

# Robot-based 3D reconstruction using Structure from Motion - Extending the Inline Computational Imaging System to a Robotic Arm

## DIPLOMARBEIT

Conducted in partial fulfillment of the requirements for the degree of a  
Diplom-Ingenieur (Dipl.-Ing.)

supervised by

Ao.Univ.-Prof. Dr. techn. Markus Vincze  
Mag. Dr. Bernhard Blaschitz  
Dipl.-Ing. Bernhard Neuberger

submitted at the

**TU Wien**

Faculty of Electrical Engineering and Information Technology  
Automation and Control Institute

by

Thomas Tramberger, BSc

Vienna, October 2021

---

**Vision for Robotics Group**

A-1040 Wien, Gusshausstr. 27, Internet: <http://www.acin.tuwien.ac.at>

---

# Preamble

I want to thank my supervisors, Markus Vincze, Bernhard Blaschitz and Bernhard Neuberger for their professional advice.

Furthermore, I would like to thank my family and especially my girlfriend for their great support throughout my studies.

Thomas Tramberger  
Vienna, October 2021

# Abstract

To meet today's high industry standards, computer vision systems are increasingly being integrated for quality control. These inspections must be carried out very quickly in order to not to slow down the production process. To meet these requirements, the Austrian Institute of Technology (AIT) has developed the Inline Computational Imaging system (ICI). The ICI is a monocular, scalable framework that is compatible with many industrial cameras. The system is limited to the use with a synchronized linear stage that guarantees image acquisition in a strict linear path and identical distance between images.

In this work, the ICI framework is extended for the use with a robotic arm. This brings many advantages, for example scanning surfaces of large objects that cannot be positioned on a linear stage. However, moving a camera with a robotic arm involves vibrations and inaccuracies. Therefore, it is expected that a direct application is not possible.

To achieve compatibility, the images must be transformed into the required arrangement. Two approaches are followed: in the first approach, the pose information is used to rectify the images. Since the position information of the robotic arm do not meet the required accuracy, they are optimized based on the image data using bound constrained bundle adjustment. In the second method, a perspective transformation is applied between the newly acquired image and the previous image to obtain the required arrangement. Feature tracks are used to keep the disparity at a constant.

The evaluation of the rectification processes shows a clear improvement of the vertical parallax via both approaches with a mean parallax in the subpixel range. The evaluation of the reconstruction confirms the improvement. Both approaches show a significant increase of quality of the resulting point clouds compared to the original images. The reconstructions deviate significantly less from the ground truth and show higher optical quality.

# Kurzzusammenfassung

Um den heute hohen Standards der Industrie gerecht zu werden, werden vermehrt computergestützte Bildverarbeitungssysteme zur Qualitätskontrollen integriert. Diese müssen sehr schnell erfolgen um den Fertigungsprozess nicht zu verlangsamen. Um diesen Anforderungen gerecht zu werden hat das Austrian Institute of Technology (AIT) das Inline Computational Imaging system (ICI) entwickelt. Das ICI ist ein monokulares, skalierbares Framework das mit vielen Industriekameras kompatibel ist. Das System ist jedoch auf den Einsatz mit einer synchronisierten Linearbühne beschränkt welche die Bildaufnahme in einem linearen Pfad und identen Aufnahmeabstand garantiert.

In dieser Arbeit wird der Einsatz des ICI auf die Anwendung mittels Roboterarme erweitert. Dies bringt viele Vorteile mit sich, beispielsweise das Scannen von Oberflächen großer Objekte die nicht auf einer Linearbühne positioniert werden können. Die Bewegung einer Kamera mittels Roboterarm ist jedoch mit vielen Vibrationen und Ungenauigkeiten verbunden. Es ist zu vermuten, dass ein direkter Einsatz nicht möglich ist.

Um die Kompatibilität zu bewerkstelligen müssen die Aufnahmen in die geforderte Anordnung transformiert werden. Dazu werden zwei Ansätze verfolgt: im ersten Ansatz werden die Poseninformation verwendet um die Bilder zu rektifizieren. Da die Positionsinformation der Bilder des Roboterarms zu ungenau sind, werden diese anhand der Bilddaten mittels grenzgebundener Bündelausgleichung (Bound Constraint Bundle Adjustment) optimiert. In der zweiten Methode wird eine perspektivische Transformation zwischen neu aufgenommenen und vorigem Bild geschätzt um das neue Bild zu rektifiziert. Um die Abstände konstant zu halten werden Merkmale auf mehreren Bildern gefunden und zur Kompensation verwendet.

Die Evaluierung der Rektifizierungsprozesse zeigt eine deutliche Verbesserung der vertikalen Parallaxe mit beiden Ansätzen mit einem Mittelwert im Subpixelbereich. Die Evaluierung der Punktwolken bestätigen die Verbesserung, die Rekonstruktionen weichen deutlich weniger vom Vergleichswert ab und zeigen optisch eine höhere Qualität.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Problem Statement . . . . .	4
1.2	Aim of the Work . . . . .	5
1.3	Proposed Solution . . . . .	5
1.4	Structure of the work . . . . .	5
<b>2</b>	<b>Related Work</b>	<b>7</b>
2.1	Inline Computational Imaging system . . . . .	7
2.2	Robot assistant 3D Reconstruction . . . . .	8
2.3	SfM and vSLAM . . . . .	9
2.4	Rectification of Camera Arrays . . . . .	11
<b>3</b>	<b>Background</b>	<b>13</b>
3.1	Camera Model and Projection . . . . .	13
3.1.1	Pinhole Camera . . . . .	13
	Lens distortions . . . . .	14
	Camera Parameters . . . . .	15
3.2	Stereo vision . . . . .	17
3.2.1	Epipolar geometry . . . . .	18
3.2.2	Essential Matrix . . . . .	19
3.2.3	Homography . . . . .	19
3.3	Structure from Motion . . . . .	20
3.3.1	Feature Extraction . . . . .	21
3.3.2	Feature Matching . . . . .	21
3.3.3	Geometric Verification . . . . .	21
3.3.4	Geometric Estimation . . . . .	22
3.3.5	Bundle Adjustment . . . . .	23
3.3.6	Visual SLAM . . . . .	24
3.4	Multi-view Stereo . . . . .	25
3.4.1	Photo-consistency . . . . .	25
3.4.2	3D Reconstruction . . . . .	27
	Depth map reconstruction . . . . .	28
	Point cloud reconstruction . . . . .	28

<b>4</b>	<b>Methodological approach</b>	<b>30</b>
4.1	Rectification via camera position . . . . .	31
4.1.1	Feature Extraction and Matching . . . . .	32
4.1.2	Geometric Verification and Triangulation . . . . .	33
4.1.3	Bound Constrained Bundle Adjustment . . . . .	34
4.1.4	SfM utilization and image rectification . . . . .	36
4.1.5	Boundary determination and mounting effects . . . . .	37
4.2	Rectification via perspective transform . . . . .	39
4.2.1	Homography estimation . . . . .	39
4.2.2	Feature Tracking and Disparity Correction . . . . .	40
<b>5</b>	<b>Experiments</b>	<b>42</b>
5.1	Data Acquisition . . . . .	42
5.1.1	Robot Control . . . . .	43
5.1.2	Optical settings . . . . .	46
5.2	Inspection of the trajectory . . . . .	47
5.2.1	Local Bundle Adjustment Approach . . . . .	48
5.3	Rectification Evaluation . . . . .	49
5.4	Reconstruction Evaluation . . . . .	53
	Comparison Step Mode vs Velocity Mode . . . . .	56
<b>6</b>	<b>Conclusion and Future Work</b>	<b>58</b>
6.1	Conclusion . . . . .	58
6.2	Future Work . . . . .	59
<b>7</b>	<b>Appendix</b>	<b>60</b>
7.1	Point clouds . . . . .	60

# List of Figures

1.1	Inline Computational Imaging system of the Austrian Institute of Technology [1]. The system requires a linear transport stage for almost perfect linear motion and constant disparity between images. . . . .	3
1.2	Graphical overview of the proposed solution. . . . .	6
2.1	Processing pipeline of the ICI[3]. . . . .	8
2.2	Algorithm comparison of open-source SfM software [19]. . . . .	11
3.1	Pinhole camera model [30] . . . . .	14
3.2	Lens distortions [30] . . . . .	15
3.3	Transformation from world coordinates into pixel coordinates. . . . .	16
3.4	Illustration of the concept of stereo vision [30]. . . . .	17
3.5	Concept of the epipolar geometry [34]. . . . .	18
3.6	Generic Structure from Motion Pipeline . . . . .	20
3.7	Incremental SfM Pipeline [19] . . . . .	22
3.8	Principle concept of the reprojection error. Rays from the camera centers through the 2D point on the image do not coincide at the same 3D point. The shift in pixels between the 2D point $u_{ij}$ and the re-projected 3D point $X_i$ onto the image plane $u'_{ij}$ is its re-projection error. The sum of all these errors for all 3D points visible from each image is the reprojection error [40]. . . . .	23
3.9	Concept of pixel matching. The surrounding area of a pixel and its intensity values are used to find pixel correspondences [41]. . . . .	26
3.10	Function responses of SSD and NCC along the epipolar lines of a textured (left) and textureless (right) image using a 3x3 squared region [32]. . . . .	27
3.11	Disparity map example. Original image (left) and its disparity image (right)[42]. . . . .	28
3.12	Point cloud example [43]. . . . .	29
4.1	The inconsistent arrangement seen on top has to be transformed into a uniform arrangement to meet the requirements of the ICI [27]. . . . .	30
4.2	Improved SfM pipeline . . . . .	32

4.3	Schematic influence of different factors on the boundaries. . . . .	38
4.4	Processing steps for an incoming image $I_k$ . . . . .	40
4.5	Feature tracks [46]. . . . .	41
5.1	Full experimental setup. . . . .	43
5.2	Block diagram of the KUKA LBR iiwas system. . . . .	46
5.3	Optical system . . . . .	46
5.4	Pose information scanning the object Random Pattern in Step Mode (1mm). . . . .	47
5.5	Pose information scanning the object Random Pattern in Velocity Mode (40mm/s) . . . . .	48
5.6	Pose information of local SfM approach using differnt window sizes. . . . .	49
5.7	Synthetic image with found feature tracks longer than 20 subsequent images. Ideally, feature tracks appear straight with uniform spacing between single feature points. . . . .	50
5.8	Mean vertical parallax between image pairs of the dataset board in mode Step. . . . .	51
5.9	Mean horizontal parallax between image pairs of the dataset board in mode Step. . . . .	52
5.10	(a) Linear Stage (b) Original (c) Homography (d) Global Synthetic image of a single feature trace of the dataset board. . . . .	53
5.11	Ground truth of evaluated objects. . . . .	54
5.12	(a) Original (b) Homography (c) Global (d) GT Top view of the reconstruction of dataset board in Step mode. . . . .	55
5.13	(a) Original (b) Homography (c) Global (d) GT In detail comparison of the key and coins on the object board. . . . .	56
5.14	Comparison of the Cornflakes dataset in both acquisition modes using the Global approach. . . . .	57
7.1	Object Random Pattern, Step Mode, 1mm . . . . .	60
7.2	Object Random Pattern, Velocity Mode, 40 mm/s . . . . .	61
7.3	Object Cornflakes, Step Mode, 1mm . . . . .	61
7.4	Object Cornflakes, Velocity Mode, 40 mm/s . . . . .	62
7.5	Object Board, Step Mode, 1mm . . . . .	62
7.6	Object Board, Velocity Mode, 40 mm/s . . . . .	63



# List of Tables

5.1	The recorded objects, each featuring a different level of difficulty.	44
5.2	Record settings . . . . .	45
5.3	Results of the evaluation processes. SIFT feature tracks are found, and their mean change in location are calculated. Then the variance is used to evaluate the rectification process vertically and horizontally. . . . .	51
5.4	Accuracy and Coverage of the different approaches. . . . .	54

# List of Abbreviations

**2D** two-dimensional.

**3D** three-dimensional.

**AIT** Austrian Institute of Technology.

**BA** Bundle Adjustment.

**BCBA** Bound constrained Bundle Adjustment.

**DLT** Direct Linear Transform.

**DoF** Degrees of Freedom.

**ICI** Inline Computational Imaging system.

**ICP** Iterative Closest Point.

**IMU** Inertial Measurement Unit.

**MVS** Multi-view Stereo.

**NCC** Normalized Cross Correlation.

**ORB** Oriented FAST and Rotated BRIEF.

**PMVS** Patch-based Multi-view Stereo.

**PnP** Perspective-n-Point.

**RANSAC** Random Sample Consensus.

**ROS** Robot Operating System.

**SfM** Structure from Motion.

**SIFT** Scale-invariant Feature Transform.

**SLAM** Simultaneous Localization and Mapping.

**SSD** Sum of Squared Differences.

**SURF** Speeded up Robust Features.

**SVD** Singular Value Decomposition.

**TGV** Total Generalized Variation.

**vSLAM** Visual Simultaneous Localization and Mapping.



Robotic arms combined with a vision system provide the possibility of scanning objects of different size and shape. While commonly available in industrial environments, they can be reprogrammed with low effort. The advantages of flexibility comes at a price of positional uncertainties and vibrations making it more difficult to robustly reconstruct. This work aims to combine best of both worlds. The ICI is applied on an industrial robotic arm to take advantages of both methods: a fast and precise reconstruction pipeline that can be applied in a flexible way.

## 1.1 Problem Statement

Industrial inspection has high requirements in terms of accuracy and speed. Depending on the task, these can surpass an acquisition speed of  $100\text{ mm/s}$  and depth resolutions in the lower micrometer range. Stated by the ETH3D High-resolution multi-view benchmark, multi-view stereo algorithms to recover 3D information are slow and far away from real-time applicability [2]. For instance, the fastest algorithm in the benchmark to recover the 3D structure of the door dataset, including only 7 high-resolution images, took over 80 seconds. Pure stereo reconstruction with subsequential depth map fusion can be applied in real-time, however these methods lack the benefits of multi-view stereo using only the pixel values of two images to recover its depth. Double the hardware is needed, increasing the mount on the robotic arm even more and leading to additional noise when following a trajectory. The ICI of the AIT features a flexible and scalable framework for fast and reliable recovery of depth information using a single camera. Different to conventional stereo, the ICI includes consistency checks using multiple images. However, the pipeline is designed for linear stages transporting the objects in almost perfect linear motion while the camera remains in a static position. This linear motion is essential as the pipeline assumes pixel correspondences lie on the same pixel row. When using a standard industrial robotic arm with several rotational joints to move the camera in a linear way, significant deviations to the perfect linear path occur. Robotic arms feature vibrations and inaccuracy depending on the underlying robot model. Further planning and moving a linear path with multiple rotational joints is also not straight forward and can lead to drift of the trajectory.

## 1.2 Aim of the Work

The main objective of this work is to extend the ICI to the use with a robotic arm. For this purpose, the robotic arm is moving a mounted camera system to scan the surface of an object. The robotic arm follows a given linear trajectory, mimicking a linear transport stage. The quality of the resulting reconstruction should not decrease using this form of acquisition. It is tested, if the positional quality the robot is sufficient for a direct adaption. Therefore, raw positional information retrieved from the robot model is inspected. The aim is to ensure that the image sequence complies with the ICI specifications in the best possible way.

## 1.3 Proposed Solution

To rectify the image sequence to fit the ICI requirements two methods are introduced. The first approach uses the position of the cameras to transform the images to an ideal camera arrangement. The robot model outputs the positions of the cameras at time of acquisition. Since these positions are not reliable, a constrained Bundle Adjustment (BA) is introduced. It is shown how the standard SfM pipeline can be improved by the additional information.

The second approach utilizes the homography between two image planes to project images into a common plane. The homography between images is retrieved by matching image features. It is used to transform the subsequential image onto the plane of the previous image. Therefore, the epipolar lines of the images are on the same row. To keep the baseline between images constant, feature tracks are used. Feature tracks are unique points found in multiple successive images.

Both approaches are applied incrementally to reduce the slowdown of the ICI pipeline. To assess the quality of the rectification, a comparison between a rectified image sequence and a image sequence taken with a linear stage using the ICI is performed. Figure 1.2 gives an overview of the proposed solutions. First an image sequence is acquired using the camera mounted on a robotic arm. Then, the images are transformed using two different approaches. Finally, the image sequence is applied to the ICI to receive a 3D reconstruction.

## 1.4 Structure of the work

In Chapter 2, state-of-the-art research of related fields are discussed. Chapter 3 introduces camera models and presents the fundamentals of stereo vision and Structure from Motion (SfM). Additionally, the incremental and global

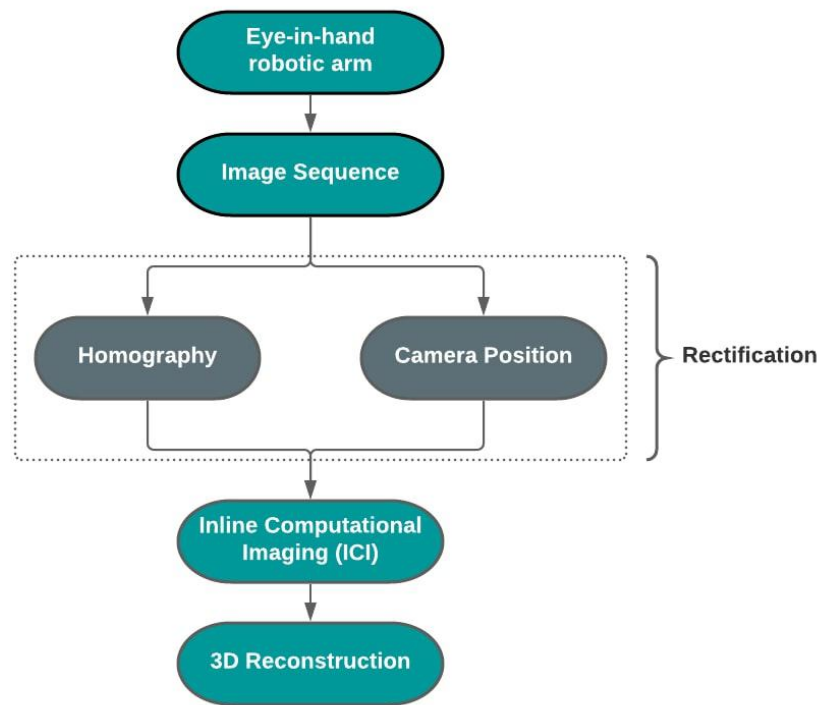


Figure 1.2: Graphical overview of the proposed solution.

approach of BA are introduced and methods for Multi-view Stereo (MVS) for dense reconstruction are presented. Chapter 4 explains the proposed solution in detail and how the positional prior is exploited for improved feature matching. Chapter 5 gives insight of the experimental setup and the results. Finally, a conclusion of the work is given and further improvements are discussed.

## 2 Related Work

This work aims to achieve a high-quality 3D reconstruction by utilizing the ICI combined with images taken by an optical system attached to a robotic arm. Therefore, the following chapter gives current approaches of robot-based 3D reconstruction. Then, research related to the proposed solutions about SfM and Simultaneous Localization and Mapping (SLAM) are presented and current state-of-the-art variations are introduced. Finally, methods for rectification of linear camera arrays are shown as the goal is similar to the aim of this work.

### 2.1 Inline Computational Imaging system

The Inline Computational Imaging system of the Austrian Institute of Technology [1] is a fast, reliable way for inspection and 3D reconstruction in industrial applications. The algorithms are flexible and can operate with most standard industrial cameras under many different lighting conditions and image resolutions. The setup consists of an industry grade camera and a transport stage that moves an object in front of the camera. The system is also capable of photometric stereo. The typical setup includes four high power LED light sources, arranged in four different directions, that are strobed sequentially. Compared to this work, where the robotic arm is moving the camera in a linear way, in the original use, the object is moved on a linear transport stage and the optical system is static.

