ADVANCED ELECTRON MICROSCOPY TECHNIQUES FOR LARGE-AREA STITCHING APPLICATIONS

ADVANCED PROCESSING OF SCANNING ELECTRON MICROSCOPY IMAGES IN 2-D AND 3-D DATASETS

By NASIM KHOONKARI, B.Sc., M.Sc.

A Thesis Submitted to the School of Graduate Studies in Partial
Fulfillment of the Requirements for
the Degree Doctor of Philosophy

McMaster University

DOCTOR OF PHILOSOPHY (2023)

Hamilton, Ontario, Canada (Computing and Software)

TITLE:	Advanced Processing of Scanning Electron Microscopy Images in 2-D and 3-D Datasets
AUTHOR:	Nasim Khoonkari B.Sc., M.Sc.
SUPERVISOR:	Dr. Christopher Anand Dr. Nabil Bassim
NUMBER OF PAGES:	xxi, 161

Lay Abstract

Applications in materials science often require the acquisition of images of semiconductor computer chips at very high resolution. Using cameras with even tens of millions of pixels might not give us enough resolution over a wide field of view. One approach is to acquire several images of parts of the sample at high magnification and assemble them into a single composite image. This way, we can preserve the high resolution over a wide area. Algorithms developed for assembling the composite image are known as tiling or mosaicing. This whole process is known as image stitching (and includes image registration). In this thesis, we develop specialized algorithms suited for the $2D$ stitching of semiconductor images, including the generalization to $3D$. This case is challenging because slight alignment errors may completely change the reconstructed circuit, and the images contain both repeated patterns (such as many parallel wires) and changes in brightness and distortions caused by the scanning device.

Abstract

To acquire high-resolution Scanning Electron Microscopy (SEM) images over wide areas, we must acquire several images “tiling” the surface and assemble them into a single composite image, using a process called image stitching. While for some applications, stitching is now routine, SEM mosaics of semiconductors pose several challenges: (1) by design, the image features (wire, via and dielectric) are highly repetitive, (2) the overlap between image tiles is small, (3) sample charging causes intensity variation between captures of the same region, and (4) machine instability causes non-linear deformation within tiles and between tiles. In this study, we compare the accuracy and computational cost of three well-known pixel-based techniques: Fast Fourier Transform (FFT), Sum of Squared Differences (SSD), and Normalized Cross Correlation (NCC). We compare well-known $2D$ algorithms, as well as novel projection-onto-$1D$ versions. The latter reduces the computational complexity from $O(n^2)$ to $O(n)$, where n is the number of pixels, without loss of accuracy, and in some cases, with greater accuracy. Another approach to reducing the computational complexity of image alignment is to compare isolated landmarks, rather than pixels. In semiconductor images, there are no natural fiducials and adding them would destroy the information required to reconstruct their circuits, so we introduce a new class of landmarks which we call numerical landmarks. Related to Harris corners, the novel numerical landmarks are insensitive to brightness variations and noise.

Finally, we consider the alignment problem between layers of image mosaics. Unlike in the “horizontal” directions, the vertical dimension is only sparsely sampled. Consequently, image features and landmarks cannot be used for alignment. Instead, we must rely on the relationship between vias (through-plane metalization) and wires (in-plane metalization), and we have developed a novel algorithm for matching vias in the lower layer with wires above, and use this to align subimages.

To my “Mom” for giving me the courage to follow my dreams

To my “Dad” for motivating me to stay strong

To “Mohammad”, my beloved brother, for his unconditional love and support

To “AlamAra”, my lovely aunt, for supporting my dreams

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Dr. Christopher Anand and Dr. Nabil Bassim, for their valuable guidance, support, and expertise throughout my doctoral research. The excellent and profound combination of their knowledge of mathematics, computer science, and materials science has greatly influenced the outcome of this thesis. I am truly grateful for their mentorship, insightful feedback, constant support, patience, and belief in my abilities. I am immensely grateful to them for always challenging me, which encourages my intellectual growth. I could not have imagined having better advisors and mentors during this four-year Ph.D. journey. I learned from them not only academically but also in my personal life, which is greatly valuable to me. I appreciate all their support and will always value this precious experience of being honoured to work under their supervision. I will always remember our discussions and how we turned problems into innovative ideas to solve real-world problems. I want to thank my Ph.D. committee members, Dr. Ned Nedialkov and Dr. Michael Greenwood, for all their efforts, valuable feedback, support and encouragement during committee meetings. I want to thank Dr. William Farmer for his great impact on my attitude toward learning logic and programming languages from a computer scientist's point of view. I am also thankful to the academic team of the Computing and Software department, especially Laurie Leblanc, Connie Carrabs,

and Stefanie Bittcher, for all their support. I want to thank the external examiner Dr. Bradley T. De Gregorio, for his insightful feedback on my thesis.

I want to thank our industrial partner, TechInsights, especially Chris Pawlowicz, Mike Green, Hamed Bagheri, and Bruno Trindade, for describing the stitching problem and providing the data for us. It was a pleasure to work in such an innovative and scientific atmosphere. I learned a lot during our biweekly meetings while we shared our knowledge between academia and industry. My sincere thanks also are extended to the many research scientists at the Canadian Centre for Electron Microscopy for providing their help and expertise. Also, I want to thank Fibics Inc. for their support.

I would thank all my colleagues in the Computer Science department and Materials Science department for their support and encouragement throughout my Ph.D. experience. Especially, I want to thank Christopher Schankula, Ahmed Al Mudarris, Akshay Kumar Arumugasamy, and Curtis D'Alves from Computer Science and Caleb Whittier, Bhaveshkumar Kamaliya, and Hesham El-Sherif from Materials Science for all their help and insightful feedback.

My friends and families from around the world, I could not be luckier than having had the privilege of having you in my life. I want to thank Yasamin Sartipi and Dr. Amid Shakeri for taking care of me from the very first day of my entrance to Canada. I am so grateful because of knowing you and great friends that I found because of you. I want to thank my lovely roommate Dr. Sara Jamali, and sweetheart friends Maryam, Mohsen, Atife, Dr. Sara Imani, and Sahand for all their support and care. My friends in Germany supported me no matter of distances and time differences; I carried you in my heart all these years. Pegah, Reza, Farzaneh, Mohammad, Hamid, Behnoosh, Zeynab, Mahshid, Saba, Mahdi, Neda, Mohammadreza, Amin, Eileen, and Ali; thank you for being a warm embrace to all my homesickness and encouraging my decisions. You are all precious entities in my life. I want to thank Dr. Matthias Dütsch for taking care of me and supporting my health and well-being. My friends in Iran, Leila, Vahid and Fateme, thank you for all your support and encouragement.

Mom, Dad, I am writing to save these words here. I woke up every day by your messages or phone calls and your sweet words to support my dreams. I started my days with your best wishes and tried to be my best. Thank you for giving me life, bringing me into this world as your daughter, and giving my dreams the wings to fly. Dr. Mohammad Khoonkari, my sweet brother, you are always the first embrace I search for in joy and sorrow. We started our journey for a better future together, thank you for always being there for me, and you should know I will be always there for you no matter the situation or distance. Finally, I want to thank my sweet aunt, Dr. AlamAra Karimi Abadei. You are a role model, and I am so lucky to have your passion, advice and help in my life.

Table of Contents

List of Figures

Notation, Definitions, and Abbreviations

Notation

$\forall$	For all elements in a set
ℓ^1	Manhattan distance
ℓ^2	Shortest distance between two points, or minimum difference
P	Projection of the image
P^r	row-wise summation, projection on $x-$axis
P^c	column-wise summation, projection on $y-$axis
I_j	jth column
I_i	ith row
λ	Eigen value
λ_{min}	Minimum eigen value

λ_{max}	Maximum eigen value
λ_1, λ_2	Eigen values
$\frac{\partial}{\partial f}$	Partial derivative
w	Weight function
L	Lanczos window
R	Rotation matrix
$E(x, y)$	Energy function at the pixel coordinate (x, y)
$I(x, y)$	Pixel coordinate (x, y) in image I
$I(x + u, y + v)$	Pixel coordinate (x, y) in image I shifted by u in $x-$direction and v in $y-$direction.
∇	Gradient
∇^2	Hessian
$\approx$	Approximately equal
B	Subimages (Box)
S	Skew Matrix
I_T	Image refer to the tile in the stitching grid
Δ	Shift is subimage

f	Frequancy domain
t	Time domain
$\omega(a_{00}, b_{00})$	Normalization function with respect to the shift
$E(X)$	Expected value of X
$Var(X)$	Variance of random variable X
c_{xy}	Covariance between variables x and y
r_{xy}	Pearson correlation coefficien
$k = (k_x, k_y)$	Coordinate in frequency space (when discussing Fourier Transform)
(k, l)	Coordinate of pixel in image

Definitions

Gate Gate or a logic gate in a microchip is a specific arrangement of transistors.

Via Through-plane wires connecting different layers vertically in semiconductors.

Wire metal under a layer of dielectric in semiconductors.

Dielectric Insulating material or a very poor conductor of electric current.

Abbreviations

SEM	Scanning electron microscopy
IC	Integrated circuit
CPU	Central processing unit
OM	Optical micrpscopy
EM	Electron micrpscopy
BSE	Backscattered electron
SE	Secondary electron
Z	Atomic number
PE	Primary electrons
R	Penetration depth in SEM
R	Response function in Harris feature detection
R	Rotation matrix in image distortion
ET	Everhart-Thornley detector
SNR	Signal to noise ratio
FOV	Field of view
ULSI	ultra-large-scale integration

SIFT	Scale-invariant feature transform
MIST	Microscopy Image Stitching tool
FFT	Fast fourier transform
SSD	Sum of squared differences
NCC	Normalized cross correlation
ACS	Auto-correlation surface
Det	Determinant
Tr	Trace
MST	Minimum spanning tree
MRI	Magnetic resonance imaging
CT	Computed tomography

Declaration of Academic Achievement

In this thesis, we present three novel algorithms. The first algorithm is a method of identifying numerical landmarks (a definition coined in this thesis). The second algorithm uses the projection of image regions onto $x-$ and $y-$ axes and the matching of the resulting $1D$ projections to determine an overall $2D$ translation for use in registration. The third algorithm aligns SEM images of successive layers of a semiconductor device by first extracting the positions of vias in the lower layer, and then searching for the best translation for subsets of vias such that they all or mostly connect to metalization in the upper layer.

Chapter 1

Introduction

Scanning electron microscopy (SEM) problems are complex and thus require interdisciplinary solutions involving mathematics, computer science and materials science. SEM probes a sample with an incident electron beam and then detects scattered electrons emitted from the sample's surface. This provides several signals with qualitative and quantitative information about the specimen. This information includes the sample's topography (through secondary electrons), composition (through backscattered electrons and characteristic x-rays), and crystallographic information (electron channelling and backscattered diffraction) [19].

The SEM ability to observe compositional variations within a specimen is by using atomic number dependence of electron backscattering. Experimental studies have been conducted to determine the relationship between the back-scattered coefficient and the specimen atomic number over a wide range of atomic numbers [47]. By selecting appropriate operating conditions, the detected back-scattered signal can be used to determine the atomic number of a specimen as well as the relationships between atomic number, which establishes the ratio of the two components. The composition of the area of interest can be directly determined using measurements of the back-scattered signal for suitable specimens, where the component elements are well separated in atomic number [3].

In this thesis, we analyze the network of semiconductor circuit metallization lines. The circuits that we image using SEM, have two major compositions of metal (typically copper, aluminum or tungsten) and dielectric, with three visible structures: dielectric without metal, metal wires under a layer of dielectric, and metal vias (through-plane wires connecting different layers vertically—i.e., into the sample [43], see Figure 1.1).

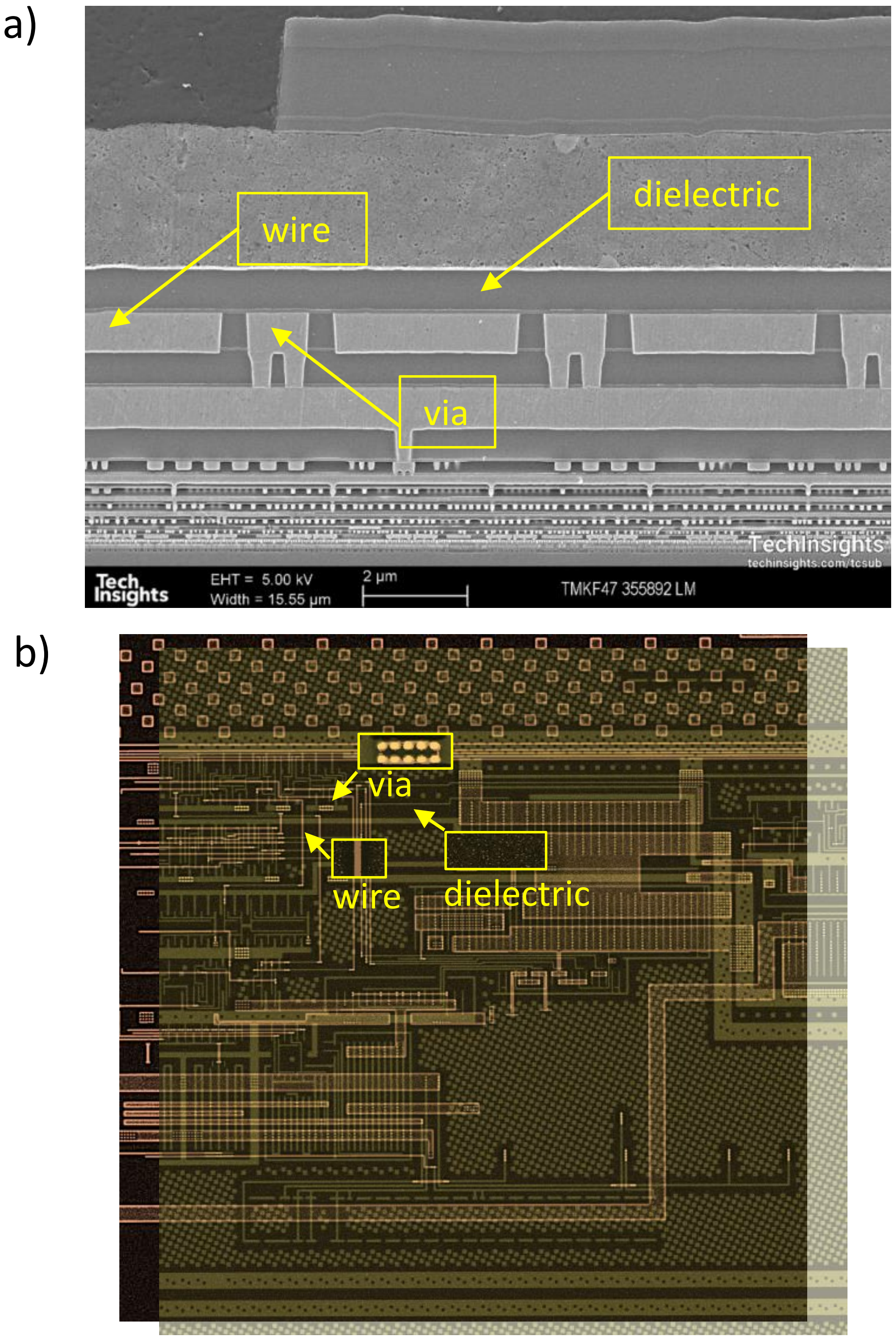

Figure 1.1: a)SEM image of the cross-section of integrated circuit metallization stack showing the dielectric, metal lines and vias, b) plane view of the two aligned layers showing the dielectric without metal, metal wires under a layer of silicon, and metal vias (through-plane wires connecting different layers vertically.

Figure 1.1 is the cross-section of the semiconductor sample, in which the three gray levels are not distinguished sharply. The reason for this observation is that when the images are taken from the top (i.e. a plan view), the depth of each material type varies at each position. For instance, vias are metal lines connecting two layers, so the amount of metal at the via position is high, and it looks brighter because of backscattered electron contrast. However, looking at the vias from the side assumes that the amount of material doesn't change with the depth in the image. The same is true for wires, but the depth of wires and the depth of the metal into the sample is lesser than vias. The amount of metal in the sample from a cross-section view is all the same because all of the metal is relatively "infinite" going into the plane. Conversely, parts of the sample with vias contain more metal in plain view, while the horizontal lines, which are wires, are made up of less metal. Therefore, in the plan view, the density of metal is higher for through-plane wires connecting different layers vertically than the horizontal wires (a via has a brighter gray level compared to wire). This is also true for all other components and materials in the structure of semiconductors. Dielectric, specifically, is not pure silicon (sometimes dielectric oxide, sometimes poly-Si for packaging) in the semiconductor samples, so its gray level depends on the local material density and composition. In [55], it is explained how different types of dielectric affect the emissivity of materials. It is also mentioned how the dielectric is decomposed and reacts.

Figure 1.2 shows a typical schematic of this structure, with five metallization layers in a cross-sectional view. To image each layer in plan view, the upper layers are cut away to a horizontal plane that intersects the vias, and the sample is then polished. Because the vias connect wires in the upper and lower metallization layers, they appear as vertical wires in the resulting image. We will image these structures in plan view for our work, thus imaging individually polished levels of this layer stack, see Figure 1.2. Because the polishing process exposes the vias' metal, the vias are the brightest features of the image as they have the highest amount of high atomic number material at those pixels. Each sample has a characteristic wire size, with slight variation and regular patterns repeated throughout each layer [42].

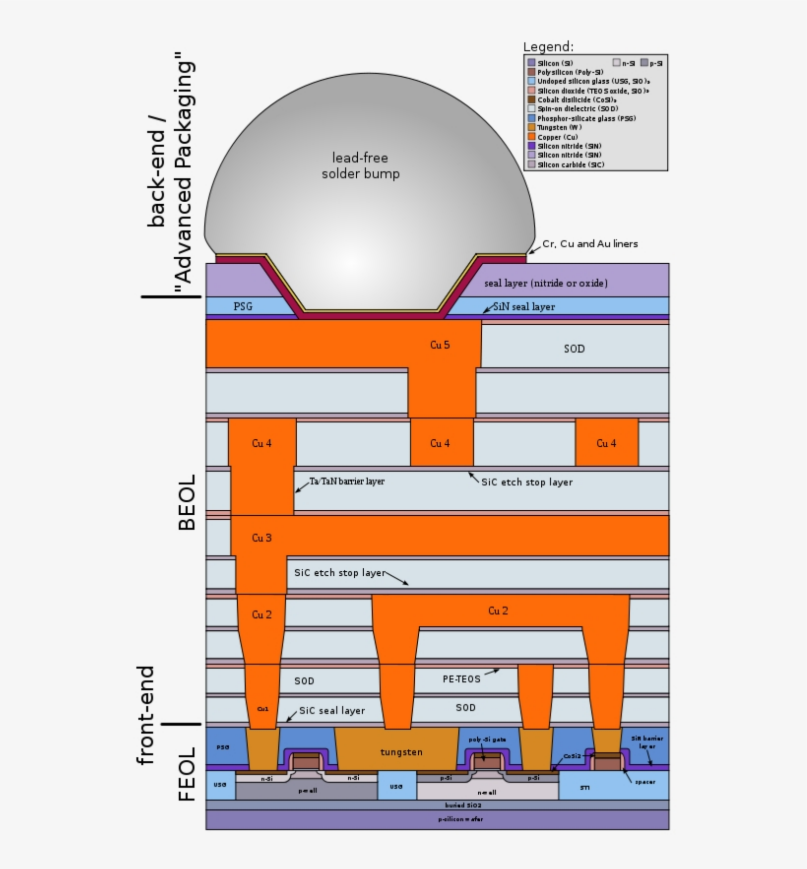

Figure 1.2: The electronic chip with five layers of metallization. This image is licensed under Creative Commons Attribution 2.5 Generic by user Cepheiden at https://commons.wikimedia.org/wiki/File:Cmos-chip-structure-in-2000s-(en).svg.

In a computer, an integrated circuit is an ensemble set of electronic circuits on one small flat piece (or chip) of semiconductor material, most often silicon. The current that flows through a gate establishes a voltage at a particular point in the circuit. This voltage represents a single bit of information. The voltage may be high (representing the value “1”) or low (representing the value “0”). At the most elementary level, transistors control the flow of electrical current through a circuit, with a metallic gate, allowing the device to form a “0” or a “1”[40]. A transistor is a miniature electronic component that performs two functions: switch or amplifier. The lowest level of a computer chip (with the smallest feature size) consists of transistors. The chip designer selects the transistors’ logical operations that can be performed using the software. As an amplifier, it acts as a current booster. This means the input current is small, and the output current is much bigger. In other words, it’s a kind of current booster. It is also possible to use transistors as switches. Computer chips work in this way. When one part of a transistor is undergoing a very small electric current, another can experience a much larger current, which means the small current switches on the larger current. A memory chip, for instance, contains hundreds of millions or even billions of transistors, each of which can be switched on and off independently. Transistors can store either zero or one as they can have two distinct states as a switch. By having billions of transistors, a chip can store billions of zeros and ones which represent many characters. Having more transistors means having bigger memory.

This is a hierarchical structure. The current is passed horizontally and vertically between semiconductor layers to make up a computer’s logic operations and this builds to higher and higher levels as logical blocks are connected so different parts of the computer communicate with each other. Two individual transistors are getting connected at the smallest level, with bus lines and major sectors of the chips connecting further up the IC. A semiconductor's size, known as node size, is measured in nanometers. Its unit is equal to one billionth of a meter (or 0.000000001m). The current 'node' sizes of the semiconductor chips are 7 nm and 5 nm, which is the channel width at the lowest level of metallization, while the resolution of an SEM is about 0.6 nm at best. So, we are nearing the point where we cannot easily image the circuits with confidence and employ more advanced data processing techniques to develop a full process. In the following, the motivation behind this Ph.D. project is discussed.

1.1 Moore's Law

Moore's law is a computing term that originated by Intel CEO Gordon Moore in 1965; the simplified version of this law states that processor speeds or overall processing power of computers will double every two years. To break down the law further, it specifically stated that the number of transistors on an affordable CPU would double every two years (which is essentially the same thing), but measuring the increase by increasing the density of transistors is more accurate. This is not a physical law but rather a driving motivation to miniaturize and improve the chips' speed.

From 2000–2009, there was not much of a speed difference as the clock rates ranged from 1.3 GHz to 2.8 GHz, suggesting that the processor speeds have barely doubled within that 10-year timeframe. In 2000, the number of transistors in the CPU numbered 37.5 million, while in 2009, the number went up to an astounding 904 million; this is why it is more accurate to apply the law to transistor number/density than to speed. In 2020, Apple announced $M1$, the most powerful chip it has ever created and the first chip designed specifically for the Mac with 16 billion transistors. Three years later, in 2023, they announced $M2$ chips, consisting of 20 billion transistors, which is 25 percent more than $M1$.

As transistors in integrated circuits can be fabricated at smaller sizes, computers become smaller and faster. Chips and transistors are microscopic structures that usually contain metal, oxides, and silicon aligned perfectly to move electricity along the circuit faster. Smaller transistors mean less distance for electrical signals to travel. The faster a microchip processes electrical signals, the more efficient a computer becomes.

The word resolution refers to the smallest size that can be observed (the shortest distance between two neighbouring points). The resolution of the human eye is said to be 0.2mm. Depending on the instrument, the resolution of a scanning electron microscope can fall somewhere between less than 1nm and 20nm. Currently, transistors are around 5nm in scale and are expected to shrink in the next few years, while M1 and M2 are already 5nm. If one wanted to take an image of the layout of a 16 billion transistor CPU, SEM is not able to scan them all at once. So, at the transistor level, we are pushing the limits of the resolution of the SEM images. In this case, we need to acquire a high-resolution image (perhaps at $16k$ by $16k$ pixel dimension with 1nm pixel size), move the sample, acquire another image, move, and so on. Ultimately, these images need to be reassembled. These three steps are the main focus and motivation of this research. This leads to the following sections of this document, which are about SEM and image processing algorithms to tackle this issue of imaging semiconductors.

