree K Nayar et al. “Fast separation of direct and global components of a scene using high frequency illumination”. In: *ACM Transactions on Graphics (TOG)*. Vol. 25. 3. ACM. 2006, pp. 935–944.

[122] Pekka Neittaanmäki and Jukka Saranen. “On finite element approximation of the gradient for solution of Poisson equation”. In: *Numerische Mathematik* 37.3 (1981), pp. 333–337.

[123] Andreas Nuchter, Kai Lingemann, and Joachim Hertzberg. “Cached kd tree search for ICP algorithms”. In: *Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM 2007)*. IEEE. 2007, pp. 419–426.

[124] Matthew O'Toole, John Mather, and Kiriakos N Kutulakos. "3d shape and indirect appearance by structured light transport". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 3246–3253.

[125] Matthew O'Toole, Ramesh Raskar, and Kiriakos N Kutulakos. "Primal-dual coding to probe light transport." In: *ACM Trans. Graph.* 31.4 (2012), pp. 39–1.

[126] Nay Oo and Woon-Seng Gan. "On harmonic addition theorem". In: *International Journal of Computer and Communication Engineering* 1.3 (2012), p. 200.

[127] Anton Osokin, Denis Sumin, and Vasily Lomakin. "Os2d: One-stage one-shot object detection by matching anchor features". In: *European Conference on Computer Vision*. Springer. 2020, pp. 635–652.

[128] Jaesik Park et al. "A tensor voting approach for multi-view 3D scene flow estimation and refinement". In: *European Conference on Computer Vision*. Springer. 2012, pp. 288–302.

[129] Artem L Pavlov et al. "AA-ICP: Iterative closest point with Anderson acceleration". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 1–6.

[130] Tomislav Petković, Tomislav Pribanić, and Matea Donlić. "Single-shot dense 3D reconstruction using self-equalizing De Bruijn sequence". In: *IEEE Transactions on Image Processing* 25.11 (2016), pp. 5131–5144.

[131] Nicola A Piga et al. "ROFT: Real-Time Optical Flow-Aided 6D Object Pose and Velocity Tracking". In: *IEEE Robotics and Automation Letters* 7.1 (2021), pp. 159–166.

[132] Daniel Pizarro and Adrien Bartoli. "Global optimization for optimal generalized procrustes analysis". In: *CVPR*. IEEE. 2011.

[133] Marc Pollefeys, Reinhard Koch, and Luc Van Gool. "Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters". In: *International Journal of Computer Vision* 32.1 (1999), pp. 7–25.

[134] Tomislav Pribanic, Nenad Obradovic, and Joaquim Salvi. "Stereo computation combining structured light and passive stereo matching". In: *Optics Communications* 285.6 (2012), pp. 1017–1022.

[135] Kari Pulli. "Multiview registration for large data sets". In: *Second International Conference on 3-D Digital Imaging and Modeling (Cat. No. PR00062)*. IEEE. 1999, pp. 160–168.

[136] Julian Quiroga et al. "Dense semi-rigid scene flow estimation from rgbd images". In: *European Conference on Computer Vision*. Springer. 2014, pp. 567–582.

[137] Ananth Ranganathan. "The levenberg-marquardt algorithm". In: *Tutoral on LM algorithm* 11.1 (2004), pp. 101–110.

[138] Rishav Rishav et al. “DeepLiDARFlow: A Deep Learning Architecture For Scene Flow Estimation Using Monocular Camera and Sparse LiDAR”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 10460–10467.

[139] Leonardo Romero and Cuauhtemoc Gomez. “Correcting radial distortion of cameras with wide angle lens using point correspondences”. In: *Scene Reconstruction Pose Estimation and Tracking*. InTech, 2007.

[140] Szymon Rusinkiewicz and Marc Levoy. “Efficient variants of the ICP algorithm”. In: *Proceedings third international conference on 3-D digital imaging and modeling*. IEEE. 2001.

[141] Sean Ryan Fanello et al. “Hyperdepth: Learning depth from structured light without matching”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 5441–5450.

[142] Joaquim Salvi et al. “A state of the art in structured light patterns for surface profilometry”. In: *Pattern recognition* 43.8 (2010), pp. 2666–2680.