The computational approach features four stages of processing: feature calculation and multi-view matching, fusion of the disparity maps, generation of the 3D model and final regularization and denoising. In the first step,  $m$  different feature sizes are matched for a selected baseline in both directions, leading to  $m$  disparity maps and confidence maps for each image. Taking the mean of these confidence maps and a weighted mean over the disparities, including some further consistency checks with different baselines, the disparity maps are fused into a single disparity map per image. Then the single disparity maps are integrated into a single scene. Finally, the 3D model is postprocessed using the confidence values and an iterative Total Generalized Variation (TGV) solver for denoising [1].



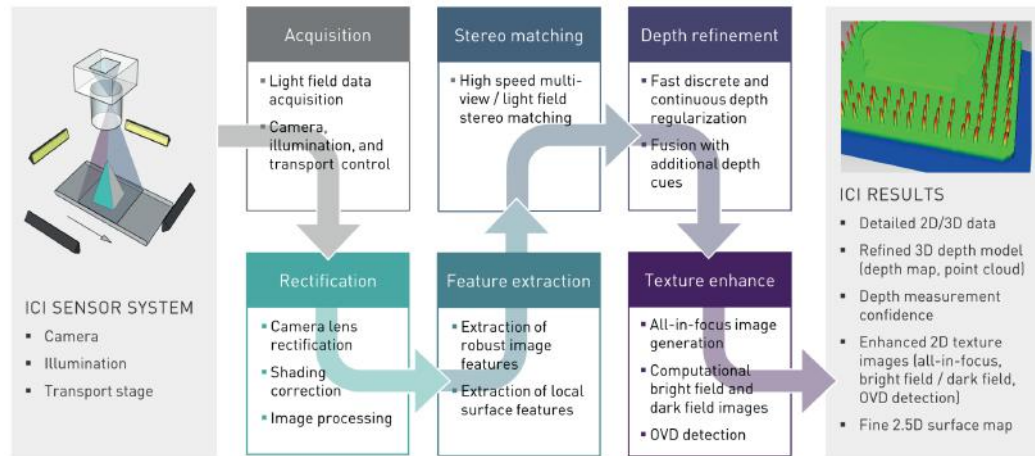


Figure 2.1: Processing pipeline of the ICI[3].

Figure 2.1 shows the original hardware setup and the main steps of the processing pipeline of the ICI.

## 2.2 Robot assistant 3D Reconstruction

Other works for 3D reconstruction utilizing a robotic arm mainly use different sensors for reconstruction. Rossi and Savino [4] as well as Callieri et al. [5] use laser scanners mounted to a robotic arm to scan objects. Huang et al. [6] combined a robotic arm with an ultrasonic scanner for medical application and Alenya et al. [7] created a eye-in-hand system by mounting a Time-of-Flight camera. For monocular reconstruction,  $R^2$ OBBIE-3D by Martins et al. [8] features a robotic arm combining a high resolution camera and an illumination basket to reconstruct biological objects. They connect photometric stereo and Patch-based Multi-view Stereo (PMVS) for high quality reconstruction, constraining the PMVS algorithm directly with the robot pose. Even if they produced high accurate reconstruction results, the runtime is not even close to real-time and took several hours. In the industrial field, Zambal et al. [9] developed an inspection system for carbon fibre reinforced plastic. This method does not directly 3D reconstruct the scene, but visualizes the fibre orientations.

## 2.3 SfM and vSLAM

SfM is a technique to retrieve the 3D geometry. This process includes both problems of estimating camera poses and reconstructing a scene in sparse 3D. Visual Simultaneous Localization and Mapping (vSLAM) is a comparable problem introduced by the robotic research community while SfM was developed in the computer vision community. Both feature similar pipelines using mainly feature correspondences and either BA, or in vSLAM different filter-based approaches to optimize over selected images. SfM was initially introduced 1981 in the seminal paper of Longuet-Higgins [10], reconstructing a scene from two images and estimating the related camera rotation introducing the eight point algorithm. The first filter-based approach, namely MonoSLAM [11], was developed 2003 using Bayesian filtering. As the goals of the methods in these communities are different, the evolution of SfM and vSLAM proceeded differently. However, the introduction of incremental SfM, that is capable of operating in real-time, brought the two methods back together [12]. The first vSLAM system to reunite the two branches is known as Parallel Tracking and Mapping (PTAM) [13].

### SLAM

Today's research in SLAM mainly tackles the problem of fusing information of multiple different sensors for more robustness and more accurate results. These are often referred to as visual-inertial SLAM systems. Most research is done using an Inertial Measurement Unit (IMU) and a camera sensor e.g. OKVIS [14], VINS-MONO [15]. In the last years, many SLAM systems, utilizing different approaches using different sensors, were developed. The following section focuses only on pure monocular visual, classical SLAM systems using graph-based optimization equal to BA. While these either use direct, semi-direct or feature-based methods, all of following systems can lead to good or even very good results according to [15]. Direct methods use pixel matches instead of features. They work more robust on low-textured surfaces while also retrieving a denser map of the environment. Their limitations, on the other hand, include the assumption of a constant surface reflectance model and a limited baseline in consequences of the photometric consistency. Additionally, they are more computationally heavy because pixel-base matching and the denser map [16].

**ORB SLAM** was initially developed by Mur-Artal et. al. [16]. The feature-based system uses Oriented FAST and Rotated BRIEF (ORB) features to allow real-time performance even without a GPU. Their extraction needs less time per image than the popular Scale-invariant Feature Transform (SIFT) or Speeded

up Robust Features (SURF) features. ORB also features good invariance to viewpoints resulting in better matches when dealing with a wider baseline. The main idea of ORB SLAM is to reuse features that are used for mapping and tracking also for place recognition, for relocalization and loop detection. Each of the tasks, tracking, mapping and loop closing are running on an own thread. For tracking, a constant velocity model is used to get initial pose estimations for new frames. A motion-only BA is used to optimize the position of new frames. If the track is lost because of e.g. not enough matches the frame is converted into a bag of words and searched for in the global map. The map is projected into the frame for further features matches and it is decided if new keyframes are added to the local map. ORB SLAM 2 [17], the first follow up paper, added additional support for stereo and RGB-D cameras. Furthermore, ORB SLAM 3 [15] extended the system to be able to use inertial data.

**Semi-direct visual odometry (SVO)** was one of the first visual odometry systems to discard the feature-based approach of monocular systems. To estimate the camera motion, the pixel's intensity values of an image are directly used exploiting the information of the hole image. For an initial camera pose estimation, direct motion estimation is utilized using small patches. The name semi-direct comes from the fact that features are still used when adding new 3d points. The extension of SVO Slam, SVO Slam 2 [18], features additional support for edgletes, IMU prior, wide angle cameras, multi-camera configurations, and forward looking camera motion.

Figure 2.2 lists the most popular open-source software for SfM and their provided algorithms for each pipeline stage. These include or can be easily processed with MVS algorithms for dense 3D reconstruction.

## Bundle Adjustment

The following section presents state-of-the-art work of the final BA stage. For further reading, Ozyesil et al. [20] presents an methodical overview of the main steps of camera location estimation and BA, also including early works.

The mathematical basics of BA are well understood and the Levenberg-Marquardt algorithms has itself proven as the most successful method for optimization. In the work of Chen et al.[21], the primary field of research are presented. The main research areas are to increase efficiency, to reduce runtime and memory usage, and the utilization of a GPU for parallel bundle adjustment. For very large datasets, distributed approaches were developed [21].

Further works covering constraint BA, infuse additional knowledge. Irschara et al. [22] use prior pose information from GPS and IMU sensors for view

	Feature Extraction	Feature Matching	Geometric Verification	Image Registration	Triangulation	Bundle Adjustment	Robust Estimation
<b>COLMAP</b>	SIFT [31]	Exhaustive Sequential Vocabulary Tree [37] Spatial [14] Transitive [14]	4 Point for Homography [20] 5 Point Relative Pose [33] 7 Point for F-matrix [20] 8 Point for F-matrix [20]	P3P [32] EPnP [34]	sampling-based DLT [14]	Multicore BA [27] Ceres Solver [35]	RANSAC [19] PROSAC [36] LO-RANSAC [38]
<b>OpenMVG</b>	SIFT [31] AKAZE [39]	Brute force ANN [40] Cascade Hashing [41]	affine transformation 4 Point for Homography [20] 8 Point for F-matrix [20] 7 Point for F-matrix [20] 5 Point Relative Pose [33]	6 Point DLT [20] P3P [32] EPnP [34]	linear (DLT) [20]	Ceres Solver [35]	Max-Consensus RANSAC [19] LMed [42] AC-Ransac [43]
<b>Theia</b>	SIFT [31]	Brute force Cascade Hashing [41]	4 Point for Homography [20] 5 Point Relative Pose [33] 8 Point for F-matrix [20]	P3P [32] PNP (DLS) [44] P4P [46] P5P [49]	linear (DLT) [20] 2-view [45] Midpoint [47] N-view [20]	Ceres Solver [35]	RANSAC [19] PROSAC [36] Aracsac [48] Evsac [50] LMed [42]
<b>VisualSFM</b>	SIFT [31]	Exhaustive Sequential Preemptive [16]	n/a	n/a	n/a	Multicore BA [27]	RANSAC [19]
<b>Bundler</b>	SIFT [31]	ANN [51]	8 Point for F-matrix [20]	DLT based [20]	N-view [20]	SBA [52] Ceres Solver [35]	RANSAC [19]
<b>MVE</b>	SIFT [31] + SURF [53]	Low-res + exhaustive [29] Cascade Hashing	8 Point for F-matrix [20]	P3P [32]	linear (DLT) [20]	own LM BA	RANSAC [19]

Figure 2.2: Algorithm comparison of open-source SfM software [19].

selection, but the information is not used to restrict the final optimization step. Lhuiller [23] introduces two constraint BAs fusing GPS data to enforce an upper bound on the reprojection error. Bound constrained Bundle Adjustment (BCBA) by Gong et al. [24] gives a more general approach to constraining the problem and is also utilized in this work. The work also mention, that previous works for constraining the BA are not generally applicable and only restricted to special motion or geometric constrains.

## 2.4 Rectification of Camera Arrays

The following section present research about rectification of linear camera arrays and also image mosaicing. For uncalibrated cameras, Zhang et al. [25] introduced a block-division feature extraction method for robust fundamental matrix estimation and a projection shift method to transform all images to a common plane. Finally, a disparity adjustment is applied to ensure constant disparity between images. In the work of Zilly et al. [26] the trifocal tensor is utilized for image rectification into a common baseline and further targeting the horizontal alignment. For known intrinsic and extrinsic camera parameters, Kang and Ho [27] present a multi-view image rectifying transform to reproject the images into a ideal parallel multi-camera arrangement reducing geometrical errors.

For areal imaging, Zhu et al. [28] proposed an automatic method for image mosaicing under constrained 6 Degrees of Freedom (DoF) motion using GPS measurements. In Geng et al. [29] a real-time algorithm for epipolar resampling of linear pushbroom images is introduced.

## 3 Background

The following section presents the basics that are required to reconstruct a scene from multiple 2D images. Starting with the most essential tool in computer vision, the camera, to record the scene, to further introducing the basics of stereo vision to retrieve depth from two images. Afterwards, a generic pipeline for SfM and the concept of MVS are presented. The main sources are Kaehler and Bradski [30] for Section 3.1 and Section 3.2, Szeliski [31] for Section 3.3 and Fukurawa [32] for Section 3.4.

### 3.1 Camera Model and Projection

Retrieving 3D information from a set of images requires knowledge of the used camera and optics. The camera model describes the relationship of 3D points in the world and their projection onto the 2D image plane. The parameters that present the camera model can be obtained by using a camera calibration process. Optics introduce radial distortions due to the shape of the lens as well as tangential distortions from the assembly process of the camera. These can harm the quality of the reconstruction, but can be also be eliminated by more complex camera models.

#### 3.1.1 Pinhole Camera

The simplest form of a camera model is the pinhole camera model. The camera aperture is only a single point, rather than a lens, just the size that light rays are able to pass through. From each scene point, a single light ray is emitted through the pinhole and projected onto the image plane. The image plane and pinhole plane are parallel. The distance between these two planes is called the focal length  $f$  while the distance from the scene point to the pinhole is its depth  $Z$ .  $X$  is the distance of the scene point to the optical axis and  $x$  its projection onto the image plane. The optical axis is the axis that goes through the pinhole while being normal to both planes. Figure 3.1 displays the simple pinhole model.

The geometry of the model leads to similar triangles, both having the projection line between  $X$  and  $x$  as hypotenuses. This can be transformed to

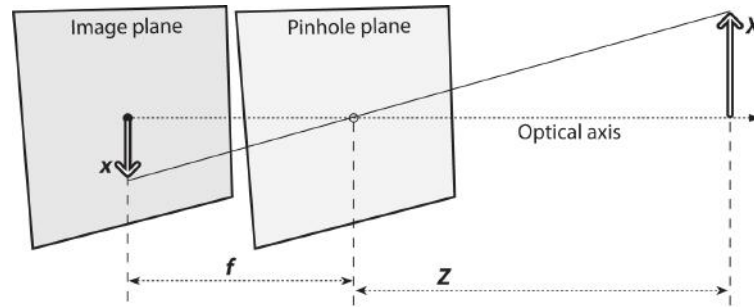


Figure 3.1: Pinhole camera model [30]

equation (3.1) to describe the relation between the 3D coordinates and the 2D image coordinates of a point. Consequently the projection of the scene will be flipped on the image plane.

$$-x = f * \frac{X}{Z} \quad (3.1)$$

The simple pinhole model does not take into account that most practical cameras have only discrete image coordinates. This means that the pinhole camera model can only be used as a first order approximation of the mapping from a 3D scene to a 2D image. In practice the small hole does not produce bright enough images. It also ignores the effects of distortion that are introduced by using a lens to capture more light.

### Lens distortions

As already mentioned, compared to the pinhole model, cameras use optical lenses to capture more light. The use of an aperture is necessary, but in practice does not come without disadvantages. This comes from the fact that in practice, spherical lenses are used, as they are easier to produce, and the imperfect mechanical alignment of the lens to the image sensor. Radial distortions are the result of the shape of the lens, tangential distortions are introduced by the faulty camera assembly process.

Radial distortions are mainly noticeable at the edges of an image, as the effect increases with the distance from the optical axis. The distortions occur because the light rays further away from the optical axis are more bent compared to rays closer to the optical axis. The fact that the lens is thicker at the optical axis results in a "barrel" effect that can be seen in Figure 3.2. Mathematically, the radial distortion can be approximated by the first few terms of the Taylor series expansion around  $r=0$ . To correct the pixel values  $x$ ,



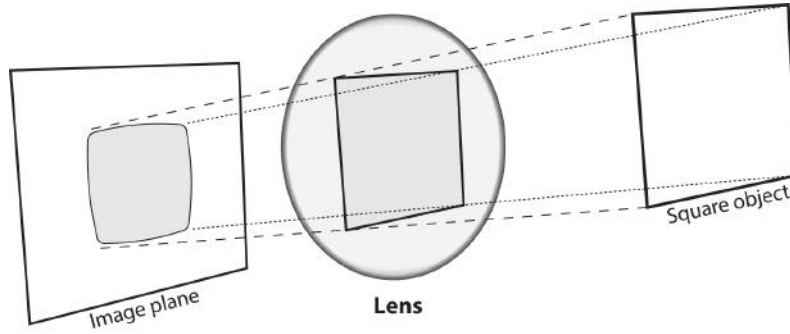


Figure 3.2: Lens distortions [30]

y the first three search terms  $k_1, k_2, k_3$  are used as in equations (3.2) [30].

$$\begin{aligned} x_{corrected} &= x * (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ y_{corrected} &= y * (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \end{aligned} \quad (3.2)$$

Tangential distortion occurs when the image plane and the lens are not parallel and can be expressed using two parameters  $p_1$  and  $p_2$ . To correct the pixel values, equations (3.3) are used [30].

$$\begin{aligned} x_{corrected} &= x * (2 * p_1 * x * y + p_2 * (r^2 + 2 * x^2)) \\ y_{corrected} &= y * (p_1 * (r^2 + 2 * y^2) + 2 * p_2 * x * y) \end{aligned} \quad (3.3)$$

### Camera Parameters

After undistorting the image and projecting a 3D scene point through the ideal pinhole model onto our image plane, it still has to be transformed to resulting pixel coordinates, as these start at the upper-left corner of the image. The intrinsic parameters of a camera include all factors needed to project a 3D scene point onto an image pixel. It includes the focal length ( $f_x, f_y$ ), the optical center ( $c_x, c_y$ ), also known as the principal point, and a skew coefficient  $s$  between the x and y axis. When using perspective transformations it is convenient to work in homogeneous coordinates. Therefore, the camera parameters are often expressed in matrices. (3.4) shows the camera intrinsics matrix  $K$ .

$$K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$



To retrieve the camera intrinsic parameters a calibration process has to be performed. For this purpose, known calibration points are used, mainly in the form of patterns e.g. of a checkerboard pattern with known spacing between corners. A popular calibration algorithm is Zhang's method [33].

The camera extrinsics matrix  $E$  encodes the camera's position and orientation in world coordinates. The origin of the camera's coordinate system is located at its optical center. It consists of a rotation matrix  $R$  and a translation vector  $t$  as shown in (3.5).

$$E = [R \ t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

When using both, the camera intrinsics matrix and the camera extrinsics matrix, we can fully describe how points of the world are projected to an image. These two matrices can be combined to form the projection matrix  $P$  (or camera matrix) that can be directly used for the transformation from world to pixel coordinates.

$$P = K [R \ t] \quad (3.6)$$

$$x = PX = K[Rt]X$$

Figure 3.3 displays the rigid transformations of world coordinates into camera coordinates using the extrinsic parameters and the projective transformation, from camera coordinates into pixel coordinates, using intrinsic parameters.

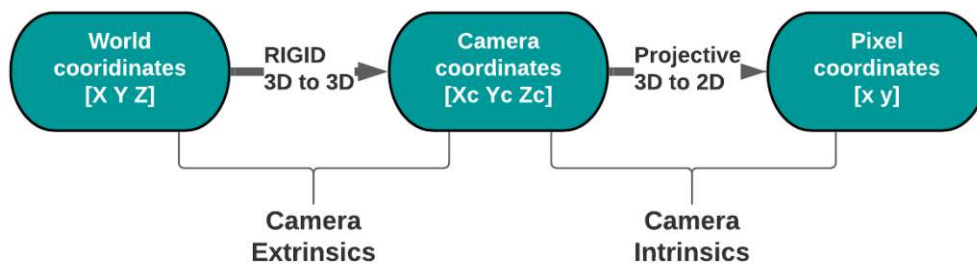


Figure 3.3: Transformation from world coordinates into pixel coordinates.

## 3.2 Stereo vision

Inspired by the human binocular vision system, stereo vision is the extraction of 3D information from two images observing the same scene from different viewpoints. In comparison with the previous section, we go the other way to get the depth of a scene point by using only 2D information and known camera parameters. A single image cannot be used to regain the depth of a point, but it projects possible solutions along a line. If another image displays the same point, the depth of the scene point can be found at the intersection of the two rays. By knowing the distance between the cameras, the depth information of a point can be retrieved easily by triangulation. Figure 3.4 shows the principle of stereo vision. The depth  $Z$  of the Point  $P$  is located at the intersection of the two rays from the optical centers ( $O_l, O_r$ ) through the pixel locations of the point ( $p_l, p_r$ ). The distance between the cameras  $T$  has to be known to be able to calculate the depth. As pixel values are discrete the depth can only be calculated up to a certain accuracy.