SEM details are discussed in chapter 2. This thesis contains several related projects in which we try to solve two important problems we found while working with semiconductor metallization samples. In chapter 3, We explain 2-dimensional stitching, which is assembling the composite image from several images, also known as tiling or mosaicing. This requires an algorithm for image stitching (inter-tile registration. The common registration approaches are also discussed in this chapter. We discuss and compare three main approaches: i) Fourier transform, ii) the sum of square difference, iii) and cross-correlation based on important metrics, including how they fit specific aspects of the experimental acquisition conditions. Automating this process is an essential issue as it is difficult and time-consuming to achieve manually. One of the famous landmark-based registration techniques is also discussed in this chapter. Our novel approach in 2D stitching is completely discussed in chapter 4. There are two methods, one based on a new set of landmarks which we named "Numerical Landmark", and one based on dimension reduction to accelerate the computation. Both methods increase the accuracy and efficiency of the stitching process.

Chapter 5 is on the alignment of stitched mosaics from layer to layer in three dimensions. In the stitching process, the tiles get stitched in 2D, and then the assembled images are needed to align in 3D to make a connectivity diagram of the target sample. This problem is known as 3D reconstruction, and it is not simply a 3D interpolation problem. Chapter 6 main discussion is on our novel approach in volume stitching. We introduce an algorithm that aligns consecutive tiles based on the centroid of specific features in SEM images.

There are two related projects, stitching in 2D and 3D, based on the various data sets we received from an industrial partner. We solved real-world applied problems to benefit both industry and academia. Our research questions were:

1. Is it possible to identify “numerical landmarks”, that is image regions which correlate to themselves much more strongly than they correlate with neighbouring regions, without having a predefined set of image features to match to? If so, how can they best be determined?

2. What approaches reduce the computational complexity of stitching? Can dividing the overlap area into smaller subimages lead to effective parallelization?

3. How can we align $2D$ stitched images from multiple layers without common features in order to reconstruct circuits?

Chapter 7 answers our research questions using the novel algorithms and analysis presented in this thesis.

All of the algorithms are implemented using Python 3.7, 3.8, and 3.10. There are multiple libraries which are used in our implementations, e.g. Numpy and Scipy are used for fitting and calculations, Skimage and OpenCV are used for calculation on images, Matplotlib is used for visualization, and Pickle is used for saving the data. In order to accelerate the “for” loops, another library is used, which is called “Numba”. Our data sets are big and include thousands of tiles. An automatic approach is needed to solve stitching problems for big data sets. This explains the high demand for computerized algorithms, which can accelerate the stitching process. The resolutions varied based on the imaging tool, but most of the microscope parameters to take those images were confidential and belonged to our industrial partner.

Chapter 2

Scanning Electron Microscopy

The human eye can resolve two points that are $0.2mm$ apart from each other. However, with the help of a lens or group of lenses (aided eye), this power could be modified to resolve much smaller features. Anything below $0.1 - 0.2mm$ requires some magnification. The microscope was developed to help human eyesight limitations and works as an efficient magnifying tool. That is an excellent aid for having a higher resolution to study and investigate the characteristics of a wide range of materials [1] since their properties are often dictated by features at the nano and micro scales.

The microscope resolving power is based on multiple factors, such as the quality and number of the lenses used and the probe source used to produce an image. According to [1], microscopy radiation wavelength can be divided into categories based on the source of radiation required to produce an image. Two are optical microscopy (OM) and electron microscopy (EM). OM uses light of different wavelengths (x-ray, visible light, infrared light) as a source, while EM uses accelerated electrons into a beam using electromagnetic lenses. Both microscopes are qualified for surface investigation of materials, but the optical microscope resolution is often limited by diffraction (or interference on the radiation waves). Using electron microscopy methods can solve this problem and help to attain a much higher resolution, though it too has some limitations, including requiring a high vacuum chamber to prevent scattering of the radiation.

Scanning electron microscopy (SEM) is one of the most common electron microscopy techniques. The SEM principle was originally demonstrated by Knoll (1935; Knoll and Theile 1939), with the first true SEM being developed by von Ardenne (1938). The modern commercial SEM emerged from extensive development in the 1950s and 1960s by Prof. Sir Charles Oatley and his many students at the University of Cambridge (Oatley 1972) [19]. This is a multipurpose instrument that can examine and analyze materials with high resolution.

In SEM, an electron gun produces an electron beam in EM. The image is created by the interaction of the primary electron beam focused on the material surface. The beam scans the surface of the sample with the subsequently scattered secondary and backscattered electrons gathered by electron detectors in a timed acquisition. Accelerated electrons used in EM have a very short wavelength ($0.859 - 0.037$Å, $20eV - 100keV$). So, you can see microscopic features, although the resolution is not simply determined by the wavelength (the quality of the lens and source, and the spatial extent of electron-sample interactions are also important). EM also gives information about composition as the electron beam can interact with inner-shell electrons, causing energy transitions in the atom that can act as a fingerprint of the chemical bonds within the sample. It also gives crystallography information because its wavelength is on the order of the lattice spacings in crystals, giving rise to Bragg diffraction.

SEM can provide information about a sample surface's features, including their shape, size, and arrangement. Materials need to be electrically conductive to be imaged in the SEM; otherwise, they experience electron charging, causing sample damage and confusion in image interpretation [16]. Non-conductive materials must be coated with a conductive layer of metal or carbon. If the specimen is observed without a conductive coating, it can be electrically charged because, at certain locations on the specimen surface, the number of electrons emitted is larger (or smaller) than the amount of incident (primary) electrons. The electron beam scans over the sample's surface, and the scattered electron detectors register a grayscale value as a function of position, see Figure 2.1. This gives a pixel grayscale value that corresponds to a specific scattering mechanism and electron yield. Commonly, with a scanned electron beam with a particular raster pattern (in this case, the beam flies back to the origin as it initiates a new row while the beam is blanked), X-Y information about the specimen can be collected. In the next three sections, we discussed two major signals of our interest in this project.

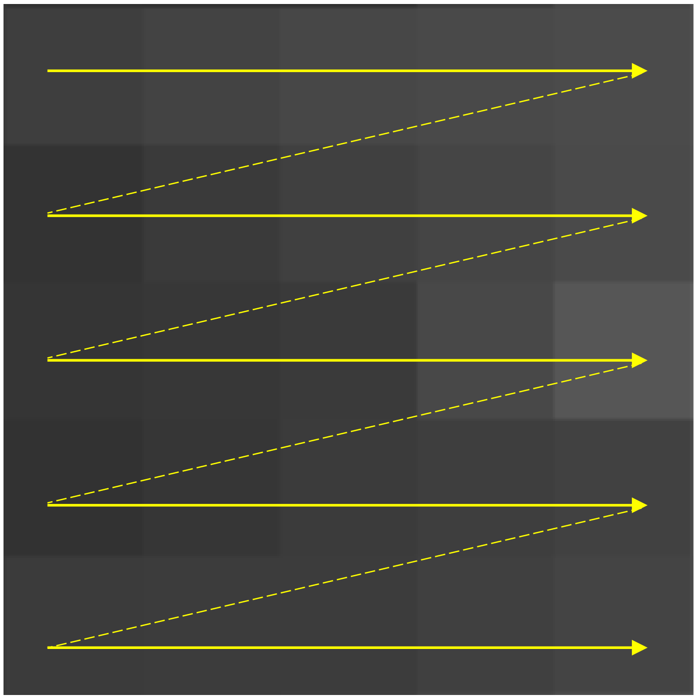

Figure 2.1: Scan path and pixel set in an SEM [2]. In raster scanning, the electron beam sweeps horizontally left-to-right at a steady rate (solid yellow line), then blanks (i.e. the beam is blocked while the scanning generating electronics return the beam to the start of the next row) and rapidly moves back to the left (dashed yellow line), where it turns back on (unblanks) and sweeps out the next line. The sample emits different kinds of electrons that are detected. Each pixel of the image gets a unique grey value based on the number of detected electrons and types. This image is a crop of an 8k (i.e. 8192 pixel) image. The pixel dimension is approximately 0.6nm, which is at the limit of the resolution of the instrument.

2.1 Electron-Specimen Interactions

SEM images are formed based on signals which are produced by the incident electron beam and its interactions with specimen atoms. Generally, there are two kinds of interactions: elastic interactions and inelastic interactions.

1. **Elastic scattering** results from the deflection of the incident electron by the

nucleus of a specimen atom. This interaction is characterized by negligible energy loss during the collision and a wide-angle directional change of the scattered electron. Backscattered Electrons (BSE) are produced by the incident electrons elastically scattered through an angle of more than 90 degrees, see Figure 2.2. BSEs produce a useful signal for imaging the sample. High atomic number (Z) nuclei scatter BSE electrons more strongly and result in a higher electron yield collected in the detector. This is the origin of the atomic number contrast (or Z-contrast), which shows that heavier elements appear brighter in a BSE image. BSE emitted with negligible energy loss but a wide angle of scattering are capable of providing information about complex properties of the target bulk down to a certain subsurface depth, revealing its material composition and crystalline structure. Contrast mechanisms reflecting these contributions to micrographs are combined according to the energy of incident electrons and the energy and angular acceptance of BSE detection.

2. **Inelastic scattering** occurs through a variety of interactions between the incident electrons and the electrons and atoms of the sample, which results in the primary beam electron transferring energy to that atom. The amount of energy loss depends on whether the specimen electrons are excited singly (secondary electrons), in multiple-scattering processes (x-ray, Auger), or collectively (plasmons) and on the electron's binding energy to the atom. Secondary Electrons (SE) are produced due to the excitation of specimen electrons during the ionization of specimen atoms. SEs are conventionally defined as possessing less than $50eV$, but the vast majority are less than $10eV$ in energy. Their information depth, about 0.5–$1.5nm$ for metals, is almost independent of the landing energy

of exciting electrons because its limitation is based on the low energy of excited electrons and their corresponding range [36]. Newly-generated secondary electrons have additional elastic and inelastic scattering events, often losing their energy further, see Figure 2.2. However, if they originate close to the sample surface, some secondary electrons may be emitted into the vacuum of the microscope chamber, and then travel to the detector. Because of the reabsorption mechanism of deeply-generated secondary electrons, only surface SEs tend to be measured, giving the SE signal a topographical contrast illustrative of the sample surface, which reflects the surface properties of the sample.

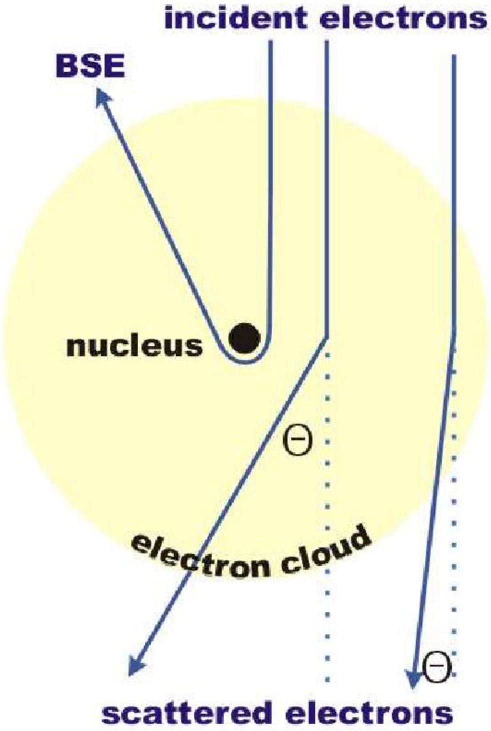

Figure 2.2: Scattering of an electron inside the electron cloud of an atom. This image is licensed under properties of Electrons, their Interactions with Matter and Applications in Electron Microscopy by Frank Krumeich (https://www.microscopy.ethz.ch/downloads/Interactions.pdf)

In addition to those signals utilised to form an image, several other inelastic signals are produced when an electron beam strikes a sample, including the generation of plasmons (oscillations of the electric field of the electron cloud) or phonons (manifested as heat - the collective oscillations of the nuclei themselves). Other useful signals include the emission of characteristic x-rays and Auger electrons, which give compositional information related to inner or outer-shell electrons (and can be mapped to give spatial chemical information as well). When the sample is bombarded by the electron beam of the SEM, electrons are ejected from the atoms comprising the surface of the sample. A resulting electron vacancy is filled by an electron from a higher shell, and an X-ray is emitted to balance the energy difference between the two electrons or an Auger electron transition (beyond the scope of this work) can take place giving surface information about the sample. Figure 2.3 shows the regions from which different signals are detected [56].

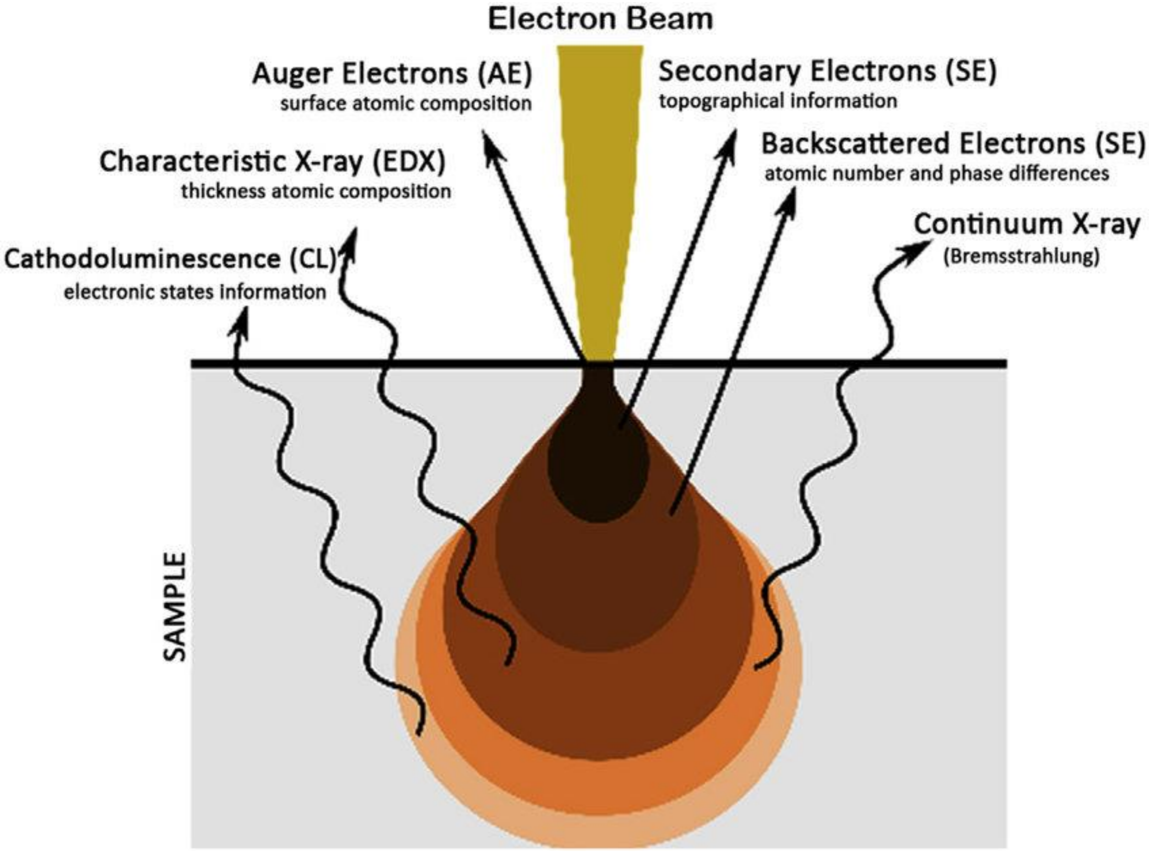
Electron Beam
Auger Electrons (AE)
surface atomic composition
Secondary Electrons (SE)
topographical information
Characteristic X-ray (EDX)
thickness atomic composition
Backscattered Electrons (SE)
atomic number and phase differences
Continuum X-ray
(Bremsstrahlung)
Cathodoluminescence (CL)
electronic states information
SAMPLE

Figure 2.3: Signal generation and detection in SEM. This image is licensed under the Creative Commons Attribution-Share Alike 4.0 International by user Rob Hurt

A direct consequence of the decrease in beam energy, E_0 is the correlated decrease in the penetration depth, R, of primary electrons (PE) interacting with the specimen. The basis of the combined directional elastic and inelastic scattering along the direction of the electron beam path can be described in simple analytical term proposed by the Bethe equation in power law [6]. The penetration of incident electrons decreases with increasing atomic number (Z). The penetration depth of incident electrons can be given by the Kanaya-Okayama Formula [24],

$$R = \frac{0.0276 A E_0^n}{z^{0.89} \rho} \mu m \tag{2.1.1}$$

where, R is the depth penetration, A is the atomic weight ($g/mole$), n is a constant, E_0 is the beam energy (KV), Z is the atomic number, and ρ is the density $(g/cm)^2$. "n" is often chosen to be ≈ 1.35 when the primary beam energy E_0 is $< 5keV$, and to be 1.67 when $E_0 > 5keV$.

The mean free path length of secondary electrons varies in different materials [46]. If the mean free path length is considered to be $10\mathring{A}$, although electrons are generated throughout the region excited by the incident beam, only those electrons that originate less than $10\mathring{A}$ deep in the sample escape to be detected. This volume of production is very small compared with BSE and X-rays. Therefore, the resolution using SE is better than either of these and is effectively the same as the electron beam size. The generation region of backscattered electrons is larger than that of the secondary electrons, namely, several tens of nm; therefore, backscattered electrons give poorer special resolution than secondary.

The specimen-electron interaction starts when the incident electron strikes the specimen surface. In this stage, the energetic electrons penetrate the sample for some distance and collide with a specimen atom. This happens in the region of primary excitation, from which a variety of signals are produced. This zone's size and shape are largely dependent upon the beam's electron energy, the specimen's atomic number, and the specimen's density. Figure 2.4 illustrates the variation of interaction volume with respect to different accelerating voltages and atomic number [56]. There is a direct relationship between volume and depth of penetration, and the beam energy, while this relation is inverse with specimen atomic number. Specimens with higher atomic numbers have increased nuclear scattering, so the higher the atomic number, the less penetration occurs. Note that high accelerating voltages will result in deep penetration and a large primary excitation region. Ultimately this causes the loss of detailed surface information, and the high voltage might degrade the specimen. This suggests that in composite materials like semiconductor metallization lines (see below), there is the possibility that sample information comes from different depths, causing some blurring and confusion about the exact feature location.

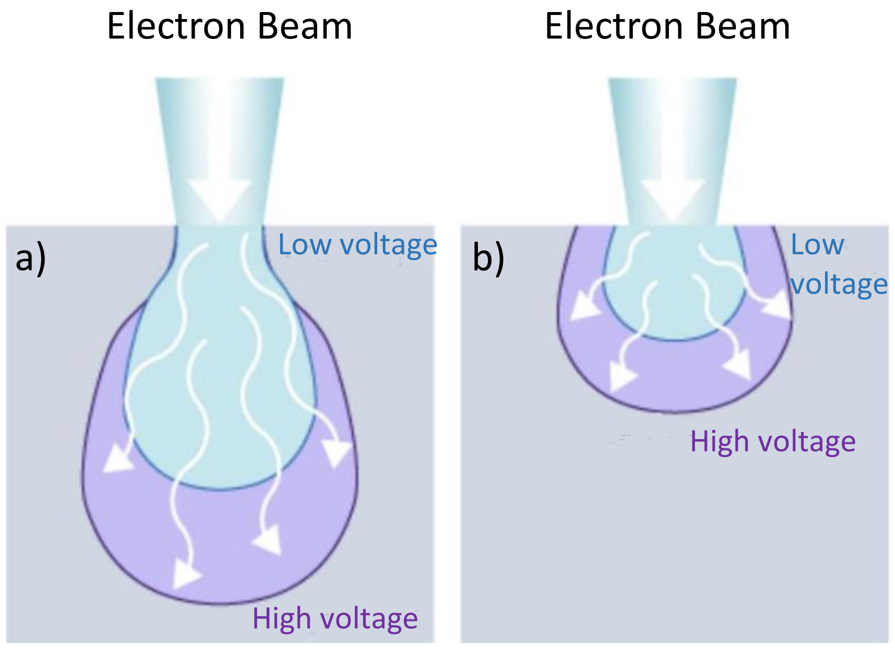

Figure 2.4: Influence of accelerating voltage and specimen atomic number on the primary excitation volume: (a) low atomic number and (b) high atomic number. This suggests that blurring is not linear, because the volume producing SE electrons will be different in different materials. This image is from (http://www.ammrf.org.au/myscope/sem/ptactice/principles/voltagevstype.php)

2.2 Secondary Electrons

When the primary beam strikes the sample surface and causes the ejection of valence electrons of low energy, these weakly-bonded electrons emitted also lose energy in interactions with adjacent atoms and are often reabsorbed into the sample. However, close to the sample surface, they can overcome a potential surface barrier and get emitted into the sample chamber's vacuum, thus becoming collectible. These escaped electrons are referred to as secondary electrons. As they have low energy, typically an average of around 3 to 5 eV, they can only escape from a region within a few nanometers of the material surface. So secondary electrons accurately mark the position of the surface atoms and give topographic information with good resolution. Because of their low energy, secondary electrons are readily attracted to a detector carrying some applied bias. The Everhart-Thornley (ET) detector, which is the standard collector for secondary electrons in most SEMs, it is designed to apply a bias $(+50 - 300V)$ to a Faraday cage that screens the detectors and sweeps up the electrons. Once inside the detector, the scintillator converts the energy of the electrons into photons (visible light). The photons then move out through the semiconductor within the detector, and the quantized energy of the photons is converted back into a measurable current that corresponds to grayscale for that pixel [56], [19] which corresponds to the number of electrons emitted from the location.

Secondary electrons are used principally for topographic contrast in the SEM, i.e., for the visualisation of surface texture and roughness. The topographical image is dependent on how many of the secondary electrons reach the detector. A secondary electron signal can resolve surface structures down to the order of $1nm$ or better. Secondary electrons that are prevented from reaching the detector will generate shadows or be darker in contrast to those regions that have an unobstructed electron path to the detector. When the specimen surface is perpendicular to the beam, the zone from which secondary electrons are emitted is smaller than found when the surface is tilted [56], [19]. The contrast is dominated by the so-called edge effect, as more secondary electrons can leave the sample on sloped than in flat areas leading to increased brightness spikes at the edges of the object. This makes topography in the sample from SE contrast intuitively easy to interpret by the microscope operator.

SEs can be generated from interactions with the primary incident electron beam, in which case they are called SE1 electrons. However, they can also be produced by backscatter electrons travelling within the sample, producing SE2 electrons as the backscattered electrons leave the sample or hitting the column or chamber wall, producing SE3 electrons, See Figure 2.5. SE2 and SE3 degrade the image's resolution because the signal emitted does not correspond to the input probe electron's position but is still gathered by the ET detector (see below) and contributes to noise in the image since its signal is delocalized.

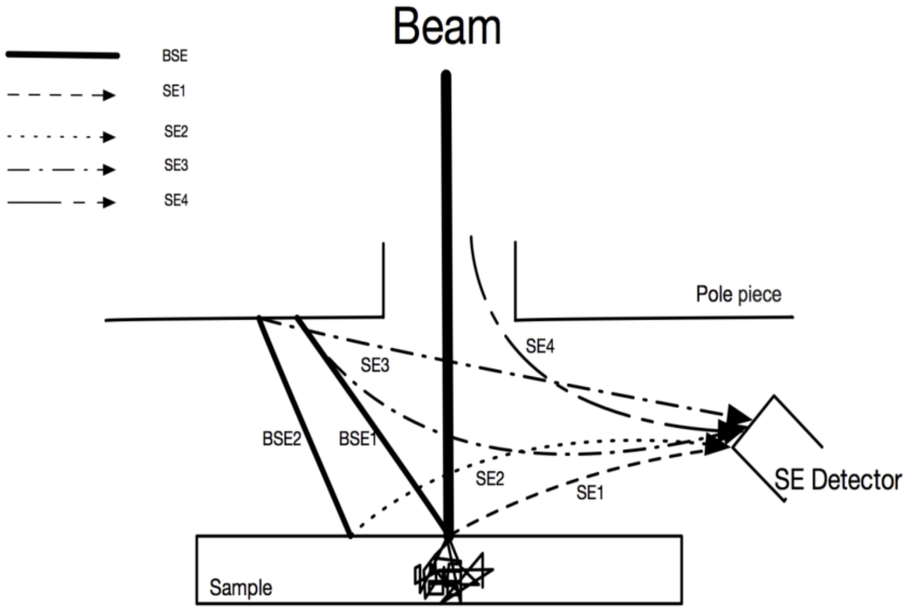

Figure 2.5: Different types of SE generated in the SEM chamber [43].