[143] Giovanna Sansoni and Elisa Redaelli. “A 3D vision system based on one-shot projection and phase demodulation for fast profilometry”. In: *Measurement Science and Technology* 16.5 (2005), p. 1109.

[144] Ashutosh Saxena, Sung H Chung, Andrew Y Ng, et al. “Learning depth from single monocular images”. In: *NIPS*. Vol. 18. 2005, pp. 1–8.

[145] Daniel Scharstein and Richard Szeliski. “High-accuracy stereo depth maps using structured light”. In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* Vol. 1. IEEE. 2003, pp. I–I.

[146] M Servin, JC Estrada, and J Antonio Quiroga. “The general theory of phase shifting algorithms”. In: *Optics express* 17.24 (2009), pp. 21867–21881.

[147] Mark Shortis. “Camera calibration techniques for accurate measurement underwater”. In: *3D Recording and Interpretation for Maritime Archaeology* (2019), pp. 11–27.

[148] *Smithsonian 3D Digitization*. `https://3d.si.edu/`. Accessed: 2022-03-10.

[149] *Stanford Scanning Repository*. `http://graphics.stanford.edu/data/3Dscanrep/`. Accessed: 2022-03-10.

[150] Gideon P Stein. “Lens distortion calibration using point correspondences”. In: *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE. 1997, pp. 602–608.

[151] Didier Stricker. “Computer-Vision-basierte Tracking-und Kalibrierungsverfahren für augmented reality”. PhD thesis. Technische Universität, 2003.

[152] Peter Sturm. "A case against Kruppa's equations for camera self-calibration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.10 (2000), pp. 1199–1204.

[153] Peter Sturm. "Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction". In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE. 1997, pp. 1100–1105.

[154] Peter Sturm. "On focal length calibration from two views". In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Vol. 2. IEEE. 2001, pp. II–II.

[155] Peter F Sturm and Stephen J Maybank. "On plane-based camera calibration: A general algorithm, singularities, applications". In: *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*. Vol. 1. IEEE. 1999, pp. 432–437.

[156] Deqing Sun et al. "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8934–8943.

[157] Shaharyar Ahmed Khan Tareen and Zahra Saleem. "A comparative analysis of sift, surf, kaze, akaze, orb, and brisk". In: *2018 International conference on computing, mathematics and engineering technologies (iCoMET)*. IEEE. 2018, pp. 1–10.

[158] Camillo J Taylor. "Implementing high resolution structured light by exploiting projector blur". In: *2012 IEEE Workshop on the Applications of Computer Vision (WACV)*. IEEE. 2012, pp. 9–16.

[159] Jos MF Ten Berge. "Orthogonal Procrustes rotation for two or more matrices". In: *Psychometrika* 42.2 (1977), pp. 267–276.

[160] Philip HS Torr and Andrew Zisserman. "Robust parameterization and computation of the trifocal tensor". In: *Image and vision Computing* 15.8 (1997), pp. 591–605.

[161] Miroslav Trajković and Mark Hedley. "Fast corner detection". In: *Image and vision computing* 16.2 (1998), pp. 75–87.

[162] Nickolay T Trendafilov and Ross A Lippert. "The multimode Procrustes problem". In: *Linear algebra and its applications* (2002).

[163] Bill Triggs et al. "Bundle adjustment—a modern synthesis". In: *International workshop on vision algorithms*. Springer. 1999, pp. 298–372.

[164] Zhigang Tu et al. "A survey of variational and CNN-based optical flow techniques". In: *Signal Processing: Image Communication* 72 (2019), pp. 9–24.

[165] Shinji Umeyama. "Least-squares estimation of transformation parameters between two point patterns". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* (1991).

[166] Sudheendra Vijayanarasimhan et al. "Sfm-net: Learning of structure and motion from video". In: *arXiv preprint arXiv:1704.07804* (2017).

[167] Victor Villena-Martinez et al. "When deep learning meets data alignment: A review on deep registration networks (drns)". In: *Applied Sciences* 10.21 (2020), p. 7524.

[168] Yang Wang et al. "Occlusion aware unsupervised learning of optical flow". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4884–4893.

