# 3D Reconstruction in Scanning Electron Microscope : from image acquisition to dense point cloud

Andrey Kudryavtsev

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UFC UNIVERSITÉ DE FRANCHE-COMTÉ

UBFC UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

# 3D Reconstruction in Scanning Electron Microscope

## from image acquisition to dense point cloud

ANDREY V. KUDRYAVTSEV

SPIM

Thèse de Doctorat

école doctorale **sciences pour l'ingénieur et microtechniques**

UNIVERSITÉ DE FRANCHE-COMTÉ

N° | X | X | X

UBFC
UNIVERSITÉ
BOURGOGNE FRANCHE-COMTÉ

THÈSE présentée par

ANDREY V. KUDRYAVTSEV

pour obtenir le

Grade de Docteur de
l'Université de Franche-Comté

Spécialité : **Automatique**

# 3D Reconstruction in Scanning Electron Microscope

## from image acquisition to dense point cloud

Soutenue publiquement le 31 october 2017 devant le Jury composé de :

| | | |
|---|---|---|
| PETER STURM | Rapporteur | Directeur de Recherche HDR, INRIA Grenoble Rhônes-Alpes |
| JACQUES GANGLOFF | Rapporteur | Professeur, Université de Strasbourg |
| OLIVIER HAEBERLÉ | Examinateur | Professeur, Université de Haute-Alsace |
| CÉDRIC DEMONCEAUX | Examinateur | Professeur, Université de Bourgogne |
| NADINE PIAT | Directeur de thèse | Professeur, ENSMM, Besançon |
| SOUNKALO DEMBÉLÉ | Directeur de thèse | Maître de Conférences HDR, Université de Franche-Comté |

*...to my loving and amazing grandmother*

# Acknowledgment

I would like to take this opportunity to thank the people who have supported, encouraged, and inspired me in the process of writing my thesis. You made all the difference.

First, I must thank my supervisors, Dr. Sounkalo Dembélé and Dr. Nadine Piat, for offering me this Ph.D. position. Your continuous support and trust helped me a lot during these three years. I will never thank you enough for the confidence you had in me, your guidance, and advice throughout this Ph.D.

I want to thank all the people in AS2M department of FEMTO-ST Institute who were always there for me. Patrick Rougeot, Jean-Yves Rauch, Guillaume Laurent, Olivier Lehmann, Cédric Clévy, and Brahim Tamadazte (order is arbitrary): thanks a lot! I cannot but mention all my colleagues who contributed in that great atmosphere where I have had an honor and a pleasure to work: Vincent Trenchant, Margot Billot, Houari Bettahar, Elodie Lechartier, Adrian Ciubotariu, Marcelo Gaudenzi de Faria, Mohamed Taha Chikhaoui, Mouloud Ourak, Bassem Dahroug, Benoit Brazey and I have certainly forgotten somebody...

I express my sincere gratitude to Dr. Peter Sturm and Dr. Jacques Gangloff for accepting to be the referees of the present work and devoting time to carefully read this manuscript. I am sure that your suggestions, both in your written reports and during the defense, have helped me to improve this work. And I convey my heartfelt thanks to all other members of the jury: Dr. Olivier Haeberlé and Dr. Cédric Demonceaux.

Last, but by no means the least, I want to thank all my family: my grandmother Lina, my parents Vladislav and Irina, my sister Olga and her husband Roma, my niece and nephew, Lena and Denis. And of course, I thank my dearly loved Tanya, who kept me fed and smiling, and supported me in times of stress and frustration. You are the best!

# Contents

# Mathematical symbols

| | |
|---|---|
| $a$ | scalar |
| $\mathbf{a}$ | vector |
| $\mathbf{A}$ | matrix |
| $\mathcal{A}$ | stack (array) of matrices |
| $\mathbf{A}^\top$ | matrix transpose |
| $\mathbf{A}^{-1}$ | matrix inverse |
| $\mathbf{A}^+$ | matrix pseudo-inverse |
| $\cdot /\!\!/$ | related to parallel projection |
| $\|.\|_F$ | Frobenius norm |
| $diag(a, b, c)$ | diagonal matrix with elements $a$, $b$, $c$ |

# Abbreviations

| | |
|---|---|
| AFM | Atomic Force Microscope (Microscopy) |
| CAD | Computer-aided design |
| DIC | Digital image correlation |
| DOF | Depth Of Field |
| DoF | Degrees of Freedom |
| FE-SEM | Field Emission Scanning Electron Microscope |
| GPL | General Public License |
| GUI | Graphical User Interface |
| LM | Levenberg-Marquardt optimization method |
| LMedS | Least Median of Squared robust estimator |
| LQ | Long Quan method of autocalibration |
| MLESAC | Maximum Likelihood Estimation Sample Consensus |
| MSAC | M-estimator Sample Consensus |
| NCC | Normalized Cross Correlation |
| POC | Phase Only Correlation |
| PS | Photometric Stereo |
| RANSAC | Random Sample Consensus |
| ROI | Region of interest |
| SAD | Sum of Absolute Differences |
| SEM | Scanning Electron Microscope (Microscopy) |
| SFM | Structure From Motion |
| SFS | Shape From Shading |
| SGM | Semi-global matching |
| SSD | Sum of Squared Differences |
| STM | Scanning Tunneling Microscope |
| SVD | Singular Value Decomposition |
| TCP | Tool Center Point |
| TEM | Transmission electron microscopy |

# Main notations

| | |
|---|---|
| $\alpha$ | aspect ratio |
| $\mathbf{A}$ | upper-left $2 \times 2$ part of calibration matrix |
| $C$ | camera center |
| $\mathcal{C}$ | cost function |
| $f$ | focal length, scale factor |
| $\mathbf{F}$ | fundamental matrix |
| $\mathbf{K}$ | calibration matrix, matrix of intrinsic parameters ($3 \times 3$) |
| $\mathbf{M}$ | upper-left $2 \times 3$ part of affine camera matrix |
| $N_{im}$ | number of images |
| $N_{iter}$ | number of iterations |
| $N_{pts}$ | number of points |
| $\mathbf{P}$ | camera matrix (perspective or affine) ($3 \times 4$) |
| $\mathbf{P}_{/\!/}$ | affine camera matrix ($3 \times 4$) |
| $\mathbf{q}$ | 2D point, $\mathbf{q} = (q_x, q_y)^\top$, or in homogeneous coordinates $\mathbf{q} = (q_x, q_y, 1)^\top$ |
| $\tilde{\mathbf{q}}$ | 2D point, $\tilde{\mathbf{q}} = (\tilde{q}_x, \tilde{q}_y)^\top$, in relative coordinates |
| $\mathbf{q}_j^i$ | 2D point number $j$ in image $i$ |
| $\mathbf{Q}$ | 3D point, $\mathbf{Q} = (Q_x, Q_y, Q_z)^\top$ or in homogeneous coordinates $\mathbf{Q} = (Q_x, Q_y, Q_z, 1)^\top$ |
| $\rho$ | out-of-plane rotation about $\vec{y}$ axis |
| $\mathbf{R}$ | rotation matrix |
| $\mathbf{R}_x$ | rotation about $\vec{x}$ axis |
| $\mathbf{R}_y$ | rotation about $\vec{y}$ axis |
| $\mathbf{R}_z$ | rotation about $\vec{z}$ axis |
| $s$ | skew |
| $\mathbf{t}$ | translation vector |
| $\mathbf{T}$ | transformation matrix ($4 \times 4$) |
| $\theta$ | slope angle of epipolar line |
| $\mathcal{W}$ | measurement matrix |
| $\mathcal{W}_r$ | measurement matrix in relative coordinates |
| $\boldsymbol{\xi}$ | vector of parameters (in context of optimization) |
| $\boldsymbol{\xi}_0$ | initial vector of parameters |
| $\boldsymbol{\xi}^*$ | solution of optimization problem |

# Introduction

The subject of 3D reconstruction at the microscale, treated in this thesis, has a direct link with the huge topic of microcharacterization as one can extract all dimensional information about the object from its 3D model. Nowadays, it is very difficult to underestimate the importance of dimensional micro- and nanocharacterization. It finds the application in a variety of fields, from biology to material science, with some of the examples given below.

Metrology is essential for the manufacturing of high quality tools with reduced dimensional tolerances which means more effective machines, improvement of their performance and longevity. One of the examples comes from micro-milling of a hardened tool steel with micro ball-end mills [EFG+15]. The purpose of this study was to observe the capability of a set of these mills to machine hard steels used for tooling applications. Authors concluded that the local geometry of micro-mills has great influence on wear resistance of the tool, thus, its characterization would allow predicting its cycle life.

In [MSS+12], *Muralikrishnan et al.* worked with the fiber probes that are increasingly used for dimensional metrology of microscale features on coordinate measuring machine. These delicate probes have slender stems (5 mm to 20 mm long, 10 µm to 100 µm in diameter) with a ball (50 µm to 200 µm diameter) at the end. Authors state that, unlike the probing spheres of a traditional coordinate measuring machine, the geometry of a fiber probe tip is difficult to control. Its analysis is of great importance as it highly influences the final quality of the measurements. Authors used another object with known geometry in order to perform the calibration.

In optics, *Thiele et al.* presented a technology where micro lenses (Figure 1) are directly printed on a CMOS image sensor as an array of hemispheres with a height less than 200 µm [TAG+17]. As a result, they obtained a miniaturized camera, mimicking the natural vision of predators, with applications in such fields as endoscopy, optical sensing, and optical metrology. The quality of the microsphere surface has a direct impact on the final image quality: it requires to measure the roughness with an uncertainty level of several nanometers.

Microcharacterization also plays an important role in material science as the mechanical properties of the material such as strength, fatigue, and cycle life depends on its microstructure. Bioinspired materials repeat the structures created by nature. Indeed, the study of lotus leaves having hydrophobic water-repellent double structure allowed creation of coatings, paints and other surfaces with the same properties (Figure 2). Thus, the study and characterization of nanostructures allow creation of new materials with needed properties and performance [MLL+09].

The tool that is used the most for observation and measurements of small-scale objects are the microscopes. They may be based on different physical principles: light,

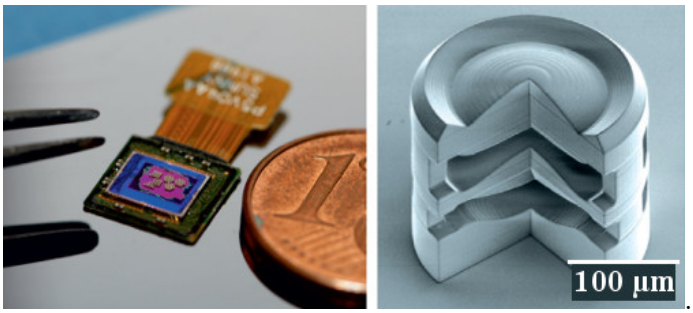



Figure 1: a) Pictures of an image sensor with doublet lenses; b) SEM image of a lens [GTH+16].

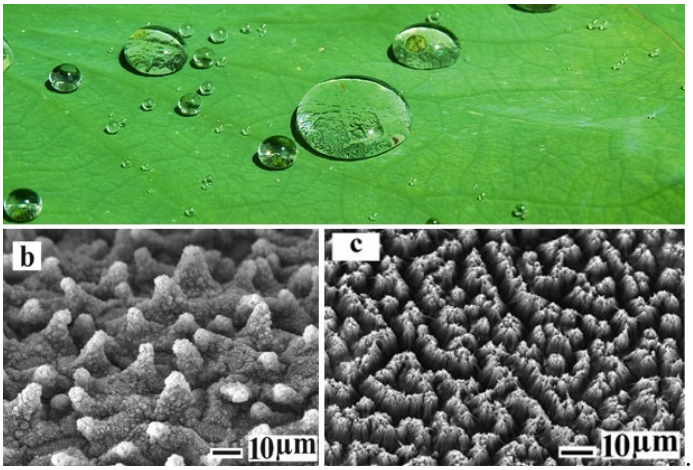



Figure 2: a) Lotus leaf and water drops; b) SEM image of the surface of lotus leaf; c) SEM view of polystyrene nanotube film surface having the same water-repellent properties [MLL+09].

electron, and ion beams, x-ray, quantum tunneling, etc. The evolution of microscopy is relative to the increase of resolution that now can be inferior to 0.1 nm. Resolution is approximated by resolving power that corresponds to the shortest distance between two points on a specimen that can still be distinguished as separate entities on the microscope image. The following paragraphs present a brief overview of most representative microscope types (Figure 3).

The oldest designed microscope is the **optical microscope** (or light microscope). It uses visible light and a system of magnifying lenses to visualize small objects. Nowadays, it is the most spread tool of dimensional control in different research fields such as microelectronics, microbiology, and pharmaceutics. First optical microscopes appeared in the XVII century and yet technical and industrial evolution turned them into an important metrology device with resolving power reaching 0.2 µm. Their main limitation is the geometry of the viewed object: the microscope is adapted only for flat objects. In order to achieve high resolution, the objectives are fabricated with a high aperture that decreases dramatically the depth of field, i.e. the distance between the nearest and farthest objects in a scene that appear acceptably sharp. It is important to note that even with a top quality optics, the resolution of light microscope is limited by the wavelength of light equal to 0.2 µm. The greatest resolving power in optical microscopy is realized with near-ultraviolet light, the shortest effective imaging wave-

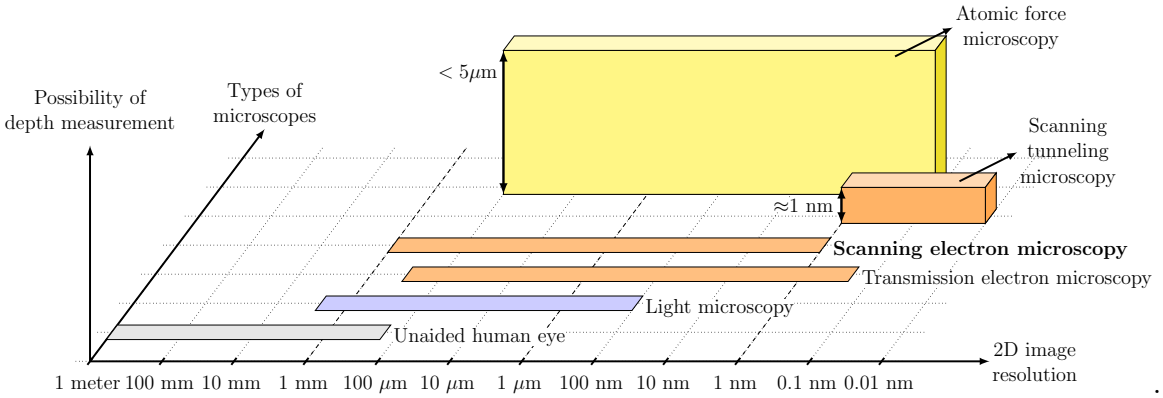
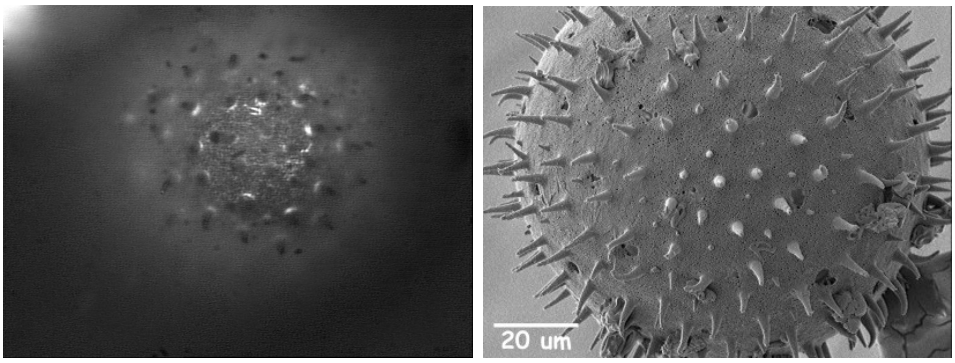



Figure 3: Different types of microscopes and their operational resolution. AFM and STM allows also the analysis of surface depth variation.




(a) optical microscope (b) SEM

Figure 4: Comparing images of the same pollen grain coming from different imaging devices.

length. That is why, scientists turned their attention towards another physical entity, electrons.

The wavelength of an electron can be up to $10^5$ times shorter than that of visible light photons, which marked the beginning of development of **electron microscopy**. In such microscopes, a beam of accelerated electrons is used as a source of illumination. Here, we mention three main types of electron microscopes: transmission electron microscope (**TEM**), scanning electron microscope (SEM) and scanning tunneling microscope (STM). The first one is based on the detection of the electrons passing through the surface of the sample. It has a much higher resolution than an optical microscope which goes up to 0.1 nm. The magnification may reach $\times 1,000,000$: it can visualize atoms! However, it requires the sample to be very thin (about 200 nm) as the electrons need to pass through it. As a result, obtaining high resolving power in 2D leads to a complete loss of the third dimension (the depth coordinate) already at the step of sample preparation.

While also using the electron beam, **SEM** image is formed by gathering the electrons reflected from the surface of the sample. Among different types of reflected electrons, the most common are the secondary electrons that are emitted by the atoms of the sample excited by the electron beam. The scanning coils allow to change the point of beam convergence and thus to obtain the effect of scanning. The number of

the detected electrons for one point of the surface is then translated into image intensity level and so the image is obtained. With SEM, we obtain images of excellent quality and high resolution of a sample with an arbitrary topography provided that it has a very big depth of field that may achieve hundreds of micrometers depending on magnification (Figure 4). However, as with every camera, the depth coordinate is lost during the image formation.

The first type of electron microscopes allowing the reconstruction of depth coordinate is the **STM**. It is based on the principles of quantum mechanics and more precisely on the phenomenon where a particle tunnels through a barrier contrary to the rules of classical mechanics. When a conducting tip is near the observed specimen surface, a voltage difference between them is applied in order to allow electrons to overcome the vacuum gap between them. Then, the value of the current determines the distance between the specimen and the tip, that corresponds to the depth coordinate. However, the depth variation of the specimen cannot exceed the value of 1 nm. Being based on tunneling, it is mainly used to observe the properties and behavior of sub-atomic particles with lateral resolution up to 0.01 nm.

Another popular type of microscopes is **Atomic-force microscope** (AFM) which is based on probing of the surface with a mechanical tip. In addition to imaging capabilities, it is also used for force measurement. Thanks to that, it can provide depth information by measuring the force of the reaction between the tip and the sample. However, due to the form of the tip (pyramid, in general), there are some limits imposed on the surface of the viewed specimen: the depth variation should not exceed 5 μm (Figure 3). Thus, it is an extremely powerful tool for characterization of flat surfaces, with resolution of fractions of nanometer, but not for objects with complex structure such as pollen grains etc.

To summarize, all of the presented microscopes have an important impact in micro- and nanocharacterization. Their main drawback consists in the impossibility of measuring the depth variation of complex objects which is of high importance for many scientific and industrial fields including characterization of biological samples and 3D micromanipulation. Many research works were concentrated on improving the microscopy performance by either adding touch sensors or using focus information to reconstruct the third coordinate. However, there is another solution that lies within a different scientific field which is computer vision and, in particular, 3D reconstruction.

In computer vision, 3D reconstruction is a process of recovering the shape of a real object from one or several images. This technique became very popular in 90's with increasing computational power and the growing need for graphic effects. Since then, it is widely used by animators that use the obtained 3D models for animation of virtual characters or for the creation of realistic, but still virtual, environment around the actor. However, computer graphics is only one of possible applications of 3D reconstruction. Another one consists in the object characterization that also finds its application in microscale, especially, since the choice of available sensors is limited comparing to macroscale solutions. 3D reconstruction has a number of advantages compared to other measurement methods. First, it represents a non-destructive and non-invasive method of measurement as there is no contact between camera and sample. Moreover, in the process of sample preparation, there is no need to slice the object. Secondly, 3D reconstruction contributes to the creation of models that simplify visual analysis and change the traditional way of micro-objects perception.

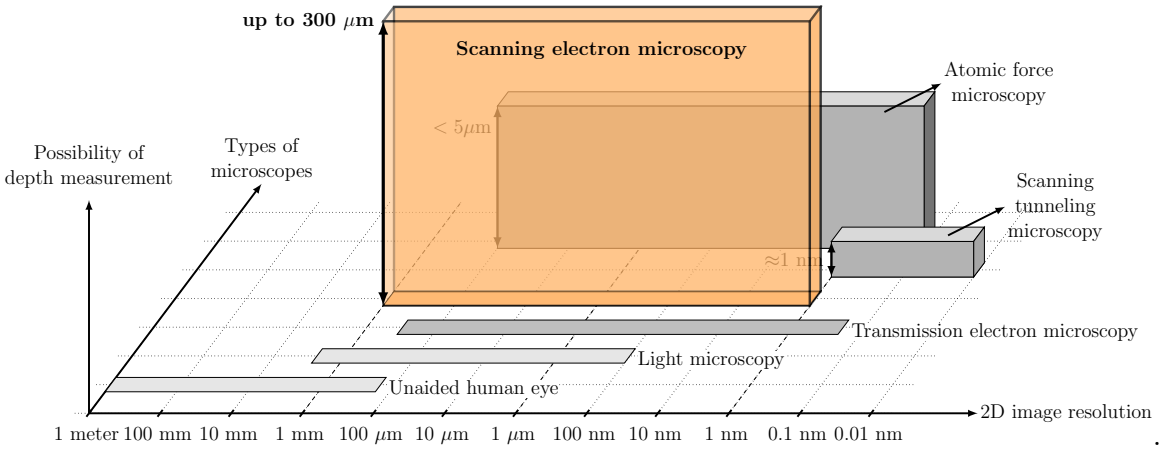



Figure 5: The goal of this study: using computer vision technique of 3D reconstruction, turn SEM into 3D measurement tool for objects with complex structure and arbitrary depth variation.

For the purpose of 3D reconstruction high quality images are needed and the choice of microscope here is quite simple. It is the SEM having the following advantages: first, the sample may have an arbitrary shape contrary to all other microscopes. Secondly, it has a big depth of field which means that even if the object has an important depth variation, all its parts will still be visible and sharp on the image. Indeed, recently, with very rapid development of computer vision, many research works have shown the potential of 3D reconstruction in SEM.

One of the first works on 3D reconstruction in SEM was realized by *Beil et al.* in 1991 [BC91]. Authors proposed a method using the technique of "Shape-from-shading". It was used for the reconstruction of surface topographies of integrated circuits (ICs) for nondestructive testing and control of the manufacturing processes. In 2003, *Cornille et al.* developed a stereovision-based approach for surface reconstruction and successfully applied it to a pair of SEM images [CGS+03]. Most recent advances, published in 2017, were realized by *Baghaiea et al.* [BTO+17] where authors presented an approach allowing to achieve a high quality dense 3D reconstruction using iterative rectification. While the extensive review of the current state of the art is reserved for later chapters, it is important to note that, to date, some problems stay unresolved. Among them are:

**Image acquisition.** High quality reconstruction may only be achieved from high quality images and while this task is easy in macroscale, to acquire a sufficient number of images with SEM may be quite challenging. The current works use mostly up to five images acquired manually by tilting the robotic stage installed inside SEM. According to our experience, besides the sample preparation, the time needed to acquire one image may achieve thirty or more minutes, which is due to the fact that operator has to adjust many of SEM parameters before a good image is obtained. Of course, if the goal is to achieve a 360 degrees reconstruction, much more images are needed and it can take a lot of time.

**Calibration.** When the dataset is available, the problem consists in finding the geometrical parameters of SEM as well as the motion of the sample between image pairs. The classical procedure allowing to achieve this goal is called calibration. Even if, mathematically, calibration is quite simple, it becomes very complex once applied to SEM. First of all, one needs to have a calibration target which also has limited quality

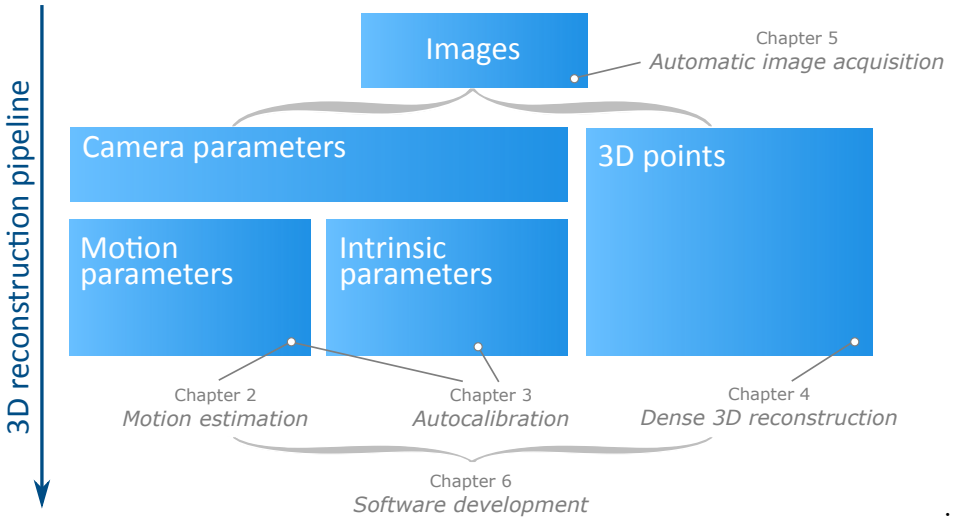



Figure 6: Structure of the thesis.

when working at high magnification. Moreover, the calibration depends on magnification which is different for every object, i.e., for every new object, SEM calibration needs to be updated.

Thus, these two problems represent the main research interest of this work and allow to formulate the main goal of this study:

> ***Main goal:*** *From multiple images obtained with uncalibrated Scanning Electron Microscope, develop a method allowing 3D reconstruction of objects with an arbitrary shape.*

Of course, this goal will be refined in the next chapters after presenting the important background information about the SEM and the current state of the art. Achieving this would allow turning SEM in a 3D measurement tool that has no analog in microscale (Figure 5) and creating new possibilities in dimensional micro- and nanocharacterization.

## Thesis outline

The final result of the present work is a dense 3D point cloud obtained from multiple SEM images of the object using 3D reconstruction. 3D reconstruction comprises several steps: from the estimation of camera motion to the triangulation of points. All these steps are covered in thesis chapters (Figure 6):

**Chapter 1** contains an important background information about SEM, its properties, and principles of image formation, from both physics and geometry points of view. It allows the deep understanding of the current state of the art presented next: its advantages and downsides. Then, the final goal of this research work is defined. As it is shown in Figure 6, 3D reconstruction may be subdivided into two major tasks which are the estimation of camera parameters (SEM in our case) and the 3D points

corresponding to the surface of the object. This distinction is also followed in the structure of the thesis.

**Chapter 2** presents a group of techniques allowing full estimation of camera motion including translations and rotations. The main complexity arises from parallel projection that creates motion ambiguities. It will be demonstrated that neither the motion nor structure can be estimated from two images. The presented methods allow avoiding these ambiguities by exploring three-view geometry and using the laws of spherical trigonometry.

**Chapter 3** covers a new method of autocalibration that includes the estimation of camera internal parameters and the refinement of motion parameters. Together with the algorithms of **Chapter 2**, the camera parameters are recovered for all images in a sequence. In contrast to the state of the art methods, we use only the images acquired for 3D reconstruction without additional sensors or prior SEM calibration.

Methods described in **Chapter 4** allow obtaining a dense point cloud in three steps. They include a new method of rectification for SEM images, dense matching, and triangulation. Using such point cloud, it is already possible to make the first measurement of 3D properties such as angles, ratios of lengths, etc. This chapter closes the work on 3D reconstruction.

As it was mentioned previously, the entry of all methods of 3D reconstruction is a sequence of images and the algorithms of **Chapter 5** are dedicated to the automation of this task. Among them are autofocus on moving object, and calibration of the robot holding the sample. Combined, they allow the definition of rotation point corresponding to the object center.

**Chapter 6** shows the result of implementation of all presented techniques in one software application called Pollen3D. Its functionality is briefly discussed in this chapter.

At the end of this introduction, we judged important to note that the presented work has a strong relation with four branches of science: robotics, optimization, microscopy and computer vision (Figure 7).

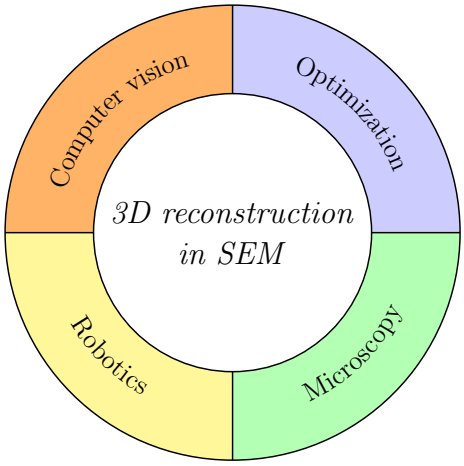



Figure 7: The present work is located between four scientific branches.

# Chapter 1

# Background of 3D reconstruction in SEM

## Contents



*This chapter presents an essential information needed to analyze the current state of the art, advantages and drawbacks of the existing methods of 3D reconstruction in SEM. First, the process of image formation in SEM is described: from physics and geometry points of view. For the remainder of this work, SEM is treated as a camera and will be often referred to like it is. Indeed, the SEM image is obtained by projecting 3D points on the common plane with one very important particularity: for an object of 10 $\mu$m the distance from detector to object may be about 10 mm. It means that the ratio of the camera-to-object distance over the object size is greater than 1000. To compare, it is the same as to watch a person from a distance of one and a half kilometer. Certainly, it plays an important role in image modeling and this chapter describes why and how. Secondly, the current state of the art on 3D reconstruction in SEM is presented and analyzed that allowed to refine and to formulate the final goal of the thesis.*

## 1.1 Image formation: physics

At the end of XIX century, *Ernst Abbé* proved the limitations of optical microscopes resolution defined by the wavelength of light [Abb73]: the smallest detail that can ever be resolved optically is of the order of two hundred nanometers. Therefore, scientists turned their attention toward another wave-like physical effect which is the electron beam that marked the appearance of electron microscopy in the beginning of XX century. In 1928, *Ernst Ruska* presented the first prototype of electron microscope giving the magnification of $\times 17$. It became possible with the use of magnetic field for electron beam focusing. By 1932, the magnification was increased to $\times 400$. By then, electron microscopy made a huge impact on biological studies: for the first time in history, scientists could see the structure of molecules, proteins and viruses. Nowadays, the value of magnification may achieve $\times 1,000,000$.

There is a number of manufacturers producing SEMs equipped with detectors of various types. Being different by design, they still have the same operation principle which is based on the interaction of the electron beam with the studied specimen. The electrons of the probe (beam) interact with the sample material and generate various types of signals: secondary electrons, back-scattered electrons, Auger electrons, characteristic X-ray radiation, etc. Namely, secondary electrons actually carry the information about the topography of the sample and, through numerical processing, allow to obtain the 2D image. Figure 1.1, represents the internal structure of SEM. The main components are described below and more details can be found in [GNE+12]. As it was already stated previously, the exact design is different from one manufacturer to another.

**Vacuum chamber.** Vacuum is an essential prerequisite for SEM functioning. A vacuum environment means that most of the air molecules have been removed from the inside of the microscope as the presence of different pollution gases makes impossible the imaging with high resolution. If there was air in the column, the beam direction couldn't be controlled as the electrons would collide with gas molecules. Typically, most of the available SEMs operate at vacuum of $10^{-5}$ to $10^{-10}$ mbar.

**Electron gun.** The electron gun generates free electrons with given kinetic energy and predefined configuration. It is located in SEM column in deep vacuum. The electron gun consists of an electron source (Tungsten cathode, cathode of lanthanum hexaboride $LaB_6$ or field emission cathode), modulator (Wehnelt cylinder), and an anode. The cathode is the main source of electrons: when heated by direct current transmission, thermal emission of electrons takes place. Wehnelt cylinder encloses the cathode and contains the hole allowing to center the electrons passing through. A positively charged anode that is rested under the Wehnelt cap is used to accelerate the electrons to energies in the range 1-40 keV.

**Electromagnetic aperture and astigmatism changer.** These electromagnetic lenses and apertures are used to reduce the diameter of the source of electrons and to place a focused beam of electrons (spot) onto the specimen. The main goal consists in guiding the rays in the desired direction. Electromagnetic aperture has an effect of a diaphragm: it allows to control the thickness of electron beam that influences directly the depth of field (DOF, Definition 1.1). Smaller aperture narrows the electron beam which means that less electrons reach the surface. It results in bigger DOF, however, it is more difficult to obtain sharp images at high magnification as the probe current

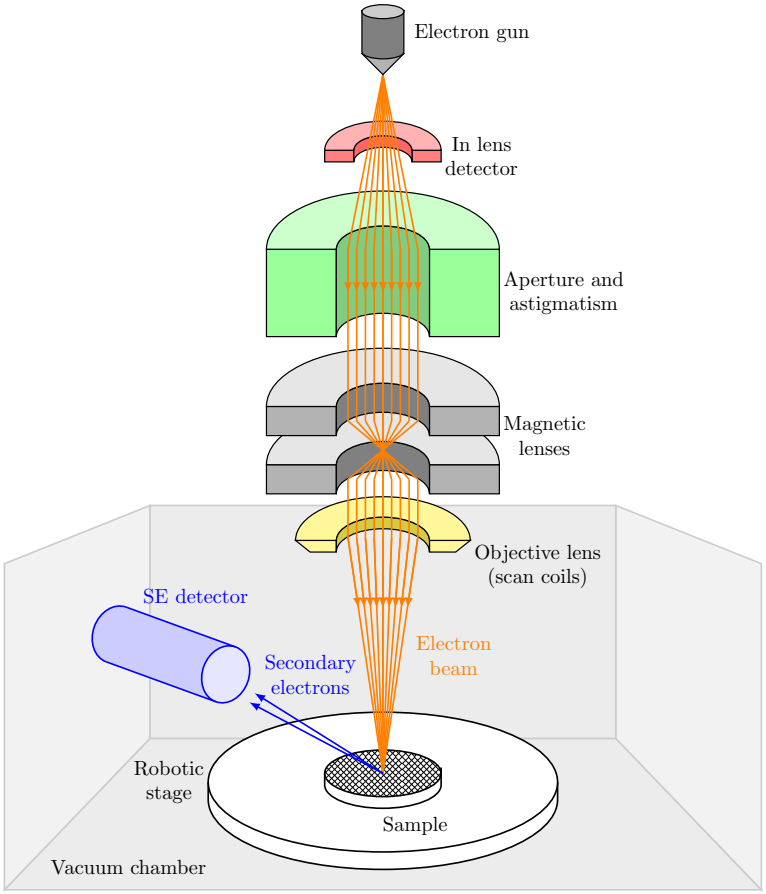



Figure 1.1: Internal structure of SEM. Scale of elements is not respected.

decreases (see Section 1.1 for more details). Astigmatism correction is needed to align the centers of aperture lens and electron gun to reduce aberrations in image formation.

**Definition 1.1**
***Depth of field*** *(DOF) represents the distance between the nearest and farthest objects in a scene that appear acceptably sharp.*

The electron beam passes then through a variety of lenses. **Magnetic lenses** or condensers are needed to shape the beam to a cylinder with diameter between 5-10 nm. These ensure a very narrow beam of electrons hits the sample. **Objective lens** is the last one that allows to finally focus the electron beam to a spot. **Scan coils** allow to define the direction of the beam, thus, position of this spot on a sample. By moving it, we reach the effect of scanning that ensures that every point (a region) on the surface of the sample is attained.

Last elements to consider are the detectors that mainly differ by the nature of detected electrons. For morphology imaging, there are two main kinds of detection principles: secondary electrons (SE) and back-scattered electrons (BSE) (Figure 1.2). All these electrons can give different information about the topography and morphology of the object. Secondary electrons are the ones that were directly rejected from specimen atoms by inelastic scattering after the sample was hit by the electron beam. SE

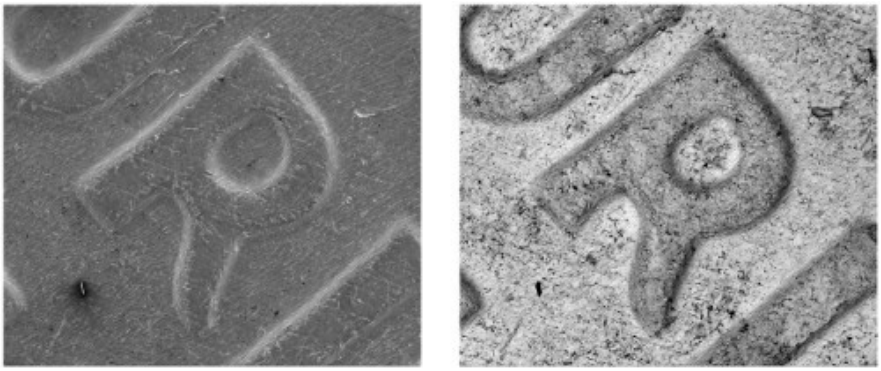


Figure 1.2: Comparison between images obtained with SE (a) and BSE (b) detectors of a penny detail. SE image reveals topography contrast and BSE image reveals atomic number contrast of the same area of a sample [Cor05].

.

are characterized by a low-energy level (<50eV). Due to that, they are originated from the top layer of the sample surface, the width of this layer lies within 10 nm [Rei72]. The number of electrons ejected from one spot is than translated in image intensity values that allow to get the image of the object surface. The electrons are detected by an Everhart-Thornley detector, which is normally positioned to one side of the specimen or directly in the column (see Figure 1.1). Contrary to SE, back-scattered electrons are the electrons coming from the electron beam that were then reflected or back-scattered out of the specimen by the nucleus of atoms. In this case, due to high energy of electrons, the image mainly depends on the atomic number of the viewed object, e.g., on the material: heavy elements appear brighter in the image. This property is very useful for analysis of the chemical homogeneity of the sample. The detector is generally positioned in the column above the sample.

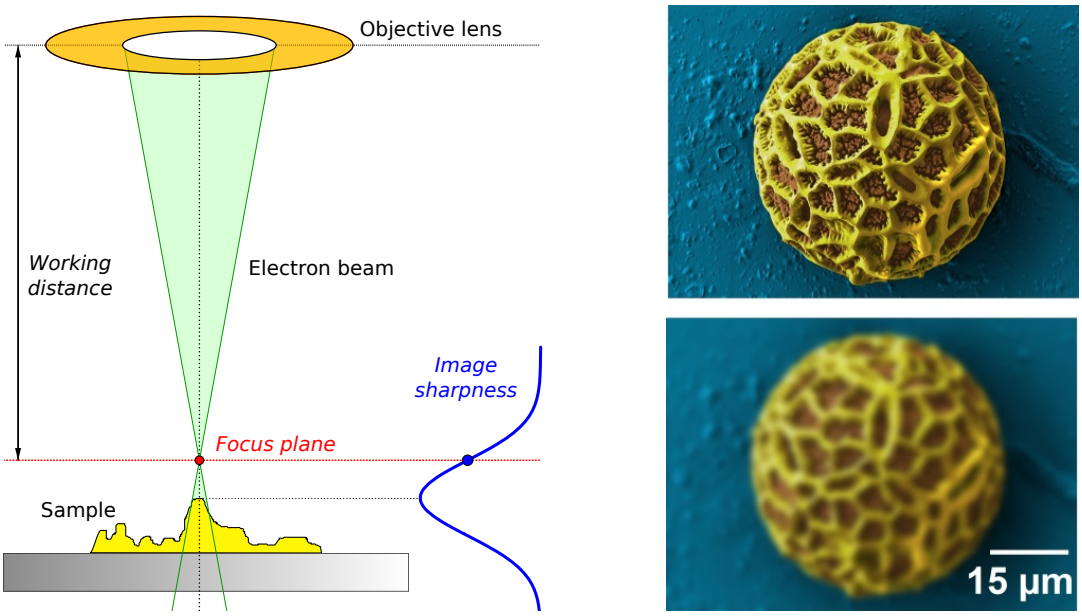
In the context of 3D reconstruction, the SE detector type is more adapted because the image corresponds exactly to the object topography. Hence, it is SE detector that will be used as the source of image in the present work. However, the realization of multimodal 3D reconstruction which means combining different imaging sources may represent a great interest for future research but goes out of the scope of this work.

## SEM main parameters

In order to produce high quality images with SEM, several parameters have to be taken into account. The order in which they are cited here corresponds to the order in which the operator generally adjusts them.

### Acceleration voltage

Acceleration voltage defines the voltage with which the electrons are accelerated down the column. Its nominal value varies between 0.5 and 30 kV. The choice of this parameter mainly depends on the nature of the sample and on its conductivity in particular. For example, when working with biological samples, i.e. polymers, the voltage is generally low, from 0.5 to 1 kV. If the voltage is too high, we observe the effect of charging that degrades dramatically the image quality, it is mainly represented by negative




(a) (b)

Figure 1.3: a) Principle of focusing in SEM with working distance defining the position of focus plane. b) examples of false colored SEM images: in-focus (top) and out-of-focus (bottom).

charges accumulated on the surface of the poor conductive specimen. More details on the charging can be found in [KAS+10]. For metal samples, the voltage may achieve 5 kV or more. Bigger values may result in the following fact: yield of secondary electrons will be reduced and surface detail will be lost as primary electrons would go deeper inside the sample which is not acceptable for the purposes of 3D reconstruction.

### Magnification

Once the acceleration voltage is properly chosen for the given specimen, the magnification is set up. It determines the size of the scanned region on the specimen. In contemporary SEMs this parameter may vary from $\times 10$ to $\times 1,000,000$. Magnification in the SEM depends only on the excitation of the scan coils and not on the excitation of the objective lens, which determines the focus of the beam.

### Aperture diameter

Next parameter is the **aperture diameter**. In practice, it represents an opening that allows controlling the thickness of electron beam. Recent models of SEMs are equipped with electromagnetic aperture, however, the older ones have a movable part with holes of different size. The value of aperture has a direct impact on the depth of field (DOF). Narrow aperture results in sharp focus at the image plane, yet, fewer electrons reach the surface which means that the acceleration voltage needs to be higher.

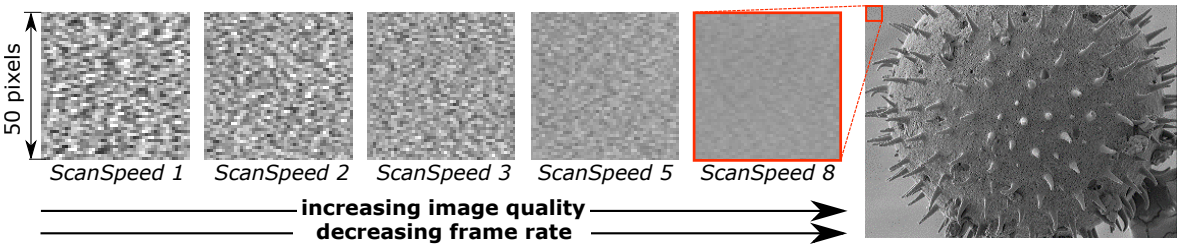



Figure 1.4: Influence of frame rate on image quality. Images represent the region of interest of 50×50 pixels from original frames acquired at different frame rates.

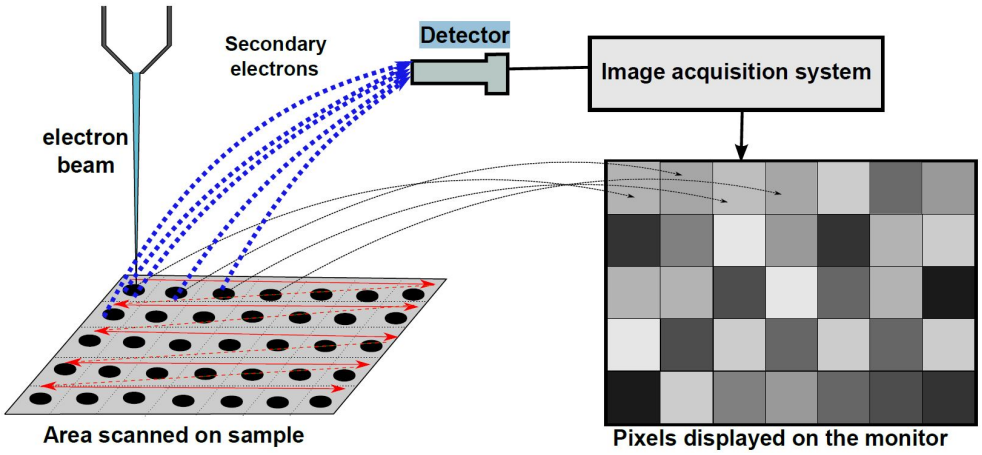



Figure 1.5: Process of image formation in SEM [Mar13].

### Working distance

Another important factor is the **working distance** (WD) which is defined as the distance between the objective lens and the focus point of electron beam (Figure 1.3). Ideally, the focus point coincides with the sample surface that gives the image with highest possible sharpness, i.e. the in-focus image. This parameter is equivalent to focal distance in macroscale cameras. It may be adjusted manually or automatically (autofocus) with typical values in range of 5 to 20 mm. It is important to note that higher values of accelerating voltage and bigger aperture allow to work with bigger working distance as electrons have enough energy to reach the surface of the sample at the higher distance.

### Frame rate

This is the last parameter we want to mention. It corresponds to the time needed to acquire one image which is indeed different from one SEM to another. The SEM used in this work has sixteen levels of frame rate: from 50 ms to several minutes. Obviously, it has drastic effect on the image quality as it is shown in Figure 1.4: with increasing frame rate (lower number of *ScanSpeed* parameter) the noise amplitude grows. For instance, with acquisition frequency of approximately 4.5 Hz (cycle time of 220 ms, *ScanSpeed2*), the standard deviation of intensity values in $50 \times 50$ ROI is 34.67, which is about 13% of the maximal value (255). To compare, the same value is equal to 5.08 for *ScanSpeed8* (cycle time 10.6 seconds).

## 1.2 Image formation: geometry

This section describes the geometric properties of image formation in SEM by which 2D images of 3D objects are obtained. Again, the image is formed by moving the electron beam across the sample surface and analyzing the emitted signal (Figure 1.5). As a result, a matrix of intensities $\mathbf{I}$, the image, is obtained. The intensity values represent the levels of gray and may vary from 0 (black) to 255 (white). Here we are mostly concentrated on secondary electrons as we are interested in object topography, although most of theoretical developments are true for other types of detectors.

In SEM, each image pixel is captured individually and its intensity depends on the result of beam/surface interaction. Geometrically, the position of every image point can be described as a projection of 3D object point onto a common surface, i.e. as a $3 \times 4$ matrix $\mathbf{P}$. This matrix is called camera matrix or projection matrix. It also defines the camera model. Among the existing camera models, two types need to be considered: perspective camera and affine camera. For both of them, the 2D projection of a 3D point has the following form (points are in homogeneous coordinates):

$$\mathbf{q} \propto \mathbf{PQ} \tag{1.1}$$

where $\mathbf{q}$ is a 2D projection ($3 \times 1$), $\mathbf{Q}$ is a 3D point ($4 \times 1$), $\mathbf{P}$ is a camera matrix ($3 \times 4$), and $\propto$ denotes the equality up to scale. The properties of each model as well as their impact in the case of SEM are discussed in the following sections. It is important to note, that, mathematically, they only differ by the form of the camera matrix, nonetheless, the change of model changes almost completely the process of 3D reconstruction.

### 1.2.1 Perspective camera

Perspective camera, or pin-hole camera, is a camera model that considers that all projection rays have an intersection point, i.e. camera center $C$ (Figure 1.6).

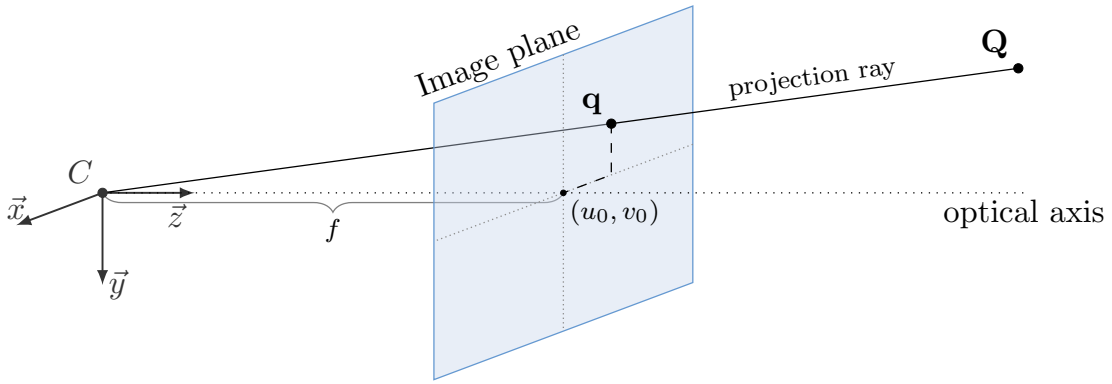



Figure 1.6: Image formation under assumption of perspective camera model.