There are different detectors for absorbing electrons. For an Everhart-Thornley detector, a Faraday cage which is biased, absorbs the electrons. The bias can be either positive or negative. It is mostly positive and attracts low-energy SEs. If we put a slight negative bias, it repels any secondary electrons from reaching the detector, making an ET a poor backscatter detector which absorbs the high-energy BSEs. Only those electrons that travel in a straight path from the specimen to the detector contribute to the backscattered image [56]. Backscattered electron detectors can be of several types (scintillators, semiconductors) and are often placed above the specimen (with a donut hole in the middle to allow the incident beam to pass through). We will mostly use backscattered electrons throughout this dissertation, but the use of SE's is possible for certain imaging scenarios.

2.3 Backscattered Electrons

Backscattered electrons (BSEs) provide both compositional and topographic information in the SEM. A BSE can have single or multiple scattering events and escape from the surface with energy in the kV range. The elastic collision between an electron and the specimen atomic nucleus causes the electron to scatter with wide-angle directional change. Roughly 60 to 80 percent of these electrons' initial energies will be retained by the time the electron is reemitted into the vacuum chamber. Typically, the backscattered electron emission follows a cosine distribution, with the highest proportion of backscattered electrons scattered towards the electron source.

Elements with higher atomic numbers have more positive charges (protons) in the nucleus, and as a result, scatter the incoming electrons more strongly, increasing the backscattered electron yield. This, in turn, gives the atomic number effect, with the resulting backscattered signal with atomic number. Thus, the amount of BSEs depends on the sample's atomic number, providing contrast in SEM images. For example, the BSE yield from a light element, such as carbon, is much lower than a more massive element, such as tungsten or gold.

Nevertheless, with a reasonably large escape depth ($10s$ of nm), BSEs can carry information about features that are deep beneath the surface. In examining relatively flat samples, one can deconvolve topography from composition, yielding an image that contains mostly compositional information.

2.4 Configuration of Scanning Electron Microscopes

Figure 2.6 shows a conventional SEM [56]. We now give a brief explanation of where the electrons are targeted on the sample and how the electron beam is generated and demagnified to a focused point on the sample. An electron gun and electron lenses perform these two processes. Other parts of the SEM are explained in detail in [56], [19].

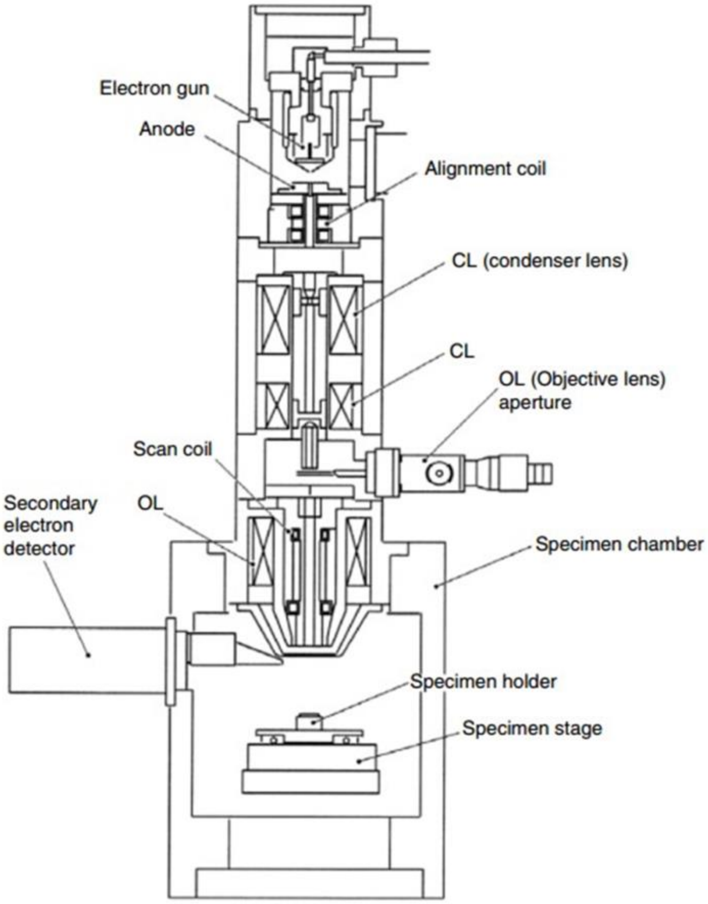

Figure 2.6: Schematic diagram of a scanning electron microscope (JSM-5410, courtesy of JEOL, USA).

2.4.1 Electron Gun

SEM systems require an electron gun that produces a stable electron beam with high current, small spot size, adjustable energy, and small energy dispersion. Electrons are produced by thermionic emission (i.e., heating a filament) or using the field-effect (putting a high extractive electric field on the electron source and overcoming the source metal's work function.). Electrons are produced by an electron gun, but their trajectory and beam characteristics need to be shaped by electron lenses which control their direction.

2.4.2 Electron Lenses

An electrostatic or electromagnetic field can focus an electron beam. Coils of wire, known as electromagnets, are used to produce a magnetic field, and the trajectories of the electrons can be adjusted by the current applied to these coils. Electron lenses can be used to demagnify the electron beam and control the convergence angle. The strength of electron lenses is variable, which results in a variable focal length. SEM always uses electromagnetic lenses to de-magnify the emission source's image to form a narrow probe on the specimen's surface. This leads to better resolution. The following are two kinds of lenses we have in SEM. Apertures are used at different lens planes to filter out aberrations of electrons that are far from the optic axis, at the expense of losing beam current.

1. **Condenser Lenses:** The electron beam will diverge after passing through the anode plate (a plate that imposes positive charges) from the emission source. By using the condenser lens, the electron beam is converged and collimated into a relatively parallel stream.

2. **Objective Lenses:** The electron beam will diverge below the condenser aperture. Objective lenses are used to focus the electron beam into a probe point at the specimen surface and supply further demagnification.

2.4.3 Scanning Coils

As mentioned in the previous sections, the electron beam excites different signals for SEM observation. The probe spot must be moved from place to place by a scanning system to form an image, akin to an old-style cathode-ray tube television. A typical image formation system in the SEM is shown in Figure 2.7 [56]. Scanning coils deflect the electron beam to scan the specimen surface, typically along the x- or y-axis. Several detectors are used to detect different signals, like solid-state BSE detectors for BSEs, the ET detector for SEs and BSEs, and an energy-dispersive x-ray spectrometer. The beam path, time spent at each spot, etc., are synchronized with the detectors' timing to make a coherent image as a function of beam position. It should be noted that scan coils can be customized in their operation to change the beam path.

Magnification is given by the ratio of the edge dimensions of the specimen area to the corresponding scanning line on the specimen. A change of the scanning area's size, controlled by the scanning coils, will result in a change of the magnification. Resolution is the ability to tell that two very close objects are distinct. If we try to magnify too much, we will not be able to see any more detail because the microscope will reach the limit of its resolution, as dictated by the beam size. A typical modern SEM may have a resolving power of about $1nm$. For SEM, the typical maximum magnification is about 1 million x. By increasing the magnification, we decrease the scan sweep, but the number of pixels remains constant. Ultimately, the beam diameter limits the magnification. In other words, the resolution of the instrument can be pixel limited (i.e. the size of the sampling area taken in the timing system) or beam limited (i.e. limited by the size of the beam, to first approximation).

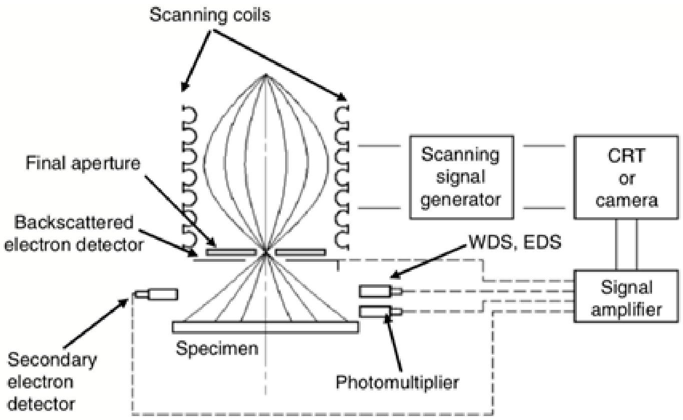

Figure 2.7: Image formation system in a typical scanning electron microscope [56].

2.5 Sample and Stage

At the bottom of the electron column, the specimen is located on the specimen stage. One can move the sample to investigate different parts of the surface with the help of the stage. The stage has motors that are capable of translation, rotation, and tilt to different orientations relative to the electron beam. The number of BSEs increases by tilting the sample stage towards the detector. The working distance in the SEM is the distance at which the beam is focussed, normally the distance from the final pole piece of the lens to the sample surface when the image is in focus. The polepiece concentrates the magnetic flux produced by the electromagnet by using soft magnetic materials like pure iron.

2.6 Measurement Analysis and Restrictions

The consistency and precision of measurements are a strong function of instrumental stability. Instability is defined as the variation in measurement results over an extended period, and stability is its inverse. To consider the instrument's stability, one needs to take measurements over an extended period or account for tool stability in an image processing model. Instrumental instability is related to thermal, acoustic, and electromagnetic variations in the instrument environment, or within the instrument itself. Noise is caused by many independent sources of error, but it is effectively modelled by adding independent, but identically, normally distributed errors to measurements. Instrumental instability can produce slowly-varying changes, and instead of being a component independent of the sample, it typically results in the sample being scanned differently than specified, either scanning too quickly, or too slowly, or with different spot sizes or voltages. It is difficult to model, and the models are unique to different imaging modalities.

Although modern manufacturing techniques have greatly improved the accuracy of SEMs, it is good practice to regularly check if the alignment of the SEM lenses and calibrated magnification remain stable. One should also take into account that magnification in the SEM changes with defined microscope working distance. Magnification and the scale bars are usually added by software, which relies on stored calibration files after the lenses have been aligned. For each current and voltage used in the microscope, the system is aligned to a calibration standard. The magnification error of an SEM can be in the range of $5-10$ percent. Calibration becomes of utmost importance if quantitative measurements are made on SEM images.

An appropriate calibration feature size must be used for calibrating at a particular magnification. The magnification calibration standards comprise features, mostly lines with a known distance (pitch). A calibration for $1000x$ magnification is not valid for $100x$ or $10,000x$ magnification.

One significant problem that can affect instrument stability is stage drift. Scanning electron micrographs at high magnification ($100,000x$ and up) are distorted by the sample's motion during image acquisition, called drift. A typical distortion source is thermal drift, see Figure 2.8. resulting from the slow thermal expansion of different materials in the sample and microscope due to small temperature changes throughout a scan or series of scans. Other sources of drift include electromagnetic fields and acoustic vibrations in the surrounding environment.

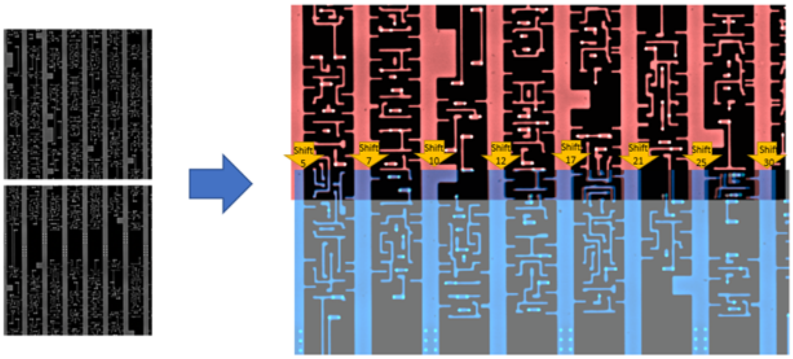

Figure 2.8: The rough estimation of thermal drift on the overlap area of two horizontally adjacent images are marked by orange arrows. Numbers in the Shift correspond to the shift in the number of x-pixels of the features required in order to align the feature. Note that the shift across the image varies in a non-linear manner with a shift of 5 pixels on the left required to bring the two images into the registry and 30 pixels on the right of the image. This precludes an alignment based on a global shift of one image onto the other.

This will affect one of the common approaches in imaging large samples with SEM, called montaging. A photomontage is a series of individual photographs taken and arranged together to create a single image. With this method, one can image large samples in high magnification and with high resolution. This could be considered as taking the sample's image as pieces of a high magnification puzzle and putting them together to have a seamless, high-resolution image of the sample's whole surface. This will be referred to going forward in our work as image stitching.

The time of the scan can be calculated considering different factors. dwell time means the time for an electron beam to scan one pixel, or by scanning continuously as the beam crosses the pixel area at a certain speed. The dwell time is an index for the scan speed of the electron beam. When the dwell time is multiplied by the number of scan pixels in one horizontal line scan, and then the flyback time is added, the scan time for one horizontal line scan is obtained. When this scan time is multiplied by the number of vertical scans (horizontal scan lines), the acquisition time of one scanning image is calculated. The probe drift distorts the features and lowers the spatial resolution when a long dwell time is used to increase the image signal-to-noise ratio (SNR). By definition, increasing signal means less noise relative to signal. The higher the number of pixels, the longer the total scan time.

2.7 Image Processing in Electron Microscopy

Getting an enhanced image or extracting useful information or features from images forms a core research area within engineering and computer science. These methods are known as image processing. It can be considered a type of signal processing, where the input is an image, and the output could be an image or characteristics/features associated with that image.

Image processing includes the following three steps:

1. Importing the image via image acquisition tools (like a camera or detectors associated with a microscope);
2. Analysing and manipulating the image;
3. Output in the form of an altered image or image analysis.

There are two types of methods used for image processing, namely, analog and digital image processing. Analog image processing can be used for hard copies like printouts and photographs. Analog signals are like television broadcasting in older days through the dish antenna systems, and in terms of images, they are manipulated by electrical signals. Digital image processing techniques help in the manipulation of digital images by using computers. The four general phases that all types of data have to undergo while using digital techniques are pre-processing, enhancement, display, and information extraction [8]. In this study, we are interested initially in image processing and finally in feature extraction in order to make connectivity for 3D reconstruction, see Figure 2.9.

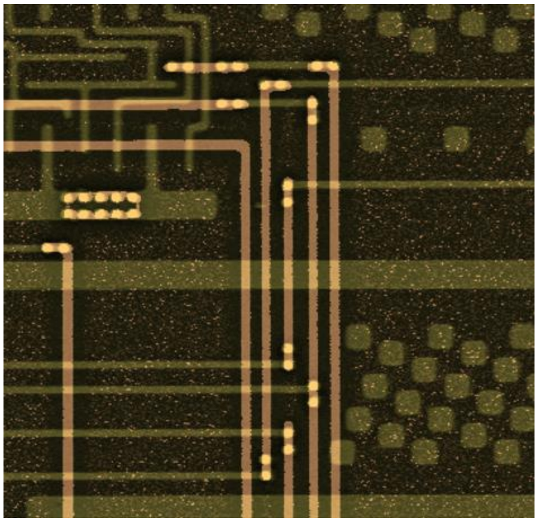

Figure 2.9: Sample alignment of two layers called M1 (bottom layer) and M2 (top layer), M1: light red, M2: Yellow, if there is a connection between the two layers in any pixel, the pixel color appears to be more bright.

Large fast computers are capable of manipulating big image matrices in a reasonable time. Images are stored as matrices in which each pixel has its own coordinate and value. The idea is that recorded electron microscope images can be improved or rendered more informative by digital image processing. Computer processing of electron microscopy images can help overcome the electron microscopist's problems, especially where radiation-sensitive specimens or instrument instabilities are concerned. Problems can include noise, jitter in the system, sample change due to radiation damage, and changes in the resolution of the acquired image.

Modern electron microscopes allow visualisation up to sub-nanometer resolutions.

Semiconductors seem to be very robust under the beam, but sometimes rapid scanning, changes in the room temperature, or limiting the number of the electrons in order to save the sample from any damage produces noise in semiconductor images as well.

In this chapter, the performance and components of SEM are discussed. The electron-specimen interactions and different types of electrons and their characteristic information are also explained. The importance of image processing in SEM images leads us to the following chapters, in which we will explain the critical problems in this field that image processing techniques could address.

Chapter 3

Stitching and Montaging Images in 2D

High-resolution images are essential in many scientific research areas, e.g. cancer research, biology, material science, and semiconductors. SEM is a tool to produce such images, capable of acquiring images at very high resolution (with pixel widths approximately equal to the diameter of the electron beam). Still, one of the drawbacks is the small field of view (FOV) of the acquired images[12]. Approaches such as MultiSEM, which splits out a primary electron beam using beam splitters into $60 - 90$ simultaneously operating beams [37] address this problem, but cost and data handling make this solution prohibitive in most cases. The most common approach to large-scale imaging at ultrahigh resolution is to divide the surface of the sample into multiple small regions with a degree of overlap, scan each region at high resolution and magnification, move the sample stage, acquire the next image until all regions have been imaged, and ultimately stitch them together using the overlaps. Stitching is an approach to creating a single SEM image. However, this approach has difficulties, especially in ultra-large-scale integration (ULSI) circuits, as the images may contain repeated patterns, distortions or nonlinear drift, leading to erroneous results [27].

Semiconductor metallization layers are among the most complex human-made structures and have many repeating patterns and features, making naive feature-based alignment challenging. Automating this process reliably is necessary as it is difficult and time-consuming to achieve manually. Circuits' SEM images are from two materials, metal and dielectric. In terms of components, they have three visible components of dielectric without metal, metal wires under a layer of dielectric, and metal vias (through-plane wires connecting different layers vertically - i.e., into the sample) [43]. This is a relatively simple set of contrast levels but is made challenging to interpret by repetitive patterns, measurement error, and the pixel-level precision required to connect nanoscale wires to reverse engineer a functioning integrated circuit. The tiles are created by acquiring an image, moving the sample with a certain degree of designated overlap, then re-assembling using a stitching algorithm. There are several challenges in stitching: (1) by design, the image features (wires, vias and dielectric) are highly repetitive, which are known as repeated patterns in SEM images of semiconductors. In the presence of such patterns, many equally likely matches between the overlap areas of two adjacent tiles are possible, and choosing the wrong one will result in global stitching errors; (2) the overlap between image tiles is small (generally around 5% overlap), which limits the search area for the best match and affects the accuracy of the computation [5]; (3) sample charging causes intensity variation between captures of the same region observed in the overlap region, (4) machine instability causes non-linear deformation within tiles, and imprecision of stage movements causes uncertainty in the motion between tiles. So, although planned stage movements are simple translations, we need to detect and correct skew, shear, rotation and scaling transformations as well.

Previous work on image stitching can be divided into two types of methods: 1. intensity-based registration and 2. feature-based registration, which registration is about maximizing similarities between images or minimizing differences in order to register or align them on each other. Although the first is relatively simple to implement by comparing corresponding pixel gray values, it is computationally expensive to compare millions of pixels against millions of adjacent pixels [32]. Feature-based methods establish correspondences between points, lines or other geometrical entities in overlapping image regions and lead to more accurate results at even higher computational costs. In 2013, the term keypoint was introduced in [30], which stands for image features. They investigate the problem in SEM image acquisition which is mostly affected by the time-varying motion of pixel positions in consecutive images, which is called drift. To perform accurate SEM measurements, it is necessary to compensate for this drift in advance. They developed a method to compensate for distorted images using keypoint correspondences and computing homography. This image registration-based drift compensation method is effective in correcting image distortions.

In the year 2003, Brown et al. [11] developed SIFT (Scale-Invariant Feature Transform), a stitching algorithm for producing panoramas. In contrast to previous methods, SIFT does not need human input or restrictions on the image sequence. Brown et al. used feature recognition techniques based on invariant local features to select matching images and a probabilistic model for verification, which is insensitive to the images' ordering, orientation, scale, illumination, and noise. In 2017, Chalfoun et al. [15] developed MIST (Microscopy Image Stitching Tool), which is a stitching tool for 2D grids of images. MIST estimates the stage mechanical model (actuator backlash, stage repeatability, etc.) from computing the translations for the image and its horizontal and vertical neighbours and then minimizes stitching errors by optimizing the translations within an area. This minimizes the maximum uncertainty related to the translation computation for any pair of images. MIST has several limitations that should be considered. This tool is to stitch microscopy images that had been acquired via a mechanical stage that moves the sample in a repeatable grid pattern, which prevents it from handling stitching problems with varying overlaps within the same acquisition. It is expected that the overlap error between images will be less than a pre-defined threshold (by default, it is set at ±3%), and any overlap value beyond that error is considered unreliable. MIST is designed for 2D time-lapse multichannel stitching and cannot perform volumetric (3D) stitching. High levels of image noise can impact stitching accuracy, so preprocessing may be necessary if the images being stitched are very noisy. These limitations prevented MIST from stitching the data sets used in this thesis, at least not with the parameters tried, including the default parameters.

In 2021, Pawlowicz et al. [34] accelerated this approach by using parallelization and cloud computing, identifying the computationally expensive step to compute correlations.

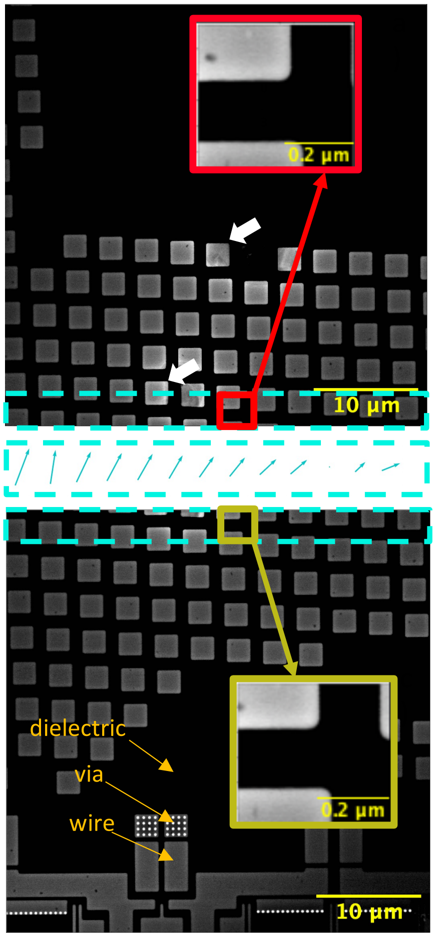

Figure 3.1: Two vertically adjacent 8k tiles with an approximately 8000 × 500-pixel overlapping seam (outlined with a dotted blue line). There are three visible contrasts: dielectric (dark), wire (grey) and via (bright), as labelled on the lower tile. During the scan, non-linear drift was noted. To estimate the drift, subimages of size 500 × 600 pixels were extracted from the seam; example subimages are outlined in red and yellow, with the lack of alignment being apparent in the inset 10× magnified versions. The translation resulting in the best match for each pair was calculated and displayed using one blue vector per subimage. The vectors are positioned in between the two tiles, and their horizontal position approximately matches the horizontal position of the subimage pair in the seam. It is also apparent from these tiles that the variation in brightness is significantly attributed to charging, as indicated by white arrows on the top tile. Since charging varies with time, this can cause changes in brightness in the overlap, which is a challenge for some registration methods.

Our goal in this thesis was to extend existing stitching algorithms to detect and correct non-linear transformations and to be robust in the presence of repeated patterns. Figure 3.1 shows the nonlinear pattern of the drift in the overlap area of the two tiles, which are adjacent vertically and have horizontal overlaps. The shift direction, which is shown in blue arrows, is calculated after dividing the overlap area into subimages and resolving the shift for each. It is obvious that the shift varies even for adjacent subimages. This figure also shows the three major features of the samples we use in our study. Although cloud computing makes it possible to stitch very large mosaics, adding to the complexity of the underlying image registration problem might make this uneconomic. Therefore, we evaluated the robustness and computational complexity of three core algorithms for stitching semiconductor metallization patterns: i) Fast Fourier transform (FFT) methods, ii) minimizing the sum of squared differences (SSD), iii) and maximizing the normalized cross-correlation (NCC). We evaluated standard implementations and two novel approaches. In the first approach, we introduce a concept named numerical landmarks. We take advantage of the Harris Corner algorithm [21] to mathematically detect landmarks in our images and apply NCC to find the maximum correlation between the overlap area of adjacent tiles and stitch them. The second one is in which the 2D images are projected by summing over columns (rows) to reduce the computational complexity of the optimization (in ii and iii) and phase-difference (in i) steps. In the course of these experiments, we discovered that "grain patterns" caused by localized variations in material properties, and modulated by filtering as part of the image acquisition, are more visible after projection and can be used to align images.