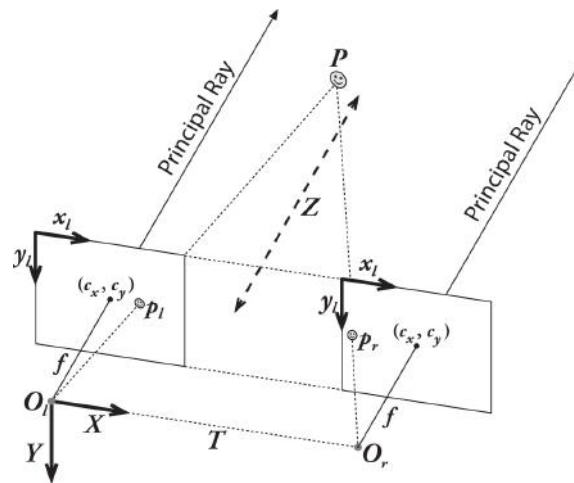


Figure 3.4: Illustration of the concept of stereo vision [30].

Via the method of similar triangles the depth can be computed as in (3.7).  $d$  is called the disparity and is the difference between the point's  $P$  pixel coordinate on the left image  $p_l$  and the right image  $p_r$

$$Z = f * \frac{T}{d} = f * \frac{T}{p_l - p_r} \quad (3.7)$$

### 3.2.1 Epipolar geometry

There are many ways to find correspondences between images, mainly using the local appearances. For each pixel on an image, each pixel of another image could be a potential match, making the process slow and unreliable. In stereo vision the additional information of the relative camera positions and the camera intrinsics are exploited to reduce the 2D correspondence search to a 1D search.

Each camera center ( $O_l, O_r$ ) is projected into the other camera's image plane at  $e$  and  $e'$ , the so-called epipoles. The ray between the camera centers and the two rays from the camera centers to the point  $P$  span the epipolar plane. The connection between the image points ( $p_l, p_r$ ) and the related epipole result in the epipolar lines. The line is equivalent to the projection of the line between  $O_l$  and  $P$  onto the right image plane and vice versa. Therefore, the corresponding point on one image can be found on the epipolar line of the other image, reducing the complexity from a 2D to a one-dimensional searching problem. Figure 3.5 displays the concept of the epipolar geometry and shows the epipoles and epipolar lines. In this case the images are tilted towards each other. In a more simple case as seen in Figure 3.4, both camera pixel coordinate systems are aligned and the cameras face into the same direction. The epipolar lines are then parallel to the line between the camera centers, resulting in a search in the same pixel row on the other image. Via a rectification process the images can be transformed into this simple case.

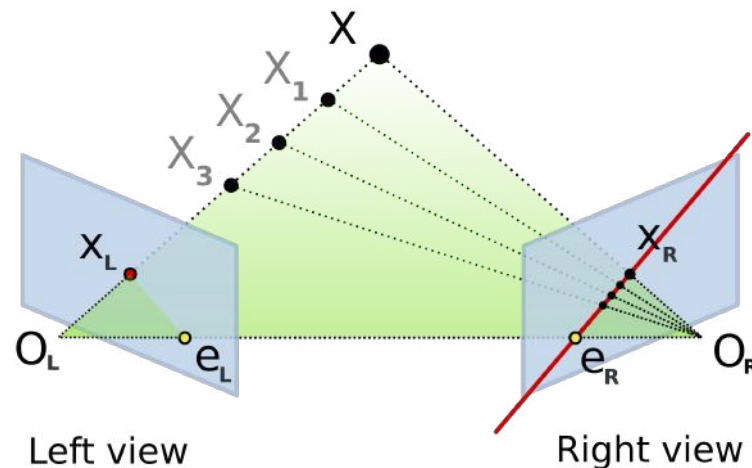


Figure 3.5: Concept of the epipolar geometry [34].

### 3.2.2 Essential Matrix

The essential matrix is a 3x3 matrix and encodes information about the geometry of the stereo configuration. The essential matrix  $E$  maps each point  $p_l$  of the plane of an image to another point  $p_r$  on the plane of another image. As shown in (3.8) it is defined as the cross product of the baseline vector between the cameras and their relative rotation. If the essential matrix is known, these can be extracted e.g. by using Singular Value Decomposition (SVD). For the baseline vector only the direction can be extracted, but not its length.

$$E = [t] \times R \quad (3.8)$$

(3.9) gives the epipolar constraint that has to hold for all corresponding points. Hence, knowing the essential matrix simplifies the search for correspondences as it leads to an equation for a line. This comes from the fact that it is a rank deficient matrix of rank 2. If the essential matrix is unknown it can be derived from given correspondences with e.g. the Five-Point Algorithm by [35] due to its five DoF, three for rotation and two for the direction of translation.

The fundamental matrix is similar to the essential matrix, but additionally includes information of the intrinsic camera parameters. Therefore, it relates two points directly in pixel coordinates, compared to the essential matrix that relates points in camera coordinates.

$$p_l^T E p_r = 0 \quad (3.9)$$

### 3.2.3 Homography

A homography describes the relationship between two images viewing the same planar surface. Its often used for image rectification, determining the motion between two images and image mosaicing. The homography matrix can be used to transform an image into the original image plane with correct perspective. Equation (3.10) shows that the homography matrix is a 3x3 matrix to directly transform 2D points. It has 8 DoF as it is estimated up to a scale [31].

$$\begin{bmatrix} x'/\lambda \\ y'/\lambda \\ \lambda \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} = 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.10)$$

To determine the homography matrix between two image planes via feature correspondences, the Direct Linear Transform (DLT) algorithm by Hartley and Zisserman [36] is used. When more correspondences are found than required, the problem can be further described as a least-squared problem minimizing the re-projection error for all correspondences  $i$  with the error function

$$\sum_i \left( x'_i - \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \right)^2 + \left( y'_i - \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \right)^2 \quad (3.11)$$

where,

$$x', y' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}. \quad (3.12)$$

is used to transform the pixel coordinates and warp the image afterwards.

### 3.3 Structure from Motion

The previous section shows how to extract depth of a scene from two images. SfM is a method to derive 3D information of multiple 2D images. As the process only takes 2D image data and potentially camera intrinsic parameters, 3D geometry (structure) and the camera poses (motion) are estimated simultaneously [31]. Figure 3.6 shows a generic SfM pipeline that divides the process into several sub tasks. First, distinct image features are extracted and matched against each image. Given these feature correspondences, the relative camera positions can be estimated and the depth of the scene can be extracted via triangulation. As mentioned in section Section 3.2.2, the depth information can only be retrieved up to a scale, since the length of the baseline vectors are unknown. Finally, to more robustly reconstruct the scene, a non-linear minimization method, called BA, is applied. The following subsections describe each step of the pipeline in detail.

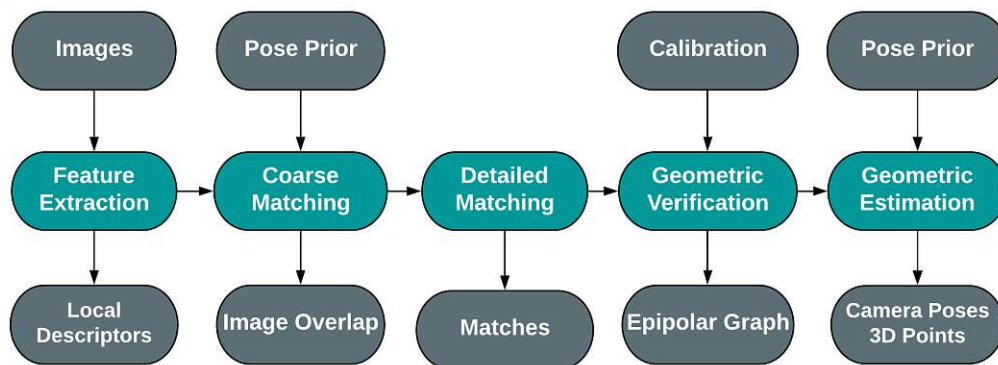


Figure 3.6: Generic Structure from Motion Pipeline

### 3.3.1 Feature Extraction

Image features are distinct areas in an image consisting of a keypoint and a descriptor including neighbourhood information. They are required to be highly unique while being recognizable in other images. There are many popular approaches to detect these unique points and to describe the neighbourhood of these points of interest. A basic approach is to use corners as they are rich on local information. Another, more complex, example are SIFT features by Lowe [37], these are also robust to translation, rotation and scaling. The more texture available in an area the more features can be extracted [31].

### 3.3.2 Feature Matching

After local features are detected in the images, corresponding point pairs are found. To avoid  $N \times N$  feature matching, coarse matching is done initially or pose priors can be utilized to find overlapping images. Then detailed matching is performed only for these overlapping images. There are several approaches to feature matching, the most basic idea is to compare each feature descriptor of an image against each other feature descriptor of another image and simply take the most similar feature as a match. This method is called brute-force matching [30].

### 3.3.3 Geometric Verification

To assure that the found 2D corresponding points of the previous step share the same 3D point and to exclude outliers, a geometric verification step is performed. Therefore, a geometric transformation between two images is found for a sufficient number of points. This geometric transformation is expressed by the essential matrix for calibrated cameras or the fundamental matrix for uncalibrated cameras. These can be determined by the epipolar constraint that has to hold for every corresponding point pair (see Section 3.2.2). To eliminate outliers, a Random Sample Consensus (RANSAC) algorithm is used to obtain a more accurate essential/fundamental matrix [22].

The output of the geometric verification and the foundation of the following geometric estimation stage is a Scene graph (or Epipolar graph). The Scene graph's nodes represent images, the edges between these nodes show the geometrically verified 2-view matches. Therefore, the Scene graph defines a sequential order for geometric processing [19], [38].

### 3.3.4 Geometric Estimation

The Geometric Estimation step is the final step of the SfM Pipeline. It takes the Scene graph as an input and outputs optimized pose estimates of the registered images and the reconstructed scene represented by a set of 3D points. If the whole Scene graph is considered at a time, the process is referred to as global SfM, in contrast to the incremental approach that takes new images one at a time to grow the reconstruction. When using the incremental method, some steps have to be repeated when adding a new image. Consequently, the global SfM approaches are, in general, faster. Studies show that global SfM also produce more accurate results than incremental SfM for small data-sets, but takes a significant larger amount of memory [39]. Figure 3.7 shows a incremental SfM pipeline and the repeating steps for each added image.

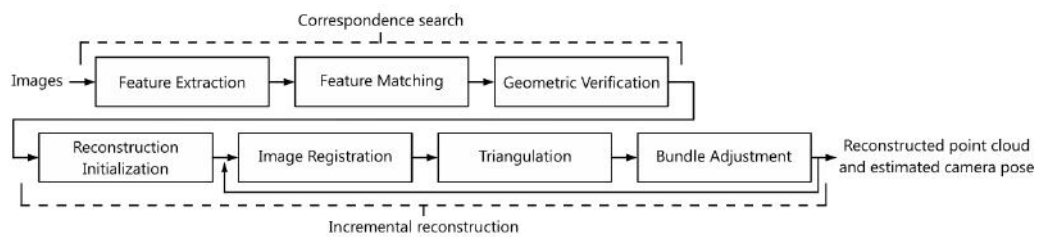


Figure 3.7: Incremental SfM Pipeline [19]

**Initialization** Initialization is crucial for good reconstruction as the later applied optimizing process, called bundle adjustment, needs a solid starting point. Therefore, a geometric verified image pair of a dense region of the Scene graph is chosen. The matching points of this image pair are used for the first reconstruction and to calculate the initial two camera positions. Via Image Registration, Triangulation and BA, additional images are added [19], [38].

**Image Registration** When adding an image to the reconstruction, the pose of its camera has to be calculated. This is done by using the correspondences of the already reconstructed 3D points and 2D features of the image to solve the Perspective-n-Point (PnP) problem. One example to do so is the P3P algorithm, which uses the least amount of information needed [31]. Once again, a RANSAC algorithm is used to eliminate outliers of the 2D-3D correspondences. A newly added image has to observe existing points of the reconstructed scene [19], [38].



**Triangulation** After calculating the camera pose of the newly added image, we now want to append the image information to the 3D reconstruction and extend the set of 3D points. However, when adding a point to the reconstruction it has to be observed by at least one registered image. Then, a triangulation process is used to calculate the coordinate of the new 3D point. There is also the case that many images have scene point in common. This problem is called multi-view triangulation [38]. Because of inaccuracies of the previous pose estimation stage, the point position does not intersect correctly resulting in a reprojection error (see Figure 3.8)[19].

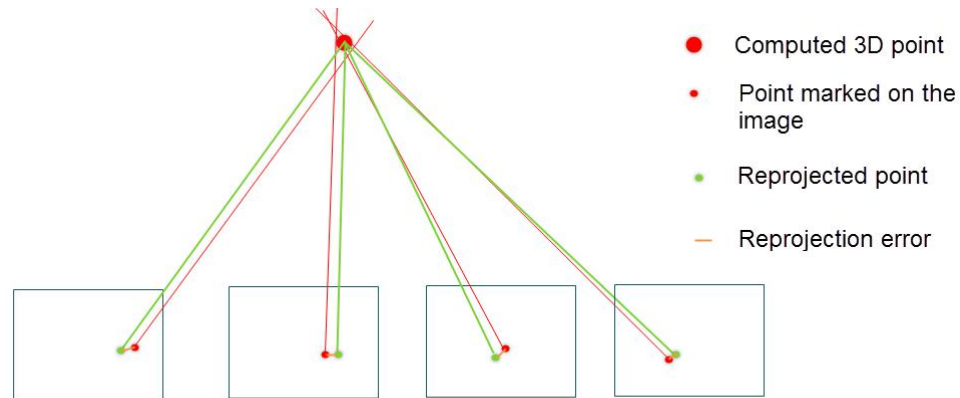


Figure 3.8: Principle concept of the reprojection error. Rays from the camera centers through the 2D point on the image do not coincide at the same 3D point. The shift in pixels between the 2D point  $u_{ij}$  and the re-projected 3D point  $X_i$  onto the image plane  $u'_{ij}$  is its reprojection error. The sum of all these errors for all 3D points visible from each image is the reprojection error [40].

### 3.3.5 Bundle Adjustment

The name "Bundle Adjustment" refers to bundles of rays arise from the camera centers to the 3D points and the adjustment of camera parameters to minimize the reprojection error [31]. It is the final refinement step that is taken before the dense 3D reconstruction to optimize the sparse 3D points retrieved from the triangulation stage as well as the camera parameters.

The previous steps of estimating the camera pose and triangulation include



errors resulting in inaccuracies in the reconstruction. When reprojecting the 3D points from the triangulation step back to the 2D image plane using the camera parameter, it often does not align with the same pixel as it presents. To reduce the accumulated error, the BA, a non-linear optimization, is performed. BA minimizes the reprojection error and can result in optimal camera parameters and 3D point locations.

$$\min \sum_{i=1}^n \sum_{j=1}^m (u_{ij} - \pi(C_j, X_i)) \quad (3.13)$$

Equation (3.13) mathematically represents the reprojection error.  $u_{ij}$  represent a set of observed points on pixel level which present the  $i$ th 3D point  $X_i$  observed by the  $j$ th camera  $C_j$ .  $\pi(C_j, X_i)$  is the projection function of 3D points  $X_i$  using the camera parameters of  $C_j$ .

The current state-of-the-art method to solve the BA problem is the Levenberg-Marquardt algorithm, also known as damped least squares method. The implementation is fairly easy while it is robust to a wide range of initialization. Chapter 4 includes a more detailed look into solving the bundle adjustment problem.

Incremental SfM has to repeat the BA process for every newly added image, taking every other already included image and corresponding point into account. This can lead to high computational cost with long processing times. To reduce the processing, local BA can be used. It only considers a small number of images, usually the most connected images or in a image sequence the latest images added. These approaches can also be combined: local BA is executed for newly added images and after the reconstructed point cloud has grown a certain size, global BA is performed.

### 3.3.6 Visual SLAM

vSLAM is a similar problem as SfM introduced by the robotics community. Due to the fact that robots often have to localize themselves in an environment, the camera pose is more relevant compared to SfM where its only used for reconstruction purposes. Technically there is no difference between the two techniques, expect that vSLAM is meant to run in real-time, however there are also online SfM methods available. Main differences are that vSLAM systems are split into three threads: tracking, mapping and loop closing. SLAM system can also include further sensor measurements, e.g. Inertial Measurement Unit (IMU), to increase the localization performance and get scale information.

Additionally, to the Graph-based approach equal to SfM, two other paradigms for SLAM exist: the extended Kalman Filter and the Particle Filter.

## 3.4 Multi-view Stereo

The following section gives a limited insight into the enormous amount of approaches to MVS, as this work does only use existing pipelines and does not contribute to this topic. However, knowledge of these methods are required since the parameters of those pipelines have to be chosen manually. While SfM is used to estimate camera positions, camera parameters and a sparse presentation of the scene, MVS takes this information to produce a dense 3D point cloud of the scene. As improvement to the stereo vision algorithms, multi-view stereo recovers dense structure from multiple viewpoints. Initially using the same principle as stereo algorithms, today's MVS algorithms are very different. The fact of varying viewpoints and often large image sets lead to significant changes in the algorithms. Known camera positions and parameters making the recovery of 3D structure equivalent to the correspondence problem for multiple views. Finding pixels in other images requires two main elements: efficiently finding potential pixel matches and a quality measure to assess these matches. As illustrated in Section 3.2, finding these matches only requires to search on the epipolar lines. The assessment measure, to evaluate the quality of pixel matches, is called photo-consistency measure [32].

### 3.4.1 Photo-consistency

Given a 3D point  $P$ , its photo-consistency  $C$  can be defined for each image pair  $i, j$  that visualize  $P$  as

$$C_{ij}(P) = \rho(I_i(\Omega(\pi_i(P))), I_j(\Omega(\pi_j(P))), \quad (3.14)$$

where  $\rho$  is a similarity measure, that compares the two projections of  $P$  onto the according image  $i, j$ . These projections are defined by the functions  $(\pi_i, \pi_j)$ ,  $\Omega$  describes its surrounding area.  $I_i, I_j$  denotes the intensity values. Equal to matching features, as introduced in Section 3.2, the region of an image pixel is used to recognize it on other images, ignoring changes in illumination and viewpoint. The choice of the region size is a critical factor for the reconstruction and not easy to achieve. This is due to the fact that uniqueness and invariance are counteractive. A larger area leads to a more unique appearance, while with an increasing size it gets harder to keep the illumination and viewpoint effects low. The simplest way to define  $\Omega$  is to use a square grid with constant size [32]. Figure 3.9 shows the concept of area function  $\omega$  of the photo-consistency

measure. In this example, the area of an image pixel is described by the 3x3 grid.

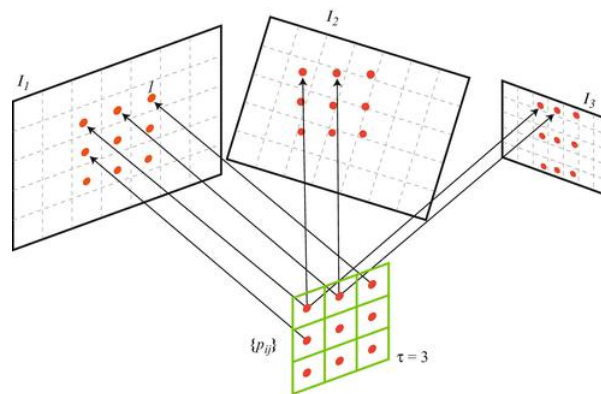


Figure 3.9: Concept of pixel matching. The surrounding area of a pixel and its intensity values are used to find pixel correspondences [41].