As a result, camera matrix $\mathbf{P}$ can be decomposed in a $3 \times 3$ matrix of intrinsic parameters $\mathbf{K}$ and a $3 \times 4$ matrix of extrinsic parameters composed of rotation and

translation components $\mathbf{R}$ and $\mathbf{t}$ respectively:

$$\mathbf{P} = \underbrace{\begin{bmatrix} \alpha f & s' & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \mathbf{\Pi}\, {}^{c}\mathbf{T}_o \tag{1.2}$$

with

$$\mathbf{\Pi} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

where $f$ is the focal length, $\alpha$ is the aspect ratio, $s'$ is the skew factor and the pair $(u_0, v_0)$ denotes the principal point of the camera; ${}^{c}\mathbf{T}_o$ denotes a $4 \times 4$ homogeneous matrix that describes the transformation between world and camera frames; $\mathbf{\Pi}$ is a perspective projection matrix.

The transformation matrix ${}^{c}\mathbf{T}_o$ can be further decomposed as:

$${}^{c}\mathbf{T}_o = \begin{bmatrix} {}^{c}\mathbf{R}_o & {}^{c}\mathbf{t}_o \\ \mathbf{0}_{3\times1} & 1 \end{bmatrix} \tag{1.3}$$

where $\mathbf{R}$ is a $3 \times 3$ rotation matrix and $\mathbf{t}$ is a translation vector $3 \times 1$. The principal point is a point in which the projection ray is perpendicular to image plane, focal distance defines the distance between camera center and image frame. The intrinsic parameters are independent of camera motion. Indeed, the camera/object motion is defined by extrinsic parameters $\mathbf{R}$ and $\mathbf{t}$ that represent the transformation between the object frame ($\mathcal{R}_o$) and camera frame ($\mathcal{R}_c$). As a result, $\mathbf{P}$ has 11 independent parameters:

$$\text{Number of parameters in } \mathbf{P} = 5_{\text{intrinsic}} + 3_{\text{rotations}} + 3_{\text{translations}} = 11 \tag{1.4}$$

This model is generally applied to classical cameras characterized by the following property: objects that are closer to the camera seems bigger on the image. In other words, if the object is moving closer it becomes bigger in the image. This property is also known as perspective effect. We are all used to it, as this is the way our eyes perceive the world. Thanks to it, we are capable to estimate an approximate distance to object knowing its dimensions. However, it is no longer true when the object is far away from the camera which is the case of SEM. For SEM, the perspective effects can be neglected for magnification values bigger than $\times$1000, which is confirmed in the literature [CGS+03; CM14; SRK+02], and affine model may be used.

### 1.2.2 Affine camera

Affine camera model assumes that all projection rays are parallel to each other. That is why affine camera model is often called parallel projection model. With regard to SEM, consider the following situation which is quite common: the object with a size of 10 µm is visualized using SEM tuned to the working distance of 10 mm. The ratio between the distance camera-object and the object size is equal to 1000. Bringing the example in the macroscale: it would be the same as to look at a standard height person from more than 1.5 km! Evidently, in such conditions, the projection rays are

very close to be parallel. This example gives the intuition behind the usage of affine camera model for SEM. Getting back to geometry, the process of image formation with an affine camera is represented schematically on Figure 1.7.

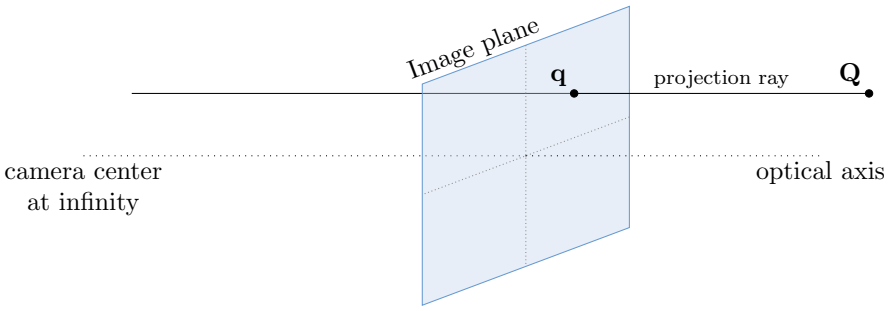



Figure 1.7: Image formation under assumption of affine camera model.

The camera matrix $\mathbf{P}_{/\!/}$ in this case is different from perspective projection by a parallel projection matrix $\mathbf{\Pi}_{/\!/}$:

$$\mathbf{P}_{/\!/} = \mathbf{K}_{/\!/}\mathbf{\Pi}_{/\!/}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix} \tag{1.5}$$

with $\mathbf{\Pi}_{/\!/} = \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$. As a result, the affine projection matrix is of the form:

$$\mathbf{P}_{/\!/} = \begin{pmatrix}* & * & * & *\\ * & * & * & *\\ 0 & 0 & 0 & *\end{pmatrix} \tag{1.6}$$

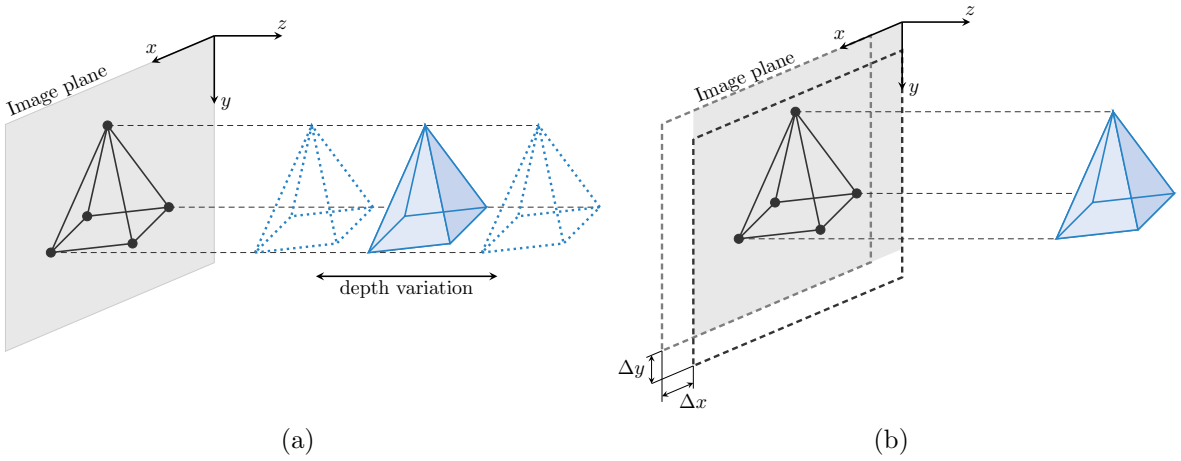
The fact of parallel projection imposes the following properties on the process of image formation (Figure 1.8,a). First, image is invariant to the object displacement along the camera optical axis. In other words, image is invariant to the distance between camera and object, which is the case in SEM. For instance, moving the sample closer to the electron beam or moving it away will not change the resulting 2D projection. As a result, the depth coordinate is lost in the process of image formation in case of parallel projection.

**Result 1.1: Invariance to depth variation**

An image acquired with affine camera is invariant to the distance between camera and object.

Secondly, the projection of the object is independent of translations in $x$ and $y$ directions of image frame if the relative coordinates are used both in 3D object frame and in camera frame [Qua96]. If such translation is performed, only the position of the object in the image changes, but not the relative disposition of its feature points (Figure 1.8,b). Indeed, for any given reference point $(q_x^r, q_y^r)^\top$ in image frame and $(Q_x^r, Q_y^r, Q_z^r)^\top$ in world frame, the expressions for relative coordinates ($\check{\mathbf{q}}$ in image frame and $\check{\mathbf{Q}}$ in a world frame) can be written as follows:

$$\begin{pmatrix}\check{q}_x\\ \check{q}_y\\ 1\end{pmatrix} = \begin{pmatrix}q_x - q_x^r\\ q_y - q_y^r\\ 1\end{pmatrix} \tag{1.7}$$




(a) (b)

Figure 1.8: Illustration of affine camera properties: a) invariance to depth variation; b) invariance to in-plane translations of camera or object.

$$\begin{pmatrix} \check{Q}_x \\ \check{Q}_y \\ \check{Q}_z \\ 1 \end{pmatrix} = \begin{pmatrix} Q_x - Q_x^r \\ Q_y - Q_y^r \\ Q_z - Q_z^r \\ 1 \end{pmatrix} \tag{1.8}$$

The reference point is often chosen as a centroid of 2D projection which is the centroid of corresponding 3D points at the same time.

**Result 1.2: Invariance to XY-translation**

The relative disposition of 2D projections does not change if camera or object is moving in plane parallel to the image plane. The exact same points may be obtained by using relative coordinates and translating the centroid in $(0, 0)^\top$.

Thirdly, in contrast to perspective camera, affine camera doesn't have principal point as all projection rays are parallel (camera center is infinitely far from the image plane). It has a direct impact on the form of intrinsic matrix $\mathbf{K}$ that has the following form in affine case:

$$\mathbf{K}_{/\!/} = \begin{pmatrix} \alpha f & s' & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1.9}$$

Finally, taking into account all the described properties one can deduce the total number of parameters for an affine camera:

$$\text{Number of parameters in } \mathbf{P}_{/\!/} = 3_{\text{intrinsic}} + 3_{\text{rotations}} + 2_{\text{translations}} = 8 \tag{1.10}$$

**Result 1.3: Affine camera matrix**

Affine camera matrix has 8 independent parameters: 3 for intrinsic parameters and 5 for motion.

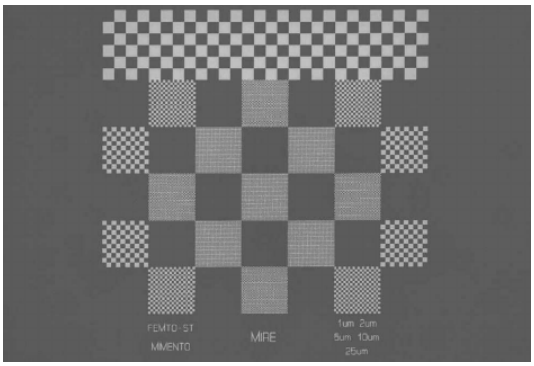



Figure 1.9: Multi-scale calibration target for SEM, square size from 1 µm up to 25 µm.

## 1.3 3D reconstruction in SEM

Before starting the description of the state of the art methods, we judged important to give some basic information about SEM calibration as for most of the algorithms this step is mandatory.

### 1.3.1 Calibration

Calibration consists in the identification of intrinsic camera parameters (either perspective or affine). Usually, it is performed using a special calibration target which is also the case of SEM (Figure 1.9).

First of all, images of the calibration target are taken from different view points (generally, up to 20). Calibration consists in finding such model parameters $\boldsymbol{\xi}$ that minimizes the error $\mathbf{e}$ between points extracted from images ($\mathbf{q}$, real projection) and their projections estimated from known 3D points of calibration target ($\hat{\mathbf{q}}$). Thus, in image $i$, the error for point $j$ is written as:

$$ {}^{i}\mathbf{e}_j = {}^{i}\mathbf{q}_j - {}^{i}\hat{\mathbf{q}}_j = {}^{i}\mathbf{q}_j - \mathbf{P}_i(\boldsymbol{\xi})\mathbf{Q}_j \tag{1.11} $$

where $\mathbf{P}(\boldsymbol{\xi})$ is the camera matrix depending on the calibration parameters $\boldsymbol{\xi}$, and $\mathbf{Q}$ is the known 3D points of calibration target. It is important to add that before subtraction in (1.11), all vectors have to be normalized, i.e., to have the element in the last row equal to one. The calibration parameters $\boldsymbol{\xi}$ are composed of intrinsic parameters (generally the same for all views) and extrinsic parameters (orientation and translation) as the motion of the target is unknown.

As a result, we are looking for calibration parameters $\boldsymbol{\xi}^*$ that minimize:

$$ \boldsymbol{\xi}^* = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \sum_{i=1}^{N_{im}} \sum_{j=1}^{N_{pts}} \left\| {}^{i}\mathbf{e}_j \right\|^2 \tag{1.12} $$

This problem may be solved by various techniques of non-linear optimization such as Levenberg-Marquardt algorithm [Mar63]. For more details see [Cor05]. An important consclusion was made by *Cui et al.* in [CM14] by comparing calibration results for FE-SEM Carl Zeiss Auriga 60 using different camera models: for magnification values bigger than ×500, instead of ×1000, a distortion-free parallel projection model (affine camera) may be used for description of image formation in SEM.

### 1.3.2 State of the art

Previous sections were devoted to background information needed to make critical judgments about current advances in the field of 3D reconstruction in SEM. The first works on 3D reconstruction appeared in the middle of XX century: in 1963 [Rob63], Roberts presented a method of 3D reconstruction from the line drawing, as a result, it was possible to generate a projection from any point of view. The main problem of 3D reconstruction is to recover the depth coordinate, i.e. $z$ component of 3D points, i.e. elevation. While different terms are used depending on the research community, we will mostly stick to the term *depth*. As the depth coordinate is lost during the process of image projection, the existing algorithms use other sources of information still contained in the images. Three main groups have emerged being based on different principles: focus, illumination, and motion. First, they will be discussed in the context of scanning electron microscopy and, after that, the analysis of their advantages and weak points contributes to the definition of the thesis goals in the last section of this chapter.

Two remarks are to be done at this point. First, there exist two-major categories of 3D reconstruction, sparse and dense. In case of *sparse* reconstruction, only local interest points are reconstructed. These are the features extracted using algorithms such as SIFT/SURF etc, or using the tracking techniques such as KLT. In contrast, *dense* methods allow computing the depth coordinate for every image point. They often include the techniques of dense matching and rectification allowing to obtain a depth (disparity) map. For dense reconstruction, the size of final point cloud may achieve millions of points while in sparse case this value typically varies from 50 to 5000.

Secondly, at this point, it is important to speak about the levels of 3D reconstruction. The first level representing the 3D object structure is the point cloud. It can be then converted to a polygon or triangle mesh to reconstruct object surface and to create CAD model. The methods of point cloud rendering are the same for all types of visual sensors and have already received much attention in past decades. For example, one can use a free open-source software MeshLad that is oriented to the management and processing of point clouds and meshes [CCC+08]. Therefore, in this work, the final result of 3D reconstruction is a dense point cloud obtained from SEM images.

#### *Shape-from-focus*

Getting back to 3D reconstruction methods, the first group is based on focus (*shape-from-focus*). The principle is the following: one needs to acquire images of the same object from the same camera position under assumption of varying focal distance or distance camera-to-object. As a result for every image, different object parts are in focus, thus, knowing the step of change, the reconstruction is obtained as a stack of slices obtained from every image. This method is especially interesting for visual sensors with low depth of field, i.e., for optical microscopes. In 1994, *Nayar et al.* presented a focus measure operator allowing the accurate estimation of focused pixels and applied it on a series of images coming from an optical microscope [NN94]. In 2013, *Marturi et al.* demonstrated the viability of shape-from-focus in SEM by obtaining the reconstruction of a microgripper [MDP13]. Authors used the normalized variance as the sharpness criteria. In order to improve the accuracy of such methods, they propose to decrease

the displacement step which can be impossible at high magnification.

#### *Shape-from-shading and photometric stereo*

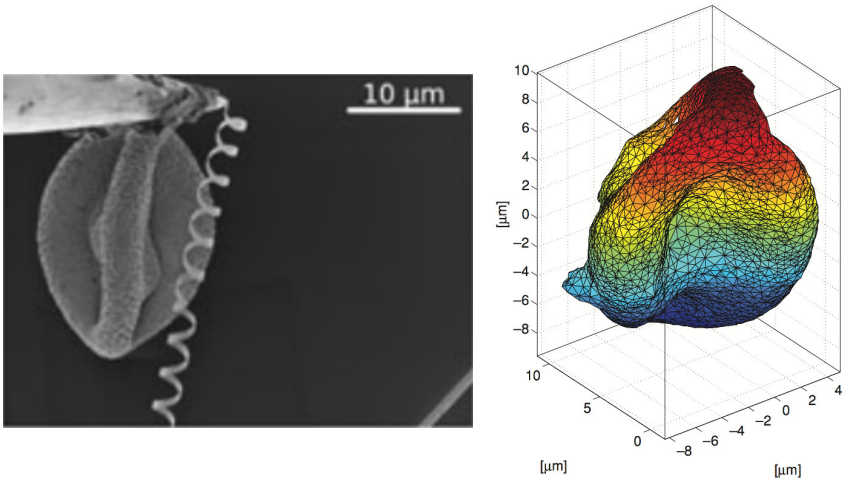
The approaches of the second group are based on illumination or light conditions, or in other words on the property of the surface to reflect light differently depending on the light incidence angle (i.e. reflectance). On the image, the lower the incidence angle, the brighter image is. This approach was first proposed by Horn in [Hor70]: the method for obtaining the shape of a smooth opaque object from one view that later received the name *shape-from-shading* (SfS). SfS was then enlarged to multiple images acquired in the following way: object and camera are fixed and only the light source is moving [Woo80]. This method is called *photometric stereo* (PS). This approach is especially used in multidetector configurations where images are obtained from different detectors at the same time. It allows simulating the movement of the light source.

For the application of the PS method to scanning electron microscopy, several methods were successfully developed and tested. *Paluszynski et al.* proposed an approach based on a photometric stereo and new numerical procedures of the signal processing [PS05]. It consists in the directional acquisition of secondary electrons generated with a known angular distribution, in a four-detector system. The method is mostly suitable for smooth surfaces with a low number of details. In [PPV08], *Pintus et al.* presented a method based on photometric stereo techniques allowing submicrometric measurements. The work is based on the capability of the SEM to use several different signals to produce images, under the same scanning conditions. Two detectors were used: standard off-axis BSE detector and the axially-detected BSE. In 2010, *Vynnyk et al.* have also applied the PS to SEM images [VSF+10]. Authors presented a new method taking into account the efficiency of the detector system through implementation of a new mathematical model of the detector signal occurrence and considering cosine Lambert's law of the electron beam distribution. In 2014, *Zhu et al.* developed a method of single-view 3D reconstruction in SEM based on shape-from-shading [ZWZ+14]: only one top-view SEM image is used to quantify the surface morphology of nanostructures in 3D. The solution is obtained by using iterative minimization of the irradiance equation for the reflectance of every image point.

#### *Structure-from-motion*

And last, but not least, the group of reconstruction algorithms based on a motion of either object or camera (both are equivalent). This approach seems very natural as the main reflex of any human being observing a new object is to take it and turn to look from different view points. The geometrical theory of structure from motion allows projection matrices and 3D points to be computed simultaneously using only corresponding points in each view. This research started with the paper of Longuet-Higgins in 1981 [Lon81] where object structure was computed from a pair of perspective views. It was next extended to multiple images in the works on factorization and Euclidean reconstruction [HZ03; TK90]. This approach is one of the most used for reconstruction in SEM.

Considering SEM, there are two ways allowing to take images from different points of view. The classical one is by moving (tilting) the stage installed in the vacuum chamber. The second one is based on deviation of an electron beam: *Jahnisch* and




(a) (b)

Figure 1.10: Results of 3D reconstruction of a pollen grain presented by *Kratochvil et al.*: a) one of the images used for reconstruction; b) sparse 3D reconstruction [KDZ+10].

*Fatikow* have developed a special hardware system for beam deflection in order to observe the sample from different angles [JF07].

In 2002, *Pouchou et al.* presented their results of 3D reconstruction in SEM for rough surfaces [PBB+02]. Authors used an image stereo pairs obtained by titlting the robotic stage, the angle of tilt is known. The 3D reconstruction is obtained using classical structure from motion techniques as the SEM was previously calibrated.

*Cornille*, in his PhD thesis, presented a very interesting method allowing to accurately calibrate the SEM, both intrinsic parameters and distortion [Cor05]. As opposed to classical calibration methods relying on a dedicated target which is usually represented as a grid of squares with known edge length, they use a randomly textured planar object (so called "speckle-pattern"). Since it is difficult to fabricate a high-quality calibration target for micro-scale, it appears to be well suited for SEM. The movement is unknown, it is then found from multiple images using fundamental and then an essential matrix that suppose that the SEM is calibrated. It is important to note that the work was realized at low magnification $\times 200$ and with perspective camera model. The resulted point cloud is then refined using accelerated bundle adjustment.

Contrary to previous works, *Kratochvil et al.* showed the results of 3D reconstruction from more than 3 images. Certainly, a larger number of viewing perspectives improves the system resolvability that improves the overall precision. However, when working with a large number of images, it demands more advanced techniques of features tracking to deal with occlusions. The proposed solution consists in the definition of visibility matrix that contains all the information about visibility/invisibility of each feature in each frame. Authors also developed a very interesting mechanichal structure allowing to simplify the acquisition of a large image dataset in SEM which is dual-charity nanobelts [DZK+09]. These structures rely on the phenomenon that if a spring is constructed with both a clockwise and counterclockwise pitch, the point at which the pitch direction changes will rotate as the ends of the spring are displaced. The results of reconstruction are demonstrated in Figure 1.10.

In 2015, *Faber et al.* presented a method for calibration-free reconstruction in SEM

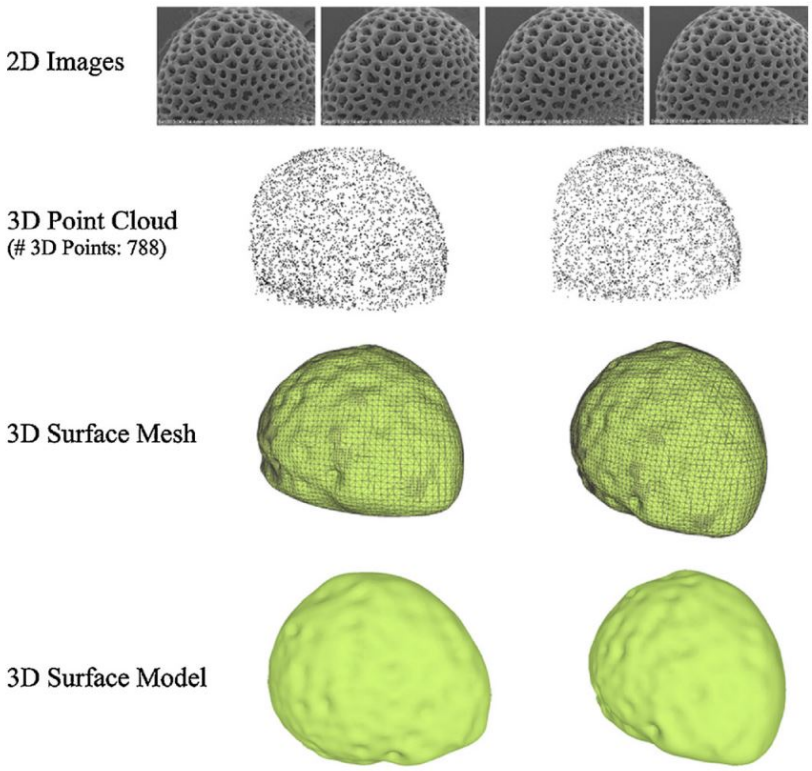



Figure 1.11: Results of 3D reconstruction of a pollen grain presented by *Tafti et al.* in [TKA+15].

for an arbitrary tilt axis [FMM+15]. They derived the expression to obtain the depth for every point of sample surface which is based on a distance from a plane: first they find a best-fit plane in 3D and then measure the distance from it that represents the depth coordinate. Digital image correlation (DIC) is used to obtain a disparity map in two image directions: horizontal and vertical. Once a dense feature set is obtained a plane is fitted to the data. Indeed, it seems to be a very promising solution for topography analysis, but not for the morphology: if the object is spherical and is mounted on a tip, there is no plane that can be used for a reference. They consider that the tilt between images is unknown and formulate the problem as a function of 7 parameters (rotations and plane parameters).

The most recent works on 3D reconstruction in SEM were presented by *Tafti et al.* in 2017 [BTO+17; TKA+15]. They also provided a 3D SEM image dataset that will be used as a part of test data in this work. In 2015, authors proposed an optimized structure from motion algorithm considering the intrinsic parameters are known. They used a population-based, stochastic optimization to refine the rotation parameters of the sample. Starting from knowing values of tilt for every image, they optimize for 7 parameters for each camera as in [FMM+15] looking for unknown rotation (axis and angle) and translation vector. In 2017, in parallel with our paper [KDP17], a dense reconstruction were obtained using iterative optimization for image rectification.

***Other***

It is also important to mention some research works that cannot be put directly in one of the presented categories. In [LLX+13], authors presented a method of 3D reconstruction based on Moiré Method. The 3D geometric model is established by combining the stereophotography technology of the SEM with traditional in-plane Moiré Method. The Virtual Projection Fringes (VPF) under different conditions are analyzed. The camera model used is the perspective one and, indeed, the method was validated at low magnifications below x1000. In [HMG+14], surface height is obtained from a pair of SEM images using stereo reconstruction based on an optical flow. The estimation was performed using differential approach starting with image smoothing. Spatial and time derivatives are then approximated and integrated to obtain a 2D flow field. For this method, it is fundamental to restrict the rotation to happen only around one axis to produce motion along a single direction, which is not always possible and highly depends on the robotic stage and on the way the object is mounted on it. *Yan et al.* presented a method of three-dimensional reconstruction using a hybrid approach: combination of shape from shading and stereoscopy [YAK17]. At the end, highly-detailed elevation maps are generated from SEM image pairs. In order to apply the proposed method, a glass sphere is used to find the relation between pixel intensities and surface normals. The size of calibration sphere is 685 µm. The presented experimental results validate the method on a millimeter sized object, thus, at low magnification. The tilt angle is also known.

## 1.4 Thesis goals

At this step, it is important to summarize the advantages and the drawbacks of each principle of 3D reconstruction. In shape-from-focus, for the reconstruction of a very small object, the precision of the focus change must be very high. Moreover, it is preferable for a visual sensor to have the lowest possible depth of field which is not the property of SEM. In fact, the best suitable value of the depth of field for shape-from-focus is lower than the focus change. Another drawback of the focus-based techniques is that lots of images are acquired for only one view point: to achieve full 3D reconstruction the number of images needed increases very fast.

The approaches based on illumination seem to be more promising as it is possible to obtain dense 3D reconstruction from one image only. However, with one image, it is not possible to obtain full 3D reconstruction as we are also limited to one point of view. In case of photometric stereo, it is mandatory to obtain images at different light conditions which is challenging in SEM if it has only one detector which is a common situation. Nonetheless, the number of views is limited to the number of installed detectors if the motion of the sample is not considered.

Finally, the structure-from-motion allows the 3D reconstruction from multiple views that can be easily obtained by moving the sample. It is important to note, that basic reconstruction from motion is sparse as it uses only the image interest points. However, it can be upgraded to dense one using the techniques of dense matching. In the ideal case, for each pair of views, thousands of points can be added to the final point cloud. Another advantage consists in the fact that adding new images to a reconstruction sequence allows adding new constraints on the final object structure,

thus to obtain more accurate results. Therefore, in this work, we are concentrated on the improvement of structure-from-motion technique and its adaptation to SEM. Therefore, we formulated the main goal of this study as follows:

**Result 1.4: Main goal of the study**

Develop a method for obtaining dense point cloud corresponding to an object with an arbitrary shape using its images taken with uncalibrated SEM at high magnification (greater than $\times 1000$).

The complexity of this task is explained by the following reasons:

*Image quality.* Several factors may be mentioned here. First, it is the high level of noise in the images that is reduced only at low scanning rates. Typically, one image acquisition may take one minute or more. Secondly, the charging effects, that are due to the build-up of electrons inside the sample, may cause a range of unusual effects such as abnormal white zones on the image, image deformation, and shift. Finally, the edge-effect, that is due to the enhanced emission of electrons from edges and peaks within the specimen which is often represented by a high brightness of object edges in the image.

*Parallel projection.* As we have already stated previously, at high magnification in SEM, the ratio between camera-to-object distance and the object size is high. As a result, the projection rays become parallel which means that the depth coordinate is completely lost during image formation: from one and even from two images, neither the object structure nor the position of the camera can be recovered.

*Calibration.* The weak point of all current techniques is the need for SEM calibration. Even if this subject is well studied, in the case of SEM, the calibration can be very complex due to the following reasons. First, in most cases, it requires a calibration object. It often means a special step of fabrication of such an object, which can be very expensive and time-consuming. Moreover, it is very difficult to guarantee the quality of its fabrication, which has a profound impact on the precision of the following image processing. Secondly, calibration is generally done offline, which can be very restrictive in some applications where the calibration object can't be placed in front of the camera once the operation started. Thirdly, which includes partially the second point, there is a problem of maintainability of calibration parameters. In order to re-calibrate the camera, the main operation task should be stopped. All these points contribute to turning attention towards the techniques of auto- or selfcalibration.

*Motion estimation.* Most works on 3D reconstruction in SEM are based on the principle that the motion of the camera from one image to another is known. It often implies the use of the internal sensors of the robot on which the sample is located. However, when working at small scales the level of confidence of micropositionning systems decreases. Moreover, in case of the complex movement (movement of more than one axis), the measured displacement of robot joints should be transformed into combined motion of the sample and, for this, the precise knowledge of robot forward kinematics is needed. Therefore, in this work, the camera motion will be considered as unknown.

*Image acquisition.* All 3D reconstruction algorithms use images as the input and, in case of structure-from-motion, images taken from different view-points. While this task is relatively easy to control with a classic camera, the situation is much more

complicated in case of SEM. Even an experienced operator may spend hours of work to acquire only ten images of a small object. This is mostly due to a very limited field of view at high magnification: even a small displacement of a robot arm makes the object leave the image.

# Chapter 2

# Motion estimation

## Contents



*In previous chapter, we presented some background information about image formation and camera models. As a result, for magnifications bigger than* $\times 1000$*, an affine camera is actually a valuable assumption for SEM. Next, a brief analysis of the current state of the art techniques showed that the main locks of 3D reconstruction in SEM are the quality of calibration and motion estimation. Both these parameters are incorporated in the camera matrices as a matrix of intrinsic parameters and a matrix of extrinsic or motion parameters. In this chapter, we present some algorithms of partial and full motion estimation for both translation and rotation. We start with the analysis of motion geometry for the minimal case of two images and then extend it to three images. It should be added that the geometry between views is subject to a set of constraints encapsulated in a so-called fundamental matrix and, at the end of the chapter, several methods of its robust estimation will be given.*

## 2.1 Detection and matching of interest points

For many computer vision applications, we need to find the relation between images taken from different perspectives. In other words, we need to find the movement of image points in order to make assumptions about the object motion in the future. This task is often defined as feature tracking (such as KLT [TK91]) or feature matching. These two terms are not interchangeable: tracking refers more frequently to videos or image sequences with very small object displacements. In contrast, feature matching is used for all other tasks: stereo-image pairs, complex contrast change etc. Their scope of application is different, the goal, however, is still the same: find similar points between two images. As a point does not contain much information (only one value of intensity), it is characterized by a region around it which is referred to as window. Its usual size is $5 \times 5$ pixels. It is important to notice that every image point is not suitable for tracking (or matching) as many points look alike. Therefore, it is important to find an algorithm allowing to detect image points that can be easily distinguished from others. Such points are called features or interest points and they are mainly characterized by strong contrast variations of pixel intensities in their neighbourhood.

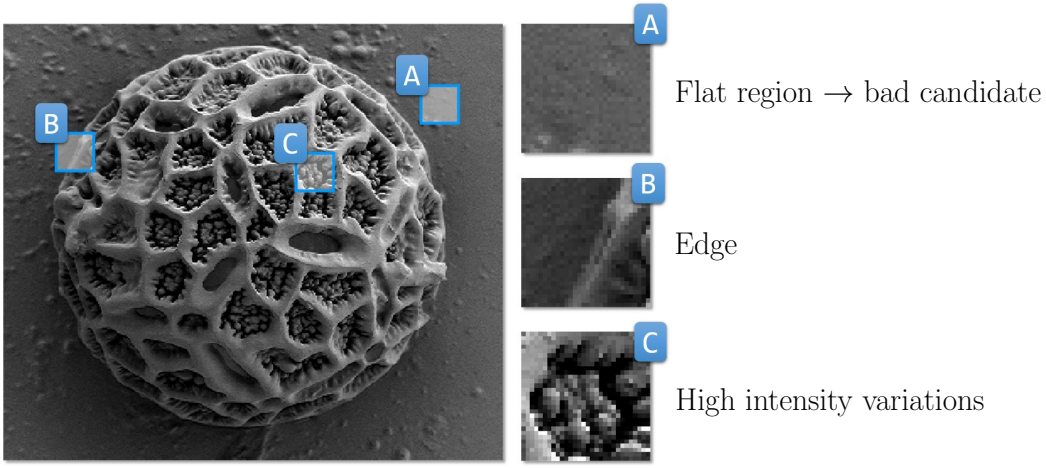



Figure 2.1: Examples of feature detection on SEM image.

Different examples related to feature detection are shown in Figure 2.1. As it can be seen, in the first case (Figure 2.1,a), the window represents a flat region corresponding to background which has no contrast variations, therefore, it is a bad candidate for tracking/matching. In the second case, the image contains an edge that is difficult to distinguish from another region obtained by moving the window across the edge. This property is used for edge tracking which goes out of the scope of this work. Finally, the last example (Figure 2.1,c) with high intensity variations represents a good candidate. These examples give an intuition behind feature detection and, obviously, nowadays, there exist a number of algorithms allowing automatic detection of interest points.

The existing algorithms of automatic feature detection (detectors) perform the scanning of image with a window of given size and decide whether a particular region is distinctive enough to consider it as a feature. Thus, the detectors allow to attribute a *distinctiveness* score to every block of pixels. Among them are Harris corner detector [HS88], FAST corner detection [RD05], Difference-of-Gaussians [Low04], etc.

Once the interest points are detected, the next step is feature description which consists in deciding how to represent the image information inside each region for later matching [Rad13]. Description algorithm should be designed in a way that the descriptor has the same values for all projections of the same 3D point under different viewing conditions: camera motion, illumination change etc. The easiest way consists in converting the interest point region to a vector. Then, two of such vectors coming from different images can be compared to find a match. However, this approach is not suitable for considerable geometric transformation such as complex rotations or changes in scale. Thus, much more robust solutions appeared recently such as SIFT [Low04] or SURF [BET+08] descriptors. In this work, we will use AKAZE algorithm for feature detection and description [ABD12; ANB11] which has a better performance comparing to the state-of-the-art SIFT/SURF techniques. Moreover, it is a part of free open-source OpenCV library [Bra00].

At this point, having applied AKAZE for feature detection and description, we have a set of feature points for every image of the sequence that now have to be matched. In order to find a correspondence, feature descriptors are compared using a matching algorithm. The simplest one consists in computing the difference between values of two vectors using either sum of absolute differences (SAD) or sum of squared differences (SSD). More efficient approaches are based on a nearest neighbor search [HAA16]: the randomized k-d forest and the fast library for approximate nearest neighbors (FLANN) [ML09; ML14]. The last one is used in this work. An example of feature matching applied on SEM images is presented in Figure 2.2.

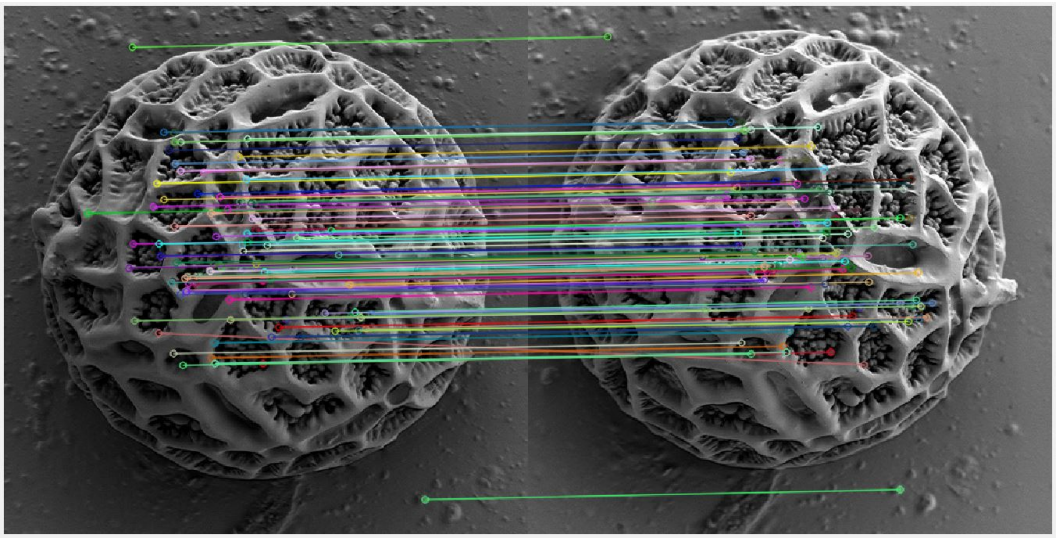


Figure 2.2: Example of feature matching on SEM images.

Mathematically, a set of feature correspondences is represented as a matrix $\mathcal{W}$ that will be further referred as *measurement matrix*:

$$\mathcal{W} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \vdots \\ \mathbf{W}_N \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1^1 & \mathbf{q}_2^1 & \dots & \mathbf{q}_P^1 \\ \mathbf{q}_1^2 & \mathbf{q}_2^2 & \dots & \mathbf{q}_P^2 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{q}_1^N & \mathbf{q}_2^N & \dots & \mathbf{q}_P^N \end{bmatrix} \tag{2.1}$$

with

$$\mathbf{W}_i = \begin{bmatrix} \mathbf{q}_1^i & \mathbf{q}_2^i & \dots & \mathbf{q}_P^i \end{bmatrix} \tag{2.2}$$

where $\mathbf{W}_i$ is the set of features extracted from the $i$-th image, a vector $\mathbf{q}_1^3 = (q_x, q_y, 1)^\top$ represents pixel coordinates of first feature extracted from the third image, $N$ is the number of images, $P$ is the number of features.

Several remarks are to be done about the measurement matrix:

- the number of columns is equal to the number $P$ of extracted features,
- one column corresponds to $N$ (number of images) projections of one 3D point,
- the number of rows is equal to $3N$ in homogeneous coordinated and to $2N$ in non-homogeneous coordinates (rows with ones are omitted),
- all features are viewed in all images.

In case where the features are not viewed in all images, we speak about *full measurement matrix* $\mathcal{W}_f$ which is filled with zeros if the features are not detected on the corresponding image. During its formation, the condition is that the feature must be present at least in two images. The full measurement matrix may have the following form:

$$\mathcal{W}_f = \begin{bmatrix} \mathbf{q}_1^1 & \mathbf{q}_2^1 & \dots & \mathbf{0} \\ \mathbf{q}_1^2 & \mathbf{q}_2^2 & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{q}_2^3 & \dots & \mathbf{q}_P^3 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{q}_P^N \end{bmatrix} \tag{2.3}$$

Here, the first feature is viewed in first two images, the second feature in images (1,2,3), etc. The measurement matrix can be easily obtained from its full form by suppressing the columns containing zeros.

It is worth noticing that for the remainder of this work, all the information about the motion and structure as well as calibration parameters will be obtained from the the measurement matrix only.

## 2.2 Camera modelling

At this point, the measurement matrix is obtained and we know that the displacement of 2D projections is due to the object motion (the motion of the robotic stage). However, for the remainder of the manuscript we will consider that the camera is moving and the object is fixed. Both situations are geometrically equivalent which is demonstrated in Appendix B, yet, the latter one simplifies a lot the presentation. We also know that a sequence of images is obtained using an affine camera, i.e., under assumption of parallel projection.

Concerning the subject of motion estimation, it is known that motion parameters as well as the camera intrinsic parameters are contained in the camera matrix which has a special form in affine case [HZ03]:

$$\mathbf{P}_{/\!/} = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ 0 & 0 & 0 & * \end{pmatrix} \tag{2.4}$$

As it is defined up to scale, the matrix has 8 independent parameters. Moreover, it can be further decomposed as:

$$\mathbf{P}_{/\!/} = \underbrace{\begin{pmatrix} \alpha f & s & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{K}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0\ 0\ 0 & 1 \end{pmatrix} \tag{2.5}$$

$$= \underbrace{\begin{pmatrix} \mathbf{A}_{2\times 2} & 0 \\ 0 & 1 \end{pmatrix}}_{\mathbf{K}} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.6}$$

$$= \begin{pmatrix} \mathbf{A}_{2\times 2}\mathbf{R}_{2\times 3} & \mathbf{A}_{2\times 2}\bar{\mathbf{t}}_{2\times 1} \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} \tag{2.7}$$

$$= \begin{pmatrix} \mathbf{M}_{2\times 3} & \mathbf{t}_{2\times 1} \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} \tag{2.8}$$

Thus, the 2D projection $\mathbf{q}$ of 3D point $\mathbf{Q}$ is written as (in homogeneous coordinates):

$$\mathbf{q} = \mathbf{P}_{/\!/}\mathbf{Q} \tag{2.9}$$

or, in non-homogeneous coordinates:

$$\mathbf{q} = \mathbf{M}_{2\times 3}\mathbf{Q} + \mathbf{t}_{2\times 1} \tag{2.10}$$

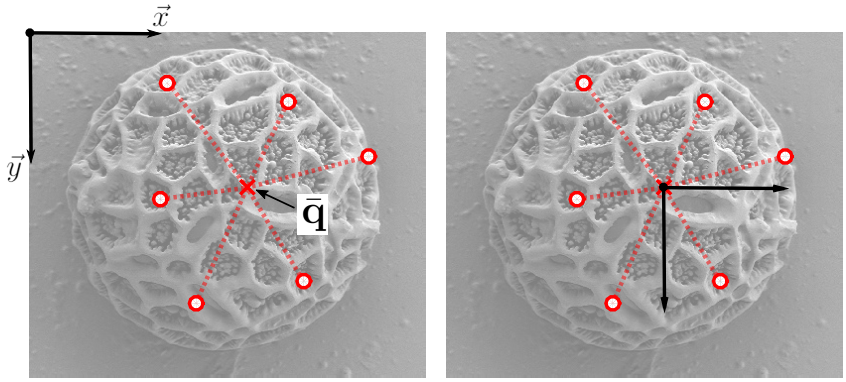
which is equivalent to

$$\begin{pmatrix} q_x \\ q_y \end{pmatrix} = \begin{pmatrix} \alpha f & s \\ 0 & f \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \end{pmatrix} \begin{pmatrix} Q_x \\ Q_y \\ Q_z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{2.11}$$

This final form allows an easy definition of all 8 parameters of the affine camera:

- motion parameters (5):
  - 2 translations: only in-plane translations are present which proves once again that the image projections obtained with affine camera are invariant to camera-object distance, $t_z$ (Result 1.2.2).
  - 3 rotation parameters for 3D rotation matrix: only two rows are used, the third one can be obtained as a cross product of the first two.
- intrinsic parameters (3): $\alpha$ is the aspect ratio; $s$ is the skew parameter; $f$ is an overall scale factor, i.e. $\frac{pixel}{m}$ ratio.

Therefore, we aim to identify all the unknowns from measurement matrix only. This Chapter is devoted to the estimation of motion parameter while the estimation of intrinsic parameters is done in Chapter 3.




a) Standard image coordinates b) Relative coordinates

Figure 2.3: Transformation from standard image coordinates to relative ones.

## 2.3 Estimating translation

The first step in identification of affine camera motion parameters consists in eliminating translation components which can be done by using relative coordinates. Remember that affine camera projections are invariant to in-plane translations (Result 1.2.2). It means that we may translate the centroid of 2D points into $(0,0)^\top$. It is equivalent to the situation presented in Figure 2.3 with

$$\mathcal{W}_r = \mathcal{W} - \begin{pmatrix} \bar{\mathbf{q}}^1 \\ \bar{\mathbf{q}}^2 \\ \vdots \\ \bar{\mathbf{q}}^N \end{pmatrix} \tag{2.12}$$

where $\bar{\mathbf{q}}^i$ is the centroid of points in $i$-th image defined as:

$$\bar{\mathbf{q}}^i = \frac{1}{P}\sum_{j=1}^{P} \mathbf{q}_j^i \tag{2.13}$$

Finally, for the $i$-th camera:

$$\mathbf{t}^i = \begin{pmatrix} t_x \\ t_y \end{pmatrix} = \bar{\mathbf{q}}^i \tag{2.14}$$

Thus, in relative coordinates the projection equation for affine camera (2.10) is reduced to:

$$\tilde{\mathbf{q}} = \mathbf{A}\mathbf{R}_{2\times 3}\tilde{\mathbf{Q}} \tag{2.15}$$

$$= \mathbf{M}_{2\times 3}\tilde{\mathbf{Q}} \tag{2.16}$$

where $\sim$ symbol denotes points expressed in relative coordinates. For the remainder of the Chapter we consider that relative coordinates are used and $\sim$ symbol will be omitted.

## 2.4 Estimating rotation

The estimation of rotation parameters is a crucial step for a variety of robotics and computer vision applications including 3D reconstruction. Recent works demonstrated that it is possible to extract the rotations from SEM images but only with some additional information, such as focus or known object structure. These methods can be subdivided in two main groups.

The first group of motion estimation techniques is based on pose computation. In [CMH+16], positions along $x$- ($t_x$) and $y$-axis ($t_y$), and rotation about the optical axis ($R_z$) were computed using Gauss-Newton method by minimizing the sum of the projection errors of some points of a pre-defined 2D model. Authors assumed that the object, while moving, stays on the plane parallel to the image plane. However, if the object performs a more complex 3D movement, the results may be inaccurate as the impact of rotations $R_x$ and $R_y$ cannot be neglected. In [KDN09], the developed solution based on augmented reality approach used the search for 3D CAD model in the 2D images of SEM by minimizing the distance between lines extracted from images and those of the model. The pose, comprising position and orientation, was computed efficiently, but the method works only for polyhedral structures. In another example, while working on 3D reconstruction in SEM, Tafti *et al.* obtain the full object motion information but the tilt angle of the stage and SEM calibration matrix were known [TKA+15].

The second group of motion estimation methods is based on the matching of a pre-defined 2D model. In [SF06] and [RZH+12], cross-correlation was used, whereas in [FWH+07] model was represented by active contours and motion was estimated from the minimization of the active contours and the object detected edges. Both methods have the same drawback of [CMH+16]: as the rotations $R_x$ and $R_y$ are not considered, measured quantities may be inaccurate. In [MTD+16] an interesting method was proposed, it is based on the work described in [KR08] which used spherical Fourier transform to compute the rotations $R_x$, $R_y$ and $R_z$. The method is promising, however, it was only tested on simulated SEM images and not in real conditions.

In this chapter, we describe a method allowing to recover full 3D rotation of the camera relying on the obtained images only, without using supplementary sensors. At our knowledge, this way of finding rotations have never been presented in microscopy, and, in particular, in SEM community. However, the estimation of object motion and structure for affine camera received much attention in computer vision community in 90's [Pri96; SK97; TK92]. One of the first and the most representative work is the paper of *Koenderink* and *Van Doorn* [KV91] with a deep analysis of two-view relations. Another outstanding work was done by *Shapiro et al.* in [SZB95]: authors developed an iterative approach allowing to find the motion from three-views. Most of these methods were tested only on synthetic data and have never been applied to SEM images. The approach of full 3D rotation estimation presented here is direct and based on spherical trigonometry. Moreover, the way of rotation decomposition shown below is the most suitable for such steps of 3D reconstruction as autocalibration (Chapter 3) and dense matching (Chapter 4). In addition, camera calibration is not needed.

The coming sections have the following structure:

- first, we present a geometric interpretation of camera rotation (Section 2.4.1) and its decomposition in three elements in a way suitable for further identification;

- secondly, we analyze the minimal configuration where only two images are available (Section 2.4.2). The conclusion is that only two rotation angles out of three may be recovered due to the *bas-relief* ambiguity (Section 2.4.3);
- finally, by adding a third image, full 3D rotation is estimated using spherical trigonometry (Section 2.4.4).

### 2.4.1 Rotation matrix decomposition

Before starting to work with images, it is important to understand under what conditions they were formed, i.e., to understand the properties of affine camera that will be used further for the derivation of geometry between views:

- affine camera is invariant to the motion along the optical axis of either object or camera itself (Result 1.2.2);
- affine camera is invariant to translations in image plane (Result 1.2.2);

These two properties allow to draw the following conclusion: for the views taken with the same affine camera, the camera centers are on the surface of a sphere if relative coordinates are used.

**Result 2.1: Geometry of multiple affine views**

For the views taken with the same affine camera, the camera centers are on the surface of a sphere if relative coordinates are used. And this sphere corresponds actually to the plane at infinity.

**Proof**. Sphere is defined as a surface with every point equidistant from its center. As it was already mentioned, the affine camera is invariant to the camera-object distance and its center is at infinity for all orientations. Thus, all camera centers are on the sphere with infinite radius. Moreover, when relative coordinates are used, the optical axis passes through the world origin [Qua96]. All these elements lead to the situation presented in Figure 2.4.