In the stitching process, some other visual problems, like differences in intensities of the two overlap sections, might appear. A blending algorithm is applied to eliminate such effects and improve the composite image's visual quality. The final image is often interpolated using computer software if the components of the images are not perfectly aligned. This is mostly the case where blending is to choose the final pixel value in a location where two images are overlapping [7]. As an example of a useful reference for understanding the concept of stitching problems and essential factors in this area, one can refer to [28], where the authors describe the application of Autostitch. Autostitch is software that is usually used for the generation of panoramas in photography and the seamless stitching of microscope images. As we are interested in SEM images, our focus is on SEM images of semiconductor metalization layers.

SEM imaging artifacts can be categorized into three classes according to [48], as it is important to distinguish the time scale of the distortions. (1) Random, time-dependent distortion due to positioning errors of the electron beam during scanning, referred to as "scan line shifts". (2) Non-random, time-independent spatial distortion, similar to distortions observed in optical systems. (3) Non-random, time-dependent distortion referred to as drift distortion. The time-varying or time-dependant distortion may be defined at each pixel location (x, y) by a time-dependent velocity function, $v(x, y; t)$. A schematic illustration of this can be found in Figure 3.2. The velocity vector is estimated by using digital image correlation to obtain the displacement of the current subset centred at this location several times, t. By integrating v over time, the drift displacement correction at each pixel can be calculated. The other two artifacts can be either time-independent or nonlinear drifts due to some changes in the temperature or situation of the microscope.

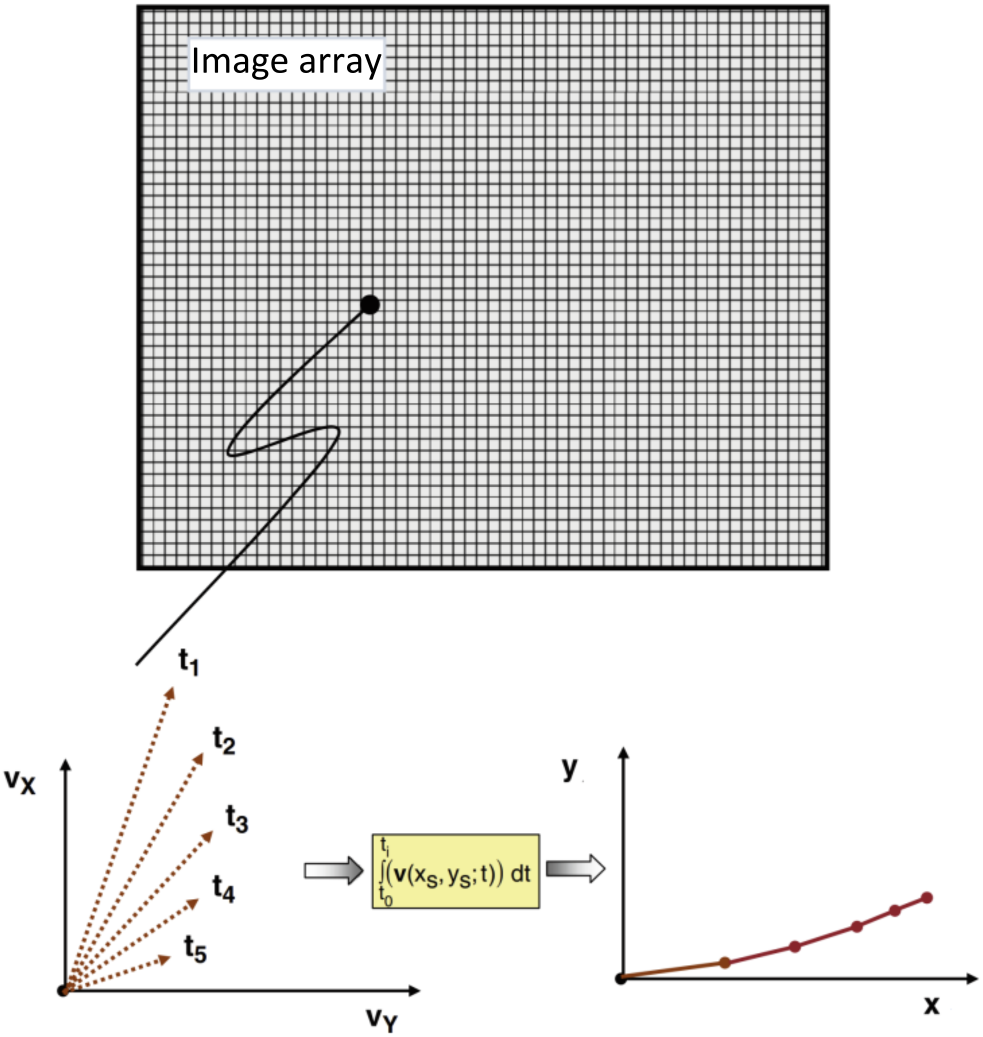

Figure 3.2: Schematic of pixel distortion that varies over time. SEM scan parameters (dwell time, beam repositioning time) and detector location are known and used to estimate the velocity vector. Image is licensed by [48].

Our research mostly focuses on resolving the distortion in SEM images during the stitching process, which is about registering two or more adjacent tiles on each other from their overlap area. For instance, two adjacent tiles are taken, and the two overlap areas will get compared. Then, the degree of shift can be resolved by minimizing the differences between the two images, matching the correlation, or locally measuring the difference in spatial frequency.

In this chapter, possible common approaches to dealing with mosaicing will be discussed. Resolving the shift in the overlap areas in order to stitch the tiles is one of the major problems in this topic; see Figure 3.3.

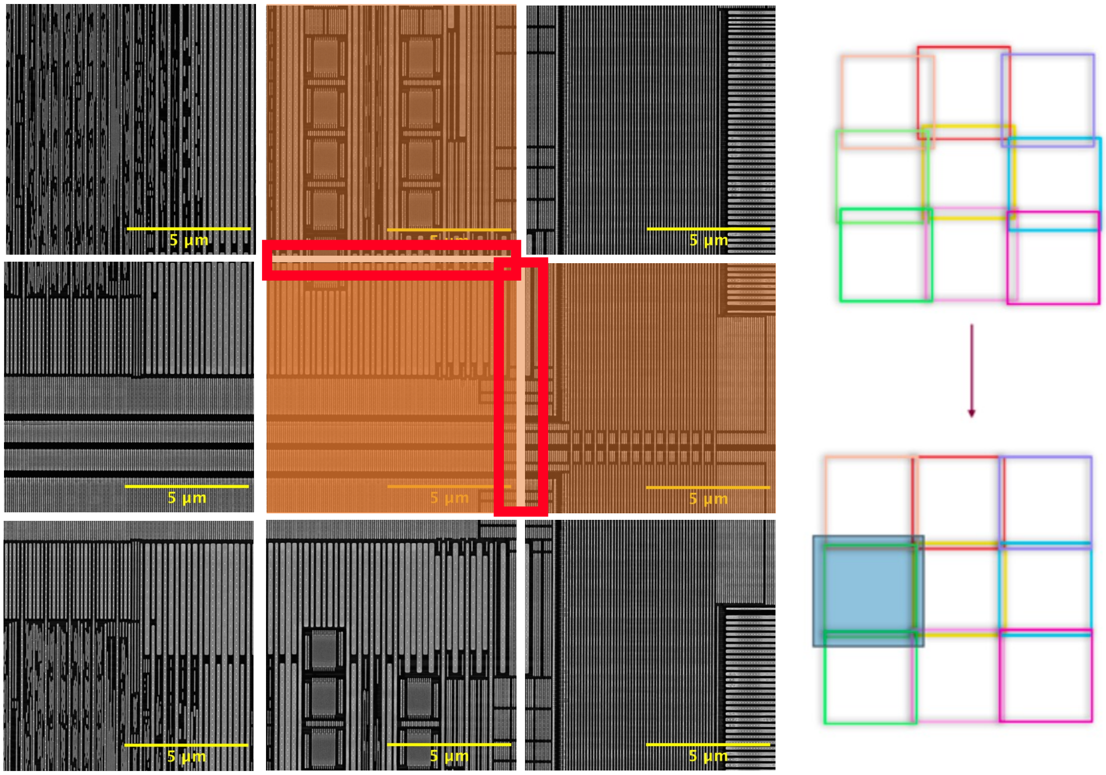

Figure 3.3: Resolving shift in matching the tiles. Each tile is slightly misaligned and requires an $X - Y$ translation to create a coherent mosaic. Horizontal and vertical overlaps of one tile where the shift needed to be resolved as a matter of stitching are marked by red rectangles.

As mentioned earlier, sometimes we encounter the measurement of complex displacement fields, and the specimen images might experience compression, shear, skew or rotation. In other words, an initially square reference subset might assume a considerably distorted shape in a later image after deformation. Our algorithms are focused on resolving the shift problem, and we believe that all other distortions can be solved in terms of shift.

There is a relation between shift and all other kinds of distortions, as $u(x,y)$ and $v(x,y)$, which are shifts in $x-$ and $y-$ directions, respectively, can be rewritten using other sorts of distortion, e.g. by using rotation and skew matrix. We can model distortions in x and y as a polynomial where $u(x,y)$ and $v(x,y)$ are distortion in x and y directions as a function of x and y, i.e.

$$x \to x + u(x,y) \tag{3.0.1}$$

$$y \to y + v(x,y). \tag{3.0.2}$$

This model depends on the order of the polynomials. If u and v are linear, we can represent pure compression, skew, rotation and all linear transformations. For example, rotation around $(0,0)$ by angle θ given by the rotation matrix

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{3.0.3}$$

is represented by the polynomials

$$u(x,y) = x(\cos\theta - 1) + y(\sin\theta) \tag{3.0.4}$$

$$v(x,y) = x(-\sin\theta) + y(\cos\theta - 1), \tag{3.0.5}$$

and the skew matrix

$$\mathbf{S} = \begin{bmatrix} 1 & a_{01} \\ 0 & 1 \end{bmatrix} \tag{3.0.6}$$

is given by polynomials:

$$u(x, y) = x + a_{01}y \tag{3.0.7}$$

$$v(x, y) = y. \tag{3.0.8}$$

Nonlinear polynomials can be thought of as interpolating between these basic forms of distortion in different image parts. This model for distortions can be defined as minimization and maximization problems to find the polynomials which best match neighbouring tiled images, I_T and $I_{T'}$. This can be solved using either SSD,

$$\min \Sigma_{x,y} ||I_T(x, y) - I_{T'}(x + u(x, y), y + v(x, y))||_{\ell_2}, \tag{3.0.9}$$

or by finding the maximum correlation,

$$\max \Sigma_{x,y} I_T(x, y) * I_{T'}(x + u(x, y), y + v(x, y)). \tag{3.0.10}$$

Solving this model is computationally expensive because our example images are mostly having 8192 × 8192 pixels, and in some cases, higher resolution makes matrix operations require holding many values in memory. Also, in such a big tile, many local extrema can happen due to repeated patterns in semiconductor SEM images (i.e. a series of parallel lines).

Instead of solving the problem for the whole overlap area at once, we can divide it into two stages. First, to divide the overlap area into smaller boxes and find the amount of the shift in each box, and second is to fit a global polynomial to the local shifts. As mentioned, the above models are for the whole image overlap, but they could be modified by dividing the overlap area into smaller boxes, B_i, as follows, which for SSD it is formulated as,

$$\min_{u,v} \Sigma_i \Sigma_{(x,y)\in B_i} ||I_T(x,y) - I_{T'}(x+u(x,y), y+v(x,y))||_{\ell_2}, \tag{3.0.11}$$

and for correlation, it is,

$$\max_{u,v} \Sigma_i \Sigma_{(x,y)\in B_i} I_T(x,y) * I_{T'}(x+u(x,y), y+v(x,y)). \tag{3.0.12}$$

As mentioned above, There is a relation between translation and all other kinds of distortions. We can model distortions in x and y as a vector-valued function on the plane. Most distortions can be approximated well by polynomials, i.e.,

$$\begin{pmatrix} x+u(x,y) \\ y+v(x,y) \end{pmatrix} = \begin{pmatrix} x + a_{00} + a_{10}x + a_{01}y + a_{11}xy + ... \\ y + b_{00} + b_{10}x + b_{01}y + b_{11}xy + ... \end{pmatrix} \tag{3.0.13}$$

where a_{00} is the overall shift in the x direction, b_{00} is the overall shift in y direction and other coefficients are different shift directions. The order of the polynomial is determined by the degree of non-linearity of the distortion. When we zoom in on a subimage, if we rescale coordinates to match the new image size, all of the coefficients a_{ij} are reduced, except for those of order zero. So for small enough subimages, the approximation is

$$\begin{pmatrix} x+u(x,y) \\ y+v(x,y) \end{pmatrix} \approx \begin{pmatrix} x+a_{00} \\ y+b_{00} \end{pmatrix}. \tag{3.0.14}$$

This justifies looking only for local coordinate shifts (rather than non-linear transformations) in subimages, knowing that it may fail for some subimages with localized non-linear distortions. Once we subdivide the overlap into subimages, and determine (x, y), as in approximation (3.0.14), we use Δ_i to represent shift in subimage $i \in \{1, 2, ...\}$ having center (x_i, y_i). We can estimate the polynomial by setting up an optimization problem which fits a polynomial to shift values by minimizing the difference as,

$$\min_{a_{00},b_{00},a_{01},...} \sum_i \|f(x_i, y_i) - \Delta_i\|, \tag{3.0.15}$$

where the norm $\|\cdot\|$ can be any norm, including the ℓ^1 (Manhattan distance) or ℓ^2 (squared Euclidean distance), see Appendix E. For computational simplicity, we consider the x and y shifts separately, resulting in one problem to fit the $a_{00}, a_{01}, ...$ and a separate problem for the $b_{00}, b_{01}, ...$ Due to nonlinear distortions, specific patterns in the overlap area (repeated patterns) or changes in the brightness, outliers in shift calculation might appear. Using the ℓ^1 norm, see Appendix E, is generally recommended in the presence of outliers [45], as it is less sensitive to them compared to other norms.

The original stitching problem is a global optimization which is hard and expensive to solve. It is preferable to solve convex problems, Appendix A, which have unique global extrema, but the proposed above models are not convex problems. Therefore maxima and minima may require an exhaustive search. Having discretization by breaking up the overlap into boxes and searching in the space of integral pixel shifts, one can find Δx_{B_i} and Δy_{B_i} using exhaustive search. This is done by shifting one pixel at a time and calculating the differences, storing the difference values in a matrix as a shift matrix, and setting the coordinate of the minimum element in the shift matrix as the shift values in $x-$ and $y-$ directions. See Appendix B for definitions of exhaustive search, Appendix C for coordinate search, and Appendix D for local extrema.

The objective function for finding the amount of the shift for each box using Cross Correlation is,

$$\max_{\Delta x_{B_i}, \Delta y_{B_i}} \Sigma_{(x,y)\in B_i}(I_T(x,y) * I_{T'}(x + \Delta x_{B_i}, y + \Delta y_{B_i}) \tag{3.0.16}$$

$$\forall i = 1, 2, 3, ..., N, \tag{3.0.17}$$

where N is the number of boxes and B_i is the selected box from the overlap and i is the index of each box. x_{B_i} and y_{B_i} are the coordinates of the boxes, and Δx_{B_i} and Δy_{B_i} are the amount of the shift in each direction, which is the information that we have about the position of the pixels and the amount of the shift. The goal is to interpolate the Δ values to determine a shift for each pixel across the whole image.

$$\min \Sigma_i w_{B_i} * (||(u(x_{B_i}, y_{B_i}) - \Delta x_{B_i}, v(x_{B_i}, y_{B_i}) - \Delta y_{B_i})||_{L_2}) \tag{3.0.18}$$

where w_{B_i} is a per-box weighting. In the course of this research, we will identify and compare different computations of the weighting.

The order of the polynomial in the interpolation problem is important. More coefficients mean overfitting, which can appear by fitting to the noise and not just the data. Less coefficients provide more confidence, but using only four coefficients gives us linear interpolation, but we know the nature of the drifts in our problem is nonlinear. In this regard, we are looking for more than two points (boxes) while we should take care of overfitting. In summary, the number of boxes should be bigger than the number of coefficients. The same method could be applied to SSD.

According to the above formulation, dividing one big maximization problem into smaller maximization problems followed by a minimization problem is computationally affordable and possible to solve. There are various kinds of norms we can use for modelling such a problem, Appendix E. ℓ^2 is useful in the presence of normal noise and ℓ^1 is insensitive to outliers, which is an important feature to consider as there might be some match in some parts of the image which are wrong. ℓ^2 norm squares the error, so the model will see a much larger error than the ℓ^1 norm, see Appendix E. In the presence of an outlier, the ℓ^2 will be adjusted to minimize this single outlier case, at the expense of many other common examples, since the errors of these common examples are small compared to that single outlier case, see Figure 3.4.

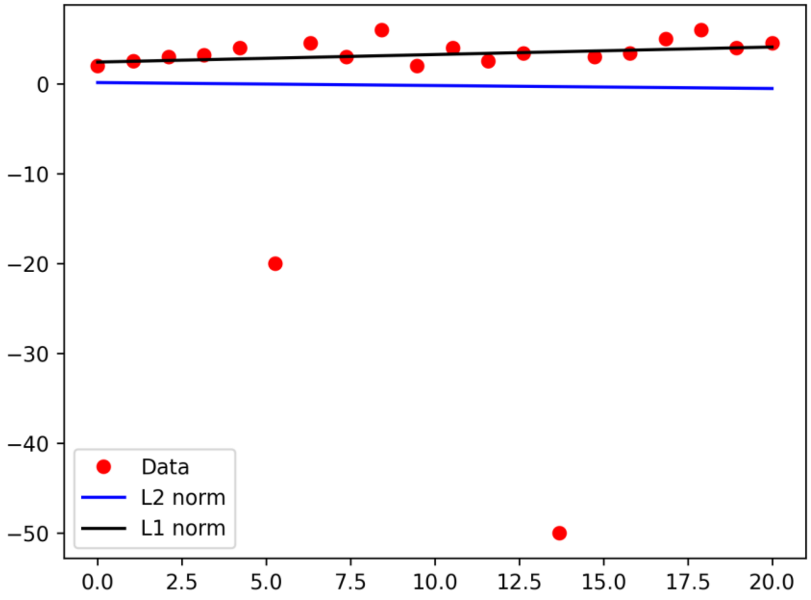

Figure 3.4: ℓ^2-norm squares the error, so the model will see a much larger error than the ℓ^1-norm. The model is much more sensitive to this example and adjusts the model to minimize this error.

In our image processing problem, outliers can happen when we get similar matches based on similar features (like vias) in different parts of the image. There are some other factors like skew and carbon deposit from the electron beam scanning process, which can change the intensities, and can produce outliers. The other important factor that can produce an outlier lies in the type of the shift.

There are two kinds of shifts in the mosaicing of semiconductor metallization layers for microelectronics applications:

1. **Translation**, in which one slides one tile on the overlap area of the other tile with prior knowledge of the intended stage movement from tile to tile in order

to resolve the shift and stitch the images from the overlap area;

2. **Instrument drift**, which is usually a change in the imaging conditions related to a change in temperature in the laboratory.

Translation resolves the linear shift between two adjacent tiles that will be stitched from their overlap area. Instrument drift is more general and may not be rigid and linear. In our experiments, instrument drift was mainly nonlinear (skew, shear or scale).

To be able to give mathematical solutions to image processing problems, we first need to know what an image is in a mathematical concept [32].

Definition Let $d \in N$, a function $b : R^d \to R$ is called a d-dimensional image, if

1. b is compactly supported,
2. $0 \leq b(x) < \infty$ for all $x \in R^d$,
3. $\int_R^d b(x)^k dx$ is finite, for $k > 0$.

The set of all images is denoted by

$Img(d) \in \{b : R^d \to R|b\}$ is a d-dimension $images(Img(d))$.

In the following sections, we will describe mathematical tools that may be used to accurately determine the shift between overlapping adjacent images. Each has advantages and drawbacks and is studied for subtly different aspects of stitching montages.

3.1 Fast Fourier Transform

Any wave, like sound waves, indeed any repeating function, can be broken up into a number of sine waves of various frequencies and amplitudes (intensities). The expression of a sound wave, or any signal varying over time, as the sum of its constituent sine waves, is known as the fast fourier transform of that signal.

The fast fourier transform is defined as

$$(1D)FFT(f) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi tf}dt, \tag{3.1.1}$$

$$(2D)FFT(k_x, k_y) = \int_{R^2} g(x, y)e^{-i2\pi(k_x*x+k_y*y)}dxdy, \tag{3.1.2}$$

where g is an input signal or image. Many applications involve sampling over time (t). With the help of the fast fourier transform, we can do calculations that seemed difficult in the time domain, simply in the frequency domain (f). These techniques were developed for time-series but can be applied to images in 2D with spatial frequencies. Images can also be expressed as a sum of sine waves, but, instead of one-dimensional waves, they are waves that vary in two-dimensions, like ripples on a sheet. Two-dimensional sine waves are written as

$$z = a\sin(k_x * x + k_y * y) \tag{3.1.3}$$

where x and y give the coordinates for points on the "sheet", z is the height, or intensity, of the wave at that point, a gives the amplitude (the maximum height) of the wave, and k_x and k_y give the number of times the wave repeats in the x and y

directions respectively (they are the x and y frequencies). The following theorem can be used to find the amount of shift in a delayed (shifted) signal. According to the shift theorem, a delay in the time domain corresponds to a linear phase term in the frequency domain.

Theorem (Shift Theorem) [26] Let $g : R \rightarrow R$ be a function which has the fast fourier transform $G : R \rightarrow R$, then $g'(t) = g(t - \Delta)$ has the fast fourier transform $G'(k) = e^{-2\pi i \Delta k} G(k)$.

This theorem says that a shift in the image domain, Δ, becomes a linear phase term, $e^{-2\pi i \Delta k}$, in the frequency domain, where k is the coordinate in frequency space. Given G and G', for any k, we can calculate

$$\Delta = \frac{\arctan\left(G'(k)/G(k)\right)}{k} = \frac{\arctan\left(G'(k)\overline{G(k)}\right)}{k} \tag{3.1.4}$$

Using complex conjugate, $\overline{G(k)}$, eliminates a possible division by zero, but this formula still has a non-linear sensitivity to noise and Gibbs' ringing [17]. There are different ways of averaging over k, and pre-filtering the subimages. We used a Lanczos window which is the central lobe of the *sinc* function [29], and reduces the artifacts. Mathematically, outside of the window interval has a zero value when a wave is multiplied by a window function. So, all that is left is the part where they overlap, which is called the view through the window. The isolated segment of the data within the window is multiplied by the window function values called tapering. In particular, this reduces ringing artifacts caused by abrupt changes in the sample values. The NumPy library function *atan2* in Python calculates the angle we need from the real and imaginary parts. In this way, one can calculate the number of times the phase changes by summing the difference pixel-to-pixel or by taking *atan2* to convert the complex number into an angle and divide the difference by the distance from 0. For the periodic structure case, one could probably get a better answer by finding the peak away from 0 and using *atan* in $2D$.

The cells of k-space are commonly displayed on rectangular grids with principal axes k_x and k_y. The k_x and k_y axes of k-space correspond to the x- (horizontal) and y-axes (vertical) of the image. The k-axes, however, represent spatial frequencies in the x- and y-directions rather than positions. The individual points (k_x, k_y) in k-space do not correspond one-to-one with individual pixels (x, y) in the image. Each k-space point contains spatial frequency and phase information about every pixel in the final image. Conversely, each pixel in the image maps to every point in k-space. In particular, the row $k_y = 0$ is the 1D fast fourier transform of the projection onto the x axis. This was our motivation to use projection in our novel stitching approach, which also contains information about the whole image.

3.2 Sum of Squared Difference

A straightforward approach to deal with image stitching is based on minimizing the so-called sum of squared differences (SSD) between two images within the overlapping region [32].