The definition above only holds if for each scene point, the viewing images are known. There are many different photo-consistency measures. In this work, two representatives are introduced, the Normalized Cross Correlation (NCC) and the Sum of Squared Differences (SSD). For easier representation, the photo-consistency function is simplified as a comparative function of two signals  $f, g$ . The NCC is a common choice and is mainly used in when a wide range of illuminations and materials appear, as it is invariant to changes in gain and bias. While leading to high accuracy, lack of texture is its main failing mode. SSD is the  $L^2$  squared distance between the input signals.

$$\begin{aligned} NCC : \rho_{NCC}(f, g) &= \frac{(f - \bar{f}) * (g - \bar{g})}{\sigma_f \sigma_g} \\ SSD : \rho_{SSD}(f, g) &= ||f - g||^2 \end{aligned} \quad (3.15)$$

Equation (3.15) shows the mathematical definitions of the NCC and SSD.  $\sigma_f, \sigma_g$  are the standard deviation of the input signals. Figure 3.10 displays the resulting functions for SSD and NCC measurement using a 3x3 kernel. On the left image a lot of texture is available near the red pixel. The right image is almost textureless. The correct depth is close to 0.5 depth units. Therefore, the function responses along the epipolar lines show that texture is the key factor when searching for pixel correspondences.

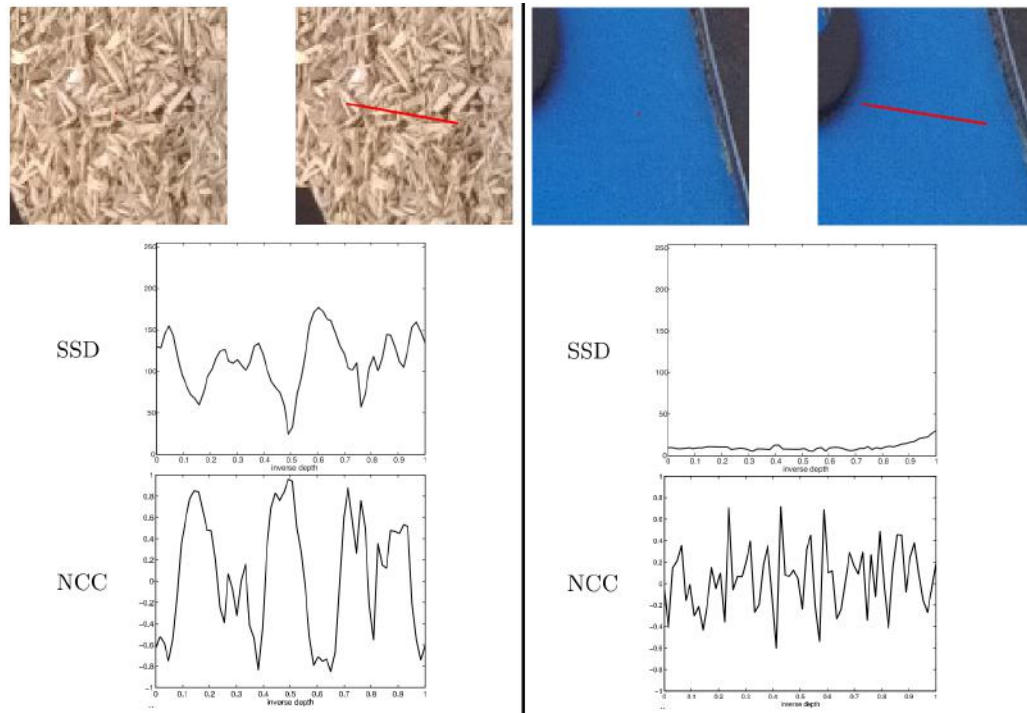


Figure 3.10: Function responses of SSD and NCC along the epipolar lines of a textured (left) and textureless (right) image using a  $3 \times 3$  squared region [32].

The photo-consistency functions are transformed, to normalize the different values to the same range, and filtered, as the function response is often noisy. The photo-consistency is a volumetric quantity. It can be either stored in a discretized 3D volume or list of pairs of photo-consistencies and positions.

### 3.4.2 3D Reconstruction

Based on various factors e.g. the photo-consistency measure or visibility computation, multiple algorithms were invented. These can be classified by the output scene representation they produce. The representation highly depends on the visualization application. Two major scene representations are discussed: depth maps and point clouds.

### Depth map reconstruction

Depth map reconstruction is a simple, but scalable way of representing a 3D scene. For each input image, a depth map is reconstructed. A depth map is a 2D array while its field value contain information about the distance between the viewpoint and the scene point. Figure 3.11 shows an example of a depth map of an input image. Depth maps can be easily computed considering neighbouring images and a reasonable photo-consistency measure and is even simpler than stereo-construction due to redundancies. Uniform depth sampling is crucial to achieve a high quality reconstruction and merging multiple depth maps into a global 3D model is the main problem of this representation. Therefore, this method is used for small baselines between camera locations [32]. Many depth map reconstruction algorithms have been proposed. [32] presents a good overview of existing methods. The Winner-Takes-All strategy is a simple approach that will be discussed below.

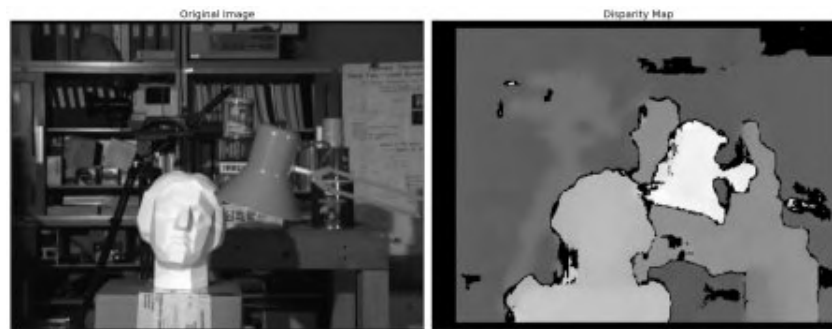


Figure 3.11: Disparity map example. Original image (left) and its disparity image (right)[42].

**Winner-Takes-All** This simple depth map reconstruction algorithm utilizes the photo-consistency function for each pixel of a reference image. As the name proposes, for each pixel only the maximum photo-consistency score determines the depth of the pixel. Additionally, a confidence measure is calculated from the photo-consistency function to further filter out low-confident depth values or to help in the merging process. Typically, the accuracy of depth map estimates is inversely proportional to the distance of the surface.

### Point cloud reconstruction

A point cloud is a single 3D model consisting of a set of points in space, where each point has a X,Y and Z coordinate. Figure 3.12 displays the example of

a point cloud representation of an object. The main difference in the reconstruction, compared to depth maps, is the consideration of spatial consistency assumptions as the region grows around single points. Consequently, reconstruction algorithms producing point clouds are often more complex and take more time and processing power. One famous technique are patch-based algorithms. In short, via the use of patches, the photo-consistency function is extended to the use of the surface normal besides the position of a pixel [32].

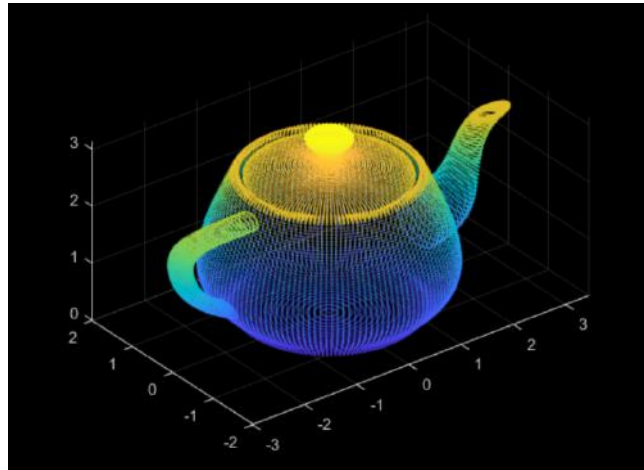


Figure 3.12: Point cloud example [43].

## 4 Methodological approach

The focus of this work is to make 3D inspection systems via an robotic arm applicable. For robust and fast reconstruct, the ICI of the AIT is utilized as introduced in Chapter 2. As already mentioned in the introduction, only perfect linear acquisition paths are supported by the framework. This guaranties pixel correspondences in the same pixel row in all images. Another condition is the uniform baseline between taken images. Through these requirements, the image path is similar to an ideal linear camera arrangement. The robots movement includes many vibrations and positional uncertainties due to an imperfect robot model. These inaccuracies include translations and rotations in the acquired images, harming the result of 3D reconstruction. Therefore, the acquired image sequence has to be transformed to fit the requirements of the ICI. Figure 4.1 shows a schematic of how the images taken via the robot system, seen on the upper image arrangement, have to be transformed to the uniformly aligned arrangement seen on the bottom.

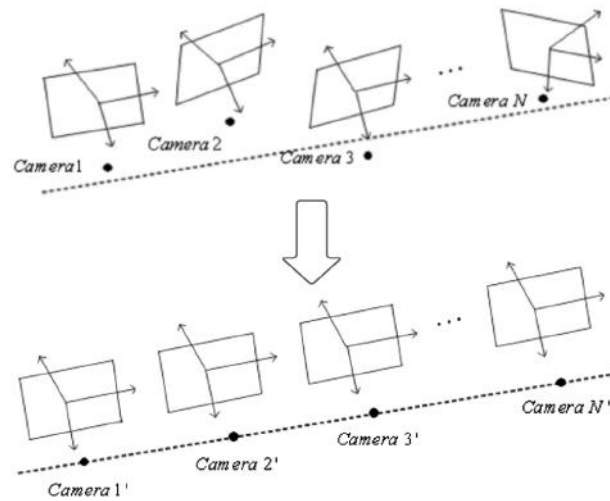


Figure 4.1: The inconsistent arrangement seen on top has to be transformed into a uniform arrangement to meet the requirements of the ICI [27].



To transform the acquired sequence to the ideal path, two different approaches are tested and evaluated. The first approach uses the camera position to transform each image. As the derived robot position is expected to not be accurate enough, the camera positions are optimized using image features. Since many MVS algorithms use the images and the camera position as input, it is presented, how the generic SfM pipeline can be improved in terms of robustness and speed via the additional data from the robot. The second approach calculates the homography via image features to sequentially map the images onto the initial image plane. Then feature tracks are used to keep a constant baseline between images. In conclusion, this work presents solutions to answer following research questions.

### Research Questions

1. How to rectify an image sequence with provided inaccurate pose information into an ideal arranged linear array?
  - Is the provided robot pose information sufficient to transform the image sequence?
  - Is an iterative method sufficient or is an optimization-based approach required?
  - How does the image transformations influence the outcome of the ICI.
2. How can the generic SfM pipeline be improved by the provided information of a robotic arm?

In the following section the two approaches are elucidated. The improved SfM pipeline is present in detail. Both methods are then tested on datasets recorded with a custom optical setups mounted on a robotic arm. In chapter 5, the results are evaluated regarding parallax and the final 3D reconstruction.

## 4.1 Rectification via camera position

The first method aims to rectify the image sequence by using refined camera positions. Therefore, the approach utilizes the SfM pipeline. Through the setup, prior, inaccurate information about the camera position of the images is available. These are used to advance the general SfM pipeline seen in Figure 3.6. The advanced pipeline, seen in Figure 4.2, includes following steps: For each image, features are extracted and matched with previous images as described



in Section 4.1.1. Then the feature correspondences are triangulated to form a Scene graph. Finally, a constrained BA is applied to optimize the camera locations and orientations. Next, the single steps of the pipeline are explained in detail.

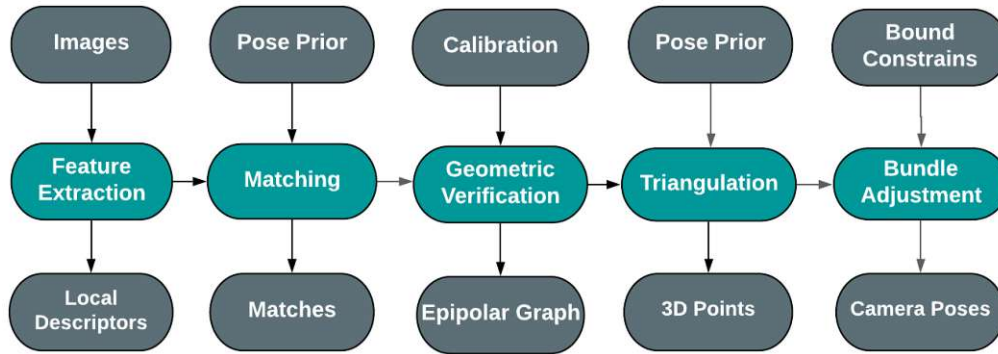


Figure 4.2: Improved SfM pipeline

### 4.1.1 Feature Extraction and Matching

In both approaches of this work, image features are extracted and matched against each other. Due to the fact that the viewpoint of neighbouring images changes are marginally and mainly occur in the single reference direction, the selection criteria is restricted to speed of computation. Hence, Oriented FAST and Rotated BRIEF (ORB) are the best choice of feature detector-descriptor for this work since they outperform other state-of-the-art methods according to Tareen and Saleem [44]. ORB features are also used in the popular ORB-SLAM that was introduced in Section 3. Tareen and Saleem [44] provide an in-depth comparison of multiple image features. They also conclude that ORB is one of the most efficient choices.

For feature matching the positional prior of the robotic system is combined with chosen maximum deviations to limit the search space for potential feature matches. Thus, the optical parameters have to be known. In an optimal camera arrangement a image pixel can be found on the same pixel row and a shift in the reference direction, called disparity. The disparity depends on the baseline and the points depth (see (3.7)). However, the depth of the pixel is unknown, as our goal is to reconstruct its depth, but it can be constrained with the parameters of the optical system. As all objects are expected to be in focus, the pixel depth is expected to lie between the near and far depth of field limit

$(D_N, D_F)$ . Therefore, the maximum and minimum disparity  $d_{min}, d_{max}$  can be calculated as

$$\begin{aligned} d_{min} &= \frac{B_x \cdot f}{D_F} \\ d_{max} &= \frac{B_x \cdot f}{D_N}. \end{aligned} \quad (4.1)$$

$B_x$  is the baseline in the reference direction  $x$  and can be computed from the two camera locations provided by the robotic system. Nevertheless, the camera arrangement is not optimal and the constrained disparity has to be extended with limitation of the positional accuracy. The maximum error is taken to be equal for all translational and equal for all rotational degrees of freedom and labelled as  $\epsilon_t[m], \epsilon_r[deg]$ . These maximum error values are expected to hold for all image pairs. To acquire the final search area for a feature location  $(x_p, y_p)$ , the search boundaries in pixel coordinates are calculated by

$$\begin{aligned} [x_{min}, x_{max}] &= [x_p - d_{min} + E(\epsilon_t, \epsilon_r), x_p - d_{max} - E(\epsilon_t, \epsilon_r)] \\ [y_{min}, y_{max}] &= [y_p - E(\epsilon_t, \epsilon_r), y_p + E(\epsilon_t, \epsilon_r)], \end{aligned} \quad (4.2)$$

with the x-axis as the reference direction. The function  $E$  outputs the error in pixel, taking the constant maximum positional errors of the translation and rotation as input.  $E$  is defined as

$$E(\epsilon_t, \epsilon_r) = \frac{\epsilon_t}{P_{RES}} + \frac{\tan(\epsilon_r \cdot \pi/180) \cdot D_F}{P_{RES}}, \quad (4.3)$$

where  $P_{RES}$  is the pixel resolution. The error function ignores the rotation and translation around the z-axis. Their values are expected to be negligible. For the translation, this is due to the fact that the distance of the camera to the object is much larger than the occurring error. The small baseline between images shrinks the rotation error around the z-axis to a minimum.

If there are several potential correspondences in the search area, the best match is chosen. For a more robust feature matching result, the correspondences are further filtered by applying a distance threshold and a RANSAC algorithm.

### 4.1.2 Geometric Verification and Triangulation

To further include outliers, the standard RANSAC algorithm is used. The image registration step is eliminated and the relative camera pose estimation is replaced by taking the positional information of the robotic arm. If the positions

of the images are unknown, the relative poses are received by estimating the essential matrix. As the essential matrix has a rank deficiency of rank 2, this works only up to a scale. Feature tracks are found for the overlapping images and multiview triangulation is performed using the DLT algorithm by Hartley and Zisserman [36].

### 4.1.3 Bound Constrained Bundle Adjustment

The main idea of utilizing the position information of the robotic arm with a SfM pipeline is still not sufficient as the conventional BA process is not scale dependant. Conventional BA sets parameters either as fixed or as unknown. This can lead to certain camera positions possibly undergo an excessive shift when converging into global or local minimum. Consequently, Bound constrained Bundle Adjustment (BCBA) is applied. As briefly mentioned in Section 2.3, BCBA by Gong et al. [24] is the first constrained BA for general use. Previous works were limited to geometric constrains, assuming perfect shapes, for special applications. Here we cannot use shape constrains as the scene is unknown. The camera locations and potentially other parameters are bounded by estimated maximum errors. Although the method was tested on medical application, the author proposes his work for general use, even mentioning industrial robots. In the following part, conventional BA algorithm is presented, since it was only introduced briefly in Section 3. Then, the BCBA algorithm is summarised. For further details see the original paper by Gong et al. [24].

BA is a non-linear, least square problem, minimizing the reprojection error between 3D points and their reprojection, using the projection matrix, onto an image. In equation (3.13) this is formulated using a mapping function  $\pi$ . This projection function  $\pi$  can be expressed as the projection matrix, or the camera intrinsic and extrinsic matrix. Therefore, the reprojection of a 3D point  $P_{ij} = [X_{ij}, Y_{ij}, Z_{ij}]^T$ , where  $i$  is the index of the point in a 3D point set and  $j$  denote the image, can be calculated by

$$\omega \begin{bmatrix} x_{ij} \\ y_{ij} \\ 1 \end{bmatrix} = K \begin{bmatrix} R_{ij} & t_{ij} \end{bmatrix} \begin{bmatrix} P_{ij} \\ 1 \end{bmatrix}, \quad (4.4)$$

where  $\omega = Z_{ij}$ . The error function is then defined as the least squares problem with the objective to

$$\text{minimize } f(s) = \frac{1}{2} r(s)^T r(s), \quad (4.5)$$

where  $r(s)$  is the distance between measured and reprojected points, the reprojection error. It depends on the input vector  $s$  that is composing  $3n$  feature points,  $6m$  camera locations and  $4$  camera intrinsic parameters.

To optimize our non-linear function  $f$ , the function can locally be approximated around our vector  $s$  by the quadratic Taylor series

$$f(s + \delta) = f(s) + g(s)^T \delta + \frac{1}{2} \delta^T H(s) \delta, \quad (4.6)$$

where  $g(s) = \frac{df}{ds}(s)$  is the gradient vector and  $H(s) = \frac{d^2f}{ds^2}(s)$  the Hessian matrix of  $f(s)$ . This approximation has a quadratic form with a unique global minimum that can be calculated.

The  $\delta$  that minimizes our approximation can be found by setting the derivative equal to zero and rearrangement to form a system of linear equations

$$g(s) + H(s)\delta = 0 \rightarrow H(s)\delta = -g(s) \quad (4.7)$$

This iterative process to find local minima is called Newton's method. The computation of the Hessian matrix is computationally demanding, therefore, in the Gauss-Newton method, the Hessian matrix is approximated by

$$\bar{H}(s) = J(s)^T J(s) \quad (4.8)$$

The Schur complement method is then used to reduce the size of the linear systems, splitting the linear equations into two smaller linear systems.

In the method of Gauss-Newton the rate of convergence is not controlled. If the second order approximation does not correspond well with the shape of the function, the step  $\delta$  of an iteration could be too large, leading to an overshoot and potentially worse results. The Levenberg-Marquart algorithm deals with this problem by effectively adapting the step size, making it the state-of-the-art solution to solve the BA problem. Additionally to the Gauss-Newton method, a damping factor  $\lambda$  is introduced which controls the step size and the direction. The damping factor is adjusted in every iteration. If it is low the method is close to the Gauss-Newton method, if  $\lambda$  is high the method is similar to method of gradient decent. Thus, the best of both methods are exploit, the fast convergence of the Gauss-Newton method and the slow, but guaranteed, convergence of the gradient decent.