From the Figure 2.4, one can remark that the rotation matrix relating a pair of views $(\mathbf{I}, \mathbf{I}')$ may be decomposed as follows:

1. $\theta$, angle between $\vec{x}$ axis of the first frame $\mathbf{I}$ and the plane $COC'$;
2. $\rho$, angle of out-of-plane rotation around the axis perpendicular to plane $COC'$;
3. $\theta'$, angle between $\vec{x}$ axis of the second frame $\mathbf{I}'$ and the plane $COC'$.

All these steps are represented in Figure 2.5. Finally, for any pair of images taken with affine camera, the rotation can be written as:

$$\mathbf{R} = \mathbf{R}_z(\theta')\mathbf{R}_y(\rho)\mathbf{R}_z^\top(\theta) \tag{2.17}$$

The following sections are devoted to methods allowing the identification of these angles from images only.

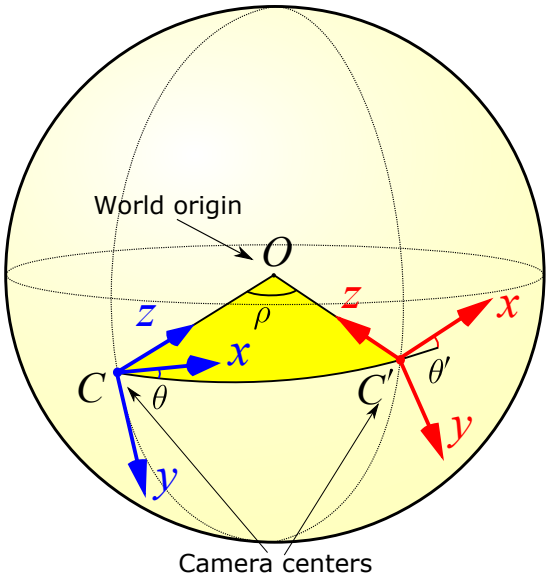



Figure 2.4: Geometry of motion between two views taken with an affine camera ($\mathbf{C}$ and $\mathbf{C}'$). The motion between cameras is a composition of three rotations with angles $\theta, \rho$, and $\theta'$.

### 2.4.2 Two-view geometry

Assume two images $(\mathbf{I}, \mathbf{I}')$ were taken with the same but moving affine camera. Obviously, there is a strong link between them. Consider first that the images were taken with a perspective camera (Figure 2.6,a):

A projection of 3D point $\mathbf{Q}$ on the first image is given by the intersection of the projection ray with the image plane. Projection ray, is a ray that connects the 3D point with the camera center $\mathbf{C}$. Remark that all 3D points lying on the projection ray $\vec{CQ}$ will give exactly the same projection on the first image. However, in the second image, this set of points projects into a line $\mathbf{l}'$. This is also true for the inverse situation, with the line $\mathbf{l}$ in the first image. These properties form an epipolar geometry, and lines $\mathbf{l}$ and $\mathbf{l}'$ are called epipolar lines. Algebraically, epipolar geometry is represented by a $3 \times 3$ matrix $\mathbf{F}$, fundamental matrix, allowing to define the epipolar constraint (in homogeneous coordinates)[1]:

$$(\mathbf{q}')^\top \mathbf{F} \mathbf{q} = 0 \tag{2.18}$$

and

$$\begin{cases} \mathbf{l} = \mathbf{F}^\top \mathbf{q}' \\ \mathbf{l}' = \mathbf{F}\mathbf{q} \end{cases} \tag{2.19}$$

where $\mathbf{l}$ represents the line $l_1 x + l_2 y + l_3 = 0$, the same is true for $\mathbf{l}'$.

In case of affine camera the situation is slightly different as camera centers are at infinity (Figure 2.6,b). Actually, as the projection rays are parallel, all epipolar lines are parallel, and, which is of high importance, the epipolar planes are also parallel to

[1]Fundamental matrix is estimated from a set of point correspondences, i.e., from measurement matrix $\mathcal{W}$ in Section 2.6

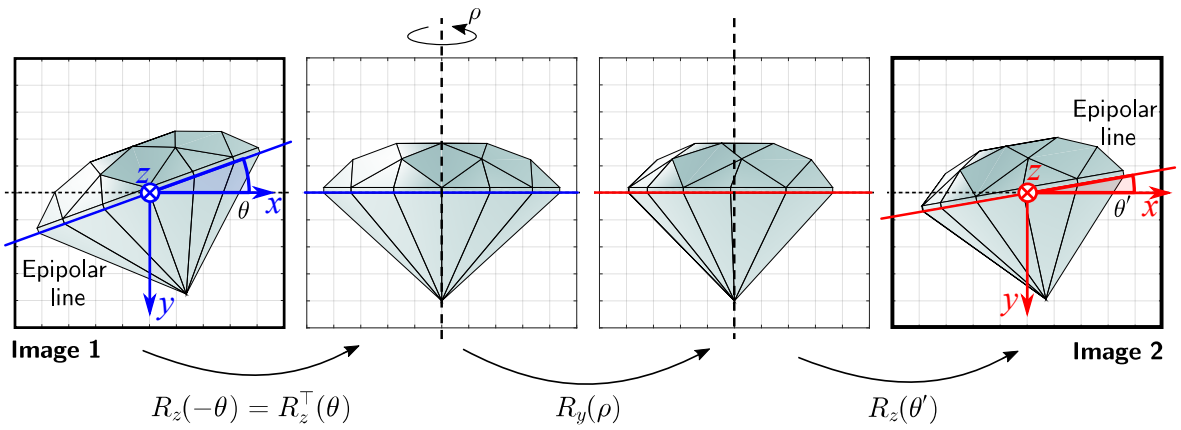
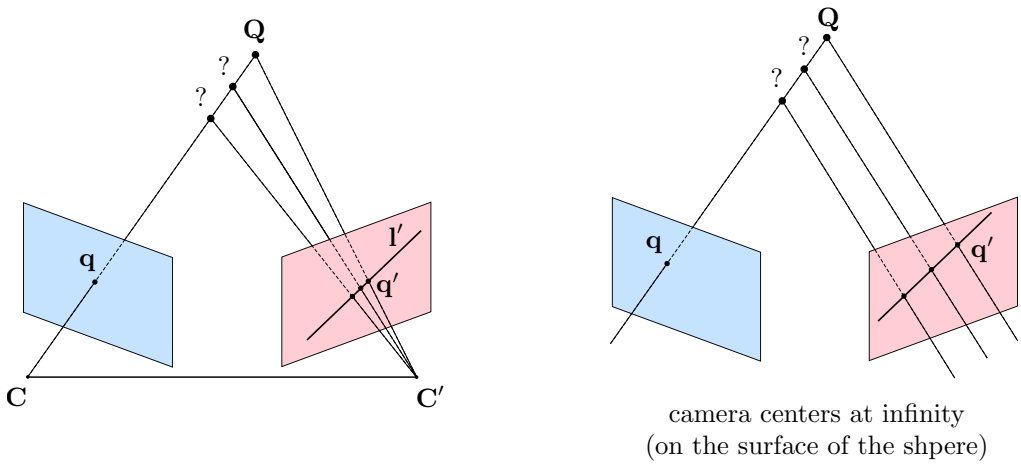



Figure 2.5: Rotations decomposition between two images taken with an affine camera (relative coordinates are considered).




(a) Perspective (b) Affine

Figure 2.6: Epipolar geometry.

each other and to the plane $COC'$ (Figure 2.4). Therefore, the epipolar line passing through the point $(0,0)^\top$ is the projection of the plane $COC'$ to the corresponding image frame (as in Figure 2.5). As a result, the slopes of epipolar lines define exactly the angles $\theta$ and $\theta'$.

As in (2.19), the epipolar lines can be identified from the fundamental matrix that has a particular form for affine camera [SZB95]:

$$\mathbf{F} = \begin{pmatrix} 0 & 0 & a \\ 0 & 0 & b \\ c & d & e \end{pmatrix} \tag{2.20}$$

where $a$,$b$,$c$,$d$, and $e$ are real numbers. Using the expression for epipolar lines (2.19), for an affine camera, the following can be written:

$$\begin{cases} \mathbf{l} = (c, d, aq'_x + bq'_y + e)^\top \\ \mathbf{l}' = (a, b, cq_x + dq_y + e)^\top \end{cases} \tag{2.21}$$

It proves once again, that for all image points, the epipolar lines are parallel as their slopes do not depend on point coordinates.

As a result, the slopes of epipolar lines, $\theta$ and $\theta'$ are:

$$\begin{cases} \theta = \arctan\left(-\frac{d}{c}\right) \\ \theta' = \arctan\left(-\frac{a}{b}\right) \end{cases} \tag{2.22}$$

where $a$,$b$,$c$,$d$ are the elements of affine fundamental matrix. Visibly, the precision of angle's estimation highly depends on the quality of $\mathbf{F}$ and in Section 2.6, we present and compare several algorithms allowing accurate and robust estimation of fundamental matrix.

Next step of rotation estimation would be the estimation of out-of-plane rotation $\rho$, however, it is impossible to do it from two images only due to the *bas-relief ambiguity* to which the next section is devoted.

### 2.4.3 Bas-relief ambiguity

When an unknown object is viewed under parallel projection, there is an ambiguity in determining its 3D structure and motion. Actually, from two views of an object with depth variation $\Delta Z_1$ rotating to $\rho_1$ is indistinguishable from the object with depth $\Delta Z_2$ rotating to $\rho_2$ (Figure 2.7). In other words, a shallow object experiencing a large turn (i.e. small $\Delta Z$ and large $\rho$) generates the same image as a deep object experiencing a small turn (i.e. large $\Delta Z$ and small $\rho$) [HZ03].

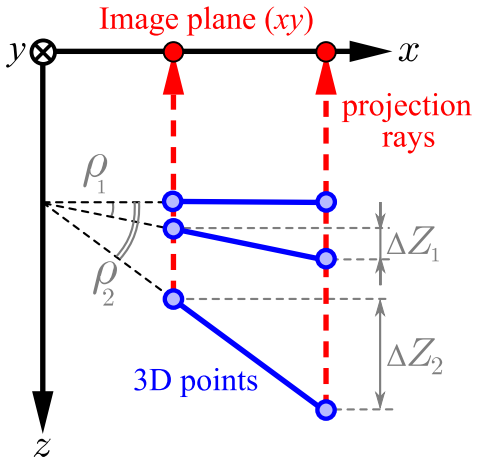



Figure 2.7: Bas-relief ambiguity.

Therefore, from two affine views, the scene (motion and structure) may be recovered **only** up to the combination of depth variation and out-of-plane rotation $\rho$ [SZB95].

**Result 2.2: Bas-relief ambiguity**

From two affine views, the scene (motion and structure) may be recovered **only** up to the combination of depth variation and out-of-plane rotation $\rho$.

**Proof**. Consider two images of a 3D point $\mathbf{Q}$ taken with the same affine camera from different view points: $\mathbf{P}$ and $\mathbf{P}'$. As a first step, the problem may be simplified

assuming that the cameras are in canonical forms, so that:

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{P}' = \begin{pmatrix} \mathbf{R}' & \mathbf{t}' \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix}$$

where $\mathbf{R}'$ corresponds to the first two rows of the rotation matrix. Translation $\mathbf{t}'$ may be considered null by using relative coordinates.

Knowing that the projection of 3D point $\mathbf{Q}$ is:

$$\mathbf{q} = \mathbf{PQ}$$

with $\mathbf{P}$ in canonical form we obtain:

$$\{Q_x = q_x, Q_y = q_y\} \tag{2.23}$$

Assume the simplest situation where second camera performed a pure rotation by an angle $\rho$ around $\vec{y}$ axis, so that:

$$\mathbf{R}' = rot_y(\rho) = \begin{pmatrix} \cos(\rho) & 0 & \sin(\rho) \\ 0 & 1 & 0 \\ -\sin(\rho) & 0 & \cos(\rho) \end{pmatrix}$$

Finally, we can write

$$\begin{pmatrix} q'_x \\ q'_y \end{pmatrix} = \begin{pmatrix} \cos(\rho) & 0 & \sin(\rho) \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} Q_x \\ Q_y \\ Q_z \end{pmatrix}$$

that gives a system with two equations:

$$\begin{cases} q'_x = \cos(\rho)Q_x + \sin(\rho)Q_z \\ q'_y = Q_y \end{cases} \tag{2.24}$$

or, by substituting (2.23) in (2.24):

$$\begin{cases} q'_x = \cos(\rho)q_x + \sin(\rho)Q_z \\ q'_y = q_y \end{cases} \tag{2.25}$$

As a result, we see that $x$-projection is non-linear and depend on a term "$\sin(\rho)Q_z$", i.e. on a combination of out-of-plane rotation (one for each image) and object depth variation (one for each correspondence). It means that, from two images, it is impossible to obtain 3D reconstruction if angle $\rho$ and depth variation $Q_z$ are both unknown.

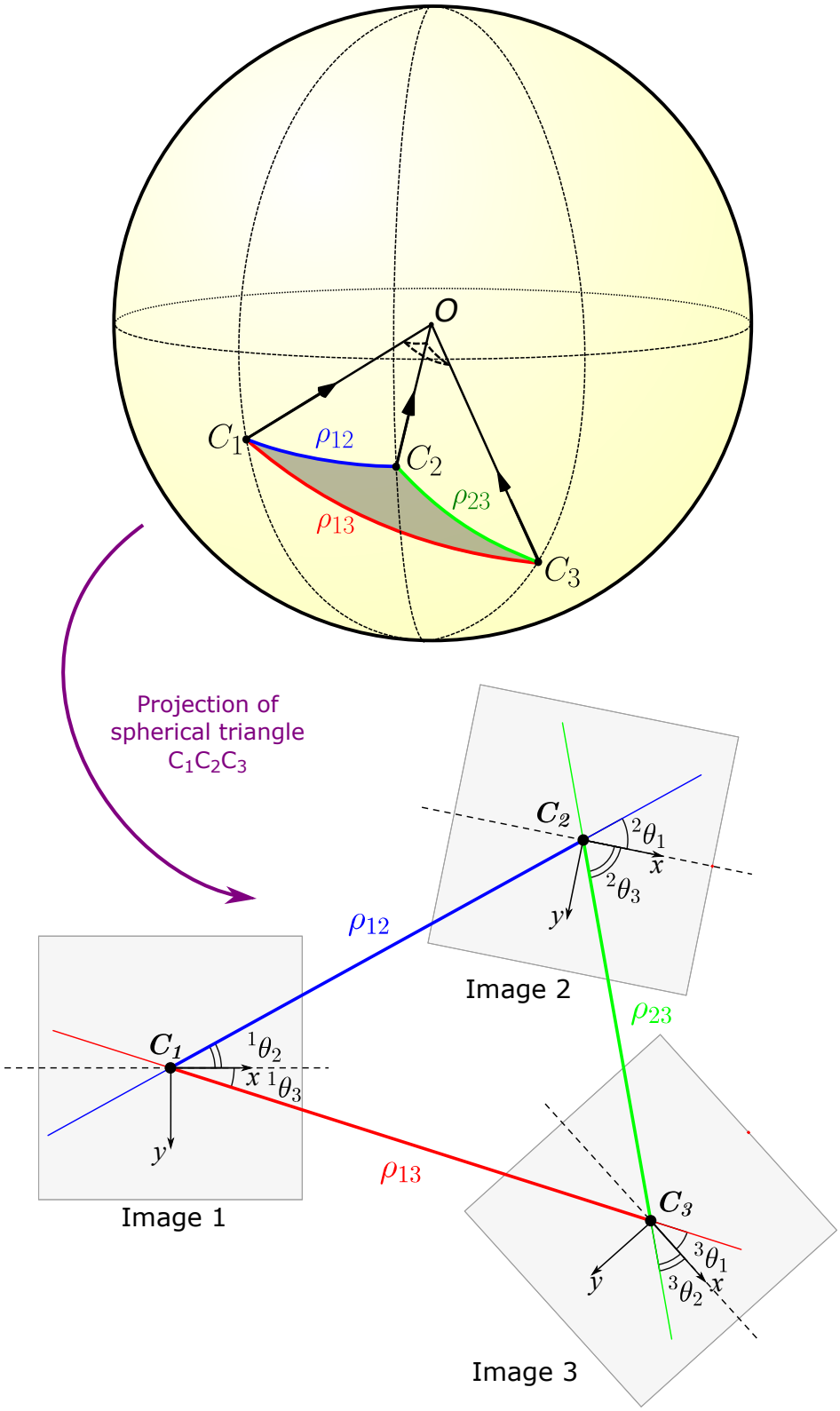



Figure 2.8: Three-view geometry: camera centers $C_1$, $C_2$ and $C_3$ are located on the surface of a unit sphere and form a spherical triangle.

### 2.4.4 Three-view geometry

To avoid problems related to bas-relief ambiguity, we propose to add one more image that will add new constraints on both 3D structure and motion. Three view configuration in case of an affine camera is represented in Figure 2.8. Assuming that relative coordinates are used, we may consider that the origins of all image frames lie on the sphere with the world origin $O$ as its center. The polygon $C_1C_2C_3$ (on the surface of the sphere) represents a spherical triangle as it is formed by intersection of three great circles, one for each pair of frame centers. Great circle is a circle that has the same radius as the sphere. It means that it is possible to use the whole branch of geometry describing the relations between sphere elements, spherical geometry and trigonometry. Spherical triangle has the following properties:

1) Angles $\sphericalangle C_1, \sphericalangle C_2, \sphericalangle C_3$ (symbol $\sphericalangle$ denotes spherical angle). The angles of the triangle are equal to the the angles between the tangent vectors of the great circle arcs where they meet at the vertices. In our case, it represents the angle between epipolar lines from two other images, e.g., the angle $\sphericalangle C_1$ is measured in the first view as the angle between epipolar lines is defined by images 2 and 3 (Figure 2.8):

$$\sphericalangle C_1 = {}^1\theta_2 - {}^1\theta_3 \tag{2.26}$$

where ${}^i\theta_j$ is the slope of epipolar line in image $i$ defined by image $j$. These angles are found using (2.22).

2) Sides $C_1C_2$, $C_1C_3$ and $C_2C_3$. According to the theory of spherical geometry, the lengths of the sides of spherical triangle are numerically equal to the radian measure of the angles that the great circle arcs subtend at the centre. It means that, revising our configuration, one can conclude that the sides of the triangle are equal to the angles $\rho$ that could not be measured in two-view case:

$$\rho_{12} = \angle C_1OC_2 \tag{2.27}$$

Thus, we have all the elements of the triangle expressed using rotational parameters of camera positions:

$$\begin{aligned} \sphericalangle C_1 &= {}^1\theta_2 - {}^1\theta_3 \\ \sphericalangle C_2 &= {}^2\theta_1 - {}^2\theta_3 \\ \sphericalangle C_3 &= {}^3\theta_1 - {}^3\theta_2 \end{aligned} \tag{2.28}$$

and

$$\begin{aligned} \rho_{12} &= C_1C_2 = \angle C_1OC_2 \\ \rho_{13} &= C_1C_3 = \angle C_1OC_3 \\ \rho_{23} &= C_2C_3 = \angle C_2OC_3 \end{aligned} \tag{2.29}$$

In the presented configuration, the angles of the spherical triangle are recovered using image pairs and (2.22). The problem of solving spherical triangle, with its angles known, is quite common and can be solved by applying a supplemental cosine law of spherical trigonometry. Using presented notations, it has the following form:

$$\begin{aligned} \rho_{12} &= \arccos\left(\frac{\cos(C_3)+\cos(C_1)\cos(C_2)}{\sin(C_1)\sin(C_2)}\right) \\ \rho_{13} &= \arccos\left(\frac{\cos(C_2)+\cos(C_1)\cos(C_3)}{\sin(C_1)\sin(C_3)}\right) \\ \rho_{23} &= \arccos\left(\frac{\cos(C_1)+\cos(C_2)\cos(C_3)}{\sin(C_2)\sin(C_3)}\right) \end{aligned} \tag{2.30}$$

Thus, all rotational parameters are recovered: slope angles ${}^{j}\theta_i$ were calculated using (2.22) and $\rho_{ij}$ using (2.30). The rotation ${}^{j}\mathbf{R}_i$ matrix can be then obtained by substituting these values in (2.17). The final algorithm is summarized in Algorithm 1.

**Algorithm 1** 3D Rotation estimation for SEM from three images

1: acquire three images of the scene with different orientations: $\mathbf{I}_1$, $\mathbf{I}_2$, $\mathbf{I}_3$
2: extract and match features (at least four correspondences are needed)
3: find fundamental matrices $\mathbf{F}_{12}, \mathbf{F}_{13}, \mathbf{F}_{23}$ (Section 2.6)
4: estimate slope angles (2.22)
5: solve spherical triangle $\mathrm{C}_1\mathrm{C}_2\mathrm{C}_3$ (2.28, 2.30)
6: Full 3D rotation matrices are:

$$\begin{cases} \mathbf{R}_1 = \mathbf{I}_{3\times 3} \\ \mathbf{R}_2 = \mathbf{R}_z({}^{2}\theta_1)\mathbf{R}_y(\rho_{12})\mathbf{R}_z^{\top}({}^{1}\theta_2) \\ \mathbf{R}_3 = \mathbf{R}_z({}^{3}\theta_1)\mathbf{R}_y(\rho_{13})\mathbf{R}_z^{\top}({}^{1}\theta_3) \end{cases}$$

## 2.5 Experimental validation

In order to evaluate the performance of the proposed method, two types of experiment were conducted. First, the method was tested on the manually generated sequence of virtual images using MATLAB. Secondly, two image datasets coming from SEM Carl Zeiss Auriga 60 were used. The features were obtained using AKAZE descriptors [ANB11] from OpenCV library [Bra00] and then matched (Section 2.1). The fundamental matrices were obtained using the algorithm presented in Section 2.6.

### 2.5.1 Synthetic images

Virtual image sequence represents an image set containing 150 images of a diamond (Figure 2.13) with predefined pose. The orientation of the object (diamond) was estimated for all frames using the method presented above. The resulting graphs (Figure 2.9) allow to compare the estimated values with the predefined ones. As a result the error stays inferior to 1 microdegree for all orientation Euler angles, which allows first validation of the method. It is important to notice that it is possible to measure object orientation for the second image but only at the moment when the third one becomes available. This fact is also reflected in Figure 2.9.

### 2.5.2 SEM images

Estimation of camera rotations was conducted on two SEM image datasets. First, seven images of *Potamogeton*, a pollen grain of an aquatic plant acquired with Carl Zeiss AURIGA 60 FE-SEM (Figure 2.10,a). The rotation was performed by tilting the stage of 3 degrees for every image. Second, *End effector* dataset which contains seven images of a tip of the end effector of a microgripper (Figure 2.10,b). The movement between images was realized using a 6-DoF robot mounted inside the microscope, first, by the steps of 3 degrees about each axis and then by 5 degrees. Features were extracted and matched, the fundamental matrices were found using Gold Standard algorithm

inside MLESAC scheme (Section 2.6) and the rotations were then recovered using the presented method. The estimated values are presented in Table 2.1. Obtained angles are very close to true ones with a deviation inferior to 0.3 degree.

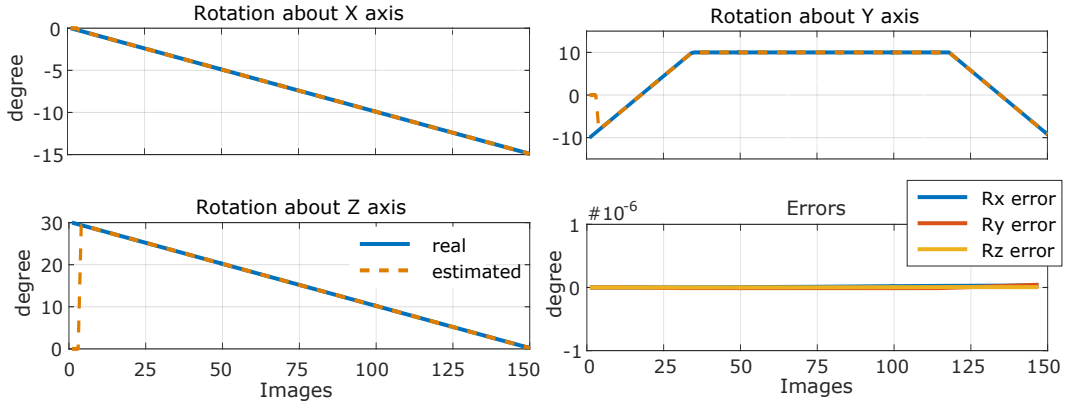



Figure 2.9: Errors in orientation measurement for virtual image sequence of a diamond. Sequence contains 150 images.

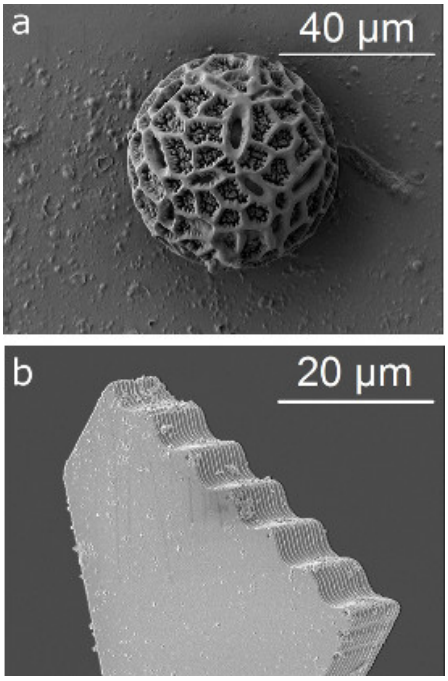



Figure 2.10: One of the images from: a) *Potamogeton* dataset, pollen grain (magnification: ×1000, image size: 2048 × 1536); b) *End effector* dataset, end effector of a microgripper (magnification: ×2000, image size: 1024 × 768).

| | Euler angles in degrees | | | Estimated angles in degrees | | |
|---|---|---|---|---|---|---|
| | $R_z$ | $R_y$ | $R_x$ | $\hat{R}_z$ | $\hat{R}_y$ | $\hat{R}_x$ |
| *Potamogeton:* | | | | | | |
| $I_1$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $I_2$ | 0 | 3.00 | 0 | -0.03 | 3.02 | 0.07 |
| $I_3$ | 0 | 6.00 | 0 | 0.07 | 6.03 | 0.10 |
| $I_4$ | 0 | 9.00 | 0 | 0.19 | 9.02 | 0.17 |
| $I_5$ | 0 | 12.00 | 0 | 0.19 | 12.03 | 0.22 |
| $I_6$ | 0 | 15.00 | 0 | 0.38 | 15.06 | 0.29 |
| $I_7$ | 0 | 18.00 | 0 | 0.33 | 18.04 | 0.32 |
| *End effector:* | | | | | | |
| $I_1$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $I_2$ | 0 | 0 | 3.00 | 0 | 0 | 2.96 |
| $I_3$ | 0 | 3.00 | 3.00 | 0 | 2.98 | 2.96 |
| $I_4$ | 3.00 | 3.00 | 3.00 | 2.98 | 2.98 | 2.96 |
| $I_5$ | 3.00 | 3.00 | 8.00 | 2.98 | 2.98 | 8.00 |
| $I_6$ | 3.00 | 8.00 | 8.00 | 2.98 | 7.93 | 8.00 |
| $I_7$ | 8.00 | 8.00 | 8.00 | 7.97 | 7.93 | 8.00 |

Table 2.1: Comparison between true Euler angles and estimated (in degrees) for two SEM image datasets, *Potamogeton* and *End effector*.

## 2.6 Affine fundamental matrix

From previous sections, the method allowing estimation of full 3D rotation is entirely based on the elements of fundamental matrix which is why we decided to devote a separate section to its estimation and analysis of its quality.

Recall that, in the case of parallel projection, the fundamental matrix is called affine and has the following form:

$$\mathbf{F} = \begin{pmatrix} 0 & 0 & a \\ 0 & 0 & b \\ c & d & e \end{pmatrix}$$

where $e$ is often taken as one, $a$,$b$,$c$,$d$ are real numbers. The fundamental matrix is estimated from a set of point correspondences (measurement matrix). One of the methods dedicated for the estimation of affine fundamental matrix is the *Gold Standard* method [HZ03] given below.

**Result 2.3: Gold Standard algorithm**

Assume one correspondence is represented by the vector $\mathbf{c}_i$:

$$\mathbf{c}_i = \left(q'_x, q'_y, q_x, q_y\right)^\top$$

Then, in order to work with relative coordinates all points are centered in $(0,0)^\top$:

$$\tilde{\mathbf{c}}_i = \mathbf{c}_i - \bar{\mathbf{c}}$$

where $\bar{\mathbf{c}}$ is the centroid of points computed as:

$$\bar{\mathbf{c}} = \frac{1}{N}\sum_i^N \mathbf{c}_i$$

with $N$ the total number of correspondences found. It allows the construction of $N \times 4$ matrix $\mathbf{A}$ with rows $\tilde{\mathbf{c}}_i^\top$. Then, if the singular vector corresponding to the smallest singular value of $\mathbf{A}$ is denoted as $\mathbf{N}$, all five elements of $\mathbf{F}$ can be found using:

$$(a,b,c,d) = \mathbf{N}^\top$$
$$e = -\mathbf{N}^\top \bar{\mathbf{c}}$$

The fundamental matrix is then obtained using Equation (2.20).

The Gold Standard allows to find such fundamental matrix that minimizes the residual error which represents the mean distance from all points to the corresponding epipolar lines:

$$\varepsilon_i = \frac{1}{N}\sum_i^N \left[d(\mathbf{q}'_i, \mathbf{F}\mathbf{q}_i)^2 + d(\mathbf{q}_i, \mathbf{F}^\top \mathbf{q}'_i)^2\right] \tag{2.31}$$

where $\mathbf{q}_i$ is the $i$-th feature extracted from the first image, $\mathbf{q}'_i$ from the second image, $d(\cdot,\cdot)$ is the geometrical distance.

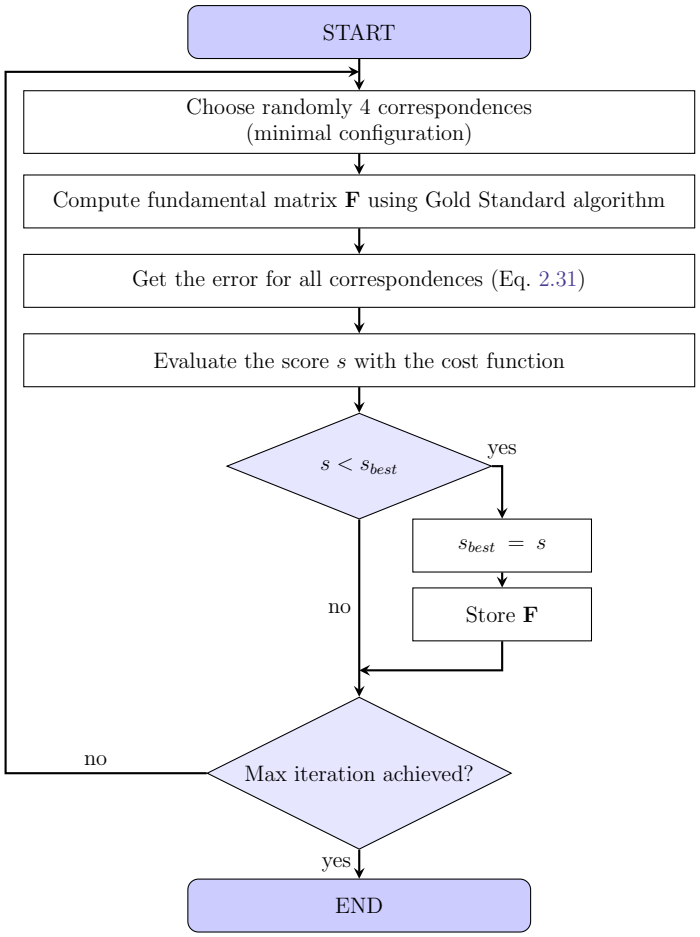



Figure 2.11: Algorithm of robust estimation of fundamental matrix for a SEM image pair.

The main drawback of this method is the lack of robustness towards the presence of outliers or mismatches. Thus, we turned our attention to a family of solutions regrouped by the name of robust estimators such as Least Median of Squared (LMedS) or RANSAC. Both methods are iterative and based on a random selection of a small subset of samples which is used for model estimation. Then, using their proper loss function, they obtain a score for the given model, i.e. how well the model fits to the data. After a number of iterations, the model with the best score is retained as the solution of the problem. The algorithm is presented in Figure 2.11. It is the same for all robust estimators reviewed in this work: the difference is in their cost functions. Four estimators were chosen for tests: LMedS [Rou84], RANSAC [Hub05], MSAC [TZ00], and MLESAC [TZ00]. At first, we describe the steps that are common for all methods and then explain the difference between them.

*How many samples represent the minimal configuration?* The first step of the algorithm consists in a random choice of a number of samples that represent the minimal configuration, or, in other words, the minimal number of samples needed for model estimation. In case of affine fundamental matrix this number is four. Only four correspondences are need to obtain an unique fundamental matrix for a pair of affine views. It can be verified by developing an epipolar constraint (by substituting 2.20 in 2.18):

for a correspondence defined by the vector $(q'_x, q'_y, q_x, q_y)^\top$, the epipolar constraint is written as:

$$aq'_x + bq'_y + cq_x + dq_y + e = 0 \tag{2.32}$$

which is a function of four parameters as $e$ is the common scale factor. Thus four of such equations are sufficient.

*How is the model estimated from samples?* In order to estimate the model, we use the Gold Standard algorithm presented above (Algorithm 2.6).

*How the error is estimated?* For all algorithms, the error is estimated as the distance from the 2D point to the corresponding epipolar line given by the current model (2.31). By developing it, we obtain that for the $i$-th correspondence, the error is given as:

$$\varepsilon_i = \left(\frac{1}{a^2+b^2} + \frac{1}{c^2+d^2}\right)(aq'_{xi} + bq'_{yi} + cq_{xi} + dq_{yi} + e)^2 \tag{2.33}$$

All errors can then be assembled in a vector $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, ..., \varepsilon_N)^\top$.

*How many iterations are necessary?* For the successful estimation, the number of iterations should be sufficient to pick a subsample with all the inliers at least once. This aspect is well studied in the litterature [CKY09]. As a the result the number of iterations, sufficient with probability $p$, is taken greater than:

$$N_{iter} \geq \frac{\log(1-p)}{\log(1-\gamma^m)} \tag{2.34}$$

where $m = 4$ in our case, as it represents the minimal configuration, $\gamma$ is the probability to pick an inlier, or, ratio of inliers over the total number of points.

### Algorithms

Consider first the LMedS algorithm. The cost function is defined as follows:

$$\mathcal{C}_{\text{LMedS}} = \text{median}(\boldsymbol{\varepsilon}^2) \tag{2.35}$$

which shows that the goal of the algorithm is to minimize the median of errors. In order to give a reliable estimate, the sample set must contain at least 50% of inliers, i.e. correct points. Remark, that the cost is estimated from the vector of errors which is no longer true for the remaining algorithms. For them, the cost is estimated for every data point separately, while the global cost is obtained as the sum of this costs:

$$\mathcal{C} = \sum_{i=1}^{N_{pts}} \mathcal{C}(\varepsilon_i) \tag{2.36}$$

The first we want to mention and the most used one is RANSAC (Random Sample Consensus) that seeks to maximize the inliers ratio. The cost function is given as:

$$\mathcal{C}_{\text{RANSAC}}(e_i) = \begin{cases} 0 & \varepsilon_i^2 < t^2 \\ \text{const} & \varepsilon_i^2 \geq t^2 \end{cases} \tag{2.37}$$

where $t$ is a threshold allowing to judge whether a given point is an inlier. We can see, that this algorithm does not take into account the quality of inliers as they all have the same cost.

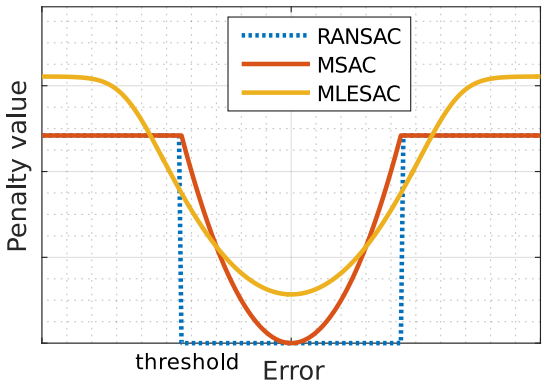



Figure 2.12: Cost functions for different RANSAC-based robust estimators.

Both these algorithms, LMedS and RANSAC, appeared in 1980's and received much attention ever since in many different fields including computer vision. RANSAC is a parent of more than ten methods that were carefully analyzed in [CKY09]. Authors divide the algorithms into three main groups having different objectives: accurate, fast and robust. In present work, we aim to obtain the 3D reconstruction of the highest possible quality, thus, two algorithms were chosen. MSAC, which stands for M-estimator Sample Consensus, is the algorithm in which every inlier have a penalty score given by how well the point corresponds to a model:

$$\mathcal{C}_{\text{MSAC}}(\varepsilon_i) = \begin{cases} \varepsilon_i^2 & \varepsilon_i^2 < t^2 \\ t^2 & \varepsilon_i^2 \geq t^2 \end{cases} \tag{2.38}$$

The last method, MLESAC (Maximum Likelihood Sample Consensus), uses the probability distribution of error by inlier or outlier to evaluate the estimated model which allows to find a maximum likelihood solution. The cost function is of the form:

$$\mathcal{C}_{\text{MLESAC}}(\varepsilon_i) = -\ln\left(\gamma \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\varepsilon_i^2}{2\sigma^2}\right) + (1-\gamma)\frac{1}{\nu}\right) \tag{2.39}$$

where, $\gamma$ is the inliers ratio, $\sigma$ is the noise variance and $\nu$ - size of error space. Thus, three parameters need to be defined. $\gamma$ is often chosen to have the value of 0.5 that is than recalculated at each iteration using expectation minimization algorithm. Figure 2.12 shows the comparison between cost functions of RANSAC family algorithms.

### Experiments

The application of these algorithms (LMedS, RANSAC, MSAC, MLESAC) for the estimation of fundamental matrix has already been proved by several works [LF96; TZ00; ZDF+95]. The presented analysis of accuracy has a different purpose: find an algorithm that gives the most accurate estimation of slope angles $\theta$ and $\theta'$. Thus, it is this criteria that will allow to judge what algorithm is the most suitable for motion estimation and 3D reconstruction.

As we do not have any information about epipolar lines in real images, we generated a pair of synthetic images. The object in them has a form of a diamond containing 22

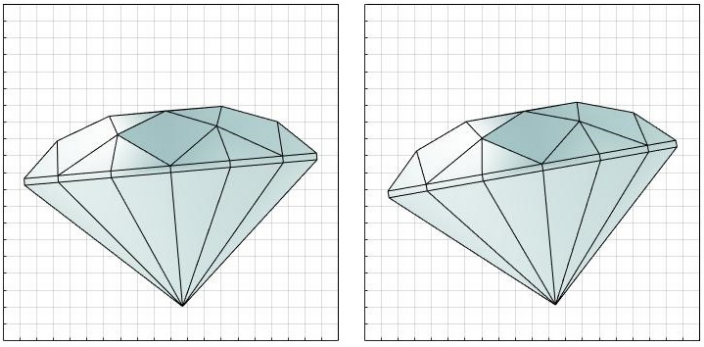


Figure 2.13: A pair of synthetic images of a diamond.

vertices[2]. The goal is to estimate the robustness and accuracy of slope angles estimation towards noise. According to [HZ03], a valuable assumption may be to represent the noise by Gaussian distribution with zero mean and variance $\sigma^2_{noise}$. This noise was generated using Box-Muller algorithm [BM58] and added to point coordinates.

Four different tests were conducted and an image pairs for each were generated in different conditions presented in Table 2.2. All of them have the same procedure. First, image pairs are generated, and the correspondences are obtained directly by projecting the 3D points. Then, a noise with corresponding variance is added to image coordinates and the fundamental matrix and slope angles are estimated. The procedure is repeated 1000 times for every test. Final results presented in Figure 2.14 are in the form of standard deviation of angles ($\theta_1$ and $\theta_2$, respectively).

Table 2.2: Conditions for four different tests of accuracy of slope angles estimation.

| | $\theta$, deg | $\theta'$, deg | $\rho$, deg | % of outliers |
|---|---|---|---|---|
| 1 | 5 | 10 | 5 | 0 |
| 2 | 5 | 10 | 5 | 10 |
| 3 | 5 | 10 | 5 | 20 |
| 4 | 5 | 10 | 20 | 0 |

From the obtained curves, we can draw the following conclusions:

*Test 1* shows the performance of the algorithm in the absence of outliers. The best estimation is given by MLESAC algorithm with the standard deviation of error not exceeding 3 degrees which is not negligible. Other algorithms have similar performance, however, the error is bigger for approximately one degree.

In *tests 2* and *3*, the set contained 10% and 20% of outliers, respectively. The best accuracy is still given by MLESAC algorithm and it shows the lowest quality decrease, i.e., the highest robustness: error stays inferior to 2 degrees for noise levels up to 0.5 pixels and up to 4 degrees for noise variance equal to one pixel.

A very interesting result is given by the *test 4* in which there is no outliers, however, the angle of out-of-plane rotation was increased by a factor of four: from 5 to 20 degrees. The test shows that it has a drastic impact on the quality of angles estimation: the error stays inferior to one degree for all noise levels and with all algorithms (MLESAC is still the best).

[2]The 3D coordinates of vertices are given in Appendix C

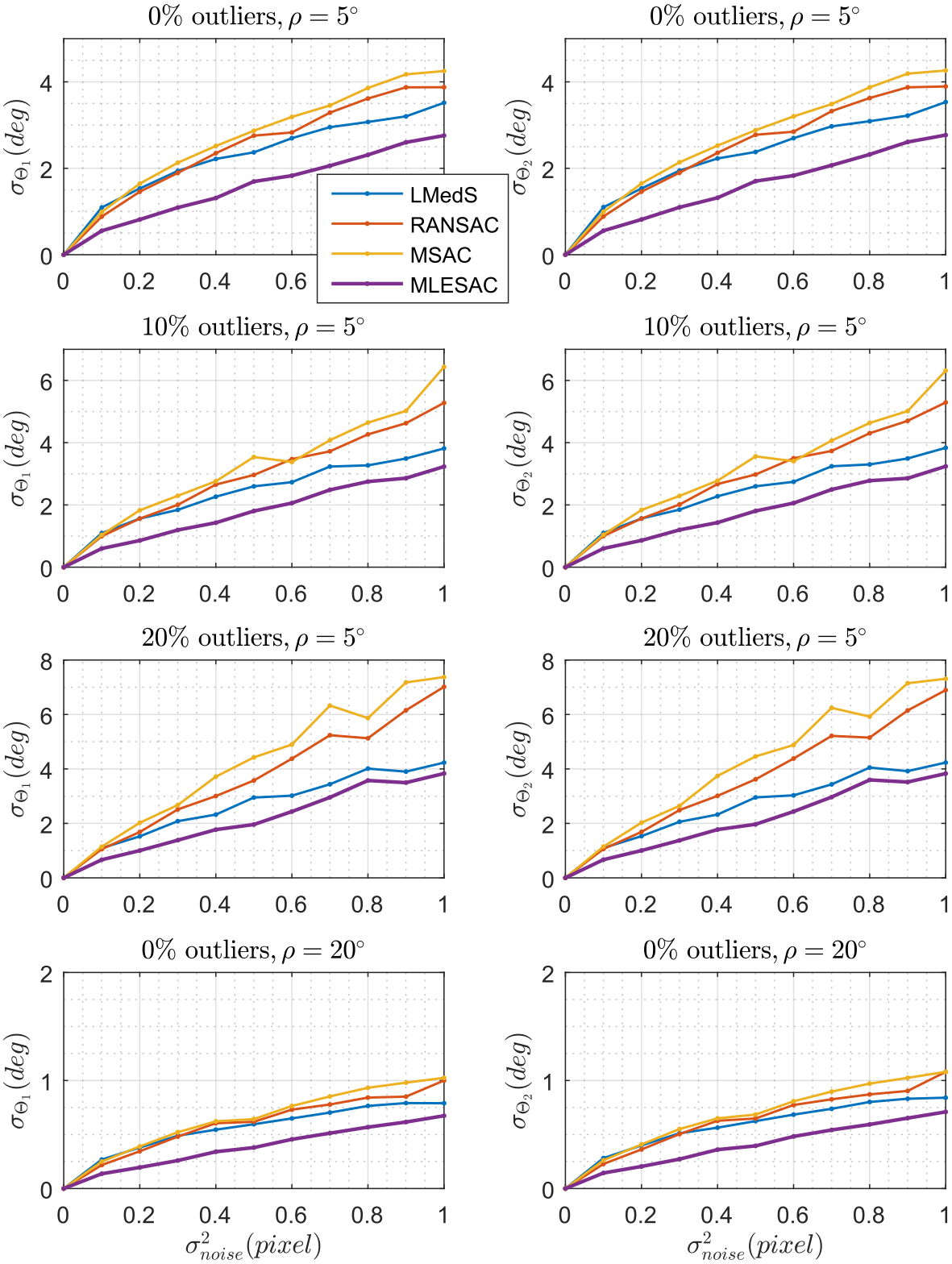



Figure 2.14: Robustness of fundamental matrix estimation algorithms. Every row corresponds to a test from Table 2.2. Left column corresponds to the standard deviation of $\theta$. Right column gives the same information about $\theta'$.

This fact plays also a crucial role for 3D reconstruction: it demonstrated that, for better quality, it is important to use images obtained with a wide baseline (with an out-of-plane rotation bigger than 15 degrees). However, for a pair of such images, the search for correspondences may be complicated as image endures an important change in the luminosity, contrast and the form of the object. Therefore, to obtain a high quality 3D reconstruction, we propose to use not two but several images obtained with small $\Delta\rho$ of about 3 degrees, but, the rotation between the first and the final image in the sequence should be as big as possible.

## 2.7 Conclusion

In this chapter, we studied the problem of motion estimation from SEM images. While for classical cameras this problem can be solved from only two images, it is impossible in SEM case due to the parallel projection that creates the bas-relief ambiguity. In contrast to other works in the field that suggest adding new sensors to the system (mostly focus), we proposed a method based on images only, by using three images instead of two.

Our solution is based entirely on the elements of the fundamental matrix that defines the epipolar geometry between views. By exploring the geometry between any pair of images taken with an affine camera, we came to the conclusion that, for all frames, camera centers are located on the surface of a sphere. The latter opened a possibility to apply all methods of spherical trigonometry which allowed to estimate full 3D camera rotation by using its decomposition on slope angles of epipolar lines and one out-of-plane rotation.

Concerning the performance of the algorithm, it should be noted that its robustness highly depends on the robustness of fundamental matrix estimation. Thus, several robust estimators were tested and compared and the best results are given by MLESAC estimator. The rotation estimation was validated on synthetic image sequence and real SEM image.

It is important to add that the method still have some limitations. Actually, for a particular motion sequences, the estimation of full camera rotation with the given method is impossible. Such motion sequence is called critical (CMS, for critical motion sequence) [Stu97]. In the present case, any motion that results in the fact that all camera centers are located on the same great circle is critical. In other words, camera centers do not form a spherical triangle. First, the estimation is impossible if camera rotates about its center. Secondly, if the motions are pure translations. Thirdly, orbital motion, i.e. all camera centers are on the same great circle.

Finally, this chapter gives a group of methods allowing to estimate a part of camera matrices, i.e. extrinsic parameters (motion). Next chapter is devoted to the estimation of intrinsic parameters using the technique of autocalibration that includes the refinement of motion parameters. Moreover, for this algorithm, the orbital camera motion, which is a very common way of image acquisition in SEM, is not critical.

# Chapter 3

# Autocalibration

## Contents



*This chapter deals with the task of autocalibration of SEM which is a technique allowing to compute camera intrinsic parameters. In contrast to classical calibration, which often implies the use of calibration object, auto- or selfcalibration is performed directly on the images acquired for a different visual task, which is 3D reconstruction in our case. First, we start with a motivation towards autocalibration and description of intrinsic parameters for parallel projection case. Then, as autocalibration represents an optimization problem, we present all the steps contributing to the success of the algorithm: formulation of the cost function incorporating metric constraints, initial estimate of parameters, definition of bounds, regularization, and optimization algorithm. Combined with the algorithms from the previous chapter, the method allows full estimation of camera matrices for all views in the sequence which is then validated on SEM images.*

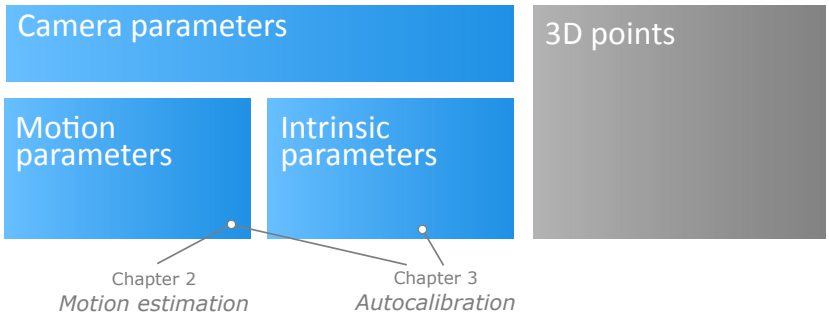



Figure 3.1: Position of the chapter in the project.