[169] ZhenZhou Wang and YongMing Yang. "Single-shot three-dimensional reconstruction based on structured light line pattern". In: *Optics and Lasers in Engineering* 106 (2018), pp. 10–16.

[170] Zhongshi Wang et al. "Recognition and location of the internal corners of planar checkerboard calibration pattern image". In: *Applied mathematics and computation* 185.2 (2007), pp. 894–906.

[171] Michael Weinmann et al. "A multi-camera, multi-projector super-resolution framework for structured light". In: *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*. IEEE. 2011, pp. 397–404.

[172] Anthony Whitehead and Gerhard Roth. "Estimating intrinsic camera parameters from the fundamental matrix using an evolutionary approach". In: *EURASIP Journal on Advances in Signal Processing* 2004.8 (2004), p. 412751.

[173] Junfeng Xie et al. "A novel sub-pixel matching algorithm based on phase correlation using peak calculation". In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 1 (2016).

[174] Lili Yang et al. "Single-shot dense depth sensing with frequency-division multiplexing fringe projection". In: *Journal of Visual Communication and Image Representation* 46 (2017), pp. 139–149.

[175] Zhenheng Yang et al. "Every pixel counts: Unsupervised geometry learning with holistic 3d motion understanding". In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 2018, pp. 0–0.

[176] Ziv Yaniv. "Rigid Registration: The Iterative Closest Point Algorithm". In: *School of Engineering and Computer Science, The Hebrew University, Israel* (2001).

[177] Zhichao Yin and Jianping Shi. "Geonet: Unsupervised learning of dense depth, optical flow and camera pose". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1983–1992.

[178] Christopher Zach. "Robust bundle adjustment revisited". In: *European Conference on Computer Vision*. Springer. 2014, pp. 772–787.

[179] Pietro Zanuttigh et al. “Time-of-flight and structured light depth cameras”. In: *Technology and Applications* (2016), pp. 978–3.

[180] Ke Zhang et al. “Robust stereo matching with fast normalized cross-correlation over shape-adaptive regions”. In: *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE. 2009, pp. 2357–2360.

[181] Li Zhang, Brian Curless, and Steven M Seitz. “Rapid shape acquisition using color structured light and multi-pass dynamic programming”. In: *Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission*. IEEE. 2002, pp. 24–36.

[182] Ruo Zhang et al. “Shape-from-shading: a survey”. In: *IEEE transactions on pattern analysis and machine intelligence* 21.8 (1999), pp. 690–706.

[183] Song Zhang. “Recent progresses on real-time 3D shape measurement using digital fringe projection techniques”. In: *Optics and lasers in engineering* 48.2 (2010), pp. 149–158.

[184] Song Zhang and Peisen Huang. “High-resolution, real-time 3D shape acquisition”. In: *2004 Conference on Computer Vision and Pattern Recognition Workshop*. IEEE. 2004, pp. 28–28.

[185] Song Zhang and Peisen S Huang. “High-resolution, real-time three-dimensional shape measurement”. In: *Optical Engineering* 45.12 (2006), p. 123601.

[186] Song Zhang and Shing-Tung Yau. “Three-dimensional shape measurement using a structured light system with dual cameras”. In: *Optical Engineering* 47.1 (2008), p. 013604.

[187] Xiaowei Zhang et al. “Dense scene flow based on depth and multi-channel bilateral filter”. In: *Asian Conference on Computer Vision*. Springer. 2012, pp. 140–151.

[188] Zhengyou Zhang. “A flexible new technique for camera calibration”. In: *IEEE Transactions on pattern analysis and machine intelligence* 22.11 (2000), pp. 1330–1334.

[189] Zhengyou Zhang. “Determining the epipolar geometry and its uncertainty: A review”. In: *International journal of computer vision* 27.2 (1998), pp. 161–195.

[190] Zhengyou Zhang. “Iterative point matching for registration of free-form curves and surfaces”. In: *International journal of computer vision* 13.2 (1994), pp. 119–152.

[191] Zhiyuan Zhang, Yuchao Dai, and Jiadai Sun. “Deep learning based point cloud registration: an overview”. In: *Virtual Reality & Intelligent Hardware* 2.3 (2020), pp. 222–246.