Compared to the traditional BA problem, the BCBA problem can be defined as

$$\begin{aligned} \text{minimize } f(s) &= \frac{1}{2} r(s)^T r(s) \\ \text{subject to } l_i &\leq s_i \leq u_i. \end{aligned} \quad (4.9)$$

BCBA modifies the traditional approach in the following way to use bounds on the parameters. The parameter vector is projected onto a feasible set as

$$\text{proj}(s) = \min(\max(l, s), u), \quad (4.10)$$

with  $[l, u]$  as lower and upper bounds and an active set  $A(s)$  is defined as

$$A(s) = \left\{ i \left| \begin{array}{l} s_i = l_i \quad \text{and} \quad g_i < 0 \\ \text{or} \\ s_i = u_i \quad \text{and} \quad g_i < 0 \end{array} \right. \right\}. \quad (4.11)$$

The complementary set of  $A(s)$  is called the inactive set  $I(s)$ . Then the gradient projection method is used to solve the constrained optimization problem. The projected gradient  $\hat{g}_i$  is either  $g_i$  if  $i \in I(s)$ , or is 0 to keep the parameters at their bounds.

$$\hat{g}_i = \begin{cases} g_i, & i \in I(s) \\ 0, & \text{otherwise} \end{cases} \quad (4.12)$$

The second modification is done at the approximation of the Hessian matrix  $\bar{H}(s)$ . As the projected gradient of the parameters of the active set  $A(s)$  are zero, all corresponding rows and columns can also be set to zero forming the reduced Hessian matrix  $\hat{H}$  that is defined as

$$\hat{H}_{ij} = \begin{cases} \bar{H}_{ij}, & \text{if } i \in I(s) \text{ and } j \in I(s) \\ \bar{H}_{ii}, & \text{if } i \in A(s) \\ 0, & \text{otherwise} \end{cases} \quad (4.13)$$

The final BCBA is then described

$$\hat{B}\delta = -\hat{g}(s), \text{ with } \hat{B} = \hat{H}(s) + \lambda\hat{D} \quad (4.14)$$

, where  $\hat{D}$  is the diagonal matrix of  $\hat{H}(s)$ . To calculate  $\delta$ , the Schur complement is used and updates are only accepted if the error function  $f(s)$  is reduced.

In this work, the camera intrinsic parameters are fixed. This is due to the fact that the ICI pipeline includes a preceding, enhanced camera calibration to determine the camera parameters. Only the inaccurate camera extrinsic parameters, obtained from the robotic arm, are bounded. The determination of the boundaries is discussed in Section 4.1.5.

#### 4.1.4 SfM utilization and image rectification

Two strategies are tested for the presented SfM pipeline. The first approach is to use the global method. All images run through the reduced pipeline as

in Figure 4.2. After all images are registered, BCBA is applied for refinement. The second approach is to utilize an incremental method with local BCBA. This means only the last  $N$  registered images are refined in the BCBA step. This comes with the advantage that the refined images can immediately be transformed and forwarded into the ICI pipeline. Consequently, the ICI does not get slowed down by the image processing.

After the refinement of the camera positions, the images are rectified into a common image plane. The common image plane is equivalent to the first camera's image plane. Therefore, all images  $I_N$  are transformed onto the plane of image  $I_1$  by retrieving the homography by the corresponding camera displacement. The homography  $H_{k \rightarrow 1}$  between the first image  $I_1$  and an image  $I_k$  can be retrieved from their camera location by

$$H_{k \rightarrow 1} = R_{k \rightarrow 1} - \frac{t_{k \rightarrow 1} \cdot n^T}{d}, \quad (4.15)$$

where  $n$  is the plane normal vector of camera  $I_k$ , that is computed using the rotation matrix  $R_k$  and the distance to the image plane  $d$ .  $d$  can be retrieved by the dot product between the plane normal vector and a point on the plane. The camera displacement components,  $R_{k \rightarrow 1}$  and  $t_{k \rightarrow 1}$  from camera position of image  $I_k$  to the ideal camera position of the first image  $I_1$ , can be calculated by

$$\begin{aligned} R_{k \rightarrow 1} &= R_1 \cdot R_k^T \\ t_{k \rightarrow 1} &= R_1 \cdot (-R_k^T \cdot t_k) + t_{1 \rightarrow k}. \end{aligned} \quad (4.16)$$

Here,  $t_{1 \rightarrow k}$  presents the ideal position of camera  $k$  by adding the intended displacement to the reference direction. The final projective homography  $G_{k \rightarrow 1}$  is then computed via the camera intrinsic matrix

$$G_{k \rightarrow 1} = K \cdot H_{k \rightarrow 1} \cdot K^{-1}. \quad (4.17)$$

Using the homography, image pixels are finally transformed via (3.12). For the global approach, the mean image plane is used instead of the plane of the initial image.

#### 4.1.5 Boundary determination and mounting effects

Determining the boundaries is similar to Section 4.1.1, but considered over multiple images. Restricting the camera locations decreases the chance of getting stuck in local minima in the optimization process. This prevents over-optimization by stopping the process at the boundaries. Additionally, this

reduces the runtime of the bundle adjustment. Furthermore, setting reasonable boundaries avoids heavily deviating camera locations.

To finding well suiting boundaries for the camera locations, several factors have to be taken in mind:

- The robots accuracy
- The mounting of the camera
- The type of bundle adjustment

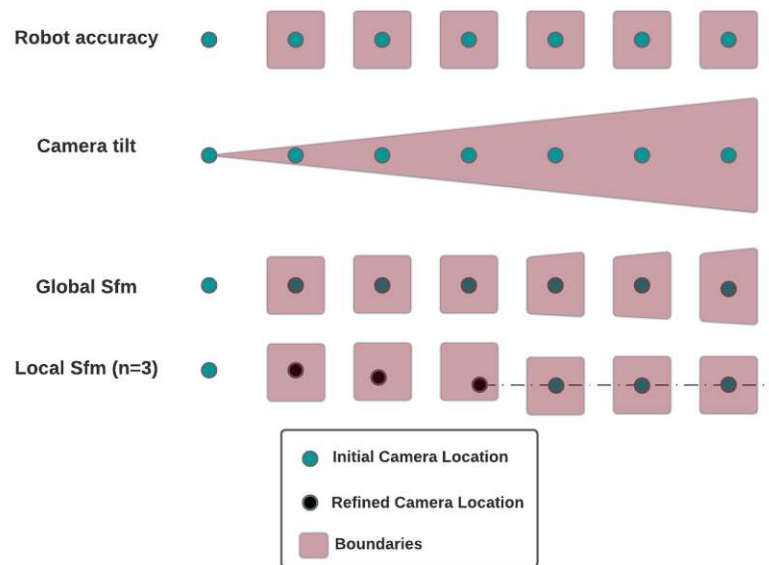


Figure 4.3: Schematic influence of different factors on the boundaries.

Figure 4.3 displays the different error types and how they influence the tolerance range. The red areas show the tolerance areas in which the cameras are located. Manufacturers often list the repeatable accuracy of the robotic arm, rather than the actual accuracy. The real accuracy depends on many factors and changes over time. Therefore, it is almost impossible to accessed precise values. However, it is possible to restrict the positional error to a tolerance range. The restricted area is constant for each position the robot is navigation to.

Mounting the camera to the robot is error prone. The main error that occurs is a tilt of the camera. Moving a tilted camera leads to a vertical shift in the



images, influencing the bounds in the x-y direction. This problem is also faced in original application with a fixed camera on the ICI setup. In the static case, this can be compensated by a custom calibration process [45]. Here, a constant transport vector is determined which reflects the camera tilt. The vector is then feed forward to the pixel matcher. A single transport vector would not be sufficient for a setup including a robotic arm. Due to other error effects, the transport vector deviates between images.

The error of the camera tilt can simply be calculated via trigonometry using the distance in the main direction of movement and a fixed maximum error angle.

The type of SfM defines how these errors are utilized to set boundaries for each image. In the global process, the boundaries have to be set beforehand. The robot accuracy stays constant for each image, but the influence of camera tilt increases with distance to the initial image. At some distance, the camera tilt error extends the error of the robot. In the local SfM the initial camera locations are reset to the last refined location in all degrees of freedom except the main direction of movement. If the refinement size is small enough, the camera tilt has no influence on the boundaries.

## 4.2 Rectification via perspective transform

A very intuitive solution to the problem of image sequence rectification is to sequentially estimate a perspective transformation between images and apply these transformation to rectify the images into a common image plane. The epipolar lines will then be parallel and on the same pixel row. Unfortunately, the disparity will still vary as movement in the reference direction is not constant. Therefore, features are tracked over multiple consecutive images to keep the shift between images constant. Figure 4.4 shows the processing steps that are applied for incoming images. First features are extracted and matched against the previous transformed image. Feature tracks are created if the feature of the previous image was matched before. Then a homography is estimated and used to transform the image. Finally, the feature tracks are used to transform the image again to insure a constant disparity between images.

### 4.2.1 Homography estimation

As introduced in Section 3.2.3, a homography matrix describes the relationship between two images viewing a planar surface. Using a homographic transform possibly leads to a sufficient result, as the scanned structures have a very low



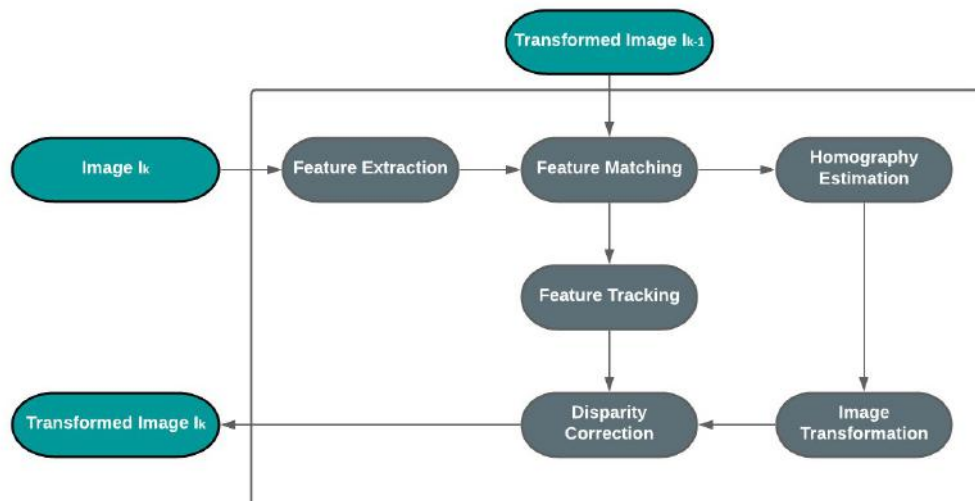


Figure 4.4: Processing steps for an incoming image  $I_k$ .

depth compared to the working distance of the optical system. Additionally, the scene is often planar, e.g. small objects are placed on a ground plane, and even through the motion has 6 DoF, all, except the reference direction, has low deviation. The homography matrix is extracted using the DLT algorithm and the images are then transformed using (3.12). To exclude the shift in the reference coordinate that occurs because of the baseline, the last column of the according axis  $h_{13}$  or  $h_{23}$  is set to zero.

### 4.2.2 Feature Tracking and Disparity Correction

To track features over multiple images, the features of image  $I_k$  that matched with feature of image  $I_{k-1}$  are associated with matched features of image  $I_{k-1}$  and  $I_{k-2}$ . When matching with the transformed image  $I_{k-1}$  the features locations are transformed as well, but the feature descriptors do not get recalculated. These transformed feature locations are used to estimate the homography between the newly added image  $I_k$  and the previous transformed image  $I_{k-1}$ .

The feature tracks are then used to keep the same disparity of features over multiple images. Therefore, the image is shifted in the reference direction to reduce the total disparity difference between all features tracks  $T$  that includes matches of the preceding two images  $I_{k-2}, I_{k-1}$  and the current images  $I_{k-1}, I_k$ . Mathematically, the error function is defined as

$$f(T) = \sum_{t \in \text{Tracks}(I_{k-2}, I_{k-1}, I_k)} d(t_{k-2, k-1}) - d(t_{k-1, k}), \quad (4.18)$$

where  $d$  is the disparity of a track  $t$  between two images. The minimizing value is then used to shift the image to the same disparity and the feature locations are updated again. Figure 4.5 shows the feature tracks over multiple images.

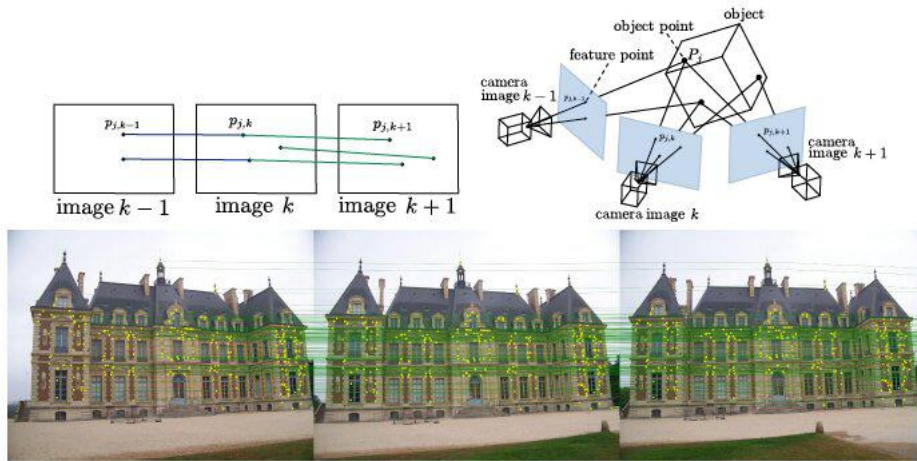


Figure 4.5: Feature tracks [46].

## 5 Experiments

To evaluate the introduced approaches of the previous chapter, an experimental setup was built to capture image sequences and associated camera pose information using a robotic arm. The purpose of the experiments is, on the one hand, to identify issues when operating an eye-in-hand system to move a straight trajectory. On the other hand, to characterize the strengths and weaknesses of the methods for rectification and reconstruction. First, the improved SfM pipeline is tested. The camera positions from the robot and the refined positions are compared to captured positions via an Optitrack system. Then, the quality of the rectification process is evaluated to see if corresponding images pixel lie on the same row. Finally, the 3D reconstruction of the ICI using the rectified images is compared to records taken from a linear stage.

### 5.1 Data Acquisition

For the acquisition of test data for the rectification process and the subsequent dense 3D reconstruction via the ICI, an eye-in-hand system was implemented. For the robotic arm, a Kuka LBR iiwa was used. It features seven rotational joints with a precision of  $\pm 0.1$  mm and a maximum loading capacity of 14 kg with the included control unit provided by the manufacturer. As camera, the Basler acA2440-75um was used. It contains a Sony IMX250 CMOS sensor with an resolution of 2448x2048 pixel (5 MP) at maximum 75 frames per second. For image acquisition, it supports both triggering over software or hardware and it features a global shutter. These specifications make it very suitable for experiments including motion. Compared to rolling shutter, global shutter exposes and captures every pixel at the same time, eliminating the effect of rolling shutter in dynamic scenes, e.g. wobble, skew and aliasing. The hardware trigger is necessary to run on the AIT's ICI operating on a linear stage while the software trigger is used when running on the robotic arm setup. For illumination, an YK-B144T ring lamp including 144 LEDs with a power of 4.5W with adjustable light intensity was utilized. Both components were mounted to the robotic arm using a custom designed, 3D printed part. Figure 5.1 shows the experimental setup. Three different objects were scanned (see Table 5.1). The first object features a flat surface with a random pattern

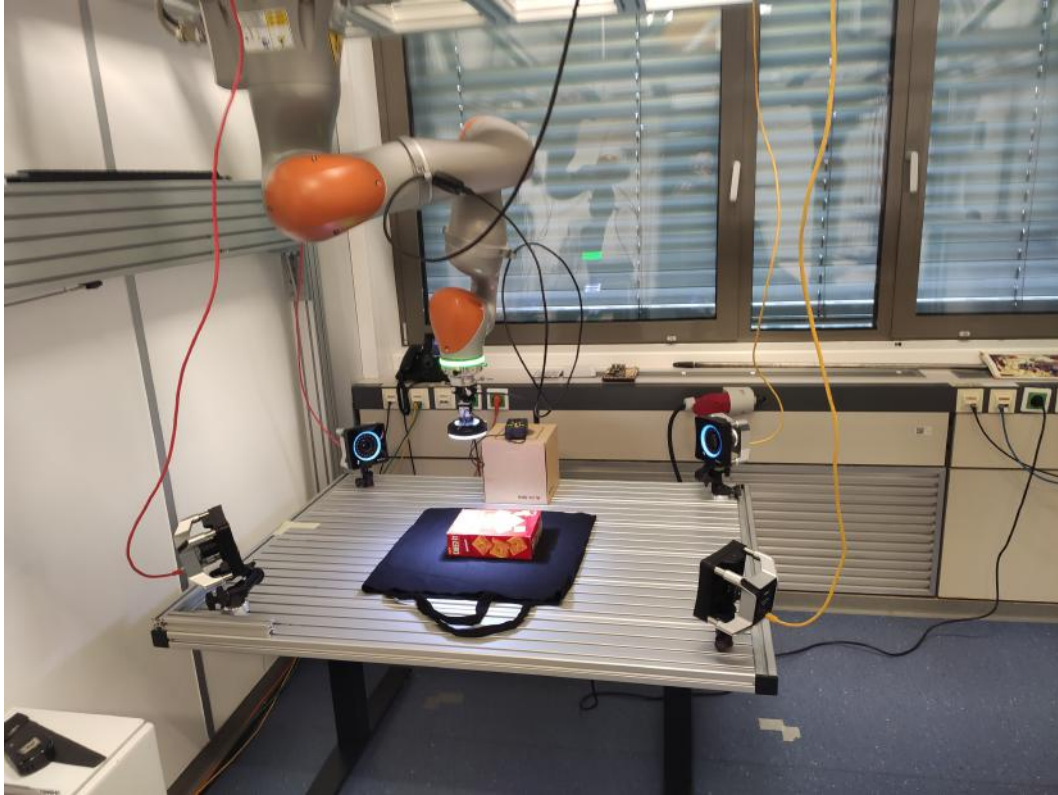


Figure 5.1: Full experimental setup.

printed on it. It provides very good texture over the hole area making it easy to find correspondences between images. The Cornflakes object increases the difficulty by having low textured areas, but it keeps the property of an almost flat surface. The hardest difficulty presents the board object. It features a lot of detailed structures, like coins and a printed circuit board which lead to several levels of depth and texture.

### 5.1.1 Robot Control

For each object, a datasets in two different modes was acquired, namely "Step Mode" and "Velocity Mode" (see Table 5.2). When recording in "Step Mode", the robot was navigating to discrete positions in a linear path. The robot stopped at these positions and an image and the positional information of the robots model was saved. Three procedures with the distances of 3mm, 1.5mm and 1mm between images were performed, leading to sets of 101, 201, and 301

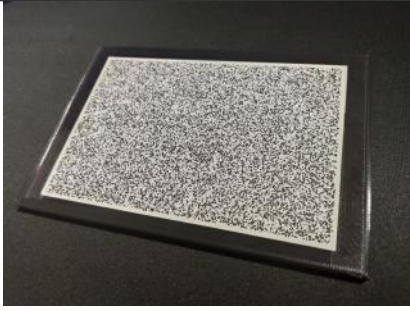


Recorded Objects	
Random Pattern	
Cornflakes	
Board	

Table 5.1: The recorded objects, each featuring a different level of difficulty.

images with positional information of the robot model. The path between the discrete positions in the linear path were precalculated using the robot model to get movement with minimal motion of the seven axis of the robotic arm. The "Velocity Mode" drives a linear path with constant velocity. It uses part-wise linearization to move in a linear motion with a speed of 40, 60 and 120 mm/s. Images were recorded constantly at 40 frames per second. The position information from the robot and the image frames were synchronized using timestamps of the Robot Operating System (ROS). Through testing on a setup using a linear stage, only the largest datasets of both modes are used in the evaluation, as they produce higher quality outputs.