## 3.1 Introduction

In previous chapters, we have seen some crucial elements of 3D reconstruction algorithm: analysis of affine camera properties and the estimation of camera motion during image acquisition process. Recall that camera matrix includes two main types of parameters: motion parameters and intrinsic parameters. The estimation of latter ones is the core of this chapter that, in combination with the previous chapter, will allow full estimation of camera matrices (Figure 3.1). In the classic case, when a special object is used, the procedure devoted to the estimation of intrinsic parameters is called calibration. However, in the present work, we aim to estimate these parameters directly from the images taken for 3D reconstruction which can be done by using the techniques of self- or autocalibration.

Actually, in case of SEM, autocalibration has even more advantages than with standard cameras. Even if this subject is well studied, for an affine camera the calibration can be very complex due to the following reasons. As mentioned in Section 1.4, first, in most cases it requires a calibration object. It often means a special step of fabrication of such an object, which can be very expensive and time consuming especially when working with SEM. Moreover, it is very difficult to guarantee the quality of its fabrication, which has a profound impact on the precision of further image processing. Secondly, the classic calibration [Zha99] needs to be done offline, which can be very restrictive in some applications where the calibration object can't be placed in front of the camera once the operation started. Thirdly, which includes partially the second point, there is a problem of maintainability of calibration parameters. In order to re-calibrate a camera, the main operation task should be stopped. All these points contribute to turning our attention towards the techniques of auto- or selfcalibration.

At this point, it is important to mention that computation of intrinsic camera parameters is inseparable from camera motion. In fact, autocalibration is a method of calibration, allowing to recover both the intrinsic and extrinsic camera parameters, that is carried out using the same images required for performing the visual task [FLM92]. Generally, all autocalibration methods use a projective reconstruction as a starting point. It is important to notice that there are several levels of reconstruction. Projective reconstruction that is different from the true one only up to a 3D projective transformation [Har94]. Affine reconstruction differs up to an affine transformation and the euclidean reconstruction up to a similarity transformation. In the literature, Euclidean reconstruction is often referenced as similarity or metric reconstruction. Furthermore, in this work, the term metric will be referred only to a true reconstruction

with the known scale. In our case, as there is no particular information about the object size only Euclidean reconstruction is achievable, i.e., an up-to-scale reconstruction. There is a variety of methods for projective reconstruction computation and the most common ones are factorization-based methods [DLH10; ST96]. Then, once the projective reconstruction is obtained, the goal of autocalibration algorithms is to determine a rectifying homography $\mathbf{H}$ from autolibration constraints and transform the reconstruction to Euclidean one. However, these methods are often blamed for instability [Oli00].

In case of affine camera, by using a factorization techniques, it is possible to directly obtain an affine reconstruction [TK92]. It means that the plane at infinity is already in its canonical position and the goal of autocalibration is to determine the intrinsic parameters of the camera in order to upgrade the reconstruction form affine to Euclidean. A variety of methods are based on the use of some calibration constrains (zero skew, known ratio, etc.) or motion constraints (pure rotation, planar motion, etc.) [PV99; Stu97; Tri98]. One of the most significant article in the field of autocalibration of affine camera is the work of *Long Quan* [Qua96]. In his paper, the author proposes a method allowing to upgrade the affine reconstruction obtained with a factorization algorithm to the Euclidean one by using an optimization algorithm. The goal is to find a non-singular $3 \times 3$ homography matrix and the criteria of optimization is based on Euclidean motion constraints. One of the steps of estimation contains a Cholesky decomposition, which accepts only positive-definite matrices, which cannot be guaranteed in the presence of noise in real images. Thus, the author proposed an elegant way to avoid this problem by imposing the constraints on the matrix to decompose and assures by this that it is positive-definite. However, the experiments that will be presented further have shown that the method of *Long Quan* (further referenced as LQ) can not guarantee the respect of the metric constraints, such as aspect ratio close to one. A more detailed description of the intrinsic parameters will be given in Section 3.2.

As many other selfcalibration algorithms, LQ uses local optimization that aims to find a local minimum of the objective function, the minimum that is the closest to the initial solution. However, it is not sufficient for the application presented here, because the objective function contains non-linearities which results in non-convex cost function. It means that one should use more complex techniques allowing to expand the search space. More recently, this problem was adressed by the methods of global optimization. *Fusiello et al.* in [FBF+04] address this problem by an interval branch-and-bound method employed for numerical minimization based on constraint on the fundamental matrix. In [CAK+07], the global optimization is used to compute the dual image of the absolute conic to find a rectifying homography. However, even if these global optimization algorithms can guarantee a theoretical global optimality, their lock is a computational time which turns out to be a critical issue for some applications. According to *Heinrich et al.*, the fundamental limitation of these algorithms is that the minimized objective function has no particular geometric meaning which results in instability of these methods [HSF11]. In return, they propose a method based on a maximum likelihood objective function.

This chapter presents a new method of selfcalibration of SEM, an affine camera, which allows to directly obtain Euclidean set of camera matrices by means of global optimization, without passing by affine reconstruction. It is worth mentioning that we

consider that camera intrinsic parameters are constant and the scene is rigid, i.e. the change of 2D projections across the images is due only to the camera motion. The sections of this chapter cover the following crucial topics contributing to the success of the algorithm:

- formulation of optimization criteria that includes:
  - error function
  - bound constraints
  - regularization
- global optimization algorithms.

Finally, similar to previous chapter, the auto-calibration will be tested on synthetic images and real SEM images in Section 3.5.

## 3.2 Intrinsic parameters

For an affine camera, the matrix of intrinsic parameters has the following form:

$$\mathbf{K} = \begin{pmatrix} \alpha f & s' & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} = diag(f, f, 1) \begin{pmatrix} \alpha & s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = diag(f, f, 1) \begin{pmatrix} & \mathbf{A} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

with three parameters summarized in Figure 3.2:

- Aspect ratio $\alpha$ represents the ratio between height and width of one pixel. In ideal situation its value is equal to one. It is also true for SEM images which was confirmed in [CM14].
- Skew parameter $s$ reflects the orthogonality level of $\vec{x}$ and $\vec{y}$ axis in image frame. Its value should be close to zero.
- Overall scale factor $f$ represents $\frac{pixel}{m}$ ratio. This value will be considered constant and equal to one as we aim to achieve Euclidean reconstruction and not metric one. In order to make the upgrade to metric reconstruction, one can use the information about the pixel size for a given magnification which is generally given by SEM manufacturer.

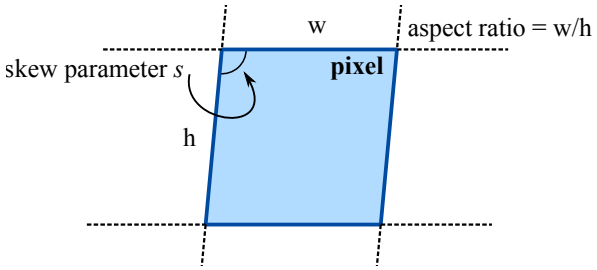



Figure 3.2: Camera intrinsic parameters.

## 3.3 Cost function formulation

The problem of autocalibration can be formulated as the following optimization problem:

$$\arg\min_{\boldsymbol{\xi}\in\mathbb{R}^n} f(\boldsymbol{\xi}) \tag{3.1}$$

where $\boldsymbol{\xi}$ is a vector of camera's parameters (both intrinsic and extrinsic) and $f(\boldsymbol{\xi})$ is the cost or objective function representing the error. Obviously, the formulation of this function determines all properties of autocalibration such as robustness and accuracy.

Consider that $N_{im}$ images of the same object were taken from different view points by the same but moving affine camera. As we have already shown, it is possible to extract a measurement matrix $\mathcal{W}$ that contains the projections of 3D points in different images (Section 2.1). Assume it contains $N_{pts}$ points. Knowing that every element of $\mathcal{W}$ can be obtained by the multiplication of 3D point coordinates and the matrix of camera in which it is projected, the expression for the estimation of measurement matrix ($\hat{\mathcal{W}}$) can be written as follows:

$$\hat{\mathcal{W}} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \vdots \\ \mathbf{W}_i \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_i \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 & \cdots & \mathbf{Q}_j \end{bmatrix} = \mathcal{P}\mathcal{Q} \tag{3.2}$$

where $\mathcal{P}$ is a stack of camera matrices ($3N_{im} \times 4$ matrix containing camera matrices for all views and $\mathcal{Q}$ is the set of 3D points in homogeneous coordinates with the size $4 \times N_{pts}$. As it was mentioned before, every camera matrix ($\mathbf{P}$) has 8 degrees of freedom (1.2.2). Furthermore, it will be considered that the images are taken with the same camera, thus, only 5 extrinsic parameters have to be estimated *separately* for every image. Thus, we need to find such $\mathcal{P}$ and $\mathcal{Q}$ that would minimize the difference between $\mathcal{W}$ and $\hat{\mathcal{W}}$. In this formulation the total number of parameters is equal to:

$$\text{length}(\boldsymbol{\xi}) = \underbrace{2}_{\text{intrinsic}} + \underbrace{3N_{im}}_{\text{rotation}} + \underbrace{2N_{im}}_{\text{translation}} + \underbrace{3N_{pts}}_{\text{3D points}} \tag{3.3}$$

The first item in this equation stands for two varying intrinsic parameters (aspect ratio and skew) of the camera; the second and third ones represent motion parameters that are different for each view; the last component is the 3D coordinates of object points.

Next, we undertake several steps allowing to reduce the number of parameters. Actually, as general optimization problems involve more variables to be optimized, it is harder to make them well-constrained. If they are not well-constrained, the optimization will proceed in a way that minimizes the global mathematical error, but that does not correspond to a solution of a real problem, i.e. has no physical meaning. Thus, it is very important to reduce the number of parameters as much as possible.

**Step 1**. As the position of the world frame is unknown, we are free to fix its orientation equal to the frame of the first camera, so that:

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

which allows to exclude three parameters:

$$\text{length}(\boldsymbol{\xi}) = 2 + 3(N_{im} - 1) + 2N_{im} + 3N_{pts} \tag{3.4}$$

**Step 2**. The biggest term in (3.3) corresponds to 3D points. One of the possible ways to eliminate it is to use the pseudo-inverse of the matrix $\mathcal{P}$. Indeed, the stack of 3D points $\mathcal{Q}$ may be replaced by:

$$\mathcal{Q} = \mathcal{P}^{+}\mathcal{W} \tag{3.5}$$

where $\mathcal{P}^{+}$ is the pseudo-inverse of the matrix $\mathcal{P}$. Here, we obtain it using SVD (Singular Value Decomposition) algorithm. Therefore, (3.2) transforms into:

$$\hat{\mathcal{W}} = \mathcal{P}\mathcal{P}^{+}\mathcal{W} \tag{3.6}$$

and the number of parameters is reduced to:

$$\text{length}(\boldsymbol{\xi}) = 2 + 3(N_{im} - 1) + 2N_{im} \tag{3.7}$$

**Step 3**. The number of parameters can be further reduced by using relative coordinates and eliminating translations (Section 2.3). It will allow to eliminate $\mathbf{t}_i$ and the number of parameters to estimate can be reduced to:

$$\text{length}(\boldsymbol{\xi}) = 2 + 3(N_{im} - 1) \tag{3.8}$$

and the equation (3.6) takes the following form:

$$\hat{\mathcal{W}} = \mathcal{M}\mathcal{M}^{+}\mathcal{W}_r \tag{3.9}$$

In this expression, $\mathcal{W}_r$ is the measurement matrix in non-homogenious relative coordinates and $\mathcal{M}$ is:

$$\mathcal{M} = \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \vdots \\ \mathbf{M}_i \end{bmatrix} = \begin{bmatrix} \mathbf{A}\mathbf{R}_1 \\ \mathbf{A}\mathbf{R}_2 \\ \vdots \\ \mathbf{A}\mathbf{R}_i \end{bmatrix} \tag{3.10}$$

where $\mathbf{M}$ is a $2 \times 3$ upper-left component of affine camera matrix $\mathbf{P}$, similar to (2.16).

Recall that

$$\mathbf{A} = f \begin{pmatrix} \alpha & s \\ 0 & 1 \end{pmatrix} \tag{3.11}$$

and $\mathbf{R}_i$ has first two rows of the rotation matrix defined as:

$$\mathbf{R}_i = \mathbf{R}_z({}^{i}\theta_{i-1})\mathbf{R}_y(\rho_{i,i-1})\mathbf{R}_z^{\top}({}^{i-1}\theta_i)$$

Ideally, we are looking for such set of parameters $\boldsymbol{\xi}$ that $\hat{\mathcal{W}} = \mathcal{W}_r$. However, due to the presence of noise in the estimation of image features, this equality is never satisfied exactly, which means that a way of comparison needs to be found. As in common, a geometric distance $d$ between estimated and measured points will be used in present work. *Hartley and Zisserman* in [HZ03] specify that the minimization of geometric error between measured and estimated 2D points is equivalent to finding such a $\hat{\mathcal{W}}$ as

close as possible to $\mathcal{W}_r$ in Frobenius norm. Thus, the autocalibration task converges to an optimization problem with the following cost function:

$$f(\boldsymbol{\xi}) = \|\mathcal{W}_r - \hat{\mathcal{W}}\|_F^2 \tag{3.12}$$
$$= \|\mathcal{W}_r - \mathcal{M}(\boldsymbol{\xi})\mathcal{M}^+(\boldsymbol{\xi})\mathcal{W}_r\|_F^2 \tag{3.13}$$
$$= \|\mathcal{W}_r - \mathcal{M}\mathcal{M}^+\mathcal{W}_r\|_F^2 \tag{3.14}$$

The goal is to find a global minimum of this function subject to a set of constraints that will be developed in the next sections.

It is important to add that, in Euclidean reconstruction, the matrix $\mathbf{M}_i$ is the result of the product of two matrices: upper-triangular matrix of intrinsic parameters $\mathbf{A}_i$ and a rotation matrix. Such matrices can be obtained using RQ-decomposition of $\mathbf{M}_i$ matrix. It is important to notice that the result of such decomposition is unique if $rank(M_i) = 2$. Thus, by computing $\mathbf{M}_i$ as the product of upper-triangular matrix and a rotation matrix, we ensure that the obtained reconstruction is an Euclidean one.

### 3.3.1 Initial values

In order to assure the convergence to the true solution, it is important to provide good initial estimates for parameters $\boldsymbol{\xi}$, the vector $\boldsymbol{\xi}_0$. As it was mentioned in previous section, the size of parameter vector is $2 + 3(N_{im} - 1)$. For example, in a three-view case (minimal configuration), it has the following form:

$$\boldsymbol{\xi} = (\underbrace{\alpha, s}_{\text{intrinsic}}, \underbrace{{}^2\theta_1, \rho_{12}, {}^1\theta_2}_{\text{image 2}}, \underbrace{{}^3\theta_2, \rho_{23}, {}^2\theta_3}_{\text{image 3}})^\top \tag{3.15}$$

Consider first the intrinsic parameters. The starting value of aspect ratio is taken to be one and the skew is zero. From other works on SEM calibration, we know that it corresponds well to the real values of these parameters for different magnification levels [CM14].

Secondly, rotation parameters: for all cameras, except the first one, the initial slope angles are found from the fundamental matrices using the methods presented in the previous sections, using the Algorithm 1, and then substituted in (2.17). The out-of-plane rotation angles $\rho$ may also be estimated using this algorithm, however, the tests have shown that for narrow baseline the algorithm lacks of accuracy. In autocalibration, this problem is avoided by fact of using more images than three, which allows to better constraint the structure and the movement. Another problem consists in the fact that in case of orbital motion (same rotation axis for all views) the estimation is not possible: all camera centers are on the same circle, i.e. there is no triangle. Indeed, orbital motion is a very common way of acquiring SEM images, because very often, the robotic stage inside the SEM allows to perform only one rotation (tilt). Thus, if Algorithm 1 fails in estimation of $\rho$, the initial value should be provided just as a very rough estimation of it. It should not be precise neither: global optimization is capable to compensate the error up to 45 degrees (see Section 3.5). In this work, we will typically use the value of 5 degrees as initial guess.

Table 3.1: Examples of initial values and bound constraints for autocalibration. The rotation of the first camera is fixed and equal to $\mathbf{I}_{2\times 3}$.

| | Initial value | Bounds |
|---|---|---|
| Aspect ratio $\alpha$ | 1 | $[0.9; 1.1]$ |
| Skew $s$ | 0 | $[-0.1; 0.1]$ |
| $\theta_1$ | Equation (2.22) | Initial + $[-8°; 8°]$ |
| $\rho$ | 5 degrees | $[-20°; 20°]$ |
| $\theta_2$ | Equation (2.22) | Initial + $[-8°; 8°]$ |
| Total number of varying parameters | $2 + 3(N_{im} - 1)$ | |

### 3.3.2 Bound constraints

Find a minimum of the objective function obtained previously is the subject of nonlinear optimization, which is typically a challenging undertaking without any additional information and thorough understanding of the nature of the objective function. In presented formulation of the objective function all of the parameters $\boldsymbol{\xi}$ have an actual geometric meaning which allows to largely reduce the search space of the solution by implementing the bound constraints. It is all the more pertinent for global optimization, where starting points are actually generated inside predefined bounds. It means that tighter bounds lead to faster convergence and higher probability of finding the solution.

The constraints for the elements of $\mathbf{M}_i$ matrix can be defined as follows. First, as regards the matrix of intrinsic parameters, as the common scale factor $f$ is factored out, the constrains can be easily imposed: the value of aspect ratio should be close to one, because the pixels are generally squared, and the skew factor should be close to zero, which would denote that $\vec{x}$ and $\vec{y}$ axis of camera are perpendicular. Furthermore, as the metric reconstruction can not be obtained (the only possible is the Euclidean "up-to-scale" reconstruction), in case of constant focal length, the value of $f$ can be fixed to any positive real value, e.g., to one. Secondly, the rotation matrix is decomposed into a sequence of three elemental rotations in a spherical coordinate system with angles $\theta_1$, $\rho$ and $\theta_2$. Hereafter, we speak about intrinsic rotations which means that they don't occur about the axes of the fixed coordinate system, but about the axes of the rotating coordinate system, which changes its orientation after each elemental rotation. It results in a following constraints for the angles. Physically, $\theta_1$ and $\theta_2$ can vary in a range of $(-\pi, \pi)$, however, the analysis of fundamental matrix in Section 2.6 showed that even at high noise level the uncertainty on the estimated angles does not exceed eight degrees. Therefore, the bounds can be defined as a range of $(-8°; 8°)$ around the initial estimate. The angle $\rho$, the out-of-plane rotation may vary in range that depends on the images. In typical situation, this angle is equal to several degrees. Here it will be fixed in the range $(-20°; 20°)$. Thus, the constraint for all elements of $\mathbf{M}_i$ matrix are defined (Table 3.1).

### 3.3.3 Regularization

Many optimization algorithms encounter the following problem: they do not take into account the physical meaning of the optimization parameters. Thus, due to the presence of noise, they find a solution that actually makes the global error smaller, but the values of parameters do not correspond to a realistic model. Moreover, the autocalibration problem may be considered ill-posed [Had02] as it is strongly non-linear due to presence of multiplication of sine and cosine functions. In such cases, the method allowing to compensate this issue is called regularization and more specifically Tikhonov regularization [TAJ77]. It allows to ensure that the minimization converges to the desired solution.

In our case, for the problem defined in (3.14), all parameters have the same weight. However, we would like to *tell* the algorithm that the values of intrinsic parameters should be close to the initial ones and that the biggest impact should be made on the values of out-of-plane rotations. In other words, during the optimization, the error should be minimized mainly by adapting motion parameters and not the intrinsic ones.

Mathematically, it translates in adding the regularization term of the following form:

$$r = \|\mathbf{\Gamma}\boldsymbol{\xi}\|_F^2 \tag{3.16}$$

where $\mathbf{\Gamma}$ is a regularization diagonal square matrix that has the same number of rows as $\boldsymbol{\xi}$. Each value on the diagonal represents the desirable impact of corresponding parameter, i.e. the weight. The bigger the value, the less the algorithm would want to change the given parameter.

In this work, we defined all weights as the power of 10. So, that:

- 100, for intrinsic parameters
- 0.1, for slope angles $\theta_1$ and $\theta_2$
- 0.01, for out-of-plane rotation $\rho$

These values were found experimentally.

Finally, the cost function with regularization term is written as:

$$f_r(\boldsymbol{\xi}) = \|\mathcal{W}_r - \mathcal{M}(\boldsymbol{\xi})\mathcal{M}^+(\boldsymbol{\xi})\mathcal{W}_r\|_F^2 + \|\mathbf{\Gamma}\boldsymbol{\xi}\|_F^2 \tag{3.17}$$

## 3.4 Global optimization

At this point, two cost functions are defined for autocalibration, $f(\boldsymbol{\xi})$ and $f_r(\boldsymbol{\xi})$, the latter one containing the regularization term. Generally, local optimization is often used in computer vision and autocalibration in particular, which works well for convex problems. However, for strongly non-linear cost functions, in order to find the minimum, it is preferable to use global optimization algorithms.

According to classification given by *Weise* in [Wei09], the global optimization algorithms can be subdivided in two main classes: deterministic and probabilistic. In case of deterministic algorithms, the search space can be, e.g., subdivided into multiple pieces, similarly to the divide and conquer strategy. This step is then followed by the exploration of each smaller region by a local solver. After that, the results are combined

and the best one is taken as a global minimum. Deterministic methods can provide a certain level of assurance that the global optimum will be located. However, even if they can guarantee that the solution found is the global one, no algorithm can do it in a finite time [MMB03]. It comes from the fact that the smaller the regions on which the space search is subdivided are, the more times the local solver should be launched. At the same time, it becomes evident that if the size of the regions are close to zero, the likelihood of finding a global minimum increases. In return, the probabilistic algorithms generate the solution based on random variables. They include algorithms like simulated annealing and evolutionary algorithms. In this this case, the solution can be found with relatively high time efficiency, however, the global optimality can not be guaranteed. In present work, we decided to apply two types of algorithms: the Scatter Search with a local solver, both presented by *Ugray et al.* in [ULP+07], and the Genetic Algorithm [Gol89]. For both algorithms, their MATLAB implementations were used.

Scatter Search (futher referenced as GS for Global Search) is a population based meta-heuristic algorithm devised to intelligently perform a search on the problem domain algorithm. It consists in a generation of multiple trial points within finite bounds, which are candidate starting points for a local solver. These are then filtered to provide a smaller subset from which the solver attempts to find a local optimum. Then it evaluates remaining possible candidates and start a local solver. The best solution is retained as the global minimum of the problem. More detailed description of the scatter search algorithm is given in [Glo98].

Genetic Algorithm (GA) is an heuristic algorithm of search, which is used to solve different optimization problems using random search, mutation and variation of parameters. It belongs to the class of evolutionary algorithms, which generate solutions by the techniques inspired by natural evolution. GA differs from other evolutionary algorithms by putting an accent on the use of cross-over operator, which conducts the operation of recombination of solution candidates.

To summarize, the proposed autocalibration scheme is represented on Figure 3.3.

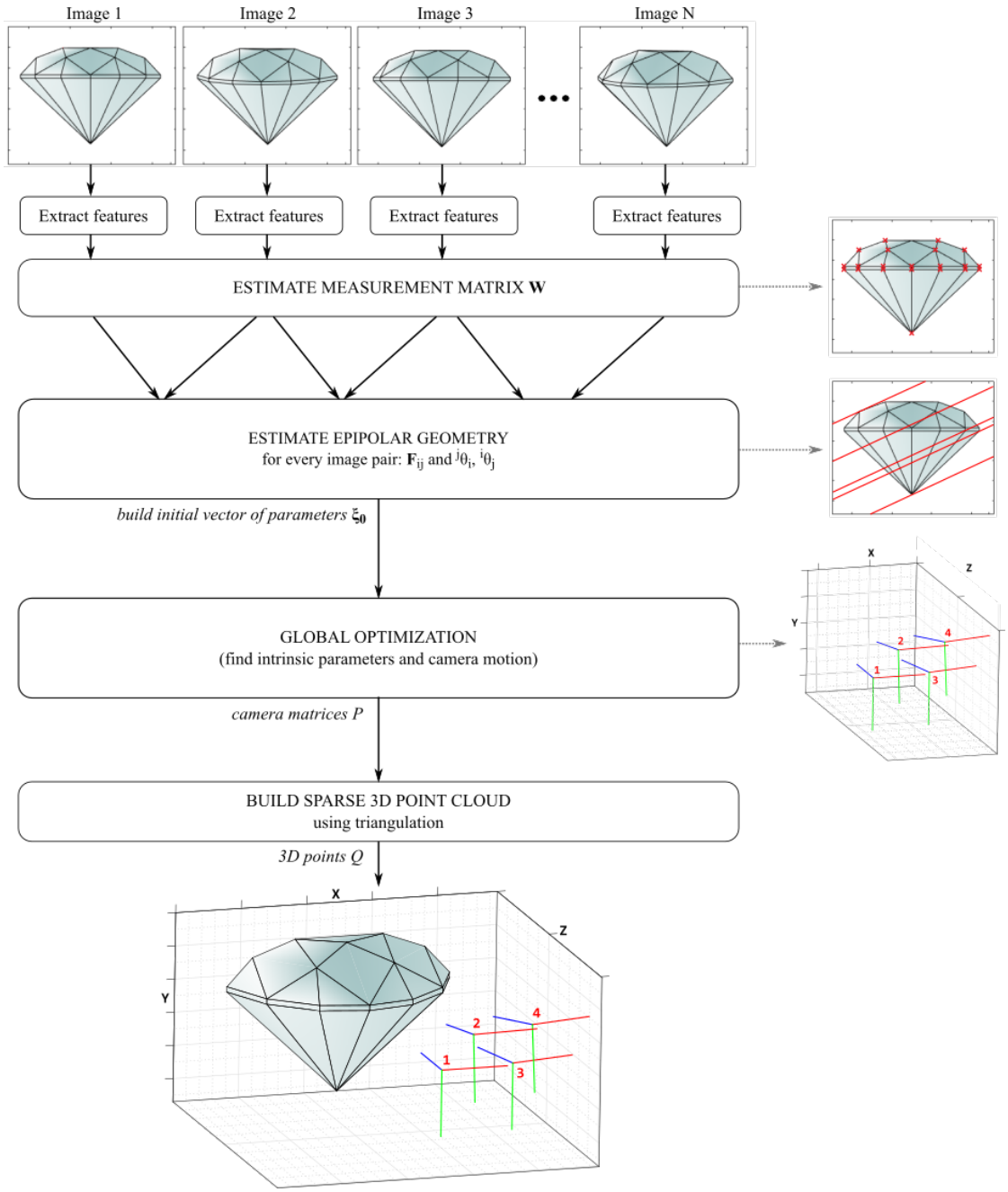



Figure 3.3: Outline of autocalibration algorithm.

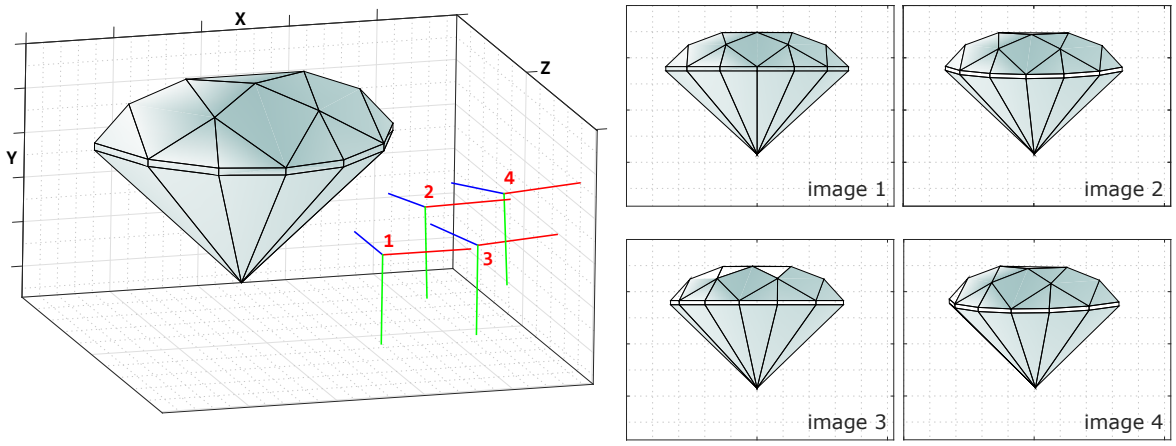
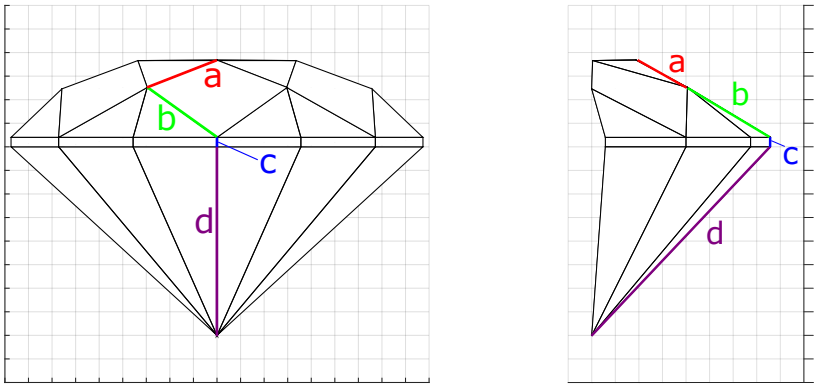



Figure 3.4: Four images of diamond sequence with 3D view. Cameras are represented by local coordinate axes.




(a) (b)

Figure 3.5: Front ($a$) and side view ($b$) of the virtual object. The marked sides are used for the analysis of performance of the autocalibration algorithms.

## 3.5 Experiments

Next step in solving the problem autocalibration is the proper choice of the cost function and of the global optimization algorithm. Four combinations are analyzed:

- GSR, scatter search optimization with regularization
- GS, scatter search optimization without regularization
- GAR, genetic algorithm optimization with regularization
- GA, genetic algorithm optimization without regularization

Their performance will be compared to the state-of-the-art algorithm of *Long Quan* [Qua96]. At first, we use a sequence of four synthetic images of a diamond (Figure 3.4). As both, the 3D structure and a set of cameras with all parameters, are known, the measurement matrix contains perfect noise-free correspondences. The faces between

vertices are given only in a purpose of better visualization of the object. As it was mentioned previously, it is not possible to obtain the metric reconstruction without any additional information on the object structure or camera focal length. Thus, in order to analyze the results, only the properties that are preserved under similarity transformation might be used. Among them are ratio of lengths and angles. For the analysis 4 lines of the object were chosen (see Figure 3.5). The real values of the ratios of lengths and the angles as well as the estimated ones are presented in Table 3.2. It can be noticed that all methods give the same results, the estimation error for both ratios and angles is lower than thousandth of percent. These results confirm the viability of presented approach and allows to proceed to the tests on real images.

It is also interesting to look at the execution time (Table 3.3). It was measured for MATLAB implementation of global optimization on the computer with the following parameters: Windows7 x64, 3.20 GHz Intel Core i5 CPU, 6 MB cache and 8 GB of RAM. One can remark, that for the same output results, the genetic algorithm takes more 20 times more time than scatter search method. Thus, genetic algorithm will no longer be used for tests.

Table 3.2: Comparison of performance of different algorithms on a noise free sequence of synthetic images.

| | Ratios of lengths | | | Angles | | |
|---|---|---|---|---|---|---|
| | a/b | b/c | c/d | $\alpha_{ab}$ | $\alpha_{bc}$ | $\alpha_{cd}$ |
| Real value | 0.761 | 12.660 | 0.036 | 94.027 | 114.747 | 136.544 |
| LQ | 0.761 | 12.660 | 0.036 | 94.027 | 114.747 | 136.544 |
| GSR | 0.761 | 12.660 | 0.036 | 94.027 | 114.747 | 136.544 |
| GS | 0.761 | 12.660 | 0.036 | 94.027 | 114.747 | 136.544 |
| GAR | 0.761 | 12.660 | 0.036 | 94.027 | 114.747 | 136.544 |
| GA | 0.761 | 12.660 | 0.036 | 94.027 | 114.747 | 136.544 |
| Errors: | | | | | | |
| $\varepsilon_{LQ}, \%$ | 6.56e-13 | 4.21e-14 | 1.34e-13 | 4.53e-13 | 5.70e-13 | 6.24e-14 |
| $\varepsilon_{GSR}, \%$ | 3.35e-05 | 1.51e-04 | 2.21e-04 | 1.87e-04 | 4.69e-05 | 8.11e-05 |
| $\varepsilon_{GS}, \%$ | 2.32e-05 | 1.23e-04 | 1.74e-04 | 1.46e-04 | 3.41e-05 | 6.45e-05 |
| $\varepsilon_{GAR}, \%$ | 3.35e-05 | 1.51e-04 | 2.21e-04 | 1.87e-04 | 4.69e-05 | 8.11e-05 |
| $\varepsilon_{GA}, \%$ | 2.32e-05 | 1.23e-04 | 1.74e-04 | 1.46e-04 | 3.41e-05 | 6.45e-05 |

Table 3.3: Mean execution time of different autocalibration methods for 4 images and 22 points.

| Method | LQ | GSR | GS | GAR | GA |
|---|---|---|---|---|---|
| Time, s | 0.0120 | 4.4152 | 4.8811 | 144.9412 | 100.3336 |

### 3.5.1 Robustness to noise

To summarize the previous sections, we have several methods (LQ,GSR,GS) that perform equally well for noise free images. However, this situation is far from reality as the set of feature coordinates always contain noisy measurements. Hence, it is important to test the performance of the algorithms at different noise levels. The procedure is exactly the same as above, except for a Gaussian noise with different variance is added to

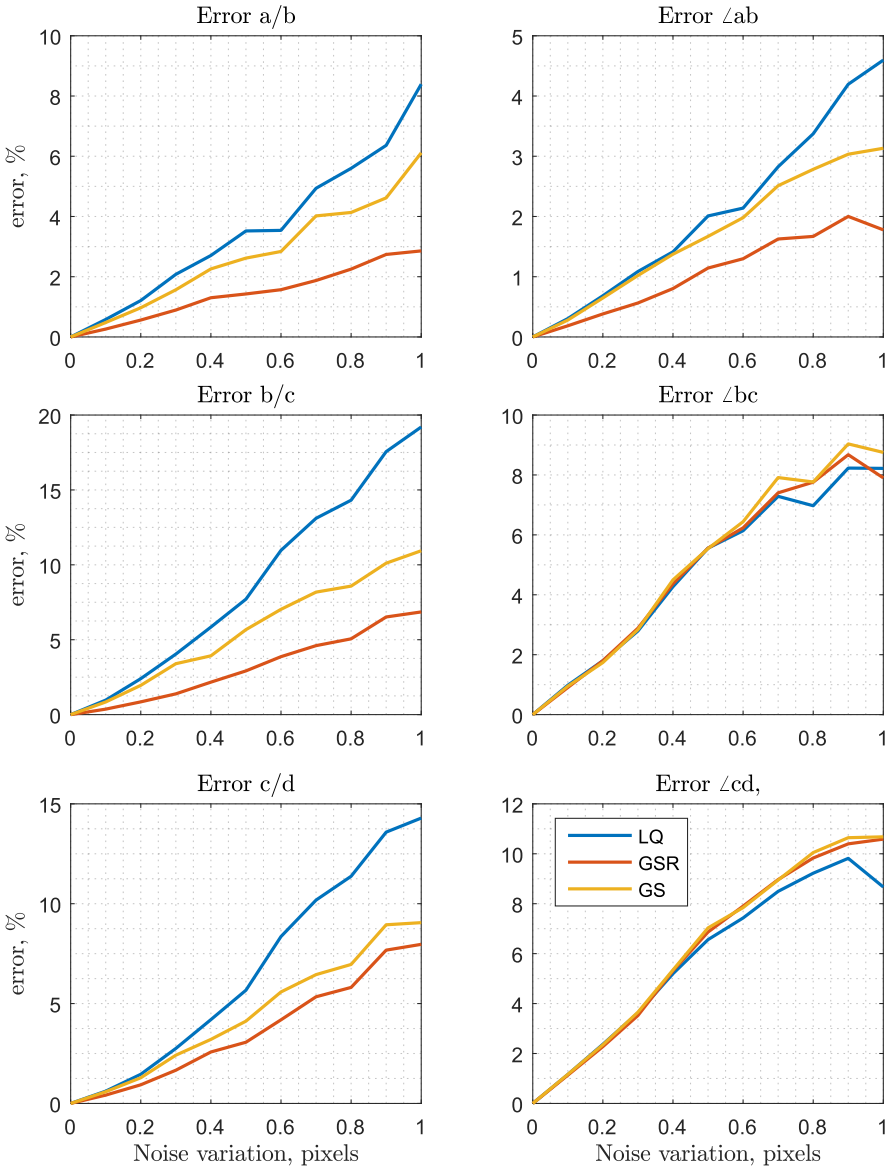



Figure 3.6: Robustness of different autocalibration algorithms to noisy measurements.

2D image coordinates, similar to tests on fundamental matrix in Section 2.6. For every noise level, the autocalibration was repeated 100 times. The results are represented in Figure 3.6.

At this point we remark that GRS outperforms both the methods LQ and GR: even for high noise level, the error for estimation of length ratios stays inferior to 6%. In contrast, it may reach 20% for LQ method and 10% for the scatter search optimization without regularization (GS). The situation with angles is similar, but the difference between methods is lower: the biggest error is of 12% for GS and GRS, and 10% for LQ. For lower noise levels, the error stays inferior to 5% for all measurements using GSR method. These results prove at the same time the efficiency of presented method and the importance of regularization term.

### 3.5.2 Convergence range

As it was already shown, global optimization is a time consuming technique. Typically, for a sequence of four images with 22 points, the autocalibration takes about 4 seconds. And an important question arises: is it worth using global optimization instead of local one? The answer is yes. To prove that, we conducted the following experiment. Once again, four images are obtained in a way that the out-of-plane rotation between the first image and the second one is 5 degrees. Then, we run autocalibration with different initial conditions using GSR method and the same method but with Levenberg-Marquardt (LM) local optimizer (LMR, for Levenberg-Marquardt with regularization). The results are summarized in Table 3.4.

Table 3.4: Comparison of convergence range between local optimizer (Levenberg-Marquardt, LMR) and global optimizer (GSR). Tests were run on noise-free images.

| Initial $\rho_{12}$, in degrees | -20 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 20 | 45 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LMR | 5.23 | 0.00 | 1.69 | 4.07 | 4.67 | 4.81 | 5.00 | 5.06 | 5.08 | 5.10 | 5.23 | 5.4 |
| GSR | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 | 5.00 |

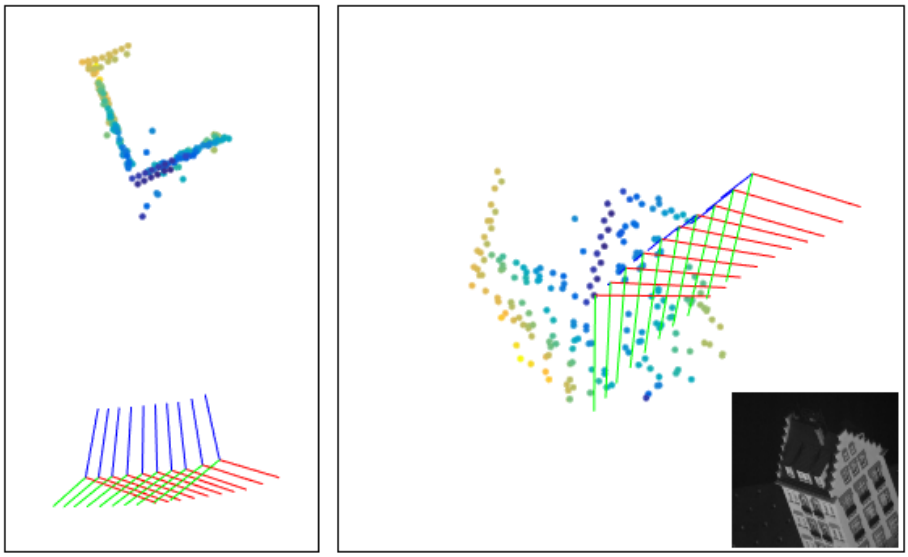
As expected, with LM algorithm, optimization suffers from multiple local minima of the cost function, i.e., one need a very good initial estimate to use this algorithm. In contrast, global optimization allows indeed the arbitrary choice of starting point as it is capable to compensate the error of 45 degrees.

### 3.5.3 Real images

In present work, six image datasets were used (detailed description in Appendix D):

- affine camera images:
  - **hotel**: 10 images from the hotel image sequence [PK97].
- SEM images from [TKH+15]:
  - **brassica**: 4 images of pollen grain of white turnip plant from a Hitachi S-4800 FE-SEM.
  - **grid**: 5 images of TEM copper grid from a Hitachi S-4800 FE-SEM.
- SEM images acquired at FEMTO-ST Institute:
  - **cutting**: 4 images of the cutting edge of a microfabrication tool. Images were taken using a SEM Zeiss AURIGA 60.
  - **pot**: 7 images of pollen grain of aquatic, mostly freshwater, plant of the family *Potamogetonaceae*. Images were taken using a SEM Zeiss AURIGA 60.
  - **pot2**: 15 images of another *Potamogetonaceae* pollen grain. Images were taken using a SEM Zeiss AURIGA 60.

The **hotel** sequence is the image dataset used by *Long Quan* to validate his algorithm. In our implementation, 196 matches were extracted and the results of autocalibration, as expected, are exactly the same for all three methods (LQ,GSR,GR). Positions of camera as well as the sparse 3D reconstruction are shown in Figure 3.7.



a) Top view b) View aligned with the first camera

Figure 3.7: Result of autocalibration for **hotel** image dataset.

Back to SEM images, five datasets were analyzed. As all image datasets were taken using SEM, there is no available information on the location of the camera. Thus, we will use two criteria allowing to judge on the algorithm performance. First, the respect of metric constraints and, secondly, the reprojection error (Figure 3.8). It is important to note, that we consider that metric constraints are respected if the aspect ratio is within the interval $(0.9; 1.1)$ although we know that it has to be close to one. For LQ method, the constraints are not respected in 4/5 cases, and 3/5 cases for GS. GSR algorithm gave the desired result for all datasets as the constraints on aspect ratio and skew are already included in cost function formulation. In terms of reprojection error, the algorithms give very close results: for all SEM images, a subpixel estimation is achieved for at least 85% of points. This result is important because it proves that even if we prevent the change of intrinsic parameters, the algorithm (GSR) succeeds in keeping the same level of accuracy as LQ and GS. Obviously, these two criteria do not allow full validation of the autocalibration: one can consider them as mandatory but not sufficient.

Unfortunately, when working at such scales and with small objects, there is no measurement device capable to provide the ground truth. However, for some datasets, we know the actual angle of tilt: 3 degrees for **brassica**, 7 degrees for **grid**, and 3 degrees for **pot**. For other datasets, the rotation between images were performed as a combination of two articular robot movement which was not calibrated, which means that the angle of out-of-plane rotation cannot be extracted. To compare the rotation angles, the following procedure was performed. Once the sequence passed through autocalibration, the rotation matrices for every frame were extracted and then

transformed into axis-angle representation. Therefore, if the rotation were performed in one movement, these angles should match the angles given by robot sensor. The results are presented in Table 3.5. For **brassica** and **pot** datasets, the difference do not exceed 0.15 degrees. Hovewer, for images of a copper **grid**, the deviation is bigger, 0.2 degrees, except for the first image pair where deviation is *exactly* one degree.

Table 3.5: Comparison between rotation angles in degrees given by robot sensors (true), and autocalibration algorithm (estimated).

| | **brassica** | | **grid** | | **pot** | |
|---|---|---|---|---|---|---|
| | true | estimated | true | estimated | true | estimated |
| $\rho_{12}$ | 3 | 3.0797 | 7 | 6.0026 | 3 | 3.0427 |
| $\rho_{23}$ | 3 | 3.1529 | 7 | 7.0387 | 3 | 3.1072 |
| $\rho_{34}$ | 3 | 3.0293 | 7 | 6.7823 | 3 | 3.0999 |
| $\rho_{45}$ | - | - | 7 | 6.9036 | 3 | 3.1430 |
| $\rho_{56}$ | - | - | - | - | 3 | 2.9871 |
| $\rho_{67}$ | - | - | - | - | 3 | 3.0572 |

## 3.6 Conclusion

In this chapter, a new method of autocalibration for affine camera was presented finalizing the estimation of camera matrices. All its components have been computed: intrinsic parameters as well as motion parameters. The presented method, being based on global optimization, has the following advantages comparing to the state-of-the-art techniques:

- all metric constraints are imposed directly on the optimized parameters. Moreover, with regularization, it is possible to guide the optimization towards the desired result, i.e. add a penalty score for excessive change of such parameters as aspect ratio and skew;
- all optimization parameters have an actual physical meaning. It means that all known elements about camera location can be easily imposed. For example, one can impose the equality of out-of-plane rotation if the angle of tilt is known to be constant for all image sequence;
- thanks to the deep analysis of camera properties, the final number of optimized parameters is low and good initial estimate is provided from the elements of fundamental matrices. These are the key success factors of optimization. Moreover, in contrast to classical bundle adjustment techniques, the 3D points are excluded from optimization process.
- the fact of using global optimization ensures high convergence range;
- the execution time is relatively low: less than 5 seconds for 4 images. It can be improved, first, by implementing global optimization in C++. Secondly, the optimization for multiple starting points may be carried out in parallel which would also decrease the optimization time.

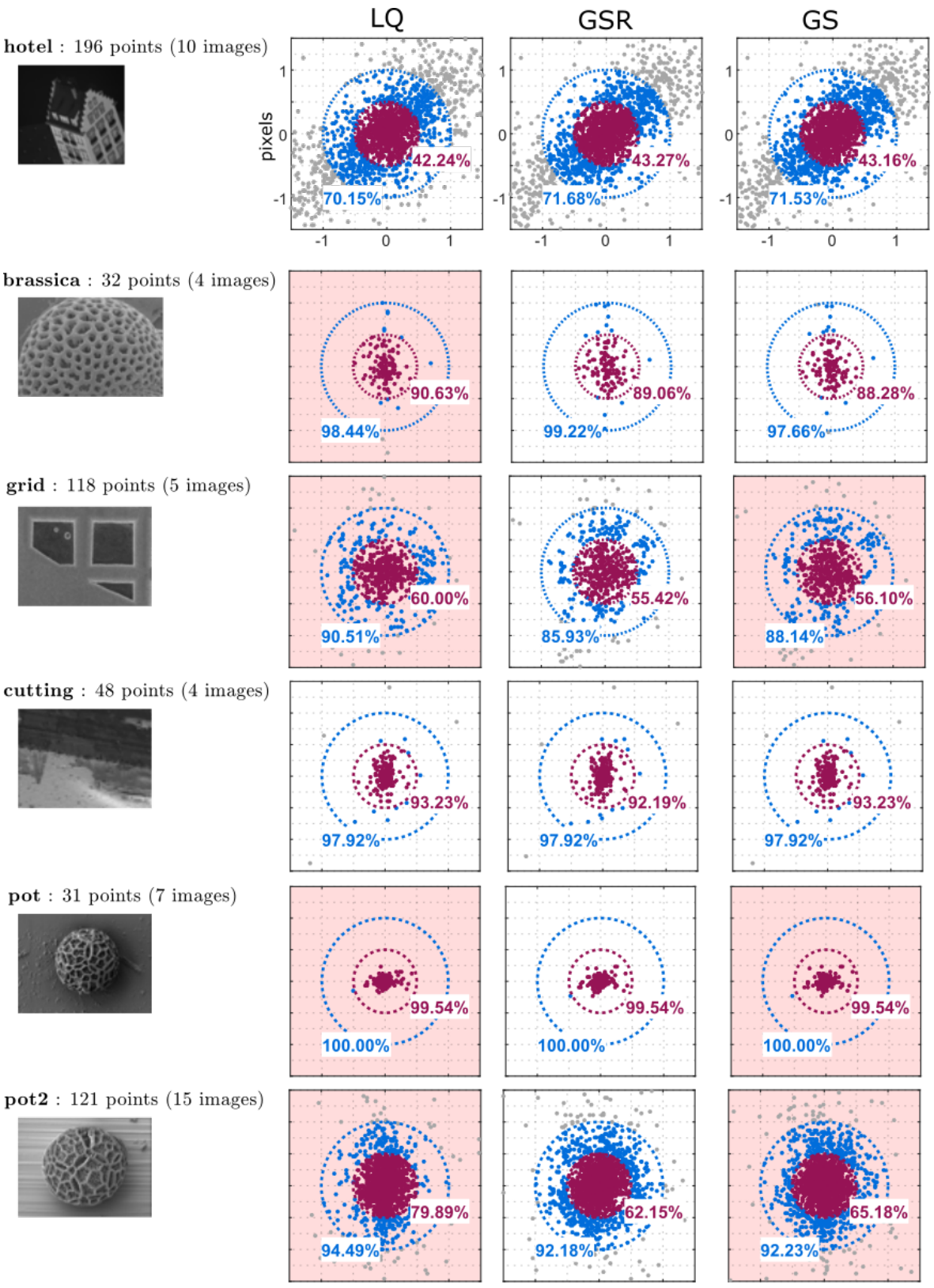



Figure 3.8: Analysis of reprojection error for different autocalibration algorithms: LQ, GSR, GS. Red background means that metric constraints are not respected: typically, aspect ratio is greater than 1.5 or lower than 0.5. For points inside circles, the reprojection error is lower than: 1 pixel for blue circle, 0.5 pixel for red circle. Numbers reflect the percentage of points inside the corresponding circle.