Theorem Let I, I' be d-dimensional arrays of gray-valued pixels. The SSD distance is defined by

$$D_{\mathrm{SSD}}(I, I') = \frac{1}{2}||I - I'||^2_{\ell^2}. \tag{3.2.1}$$

The best SSD alignment is found by solving the optimization problem

$$\min_{a_{00},b_{00}} \Sigma_{k,l}||I(k,l) - I'(k+a_{00}, l+b_{00})||_{\ell^2} \tag{3.2.2}$$

where the subimages I and I' come from the overlap region of adjacent tiles. In practice, we only calculate the SSD difference for integral shifts, but the optimization problem can be defined for fractional shifts, and for nonlinear transformations by using interpolation to turn the pixel arrays into real functions. As I' shifts relative to I, the objective function would be sensitive to bright pixels shifting out of one side while darker pixels shift in the other, etc. It is discussed in [22] that there are different methods for calculating the similarity between images, but SSD is the most straightforward and requires the least amount of computation since it only involves a square operation and subtraction of pixels between the template and the original images. It is also mentioned that is sensitive to changes in illumination.

3.3 Normalized Cross Correlation

In statistics, correlation usually refers to a linear relationship between variables. Pearson's correlation coefficient [41] for two variables is calculated by multiplying them after subtracting their means, summing over the samples and dividing by the product of their standard deviations. In this case, we are looking for the linear relationships between the pixels at corresponding locations in two images. The best alignment is when this correlation is strongest. Due to [22], NCC has more complex calculations than other measures of finding similarities like SSD, as it involves multiplication, division and square root operation. It should be considered that it is more robust compared to SSD in illumination changes.

The expected value or mean of X as a discrete random variable is a weighted average of the possible values that X can take. Each weight is according to the probability of that event occurring. The expected value of X is usually written as $E(X)$ or m.

$E(X) = \sum$(each possible outcomes) × (the probability of the outcome occurring).

The variance of a random variable tells us something about the spread of the variable's possible values. For a discrete random variable X and mean m, the variance of X is written as Var(X).

$$Var(X) = E(X - m)^2, \tag{3.3.1}$$

$$Var(X) = E(X^2) - m^2, \tag{3.3.2}$$

$$Var(X) = E(X^2) - E(X)^2. \tag{3.3.3}$$

In Mathematics, squaring is the same as multiplying by itself. So the variance equation can be rewritten as,

$$Var(X) = E(XX) - E(X)E(X). \tag{3.3.4}$$

One of the Xs can be substituted by another random variable Y,

$$Var(XY) = E(XY) - E(X)E(Y). \tag{3.3.5}$$

If the variance measures how a random variable varies with itself, then covariance measures how one variable varies with another.

Covariance measures the joint variability of two random variables in probability theory and statistics [20]. If the higher values of one variable mainly correspond with the other variable's higher values, and the same holds for the lesser values (i.e., the variables tend to show similar behaviour), the covariance is positive. In the opposite case, when the higher values of one variable mainly correspond to the other's lesser values (i.e., the variables tend to show the opposite behaviour), the covariance is negative.

The sign of the covariance, therefore, shows the tendency in the linear relationship between the variables. The covariance's magnitude is not easy to interpret because it is not normalized and depends on the variables' magnitudes. The normalized version of the covariance, the correlation coefficient, however, shows by its magnitude the strength of the linear relation.

Covariance between variables $x = x_t$ and $y = y_t$ is,

$$c_{xy} = \frac{1}{N}\sum_{i}^{N}(x_i - \bar{x})(y_i - \bar{y}). \tag{3.3.6}$$

Cross-covariance is a function in which larger values of x are associated with larger values of y and smaller values of x are associated with smaller values of y if the covariance is positive. It is doing opposite when the covariance is negative. So, for positive values of displacement between variables x_t and y_{t+k} is,

$$c_{xy}(k) = \frac{1}{N}\sum_{t=1}^{N-k}(x_t - \bar{x})(y_{t+k} - \bar{y}). \tag{3.3.7}$$

As mentioned, the Pearson correlation coefficient (correlation) is defined as the covariance of two variables divided by the product of their standard deviations (which are the square roots of their respective variances),

$$r_{xy} = \frac{c_{xy}}{\sqrt{c_{xx}c_{yy}}}. \tag{3.3.8}$$

The cross-correlation function is defined as,

$$r_{xy}(k) = \frac{c_{xy}(k)}{\sqrt{c_{xx}(0)c_{yy}(0)}}, \tag{3.3.9}$$

where, c_{xx} and c_{yy} are the variances of x and y, respectively. Also, $c_{xx}(0)$ and $c_{yy}(0)$ are the variances of x_t and y_t respectively.

So, correlation is used to test relationships between quantitative or categorical variables. In other words, it is a measure of how things are related.

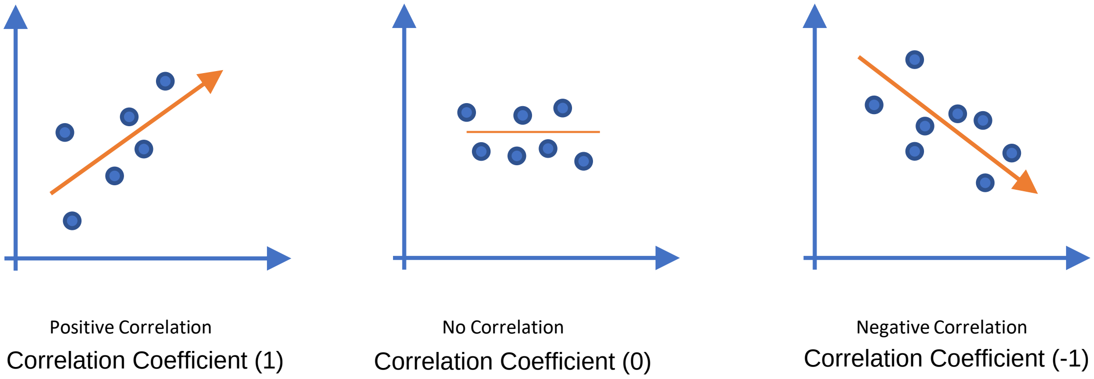

Figure 3.5: Correlation between the scatter of the points regarding their x and y positions. It is also the same in 2D histograms.

A correlation coefficient is a way to quantify the relationship. Correlation coefficients have a value between -1 and 1. 0 means there is no relationship between the variables at all, while -1 or 1 means that there is a perfect negative or positive correlation, as shown in Figure 3.5.

Cross-correlation compares two time series to detect a correlation between metrics with the same maximum and minimum values, like two audio signals in phase or how two images align. Normalized cross-correlation is also the comparison of two-time series. It can compare metrics with different value ranges. As all ranges are normalized in the same range, mostly in $[0, 1]$, one can compare values in different ranges. In [39], cross-correlation and normalized cross-correlation are used in image registration, which is about comparing the two images to find the maximum correlation between them, so one can do registration or mapping.

Given two input Images, I and I', that are size $M \times N$ and $P \times Q$, the 2-D cross-correlation value at the point (k, l) is given by:

$$C(k,l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m,n) I'(m+k, n+l). \tag{3.3.10}$$

The normalized cross-correlation value at the point (k, l) is calculated as:

$$C_N(k,l) = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m,n) I'(m+k, n+l)}{\sqrt{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m,n)^2} \sqrt{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I'(m+k, n+l)^2}} \tag{3.3.11}$$

where:

$0 \leq k < M + P - 1$

$0 \leq l < N + Q - 1$

After algebraic simplification, we find that the best NCC alignment is found by solving the optimization problem

$$\max_{a_{00},b_{00}} \Sigma_{x,y} \frac{(I(x,y) \cdot I'(x+a_{00},y+b_{00}))}{\sqrt{I^2(x,y)}.\sqrt{I'^2(x+a_{00},y+b_{00})}} \tag{3.3.12}$$

If an efficient library function in any programming language is available to calculate image convolutions, it can be used here. In our case, we are using the Numpy library in Python.

The stitching quality is expressed by measuring the correspondence between adjacent stitched images that form the composite image [38]. Image stitching (registration) methods have been explained in detail in [10], which contains the correlation-based and Fourier-based methods. Different sorts of distortions and transformations, as well as point mapping methods, are discussed. In the paper written by Flynn et al. [18], cross-correlation is the preferred method for automatically registering a large number of images. Various registration methods were compared in this paper, and it was shown that the cross-correlation method provided the smallest error, but we noticed different kinds of shifts and data sets need algorithm adjustments.

3.4 Harris Corner and Edge Detection

In 1988, Harris and Stephens introduced the Harris response function for corner and edge detection [21]. The Harris response function uses differences in pixel values for a pixel and its neighbours. Figure 3.6 indicates different choices of neighbours for comparison.

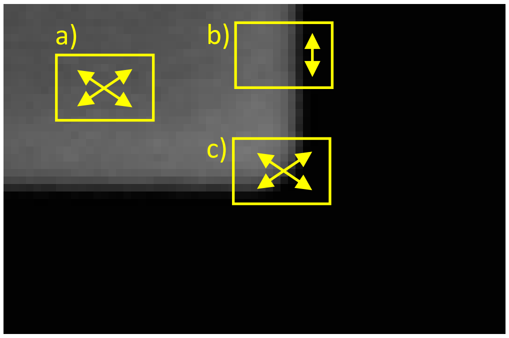

Figure 3.6: a) flat region, no changes in intensity in any direction, b) Edge, no changes in intensity along the edge, c) Corner, in any direction intensity changes, can get recorded.

They detect the features of the image using the eigenvalues and response function based on the auto-correlation surface, which will be discussed in this section.

Harris et al. [21] used the energy function E measuring differences caused by translating in (x, y) the image I as,

$$E(k, l) = \Sigma_{(i,j)\in\Omega} |I(k + i, l + j) - I(k, l)|^2, \tag{3.4.1}$$

where Ω is a neighbourhood of translations, e.g., the set of one-pixel shifts would be $\Omega = \{(1, 0), (1, 1), (0, 1), (-1, 1)\}$. One can fit a polynomial to the E and have an approximation of E using a quadratic surface. A function can be shown by its Taylor expansion of it [53]. Taylor series can be considered as polynomials with infinite terms used to approximate a function with its derivatives. The first few terms of the Taylor series can be used to approximate a function, but taking more terms will give a better approximation. The quadratic approximation of $E(k, l)$,

$$E \approx E(0,0) + (k,l)\nabla E + \frac{1}{2}(k,l)\nabla^2 E(k,l)^T, \tag{3.4.2}$$

in the neighbourhood of $(0, 0)$ is the second-order Taylor expansion. We use the second derivative of E in its Taylor expansion to approximate the auto-correlation surface (ACS). The fitted paraboloid, which approximates ACS, gives us information about the eigenvalues and their shape, which help to decide whether a chosen window shift gives us sufficient information. The quadratic term in the approximation can be written as

$$E = (k,l)M(k,l)^T, \tag{3.4.3}$$

where the partial derivatives in M can be approximated by the image derivatives.

$$M \approx \begin{bmatrix} \Sigma\frac{\partial I}{\partial k}\frac{\partial I}{\partial k} & \Sigma\frac{\partial I}{\partial k}\frac{\partial I}{\partial l} \\ \Sigma\frac{\partial I}{\partial k}\frac{\partial I}{\partial l} & \Sigma\frac{\partial I}{\partial l}\frac{\partial I}{\partial l} \end{bmatrix} \tag{3.4.4}$$

Note that Harris et al. [21] do not give conditions under which this approximation is valid, but rather validate the corner detector on real images.

Using diagonalization [4], M can be written as,

$$M = R^{-1}\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R \tag{3.4.5}$$

where λ_1 and λ_2 are the eigenvalues which indicate the length of the paraboloid in

x and y directions, and R is the rotation matrix which determines the orientation of the paraboloid.

If we consider the eigenvalues as local curvatures of the auto-correlation surface, then they have three different conditions to extract features:

- If both curvatures are small so that the local auto-correlation function is flat, then the windowed image region is of approximately constant intensity (ie. arbitrary shifts of the image patch cause little changes.)
- If one curvature is high and the other low, so that the local auto-correlation function is ridge shaped, then only shifts along the ridge (ie. along the edge) cause little change: this indicates an edge;
- If both curvatures are high so that the local auto-correlation function is sharply peaked, then shifts in any direction will increase our energy function: this indicates a corner.

It is not only necessary to classify corners and edges but also to measure their quality or response. Edges show the variation in one direction, while corners give us a better response as they measure the variation in both x and y directions. To select isolated corner pixels and edge pixels, the size of the response is considered. They introduce the measure of the response, R, which is required to be a function of α and β (eigenvalues). One can use the trace and determinant in the formulation, as this avoids the explicit eigenvalue decomposition. Thus the response function is formulated as,

$$R = Det - kTr^2, \tag{3.4.6}$$

where Det is the determinant and Tr is the trace of the Hessian (second derivative) of the paraboloid. Constant k is to balance the tradeoff between trace and determinant, which in literature, is chosen to be $[0.04, 0.06]$, [51] [49].

The Harris Algorithm is based on approximating the distinctiveness using ACS. This approximation is calculated based on the second derivative of the energy function E. The criteria for choosing landmarks is based on the eigenvalues, which divide our choices into flat, edge, or corner regions.

In sum, stitching can be done by translation, and the applied procedure can be referred to as mosaicing, tiling, montaging, or stitching. First, we have to find the relative positions of images that we want to match. We also need an empty image array that we will define in our program code where these images will be placed. We need to move image edges in both directions for the next step until the best match of features is found. This is how we will find the best correlation point between adjacent image edges[38].

Based on all these experiments (SSD, NCC, FFT, and Harris), we devised novel and efficient algorithms to solve the stitching problem. Our method increases the accuracy and efficiency of the stitching by detecting the landmark and reducing the computation dimension from $2D$ to $1D$, which we will discuss in Chapter 4.

Chapter 4

Novel Approaches in 2D Stitching

Stitching algorithms need powerful registration techniques. There are two types of methods that have been used for registration: 1. Intensity-based and 2. feature-based. Although the first method is relatively simple to implement, comparing millions of pixels against millions of pixels is computationally expensive. The feature-based approach establishes correspondences between points, lines, or other geometrical entities in overlapping image regions with higher computational costs. They are collectively called landmarks. In section 4.1, we introduce a new set of landmarks and resolve the registration problem in stitching neighbouring tiles. In section 4.2, we reduce the dimensionality to accelerate the stitching process and increase accuracy.

4.1 Numerical landmarks

Stitching, mathematically, turns out to be a registration problem, defined as finding the maximum correlation or minimum difference between the overlapping area of the neighbouring images. An accurate procedure is necessary, particularly if one wants to extract quantitative data from a stitched image. Typical features are, for example, hard or soft landmarks in the images. A landmark is the location of a typically outstanding feature of an image, particularly intense vias in semiconductor metallization datasets. **Hard landmarks** are so-called fiducial markers added to the sample before imaging at certain spatial positions on a sample. In SEM imaging, fiducials can be added by using the FIB (focussed ion beam) to create a hole or trench in the sample. These landmarks can be identified in the images with high accuracy. However, fabricating this type of landmark might damage the sample or interfere with the material's functionality if they are deliberately etched or patterned into the sample. In contrast, **soft** or **retrospective landmarks** are deduced from the images, based on similarity to predetermined features, e.g., corners in any image, figures in medical imaging, or cell components in biological imaging. The spatial location of these landmarks requires expert knowledge and/or sophisticated image analysis tools for automatic detection.

In this section, we introduce a new set of landmarks called **numerical landmarks** based on the similarity of a subimage to its own translates. Specifically, if the autocorrelation surface measuring the closeness of fit as a function of the translation vector has a steep and isolated valley/peak (for SSD/NCC). Harris et al. [21] based their corner-detector on the same concept: the energy function given by the SSD between the image and a shifted version of itself, but their derivation diverges from ours by using the image derivatives (differences between neighboring pixels) to approximate the first order coefficients of the Taylor polynomial to the energy function, which makes their corner detector a pixel-wise function, whereas we consider much large subimages.

In Figure 4.1 we compare Harris' original method with our own. Both methods define a response function whose numerical value can be used to identify landmarks. In Harris' case, the response function was calculated based on image derivatives and minimization. The response function identifies areas of positive curvature as corners. In our case, the response function takes the same form as 3.4.6, but we use a polynomial approximation to the NCC-based ACS, which is maximized at what we call *numerical landmarks*.

All other methods (FFT, NCC, and SSD) directly find the shift in $x-$ and $y-$ directions to match the subimages in the overlap area of the adjacent tiles. This is not the case for Harris and our numerical landmark method. Harris assigns a score to each pixel, with the highest scores for definite corners and lower scores for possible corners or edges. The location of isolated corners could be used for registration, but Harris' response function does not generally find isolated corners. Our numerical landmarks are subimages that would produce maximum correlations with themselves and much lower correlations with their neighbourhood. This makes it hard to compare numerical landmarks to the first three methods or Harris' response function.

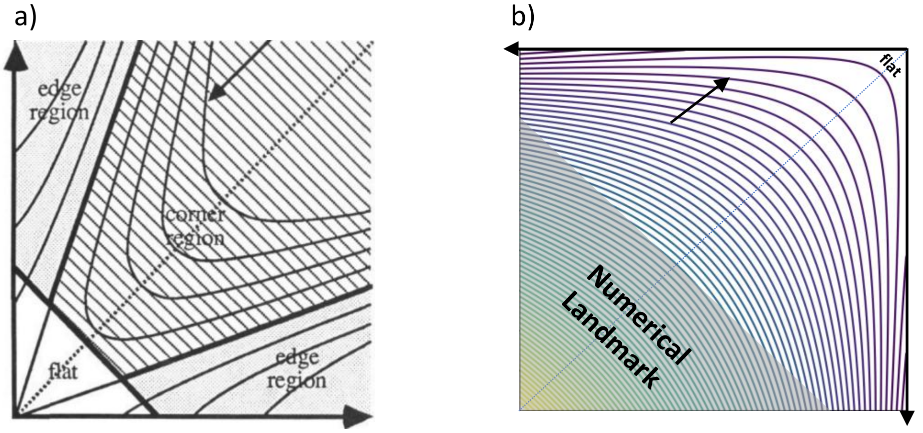

Figure 4.1: Contour plot of Harris Corner ACS principal curvature space, in which the horizontal and vertical coordinates represent the eigenvalues of the Hessian of the best-fit paraboloid. Curved lines represent equal R values, and shading represents different image features (corner, edge, or flat regions). (Reproduced from [21].) b) Shows iso-R values for the modified response function based on NCC. Since we use NCC rather than SSD because it is more robust in changes of brightness, the eigenvalues in (b) are negative, so the orientation is reversed. Since the ACS is calculated for a subimage in (b) rather than for a single pixel in (a) we cannot characterize regions as corners and edges, but rather as numerical landmarks.

In the first step, the relative positions of the images we want to match are chosen based on the information from the imaging tool (SEM). Then the overlap area is divided into thinner strips with a degree of overlap. Each strip is divided into smaller subimages, and self-correlation is applied to each; see Figure 4.2. As we are facing changes in brightness due to charging during the course of imaging multiple tiles, we use NCC for self-correlation, as it is less sensitive to such changes compared to SSD, which is used in the original Harris corner detection algorithm.

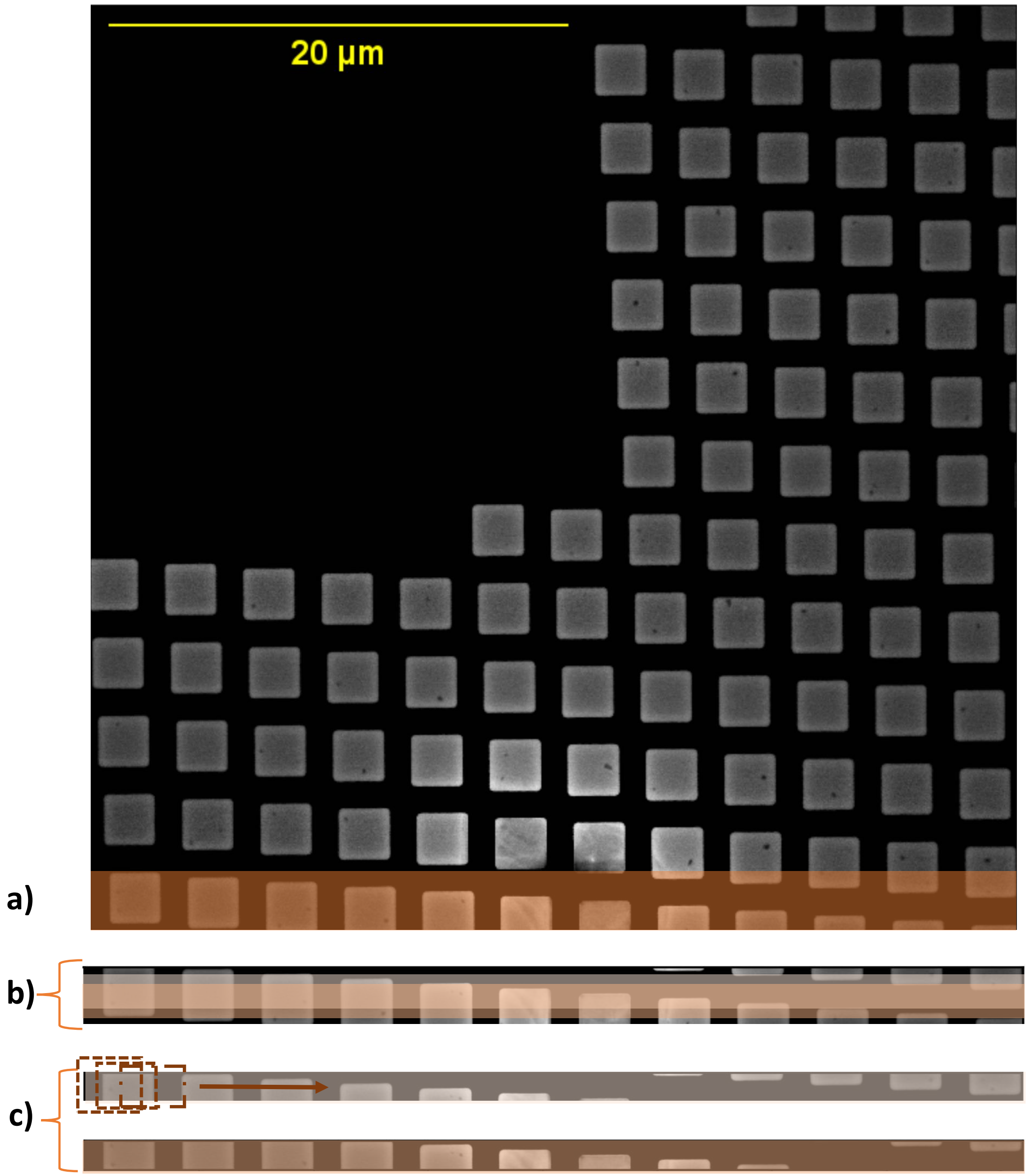

Figure 4.2: Self-Correlation to find numerical landmarks. a) The height of the overlap between adjacent tiles in this data set is 512 pixels. b) In the overlap region, two stripes with the height of 256 pixels and an overlap of 100 pixels are highlighted. c) In each of the stripes, dashed rectangles show how each stripe is divided into smaller subimages overlapping horizontally by 100 pixels.