[192] Kun Zhou et al. “Texturemontage”. In: *ACM SIGGRAPH 2005 Papers*. 2005, pp. 1148–1155.

[193] Tinghui Zhou et al. “Unsupervised learning of depth and ego-motion from video”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1851–1858.

[194] Alex Zihao Zhu et al. “Unsupervised event-based learning of optical flow, depth, and egomotion”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 989–997.

[195] Timo Zinßer, Jochen Schmidt, and Heinrich Niemann. “Point set registration with integrated scale estimation”. In: *International conference on pattern recognition and image processing*. 2005.

[196] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. “Df-net: Unsupervised joint learning of depth and flow using cross-task consistency”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 36–53.

Appendix A

Curriculum Vitae

Work Experience

since Oct. 2022	**Data Scientist** Mercedes-Benz Group, Germersheim
Jan. 2017 – Sep. 2022	**Researcher** University of Kaiserslautern Department of Computer Science, Augmented Vision Group
Jan. 2016 – Dec. 2016	**Research Assistant** German Research Center for Artificial Intelligence (DFKI), Kaiserslautern
Dec. 2014 – Dec. 2015	**Working Student** SAP, Walldorf
Feb. 2012 – Dec. 2014	**Working Student** PVM Vetterolf Maschinenbau GmbH, Mannheim

Education

Sep. 2014 – Oct. 2016	**Master of Science in Industrial Mathematics** University of Kaiserslautern Specialization: *Image Processing and Data Analysis* Technical Subject: *Mechanical Engineering*
Oct. 2010 – Sep. 2014	**Bachelor of Science in Mathematics** University of Kaiserslautern Specialization: *Image Processing and Data Analysis* Technical Subject: *Mechanical Engineering*
Sep. 2001 – Jun. 2010	**Abitur** Friedrich-Ebert-Gymnasium Sandhausen

Appendix B

List of Publications

INV-Flow2PoseNet: Light-Resistant Rigid Object Pose from Optical Flow of RGB-D Images using Images, Normals and Vertices, T. Fetzer, G.Reis and D. Stricker, *MDPI Sensors 22(22)*, 2022

ZebraPose: Coarse to Fine Surface Encoding for 6DoF Object Pose Estimation, Y. Su, M. Saleh, T. Fetzer, J. Rambach, N. Navab, B. Busam, D. Stricker, F. Tombari, *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022

Fast Projector-Driven Structured Light Matching in Sub-Pixel Accuracy using Bilinear Interpolation Assumption, T. Fetzer, G. Reis and D. Stricker, *Proceedings of the International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2021

Simultaneous Bi-Directional Structured Light Encoding for Practical Uncalibrated Profilometry, T. Fetzer, G. Reis and D. Stricker, *Proceedings of the International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2021

Joint Global ICP for Improved Automatic Alignment of Full Turn Object Scans, T. Fetzer, G. Reis and D. Stricker, *Proceedings of the International Conference on Computer Analysis of Images and Patterns (CAIP)*, 2021

Stable Intrinsic Auto-Calibration from Fundamental Matrices of Devices with Uncorrelated Camera Parameters, T. Fetzer, G. Reis and D. Stricker, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2020

Iterative Color Equalization for Increased Applicability of Structured Light Reconstruction, T. Fetzer, G. Reis and D. Stricker, *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, 2020

Ultrasound Tomography and 3D Scanning Technologies as a Tool to Constrain the Weathering State of Objects Made of Marble, J. Menningen, T. Fetzer, A. Schäfer, G. Reis and S. Siegesmund, *Proceedings of the International Congress on the Deterioration and Conservation of Stone (STONE)*, 2020

Robust Auto-Calibration for Practical Scanning Setups from Epipolar and Trifocal Relations, T. Fetzer, G. Reis and D. Stricker, *Proceedings of the International Conference on Machine Vision Applications (MVA)*, 2019

Introduction to Coherent Depth Fields for Dense Monocular Surface Recovery, V. Golyanik, T. Fetzer and D. Stricker, *Proceedings of the British Machine Vision Conference (BMVC)*, 2017

Accurate 3d reconstruction of dynamic scenes from monocular image sequences with severe occlusions, V. Golyanik, T. Fetzer and D. Stricker, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2017