With regard to microscopy community, we presented the first autocalibration algorithm for Scanning Electron Microscope.

It is worth adding that the crucial step in all autocalibration algorithms is the feature matching across the images. In case of affine camera less points are needed to constraint camera parameters, because of the reduced number of degrees of freedom. As it was shown before, we used the number of features from 20 to 200. If it is possible to increase the number of matches without loss of their quality, it will improve the final results, however, will slightly increase the execution time. Though, the quality of the correspondences is much more important than their quantity: even if a feature is an inlier, its quality may vary. Outliers are rejected at the step of fundamental matrix estimation.

At last, the result of autocalibration is a sparse point cloud and camera matrices corresponding to every image in the sequence. Certainly, a point cloud that contains only 200 points is not enough not only for the purposes of characterization but even for visualization. In the next chapter, we will demonstrate a group of techniques allowing to upgrade the reconstruction form sparse to dense one, that may contain millions of points.

# Chapter 4

# Dense 3D reconstruction

## Contents



*In the previous chapter, we demonstrated a method of SEM autocalibration allowing to recover both intrinsic parameters and motion parameters, i.e., all elements of camera matrices. At this point, we were capable to obtain sparse 3D reconstruction, containing only few points, usually up to 200. This result is not sufficient neither for object characterization nor for proper visualization. With 200 points, the object is often unrecognizable in 3D reconstruction. This issue brings us to a group of methods presented in this chapter which has a goal of upgrading the reconstruction from sparse to dense one, containing hundreds of thousands points. First, we present a rectification algorithm allowing to simplify the search for correspondences between images which would allow to build a disparity map for every image pair. The disparity is closely related to depth: this relation will be found with the triangulation technique adapted for SEM images. The output of the algorithm is a dense point cloud representing the surface of the object. Finally, some views of 3D reconstructions are presented.*

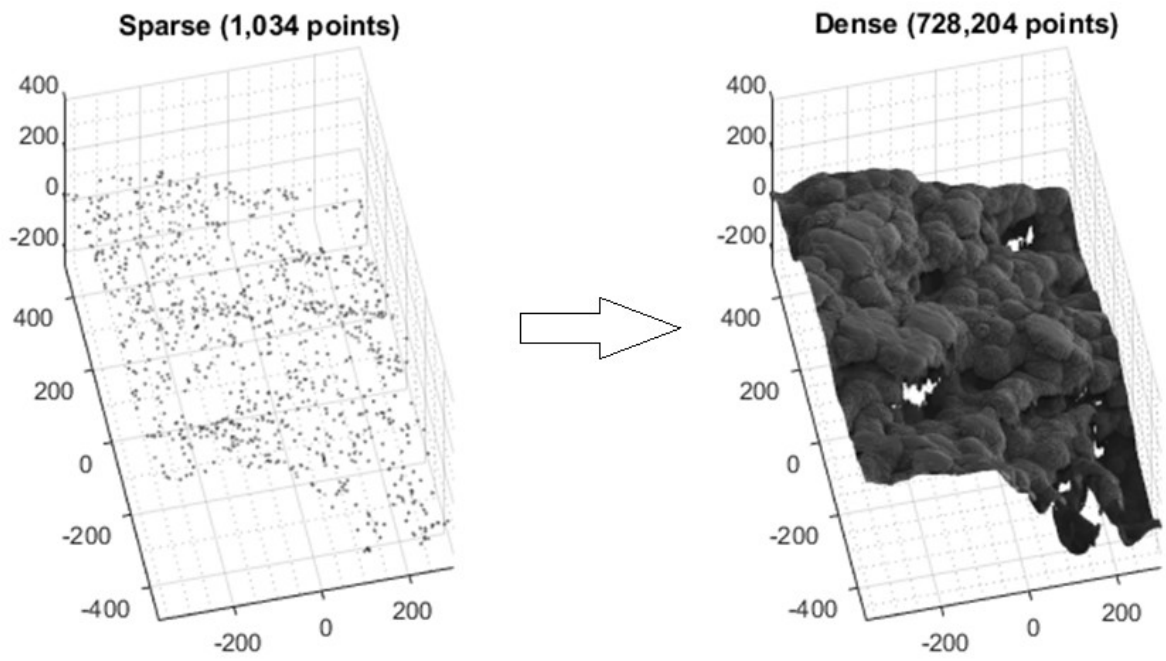



Figure 4.1: Example of upgrade of sparse 3D reconstruction to a dense one. The object is the surface of polypyrrole material viewed with SEM (**ppy** dataset, Appendix D).

## 4.1 Introduction

As a result of autocalibration, which was established in the previous chapter, we have now all the information about the camera, its intrinsic parameters and location in 3D space. Another available entity is a sparse 3D point cloud reconstructed from sparse measurement matrix. Recall that it usually contains low number of points, less then 1000. Due to the fact that such point cloud is unusable for both visualization and characterization, in this chapter we address the problem of upgrading the sparse reconstruction to dense one. Figure 4.1 reflects this goal. In the given example, the number of reconstructed points was multiplied by a factor of 700.

In order to achieve dense reconstruction, one needs to find a correspondence for every image pixel, i.e., to perform a dense matching. Classical feature matching techniques such as SIFT, SURF, etc. are not generally applied in this case as it would demand to calculate a descriptor for every pixel which is very time consuming. Moreover, matching algorithms generally perform the search over the entire image which is unnecessary if the geometry between views is taken into account. In the chapter dedicated to motion estimation, we have shown that the search for correspondence may be reduced to a line once the fundamental matrix is estimated. Actually, the correspondence for every pixel of the first image is located on the corresponding epipolar line in the second image. Moreover, rectification allows reducing the search space to a horizontal line. It means that the search is performed only along one dimension that accelerates dramatically the algorithm of dense matching. In Section 4.2 we describe the rectification algorithm for SEM images.

Once the images are rectified, we obtain a new image pair that is then supplied to dense matching algorithm that actually performs the search and finds the difference

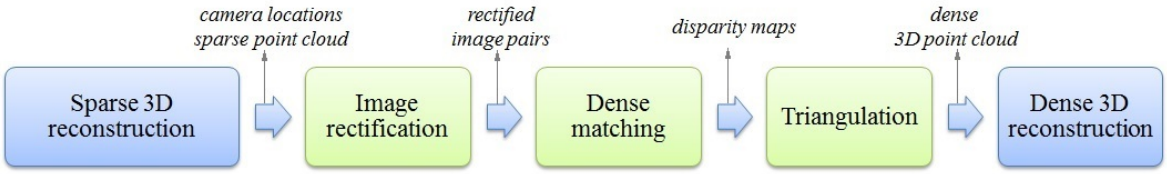



Figure 4.2: Outline of the algorithm allowing to upgrade 3D reconstruction from sparse to dense.

between the position of the pixel in the first image and in the second image. This difference is called disparity and it is closely related to the lacking depth coordinate. The methods are described in Section 4.3. The final step is the triangulation that allows to reproject 2D points in 3D space and to obtain a dense 3D point cloud (Section 4.4). The pipeline of the sparse-to-dense algorithm is summarized in Figure 4.2.

## 4.2 Rectification

Rectification consists in warping two images in the common plane (making them coplanar) to reduce the search of correspondence to one dimension, i.e., to a horizontal line. This technique is based on epipolar geometry which gives a number of geometric constraints between the 3D points and their projections onto the 2D images that can be rewritten mathematically in the form of $3 \times 3$ fundamental matrix. Usually, these constraints are based on the assumption that camera model is perspective [LZ99; PKV99] and then the goal of rectification consists in applying a pair of projective transformations to the image pair. However, the model and then rectification can be simplified regarding special imaging conditions, e.g., when the object is far away from the view point, i.e., when the focal length is much bigger than the depth variation of the object which is the case of Scanning Electron Microscope (SEM). For SEM, the perspective effects can be neglected and a parallel projection model can be used for magnification values bigger than $\times 1000$, which is confirmed in the literature [CGS+03; CM14; SRK+02]. Such model assumes that all projection rays are parallel, which means that all epipolar lines are parallel and the epipoles are at infinity.

Basically, rectification algorithms can be subdivided into two main classes depending on whether the cameras are calibrated [FTV00] or not [KMP+10]. In the case of parallel projection, the calibration is the subject of finding eight parameters, corresponding to eight degrees of freedom. The problem of affine rectification was partially addressed in [LJD09; LZ99]. Authors work with perspective cameras and separate the task of rectification on two transformations: projective and affine. They firstly find a projective transformation in order to reduce the rectification task to an affine one. The affine transformation represents scale, rotation and translation. All of these parameters are then found by using optimization approaches.

In present work, we derive a direct linear rectification method for SEM images allowing to find a rectifying transformation. The method is based on epipolar geometry, in particular, on the special form of the fundamental matrix in case of parallel projection, which was estimated using MLESAC method presented before in Section 2.6. In two sections that follows, we present the method itself and then the experimental results.

### 4.2.1 Image transformation

The goal of rectification for classical perspective cameras is often stated as follows: apply a perspective transformation to both images in order to make their optical axes parallel and their epipolar lines horizontal. However, in the case of SEM parallel projection with constant magnification, as it will be demonstrated further, the condition may be formulated in another way: to obtain a pair of rectified images, the only necessary condition is the coplanarity of $\vec{x}$ axes of both camera frames (the first is still true).

As it can be seen from Figure 2.4, for any pair of affine cameras, there is a circle passing through the centers of both cameras and the world origin. This circle defines uniquely a plane $\boldsymbol{\pi}_r$ from three points. As a result, for images to be rectified, it is enough to apply a rotation about $\vec{z}$ in order to make $\vec{x}$-axis of both cameras tangent to this circle. As the position of cameras is constrained by the epipolar geometry, the demanded rotation angle can be found using fundamental matrix, as it was already done in Section 2.4.2. In fact, all epipolar lines are parallel between themselves and to the plane $\boldsymbol{\pi}_r$, which means that the needed rotation angles can be calculated as slopes of one of epipolar lines for both images:

$$\theta = \arctan\left(-\frac{d}{c}\right), \theta' = \arctan\left(-\frac{a}{b}\right) \tag{4.1}$$

where $a, b, c, d$ are the elements of the fundamental matrix. Knowing the slope angles, one can rectify the stereo-image using the algorithm presented below.

Next step consists in applying an affine transformation of the same form to both images. The transformations of both images are:

$$\mathbf{T} = \mathbf{R}_z(\theta) \tag{4.2}$$

$$\mathbf{T}' = \mathbf{R}_z(\theta') \tag{4.3}$$

which represents the rotation by an angle $\theta(\theta')$ about the $\vec{z}$-axis, perpendicular to image frame. After these transformations, all epipolar lines are horizontal, however, they still need to be aligned vertically. In order to do that, a vertical shift $\Delta s$ should be applied to one of the image, e.g., if its sign is negative, $\Delta s$ lines should be added to the beginning of the second (right) image.

$$\Delta s = \frac{1}{N_{pts}} \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \sum_{i}^{N_{pts}} (\mathbf{T}\mathbf{q}_i - \mathbf{T}'\mathbf{q}'_i) \tag{4.4}$$

At this step, the images are rectified: all epipolar lines are horizontal after image rotation and then the vertical shift was compensated. Figure 4.3 shows the output of the algorithm for image pair in **cutting** sequence at every step of rectification.

### 4.2.2 Experiments and analysis

In this section, we present the analysis of rectification algorithm performance. The main criterion allowing to judge if the rectification was successful is the rectification error, which is calculated as the difference between $q_y$ and $q'_y$ in rectified images. In other words, the error reflects the distance between epipolar lines for one correspondence in the first and in the second image. The algorithm was applied to six image

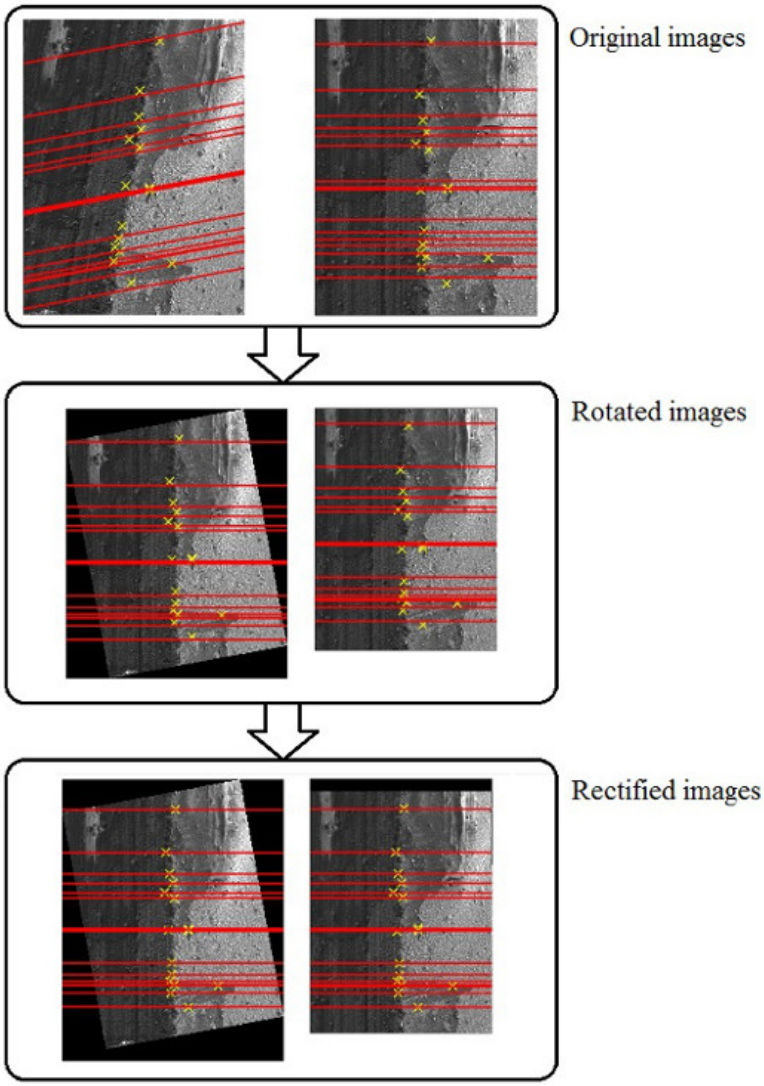



Figure 4.3: Steps of image rectification on the example of first two images of **cutting** sequence.

sequences: **brassica**, **grid**, **cutting**, **pot**, **pot2**, and **ppy**. The results are summarized in Table 4.1. The mean of the error for a set of correspondences does not exceed 0.5 pixel. The standard deviation has a bigger variation across datasets. However, for only 4 out of 34 images pairs, the standard deviation exceeds 1 pixel that may be due to the presence of outliers that have not been filtered during fundamental matrix estimation. Another point is that for datasets with higher number of features (**grid**, **pot2**, **ppy**) the error is generally bigger. This fact confirms that lower number of features of higher quality is preferable to a big number of points. The resulting rectified images as well as the original images are displayed in Figure 4.4 and we can proceed to the dense matching algorithms.

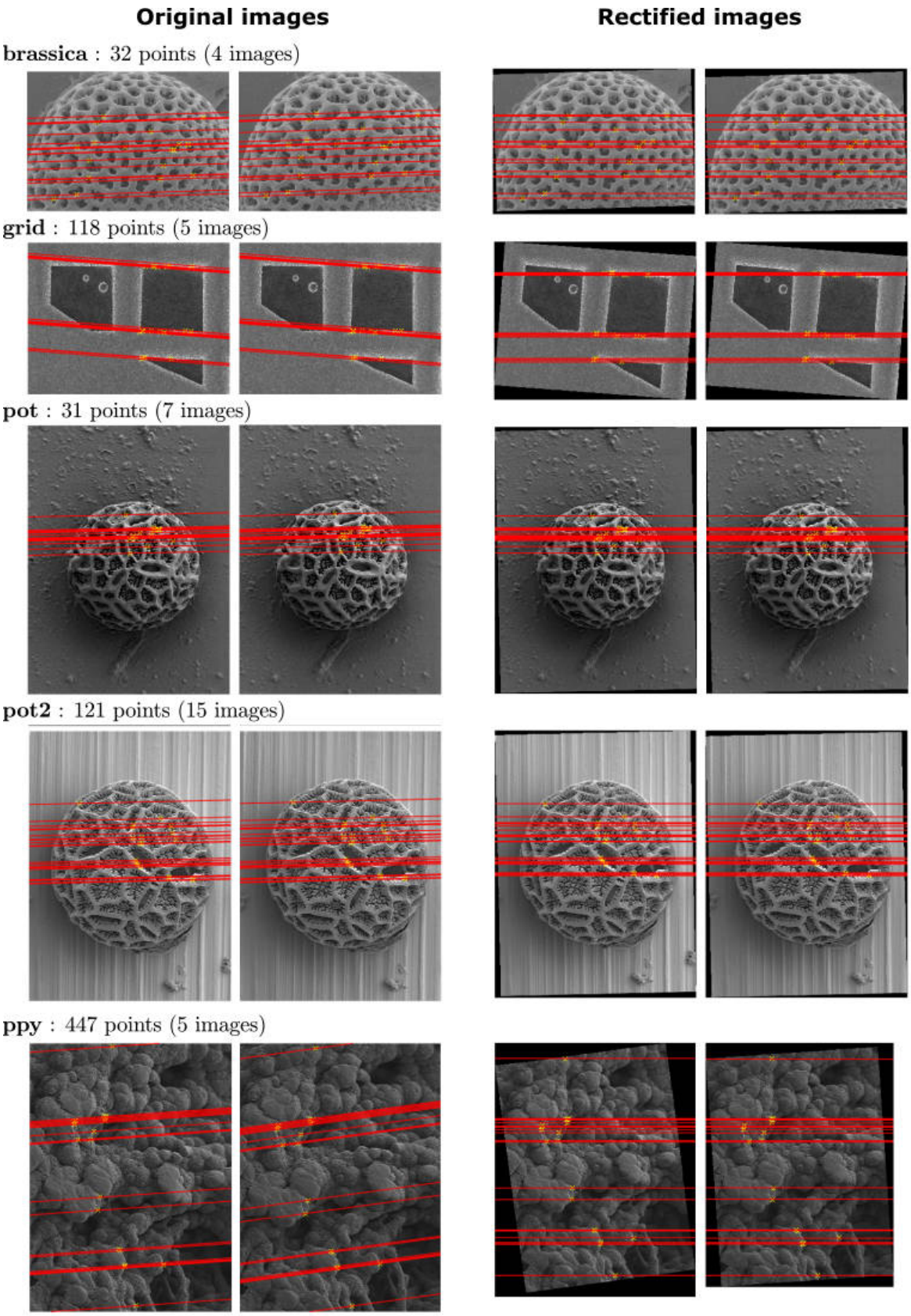



Figure 4.4: Original and rectified images pairs for different SEM image sequences. Only pairs $1 \leftrightarrow 2$ are displayed. Epipolar lines are displayed only for 20 points.

Table 4.1: Rectification errors for various SEM image datasets. Errors are given in pixels.

| Image pairs | **brassica** | | **grid** | | **cutting** | |
|---|---|---|---|---|---|---|
| | $\epsilon_{mean}$ | $\epsilon_{std}$ | $\epsilon_{mean}$ | $\epsilon_{std}$ | $\epsilon_{mean}$ | $\epsilon_{std}$ |
| $1 \leftrightarrow 2$ | 0.031 | 0.547 | -0.117 | 0.978 | -0.403 | 0.505 |
| $2 \leftrightarrow 3$ | -0.025 | 0.384 | -0.270 | 0.673 | -0.295 | 0.358 |
| $3 \leftrightarrow 4$ | -0.193 | 0.327 | 0.049 | 1.048 | 0.249 | 0.733 |
| $4 \leftrightarrow 5$ | | | 0.326 | 1.026 | | |
| | **pot** | | **pot2** | | **ppy** | |
| | $\epsilon_{mean}$ | $\epsilon_{std}$ | $\epsilon_{mean}$ | $\epsilon_{std}$ | $\epsilon_{mean}$ | $\epsilon_{std}$ |
| $1 \leftrightarrow 2$ | -0.024 | 0.096 | -0.447 | 0.185 | 0.163 | 0.166 |
| $2 \leftrightarrow 3$ | 0.407 | 0.139 | -0.100 | 0.178 | 0.466 | 0.849 |
| $3 \leftrightarrow 4$ | -0.202 | 0.172 | -0.442 | 0.192 | -0.122 | 0.458 |
| $4 \leftrightarrow 5$ | -0.461 | 0.160 | 0.160 | 0.150 | -0.135 | 1.597 |
| $5 \leftrightarrow 6$ | -0.231 | 0.113 | 0.348 | 0.203 | | |
| $6 \leftrightarrow 7$ | -0.362 | 0.082 | 0.455 | 0.143 | | |
| $7 \leftrightarrow 8$ | | | -0.342 | 0.965 | | |
| $8 \leftrightarrow 9$ | | | -0.215 | 0.186 | | |
| $9 \leftrightarrow 10$ | | | 0.417 | 0.277 | | |
| $10 \leftrightarrow 11$ | | | 0.247 | 0.176 | | |
| $11 \leftrightarrow 12$ | | | 0.379 | 0.223 | | |
| $12 \leftrightarrow 13$ | | | -0.321 | 0.197 | | |
| $13 \leftrightarrow 14$ | | | -0.463 | 1.106 | | |
| $14 \leftrightarrow 15$ | | | -0.369 | 0.364 | | |

## 4.3 Dense matching

The work on dense stereo matching appeared a long time ago but still draws the attention of computer vision community. The general setup is the following: two images, left and right or first and second, are obtained by translating a camera along the horizontal axis by a distance $b$ that determines the baseline. Such motion ensures that images are coplanar. However, we know that in case of SEM, the set of correspondences obtained by moving the camera in the image plane is useless as affine camera is invariant to such motion. It means that all 2D points will be just shifted from one image to another which does not give any information about depth. Hence, we use an out-of-plane rotation and rectification helps to make all epipolar lines horizontal. In this way, all necessary conditions for dense matching are respected.

The output of the dense matching algorithms is the disparity map. Disparity is the measure of horizontal displacement of a pixel from one image to another. Assuming that a correspondence is determined by points $(q_x, q_y)^\top$ in left image and $(q'_x, q'_y)^\top$ in right image, the disparity is given by:

$$\delta = q'_x - q_x \tag{4.5}$$

under the assumption that $q_y = q'_y$. This principle is illustrated in Figure 4.5. The disparity is closely related to the depth coordinate. Going forward, the depth may be

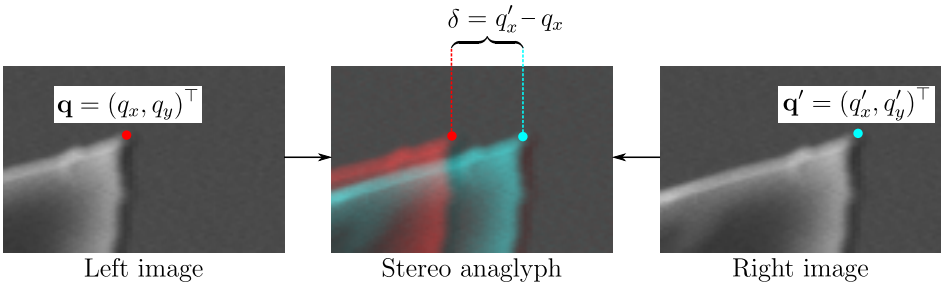



Figure 4.5: Example of disparity calculation for one image pixel. Images are rectified, i.e. $q'_y = q_y$.

obtained from disparity by means of triangulation presented in Section 4.4.

All dense matching methods may be subdivided into three groups: local or block matching, semi-global matching and global techniques [SS02]. Block matching algorithm performs a comparison between the block of pixels in the first image with the line of the same width in the second image [Kon98]. Thus, by moving the block across this line, a similarity score is attributed to every position. Finally, the pixel window (block) with the highest similarity score is considered as a match. The algorithm is repeated for all pixels of the first image. Various ways exist allowing to measure this similarity score: SAD, SSD, NCC (Normalized Cross Correlation), or POC (Phase-Only-Correlation) [SIA+11]. The obtained disparity is often subject of refinement methods [MHW+13]. Local dense matching algorithms assume that the disparity is constant inside the correlation window which is not true at the depth discontinuities. The estimation is also challenging on a low textured regions.

Methods of global disparity usually have a strong link with Markov random field (MRF) problem [FDM15]. An energy function is constructed in a way allowing to impose the smoothness constraint, i.e. to avoid strong variations of disparity for the neighboring pixels. The optimization consists in warping the first image into the second one while varying the displacement vector map until high similarity level is achieved. High execution time is often the price for the efficiency of such algorithms.

In 2005, *Hirschmuller* proposed a new method lying between global and local dense matching: semi-global matching algorithm (SGM) [Hir05]. Nowadays, it is one of the most popular ways of disparity map estimation, because of its performance and good efficiency [HK12] and this is the dense matching algorithm we use. Matching problem is expressed as a cost function containing the similarity term, intensity values of two images, and a 2D smoothness term allowing to avoid noisy disparities.

When working with SGM, we need to provide two parameters for the algorithm. The first parameter is the size of the matching block. This value depends mainly on whether the image contains a lot of features or not. For high textured images this value can be lowered. Generally, we started with the value of 15 and then refined it for every dataset separately, if needed. The second parameter is the disparity range, which depends on the baseline and on the object form. It should also be adapted for every sequence.

Once the disparity map is estimated, it then passes through bilateral filter. This technique is quite common, as it allows to assure the smoothness of final 3D reconstruction [Por08; YTA09]. The obtained disparity maps are shown in Figure 4.6 and Figure 4.7. The corresponding parameters are given in Table 4.2.

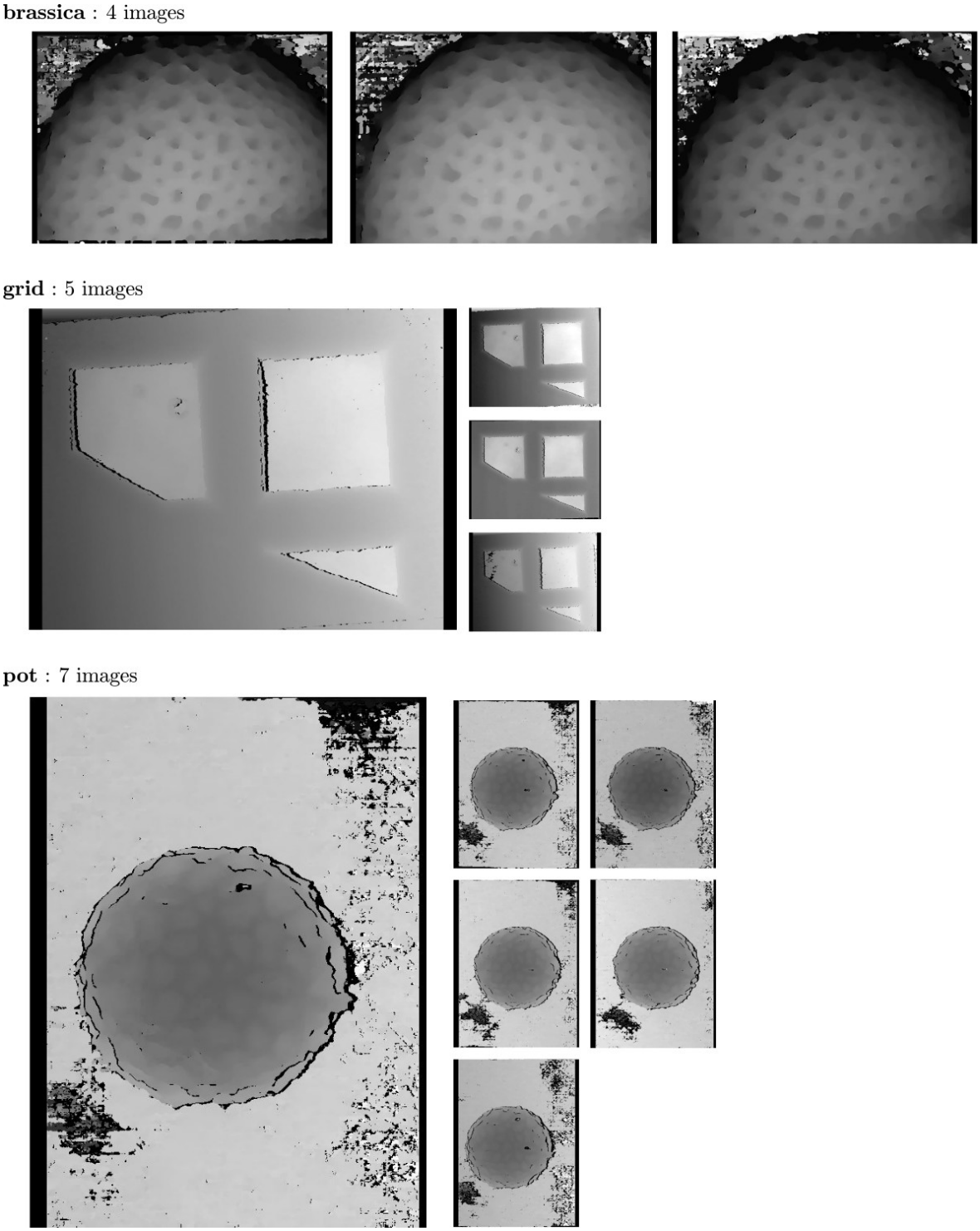



Figure 4.6: Disparity maps for SEM image sequences. Page 1.

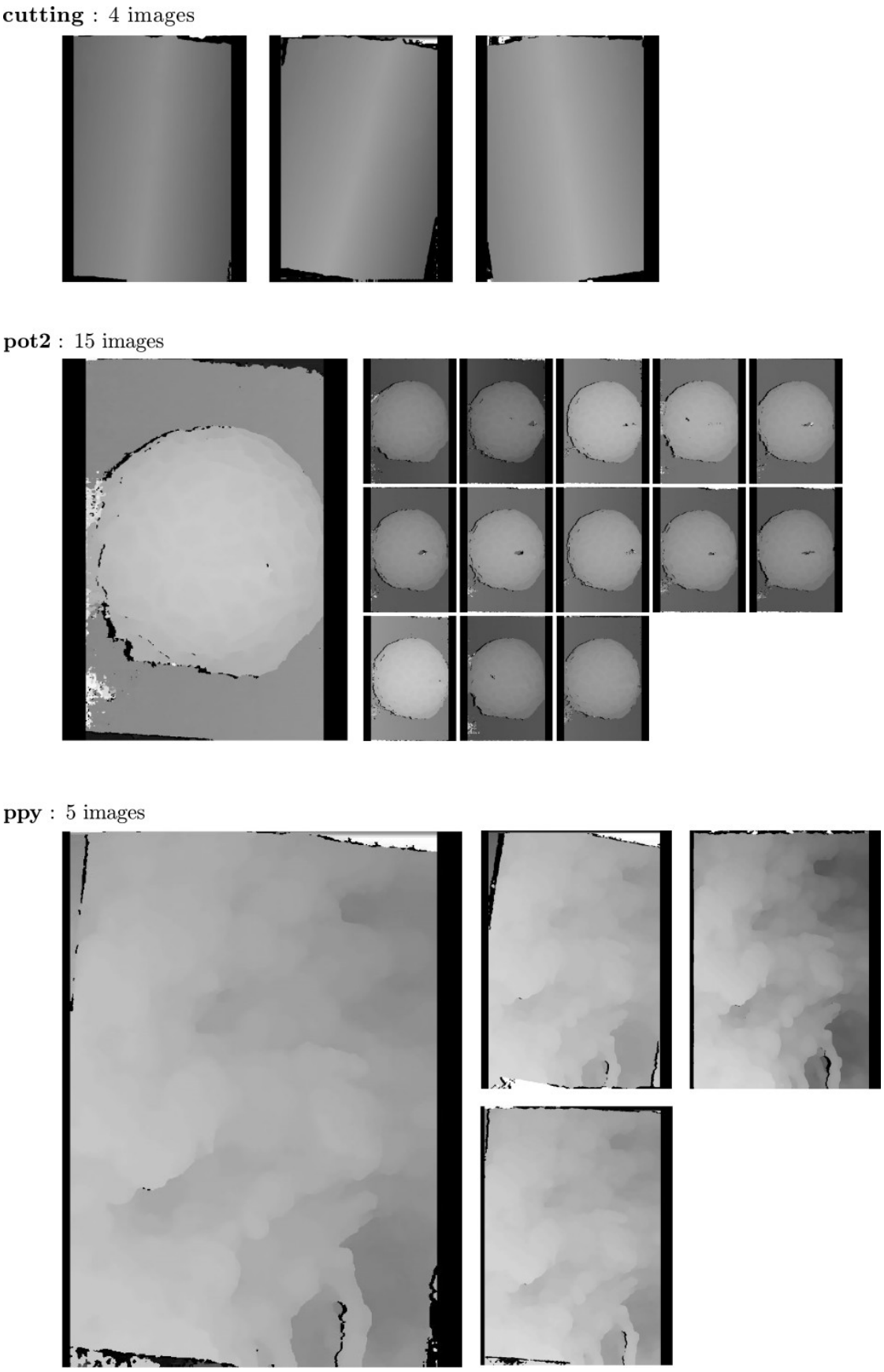



Figure 4.7: Disparity maps for SEM image sequences. Page 2.

Table 4.2: Dense matching parameters for different SEM datasets.

| | method | disparity range, pixels | block size, pixels |
|---|---|---|---|
| **brassica** | SGM | 16·[−2 1] | 15 |
| **grid** | SGM | 16·[−2 3] | 15 |
| **cutting** | SGM | 16·[−4 7] | 21 |
| **pot** | SGM | 16·[−2 5] | 19 |
| **pot2** | SGM | 16·[−4 3] | 9 |
| **ppy** | SGM | 16·[−3 2] | 9 |

## 4.4 Triangulation

Once the disparity map is computed, we can obtain a dense equivalent of measurement matrix in the following way. The coordinates of the pixels in the first image are left unchanged. Thanks to the disparity map, we know that the correspondence for every pixel is located at the same point but shifted with the value of disparity. Therefore, the new dense measurement matrix $\mathcal{W}_d$ has the following form:

$$\mathcal{W}_d = \begin{pmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_N \\ \mathbf{q}_1 + \begin{pmatrix} \delta_1 \\ 0 \end{pmatrix} & \mathbf{q}_2 + \begin{pmatrix} \delta_2 \\ 0 \end{pmatrix} & \cdots & \mathbf{q}_N + \begin{pmatrix} \delta_N \\ 0 \end{pmatrix} \end{pmatrix} \tag{4.6}$$

with $N = N_{pts}$ number of 3D points. In this example the dense measurement matrix is presented for the two-view case. This is the result of algorithms presented previously.

The goal of this chapter is, once a dense set of correspondences is estimated, find the position of 3D points which is done by means of triangulation. Therefore, the problem of triangulation is stated as follows: given camera matrices and the matched projections of 2D points, determine the position of points in 3D space:

$$\mathcal{Q} = \tau(\mathcal{W}_d, \mathcal{P}) \tag{4.7}$$

Or, for the simplest two-view case and one 3D point:

$$\mathbf{Q}_j = \tau(\mathbf{q}_j, \mathbf{q'}_j, \mathbf{P}, \mathbf{P'}) \tag{4.8}$$

In fact, the function $\tau$ allows to find the intersection of two projection rays defined by camera matrices and 2D points. This task may seem trivial, but, in presence of noise, the 3D projection rays will never intersect. So, many different triangulation methods were developed to find an optimal solution to this problem [HS97; Lin10]. The most common is the linear triangulation which allows to transform the triangulation problem to the form $\|\mathbf{AQ} = 0\|$. The matrix $\mathbf{A}$ is obtained by developing the cross product expression $\mathbf{q} \times \mathbf{PQ}$. Thus, taking two camera matrices $(\mathbf{P}, \mathbf{P'})$ we obtain a linear set of equations in the elements of a 3D point $\mathbf{Q}$:

$$\begin{pmatrix} q_x\mathbf{p}_3 - \mathbf{p}_1 \\ q_y\mathbf{p}_3 - \mathbf{p}_2 \\ q'_x\mathbf{p}'_3 - \mathbf{p}'_1 \\ q'_y\mathbf{p}'_3 - \mathbf{p}'_2 \end{pmatrix} \mathbf{Q} = 0 \tag{4.9}$$

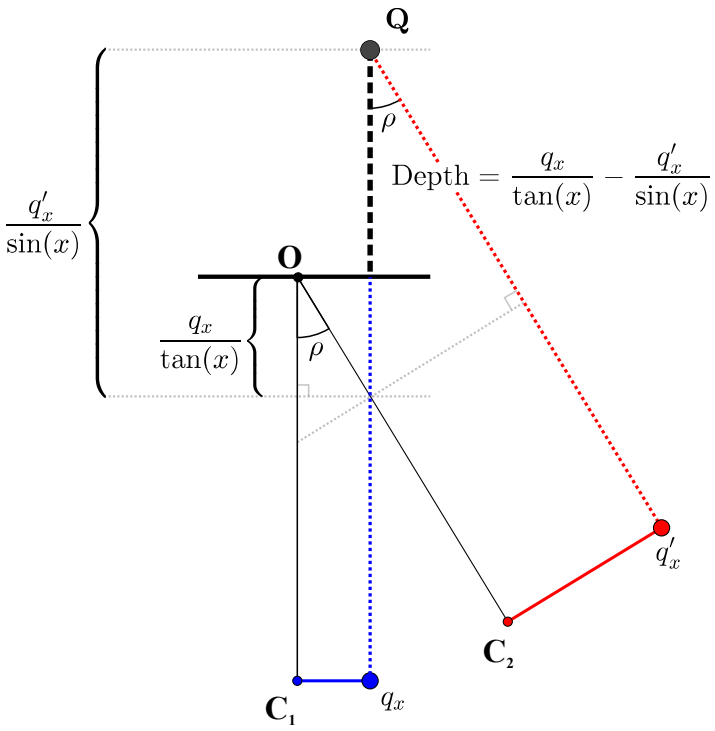



Figure 4.8: Principle of direct triangulation.

where $\mathbf{p}_i$ is the $i$-th row of the corresponding camera matrix. This equation may be solved by taking the pseudo-inverse of $\mathbf{A}$ which gives a least-squares solution. This method will be further referred as *Linear-LS*. It is important to add that if a point is visible in more than two cameras, every new camera is easily added to triangulation (by adding two new rows to the matrix $\mathbf{A}$).

While there exist many other triangulation methods, the main difference between them is the invariance or not to affine and projective transformations. Such methods are very important in applications where an affine or projective reconstruction are sufficient. In our case, Euclidean reconstruction is obtained, and in [HS97], authors came to the conclusion that for Euclidean reconstruction with the goal of 3D error minimization, the performance of all these algorithms is very similar.

On the other hand, in recent works on 3D reconstruction in SEM [BTO+17; Xie11], authors proposed a way of points triangulation based entirely on the two-view geometry in parallel projection case. In this work, it will be referred as *Direct*. The relevant geometry is displayed in Figure 4.8 and the expression for 3D coordinates is:

$$\begin{cases} Q_x = q_x \\ Q_y = q_y \\ Q_z = \frac{q_x}{\tan(x)} - \frac{q'_x}{\sin(x)} \end{cases} \tag{4.10}$$

This result has a direct link with *Linear-LS* algorithm. The expression (4.10) can be obtained by developing (4.9) for the case of pure rotation about $\vec{y}$ axis of the camera (the angle $\rho$ in present notations). However, an important difference exists. First, *Direct* works only for rectified images. Secondly, only *Linear-LS* is easily extendable to the multi-view case. The last point and the most important one is that *Direct* algorithm finds the position of 3D points that is optimal only for the first image, two

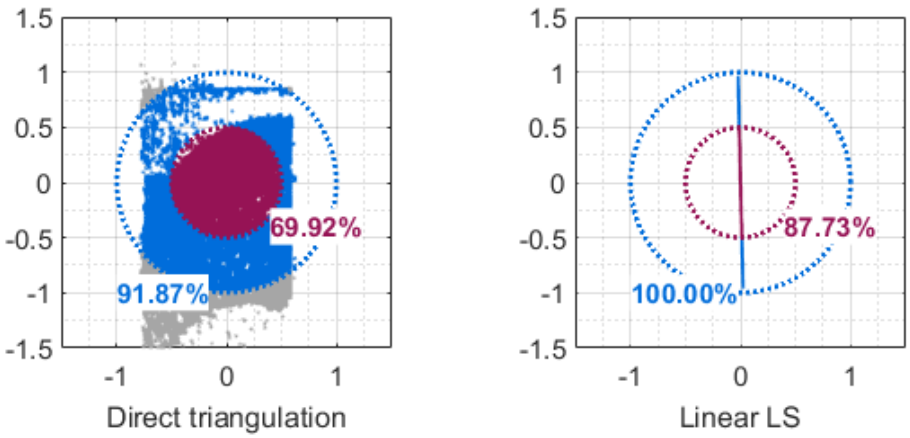



Figure 4.9: Reprojection error for different triangulation methods. An image pair from **brassica** dataset was used.

out of three coordinates depend only on the first image. This fact is also proven by the analysis of reprojection error (Figure 4.9) where triangulation was tested on the first image pair from **brassica** dataset. In case of *Direct* triangulation the reprojection error for the 2D points in the first image is zero, all errors are transmitted to the projections in the second image. We see in the results of triangulation that for 91% of points the error is below one pixel. For *Linear-LS* triangulation the results are better, all reprojection errors are inside one pixel range and are distributed between the projections in the first and second images.

The last point we want to add is that, even if *Direct* triangulation has lower accuracy comparing to *Linear-LS* triangulation, it is less time consuming as there is no need to calculate the pseudo-inverse. All depths are recovered at the same time just by using matrix subtraction and multiplication by a scalar factor. For example, triangulation of 503,082 points using *Linear-LS* took 16.9 seconds comparing to 57 ms for *Direct* method. Therefore, as both methods give very similar results, for visualization, one can choose *Direct* method, and linear least squares if higher level of accuracy is needed.

Triangulation is the last step of 3D reconstruction covered in this thesis. It allows to obtain a dense point cloud corresponding to the object structure that can already be exploitable for measuring 3D object properties. The result of triangulation, and, therefore, the final results of 3D reconstruction for all six image datasets used throughout this manuscript are displayed in Figure 4.10 and Figure 4.11.

## 4.5 Conclusion

In this chapter, devoted to dense 3D reconstruction, we presented a group of methods allowing to upgrade sparse reconstruction to dense one containing possibly millions of points. This upgrade includes the following steps. First, images are rectified in order to simplify the search space for one correspondence to a horizontal epipolar line. Being based on the special geometry between two affine views, it is direct and fast as the rectifying transformations are obtained directly from the elements of fundamental matrix. Moreover, the slopes of epipolar lines, that are used for calculation of image rotation, were already optimized previously as they are part of parameter vector in

autocalibration task. Therefore, the values obtained originally from the fundamental matrix are then refined at the step of autocalibration which improves the level of accuracy.

Once the images are rectified, we apply the algorithm of Semi-Global dense matching to obtain a disparity map, i.e., the displacement of every pixel from the first image to the second one. These values are then used in the triangulation process where a dense 3D point cloud is generated from disparities.

Finally, the methods were tested on six microscopic samples: on both biological and microfabricated objects. The resulting point clouds contain from 500,000 to about 4 millions of 3D points. Moreover, the original colors were added to 3D points which improves dramatically the quality of visual perception of 3D reconstruction.

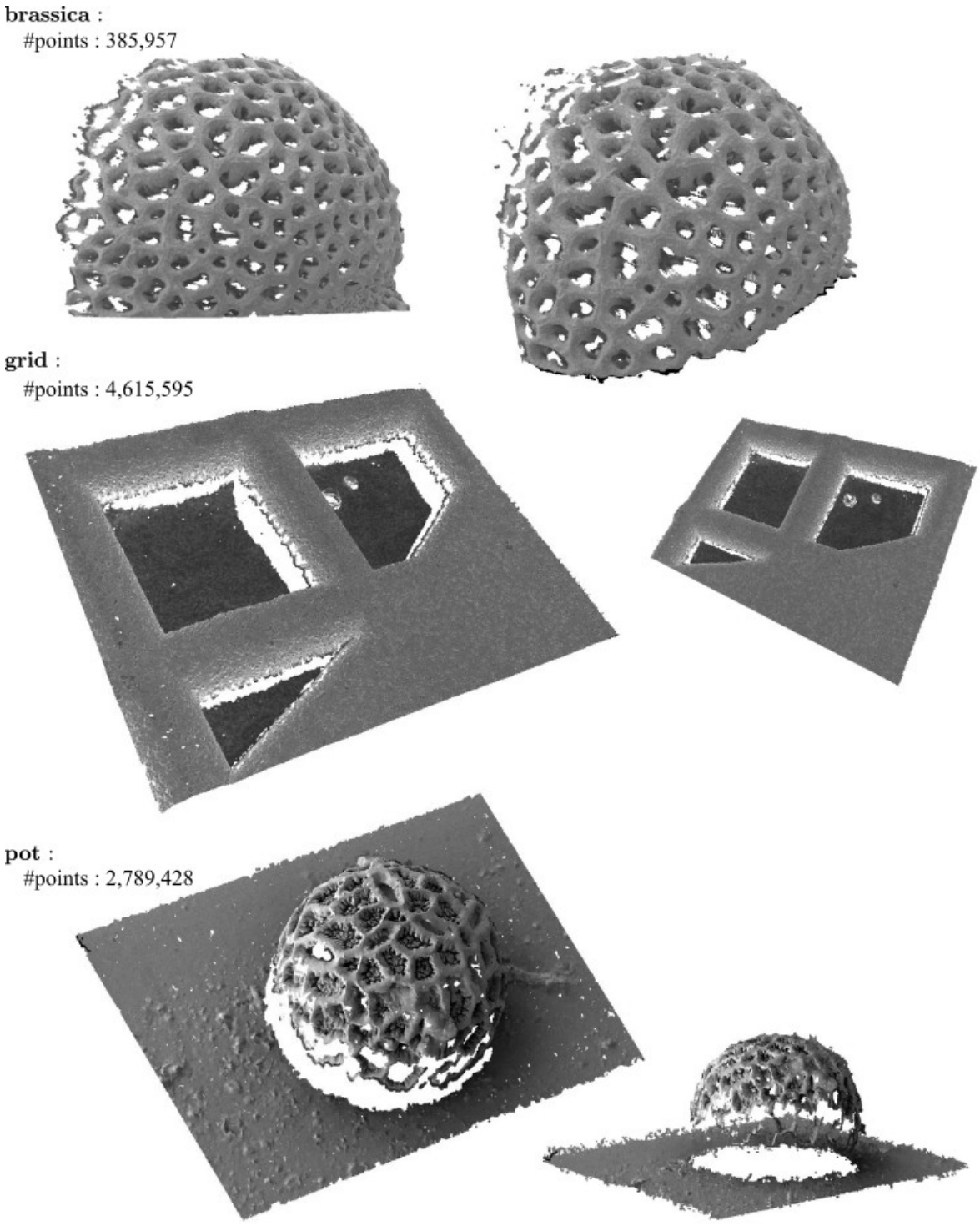



Figure 4.10: 3D reconstructions for SEM image sequences. Page 1.