Figure 4.3 (a) represents the self-correlation on an example subimage from the overlap area. After calculating the NCC for the inner (dashed-line) and outer (solid-line) subimages, we fitted a paraboloid to the auto-correlation surface using a weighted objective function.

$$\min_{a,b,c,d,e,f} \; \Sigma_{x,y} \; w_{x,y} \; \left(z_{\text{NCC}} - \left(ax^2 + 2bxy + cy^2 + dx + ey + f\right)\right)^2 \tag{4.1.1}$$

where

$$w_{x,y} = \begin{cases} 0 & x = r \\ L\left(\sqrt{(x-r)^2 + (y-r)^2}\right) & \text{otherwise} \end{cases} \tag{4.1.2}$$

$$L(u) = \begin{cases} \frac{\sin(\pi u/r)}{(\pi u/r)} & u < r \\ 0 & u \geq r. \end{cases} \tag{4.1.3}$$

In (4.1.1), we define the optimization problem for fitting the polynomial $ax^2 + 2bxy + cy^2 + dx + ey + f$ to z_{NCC}, the ACS. This problem is quadratic in the coefficients, and can be solved by setting the derivates of the objective function to zero, as follows:

$$
\begin{aligned}
0 =& \frac{\partial}{\partial a}\Sigma_{x,y} w_{x,y} \left(z_{\text{NCC}} - \left(ax^2 + 2bxy + cy^2 + dx + ey + f\right)\right)^2 \\
=& \frac{\partial}{\partial a}\Sigma_{x,y} w_{x,y} \left(z_{\text{NCC}} - ax^2 - 2bxy - cy^2 - dx - ey - f\right)^2 \\
=& \Sigma_{x,y}(-2) w_{x,y} \left(x^2\right) \left(z_{\text{NCC}} - ax^2 - 2bxy - cy^2 - dx - ey - f\right) \\
=& (-2)\Sigma_{x,y} \bigl(w_{x,y}x^2 z_{\text{NCC}} - a w_{x,y} x^4 - 2b w_{x,y} x^3 y - c w_{x,y} x^2 y^2 - \\
& d w_{x,y} x^3 - e w_{x,y} x^2 y - f w_{x,y} x^2\bigr) \\
=& (-2)\bigl(\Sigma_{x,y} w_{x,y} x^2 z_{\text{NCC}} - a\Sigma_{x,y} w_{x,y} x^4 - b\Sigma_{x,y} 2 w_{x,y} x^3 y - c\Sigma_{x,y} w_{x,y} x^2 y^2 - \\
& d\Sigma_{x,y} w_{x,y} x^3 - e\Sigma_{x,y} w_{x,y} x^2 y - f\Sigma_{x,y} w_{x,y} x^2\bigr)
\end{aligned}
$$

$$
\begin{aligned}
0 =& \frac{\partial}{\partial b}\Sigma_{x,y} w_{x,y} \left(z_{\text{NCC}} - \left(ax^2 + 2bxy + cy^2 + dx + ey + f\right)\right)^2 \\
=& \frac{\partial}{\partial b}\Sigma_{x,y} w_{x,y} \left(z_{\text{NCC}} - ax^2 - 2bxy - cy^2 - dx - ey - f\right)^2 \\
=& \Sigma_{x,y}(-4) w_{x,y} \left(xy\right) \left(z_{\text{NCC}} - ax^2 - 2bxy - cy^2 - dx - ey - f\right) \\
=& (-4)\bigl(\Sigma_{x,y} w_{x,y} z_{\text{NCC}} xy - \Sigma_{x,y} w_{x,y} a x^3 y - \Sigma_{x,y} w_{x,y} 2b x^2 y^2 - \Sigma_{x,y} w_{x,y} c x y^3 - \\
& \Sigma_{x,y} w_{x,y} d x^2 y - \Sigma_{x,y} w_{x,y} e x y^2 - \Sigma_{x,y} w_{x,y} f xy\bigr)
\end{aligned}
$$

$$
\begin{aligned}
0 =& \frac{\partial}{\partial c}\Sigma_{x,y}w_{x,y}\left(z_{\mathrm{NCC}} - \left(ax^2 + 2bxy + cy^2 + dx + ey + f\right)\right)^2 \\
=& \frac{\partial}{\partial c}\Sigma_{x,y}w_{x,y}\left(z_{\mathrm{NCC}} - ax^2 - 2bxy - cy^2 - dx - ey - f\right)^2 \\
=& \Sigma_{x,y}(-2)w_{x,y}\left(y^2\right)\left(z_{\mathrm{NCC}} - ax^2 - 2bxy - cy^2 - dx - ey - f\right) \\
=& (-2)\left(\Sigma_{x,y}w_{x,y}y^2z_{\mathrm{NCC}} - \Sigma_{x,y}w_{x,y}ay^2x^2 - \Sigma_{x,y}w_{x,y}2bxy^3 - \Sigma_{x,y}w_{x,y}cy^4 - \right. \\
& \left. \Sigma_{x,y}w_{x,y}dxy^2 - \Sigma_{x,y}w_{x,y}ey^3 - \Sigma_{x,y}w_{x,y}fy^2\right)
\end{aligned}
$$

$$
\begin{aligned}
0 =& \frac{\partial}{\partial d}\Sigma_{x,y}w_{x,y}\left(z_{\mathrm{NCC}} - \left(ax^2 + 2bxy + cy^2 + dx + ey + f\right)\right)^2 \\
=& \frac{\partial}{\partial d}\Sigma_{x,y}w_{x,y}\left(z_{\mathrm{NCC}} - ax^2 - 2bxy - cy^2 - dx - ey - f\right)^2 \\
=& (-2)(x)\Sigma_{x,y}w_{x,y}\left(z_{\mathrm{NCC}} - ax^2 - 2bxy - cy^2 - dx - ey - f\right) \\
=& (-2)\left(\Sigma_{x,y}w_{x,y}xz_{\mathrm{NCC}} - \Sigma_{x,y}w_{x,y}ax^3 - \Sigma_{x,y}w_{x,y}2bx^2y - \Sigma_{x,y}w_{x,y}xcy^2 - \right. \\
& \left. \Sigma_{x,y}w_{x,y}dx^2 - \Sigma_{x,y}w_{x,y}exy - \Sigma_{x,y}w_{x,y}xf\right)
\end{aligned}
$$

$$
\begin{aligned}
0 =& \frac{\partial}{\partial e}\Sigma_{x,y}w_{x,y}\left(z_{\mathrm{NCC}} - \left(ax^2 + 2bxy + cy^2 + dx + ey + f\right)\right)^2 \\
=& \frac{\partial}{\partial e}\Sigma_{x,y}w_{x,y}\left(z_{\mathrm{NCC}} - ax^2 - 2bxy - cy^2 - dx - ey - f\right)^2 \\
=& (-2)(y)\Sigma_{x,y}w_{x,y}\left(z_{\mathrm{NCC}} - ax^2 - 2bxy - cy^2 - dx - ey - f\right) \\
=& (-2)\left(\Sigma_{x,y}w_{x,y}yz_{\mathrm{NCC}} - \Sigma_{x,y}w_{x,y}ayx^2 - \Sigma_{x,y}w_{x,y}2bxy^2 - \Sigma_{x,y}w_{x,y}cy^3 - \right. \\
& \left. \Sigma_{x,y}w_{x,y}dyx - \Sigma_{x,y}w_{x,y}ey^2 - \Sigma_{x,y}w_{x,y}yf\right)
\end{aligned}
$$

$$
\begin{aligned}
0 =& \frac{\partial}{\partial f}\Sigma_{x,y}w_{x,y}\left(z_{\text{NCC}} - \left(ax^2 + 2bxy + cy^2 + dx + ey + f\right)\right)^2 \\
=& \frac{\partial}{\partial f}\Sigma_{x,y}w_{x,y}\left(z_{\text{NCC}} - ax^2 - 2bxy - cy^2 - dx - ey - f\right)^2 \\
=& (-2)\Sigma_{x,y}w_{x,y}\left(z_{\text{NCC}} - ax^2 - 2bxy - cy^2 - dx - ey - f\right) \\
=& \big(\Sigma_{x,y}w_{x,y}z_{\text{NCC}} - \Sigma_{x,y}w_{x,y}ax^2 - \Sigma_{x,y}w_{x,y}2bxy - \Sigma_{x,y}w_{x,y}cy^2 - \\
& \Sigma_{x,y}w_{x,y}dx - \Sigma_{x,y}w_{x,y}ey - \Sigma_{x,y}w_{x,y}f\big)
\end{aligned}
$$

Setting the above derivatives to zero, we can solve for the polynomial coefficients a , b, c, d, e, and f. Figure 4.3 (b) shows the ACS and the weighted polynomial fit using the obtained coefficients. It also shows the contour plot of the fitted ACS and the curvatures. Figure 4.3 (c) shows how fitting paraboloids eliminate the noise in our calculation.

The response function is calculated based on the eigenvalues of the Hessian, H. As the determinant is the multiplication of the eigenvalues and the trace is their sum, we use determinant and trace in our calculation because they are simpler to calculate rather than eigenvalues.

$$
\begin{aligned}
R &= \lambda_{min}\lambda_{max} - k(\lambda_{min} + \lambda_{max})^2 && (4.1.4) \\
&= \det(H) - k\,\text{tr}(H) && (4.1.5)
\end{aligned}
$$

Notice that for round paraboloids, eigenvalues are equal and $R = 0$ when $k = 0.25$. By choosing $k = 0.25$, the response function is always zero for the rounded paraboloids, and for the non-rounded ones, it is negative. In our implementation, we use $k = 0.125$, which controls for both roundness and large determinants/curvature, as being round is not enough, we want sharp local minima, and we need big eigenvalues/determinants for this. The choice 0.125 is halfway in between. All steps of the proposed algorithm are listed in Algorithm 1. In this algorithm, the subimages are taken by stepping through the overlap area with step size 100 both horizontally and vertically. The weighting for the polynomial fit in Algorithm 1 is to eliminate the noise caused by the acquisition, see Figure 4.3, and reduce the artifacts like aliasing similar to Gibb's ringing [31].

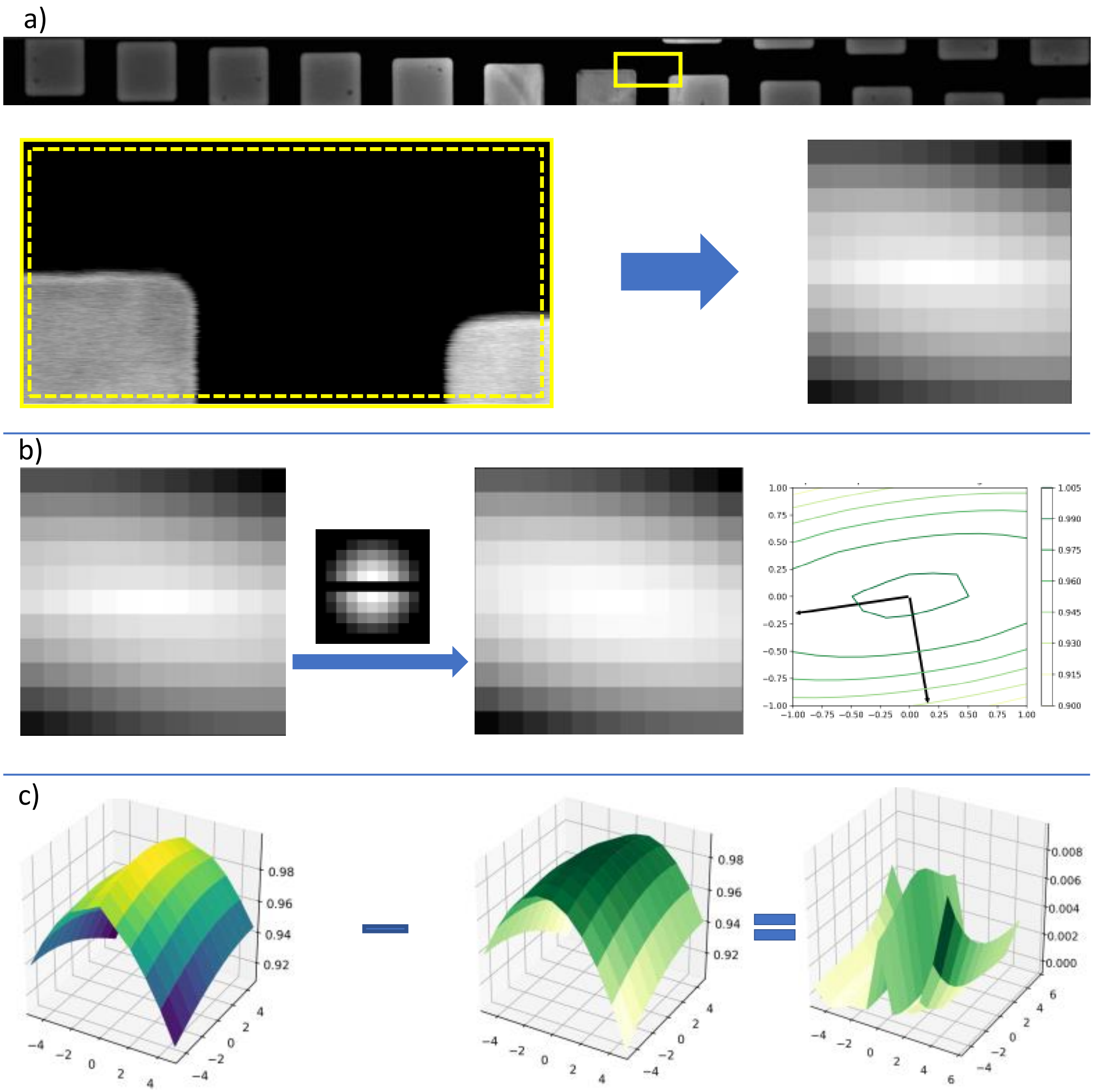

Figure 4.3: A subimage from the overlap area is shown in the top row. In section a), the solid yellow rectangle represents the subimage, and the dashed rectangle represents the inner rectangle moving inside the solid rectangle and returning the NCCs of the central subimage and its translates. The margin is 5 pixels from each side; as a result, the ACS is 11 × 11 pixels. b) A weighting factor is applied to the ACS to remove the noisy central row and weight the NCCs according to the magnitude of the translation. The contour plot shows the best-weighted-fit paraboloid to the ACS and the eigenvectors of its Hessian from which we can read the roundness of the paraboloid. c) The difference between the raw ACS and the fitted paraboloid shows how the noise is removed from the calculation. Note, in particular, the removal of the central ridge along the $x-$axis.

Algorithm 1 Numerical landmark search for horizontal overlap

```
Require: $I$                                                        ▷ Image
Require: $h$                                        ▷ Height of the overlap
Require: $l$                                         ▷ Width of the overlap
Require: $h_s$                                    ▷ Height of the subimage
Require: $l_s$                                     ▷ width of the subimage
Require: $w_{x,y}$                          ▷ Weighting for polynomial fit
Require: $k$                                     ▷ Parameter, used 0.125
Require: $\mathcal{I}$                           ▷ set of subimages (see text)
Ensure: $h_s \leq h, l_s \leq l$
  for $I' \leftarrow \mathcal{I}$ do
    $z_{\text{NCC}} \leftarrow \text{NCC}(I')$       ▷ Autocorrelation Surface. See Algorithm 2
    $a, b, c, d, e, f \leftarrow \text{ argmin } \Sigma_{x,y}\ w_{x,y}\ \left(z_{\text{NCC}} - (ax^2 + 2bxy + cy^2 + dx + ey + f)\right)^2$
    $H \leftarrow \begin{pmatrix} 2a & 2b \\ 2b & 2c \end{pmatrix}$                  ▷ Hessian of parabaloid at (0,0)
    $R \leftarrow \det(H) - k\,\text{tr}(H)^2$                  ▷ Harris response function
    if $R \geq 0$ then add to list of landmarks
    end if
  end for
```

The NCC in Algorithm 1 is calculated using Algorithm 2. In this algorithm, we pick a central subimage inside I_s and calculate the correlation between it and its translates within I up to a maximum translation of 5 pixels in either direction. Rounded paraboloid ACSs with unique maximums are the best to use in Algorithm 1.

Figure 4.4 shows three data sets from our industrial partner to analyze them with our algorithm. The response function histograms and the threshold zero (orange bars) show how we pick numerical landmarks from the overlap area based on our numerical criteria. As the NCC should have a maximum in the case of no translation, eigenvalues should be both negative. As discussed, the roundness of the paraboloids

Algorithm 2 NCC for self-correlation

```
Require: $I_s$                                   ▷ Subimage
Require: $h_s$                                   ▷ Height of the subimage
Require: $l_s$                                   ▷ width of the subimage
Require: $m$                                     ▷ margin
  $r, c \leftarrow size(I_s)$        ▷ Number of rows (r) and number of columns (c)
  $I_{\text{mid}} \leftarrow I_s[m : r - m, m : c - m]$
  for $i \leftarrow -m$ to $m$ do
      for $j \leftarrow -m$ to $m$ do
          $I_{\text{move}} \leftarrow I_s[i : i + r - 2m, j : j + c - 2m]$
          $\text{NCC}_{i,j} \leftarrow \frac{I_{\text{move}} \cdot I_{\text{mid}}}{\sqrt{I_{\text{move}} \cdot I_{\text{move}}} \sqrt{I_{\text{mid}} \cdot I_{\text{mid}}}}$
      end for
  end for
  return NCC matrix
```

corresponds to having equal eigenvalues. Small R gives us flat regions, negative R indicates edges, and positive R means that the subimage significantly changes when moving in any direction and corresponds to corners or significant landmarks like vias in semiconductor SEM images. Of the images which pass this threshold, if there are overlapping subimages, we use only the one with the maximum R value.

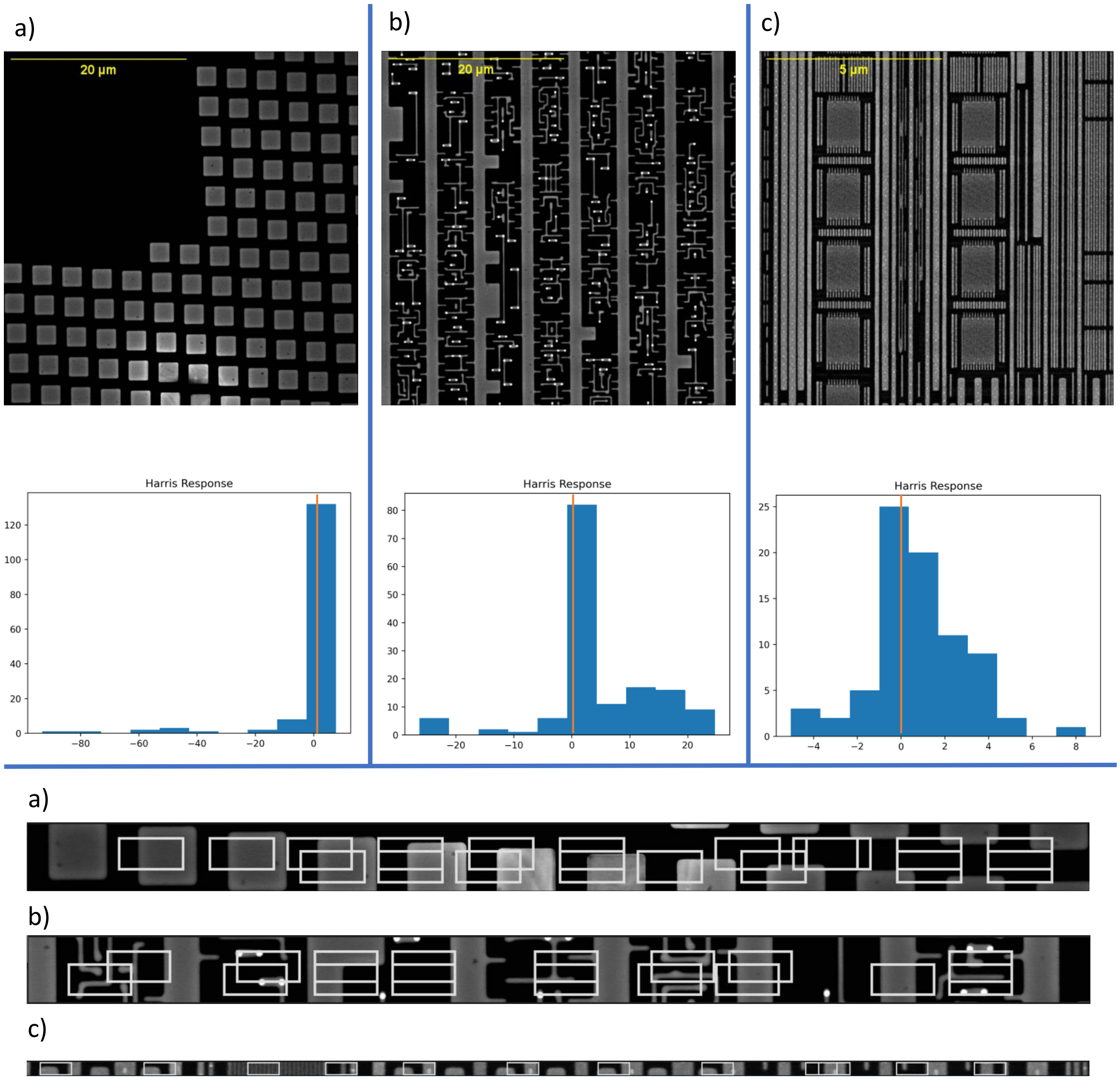

Figure 4.4: Three data sets are compared. The histograms show the modified Harris Corner response function based on NCC, with the frequency of subimages with different R values plotted against the R values. The orange line on the histogram shows the threshold for picking numerical landmarks. a, b and c indicate different data sets and white outlined subimages on the stripes show numerical landmarks with localizable features in them for the stitching.

Figures 4.5, 4.6 , and 4.7 show the 2D histograms of eigenvalues for different subimages. The maximum and minimum eigenvalues are plotted on the horizontal and vertical axes, respectively. Figure 4.5 shows one of the chosen subimages and self-correlation results, while the next two show the best curvature and the worst ones. In order to plot the $2D$ histogram, a Python library named "plotly" is used. One can plot the $1D$ histogram of the values on each axes on the edge of the $2D$ histogram to track the changes in one axis for different amounts of the other axis. In our case, the axes are eigenvalues.

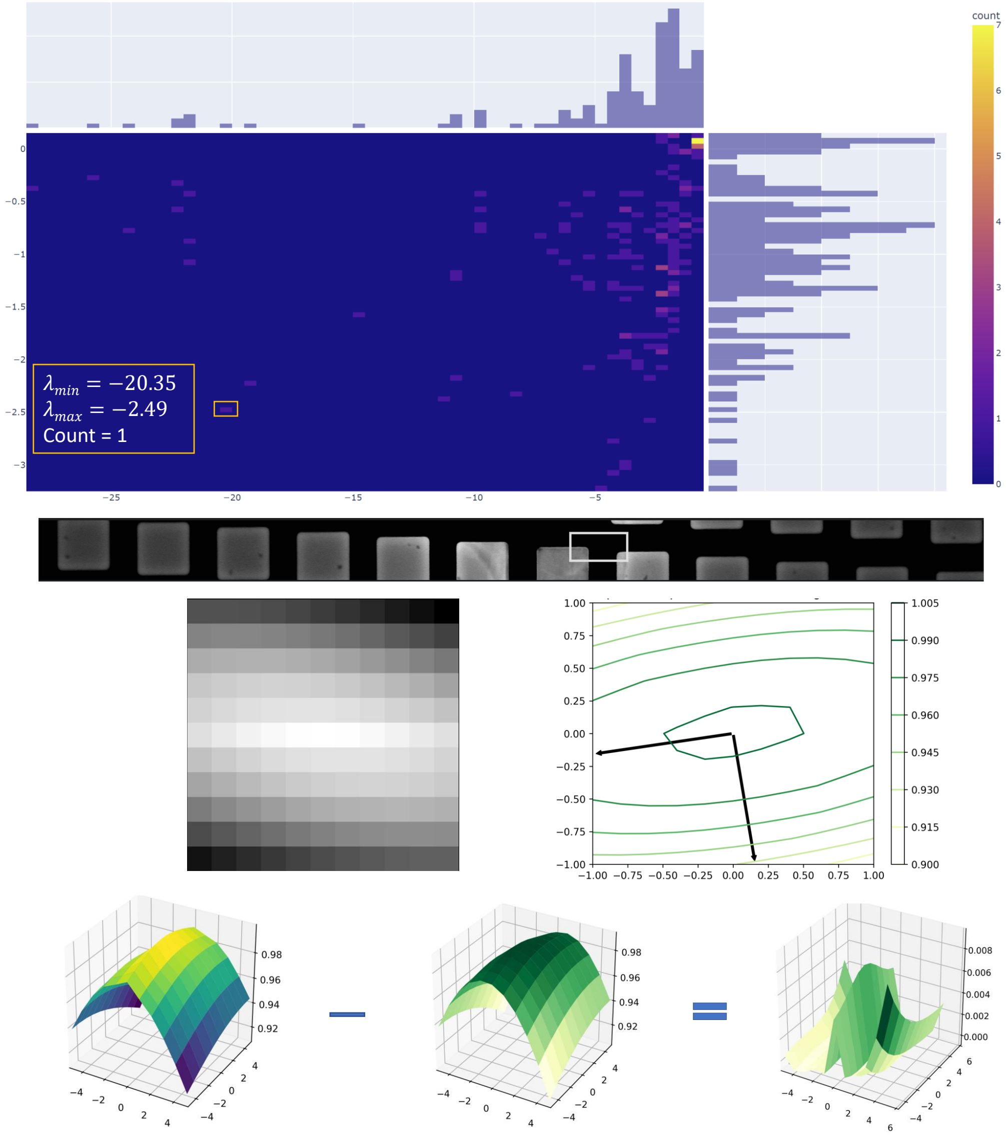

Figure 4.5: Using self-correlation to find numerical landmarks ($R = -14.53$). This subimage will be selected because it has a large R value maximum and minimum eigenvalues of the Hessian of the second-degree polynomial approximation to the ACS. Histograms of the eigenvalues are shown separately (to the top and right) and as a 2D histogram *(top)*. The $(0, 0)$ point in the histogram is near the top right corner and contains the highest counts, indicating that most subimages are not useful as numerical landmarks. The position of this subimage in the overlap region is shown next, followed by the raw ACS and a contour plot of the 2D fit. The eigenvectors of its Hessian are indicated on the contour plot. Finally, the quality of the fit is illustrated *(bottom)* by visually subtracting surface plots of the polynomial fit from the raw ACS.