For comparison, the pose was additionally recorded via an external Opti-

Track motion capture system. The OptiTrack motion capture systems works by triangulating positions of spheres between multiple cameras. The spheres are illuminated using infrared light for better segmentation. They were mounted via custom 3D print on the robotic arm. The OptiTrack did record the positions with a mean error smaller then 0.4mm.

Record settings			
Step Mode		Velocity Mode	
Step distance [mm]	Number of Images	Velocity [mm/s]	Number of Images
1	301	40	~ 300
1.5	201	60	~ 200
3	101	120	~ 100

Table 5.2: Record settings

TwinCat and OptiTracks Motion capture software were both running on the same PC. However, all information were redirected to a Linux PC via Ethernet running ROS. The camera was directly connect via an extended USB 3.0 cable and the images were accessed by the ROS node provided by Basler. To control and read data from the robot, a custom ROS node was utilized. Figure 5.2 provides an overview how the different data were redirected and collected. All the information was then exported into a rosbag file for further computation.



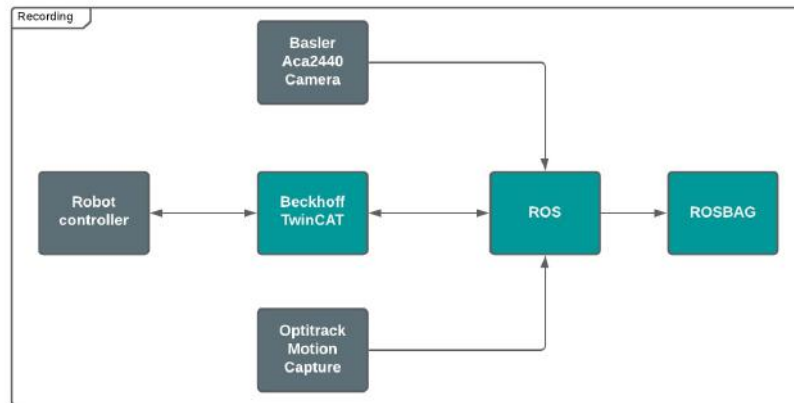


Figure 5.2: Block diagram of the KUKA LBR iiwa system.

### 5.1.2 Optical settings

A lens by Schneider-Kreuznach with a focal length of 12 mm was mounted via C-mount to the Basler camera. At a lens aperture of approximately 5.6 and a sampling object space of 0.05 mm/pixel the working distance of about 185 mm was calculate. Further the range was calculate to be about 38 mm and the lateral resolution 109  $\mu\text{m}$ . The read out of the camera was done via USB 3.0 connection. The additional USB expansion cable reduced the read-out performance from max 75 to max 40 frames per second in bayer\_rggb8 mode. The illumination YK-B144T was set to minimum power. To obtain the camera intrinsic parameters, a calibration with the camera\_calibration node that comes with ROS was performed using a chessboard pattern [47]. The rectification and reconstruction process were performed on a generic Notebook with an i5-8365U CPU running Ubuntu 18.04 LTS, reinforced with an external graphics card Nvidia 1080 TI.



Figure 5.3: Optical system

## 5.2 Inspection of the trajectory

The main approach of this work is to transform the acquired images via its pose information into a perfect linear camera arrangement. In this section we take a closer look at this information. On the one hand we want to observe, how the robot classifies his own movement and, on the other hand, how the global optimization processes changes the image poses to reduce the reprojection error. To compare the different trajectories, an Optitrack system was running along acquiring the dataset. Even though, the positional data of the Optitrack System is still uncertain, the accuracy can be narrowed down to 0.4 mm according to the calibration process. For synchronising the pose information, timestamps of the ROS are used.

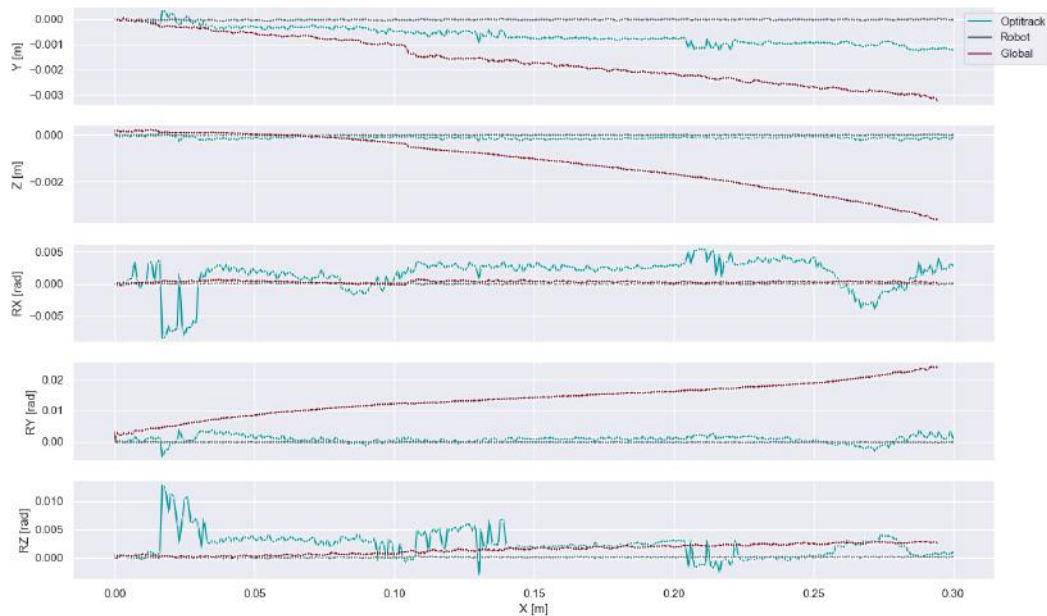


Figure 5.4: Pose information scanning the object Random Pattern in Step Mode (1mm).

Figure 5.4 show the different axis and angles in contrast to the main axis of movement  $X$ . When inspecting the robot pose of the robot model (Robot) it can be seen that it deviates from an ideal arrangement only in the range of micrometres. According to the robot model, the robotic arm moves highly accurate. However, when comparing the pose information to the Optitrack system, is shows a systematic shift in the  $Y$ -direction. The further away the



image is taken from the origin, the more shift of the pose is introduced.

The global SfM approach presents, how the image data changes the image poses. It introduces an increased shift in the  $Y$  direction that can be explained by a rotation in the camera mount described in Section 4.1.5. It also features a shift in the  $Z$  direction. A possible reason could be a tilted ground plane.

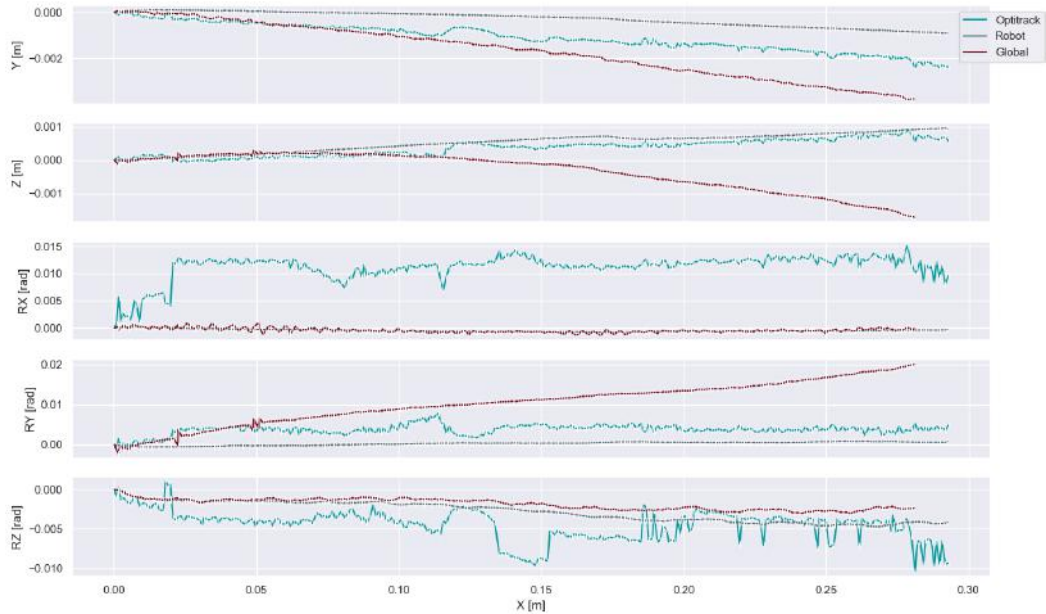


Figure 5.5: Pose information scanning the object Random Pattern in Velocity Mode (40mm/s)

The pose information in the Velocity Mode gives an similar outcome. As seen in Figure 5.5 the same effects in the global SfM approach occur. Main difference be that the robot model does not assume a perfect path. Even though, it is still off up to 2 mm, compared to the Optitrack system.

### 5.2.1 Local Bundle Adjustment Approach

Although, popular vSLAM systems do not support the input of pose priors, we tested several SLAM systems on our dataset. Unfortunately, the resulting trajectories were unusable or the SLAM system would not accept our image sequence e.g in the case of ORB-SLAM.

When inspecting the trajectory processed with out local SfM pipeline, multiple issues occur. As seen in Figure 5.6 several window sizes are tested on the Object

Random Pattern in Step mode. The window size refers to the amount of last camera poses that get optimized in a single processing step. Three different window sizes were tested: 3 (red), 11 (turquoise) and 21 (gray). Compared to the trajectory of the global SfM, consecutive poses deviate tremendously leading to an inconsistent zigzag pattern. This effect increase with the distance to the origin.

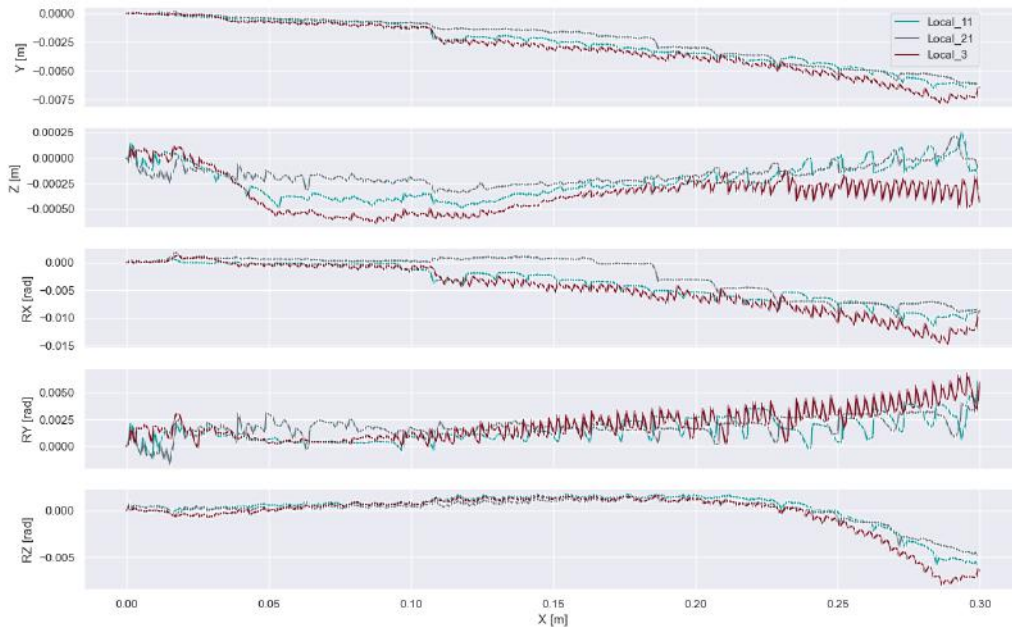


Figure 5.6: Pose information of local SfM approach using different window sizes.

As the resulting trajectory is corrupted, the approach is discarded and not further included in the evaluation.

## 5.3 Rectification Evaluation

To use the acquired images with the ICI, corresponding pixels are required to appear on the same pixel row. Therefore, the images are transformed using different approaches introduced in Chapter 4. To evaluate the quality of the rectification process, SIFT features are utilized. The features are extracted for all images and matched using the matching approach introduced in Section 4.1.1. Then, feature tracks are found over multiple subsequent images. Only tracks that are found in over 8 successive images are considered. Figure 5.7 shows

an example of the found feature tracks of a dataset. Feature tracks should appear as straight lines and the spacing between single feature points should be uniform. To evaluate the rectification, the change of location of SIFT features is used. First, the mean parallax of all features pixel locations between two successive images is calculated. This is done in vertical and horizontal direction. Also, the variance of these values are calculated to evaluate the rectification. The mean parallax and variance between two images are then averaged over all image pairs and used as quality measure.



Figure 5.7: Synthetic image with found feature tracks longer than 20 subsequent images. Ideally, feature tracks appear straight with uniform spacing between single feature points.

The results of the rectification evaluation are shown in Table 5.3. The different approaches, using the homography and the rectification via the globally refined pose information were tested against the three datasets in both modes. For comparison, images acquired using a linear stage were also evaluated. As expected, the linear stage performs best with the lowest vertical and horizontal parallax. The linear stage reaches sub-pixel accuracy in both cases with values under 0.07 in vertical and under 0.13 in the horizontal direction. Our approaches reduced the vertical parallax in all cases. The homography approach even reduced the vertical variance of the parallax down to similar values as the linear stage, under 0.1 in all cases. The horizontal parallax was significantly increased in the Homography approach.

Variance: $vertical[px^2] / horizontal[px^2]$				
Step Mode, 1mm	Linear stage	Original	Homography	Global
Random Pattern	0.03 / 0.10	0.94 / 1.10	0.01/5.15	0.32 / 0.90
Cornflakes	0.07 / 0.12	1.24 / 1.41	0.09 / 6.43	0.90 / 0.94
Board	0.03 / 0.13	1.25 / 1.34	0.05 / 5.82	0.58 / 2.83
Velocity Mode, 40 mm/s	Linear stage	Original	Homography	Global
Random Pattern	0.03 / 0.10	2.21 / 0.53	0.01 / 2.23	1.53 / 2.55
Cornflakes	0.07 / 0.12	1.96 / 0.62	0.09 / 1.35	1.79 / 0.70
Board	0.03 / 0.13	2.46 / 0.53	0.05 / 1.39	1.82 / 2.27

Table 5.3: Results of the evaluation processes. SIFT feature tracks are found, and their mean change in location are calculated. Then the variance is used to evaluate the rectification process vertically and horizontally.

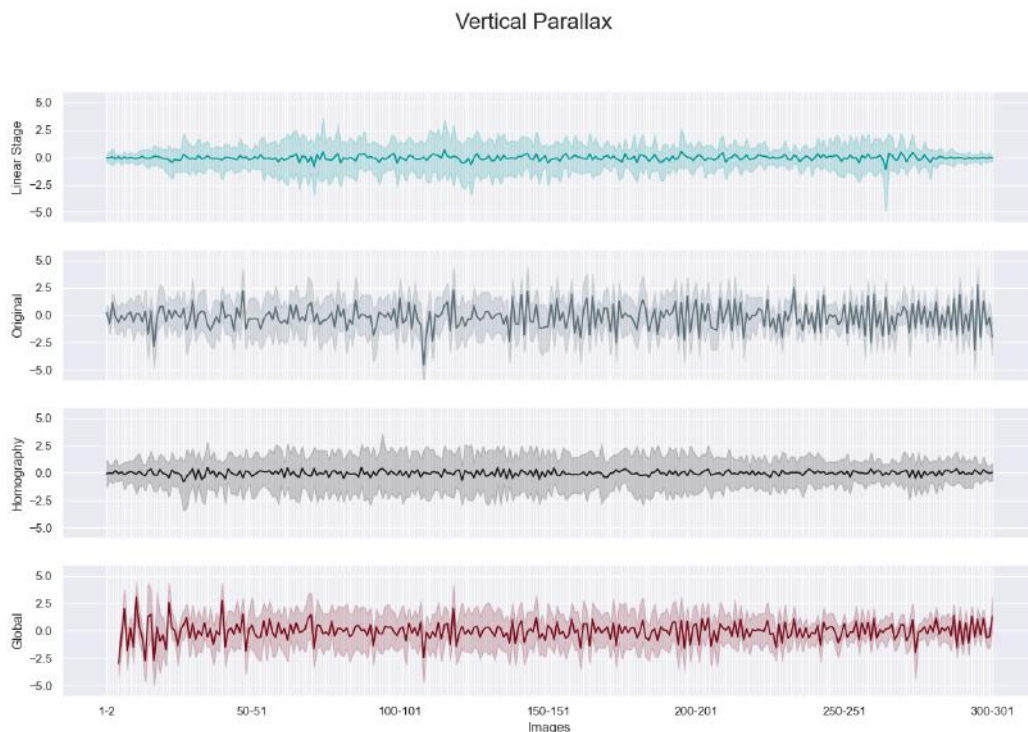


Figure 5.8: Mean vertical parallax between image pairs of the dataset board in mode Step.

Figure 5.8 and Figure 5.9 display the mean parallax between each image pair. The outer lines show the minimum and maximum parallax of a feature correspondence. In the vertical case, it can be seen that every method has

outliers of over 2.5 pixel. The mean value of the linear stage and homography is more consistent than the original sequence and global approach.

The horizontal parallax reveals major problems of our approaches at the starting section, where only a low amount of features is found and feature of the background influence the result. It is further visible, that the linear stage did use another setup with a different disparity.

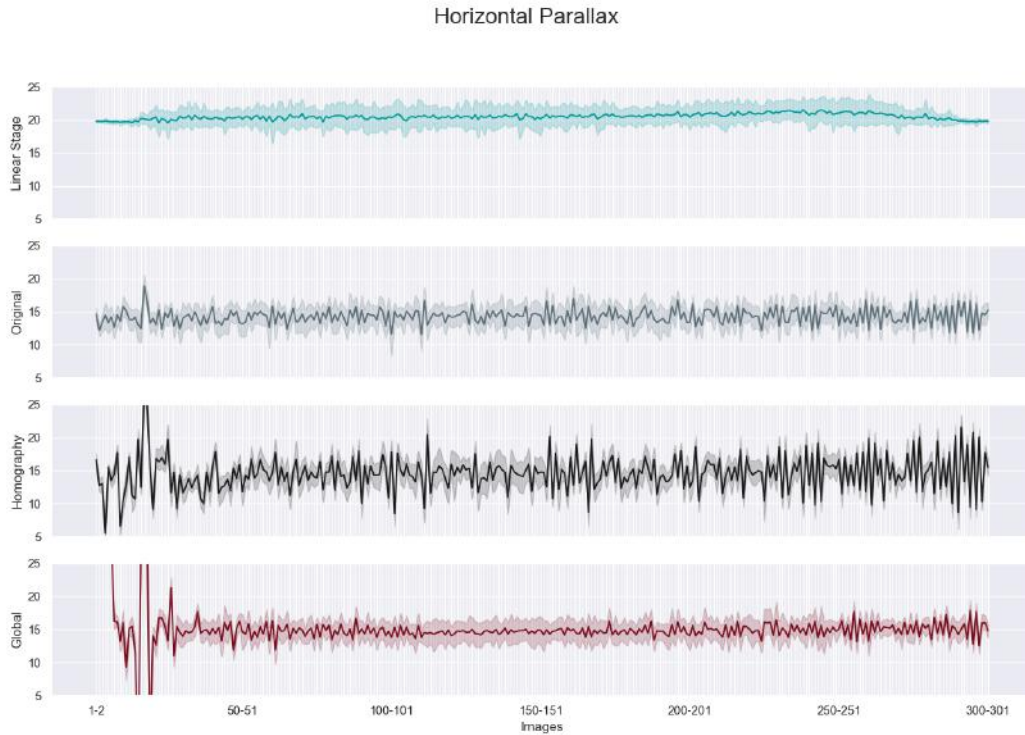


Figure 5.9: Mean horizontal parallax between image pairs of the dataset board in mode Step.