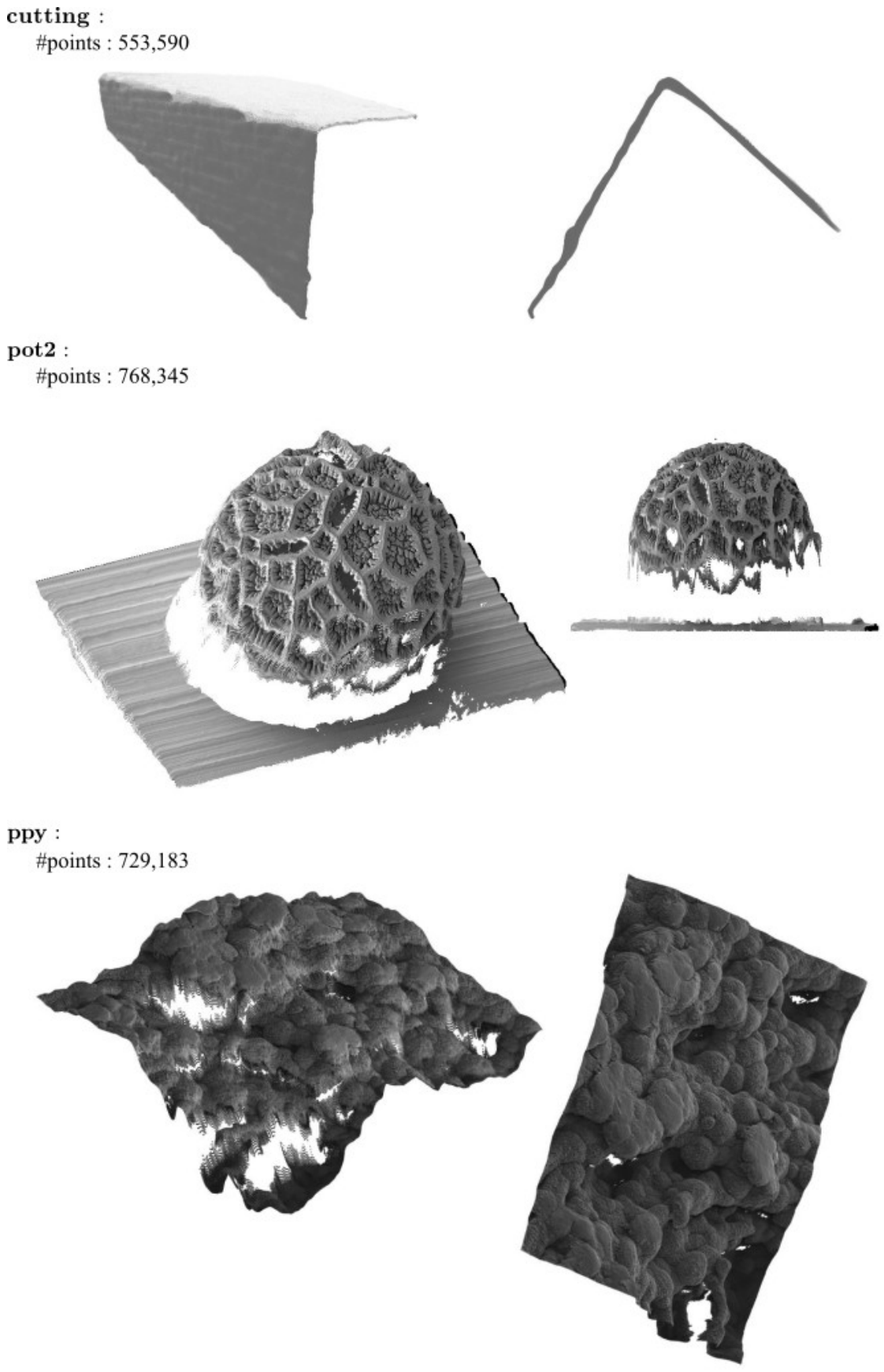



Figure 4.11: 3D reconstructions for SEM image sequences. Page 2.

# Chapter 5

# Towards automatic image acquisition

## Contents



*Every 3D reconstruction starts with image acquisition. While this process is easily controlled in macroscale, it may be really difficult in SEM. Typically, in order to acquire several images of the specimen, even an experienced operator may spend more than an hour. The problem is that when looking at small objects, the field of view is limited. It means that even a small movement of the robot arm leads to a huge object displacement comparing to the size of the viewed area. In this chapter, we make a step towards automatic image acquisition in SEM. To do this, two techniques will be presented. First, we describe an algorithm allowing to maintain the target object in focus while it is moving which is equivalent to controlling the depth-position. Secondly, a method of robot and tool center point (TCP) calibration is proposed. The precise knowledge of TCP combined with calibrated robot kinematics will allow performing the object motion exactly about the point of our choice, i.e. keeping it in sight.*

## 5.1 Problem statement

All methods of 3D reconstruction begin from a sequence of images. In order to acquire it, a user may have to spend hours of work which is mostly due to the problem of maintaining the object in the field of view and in focus. Consider the following example. The sample is about 100 µm size and it is located somewhere on the robotic stage (e.g. a robotic arm), several centimeters from the rotation center. Even if we apply an in-plane rotation of 3 degrees, the object may be millimeters away from the initial position. Hence, an operator needs to relocate the object and the situation repeats itself for every new image. In this chapter, we describe three methods allowing to make an important step towards automatic image acquisition.

The first one is the technique based on focus information. When an object performs an in-plane rotation, it leaves the field of view which can be compensated by in-plane translations, i.e. with 2 degrees of freedom (DOF). However, in the case of more complex out-of-plane rotation, the displacement of object center is characterized by three coordinates, two same in-plane translations and translation along the optical axis, moving away or towards the camera. As it was mentioned previously, this last translation is not taken into account by the projection process in case of an affine camera (Result 1.2.2), yet, it has to be considered as SEM has limited depth of field, an area in which the object stays in focus. Obviously, blurry images is not a convenient start point for 3D reconstruction. Therefore, it is important to *maintain* the object in focus while it is moving. We will refer to this problem as dynamic autofocus and explain the proposed method in Section 5.2.

It is important to mention that autofocus controls only one degree of freedom which is the distance from camera to object. Nevertheless, the object may still go out of the field of view. The solution is simple, move the initial position of rotation point (tool center point in robotics notation) to the object center. It requires not only the precise knowledge of the robot kinematic model but also the transformation from the terminal robot link to the end effector and both these elements can be estimated by the methods presented in Section 5.3 and Section 5.4.

## 5.2 Dynamic autofocus

Autofocus is a very useful feature for all types of visual sensors and, in particular, for SEM: it makes possible to obtain sharp images with the least human intervention. Of the two usual types of autofocus, active with the use of a rangefinder, and passive with the use of images, only the last type is implemented in electron microscopy [NMY+13; RMM10; WWZ12]. Autofocus can be also classified as static or dynamic depending on whether the target object is fixed or moving, respectively, because the same principles are not appropriate for both cases. In electron microscopy, static autofocus is the most widespread because in most applications objects are static. However, with the appearance of new applications in microscopy such as 3D reconstruction that requires smooth acquisition of multiple images with different object positions [DSH+12; HDD+04; KDZ+10; TKA+15] or robotics in the microscope that requires 3D object tracking, i.e. including depth measurement, [ASP+14; BDN+06; MJC+14; RT12; SLY+16; ZTF+13] static autofocus is no longer appropriate and dynamic autofocus

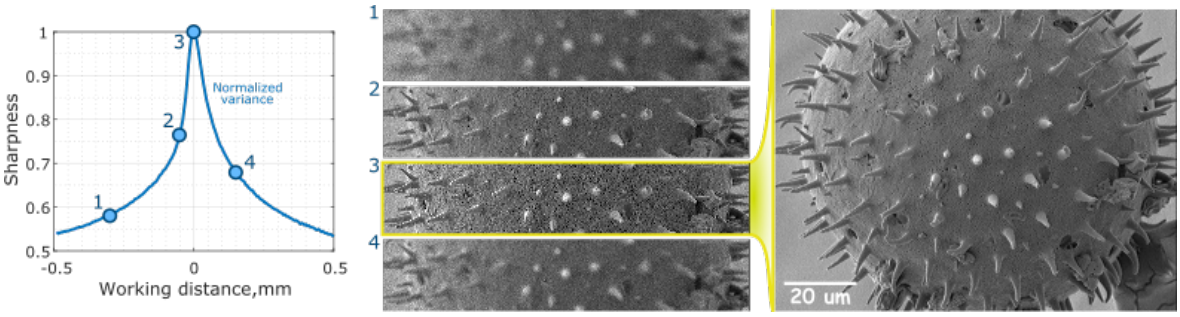



Figure 5.1: Left: sharpness function for a pollen grain of daisy flower (*Bellis Perrenis*). Middle: image ROIs for points in sharpness function. Right: in-focus image that corresponds to point 3 on sharpness function. Magnification: ×1000.

needs to be developed. The object needs to stay in focus for the whole operation time.

It is important to notice the relation between autofocus and depth estimation using defocus information. Actually, these tasks are the same. Yet, the goal of autofocus is to adapt the value of focal distance in the range determined by the depth of field while depth estimation aims at obtaining the value with the smallest possible error. The algorithm presented below is initially developed for keeping an object in focus, however, it allows to estimate the depth coordinate precisely enough for such applications as automatic manipulation, assembly or robot calibration.

When working with autofocus algorithms, the in-focus image is defined as follows: any change of focal distance, or working distance in case of SEM, or equivalently the change of object distance from the camera, will not give a sharper image than the in-focus image. There exist many different techniques of measuring how sharp an image is. They may be based on statistical information, image gradient, Fourier or wavelet transforms. The dependence of image sharpness from the depth variation or equivalently from focal distance variation is called sharpness function (see Figure 1.3). It has a special form characterized by the presence of a maximum in the point where an image is in focus. Depending on the object form, the function may have several maxima if object parts are located at different distances from the visual sensor. A very extensive comparative study of focus measure operators for general scenes was realized by Pertuz *et al.* in [PPG13]. As for microscopy domain, one can refer to the study made by Rudnaya *et al.* [RMM10]. After evaluating several sharpness functions, the normalized variance was selected:

$$S(\mathbf{I}) = \frac{1}{MN}\frac{1}{\mu}\sum_{M}\sum_{N}(\mathbf{I}(q_x, q_y) - \mu)^2 \tag{5.1}$$

where $S(\mathbf{I})$ is the sharpness of image $\mathbf{I}$, $\mu$ is the mean of intensity values, $M$ and $N$ are image width and height, respectively. An example of sharpness function as well as some images at different defocus values are presented in Figure 5.1. The object is a pollen grain of daisy flower (*Bellis Perennis*).

In case the in-focus image is constant, we speak about static autofocus. As the sharpness is a function of the working distance, autofocus may be considered as a problem of optimization: the search for the peak of sharpness along the optical axis. When the object is not moving, this function is constant and the desired image stays

the same. Besides the simple method of sweeping all possible values of working distance and choosing the one with the best sharpness score, two main types of static autofocus can be distinguished. They differ on whether the model of sharpness function is used explicitly ([NJS97; RMM+12; WWZ12]) or not [HZH03; MTD+13; OPT98].

Contrary to static autofocus, in dynamic autofocus task the in-focus image varies in time due to the object displacement. Previously mentioned methods can not be applied in this case as the in-focus image is not constant. It also implies that dynamic autofocus represents the continuous search for the maximum value of sharpness. This process is equivalent to estimate the depth coordinate using defocus information. Therefore, we propose a method of *keeping* object in focus online, during its movement, both in translation and rotation. The proposed method, being based on online stochastic optimization, has the following advantages compared to literature solutions:

- absence of calibration step, no model is used, i.e. there is no need for training data, no focus sweeping of the scene;
- the algorithm is invariant to the object structure which is directly derived from the previous point;
- no scanning procedure during operation, only two images are used to estimate best focus position (depth variation);
- being robust to noise, it allows to work with high frame rate (approximately 5 Hz) that is confirmed by experiments;
- adaptive to the variation of object speed.

### 5.2.1 Sharpness optimization

As it was stated previously, the presented approach of dynamic autofocus is based on mathematical optimization. In the context of present work, the objective function is the sharpness function with working distance as an input parameter. Equivalently, the input parameter may be replaced by object translation perpendicular to the image plane (depth coordinate). In the case of dynamic autofocus, the function is changing in time, i.e. is non-stationary. The goal of the method is to keep the sharpness in maximum, thus, continuously update the value of $\xi$, so that:

$$\xi_n = \arg\max_{\xi \in \mathbb{R}} f_n(\xi) = \arg\min_{\xi \in \mathbb{R}} (-f_n(\xi)) \tag{5.2}$$

where $\xi_n$ is the current value of the working distance that maximizes the current image sharpness ($f_n(\xi) = S$). The objective function may contain one or several maximums depending on the scene structure. However, considering that the starting point of the dynamic autofocus is a well-focused image (not necessarily the best-focused one), the function may be considered convex in the neighborhood of the maximum point. The size of this neighborhood is equal to the current depth of field of the microscope.

To find a solution $\xi_n$ we use local optimization algorithms that differ on whether they use only function evaluations, first order derivative (gradients) or second order derivative (Hessians). For the algorithms that belong to the first group, most of them,

such as Golden-section search [Kie53], are based on the reduction of the interval that contains the maximum. They are not suitable for our application because the objective function is not stationary. The second group contains the approaches based on derivative. In autofocus problem, neither the sharpness function nor its derivatives are available. The only possible solution is to use the approximations which are not readily apparent to the unknown function in the case where only noisy measurements can be obtained. To approximate the first order derivative we use the method of centered differences:

$$f'(\xi_n) = \frac{f(\xi_n + \Delta\xi) - f(\xi_n - \Delta\xi)}{2\Delta\xi} \tag{5.3}$$

The presence of noise makes irrelevant the idea to use second-order approximation: apart from the fact that it requires more images for one Hessian estimation, its value would likely be unusable due to the high noise level at high frame rate. All these factors confine the choice of the optimization algorithms to the first-order methods. The most used of them is the gradient descent or ascent in our case (the difference consists only in the movement direction). It has the following update rule:

$$\xi_{n+2} = \xi_n - \alpha f'(\xi_n) \tag{5.4}$$

where $\alpha$ denotes the gain or learning rate. Its value determines how important is the update in one iteration. In the context of the present work, $\xi_{n+2}$ represents the estimate of the working distance that would give the best value of image sharpness. As the evaluation of $f'(\xi_n)$ takes two images, the time elapsed between $\xi_{n+2}$ and $\xi_n$ is twice the time of one image acquisition, that is why at odd time moments $(t_{n+1}, t_{n+3}, t_{n+5})$ the update is not performed.

In general, gradient descent achieves good results when the objective function is not corrupted by noise, which is not true for the sharpness of SEM images taken at high frame rate ($\geq$4.5 Hz). When there is an important change in the gradient value, which is chaotic due to noise, the algorithm will change dramatically the value of working distance and lose the focus. Another drawback of gradient descent consists in high dependence on the value of $\alpha$. If the gain is too low, the convergence speed will also be low. In the case of autofocus, it would greatly limit the maximum displacement speed. In contrast, if the gain is too high, the algorithm may suffer from oscillations about the maximum value. Therefore, several techniques were proposed in the literature to improve the performance of gradient descent such as *Momentum* [Qia99], *AdaGrad* [DHS11], *RMSProp* [TH12], *Adam* [KB14] (see Appendix E for more details). The last one is then adapted for the task of dynamic autofocus.

*Adam*, is a recently introduced method that stands for adaptive moment estimation. The idea proposed here is not to use directly the gradient but its exponentially weighted moving average. In addition, it also stores the exponentially moving average of the gradient squared:

$$\begin{cases} m_{n+2} \leftarrow \beta_1 m_n + (1-\beta_1) f'(\xi_n) \\ v_{n+2} \leftarrow \beta_2 v_n + (1-\beta_2) f'(\xi_n)^2 \\ \xi_{n+2} \leftarrow \xi_n - \alpha \frac{m_{n+2}}{\sqrt{v_{n+2}+\varepsilon}} \end{cases} \tag{5.5}$$

where $m$ is the first moment variable, $v$ is the second moment variable and $\varepsilon$ is a small constant (typically $10^{-8}$) allowing to avoid division by zero at first iterations.

**Algorithm 2** Dynamic autofocus in SEM

$step \leftarrow 1$
$\alpha \leftarrow 0.004$ ▷ optimization parameters
$\beta_1 \leftarrow 0.6$
$\beta_2 \leftarrow 0.6$
$\Delta\xi \leftarrow 10^{-6}$
$\varepsilon \leftarrow 10^{-8}$
$m \leftarrow 0$
$v \leftarrow 0$
$\xi \leftarrow \xi_0$ ▷ initial working distance
set working distance $\xi = \xi + \Delta\xi$
**while** autofocus is activated **do**
    acquire image $\mathbf{I}_n$
    get sharpness score $s \leftarrow S(\mathbf{I}_n)$ (Eq. 5.1)
    **if** $step = 1$ **then**
        evaluate $f(\xi + \Delta\xi) = s$
        set working distance $\xi = \xi - \Delta\xi$
        $step \leftarrow 2$
    **else**
        evaluate $f(\xi - d\Delta\xi) = s$
        estimate gradient $\hat{g} = f'(\xi_n)$ (Eq. 5.3)
        $m \leftarrow \beta_1 m + (1 - \beta_1)\hat{g}$
        $v \leftarrow \beta_2 v + (1 - \beta_2)\hat{g}^2$
        $\xi \leftarrow \xi - \alpha \frac{m}{\sqrt{v}+\varepsilon}$
        set working distance $\xi = \xi + \Delta\xi$
        $step \leftarrow 1$
    **end if**
**end while**

Such formulation of update rule allows better filtering of the gradient while keeping the functionality needed for non-stationary function optimization. Typical values of parameters are: $\beta_1 = 0.9, \beta_2 = 0.99, \varepsilon = 10^{-8}$.

The final algorithm for dynamic autofocus is represented in Algorithm 2. It should be noted that during operation, actual working distance is never equal to $\xi$. Instead, the algorithm sets the working distance to $\xi \pm \Delta\xi$ to continuously estimate the derivative. In other words, the actual value of working distance oscillates around the best focus position, which is updated every two images.

### 5.2.2 Experiments

The validation of the presented theoretical aspects was confirmed by three groups of experiments: translation along Z axis (optical axis of the camera), translation along Z axis with varying parameters (speed, magnification), and rotational movement. The object used is a pollen grain of daisy flower that will be further referred as *Bellis Perennis*. The object is mounted on the robot (6 degrees of freedom) installed inside the SEM. For the rotational movement, additionally to the *Bellis Perennis*, the algorithm was tested on a scene with different objects including pollen grains: *Pollen Grains*. Both objects were coated with a layer of gold. The equipment used is a SEM Carl Zeiss AURIGA 60. The following parameters were constant for all of the experiments: acceleration voltage 3 kV, aperture size 30 μm. The frame rate was also constant, with the value of 4.5 Hz.

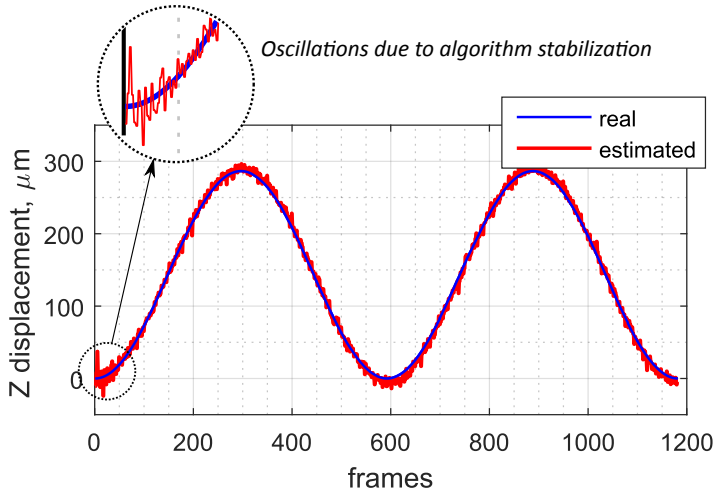



Figure 5.2: Results of dynamic autofocus in SEM. Object is performing a translation along Z axis, the software adapts the working distance to keep it in focus. Object: pollen grain *Bellis Perennis*. Magnification: ×1000. Maximal speed: 10 µm/s.

Before starting the description of the experiments, it is important to give some information about the initialization of the algorithm. Three values are to be defined: $m_0$, $v_0$ and $\xi_0$. The values of $m_0$ and $v_0$ are equal to zero. As a drawback, it leads to the fact that the values of $m_n$ and $v_n$ are biased towards zero at the initial steps as mentioned in [KB14]. It can be seen from the experiments that there are some oscillations at the first 5-15 frames in the values of estimated depth that disappear afterward (Fig. 5.2). Thus, we consider that it would be a good idea to activate the dynamic autofocus several images before the movement starts as it was done in one of the experiments. Another important variable is $\xi_0$. The dynamic autofocus represents the tracking of the best sharpness position. A good choice would be a value of $\xi_0$ that is close to the peak, i.e. in the depth of field. If the value is chosen far from maximum, the algorithm *may* catch up with the best focus value, but the convergence speed needs to be bigger than the movement speed. In addition, a different set of parameters should be used and nothing can guarantee that this set will still be optimal when the maximum will have been reached.

**Translation along Z axis**. The first experiment consisted in the object performing a translation along Z axis (along the optical axis of the camera), Figure 5.2. The speed was defined as a sine function with its maximum of 10 µm/s. It allowed comparing the results of dynamic autofocus with actual displacement which was known from the proprioceptive detectors of the robot. Results demonstrate that not only the object was correctly positioned inside the depth of field but it was performed with high accuracy (for the given example the depth of field was about 40 µm). The standard deviation of error is about 5 µm while the object dimensions are about 100 µm, magnification ×1000. It allowed confirming the viability of the proposed approach. The next step was the test that allowed evaluation of the robustness of the proposed solution.

**Varying object speed**, Figure 5.3(a). In this example, once again, the object had the sine function as speed profile. The maximal speed for each test was 10, 20 and 40 µm/s, respectively. All optimization parameters, as well as magnification (×1000), were held constant, only the object speed was subject to change. The results show that

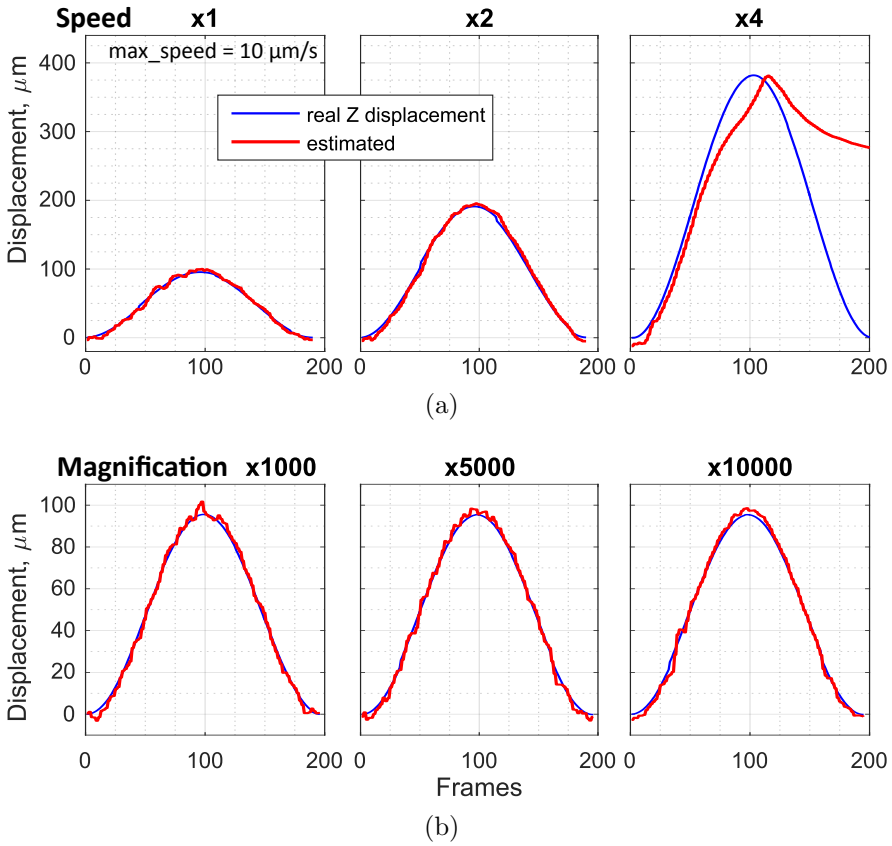
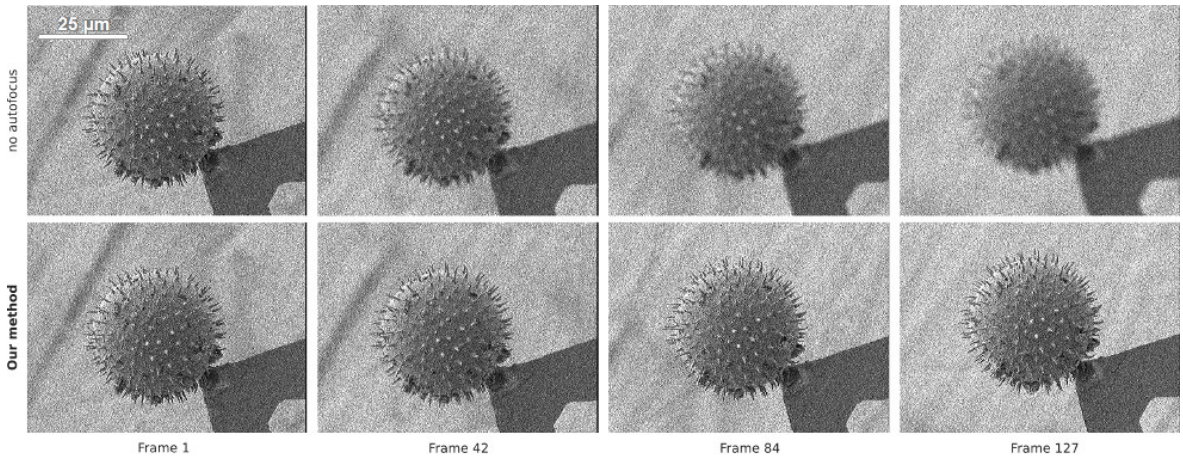



Figure 5.3: Performance of dynamic autofocus algorithm with varying: a) speed, b) magnification.

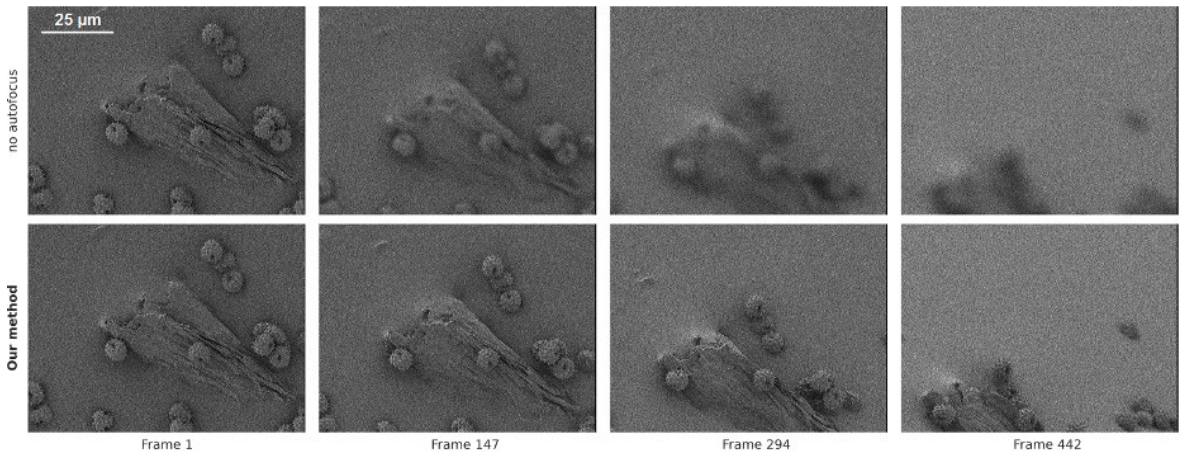
the algorithm performs well when the speed is multiplied by a factor of two. However, it fails when the speed is multiplied by four. To overcome this, it is necessary to adapt the value of the parameter $\alpha$. In general, the maximum object speed, for which dynamic autofocus is still viable, is principally limited by the value of frame rate that was equal to 4.5 Hz in this experiment.

**Varying magnification**, Figure 5.3(b). In the next experiment, the algorithm was subject to changing magnification while the speed was the same. Three different values were tested: $\times 1000$, $\times 5000$, $\times 10000$. The approach presents a high level of robustness. The value of standard deviations for all three cases is about 2 µm. For further increasing of magnification, the value of $\Delta\xi$ should be adapted as the depth of field becomes smaller with growing magnification.

**Rotating objects** (Figure 5.4). The last experiment was conducted to test the performance of the algorithm on rotating objects. Two scenes were used: *Bellis Perennis* and *Pollen Grains*. Rotation speed was constant and fixed to 0.2 degrees per second. Magnification was $\times 500$ and $\times 400$, respectively. For the first object, *Bellis Perennis*, the total rotation was 15 degrees. The scene *Pollen Grains* rotated to approximately 60 degrees. It should be noted that the center of the scene was not aligned with the rotational axis of the robot. Thus, when the robot performs the rotational movement, the object rotates but not precisely about its center. It means that the rotational movement of one robotic axis results in a more complex movement of the object, i.e.




(a) *Bellis Perrenis* scene. Rotation speed: 0.2 deg/s. Magnification: ×500.




(b) *Pollen grain* scene. Rotation speed: 0.2 deg/s. Magnification: ×400.

Figure 5.4: Dynamic autofocus algorithm on rotating objects in SEM.

rotation combined with uncontrollable translations. That is why, without autofocus, the object goes out of the depth of field. In contrast, when the dynamic autofocus is activated, the image stays sharp during the whole movement even when the scene highly differs from the beginning operation to the end like in the case of *Pollen Grains*: at the final frames, after the rotation of 60 degrees the scene was very different from the initial one, and only 10% of the image actually contained some visual information. It demonstrates that the algorithm is not only invariant to the scene itself but also to its change during operation.

## 5.3 Robot and tool center point calibration

This section covers the aspects of calibration of robot and tool center point (TCP) which is an important step towards the improvement of positioning accuracy. Actually, to automate the process of image acquisition we need to be capable of rotating around one point which is the center of the object (scene). To do that, the position of this point needs to be known with a very high level of precision.

The system we have is shown in Figure 5.5 and Figure 5.6. Inside a vacuum chamber of SEM, a serial-link manipulator, further referred as robot, is installed. As a usual robotic arm, it comprises a chain of rigid links and joints. One joint $j$ has one degree of freedom represented either as a translational movement for prismatic joints or rotational movement for revolute joints. In our configuration, the robot is PPPRRR or 3P3R, which means that it has three prismatic $(\vec{x}, \vec{y}, \vec{z})$ and three revolute joints. All joints are equipped with position sensors. Prismatic joints are actually three micropositioning stages mounted together. Regarding revolute joints, first two are implemented as a goniometer $(\vec{r}_y, \vec{r}_z)$ and the last one is a classic rotation block $(\vec{r}_x)$. The base of the robot is fixed on the chamber ceiling and the frame associated to it is also fixed and denoted as $\mathcal{R}_0$. Other frames $\mathcal{R}_i$ with $i \in (1, 2, .., 6)$ correspond to six robot joints. The end of the robot which is free to move holds the tool or end-effector which is a sample holder in our case. We will denote $\mathcal{R}_t$ the frame associated with the center of the object we want to rotate about. It is important to note, that we are interested only in its position and not orientation which means that it can be chosen equal to the orientation of last robot joint.

The position of the end-effector clearly depends on the state of each joint. The pose may be computed as a series of transformations involving every link from the base frame to the tool frame:

$$ {}^0\mathbf{T}_t = \underbrace{{}^0\mathbf{T}_1 {}^1\mathbf{T}_2 \cdots {}^{n-1}\mathbf{T}_n}_{\text{forward kinematics}} {}^n\mathbf{T}_t \tag{5.6} $$

where ${}^i\mathbf{T}_k$ denotes the $4 \times 4$ transformation matrix between $\mathcal{R}_k$ to $\mathcal{R}_i$ with $k, i \in (0, 1, .., n)$ and $n = 6$. The first part of this equation is referred to as forward kinematics. In contrast to the tool, that is interchangeable (the position is not the same for different samples), the forward kinematics may be considered constant.

Each transformation matrix in (5.6) includes geometric parameters of the robot: lengths between links, angles between stages, etc. Once the robot has been designed, the theoretical estimation of forward kinematics ${}^0\breve{\mathbf{T}}_t$ (with $\breve{\cdot}$ denoting *theoretical*) is obtained from its CAD model and position of the tool may be calculated. However, due to the process of assembly and manufacturing, these base theoretical parameters are never the true ones which results in low accuracy of object positioning. The situation is even worse at small scales. The presented robot is approximately of 15 cm length while we want to control the position of a 10 µm object. Our task is similar to manipulating the needle with a tower crane!

The accuracy of the robot may be improved by following a calibration procedure with the goal of refinement of robot geometric parameters. In the following sections, we will define these parameters of the robot and present the corresponding calibration procedure.

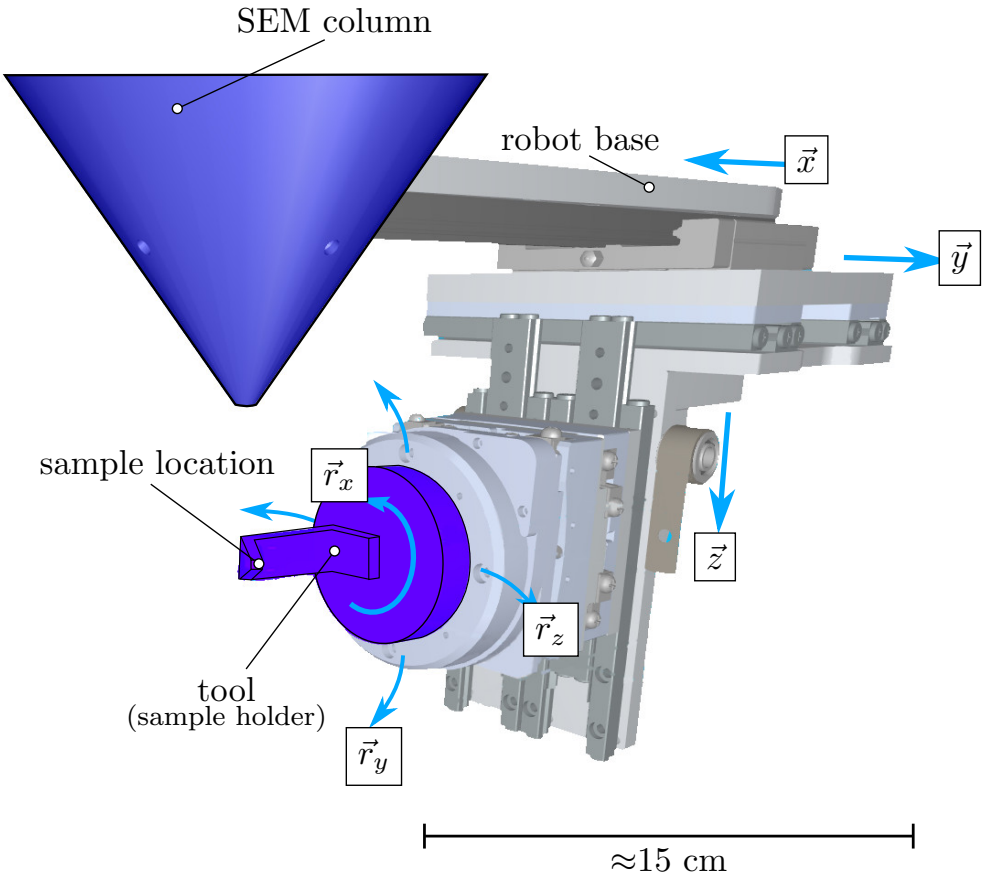



Figure 5.5: View of the CAD models of the 3P3R robot and the SEM column, inside the SEM vacuum chamber.

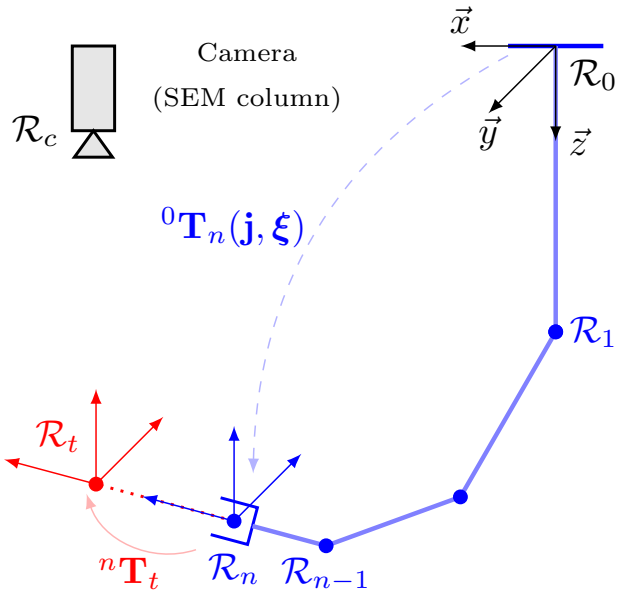



Figure 5.6: Model of the robot installed inside SEM chamber.

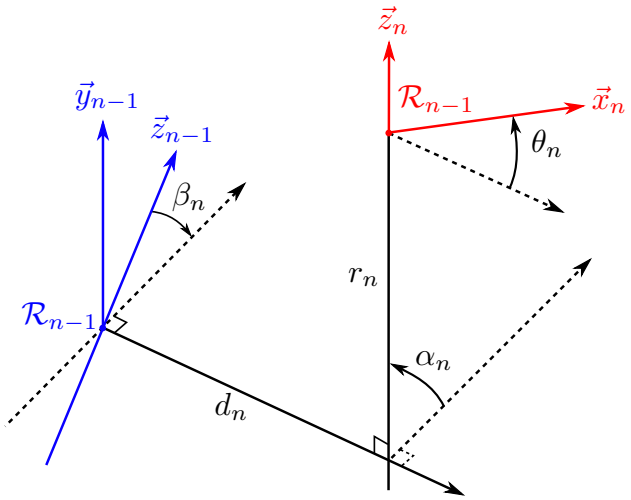



Figure 5.7: Transformation between robot links.

### 5.3.1 Point-link calibration

Calibration procedure implies the refinement of geometric parameters of forward kinematics and tool transformation. The vector of parameters $\boldsymbol{\xi}$ can be decomposed for each transformation separately. For transformations between the links of the robot we will follow Khalil and Kleinfinger notation [KK86]. The transformation matrix between two joints $j_{i-1}$ and $j_i$ can be written as follows (Figure 5.7):

$$^{i-1}\mathbf{T}_i = \text{rot}(\vec{y}, \beta_i) \cdot \text{rot}(\vec{x}, \alpha_i) \cdot \text{trans}(\vec{x}, d_i) \cdot \text{rot}(\vec{z}, \theta_i) \cdot \text{trans}(\vec{z}, r_i) \tag{5.7}$$

And for tool transformation:

$$^{n}\mathbf{T}_t = \text{trans}(\vec{x}, x_t) \cdot \text{trans}(\vec{y}, y_t) \cdot \text{trans}(\vec{z}, z_t) \tag{5.8}$$

The theoretical values of robot forward kinematics are presented in Table 5.1. It should be noted that for revolute joints, angles $\theta_i$ are replaced with joint variables (the values given by sensors). The same is true for $r_i$ and prismatic joints.

Table 5.1: Theoretical model of robot forward kinematics (initial values of parameters $\boldsymbol{\xi}_0$).

| | $\beta_i$ degrees | $\alpha_i$ degrees | $d_i$ mm | $\theta_i$ degrees | $r_i$ mm |
|---|---|---|---|---|---|
| $j_1$ | 0 | 0 | 0 | 0 | $j_1$ |
| $j_2$ | 0 | −90.00 | 0 | 90.00 | $j_2$ |
| $j_3$ | 0 | −90.00 | 0 | 180.00 | $j_3$ |
| $j_4$ | 0 | 0 | 0 | $j_4$ | 0 |
| $j_5$ | 0 | −90.00 | 0 | $j_5$−90.00 | 0 |
| $j_6$ | 0 | −90.00 | 0 | $j_6$ | −41.6 |

The standard calibration procedure comprises the following steps. First, the tool position $\hat{\mathbf{x}}_c$ is measured by a precise external sensor for many different robot configurations $c \in (1, 2, .., N_c)$. The joint variables $\mathbf{j}_c$ are also stored. Therefore, we have

two values that should match each other: tool position measured by an external sensor $\hat{\mathbf{x}}_c$ and the same computed from forward kinematics and tool transformation $\mathbf{x}_c(\mathbf{j}_c, \boldsymbol{\xi})$. The goal is then to find such values of parameters that minimize the difference between them:

$$\boldsymbol{\xi}^* = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \sum_{c=1}^{N_c} \left\| \hat{\mathbf{x}}_c - \mathbf{x}_c(\mathbf{j}_c, \boldsymbol{\xi}) \right\|^2 \tag{5.9}$$

However, as it was already mentioned, it is not possible to find a sensor allowing to provide accurate measurements of tool (object) position at microscale. Therefore, we suggest using another calibration method that relies only on joint parameters and is called *point-link calibration* [KGD95]. The idea behind this method is the following: place the robot in many different configurations and ensure that for every configuration the position of the tool is the same (with an arbitrary orientation). It means that for every pair of configurations, the tool position is the same:

$$\forall (\mathbf{j}_a, \mathbf{j}_b) : \mathbf{x}(\mathbf{j}_a, \boldsymbol{\xi}) = \mathbf{x}(\mathbf{j}_b, \boldsymbol{\xi}) \tag{5.10}$$

Therefore, the calibration problem transforms into:

$$\boldsymbol{\xi}^* = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \sum_{a=1}^{N_{c-1}} \sum_{b=a+1}^{N_c} \left\| \mathbf{x}(\mathbf{j}_a, \boldsymbol{\xi}) - \mathbf{x}(\mathbf{j}_b, \boldsymbol{\xi}) \right\|^2 \tag{5.11}$$

which can be solved by many different optimization algorithms discussed throughout this thesis: from global optimization to Levenberg-Marquardt algorithm, that is the most common approach for such tasks.

As a result, in order to calibrate the robot, we need to acquire a set of configurations with the object at the same position. One can remark that the goal of the chapter is actually to automate exactly this process and this is true. The idea is that we perform one manual acquisition in order to calibrate the robot and, then, when the object is replaced, only the tool transformation would change that can be estimated from a much lower number of poses and automatically (see Section 5.4). The methods allowing to keep the object at the same position while changing the robot configuration are presented below.

### 5.3.2 Maintaining object location

The position of the object center $\mathbf{Q}_t$ (tool) is determined by its three coordinates $(q_{tx}, q_{ty}, q_{tz})^\top$ and the goal is to maintain it in the center of the image for any changing robot configuration. It means that we need to measure the relevant displacement and to compensate it by moving the robot accordingly. The depth coordinate is measured using the autofocus technique presented above (see Section 5.2). For two other coordinates, we use visual servoing to keep the object in the center of the image.

Visual servoing is a technique allowing to control a robot by using the feedback from a visual sensor [CH06]. In our case, the visual sensor is the SEM and we need to control the translation of the object (tool) in the plane parallel to the image plane. In order to measure the displacement of one point, we use the homography-based approach.

Assume the starting point of the algorithm is an in-focus image with the object to track located in its center. This image is used to extract the features that are then

tracked using KLT algorithm in all following images [TK91]. Object center is not a feature itself, thus, in order to track it, one needs to find the transformation between the current image and first image and then find the projection the object center in this current image. This transformation is a homography that defines the following relation between two corresponding points:

$$\mathbf{q}'_t = \mathbf{H}\mathbf{q}_t \tag{5.12}$$

where $\mathbf{H}$ is a 3×3 homography matrix. Therefore, once $\mathbf{H}$ is estimated, the new position of the object center is easily found using (5.12). The estimation of homography is done from tracked features using linear algorithm inside RANSAC scheme [HZ03].

Once the homography is estimated, the error between current object position and the desired one (image center) is written as:

$$\mathbf{e}(t) = \mathbf{q}_t(t) - \mathbf{c} \tag{5.13}$$

where $\mathbf{c}$ is the center of the image and $\mathbf{q}_t(t)$ is a time-varying current position of the tool. It is important to note that the error $\mathbf{e}$ is computed in camera frame $\mathcal{R}_c$ and, in order to express it in the robot base frame to find the joint speed, it has to be multiplied by the transformation matrix ${}^0\mathbf{T}_c$. This matrix is known from CAD model of SEM vacuum chamber. Finally, the final control law for maintaining the object in the center of the image is written as[1]:

$$\dot{\mathbf{j}} = -\lambda {}^0\mathbf{T}_c \mathbf{e} \tag{5.14}$$

where $\dot{\mathbf{j}}$ is the vector of joint speeds and $\lambda$ is a proportional gain.

As a result, by using 2D visual servoing in combination with dynamic autofocus, it is possible to assure that the object always stays in the center of the image.

### 5.3.3 Results

After about 30 robot configurations have been acquired, we use the expression (5.11) to refine the set of parameters $\boldsymbol{\xi}$. The table of calibrated parameters is shown below (Table 5.2). From the resulting values, one can see that the most important correction was done on the parameter $\alpha$ reflecting the perpendicularity between robot axis. The displacements $d$ and $r$ allowed to correct the center of rotation. Actually, before correction, the axis of rotation did not intersect!

With the calibrated robot it is now possible to perform a rotation around the object center (Figure 5.8). As an object, a pollen grain of spherical form was used with the diameter of about 100 µm. After calibration, while rotating, the biggest displacement between the object and the image center is near its size. Such error is mostly due to the depth estimation. Actually, as the object has a spherical form, the autofocus is always performed on its surface, i.e. approximately 50 µm away from the center. Subtracting this value does not seem an ideal option as the surface is not a regular sphere.

[1]Note that Jacobian robot matrix is equal to identity for translational part of presented robot structure

Table 5.2: Calibrated model of robot forward kinematics $\boldsymbol{\xi}^*$.

| | $\beta_i$ degrees | $\alpha_i$ degrees | $d_i$ mm | $\theta_i$ degrees | $r_i$ mm |
|---|---|---|---|---|---|
| $j_1$ | 0 | 0 | 0 | 0 | $j_1$ |
| $j_2$ | 0 | −90.175 | 0 | 90.00 | $j_2$ |
| $j_3$ | 0 | −91.015 | 0 | 180.00 | $j_3$ |
| $j_4$ | -0.310 | 1.163 | 0.000 | $j_4$−0.013 | 0.000 |
| $j_5$ | 0 | −89.790 | −0.337 | $j_5$−89.197 | −0.533 |
| $j_6$ | 0 | −90.102 | 0.956 | $j_6$−0.046 | −41.600 |

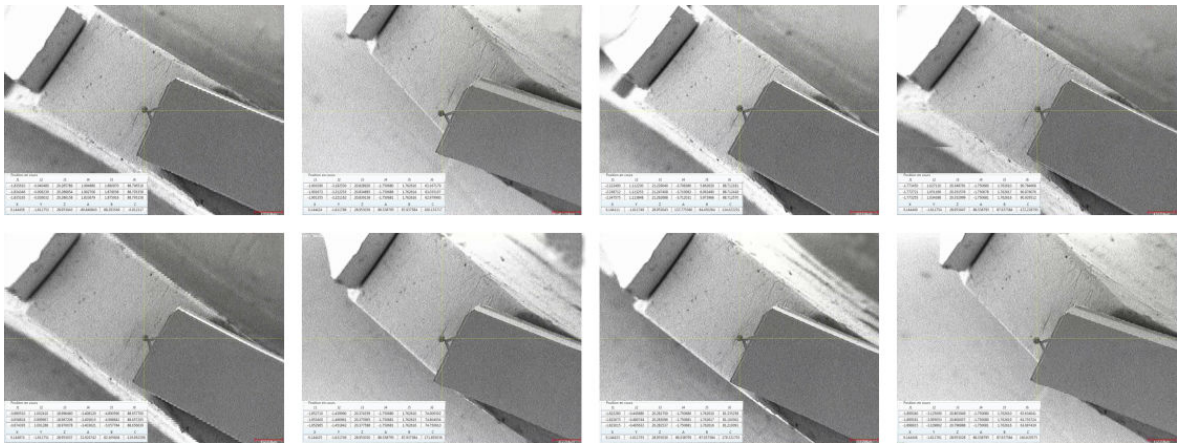


Figure 5.8: Motion of calibrated robot.

## 5.4 Tool center point calibration

In previous section we described the method of robot calibration using *point-link* technique. The robot as well as the tool transformation were calibrated together at the same time. However, the tool is often subject to change as the SEM is used by many people. While the calibrated parameters of the robot may be considered constant, the tool transformation changes with every new sample, and in this section we discuss the method of separate tool center point calibration.