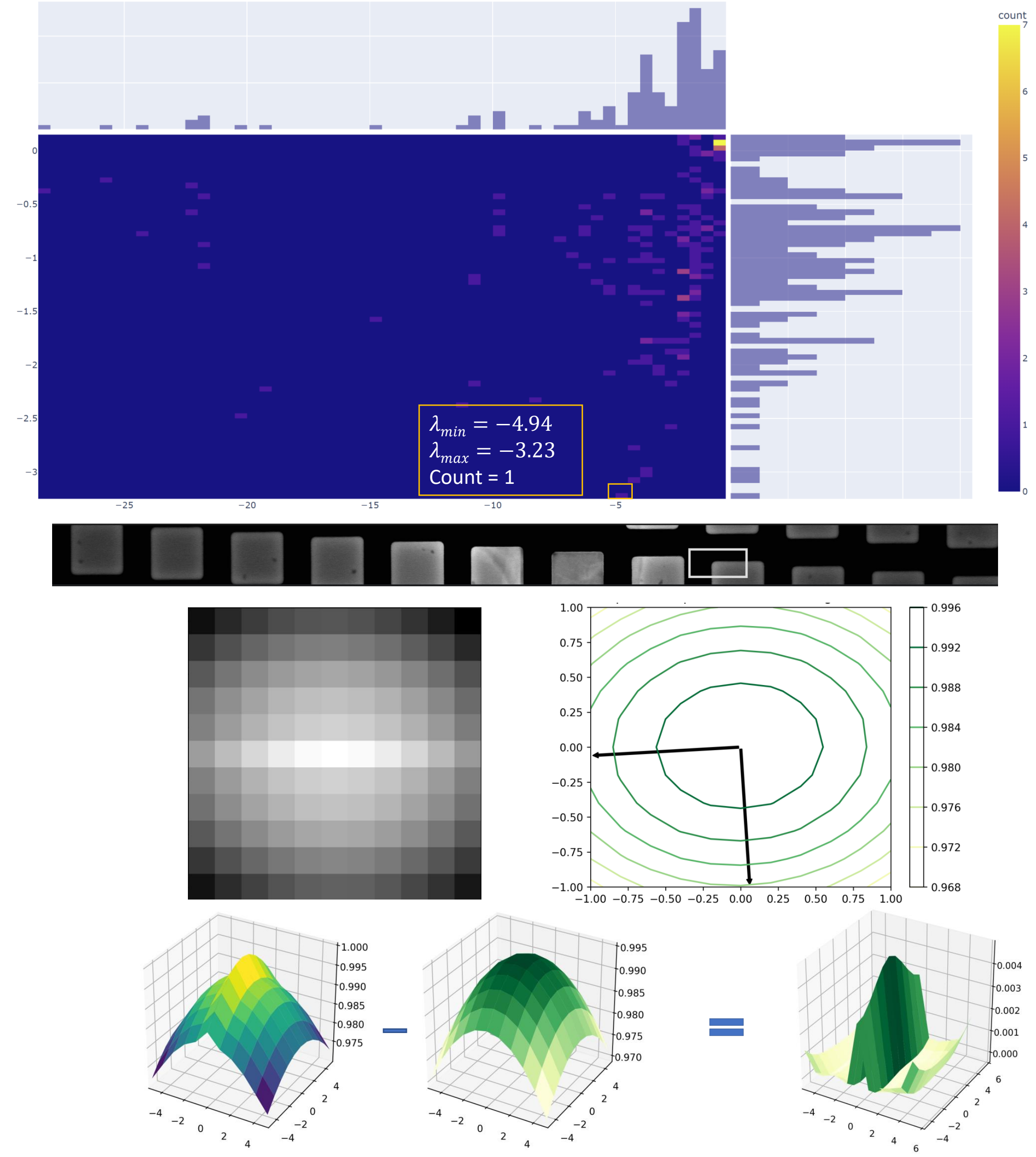

Figure 4.6: Using self-correlation to find numerical landmarks ($R = 7.61$). The best case with the best curvature in all directions.

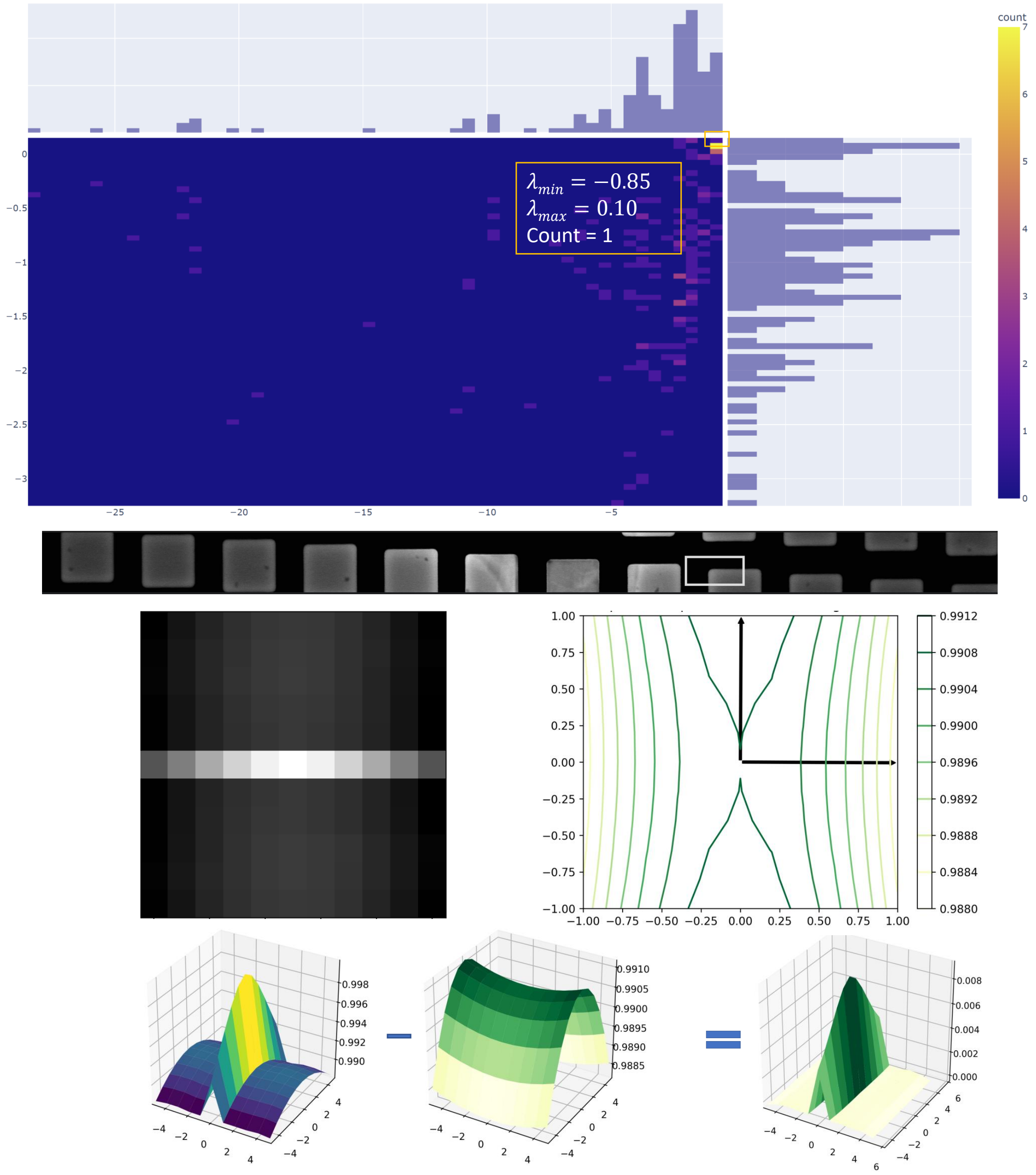

Figure 4.7: Using self-correlation to find numerical landmarks ($R = -0.15$). It indicates the worst curvature.

The next data set has more complex patterns of wires and vias, both isolated or grouped. Figures 4.8, 4.9, and 4.10 show the results of our algorithm running for this dataset.

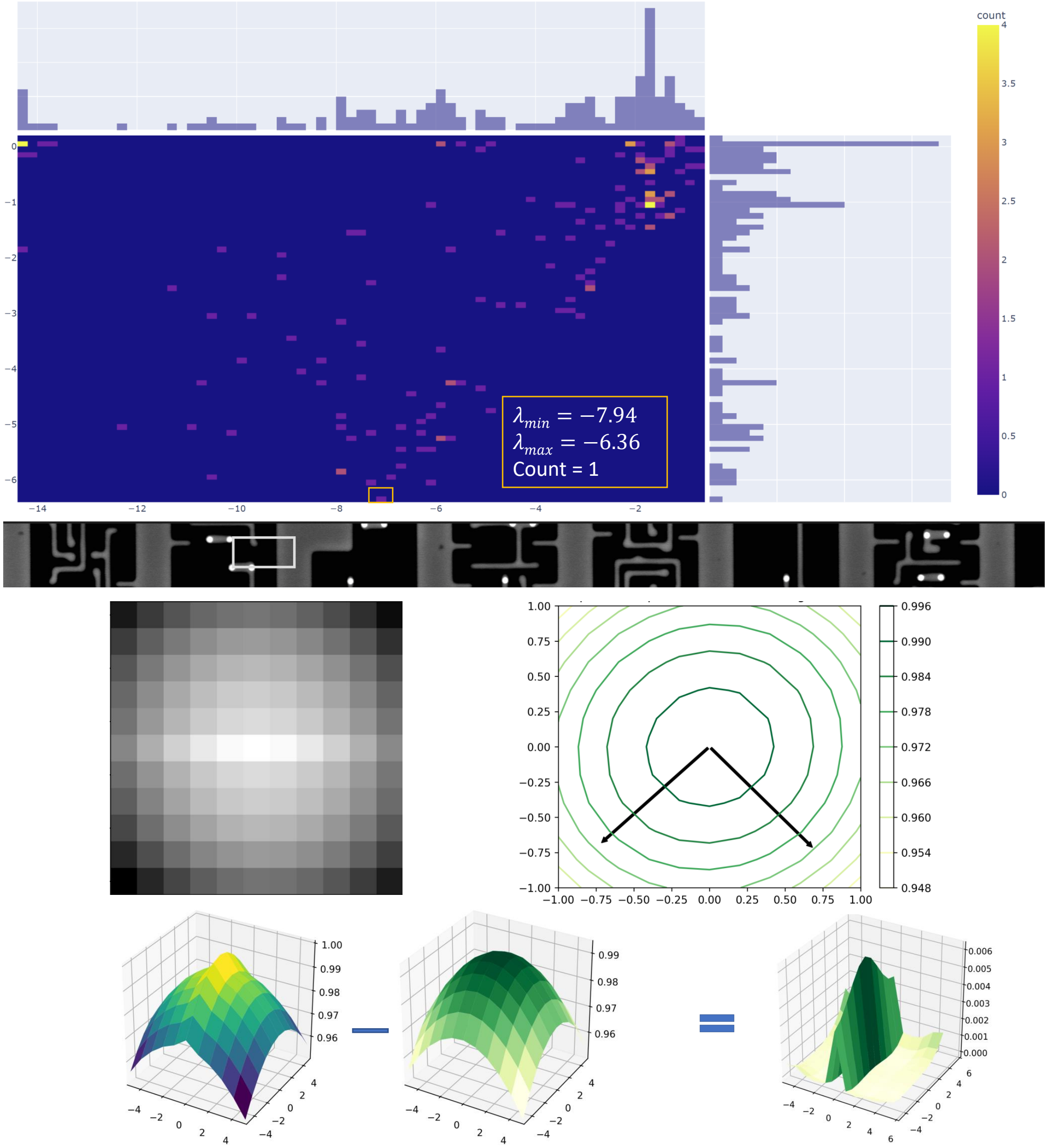

Figure 4.8: Self-Correlation to find numerical landmarks ($R = 26.71$), which indicates one of the algorithm-chosen subimages with a rounded shape paraboloid.

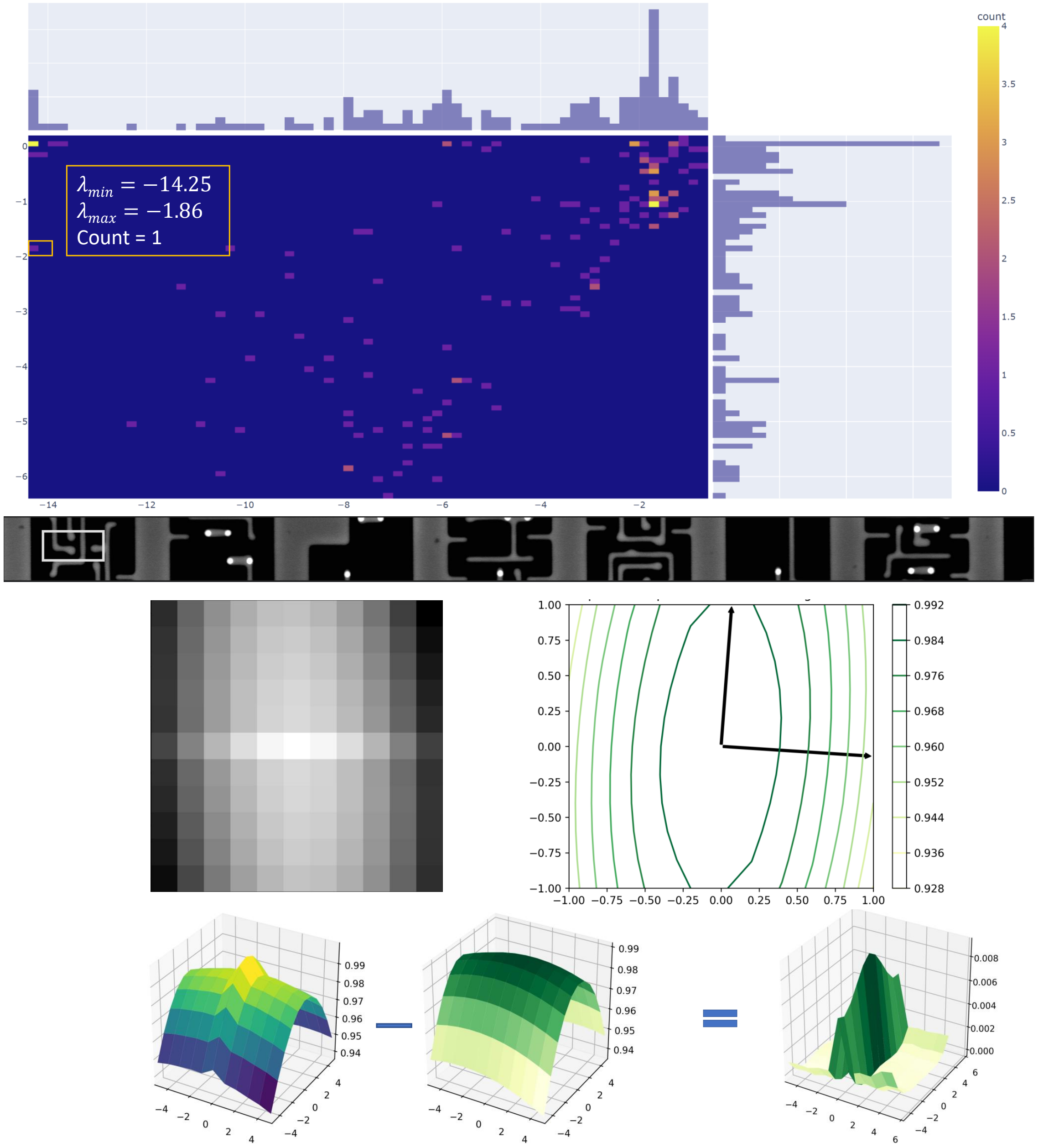

Figure 4.9: Typical numerical landmarks identified by the proposed procedure ($R = -5.93$). A preponderance of features (e.g., wires) in one orientation results in uneven eigenvalues, but both are sufficiently negative that the subimage can be used as a numerical landmark.

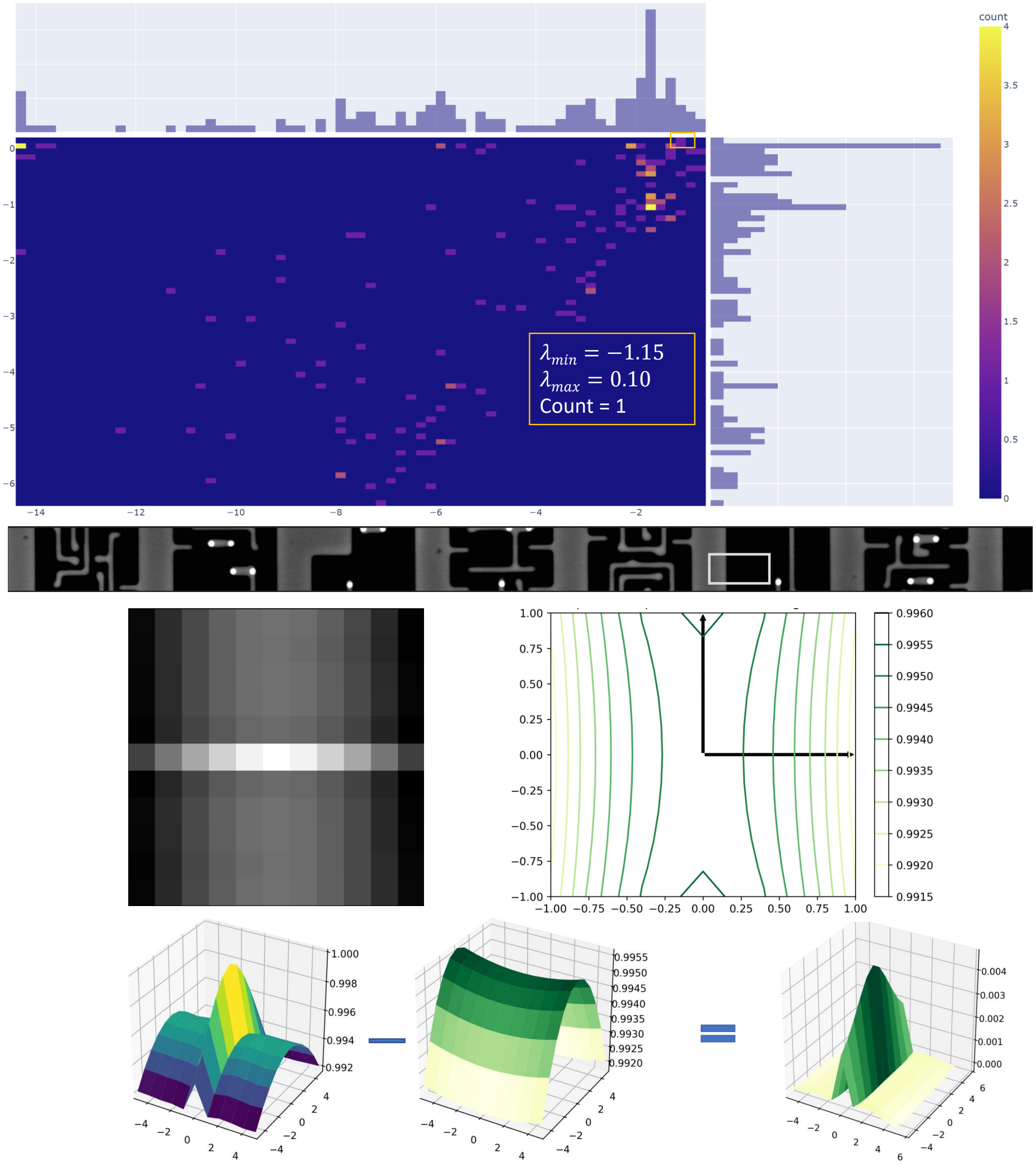

Figure 4.10: Self-Correlation to find numerical landmarks ($R = -0.3$). It represents the worst curvature for a subimage in this data set. From the contour plot, we see that the curvature is slightly negative. This happens when a subimage contains only features with one orientation.

The third data set has a noise which we named rice noise, as its pattern is like the grains of rice. The noise is in the SEM scan direction. Figures 4.11, 4.12, and 4.13 are showing the results.

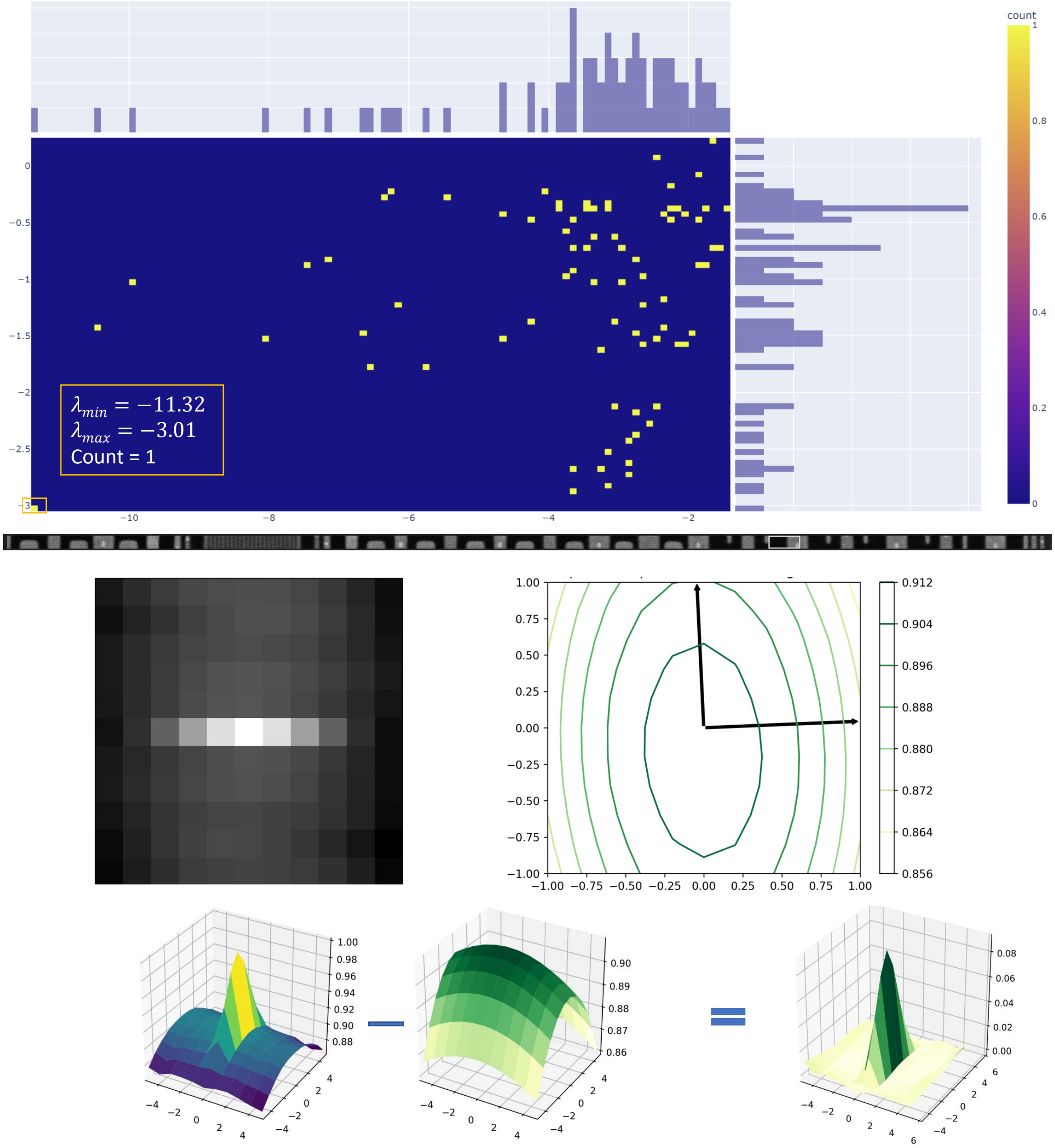

Figure 4.11: Self-Correlation to find numerical landmarks ($R = 8.44$). It represents one of the bad choices for a subimage in this data set.