Although, these values give an overall presentation of the parallax, they do not display global shifts as they only compare image pairs. These phenomena can be seen in Figure 5.10. While the variance of the vertical parallax of the original images were higher compared to the Global approach, the difference is barely visible in the previous Figure 5.8. Following a feature track, it is clearly visible that, on the original images, the feature is not evenly distributed. The feature point is constantly shifting downwards, the more the further away from the origin. The other methods did not show this behaviors, they have a uniform distribution around the starting value of the track.



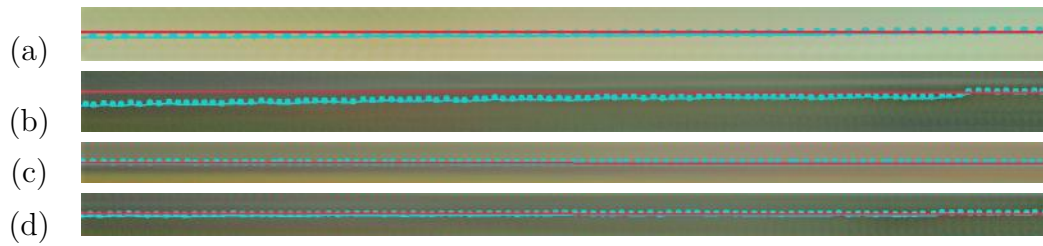


Figure 5.10: (a) Linear Stage (b) Original (c) Homography (d) Global Synthetic image of a single feature trace of the dataset board.

## 5.4 Reconstruction Evaluation

In the following section, the 3D reconstructions are compared. Therefore, the rectified image sequences of the different approaches are processed with the ICI. The reconstructions are present as point clouds (.ply). To have an external comparison, the reconstruction from the open-source software Meshroom is added to the evaluation. The standard Meshroom pipeline was feed with the original images, retrieving the unfiltered point cloud. The resulting point clouds are then compared to ground truth. As ground truth serves a reconstruction acquired at a test setup in the laboratories of the AIT. The test setup uses a linear stage with four illuminations (see Figure 1.1). It features a resolution of  $50 \mu\text{m}$ , a large increase to our tested setup, to get an improved reconstruction. This comes with the advantage that the outcome is directly comparable with the initial setup of the ICI. The coloured point clouds are displayed in Figure 5.11. For evaluation, the 3D point processing software CloudCompare is utilized. First, the background is cut out of the point clouds and the object is trimmed manually to the same field of view as the ground truth. Then, the point clouds are scaled and aligned manually using point pairs. Further, the Iterative Closest Point (ICP) algorithm is used to finely register the point clouds. Finally, the distance and standard deviation between the entities is calculated. Due to the scaling both quality measures are without unit. Additionally, the amount of points are compared to relate the denseness of the reconstructions.

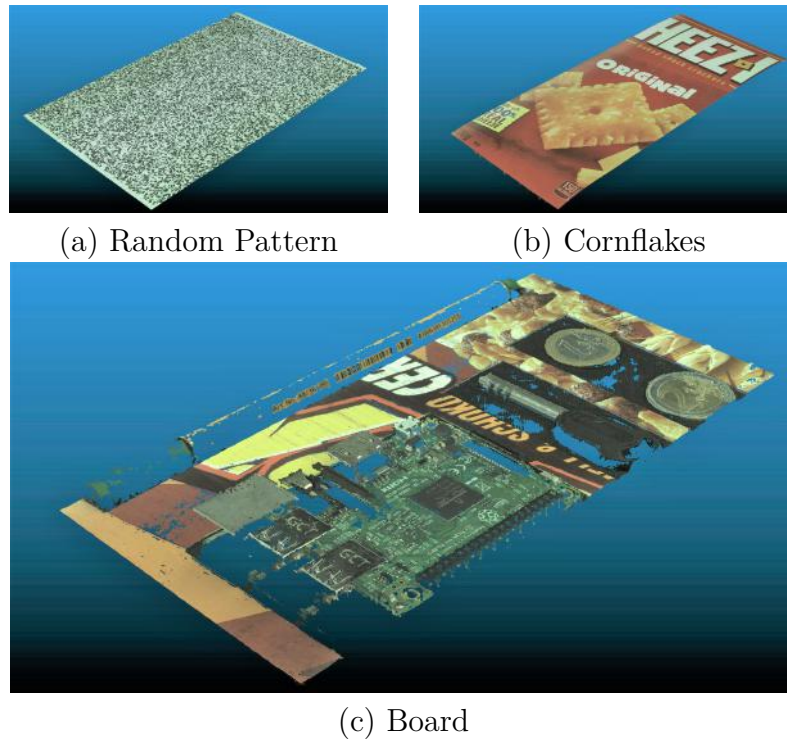


Figure 5.11: Ground truth of evaluated objects.

Accuracy [Mean distance / Std deviation]				
Step Mode, 1mm	Original	Homography	Global	Meshroom
Random Pattern	0.38 / 0.31	0.19 / 0.14	0.50 / 0.14	1.24 / 1.02
Cornflakes	0.54 / 0.58	4.13 / 3.34	0.17 / 0.17	0.48 / 0.47
Board	0.65 / 0.68	0.53 / 0.82	0.59 / 0.57	0.91 / 0.91
Velocity Mode, 40 mm/s	Original	Homography	Global	Meshroom
Random Pattern	0.41 / 0.34	0.19 / 0.14	0.38 / 0.23	1.01 / 0.86
Cornflakes	0.75 / 1.16	1.75 / 2.87	0.22 / 0.18	0.44 / 0.41
Board	0.82 / 0.77	0.65 / 0.66	0.46 / 0.51	0.94 / 0.94
Coverage [Points]				
Step Mode, 1mm	Original	Homography	Global	Meshroom
Random Pattern	2.737.605	2.739.937	2.726.209	260.424
Cornflakes	3.687.538	3.314982	3.686.828	251.371
Board	2.855.022	3.082.777	2.907.219	439.163
Velocity Mode, 40 mm/s	Original	Homography	Global Pose	Meshroom
Random Pattern	2.737.235	2.738.275	2.528.660	252.020
Cornflakes	3.688.412	3.498.901	3.602.965	232.303
Board	2.717.000	2.745.679	2.912.540	362.773

Table 5.4: Accuracy and Coverage of the different approaches.

The resulting mean distance and standard deviation of the different point clouds produced by the ICI are displayed in Table 5.4. The results show significant improvements, feeding the processed image sequences of our approaches, on all datasets in all modes over the original images. The homography approach seems to fail on the dataset Cornflakes as features of the background are found which increase the depth of the scene. With almost 10 times the amount of points, the ICI produces denser point clouds compared to Meshroom.

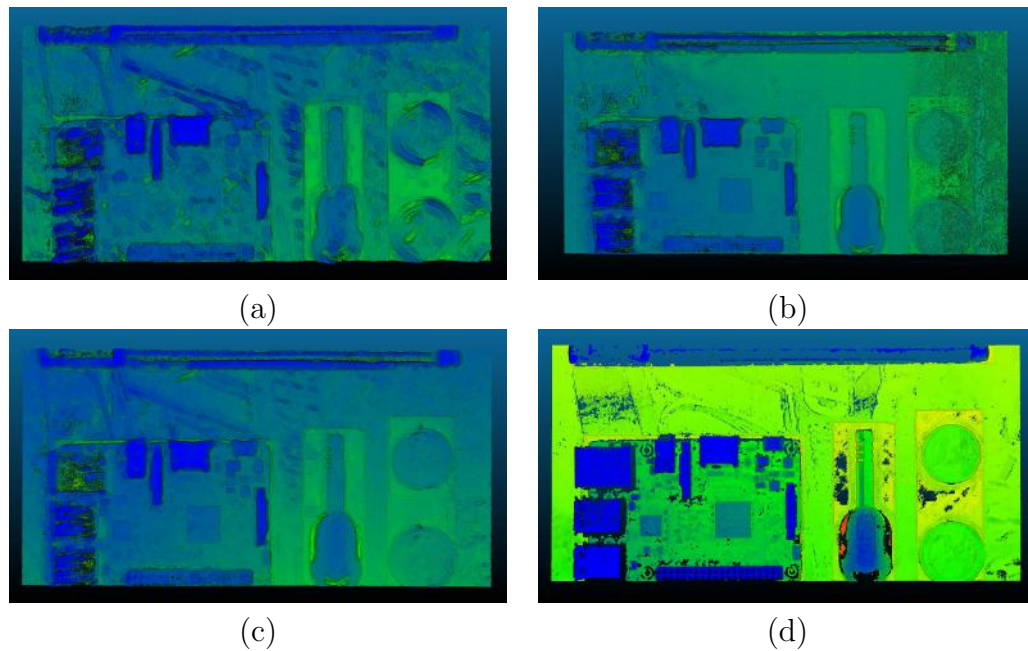


Figure 5.12: (a) Original (b) Homography (c) Global (d) GT

Top view of the reconstruction of dataset board in Step mode.

Figure 5.12 shows a top view of the Board reconstructions. Our approaches appear much cleaner on smooth surfaces while details are better preserved. The reconstruction via the original images have a overall bumpy look introducing wavy regions e.g on the edges of the coins. Both approaches did struggle to reconstruct the metal housings of the connectors on the PCB. Also, the pen was hardly reconstructed, even by the Ground Truth, as it barely features any texture. A more detailed view is presented in Figure 5.13. While the ICI produces an unsatisfying result with the original images, our approaches show a clean reconstruction with visible details. Although, the Ground Truth features a denser reconstruction, missing details are only due to restrictions in resolution. Resulting point clouds can be seen in Section 7.



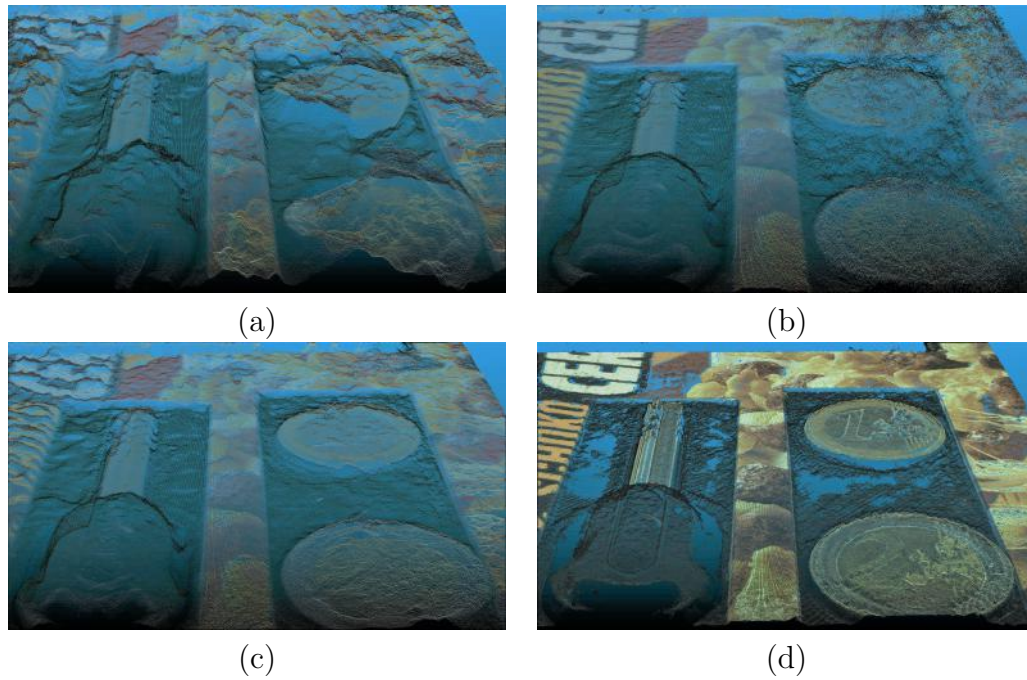


Figure 5.13: (a) Original (b) Homography (c) Global (d) GT  
 In detail comparison of the key and coins on the object board.

### Comparison Step Mode vs Velocity Mode

The Velocity mode is a fast method for image acquisition. Therefore, it is preferred over the Step mode. According to the mean distance and standard deviation, the approaches showed similar results for both recording modes. The original images are slightly worse in case of the Velocity mode. Arguable, this is due to even more drift in the trajectory. Figure 5.14 shows the reconstructed cornflakes with the global approach via both recording modes. Both results show a similar optical quality.

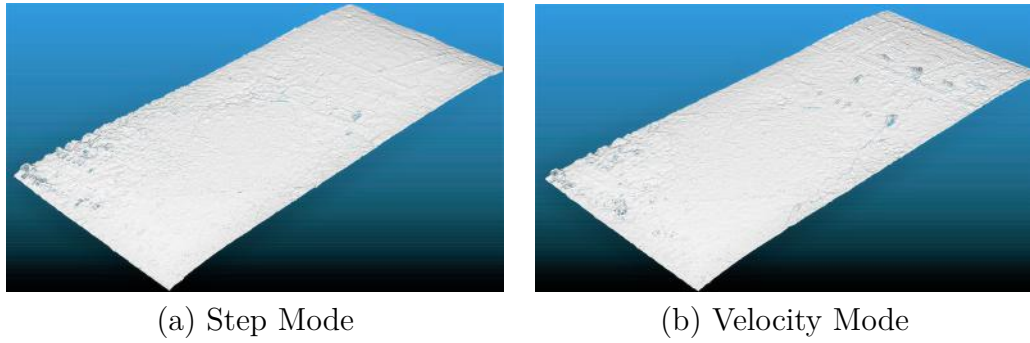


Figure 5.14: Comparison of the Cornflakes dataset in both acquisition modes using the Global approach.

## 6 Conclusion and Future Work

This final chapter summarizes the taken approaches and the subsequent evaluation of the rectification and reconstruction. The most important findings are mentioned and further improvements of the approaches are discussed.

### 6.1 Conclusion

The aim of this thesis is to apply the ICI on a robotic arm to form a flexible and fast 3D reconstruction method. Due to the requirements of the ICI, the acquired image sequence has to be taken in a linear motion, as pixel correspondences are expected to lie in the same row, with equal disparity between images. As expected, utilizing the acquired image sequence combined with the raw positional information of the robots leads to an unsatisfying result. To align the epipolar lines for the acquire image sequences, two approaches are introduced. The first approach features a simplified SfM pipeline to optimize the pose information. The optimized image poses are then used to rectify the images. The second approach directly determines a homography between images. It transforms the images sequentially including a disparity correction procedure.

Even though, the outcome always depends on several factors, like the quality of the robot model, this work shows a significant improvement when applying the ICI on a robotic arm. First the resulting rectification was evaluated using SIFT feature tracks. A clear improvement regarding the vertical parallax was noticed. The Homography approach reached similar values comparable to a linear stage. However, the horizontal parallax was increased. Both methods managed to remove global inconsistencies, like a global drift visible in the original image sequence. Finally, the reconstruction were evaluated utilizing the ground truth recorded via a linear stage with higher resolution. While the quality of the Homography approach is decreasing with the level of depth on the object, the Global approach produced consistent results in all cases. Again, both approaches show a clear improvement compared to the original images. It can also be concluded that acquisition mode barely effects the outcome of the approaches and final reconstructions. The velocity mode is preferable, as it leads to less acquisition times.

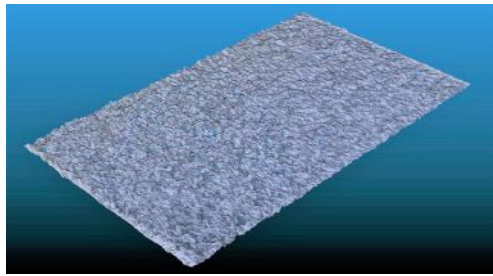
## 6.2 Future Work

The proposed approaches show a significant improvement in the reconstruction. Still, there are many potential enhancements that can be applied. The rectification evaluation shows an increase of varying disparity. A more advanced approach could be taken to address this problem in both approaches. In the case of the optimization based approach, the BA process could be supported by fixed keypoints on the objects to further stabilize the outcome of the optimization process. To reduce the impact of surface texture, an optical flow based approach could lead to improved results on low textured objects.

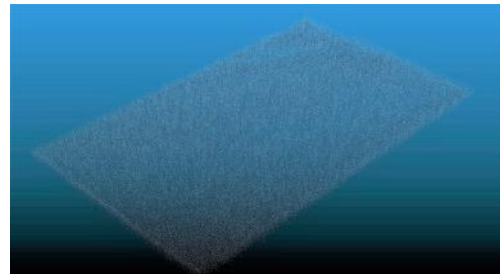
Even though, the computational speed was not part of this work, implementing the approaches efficiently by utilizing CUDA on an additional GPU, that is unused by the ICI, could lead to processing times in the range of seconds. Comparing this to Meshroom or other state-of-the-art reconstruction tools which take several hours, the system could be applied for fast, industrial inspection. Furthermore, a main feature of the ICI is not utilized in this work, photometric stereo. Using multiple light sources alternately could lead to a further increase in the quality of the reconstruction.