The simplest technique of updating the tool position is the following. If the entire system is calibrated (besides the tool), the transformation matrix between the robot base and the camera (${}^c\mathbf{T}_0$) and forward kinematics (${}^0\mathbf{T}_n$) are known. It means that if we manually place the object in front of the camera and adjust the focus, the tool position can already be calculated from the image center and the current working distance:

$$\mathbf{q}_t = {}^n\mathbf{T}_0 \, {}^0\mathbf{T}_c \begin{pmatrix} 0.5(\text{image width}) \\ 0.5(\text{image height}) \\ \text{working distance} \\ 1 \end{pmatrix} \tag{5.15}$$

This method provides a good initial estimate for further refinement. Assume current TCP is near the desired point, thus, when a single rotation is applied, the desired point

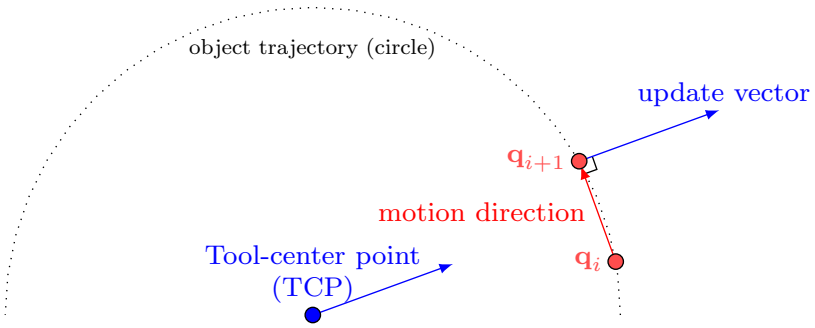



Figure 5.9: Update rule for TCP calibration.

will follow a circle (Figure 5.9), the point moves from $\mathbf{q}_i$ to $\mathbf{q}_{i+1}$. Hence, by updating the tool center point coordinates in the right direction the radius of the circle will decrease. The radius will be equal to zero when the actual TCP coincides with the object center.

In present work, we will use the update rule illustrated in Figure 5.9:

$$\mathbf{TCP}_{i+1} = \mathbf{TCP}_i - \lambda \underbrace{\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} (\mathbf{q}_{i+1} - \mathbf{q}_i)}_{\text{perpendicular to motion direction}} \tag{5.16}$$

Figure 5.10 demonstrates the convergence of the algorithm for different values of $\lambda$. For $\lambda = 5$, the algorithm converged in 21 iterations (21 images).

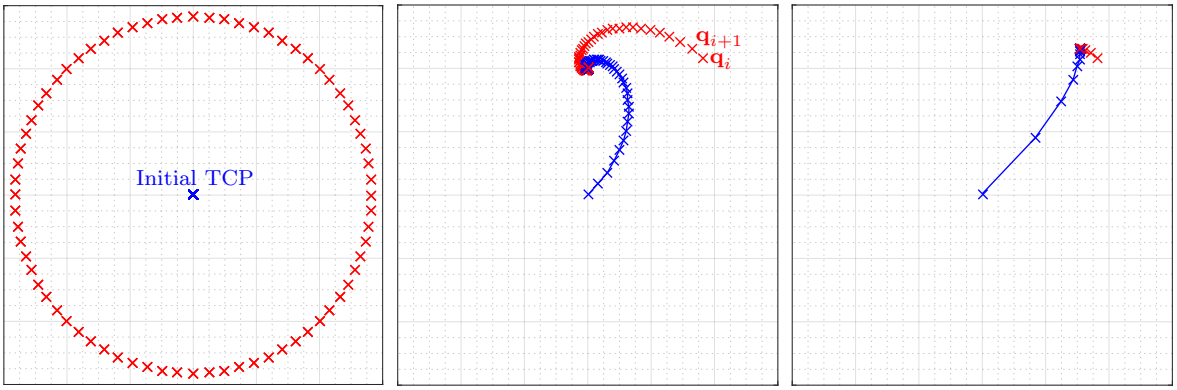



Figure 5.10: TCP calibration for different values of $\lambda$.

## 5.5 Conclusion

In this chapter, we discussed a group of methods dedicated for automatic image acquisition in SEM. The main lock consists in the problem of keeping the object inside the field of view while performing rotations and the presented algorithms are dedicated to solving it.

The first method refers to dynamic autofocus or autofocus on moving object. Sample displacement results in unknown variations of maximal sharpness, i.e. the sharpness function is non-stationary with respect to the change of object position or orientation. The presented method relies on gradient-based optimization and allows keeping the object with unknown structure in focus at high frame rate, i.e. using very noisy images. The experiments on the SEM validated the algorithm and proved its robustness to the variation of magnification and displacement speed. The working distance is adjusted automatically that serves also for depth estimation, for measuring the translation perpendicular to the image plane. With optimal parameters, it was possible to track the object with displacement speed up to 20 µm/s (at ×1000 magnification) and 0.2 degrees/s with a frame rate of 5 Hz and a format of 1024×768 pixels.

Next method demonstrates the procedure allowing to calibrate a serial robot inside SEM that allows performing the rotation around the point of our choice. It starts with the acquisition of different robot configurations keeping the same position of the tool. The vectors of joint variables corresponding to every configuration are then used in the optimization process that allows refining of initial theoretical geometric parameters of the robot and the tool transformation. The method was tested inside SEM with a pollen grain of 100 µm. With applying the rotational speed the object stays in the field of view and the shift from the desired position do not exceed the size of the object.

While the parameters of the robot may be calibrated only once, it is no longer true for the tool that is subject to change frequently. Indeed, in order to change the rotation point, one need to recalibrate the TCP (tool center point) which can be done by the method presented in the previous section. If the TCP is not defined precisely, when a rotational speed is applied, the object will follow the circle with the diameter equal to the distance between the object center and TCP. The proposed method allows making it equal to zero by progressively updating the TCP. The method was presented for in-plane displacements but can be extended to full 3D motion. For example, one may calibrate first for $\vec{x}\vec{y}$ plane and then for $\vec{y}\vec{z}$ plane.

# Chapter 6

# Software development

## Contents



*This chapter briefly presents the software modules that have been developed during this thesis. They integrate all algorithms of 3D reconstruction presented in previous chapters. The software is called Pollen3D. Currently, it exists in the form of MATLAB toolbox and a standalone C++ application with GUI. In this chapter, we will speak mostly about C++ implementation, however, it is important to note that the same results may be achieved by using the Pollen3D Toolbox for MATLAB.*

## 6.1 Context

The software developed as a part of this thesis is called Pollen3D with its logo displayed in Figure 6.1. The goal of this software is to allow users of SEM, that may have very different scientific background, to obtain a dense 3D reconstruction of a small object from a sequence of its images.


Figure 6.1: Logo of the Pollen3D software for 3D reconstruction in SEM.

Two implementations of Pollen3D exist nowadays. First, it is a MATLAB toolbox that contains only a set of functions, and, secondly, it is a standalone C++ application with a graphical user interface. Both implement all algorithms presented in previous chapters. The input is a sequence of SEM images obtained by moving the robotic stage. The standalone application was developed in C++ and is cross-platform (it was tested on Windows, Linux (Ununtu distribution) and Mac), that became possible by using *Qt 5* library for GUI. The core part of the software uses two external libraries (see Figure 6.2). *OpenCV* [Bra00] library that is used for interaction with image files, feature extraction and matching, and mathematical algorithms such as SVD, matrix multiplication, etc. Another library used is *NLopt* which stands for non linear optimization [Joh14]. Thanks to it, we could include the global optimization in the software.

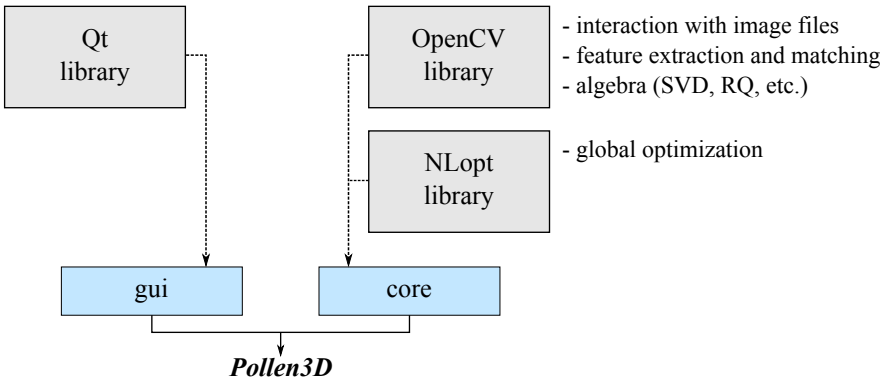



Figure 6.2: External libraries for Pollen3D software.

In the following sections, we briefly discuss different elements of Pollen3D software and the links between program and previous chapters.

## 6.2 Pollen3D software GUI

At the start of the program, user should load the images by clicking on the corresponding button. A screen-capture of the software window with images loaded is displayed in Figure 6.3. It contains the following elements: the list of images and an image viewer displaying them. At the right, there is a console displaying the log messages. The main element of control is the tab widget located on top. Four tabs are available. General tab allows some basic manipulations with images and the project itself such as save, load etc. Other tabs actually correspond to the target they are applied to:

- Image. It contains the algorithms that can be applied to one image only.
- Stereo. This tab works with image pairs.
- Multiview. This tab controls the algorithms dedicated for image sequences and point clouds.

The left side and right side widgets can be hidden by clicking on the arrow buttons.

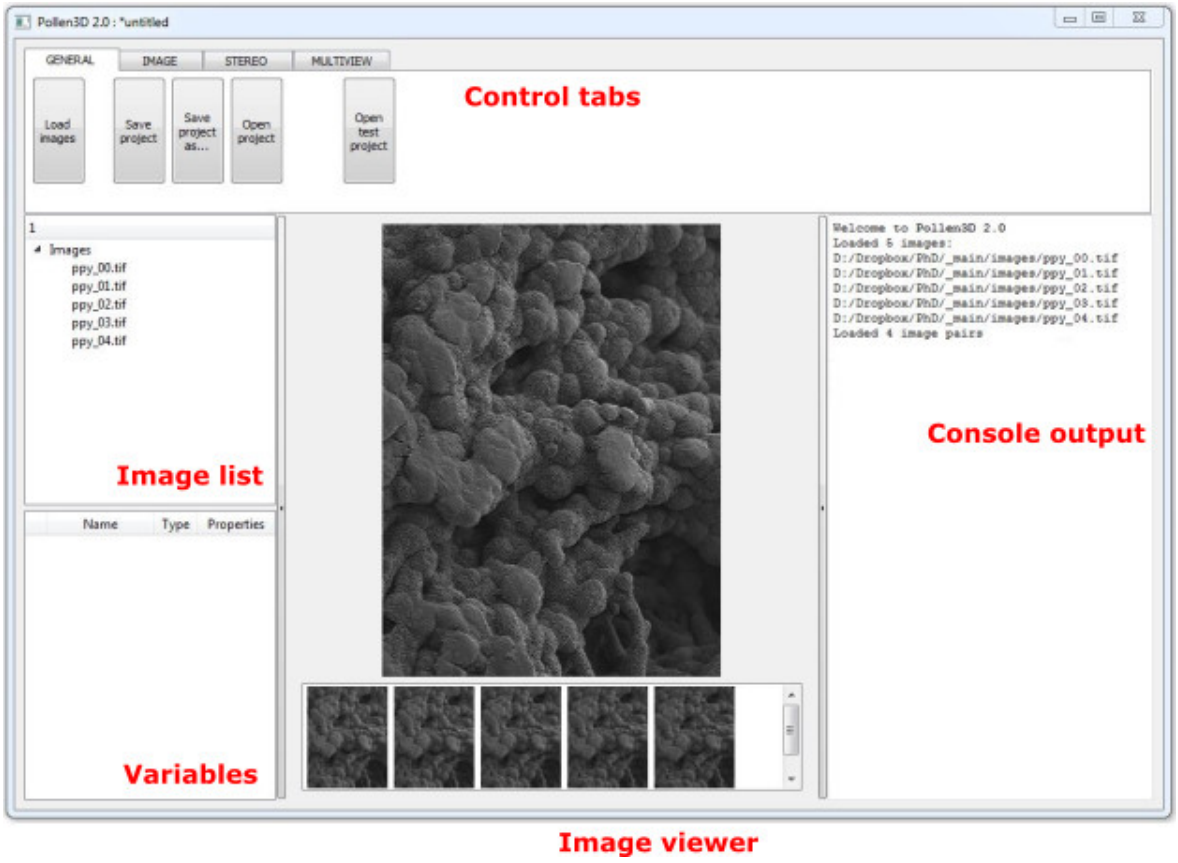



Figure 6.3: Main window of Pollen3D software with images loaded (**ppy** dataset).

### 6.2.1 Image tab

On this tab, we work with each image separately and the available function is the detection of features using the algorithms of Section 2.1. User needs to tune the threshold parameter that reflects the feature quality (see Figure 6.4). For low-textured images this parameter should be decreased. The parameter can be specified for the whole sequence and then adjusted for every image separately if needed.

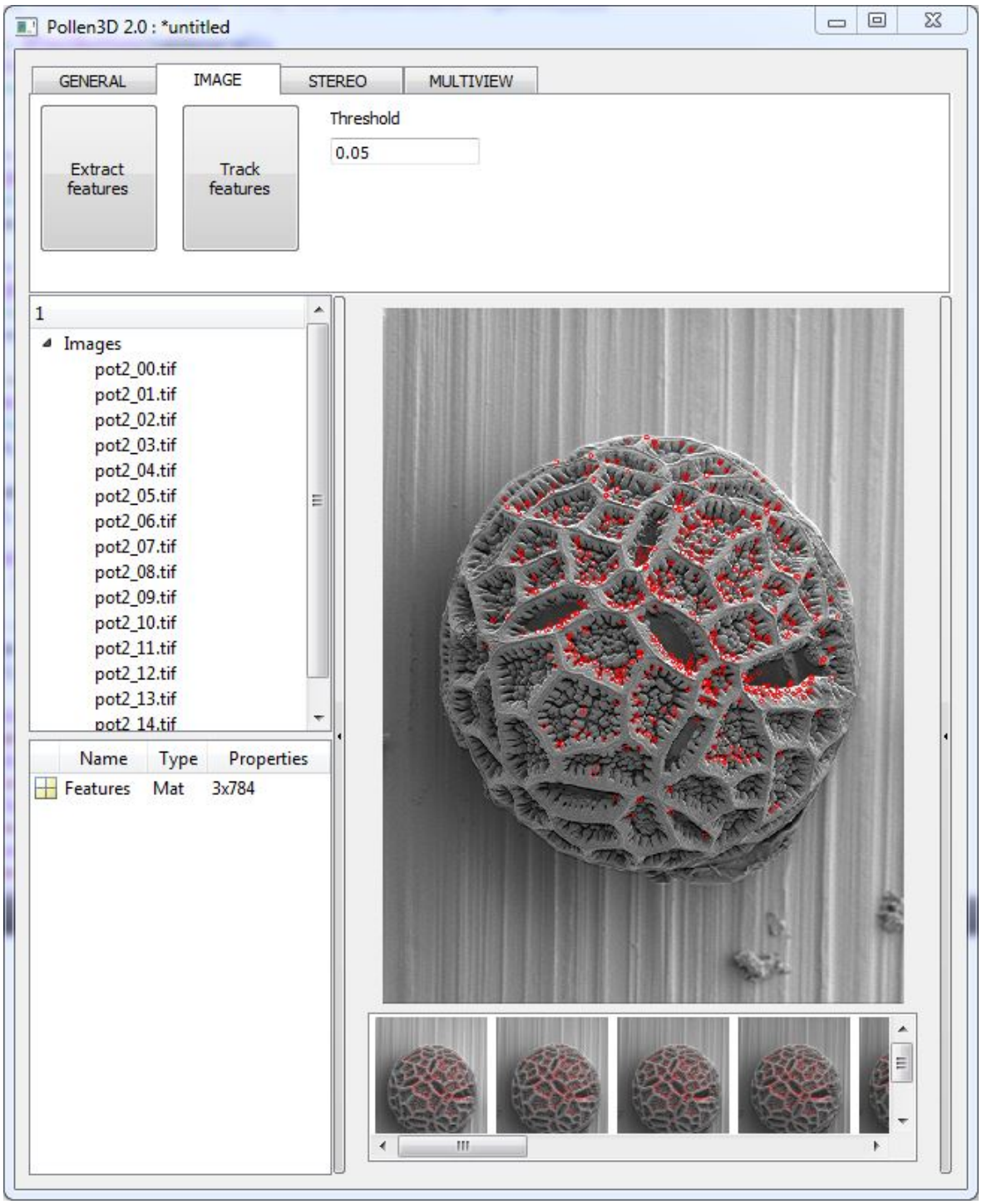



Figure 6.4: Feature extraction using Pollen3D (**pot2** dataset).

### 6.2.2 Stereo tab

This tab allows user to work with image pairs and the image list widget changes accordingly (see Figure 6.5). Every task represents a separate widget surrounded by a box. As we can see from the figure, it is possible to perform pairwise matching and estimate the fundamental matrix. All algorithms of fundamental matrix estimation presented in this thesis (Section 2.6) are available from the dropdown list. The corresponding check boxes control the display of epipolar lines of matches which can be very useful as one can immediately detect possible malfunctions: wrong filter coefficient or presence of outliers.

With the controls on this tab, a rectification algorithm presented in Section 4.2 can be applied to images. The result is shown in Figure 6.5.

The last available function consists in the disparity map computation using dense matching algorithms (Section 4.3). All dense matching parameters such as disparity range or block size can be modified directly in GUI.

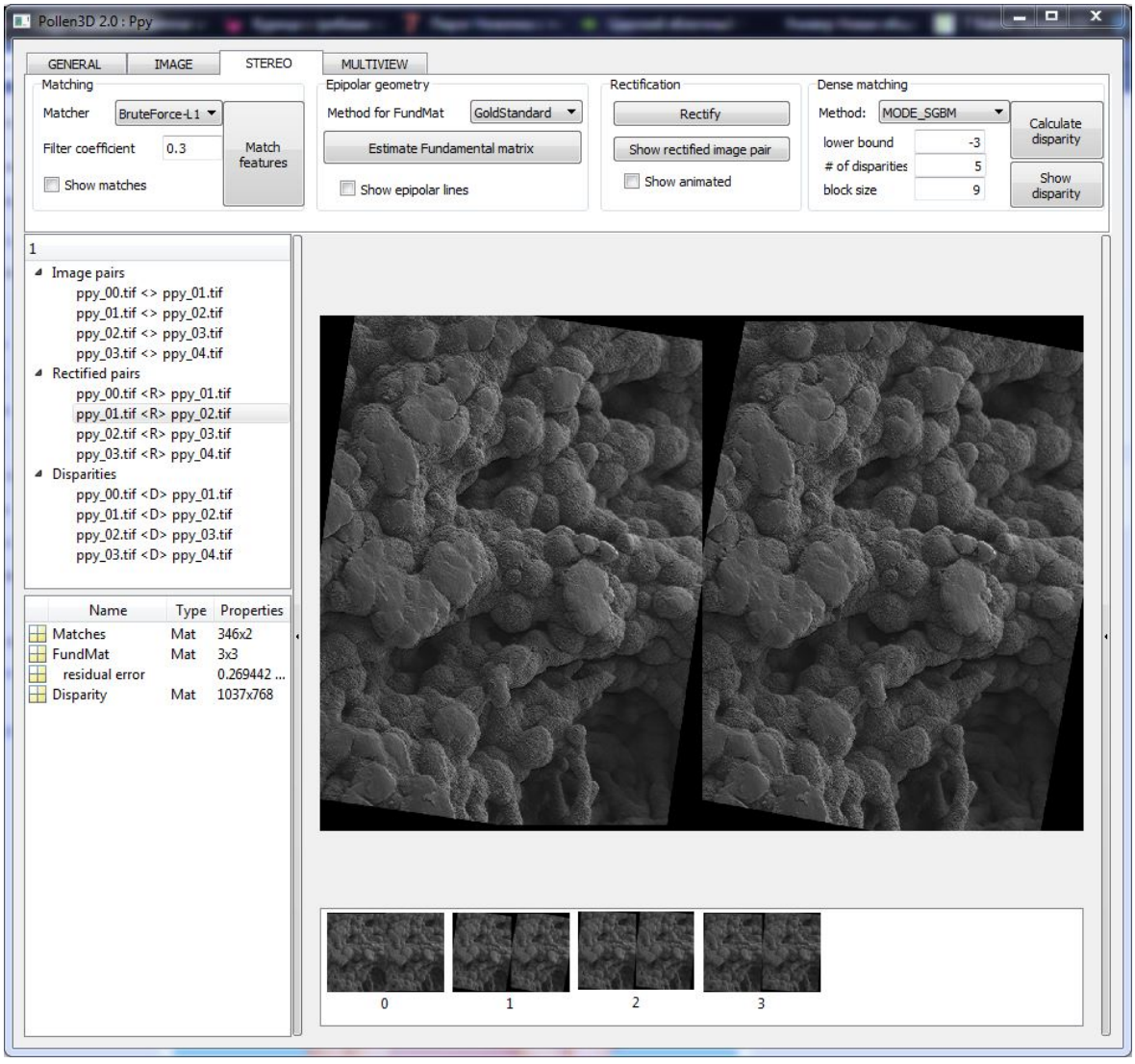



Figure 6.5: Rectification using Pollen3D (**ppy** dataset).

### 6.2.3 Multiview tab

The last control tab is the "Multiview" tab (Figure 6.6). Here, the algorithms working with multiple views are presented. First, it is possible to extract the measurement matrix $\mathcal{W}$ and full measurement matrix $\mathcal{W}_f$. Secondly, the optimization widget refers to autocalibration using global optimization (Chapter 3) that is achieved by clicking on "Optimize" button. Besides general stopping criteria which is the tolerance on the function change, it is also possible to stop the optimization after $t$ seconds. At the end, the results may be refined using local optimization.

The last control group refers to point cloud. 3D points are obtained from disparity maps and camera matrices using the triangulation techniques presented in 4.4. Finally, the point cloud may be saved to a *.ply file (Polygon File Format) that is a free data storage format that can be read by MeshLab or other 3D modelling software.

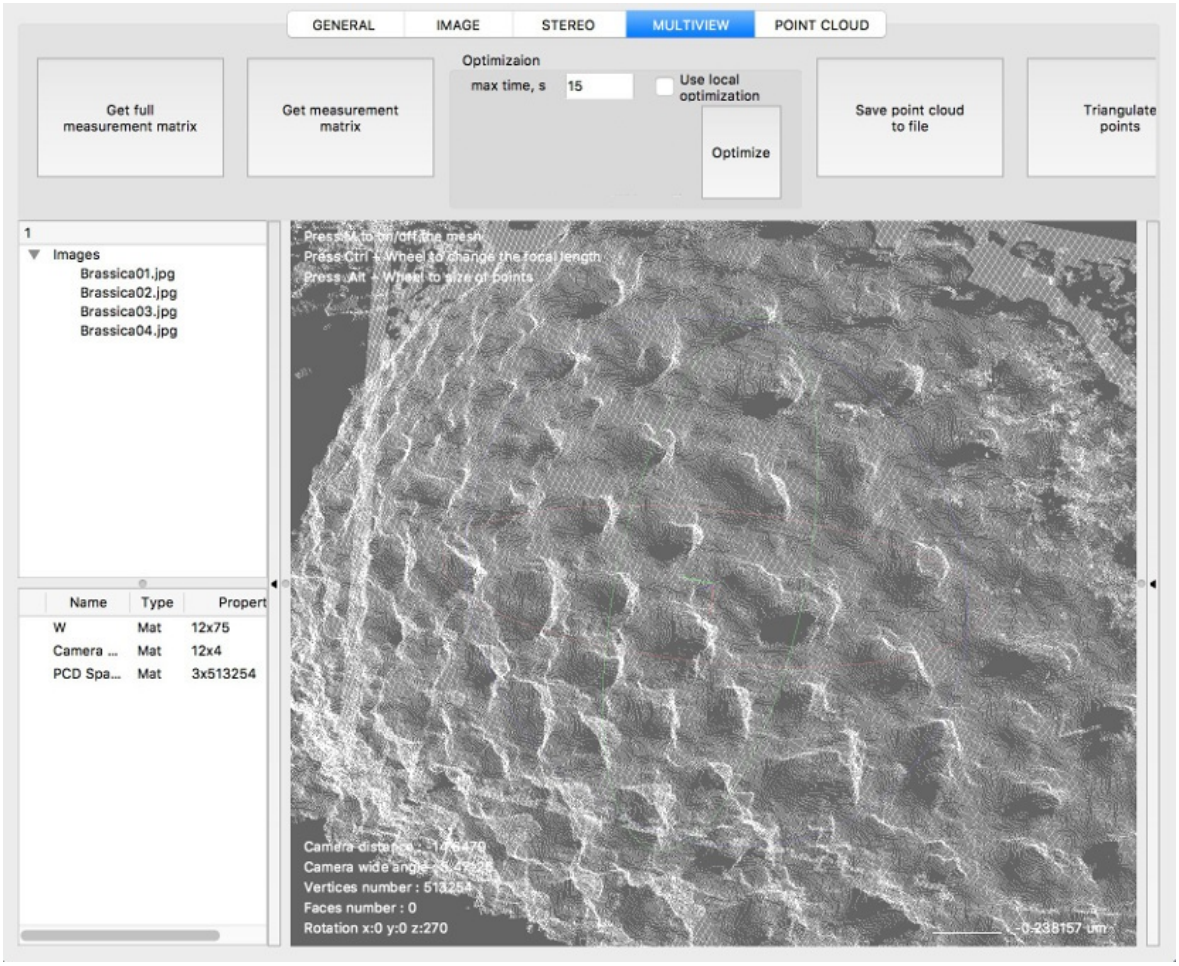



Figure 6.6: Dense 3D reconstruction using Pollen3D (**brassica** dataset).

## 6.3 Conclusion

This chapter provides only a very brief overview of the software developed in the scope of this thesis. It represents a standalone cross-platform application that allows to obtain a 3D reconstruction from multiple SEM images using the algorithms presented in this work. As an example, it is worth to mention that for four images of **brassica** sequences, approximately 400,000 3D points were obtained in less than a minute which is a good result considering that the algorithms were not optimized for speed. Therefore, their efficiency can be improved by means of parallel computing and/or GPU techniques.

Our C++ software, Pollen3D, provides all necessary controls for every step of 3D reconstruction: extraction of features, matching, estimation of fundamental matrix, rectification, dense matching, autocalibration, and triangulation. At every moment, the obtained data can be saved to a project file or exported as MATLAB data file. Its viability and performance were validated through multiple tests. The same is also true for Pollen3D MATLAB toolbox.

By lack of time, the work on the software is still in progress and we hope that the first version of it will be available online by the end of 2017 at least for MATLAB toolbox. The choice of license has not been done yet. At present, we are planning to install Pollen3D on the SEM computer to get some feedback from other researches of the team in order to make it more user friendly and to minimize the number of program failures.

# Conclusion and perspectives

## 6.4 Summary and discussion

3D reconstruction is a very powerful technique that allows obtaining a 3D model of an object from its images taken from different perspectives. 3D reconstruction is a very deeply studied subject in computer vision, however, its application to special visual devices, such as Scanning Electron Microscope, creates new challenges that have not been completely solved. The microworld, while opening new possibilities for many research fields, remains a continuing challenge. With regard to SEM, the main lock is due to specifics of image formation when working with very small objects. It is reflected by the fact that, due to the very high ratio between the distance from the camera to object and the object size, the projection rays are parallel. This feature has to be taken into account as, otherwise, 3D reconstruction is not possible.

In the introductory chapter, different solutions proposed in the state-of-the-art are discussed: while an important steps towards the problem resolution were made, some problems stayed unresolved. Among them were the camera calibration, density of 3D reconstruction and the process of image acquisition, and we believe that, in this thesis, some advances in these fields were made.

Consider first the aspect of calibration. Previously, in order to calibrate a SEM, one needed a special calibration target or grid fabricated with a very high precision. The latter is not a trivial undertaking as the fabrication precision at micrometer scale can not be compared to general macroscale results. In this work, we discussed the technique of autocalibration that was adapted for and applied to SEM for the first time. It means that 3D reconstruction can be achieved without using additional sensors or any information about the object as the algorithm computes at the same time the intrinsic and extrinsic camera parameters. Therefore, assuming that the internal parameters of the camera do not change across the image sequence, there is no more need for calibration target.

It should be mentioned that the assumption of parallel projection was confirmed for the magnification range starting form $\times 1000$. It is within this range that our method of autocalibration finds its application. For lower magnifications, the perspective effects cannot be neglected. Moreover, an important work has to be done on the distortion compensation. Another limitation of the algorithm consists in the quality of features detected in the images. For such biological samples as pollen grains, this problem is not present. However, the surfaces of the objects fabricated using such methods as photolithography often lack of features. One of the ways to anticipate this situation may be to add some random pattern on the object surface during fabrication.

Indeed, image quality has a huge impact on the quality of 3D reconstruction. There-

fore, to acquire a sequence of images for 3D reconstruction we suggest to consider the following points. First, it is important to properly adjust SEM parameters in order to reduce noise and the charging effects. For example, the accelerating voltage for biological, mostly non-conductive, samples should be lowered. Regarding the angle of out-of-plane rotation, experiments have shown that the best results are given for the angle of approximately 3 degrees, as the feature matching is quite simple. However, we suggest taking at least 5 images so the total baseline for images in the sequence is equal to at least 15 degrees.

Once the calibration is done, the intrinsic parameters of the camera as well as its location for every image are available. It brings us to the next step covered in this work which is dense 3D reconstruction. With the presented techniques it is now possible to obtain a point cloud that may contain millions of points. This result creates new possibilities in terms of micro- and nanocharacterization. Point cloud is the first level of object model where the measurements can be done. Consider the example of **cutting** dataset (see Figure 4.11 for 3D reconstruction). Our colleagues from the department of Applied Mechanics (FEMTO-ST Institute) were interested in measuring the angle between two planes of the edge of this cutting tool. The angle can be found by fitting two planes to the corresponding parts of 3D reconstruction. Our result is 79.3 degrees while the angle in the CAD model is equal to 80 degrees.

The last part of the work on 3D reconstruction covered in this thesis concerns image acquisition. When working with SEM, in order to acquire several images of the object, even an experienced operator may spend hours of work. The main problem is the impossibility of rotating the object around its center. As the center of rotation is away, even after a small movement of the robot, the object leaves the field of view. To tackle this lock, we proposed three new methods. The first one allows keeping the object in focus while it is moving. Two other methods concern the calibration of the robot and the tool center point. Both represent a great interest not only for 3D reconstruction but also for small scale robotics, microassembly etc. The results are promising but a deeper experimental investigation is needed to improve the accuracy of calibration. Both methods were validated on a pollen grain, however, for further development, we suggest using a smaller object (10 μm or less). Ideally, it would be a sphere of a very small diameter mounted on a tip.

It is worth mentioning that the method of calibration presented here does not take into account non-geometric parameters of the robot such as compliance, encoder misalignment or joint deformation. According to the results presented in the literature, non-geometric robot properties are responsible for 8%-10% of the position error of the end effector [RRB+91]. Obviously, their impact may be different for microscale robots and has to be studied in more details in the future.

Finally, we would like to point out that this work covers all the steps of 3D reconstruction in Scanning Electron Microscope: from image acquisition to dense point cloud. We also successfully combined all methods in one standalone software called Pollen3D and we hope that its final version will be available online by the end of 2017.

## 6.5 Contributions

This section summarize briefly the main scientific contributions presented in this dissertation following its structure.

- Chapter 2 presents a new method of full *motion estimation* from three images taken with an affine camera. The method is entirely based on the epipolar geometry and spherical trigonometry. In contrast to the state-of-the-art methods that use additional sensors such as focus or robot sensors, in our case, the motion is computed from images only.

- In Chapter 3 we derived a new method of *autocalibration* for SEM considering it to be an affine camera. Presented algorithm allows to achieve directly the Euclidean reconstruction without an additional affine step. The autocalibration is formulated as an optimization problem and by adding a supplementary regularization term we ensure the respect of metric constraints. Being based on global optimization, the method has a high convergence range. Moreover, all optimized parameters have an actual physical meaning so that user can easily impose new constraints if some of the information is available (known rotation angles, constant parameters etc.).

- In Chapter 4 a *rectification* algorithm for SEM images was derived in accordance with previous step of autocalibration. All parameters used for rectification are estimated directly from the fundamental matrix and then refined at the step of autocalibration. The accuracy is also improved by implementing robust estimation of fundamental matrix using MLESAC method.

- In Chapter 5 we made an important step towards *automation of image acquisition* in SEM. First, we presented a method of autofocus for moving object that has never been done for such type of visual sensors. The best focus is achieved fast without sweeping the working distance.

- At last, we proposed a *calibration method for complex robotic structure* inside SEM using only visual information (Chapter 5). It allows improving the accuracy of robot positioning. Moreover, thanks to tool center point calibration, it is now possible to change the rotation point to the center of image. The algorithms were validated on one object (pollen grain) and need further, mostly experimental, investigations.

## 6.6 Future work

The following points may be consider in the future as the next steps for this dissertation:

**Point cloud to 3D model.** Even if the point cloud can be exploited for characterization, it still has a limited application. It would be very interesting to work on the methods allowing to filter the point cloud, reduce the number of points without loss of quality, transform it into a mesh. Ideally, the final step of this process would be the creation of printable 3D models for visualization and educational purposes.

**Multimodal 3D reconstruction.** In present work, we used only one SE detector to obtain several images. However, if the SEM is equipped with another detector such as BSE, it is possible to combine the reconstructions from different detectors and obtain not only the topological information but also the information about the material of the sample. One may also consider joining the structure-from-motion technique with photommetric stereo in case of multiple detectors.

**Robot calibration through 3D reconstruction.** The method of robot calibration presented in Chapter 5 is based on the principle of maintaining the object at the same place while changing the robot configuration. However, by means of 3D reconstruction we could actually find the relative object displacement between two images and the 3D points, and use them inside the robot calibration algorithm. Assume that $N_c$ configurations were acquired and $N_{pts}$ 3D points were reconstructed, the following optimization problem may be defined:

$$\boldsymbol{\xi}^* = \underset{\boldsymbol{\xi}}{\operatorname{argmin}} \sum_{i}^{N_{pts}} \sum_{c}^{N_c} \left\| \mathbf{q}_i^c \times \mathbf{K}/\!\!/\boldsymbol{\Pi}/\!\!/\mathbf{T}(\mathbf{j}_c, \boldsymbol{\xi})\mathbf{Q}_i \right\|^2 \tag{6.1}$$

with $\mathbf{T}(\mathbf{j}, \boldsymbol{\xi}) = {}^c\mathbf{T}_0 {}^0\mathbf{T}_n {}^n\mathbf{T}_t$. It means that from just a sequence of images it would be possible to calibrate the camera and the robot holding the sample at the same time.

**Software testing**. Even though the developed software application (Pollen3D) is running well in most of the cases, it has not been tested thoroughly. Also due to the time constraint, some basic but very useful functionality such as undoing the previous command, adding one more image to the sequence, etc. was not added yet. So, for the future work, it is highly recommended to perform an overall unit testing. Once this task is finished, it is planned to distribute the software first within the department for more tests and then, freely, on one of the open-source platforms.

**Software modules**. Regarding microcharacterization, the needs of every research department are different. It may vary from the estimation of angles between two planes (as in example of cutting tool) to the comparison of two volumes. Obviously, the software cannot meet all demands. Therefore, it would be interesting to integrate a system of modules or plugins so that every user could write a small script in order to tune the application to its own needs.

# Bibliography


[Abb73] E. Abbe. "Beiträge zur Theorie des Mikroskops und der mikroskopischen Wahrnehmung". In: *Archiv für mikroskopische Anatomie* 9.1 (1873), pp. 413–418.

[ABD12] P. Alcantarilla, A. Bartoli, and A. Davison. "KAZE features". In: *European Conf. Computer Vision (ECCV'12)*. Springer, 2012, pp. 214–227.

[ANB11] P. F. Alcantarilla, J. Nuevo, and A. Bartoli. "Fast explicit diffusion for accelerated features in nonlinear scale spaces". In: *IEEE Trans. Patt. Anal. Mach. Intell* 34.7 (2011), pp. 1281–1298.

[ASP+14] J.-O. Abrahamians, B. Sauvet, J. Polesel-Maris, R. Braive, and S. Régnier. "A nanorobotic system for in situ stiffness measurements on membranes". In: *IEEE Trans. Robot.* 30.1 (2014), pp. 119–124.

[BC91] W. Beil and I. Carlsen. "Surface reconstruction from stereoscopy and "shape from shading" in SEM images". In: *Mach. Vis. Appl.* 4.4 (1991), pp. 271–285.

[BDN+06] B. J. Bell, L. Dong, B. Nelson, M. Golling, L. Zhang, and D. Grützmacher. "Fabrication and characterization of three-dimensional InGaAs/GaAs nanosprings". In: *Nanoletters* 6.4 (2006), pp. 725–729.

[BET+08] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. "Speeded-up robust features (SURF)". In: *Comput. Vis. Image Underst.* 110.3 (2008), pp. 346–359.

[BM58] G. E. Box and M. E. Muller. "A note on the generation of random normal deviates". In: *Ann. of Math. Stat.* 29.2 (1958), pp. 610–611.

[Bra00] G. Bradski. "OpenCV library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[BTO+17] A. Baghaie, A. P. Tafti, H. A. Owen, R. M. D'Souza, and Z. Yu. "SD-SEM: sparse-dense correspondence for 3D reconstruction of microscopic samples". In: *Micron* 97 (2017), pp. 41–55.

[CAK+07] M. Chandraker, S. Agarwal, F. Kahl, D. Nistér, and D. Kriegman. "Autocalibration via rank-constrained estimation of the absolute quadric". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR'07)*. 2007, pp. 1–8.

[CCC+08] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. "MeshLab: an Open-Source Mesh Processing Tool". In: *Eurographics Italian Chapter Conf.* 2008.

[CGS+03] N. Cornille, D. Garcia, M. A. Sutton, S. McNeill, and J.-J. Orteu. "Automated 3-D reconstruction using a scanning electron microscope". In: *SEM Conf. Exp. and Appl. Mech.* 2003.

[CH06] F. Chaumette and S. Hutchinson. "Visual servo control, Part I: Basic approaches". In: *IEEE Robot. Automat. Mag.* 13.4 (2006), pp. 82–90.

[CKY09] S. Choi, T. Kim, and W. Yu. "Performance evaluation of RANSAC family". In: *British Machine Vision Conf. (BMVC'09)* (2009), pp. 1–12.

[CM14] L. Cui and E. Marchand. "Calibration of scanning electron microscope using a multi-image non-linear minimization process". In: *IEEE Int. Conf. Robot. Autom. (ICRA'14)*. 2014, pp. 5191–5196.

[CMH+16] L. Cui, E. Marchand, S. Haliyo, and S. Régnier. "Three-Dimensional Visual Tracking and Pose Estimation in Scanning Electron Microscopes". In: *IEEE Int. Conf. on Intell. Robots and Syst. (IROS'16)* (2016).

[Cor05] N. Cornille. "Accurate 3D shape and displacement measurement using a scanning electron microscope". PhD thesis. INSA de Toulouse, 2005.

[DHS11] J. Duchi, E. Hazan, and Y. Singer. "Adaptive subgradient methods for online learning and stochastic optimization". In: *J. Mach. Learn. Res.* 12 (2011), pp. 2121–2159.

[DLH10] Y. Dai, H. Li, and M. He. "Element-wise factorization for n-view projective reconstruction". In: *European Conf. Computer Vision (ECCV'10)*. 2010, pp. 396–409.

[DSH+12] R. Danzl, H. Schroettner, F. Helmli, and S. Scherer. "Coordinate measurement with nano-metric resolution from multiple SEM images". In: *European Microscopy Congr.* 2012.

[DZK+09] L. Dong, L. Zhang, B. E. Kratochvil, K. Shou, and B. J. Nelson. "Dual-chirality helical nanobelts: linear-to-rotary motion converters for three dimensional microscopy". In: *IEEE J. Microelectromech. Syst.* 18.5 (2009), pp. 1047–1053.

[EFG+15] B. Escolle, M. Fontaine, A. Gilbin, S. Thibaud, and P. Picart. "Experimental investigation in micro ball-end milling of hardened steel". In: *J. Mater. Sci. Eng. A* 5 (2015), pp. 327–338.

[FBF+04] A. Fusiello, A. Benedetti, M. Farenzena, and A. Busti. "Globally convergent autocalibration using interval analysis". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (2004), pp. 1633–1638.

[FDM15] G. Facciolo, C. De Franchis, and E. Meinhardt. "Mgm: A significantly more global matching for stereovision". In: *British Machine Vision Conf. (BMVC'15)*. 2015.

[FLM92] O. D. Faugeras, Q.-T. Luong, and S. J. Maybank. "Camera self-calibration: Theory and experiments". In: *European Conf. Computer Vision (ECCV'92)*. 1992, pp. 321–334.

[FMM+15] E. Faber, D. Martinez-Martinez, C. Mansilla, V. Ocelík, and J. T. M. De Hosson. "Calibration-free quantitative surface topography reconstruction in scanning electron microscopy". In: *Ultramicroscopy* 148 (2015), pp. 31–41.

[FTV00] A. Fusiello, E. Trucco, and A. Verri. "A compact algorithm for rectification of stereo pairs". In: *Machine Vision and Appl.* 12 (2000), pp. 16–22.

[FWH+07] S. Fatikow, T. Wich, H. Hulsen, T. Sievers, and M. Jahnisch. "Microrobot system for automatic nanohandling inside a scanning electron microscope". In: *IEEE/ASME Trans. Mechatronics* 12 (2007), pp. 244–252.

[Glo98] F. Glover. "A template for scatter search and path relinking". In: *Lecture notes in computer science* 1363 (1998), pp. 13–54.

[GNE+12] J. Goldstein, D. E. Newbury, P. Echlin, D. C. Joy, A. D. Romig Jr, C. E. Lyman, C. Fiori, and E. Lifshin. *Scanning electron microscopy and X-ray microanalysis: a text for biologists, materials scientists, and geologists*. Springer Science & Business Media, 2012.

[Gol89] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, 1989.

[GTH+16] T. Gissibl, S. Thiele, A. Herkommer, and H. Giessen. "Two-photon direct laser writing of ultracompact multi-lens objectives". In: *Nat. Photonics* 10 (2016), pp. 554–560.

[HAA16] M. Hassaballah, A. A. Abdelmgeid, and H. A. Alshazly. "Image Features Detection, Description and Matching". In: *Image Feature Detectors and Descriptors*. 2016, pp. 11–45.

[Had02] J. Hadamard. "Sur les problèmes aux dérivées partielles et leur signification physique". In: *Princeton University Bulletin* (1902), pp. 49–52.

[Har94] R. I. Hartley. "Projective reconstruction and invariants from multiple images". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 16 (1994), pp. 1036–1041.

[HDD+04] Z. Huang, D. Dikin, W. Ding, Y. Qiao, X. Chen, Y. Fridman, and R. Ruoff. "Three-dimensional representation of curved nanowires". In: *J. Microsc.* 216 (2004), pp. 206–214.

[Hir05] H. Hirschmuller. "Accurate and efficient stereo processing by semi-global matching and mutual information". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 2. 2005, pp. 807–814.

[HK12] S. Hermann and R. Klette. "Iterative semi-global matching for robust driver assistance systems". In: *Asian Conf. on Computer Vision*. 2012, pp. 465–478.

[HMG+14] J. Henao, J. Meunier, J. Gomez-Mendoza, and J. Riãno-Rojas. "SEM Surface Reconstruction using optical flow and Stereo Vision". In: *Int. Conf. on Image Processing, Computer Vision, and Pattern Recognition (IPCV'14)*. 2014, p. 1.

[Hor70] B. K. Horn. *Shape from shading: A method for obtaining the shape of a smooth opaque object from one view*. 1970.

[HS88] C. Harris and M. Stephens. "A combined corner and edge detector". In: *Alvey Vision Conf.* Vol. 15. 50. 1988, pp. 10–5244.

[HS97] R. I. Hartley and P. Sturm. "Triangulation". In: *Computer vision and Image Understanding* 68 (1997), pp. 146–157.

[HSF11] S. B. Heinrich, W. E. Snyder, and J.-M. Frahm. "Maximum likelihood autocalibration". In: *Image and Vision Computing* 29 (2011), pp. 653–665.

[Hub05] P. J. Huber. *Robust statistics*. Vol. 579. John Wiley & Sons, 2005.

[HZ03] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[HZH03] J. He, R. Zhou, and Z. Hong. "Modified fast climbing search autofocus algorithm with adaptive step size searching technique for digital camera". In: *IEEE Trans. Consum. Electron.* 12 (2003), pp. 244–252.

[JF07] M. Jähnisch and S. Fatikow. "3D vision feedback for nanohandling monitoring in a scanning electron microscope". In: *Int. J. Optomech.* 1 (2007), pp. 4–26.

[Joh14] S. G. Johnson. *The NLopt nonlinear-optimization package*. 2014.

[KAS+10] K. H. Kim, Z. Akase, T. Suzuki, and D. Shindo. "Charging effects on SEM/SIM contrast of metal/insulator system in various metallic coating conditions". In: *Mat. Trans.* 51 (2010), pp. 1080–1083.

[KB14] D. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[KDN09] B. E. Kratochvil, L. Dong, and B. J. Nelson. "Real-time rigid-body visual tracking in a scanning electron microscope". In: *Int. J. Robot. Res.* 28 (2009), pp. 498–511.

[KDP17] A. V. Kudryavtsev, S. Dembélé, and N. Piat. "Stereo-image rectification for dense 3D reconstruction in SEM". In: *Int. Conf. on Manipulation, Autom. and Robot. at Small Scales (MARSS'17)*. 2017, pp. 42–47.

[KDZ+10] B. Kratochvil, L. Dong, L. Zhang, and B. Nelson. "Image-based 3D reconstruction using helical nanobelts for localized rotations". In: *J. Microsc.* 237 (2010), pp. 122–135.

[KGD95] W. Khalil, G. Garcia, and J.-F. Delagarde. "Calibration of the geometric parameters of robots without external sensors". In: *IEEE Int. Conf. Robot. Autom. (ICRA'95)*. Vol. 3. 1995, pp. 3039–3044.

[Kie53] J. Kiefer. "Sequential minimax search for a maximum". In: *Proc. of American Math. Soc.* 4 (1953), pp. 502–506.

[KK86] W. Khalil and J. Kleinfinger. "A new geometric notation for open and closed-loop robots". In: *IEEE Int. Conf. Robot. Autom. (ICRA'86)*. Vol. 3. 1986, pp. 1174–1179.

[KMP+10] S. Kumar, C. Micheloni, C. Piciarelli, and G. L. Foresti. "Stereo rectification of uncalibrated and heterogeneous images". In: *Pattern Recogn. Lett.* 31 (2010), pp. 1445–1452.

[Kon98] K. Konolige. "Small vision systems: Hardware and implementation". In: *Robotics research*. 1998, pp. 203–212.

[KR08] P. J. Kostelec and D. N. Rockmore. "FFTs on the rotation group". In: *J. Fourier Anal. Appl.* 14 (2008), pp. 145–179.

[KV91] J. J. Koenderink and A. J. Van Doorn. "Affine structure from motion". In: *JOSA A* 8 (1991), pp. 377–385.

[LF96] Q.-T. Luong and O. D. Faugeras. "The fundamental matrix: Theory, algorithms, and stability analysis". In: *Int. J. of Computer Vision* 17 (1996), pp. 43–75.

[Lin10] P. Lindstrom. "Triangulation made easy". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR'10)*. 2010, pp. 1554–1561.

[LJD09] S. Liansheng, Z. Jiulong, and C. Duwu. "Image rectification using affine epipolar geometric constraint". In: *J. of Software* 4 (2009), p. 27.

[LLX+13] C. Li, Z. Liu, H. Xie, and D. Wu. "Novel 3D SEM Moiré method for micro height measurement". In: *Optics express* 21 (2013), pp. 15734–15746.

[Lon81] H. C. Longuet-Higgins. "A computer algorithm for reconstructing a scene from two projections". In: *Nature* 293 (1981), pp. 133–135.

[Low04] D. G. Lowe. "Distinctive image features from scale-invariant keypoints". In: *Int. J. of Computer Vision* 60 (2004), pp. 91–110.

[LZ99] C. Loop and Z. Zhang. "Computing rectifying homographies for stereo vision". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR'99)* 1 (1999).

[Mar13] N. Marturi. "Vision and visual servoing for nanomanipulation and nanocharacterization using scanning electron microscope". PhD thesis. UFC, Besançon, 2013.

[Mar63] D. W. Marquardt. "An algorithm for least-squares estimation of nonlinear parameters". In: *J. Soc. Ind. Appl. Math.* 11 (1963), pp. 431–441.

[MDP13] N. Marturi, S. Dembélé, and N. Piat. "Depth and shape estimation from focus in scanning electron microscope for micromanipulation". In: *IEEE Int. Conf. Control Autom. Robot. Embedded Syst. (CARE'13)*. 2013, pp. 1–6.

[MHW+13] Z. Ma, K. He, Y. Wei, J. Sun, and E. Wu. "Constant time weighted median filtering for stereo matching and beyond". In: *IEEE Int. Conf. Computer Vision (ICCV'13)*. 2013, pp. 49–56.

[MJC+14] M. R. Mikczinski, G. Josefsson, G. Chinga-Carrasco, E. K. Gamstedt, and S. Fatikow. "Fabrication and characterization of three-dimensional InGaAs/GaAs nanosprings". In: *IEEE Trans. Robot.* 30 (2014), pp. 115–119.

[ML09] M. Muja and D. G. Lowe. "Fast approximate nearest neighbors with automatic algorithm configuration". In: *VISAPP (1)* 2 (2009), pp. 331–340.

[ML14] M. Muja and D. G. Lowe. "Scalable nearest neighbor algorithms for high dimensional data". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 36 (2014), pp. 2227–2240.

[MLL+09] C. Mao, C. Liang, W. Luo, J. Bao, J. Shen, X. Hou, and W. Zhao. "Preparation of lotus-leaf-like polystyrene micro-and nanostructure films and its blood compatibility". In: *J. Mat. Chem.* 19 (2009), pp. 9025–9029.

[MMB03] C. G. Moles, P. Mendes, and J. R. Banga. "Parameter estimation in biochemical pathways: a comparison of global optimization methods". In: *Genome Research* 13 (2003), pp. 2467–2474.

[MSS+12] B. Muralikrishnan, J. Stone, C. Shakarji, and J. Stoup. "Performing three-dimensional measurements on micro-scale features using a flexible coordinate measuring machine fiber probe with ellipsoidal tip". In: *Meas. Sci. Technol.* 23 (2012), p. 025002.

[MTD+13] N. Marturi, B. Tamadazte, S. Dembélé, and N. Piat. "Visual servoing-based approach for efficient autofocusing in scanning electron microscope". In: *IEEE Int. Conf. on Intell. Robots and Syst. (IROS'13)*. 2013, pp. 2677–2682.

[MTD+16] N. Marturi, B. Tamadazte, S. Dembélé, and N. Piat. "Image-Guided Nanopositioning Scheme for SEM". In: *IEEE Trans. Autom. Sci. Eng.* 99 (2016), pp. 1–12.

[NJS97] F. Nicolls, G. de Jager, and B. Sewell. "Use of a general imaging model to achieve predictive autofocus in the scanning electron microscope". In: *Ultramicroscopy* 69 (1997), pp. 25–37.

[NMY+13] R. Nishi, Y. Moriyama, K. Yoshida, N. Kajimura, H. Mogaki, M. Ozawa, and S. Isakozawa. "An autofocus method using quasi-Gaussian fitting of image sharpness in ultra-high-voltage electron microscopy". In: *J. Microsc.* 62 (2013), pp. 515–519.

[NN94] S. K. Nayar and Y. Nakagawa. "Shape from focus". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 16 (1994), pp. 824–831.

[Oli00] J. Oliensis. "A critique of structure-from-motion algorithms". In: *Computer Vision and Image Understanding* 80 (2000), pp. 172–214.

[OPT98] K. Ong, J. Phang, and J. Thong. "A robust focusing and astigmatism correction method for the scanning electron microscope. Part II: Autocorrelation based coarse focusing method". In: *Scanning* 20 (1998), pp. 324–334.

[PBB+02] J.-L. Pouchou, D. Boivin, P. Beauchêne, G. L. Besnerais, and F. Vignon. "3D reconstruction of rough surfaces by SEM stereo imaging". In: *Microchim. Acta* 139 (2002), pp. 135–144.

[PK97] C. J. Poelman and T. Kanade. "A paraperspective factorization method for shape and motion recovery". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (1997), pp. 206–218.

[PKV99] M. Pollefeys, R. Koch, and L. Van Gool. "A simple and efficient rectification method for general motion". In: *IEEE Int. Conf. Computer Vision (ICCV'99)*. Vol. 1. 1999, pp. 496–501.

[Por08] F. Porikli. "Constant time O (1) bilateral filtering". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR'08)*. 2008, pp. 1–8.

[PPG13] S. Pertuz, D. Puig, and M. A. Garcia. "Analysis of focus measure operators for shape-from-focus". In: *Pattern Recognition* 46 (2013), pp. 1415–1432.

[PPV08] R. Pintus, S. Podda, and M. Vanzi. "An automatic alignment procedure for a four-source photometric stereo technique applied to scanning electron microscopy". In: *IEEE Trans. Instrum. Meas.* 57 (2008), pp. 989–996.

[Pri96] M. D. Pritt. "Structure and motion from two orthographic views". In: *JOSA A* 13 (1996), pp. 916–921.

[PS05] J. Paluszyński and W. Slowko. "Surface reconstruction with the photometric method in SEM". In: *Vacuum* 78 (2005), pp. 533–537.

[PV99] M. Pollefeys and L. Van Gool. "Stratified self-calibration with the modulus constraint". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 21 (1999), pp. 707–724.

[Qia99] N. Qian. "On the momentum term in gradient descent learning algorithms". In: *Neural networks* 12 (1999), pp. 145–151.

[Qua96] L. Quan. "Self-calibration of an affine camera from multiple views". In: *Int. J. Computer Vision* 19 (1996), pp. 93–105.

[Rad13] R. J. Radke. *Computer vision for visual effects*. Cambridge University Press, 2013.

[RD05] E. Rosten and T. Drummond. "Fusing points and lines for high performance tracking". In: *IEEE Int. Conf. Computer Vision (ICCV'05)*. Vol. 2. 2005, pp. 1508–1515.

[Rei72] A. C. Reimschuessel. "Scanning electron microscopy-Part I". In: *J. Chem. Educ.* 49 (1972), A413.

[RMM+12] M. Rudnaya, H. ter Morsche, J. Maubach, and R. Mattheij. "A derivative-based fast autofocus method in electron microscopy". In: *J. Math. Imaging Vis.* 44 (2012), pp. 38–51.

[RMM10] M. Rudnaya, R. Mattheij, and J. Maubach. "Evaluating sharpness functions for automated scanning electron microscopy". In: *J. Microsc.* 240 (2010), pp. 38–49.

[Rob63] L. G. Roberts. "Machine perception of three-dimensional soups". PhD thesis. MIT, Boston, 1963.

[Rou84] P. J. Rousseeuw. "Least median of squares regression". In: *J. Am. Stat. Assoc.* 79 (1984), pp. 871–880.

[RRB+91] J.-M. Renders, E. Rossignol, M. Becquet, and R. Hanus. "Kinematic calibration and geometrical parameter identification for robots". In: *IEEE Trans. Robot. Autom.* 7 (1991), pp. 721–732.

[RT12] C. Ru and S. To. “Contact detection for nanomanipulation in a scanning electron microscope”. In: *Ultramicroscopy* 118 (2012), pp. 61–66.

[RZH+12] C. Ru, Y. Zhang, H. Huang, and T. Chen. “An improved visual tracking method in scanning electron microscope”. In: *Microscopy and Microanalysis* 18 (2012), pp. 612–620.

[SF06] T. Sievers and S. Fatikow. “Real-time object tracking for the robot-based nanohandling in a scanning electron microscope”. In: *J. Micromech.* 3 (2006), pp. 267–284.

[SIA+11] S. Sakai, K. Ito, T. Aoki, and H. Unten. “Accurate and dense wide-baseline stereo matching using SW-POC”. In: *IEEE Asian Conf. Pattern Recognition (ACPR’11)*. 2011, pp. 335–339.

[SK97] R. Szeliski and S. B. Kang. “Shape ambiguities in structure from motion”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (1997), pp. 506–512.

[SLY+16] C. Shi, D. K. Luu, Q. Yang, J. Liu, J. Chen, C. Ru, S. Xie, J. Luo, J. Ge, and Y. Sun. “Recent advances in nanorobotic manipulation inside scanning electron microscopes”. In: *IEEE Trans. Robot.* 2 (2016), p. 16024.

[SRK+02] O. Sinram, M. Ritter, S. Kleindick, A. Schertel, H. Hohenberg, and J. Albertz. “Calibration of a SEM, using a nanopositioning tilting table and a microscopic calibration pyramid”. In: *ISPRS Archives* 34 (2002), pp. 210–215.

[SS02] D. Scharstein and R. Szeliski. “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms”. In: *Int. J. Computer Vision* 47 (2002), pp. 7–42.

[ST96] P. Sturm and B. Triggs. “A factorization based algorithm for multi-image projective structure and motion”. In: *European Conf. Computer Vision (ECCV’96)*. 1996, pp. 709–720.

[Stu97] P. Sturm. “Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction”. In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR’97)*. 1997, pp. 1100–1105.

[SZB95] L. S. Shapiro, A. Zisserman, and M. Brady. “3D motion recovery via affine epipolar geometry”. In: *Int. J. Computer Vision* 16 (1995), pp. 147–182.

[TAG+17] S. Thiele, K. Arzenbacher, T. Gissibl, H. Giessen, and A. M. Herkommer. “3D-printed eagle eye: Compound microlens system for foveated imaging”. In: *Science Advances* 3 (2017), e1602655.

[TAJ77] A. N. Tikhonov, V. Y. Arsenin, and F. John. *Solutions of ill-posed problems*. Vol. 14. Winston Washington, DC, 1977.

[TH12] T. Tieleman and G. Hinton. “RMSProp, Coursera: Neural Networks for Machine Learning”. In: *Technical report* (2012).

[TK90] C. Tomasi and T. Kanade. “Shape and motion without depth”. In: *IEEE Int. Conf. Computer Vision (ICCV’90)*. 1990, pp. 91–95.

[TK91] C. Tomasi and T. Kanade. *Detection and tracking of point features*. 1991.

[TK92] C. Tomasi and T. Kanade. "Shape and motion from image streams under orthography: a factorization method". In: *Int. J. Computer Vision* 9 (1992), pp. 137–154.

[TKA+15] A. P. Tafti, A. B. Kirkpatrick, Z. Alavi, H. A. Owen, and Z. Yu. "Recent advances in 3D SEM surface reconstruction". In: *Micron* 78 (2015), pp. 54–66.

[TKH+15] A. P. Tafti, A. B. Kirkpatrick, J. D. Holz, H. A. Owen, and Z. Yu. *3DSEM: A Dataset for 3D SEM Surface Reconstruction*. 2015.

[Tri98] B. Triggs. "Autocalibration from planar scenes". In: *European Conf. Computer Vision (ECCV'98)*. 1998, pp. 89–105.

[TZ00] P. H. Torr and A. Zisserman. "MLESAC: A new robust estimator with application to estimating image geometry". In: *Computer Vision and Image Understanding* 78 (2000), pp. 138–156.

[ULP+07] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Martí. "Scatter search and local NLP solvers: A multistart framework for global optimization". In: *INFORMS J. Comput.* 19 (2007), pp. 328–340.

[VSF+10] T. Vynnyk, T. Schultheis, T. Fahlbusch, and E. Reithmeier. "3D measurement with the stereo scanning electron microscope on sub-micrometer structures". In: *J. Eur. Opt. Soc. Rapid Publ.* 5 (2010).

[Wei09] T. Weise. "Global optimization algorithms - Theory and Application". In: *Self-Published,* (2009).

[Woo80] R. J. Woodham. "Photometric method for determining surface orientation from multiple images". In: *Optical engineering* 19 (1980), pp. 191139–191139.

[WWZ12] Z. Wu, D. Wang, and F. Zhou. "Bilateral prediction and intersection calculation autofocus method for automated microscopy". In: *J. Microsc.* 2048 (2012), pp. 271–280.

[Xie11] J. Xie. *Stereomicroscopy: 3D imaging and the third dimension measurement*. Application Note, Agilent Technologies, 2011.

[YAK17] S. Yan, A. Adegbule, and T. C. Kibbey. "A hybrid 3D SEM reconstruction method optimized for complex geologic material surfaces". In: *Micron* 99 (2017), pp. 26–31.

[YTA09] Q. Yang, K.-H. Tan, and N. Ahuja. "Real-time O(1) bilateral filtering". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR'09)*. 2009, pp. 557–564.

[ZDF+95] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry". In: *Artificial intelligence* 78 (1995), pp. 87–119.

[Zha99] Z. Zhang. "Flexible camera calibration by viewing a plane from unknown orientations". In: *IEEE Int. Conf. Computer Vision (ICCV'99)*. Vol. 1. 1999, pp. 666–673.

[ZTF+13] S. Zimmermann, T. Tiemerding, S. Fatikow, T. Wang, T. Li, and Y. Wang. "Automated mechanical characterization of 2D materials using SEM based visual servoing". In: *Int. Conf. Manipulation, Manufacturing and Measurement on the Nanoscale (3M-NANO'13)*. 2013, pp. 283–295.

[ZWZ+14] F.-Y. Zhu, Q.-Q. Wang, X.-S. Zhang, W. Hu, X. Zhao, and H.-X. Zhang. "3D nanostructure reconstruction based on the SEM imaging principle, and applications". In: *Nanotechnology* 25 (2014), p. 185705.

# Appendices

## Appendix A Experimental setup

The hardware set-up architecture used in this project is shown in Figure 6.7. The SEM used in this work is a CARL Zeiss Auriga 60 SEM. It has Schottky Field Emitter in the electron column that converges the beam towards the sample surface. Its electron column is equipped with all the elements explained previously. Its objective aperture is controlled electronically and may take the following values: 7,20,30,60,120 µm. The accelerating voltage for the SEM varies from 0:1 kV to 15 kV and the achievable resolution at 15 kV is 1 nm. The magnication of the SEM varies from 12 to 1,000,000.

The SEM chamber is equipped with a mobile platform (stage) that can be controlled externally using keyboard. It has six motorized axes: three translations, one continuous rotation ($0 - 360°$), tilt ($-10$-$60°$) and analytical working distance that varies from 0–20 mm. The view on the vacuum chamber from a camera installed within is shown in Figure 6.8.

Another element installed within SEM vacuum chamber is the 6-DDL manipulation robot. It was developed in FEMTO-ST Institute (AS2M department) for the purposes of micro- and nanoassembly. It contains three translation axes with nanometric precision and three rotations. Two of them are realized as a goniometer allowing to perform rotations in range $(-7°; 7°)$. The last rotation is a continuous one $(0 - 360°)$.

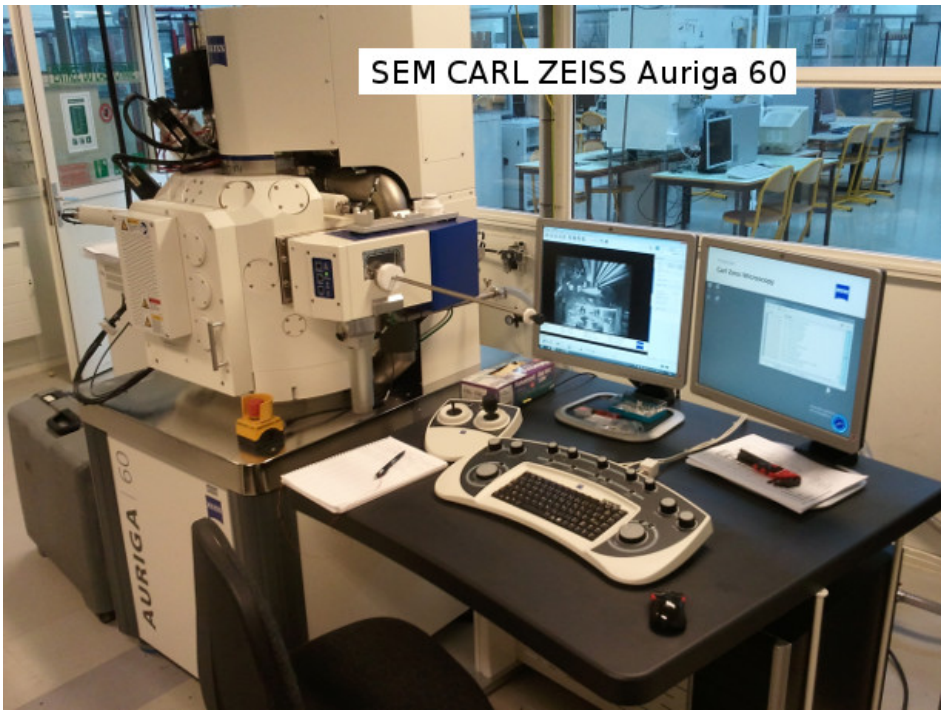



Figure 6.7: Experimental environment showing the hardware set-up.

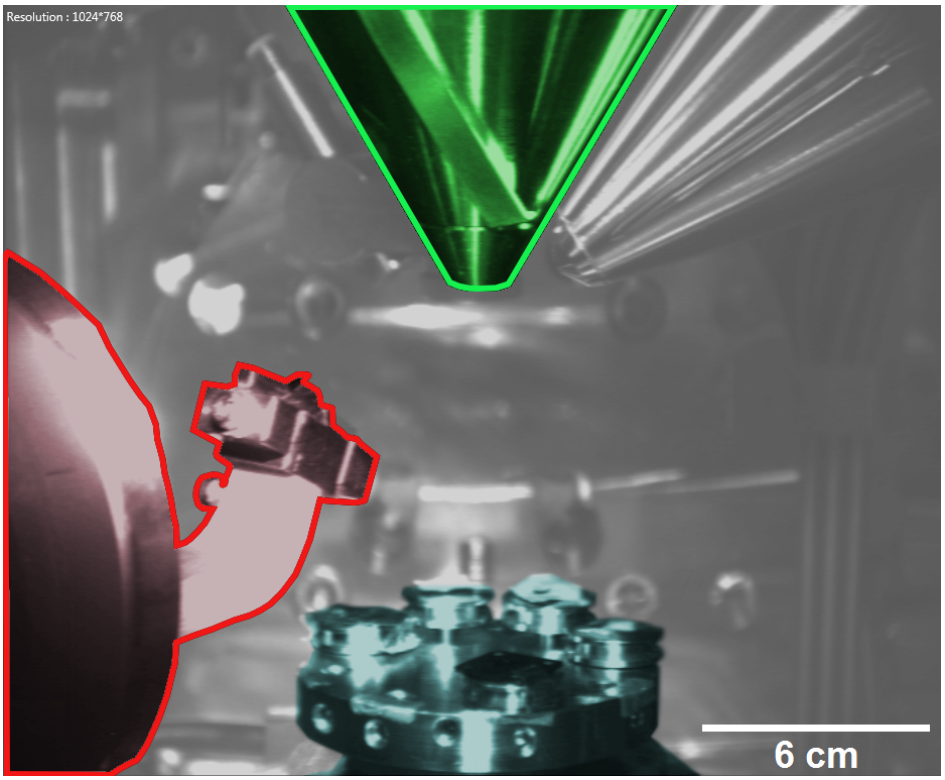



Figure 6.8: The view on the SEM vacuum chamber from a camera installed within. Green: SEM column; red: manipulation robot, blue: robotic stage.

# Appendix B Camera vs object motion

In SEM, an image pair is obtained by moving the robotic stage holding the sample: camera is fixed, object is moving. However, for the ease of presentation, we consider the object was static and that the camera (SEM) performed the motion. It can be seen from Figure 6.9 that these situations are equivalent.

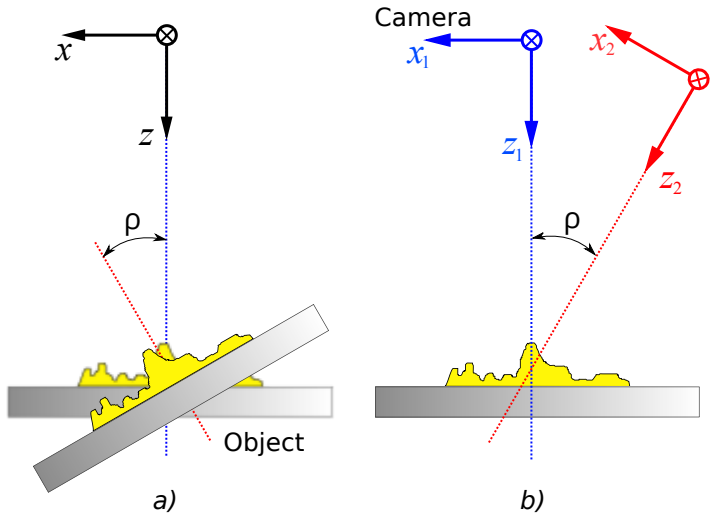
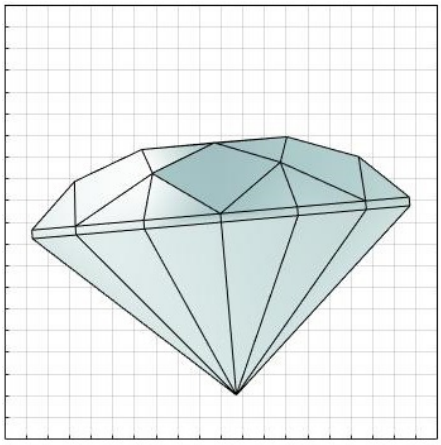



Figure 6.9: Equivalence between object motion and camera motion. a) object is moving, camera is static, b) camera is moving, object is static.

# Appendix C Diamond: synthetic image data



```
vertices = [
         0   -147.20     75.20;
         0    -16.00    303.20
   -119.20   -100.80    162.40
    118.40   -100.80    162.40;
    134.40   -146.40      1.60;
    263.20    -98.40         0;
    268.80    -16.00    160.00;
    142.40    -16.00    269.60;
    349.60    -16.00     23.20;
    268.80         0    160.00;
    142.40         0    269.60;
    349.60         0     23.20;
         0         0    303.20;
         0    320.00         0;
   -349.60         0     23.20;
   -142.40         0    269.60;
   -268.80         0    160.00;
   -349.60    -16.00     23.20;
   -142.40    -16.00    269.60;
   -268.80    -16.00    160.00;
   -263.20    -98.40         0;
   -134.40   -146.40      1.60
];
```

```
facesTriangles = [
     1    4     5;
     1    5    22;
     4    7     8;
     4    2     8;
     6    7     9;
    13   11    14;
    11   10    14;
    10   12    14;
     1   22     3;
     2    3    19;
    19    3    20;
    20   21    18;
    13   14    16;
    16   17    14;
    17   15    14
];

facesQuad = [
    17   20    18    15;
     1    4     2     3;
     4    5     6     7;
     3   22    21    20;
     2    8    11    13;
     8    7    10    11;
     7    9    12    10;
     2   13    16    19;
    16   19    20    17
];
```

# Appendix D SEM image datasets

## D.1. Brassica

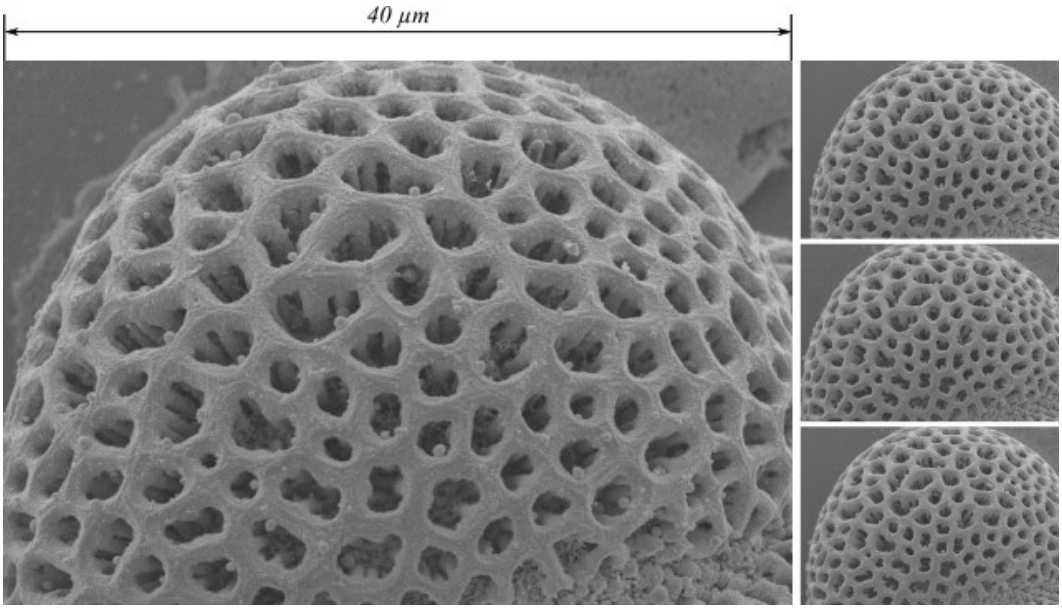



Figure 6.10: **brassica** image dataset.

Table 6.1: Properties of **brassica** image dataset.

| Name | **brassica** |
|---|---|
| Description | pollen grain from Brassica rapa |
| Origin | [TKA+15; TKH+15] |
| Number of images | 4 |
| Image size, pixels | 854×590 |
| Stage motion | 3 degrees |
| SEM | Hitachi S-4800 FE-SEM |
| Detector | SE |
| Magnification | ×10000 |
| Accelerating voltage | 3.0 kV |
| Working distance | 14.4 mm |
| Image pixel size | 47 nm |

## D.2. Grid

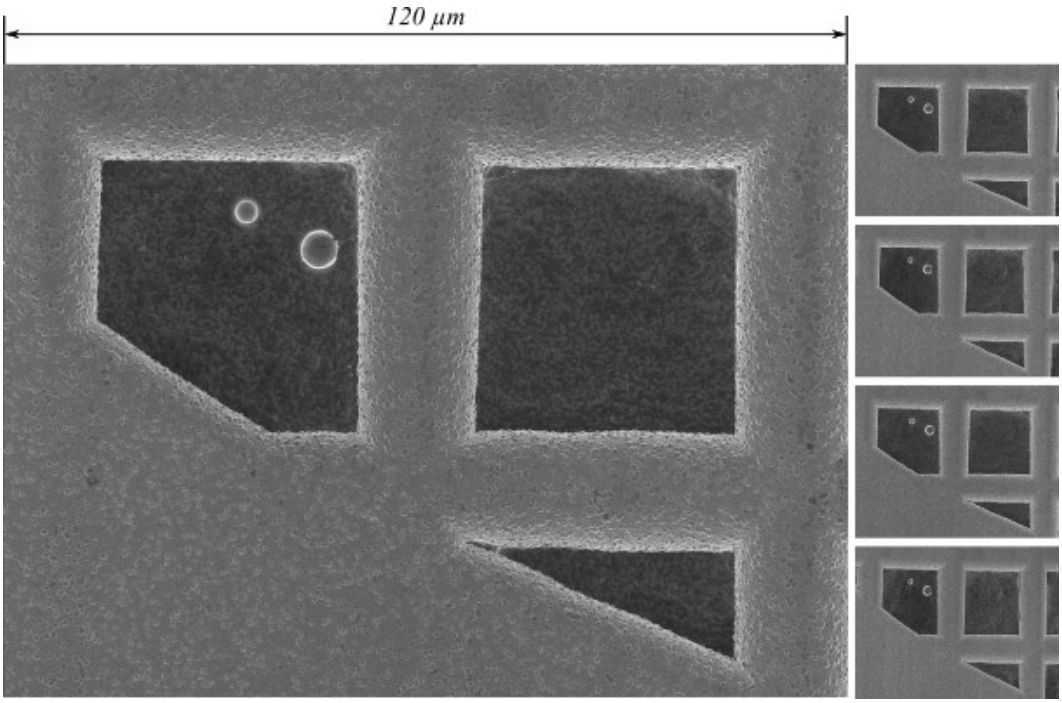



Figure 6.11: **grid** image dataset.

Table 6.2: Properties of **grid** image dataset.

| Name | **grid** |
|---|---|
| Description | material object TEM copper grid |
| Origin | [TKA+15; TKH+15] |
| Number of images | 5 |
| Image size, pixels | 2560×1920 |
| Stage motion | 7 degrees |
| SEM | Hitachi S-4800 FE-SEM |
| Detector | SE |
| Magnification | ×10000 |
| Accelerating voltage | 3.0 kV |
| Working distance | 14.4 mm |
| Image pixel size | 47 nm |

## D.3. Cutting tool

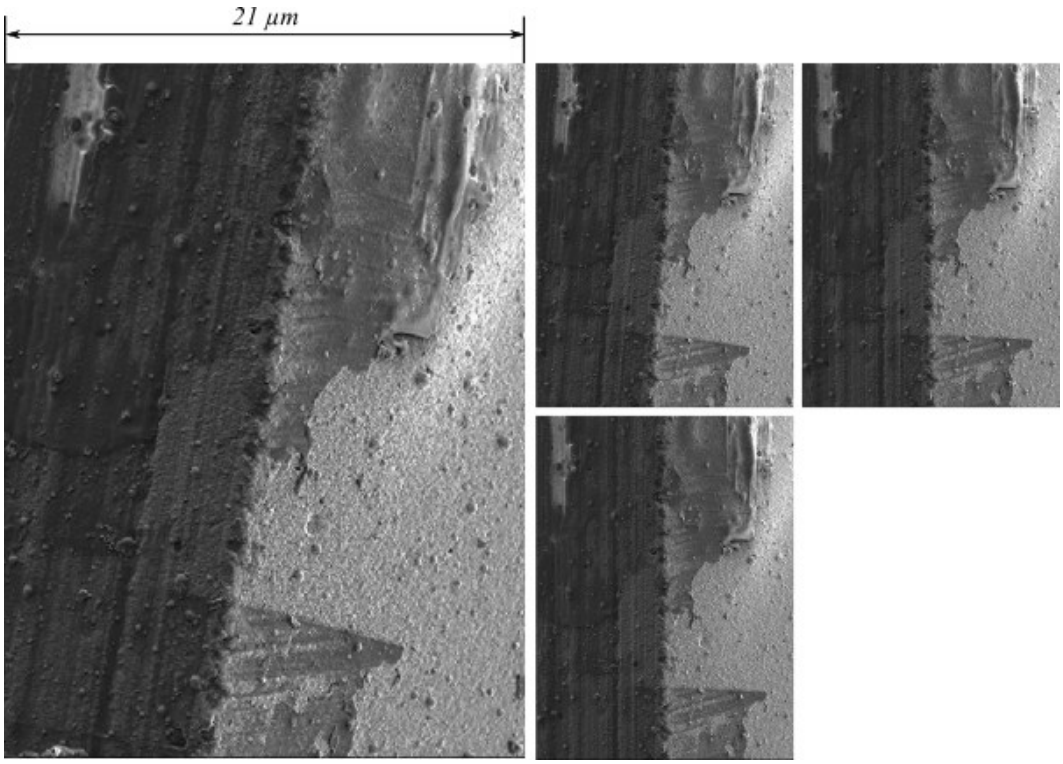



Figure 6.12: **cutting** image dataset.

Table 6.3: Properties of **cutting** image dataset.

| Name | **cutting** |
|---|---|
| Description | cutting edge of a microfabrication tool |
| Origin | FEMTO-ST Institute |
| Number of images | 4 |
| Image size, pixels | 1024×768 |
| Stage motion | complex rotation |
| SEM | Carl Zeiss AURIGA 60 SEM |
| Detector | SE |
| Magnification | ×2000 |
| Accelerating voltage | 5.0 kV |
| Working distance | 10.6 mm |
| Image pixel size | 27.91 nm |

## D.4. Potamogeton

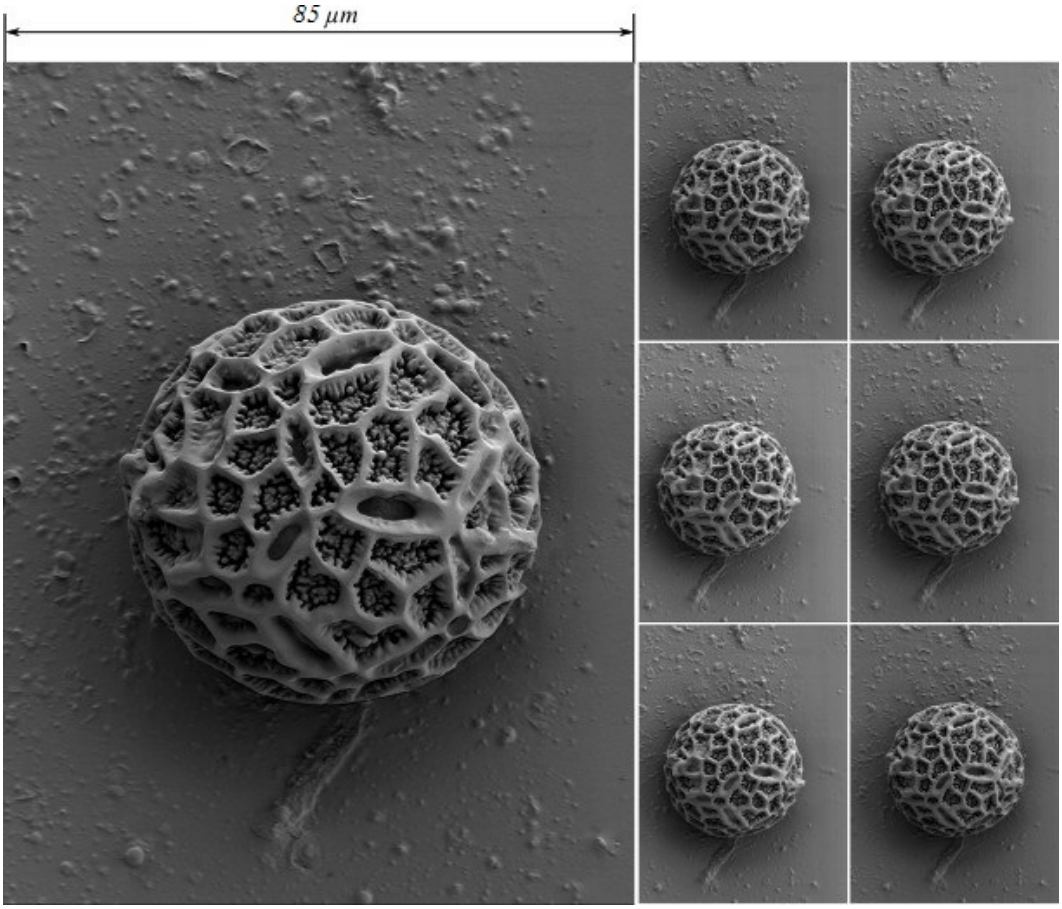



Figure 6.13: **pot** image dataset.

Table 6.4: Properties of **pot** image dataset.

| Name | **pot** |
|---|---|
| Description | pollen grain of aquatic, mostly freshwater, plant of the family *Potamogetonaceae* |
| Origin | FEMTO-ST Institute |
| Number of images | 7 |
| Image size, pixels | 2048×1536 |
| Stage motion | 3 degrees |
| SEM | Carl Zeiss AURIGA 60 SEM |
| Detector | SE |
| Magnification | ×1000 |
| Accelerating voltage | 1.0 kV |
| Working distance | 6.2 |
| Image pixel size | 55.82 nm |

## D.5. Potamogeton2

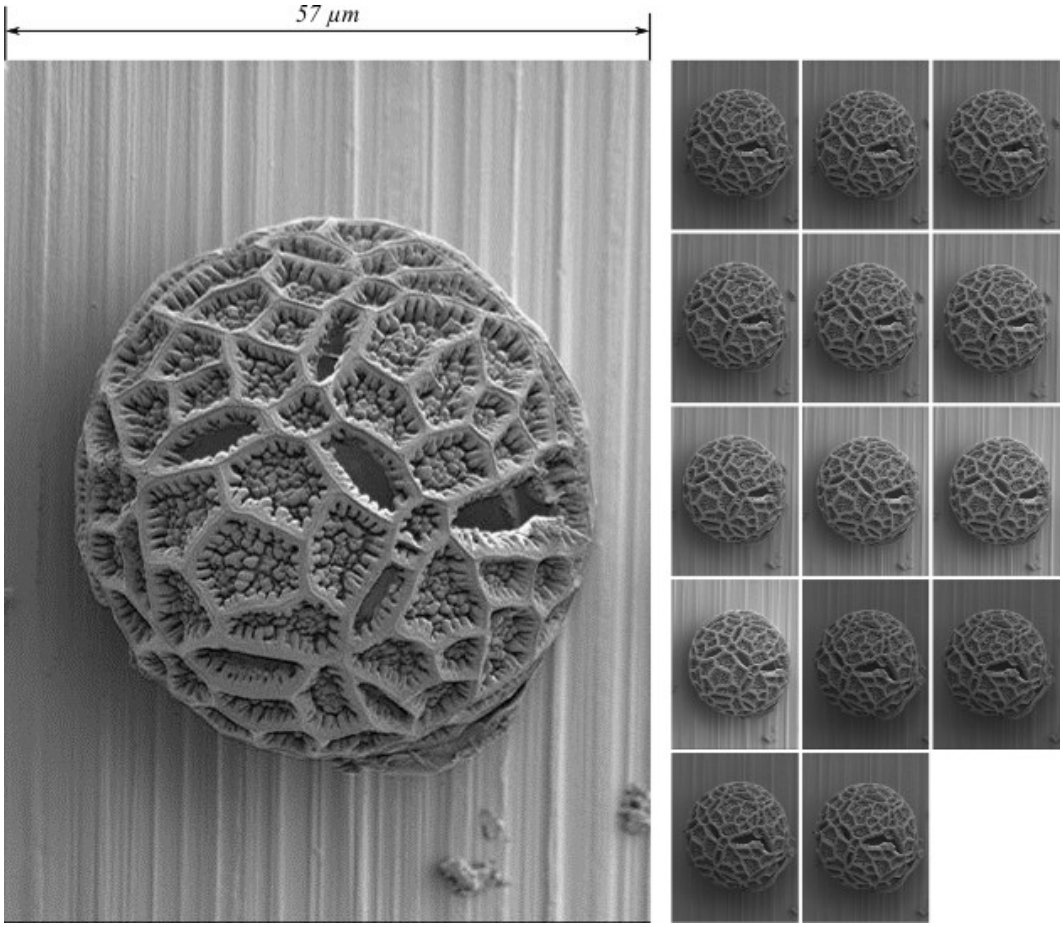



Figure 6.14: **pot2** image dataset.

Table 6.5: Properties of **pot2** image dataset.

| Name | **pot2** |
|---|---|
| Description | pollen grain of aquatic, mostly freshwater, plant of the family *Potamogetonaceae* |
| Origin | FEMTO-ST Institute |
| Number of images | 15 |
| Image size, pixels | 1024×768 |
| Stage motion | complex rotation |
| SEM | Carl Zeiss AURIGA 60 SEM |
| Detector | SE |
| Magnification | ×1500 |
| Accelerating voltage | 2.0 kV |
| Working distance | 9.0 mm |
| Image pixel size | 74.43 nm |

## D.6. PPY

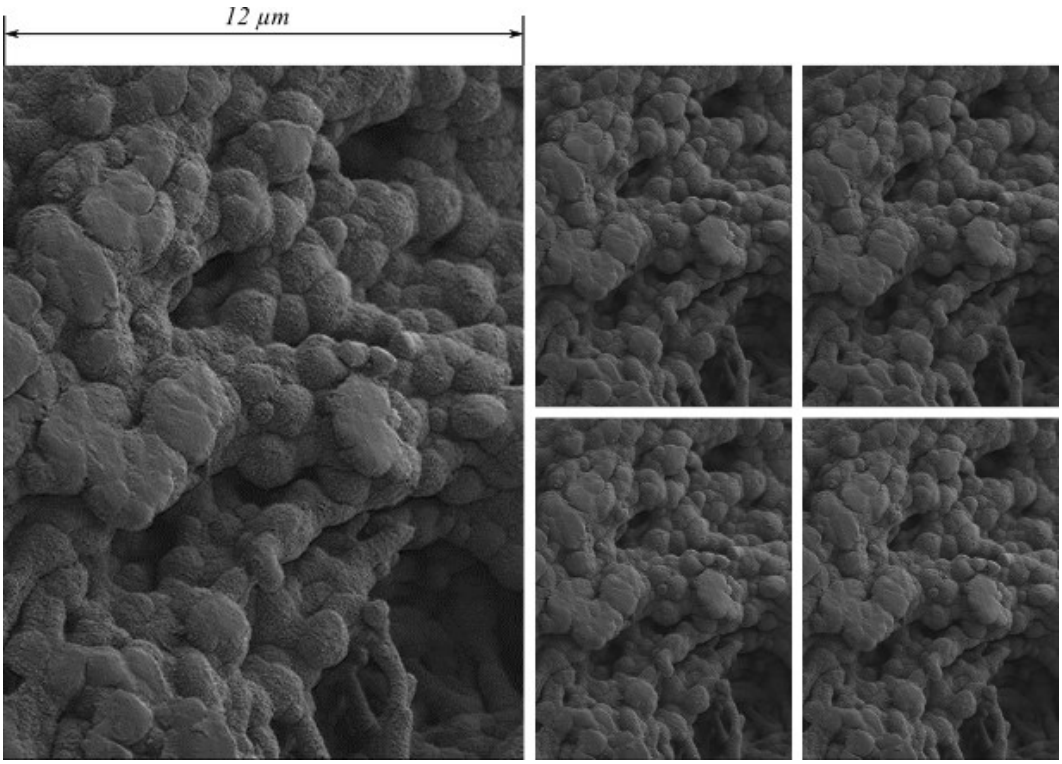



Figure 6.15: **ppy** image dataset.

Table 6.6: Properties of **ppy** image dataset.

| Name | **ppy** |
|---|---|
| Description | Polypyrrole (PPY) material surface |
| Origin | FEMTO-ST Institute |
| Number of images | 5 |
| Image size, pixels | 1024×768 |
| Stage motion | complex rotation |
| SEM | Carl Zeiss AURIGA 60 SEM |
| Detector | SE |
| Magnification | ×7,000 |
| Accelerating voltage | 1.0 kV |
| Working distance | 5.2 mm |
| Image pixel size | 15.95 nm |

# Appendix E Non-stationary function optimization.

The term non-stationary is referred to a function that may change its properties in time. This feature greatly reduce the number of optimization solutions. In this section we provide a very brief overview and analysis of the state-of-the-art optimization techniques for first-order optimization.

The most common first order algorithm allowing to find the minimum of a function $f(\xi)$ is the gradient descent which has the following update rule:

$$\xi_{n+1} = \xi_n - \alpha\hat{g} \tag{6.2}$$

where $\alpha$ denotes the gain or learning rate and $\hat{g}$ is the estimated gradient of function $f(\xi)$. Its value determines how important is the update in one iteration. In general, gradient descent achieves good results when the objective function in not corrupted by noise. When there is an important change in the gradient value, which is chaotic due to noise, the algorithm will impose too important change of parameter $\xi$ which may cause the instability of the system. Another drawback of gradient descent consists in high dependence on the value of $\alpha$. If the gain is too low, the convergence speed will also be low. In the case of autofocus it would greatly limit the maximum displacement speed. In contrast, if the gain is too high, the algorithm may suffer from oscillations about the maximum value. Therefore, several techniques were proposed in the literature to improve the performance of gradient descent.

*Momentum* [Qia99]. This first method is based on the following idea: if the sign of gradient does not change for a certain amount of time, i.e. the update direction stays the same, the update in this direction can be accelerated. It gives the following update rule:

$$\begin{aligned} m_{n+1} &\leftarrow \mu m_n - \alpha\hat{g} \\ \xi_{n+1} &\leftarrow \xi_n + m_{n+1} \end{aligned} \tag{6.3}$$

where $m$ is a first moment variable. Thus, instead of integrating the gradient, the velocity is integrated. The acceleration depends on the factor $\mu \in (0, 1)$. Its typical value is 0.9. This algorithm allows to improve the convergence speed and prevents the value of $\xi$ from chaotic jumps. However, it is not suitable for non-stationary functions. Assuming that the minimum is moving in one direction for some time and then changes it, the algorithm would not be able to respond quickly.

*AdaGrad* [DHS11]. In this case the learning rate is adaptive. It scales the current value of gradient according to the history of squared gradient values for previous iterations:

$$\begin{aligned} v_{n+1} &\leftarrow v_n + \hat{g}^2 \\ \xi_{n+1} &\leftarrow \xi_n - \alpha\frac{\hat{g}}{\sqrt{v_{n+1}}+\varepsilon} \end{aligned} \tag{6.4}$$

where $\varepsilon$ is a small constant (typical $10^{-8}$) allowing to avoid division by zero at first iterations, $v$ is a second moment variable. Despite the robustness of this algorithm, it is also not adapted for non-stationary functions: the history of gradient is stored for the whole time of optimization. As a result, it has the same drawback as Momentum, impossibility to quickly respond at the change of minimum position.

*RMSProp* [TH12]. The idea proposed here is not to store all values of the gradient but use the exponentially weighted moving average:

$$\begin{aligned} v_{n+1} &\leftarrow \beta v_n + (1-\beta)\hat{g}^2 \\ \xi_{n+1} &\leftarrow \xi_n - \alpha\frac{\hat{g}}{\sqrt{v_{n+1}}+\varepsilon} \end{aligned} \tag{6.5}$$

where $0 \leq \beta < 1$ is the parameter that determines how many previous gradients would be taken into account and with which weight factor. For instance, if $\beta = 0$ only the current estimate of the gradient will be used, and the algorithm will perform the update in its direction by the value of $\alpha$. It is worth to note that RMSProp is invariant to the scale of the gradient, as in previous example when $\beta = 0$. In practice, the value of $\beta$ is taken equal to 0.9 or 0.99. This is the first algorithm that has the necessary properties for non-stationary functions: filtering of chaotic jumps in the gradient values, robustness and quick response on function variations (if the value of $\beta$ is chosen correctly).

*Adam* [KB14]. This recently introduced method stands for adaptive moment estimation. In addition to store the exponentially moving average of squared gradients like RMSProp, it also stores the exponentially moving average of the gradient itself:

$$\begin{gathered} m_{n+1} \leftarrow \beta_1 m_n + (1-\beta_1)\hat{g} \\ v_{n+1} \leftarrow \beta_2 v_n + (1-\beta_2)\hat{g}^2 \\ \xi_{n+1} \leftarrow \xi_n - \alpha \frac{m_{n+1}}{\sqrt{v_{n+1}}+\varepsilon} \end{gathered} \tag{6.6}$$

The update rule is very similar to RMSProp, however, not the noisy gradient estimate $\hat{g}$ is used, but its averaged value $m$. It allows better filtering of the gradient while keeping the functionality needed for non-stationary function optimization. Typical values of parameters are: $\beta_1 = 0.9, \beta_2 = 0.99, \varepsilon = 10^{-8}$.

## Résumé :

L'objectif de ce travail est d'obtenir un modèle 3D d'un objet à partir d'une série d'images prises avec un Microscope Électronique à Balayage (MEB). Pour cela, nous utilisons la technique de reconstruction 3D qui est une application bien connue du domaine de vision par ordinateur. Cependant, en raison des spécificités de la formation d'images dans le MEB et dans la microscopie en général, les techniques existantes ne peuvent pas être appliquées aux images MEB. Les principales raisons à cela sont la projection parallèle et les problèmes d'étalonnage de MEB en tant que caméra. Ainsi, dans ce travail, nous avons développé un nouvel algorithme permettant de réaliser une reconstruction 3D dans le MEB tout en prenant en compte ces difficultés. De plus, comme la reconstruction est obtenue par auto-étalonnage de la caméra, l'utilisation des mires n'est plus requise. La sortie finale des techniques présentées est un nuage de points dense, pouvant donc contenir des millions de points, correspondant à la surface de l'objet.



**Mots-clés :** MEB, reconstruction 3D dense, camera affine, auto-étalonnage de cameras



## Abstract:

The goal of this work is to obtain a 3D model of an object from its multiple views acquired with Scanning Electron Microscope (SEM). For this, the technique of 3D reconstruction is used which is a well known application of computer vision. However, due to the specificities of image formation in SEM, and in microscale in general, the existing techniques are not applicable to the SEM images. The main reasons for that are the parallel projection and the problems of SEM calibration as a camera. As a result, in this work we developed a new algorithm allowing to achieve 3D reconstruction in SEM while taking into account these issues. Moreover, as the reconstruction is obtained through camera autocalibration, there is no need in calibration object. The final output of the presented techniques is a dense point cloud corresponding to the surface of the object that may contain millions of points.



**Keywords:** SEM, dense 3D reconstruction, affine camera, camera autocalibration


SPIM


■ École doctorale SPIM 1 rue Claude Goudimel F - 25030 Besançon cedex

■ tél. +33 (0)3 81 66 66 02 ■ **ed-spim@univ-fcomte.fr** ■ www.ed-spim.univ-fcomte.fr


UFC
UNIVERSITÉ
DE FRANCHE-COMTÉ