# 7 Appendix

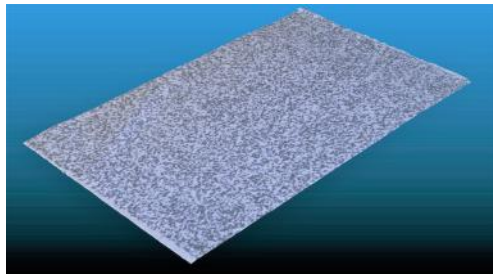
## 7.1 Point clouds



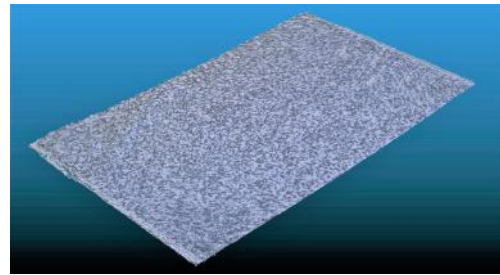
(a) Original



(b) Meshroom



(c) Homography



(d) Global

Figure 7.1: Object Random Pattern, Step Mode, 1mm

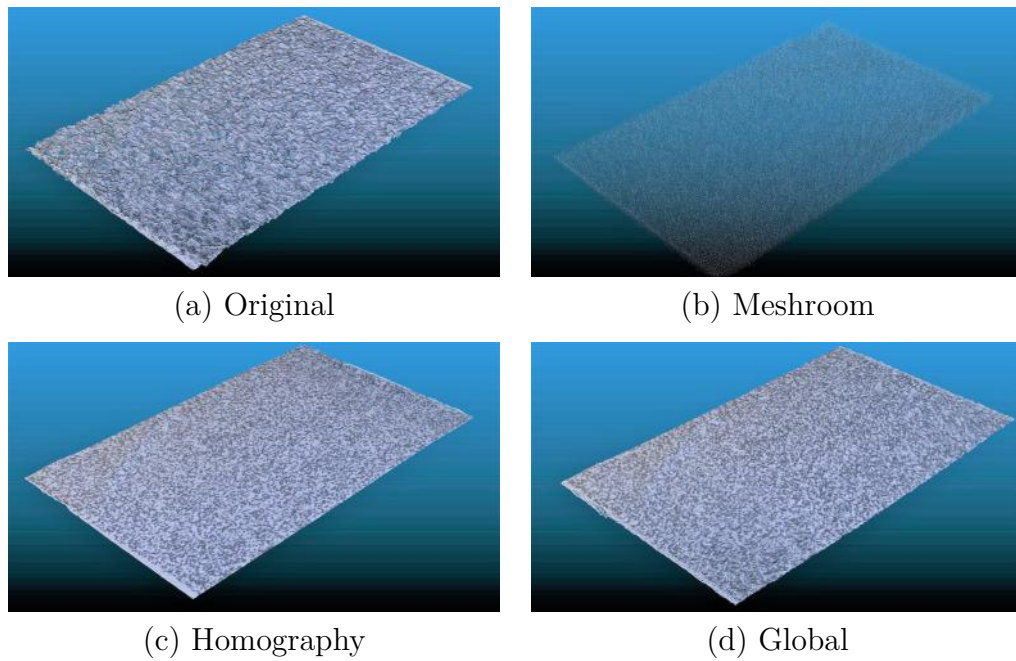


Figure 7.2: Object Random Pattern, Velocity Mode, 40 mm/s

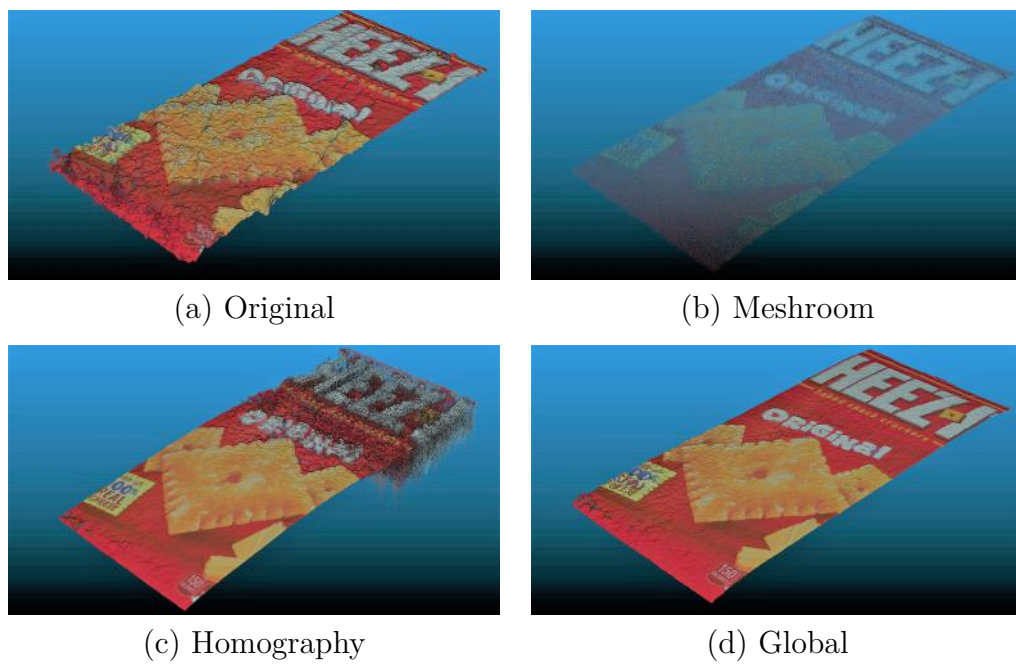


Figure 7.3: Object Cornflakes, Step Mode, 1mm



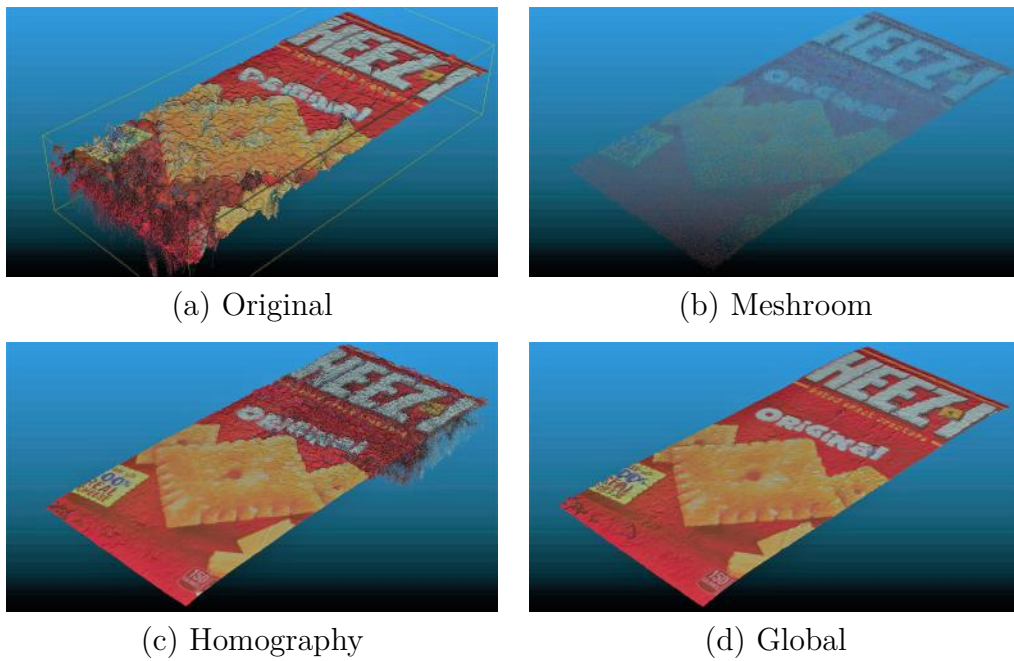


Figure 7.4: Object Cornflakes, Velocity Mode, 40 mm/s

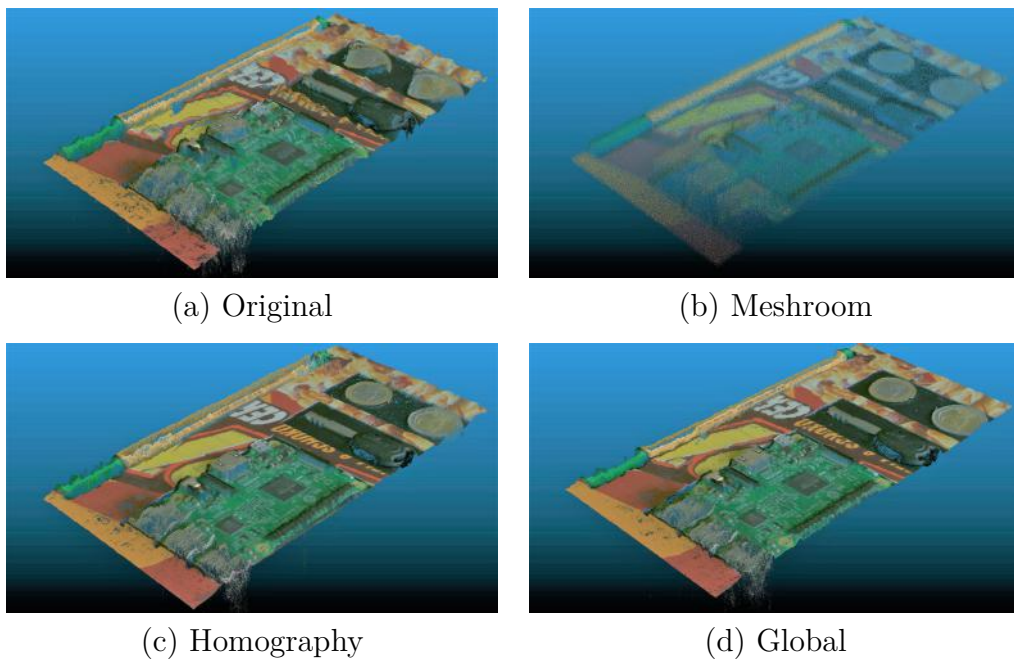
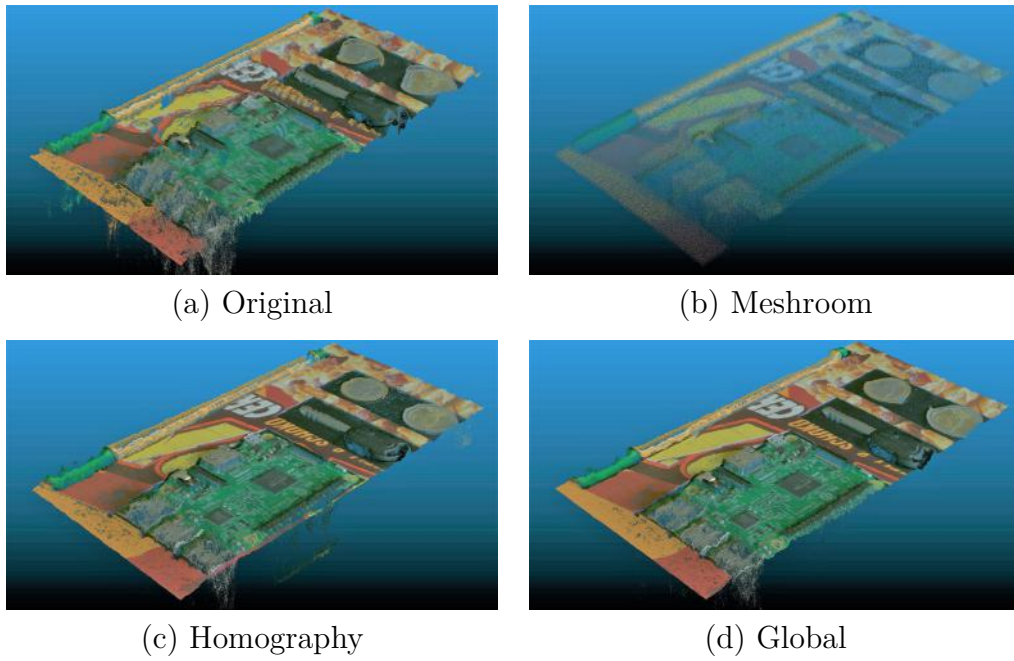


Figure 7.5: Object Board, Step Mode, 1mm



(a) Original

(b) Meshroom

(c) Homography

(d) Global

Figure 7.6: Object Board, Velocity Mode, 40 mm/s



# Bibliography

- [1] B. Blaschitz, S. Breuss, L. Traxler, L. Ginner, and S. Štolc, „High-speed inline computational imaging for area scan cameras,“ *Electronic Imaging*, vol. 2021, no. 6, pp. 301–1, 2021.
- [2] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, „A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos,“ Tech. Rep. [Online]. Available: [www.eth3d.net](http://www.eth3d.net)..
- [3] AIT, „Inline Computational Imaging,“ 2021. [Online]. Available: [https://www.ait.ac.at/fileadmin//mc/vision\\_automation\\_control/F\\_F\\_F/flyer\\_wickelfalz\\_DIN\\_A4\\_hoch\\_6-seiter\\_ICI-v10\\_2021.pdf](https://www.ait.ac.at/fileadmin//mc/vision_automation_control/F_F_F/flyer_wickelfalz_DIN_A4_hoch_6-seiter_ICI-v10_2021.pdf).
- [4] C. Rossi and S. Savino, „A Robotic System to Scan and Reproduce Object,“ *Journal of Robotics*, vol. 2011, pp. 1–11, 2011, ISSN: 1687-9600.
- [5] M. Callieri, A. Fasano, G. Impoco, P. Cignoni, R. Scopigno, G. Parrini, and G. Biagini, „RoboScan: An automatic system for accurate and unattended 3D scanning,“ in *Proceedings - 2nd International Symposium on 3D Data Processing, Visualization, and Transmission. 3DPVT 2004*, 2004, pp. 805–812, ISBN: 0769522238. [Online]. Available: <https://www.researchgate.net/publication/4091707>.
- [6] Q. Huang, J. Lan, and X. Li, „Robotic Arm Based Automatic Ultrasound Scanning for Three-Dimensional Imaging,“ *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1173–1182, 2019, ISSN: 15513203.
- [7] G. Alenyà, S. Foix, and C. Torras, „ToF cameras for eye-in-hand robotics,“ in *Optical Imaging Devices: New Technologies and Applications*, 2017, pp. 117–148, ISBN: 9781498711012.
- [8] A. F. Martins, M. Bessant, L. Manukyan, and M. C. Milinkovitch, „R2OBBIE-3D, a fast robotic high-resolution system for quantitative phenotyping of surface geometry and colour-texture,“ *PLoS ONE*, vol. 10, no. 6, Jun. 2015, ISSN: 19326203.

- [9] S. Zambal, W. Palfinger, and C. Eitzinger, „Robotic inspection of 3D CFRP surfaces,“ in *3rd IEEE International Workshop on Metrology for Aerospace, MetroAeroSpace 2016 - Proceedings*, 2016, pp. 197–202, ISBN: 9781467382922.
- [10] H. C. Longuet-higgins, „A computer algorithm for reconstructing a scene from two projections,“ *Nature*, vol. 293, no. 5828, pp. 133–135, 1981, ISSN: 00280836.
- [11] A. J. Davison, „Real-time simultaneous localisation and mapping with a single camera,“ in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, 2003, pp. 1403–1410. [Online]. Available: <http://www.robots.ox.ac.uk/~æajd/>.
- [12] M. R. U. Saputra, A. Markham, and N. Trigoni, *Visual SLAM and structure from motion in dynamic environments: A survey*, 2018. [Online]. Available: <https://doi.org/10.1145/3177853>.
- [13] G. Klein and D. Murray, „Parallel tracking and mapping for small AR workspaces,“ in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR*, 2007, pp. 225–234, ISBN: 9781424417506.
- [14] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, „Keyframe-Based Visual-Inertial SLAM using Nonlinear Optimization,“ 2016.
- [15] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, „ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM,“ 2020. arXiv: 2007.11898. [Online]. Available: <http://arxiv.org/abs/2007.11898>.
- [16] R. Mur-Artal, J. M. Montiel, and J. D. Tardos, „ORB-SLAM: A Versatile and Accurate Monocular SLAM System,“ *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015, ISSN: 15523098. arXiv: 1502.00956.
- [17] R. Mur-Artal and J. D. Tardos, „ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras,“ *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017, ISSN: 15523098. arXiv: 1610.06475. [Online]. Available: <http://arxiv.org/abs/1610.06475><http://dx.doi.org/10.1109/TR0.2017.2705103>.
- [18] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, „SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems,“ Tech. Rep. 2, 2017, pp. 249–265. [Online]. Available: <https://youtu.be/hR8uq1RTUfA>.

- [19] S. Bianco, G. Ciocca, and D. Marelli, „Evaluating the performance of structure from motion pipelines,” *Journal of Imaging*, vol. 4, no. 8, Aug. 2018, ISSN: 2313433X.
- [20] O. Ozyesil, V. Voroninski, R. Basri, and A. Singer, „A survey of structure from motion,” *Acta Numerica*, vol. 26, pp. 305–364, 2017, ISSN: 14740508. arXiv: 1701.08493.
- [21] Y. Chen, Y. Chen, and G. Wang, „Bundle adjustment revisited,” Tech. Rep., 2019. arXiv: 1912.03858.
- [22] A. Irschara, C. Hoppe, H. Bischof, and S. Kluckner, „Efficient structure from motion with weak position and orientation priors,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2011, ISBN: 9781457705298. [Online]. Available: <https://www.researchgate.net/publication/224253054>.
- [23] M. Lhuillier, „Incremental fusion of structure-from-motion and GPS using constrained bundle adjustments,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 12, pp. 2489–2495, 2012, ISSN: 01628828. [Online]. Available: <http://maxime.lhuillier.free.fr/http/ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=6332439>.
- [24] Y. Gong, D. Meng, and E. J. Seibel, „Bound constrained bundle adjustment for reliable 3D reconstruction,” *Optics Express*, vol. 23, no. 8, p. 10 771, 2015, ISSN: 1094-4087.
- [25] Y. Zhang, P. An, H. Wang, and Z. Zhang, „A rectification algorithm for un-calibrated multi-view images based on SIFT features,” *ICALIP 2010 - 2010 International Conference on Audio, Language and Image Processing, Proceedings*, pp. 143–147, 2010.
- [26] F Zilly, C Riechert, M. Muller, W Waizenegger, T Sikora, and P Kauff, „Multi-camera rectification using linearized trifocal tensor,” in *Proceedings - International Conference on Pattern Recognition*, 2012, pp. 2727–2731, ISBN: 9784990644109.
- [27] Y. S. Kang and Y. S. Ho, „An efficient image rectification method for parallel multi-camera arrangement,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1041–1048, 2011, ISSN: 00983063.
- [28] Z. Zhu, A. R. Hanson, and E. M. Riseman, „Generalized Parallel-Perspective Stereo Mosaics from Airborne Video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 226–237, 2004, ISSN: 01628828.

- [29] X. Geng, Q. Xu, S. Xing, C. Lan, and Y. Hou, „Real time processing for epipolar resampling of linear pushbroom imagery based on the fast algorithm for best scan line searching,“ in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. XL-2/W2, 2013, pp. 129–131.
- [30] A. Kaehler and G. Bradski, *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*, eng. Sebastopol: O’Reilly Media, Incorporated, 2017, ISBN: 1491937998.
- [31] R. Szeliski, *Computer Vision: Algorithms and Applications*. 2010. [Online]. Available: [http://szeliski.org/Book/..](http://szeliski.org/Book/)
- [32] Y. Furukawa and C. Hernandez, „Multi-View Stereo: A Tutorial,“ *Foundations and Trends in Computer Graphics and Vision*, vol. 9, no. 1-2, pp. 1–148, 2015, ISSN: 1572-2740. [Online]. Available: <http://dx.doi.org/10.1561/06000000052>.
- [33] Z. Zhang, „A flexible new technique for camera calibration,“ *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000, ISSN: 01628828.
- [34] *Epipolar geometry - Wikipedia*. [Online]. Available: [https://en.wikipedia.org/wiki/Epipolar\\_geometry](https://en.wikipedia.org/wiki/Epipolar_geometry) (visited on 03/22/2021).
- [35] D. Nistér, „An efficient solution to the five-point relative pose problem,“ *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004, ISSN: 01628828.
- [36] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.
- [37] D. G. Lowe, „Distinctive Image Features from Scale-Invariant Keypoints David,“ *International Journal of Computer Vision*, 2004.
- [38] J. L. Schönberger and J.-M. Frahm, „Structure-from-Motion Revisited,“ Tech. Rep. [Online]. Available: <https://github.com/colmap/colmap..>
- [39] S. Zhu, R. Zhang, L. Zhou, T. Shen, T. Fang, P. Tan, and L. Quan, „Very Large-Scale Global SfM by Distributed Motion Averaging,“ in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4568–4577, ISBN: 9781538664209.
- [40] *Reprojection error, Pix4d*. [Online]. Available: <https://support.pix4d.com/hc/en-us/articles/202559369-Reprojection-error> (visited on 11/18/2020).

- [41] Y. Furukawa, „Photo-Consistency,“ in *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed. Boston, MA: Springer US, 2014, pp. 595–597, ISBN: 978-0-387-31439-6. [Online]. Available: [https://doi.org/10.1007/978-0-387-31439-6\\_204](https://doi.org/10.1007/978-0-387-31439-6_204).
- [42] *OpenCV: Depth Map from Stereo Images*. [Online]. Available: [https://docs.opencv.org/master/dd/d53/tutorial\\_py\\_depthmap.html](https://docs.opencv.org/master/dd/d53/tutorial_py_depthmap.html) (visited on 03/22/2021).
- [43] *Point Cloud and 3D Image | Revopoint 3D Technologies Inc.* [Online]. Available: <https://www.revopoint3d.com/point-cloud-and-3d-image/> (visited on 03/22/2021).
- [44] S. A. K. Tareen and Z. Saleem, „A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK,“ *2018 International Conference on Computing, Mathematics and Engineering Technologies: Invent, Innovate and Integrate for Socioeconomic Development, iCoMET 2018 - Proceedings*, vol. 2018-Janua, pp. 1–10, 2018.
- [45] B. Blaschitz, S. Štolc, and D. Antensteiner, „Geometric calibration and image rectification of a multi-line scan camera for accurate 3D reconstruction,“ *Electronic Imaging*, vol. 2018, no. 9, pp. 240–1, 2018.
- [46] *Tracks, OpenMVG library*. [Online]. Available: <https://openmvg.readthedocs.io/en/latest/openMVG/tracks/tracks/> (visited on 03/22/2021).
- [47] *Camera Calibration - ROS Wiki*. [Online]. Available: [http://wiki.ros.org/camera\\_calibration](http://wiki.ros.org/camera_calibration) (visited on 10/16/2021).