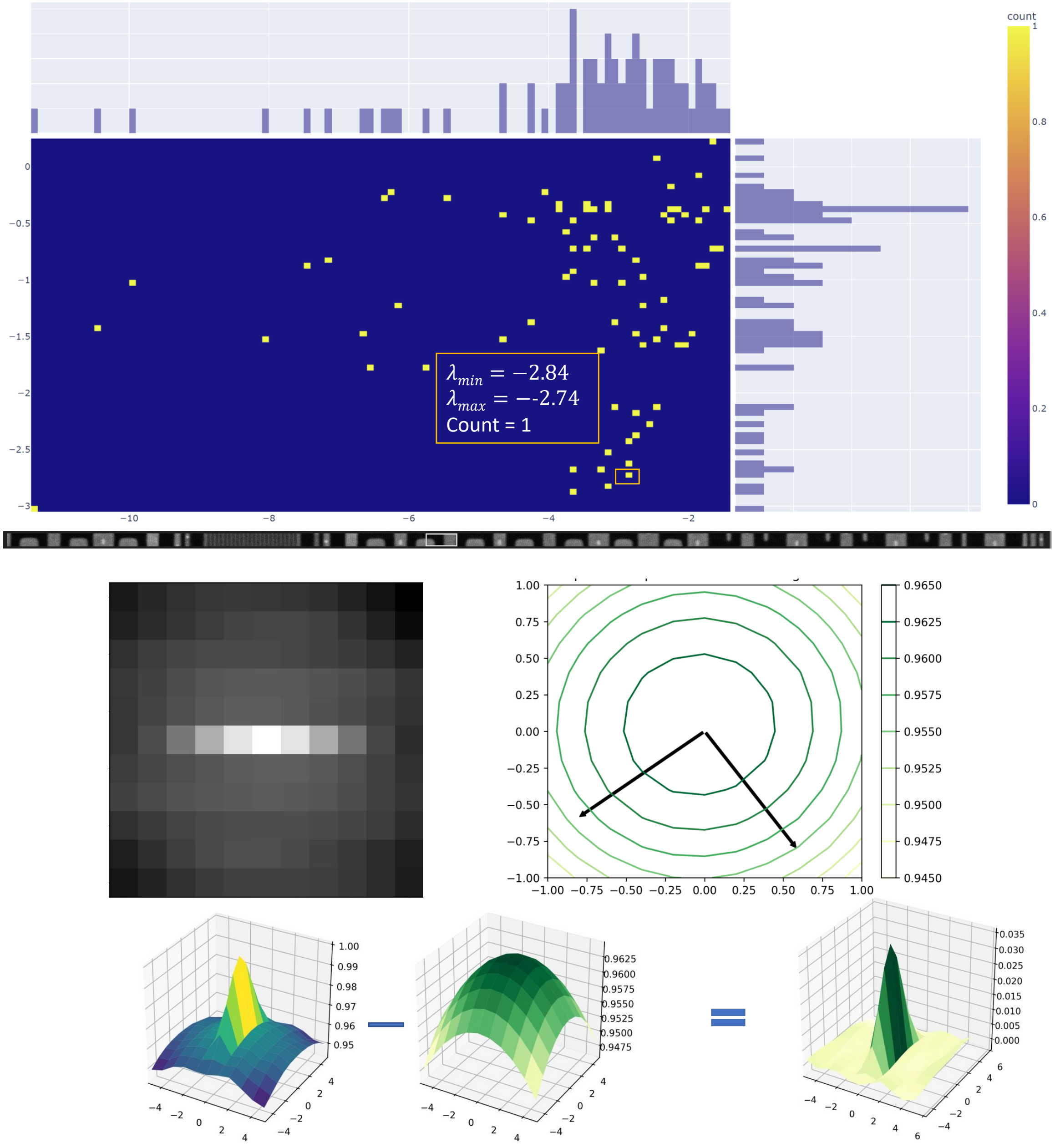

Figure 4.12: Self-Correlation to find numerical landmarks ($R = 3.90$), which indicates one of the algorithm-chosen subimages with a rounded shape paraboloid.

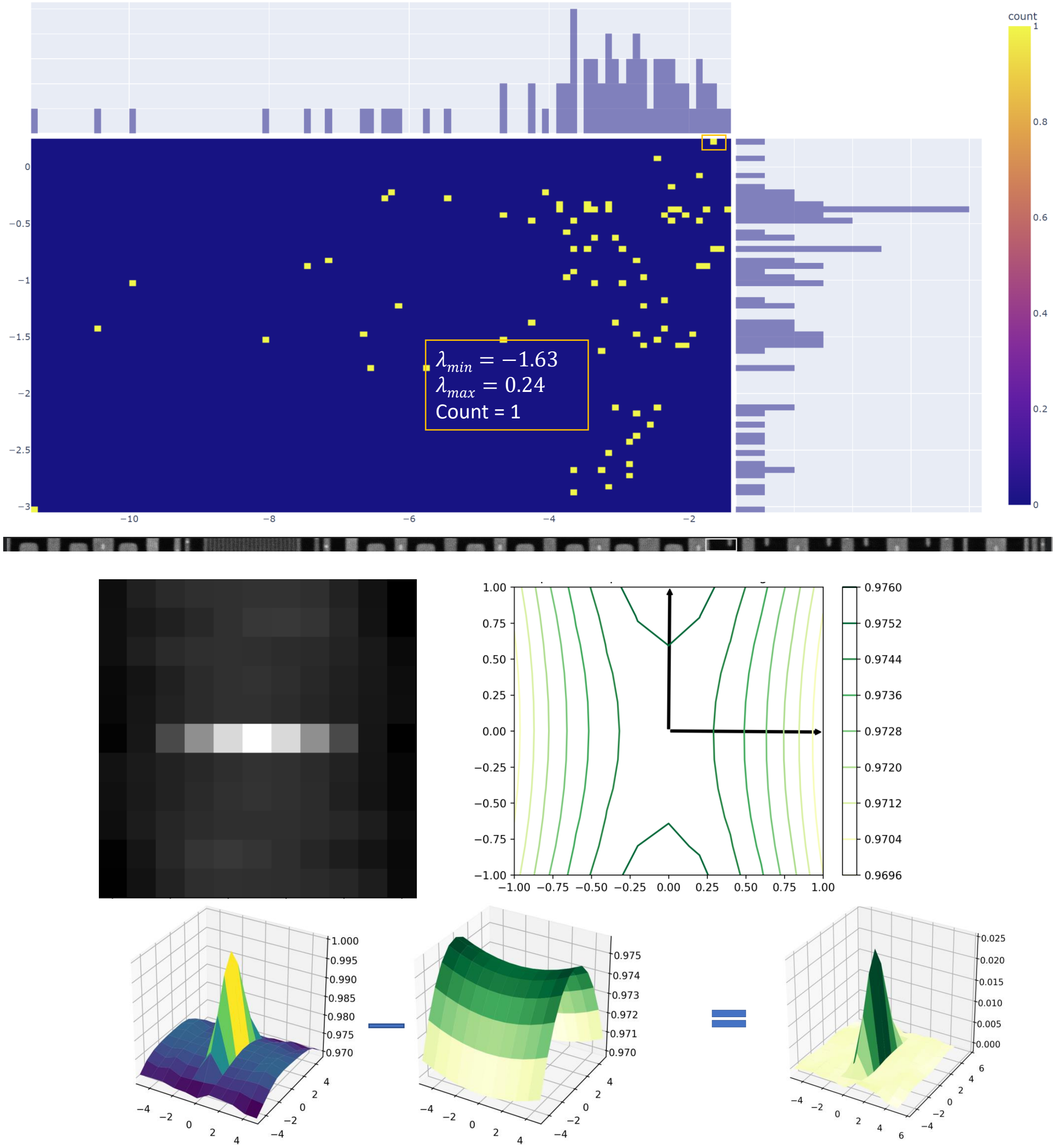

Figure 4.13: Self-Correlation to find numerical landmarks ($R = -0.64$), which indicates one of the worst cases to consider as numerical landmark.

In the next step, the amount of the translation is calculated based on the position of the maxima in the ACS of numerical landmarks. The result after correcting the translation error is shown in Figure 4.14.

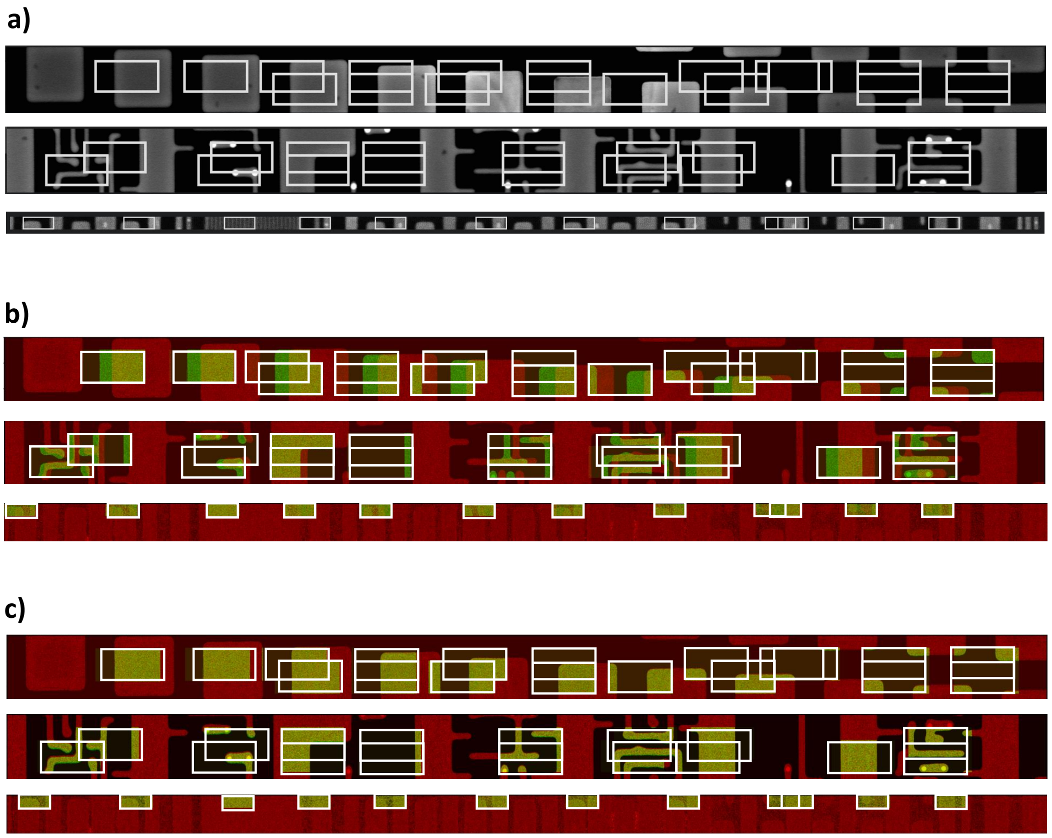

Figure 4.14: Results of correcting translation errors: a) numerical landmarks (outlined in white) after thresholding based on the modified Harris response function, b) uncorrected images superimposed red and green channels, and c) corrected images using the result of polynomial fit. The subimage coordinates are all outlined in white.

We also found that after using any of these methods to estimate local translations, global non-linear transformations could be estimated by fitting those translations to a polynomial of order 2. Since even the most robust NCC-1D estimates still produce outliers, see Figure 4.16, we recommend using ℓ^1 estimates over ℓ^2 estimates, i.e., fitting using a linear program rather than quadratic optimization.

Furthermore, after estimating local translations, it is possible to detect global nonlinear transformations by fitting those translations to a polynomial. Since even the most robust NCC estimates still produce outliers, we recommend using ℓ^1 estimates over ℓ^2 estimates.

Data sets a and b from Figure 4.4 has instrument drift during the course of the scan using SEM. As the global fit is shown in Figure 4.15 for data set a and Figure 4.16 for data set b, the drift is in the x direction, which is the same as the scan direction. The shift is not constant for all selected subimages in both axes. Therefore, fitting a polynomial of order 2 can give us a global fit for the data sets with unexpected or nonlinear drifts.

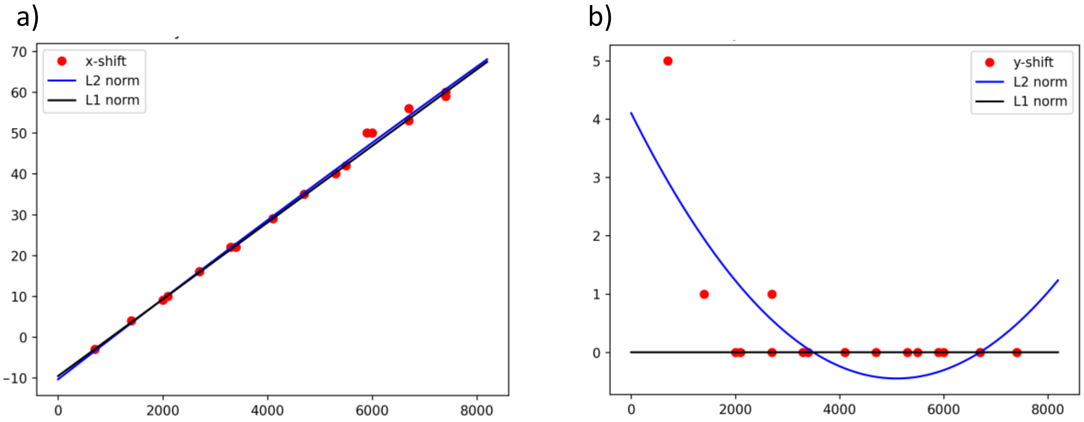

Figure 4.15: This figure illustrates the fit to the numerical landmarks determined using NCC and ACS fitting. Red dots are the $x-$ and $y-$translation values for each of the selected subimages. Outlier rejection is necessary and can be accomplished by fitting a polynomial using an ℓ^1 norm (black line) versus an ℓ^2 norm (blue line). For the definition of the norms, please see Appendix E. In both plots, the horizontal axis shows the subimage location (in pixels) within the horizontal seam of the data set a in Figure 4.4. The left plot shows the estimated correction in the x-direction, while the right shows that for the y-direction.

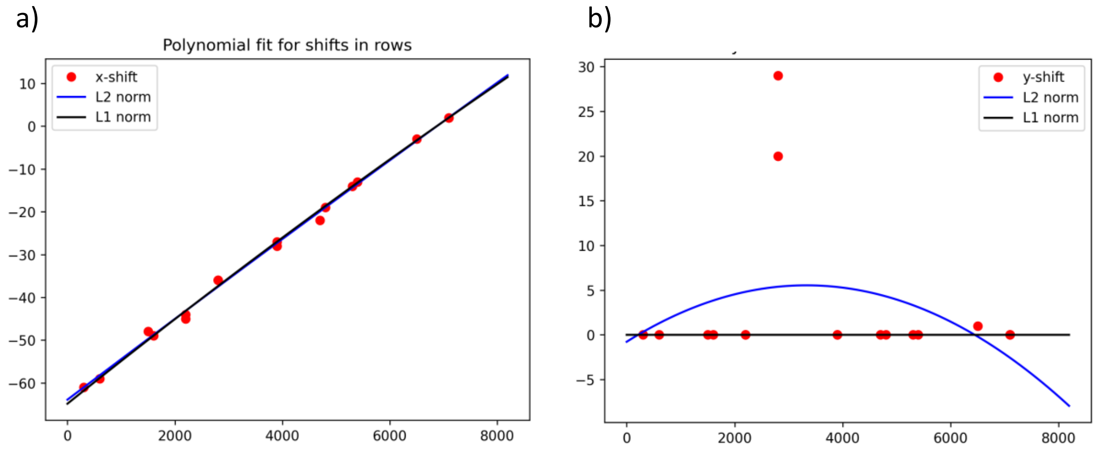

Figure 4.16: This figure illustrates the fit to the numerical landmarks obtained using NCC and ACS fitting. Red dots are the shift values for each of the selected subimages. Outlier rejection is necessary and can be accomplished by fitting a polynomial using an ℓ^1 norm (black line) versus an ℓ^2 norm (blue line). For the definition of the norms, please see Appendix E. In both plots, the horizontal axis shows the subimage number within the horizontal seam of the data set b in Figure 4.4. The left plot shows the estimated correction in the x-direction, while the right shows that for the y-direction.

Data set c from Figure 4.4 exhibits only simple translation, and there is no nonlinear instrument drift during the course of the scan using SEM. As the global fit is shown in Figure 4.17, the shift is constant for all selected subimages in both axes. Therefore, fitting a line can give us a global fit for the data sets without unexpected or nonlinear drifts.

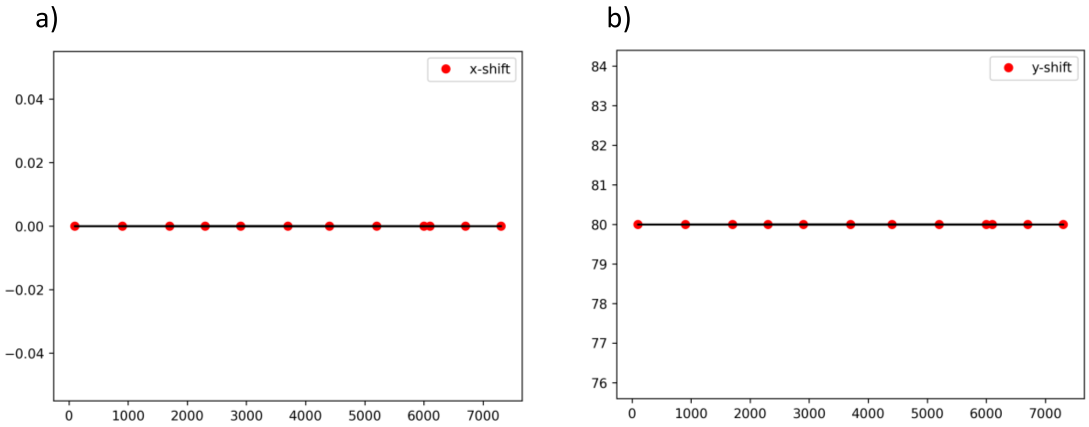

Figure 4.17: This figure illustrates the fit to the numerical landmarks. Red dots are the shift values for each selected subimages after compensating the overlap height for adjacent tiles. In both plots, the horizontal axis shows the subimage number within the horizontal seam of the data set c in Figure 4.4. The left plot shows the estimated correction in the x-direction, while the right shows one for the y-direction.

Using the fitted polynomials, one can calculate the shift for all the pixels of the overlap area of the adjacent tiles, even in the presence of nonlinear drifts and stitch the tiles by compensating for the distortion.

4.2 Projection

In this method, we project an image block onto the x and y axes by column-wise and row-wise summations of the pixel values to obtain two 1D vectors, see Algorithm 3. Considering r as row and c as column, P^r and P^c which are row-wise and columns-wise projections of image I are,

$$P^r_{I_j} = \Sigma_{k=0}^{\#rows} I_{k,j}, \text{for j = columns (row-wise summation)} \tag{4.2.1}$$

$$P^c_{I_i} = \Sigma_{k=0}^{\#columns} I_{i,k}, \text{for i = rows (column-wise summation).} \tag{4.2.2}$$

Algorithm 3 Projection

```
Require: $I$                                  ▷ Image
  $r, c \leftarrow size(I)$      ▷ number of rows(r) and columns(c) of image $I$
  for $j \leftarrow 0$ to $r$ do       ▷ row-wise summation, project on $y-$ axis
      $P^r_{I_j} = \Sigma^c_{k=0} I_{k,j}$
  end for
  for $i \leftarrow 0$ to $c$ do    ▷ column-wise summation, project on $x-$ axis
      $P^c_{I_i} = \Sigma^r_{k=0} I_{i,k}$
  end for
  return $P^r_{I_j}$ and $P^c_{I_i}$
```

Both metal and dielectric materials in our samples are non-uniform. The grain pattern is anisotropic, sometimes looking like grains of rice, aligned along the scan direction of the electron beam. By summing over the rows and columns of each subimage, the projection will be either parallel or perpendicular to the grain pattern. This pattern is different than noise. To validate this, one can choose one subimage from the overlap area, which is pure dielectric, and subtract the projections from each other (after alignment), to isolate the noise from the grain pattern, see Figure 4.18.

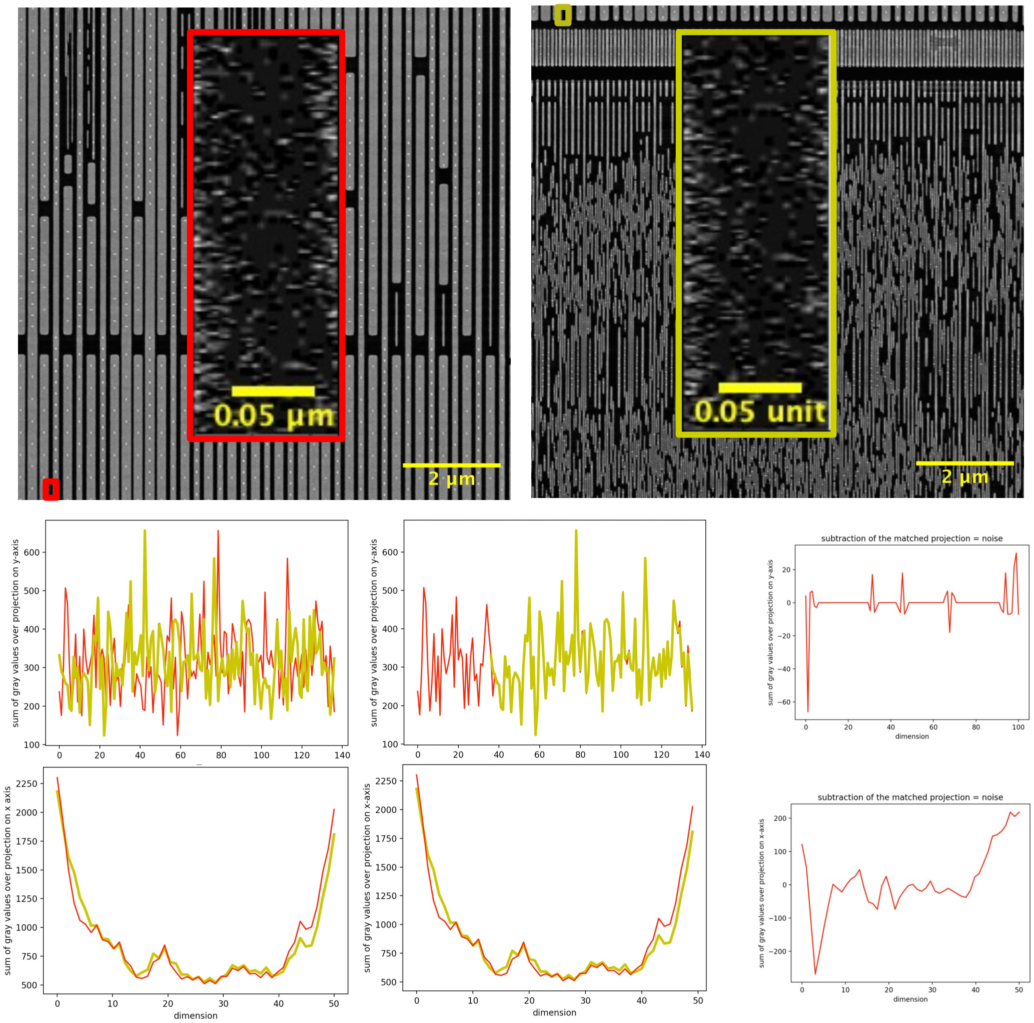

Figure 4.18: A sub-image is taken from the solid part of the SEM image of a semiconductor (dielectric part of the image). Translation values for both the $x-$ and $y-$ directions are calculated using NCC-1D. By subtracting the matched signals from each other, one can calculate the noise, which is different than the grain pattern.

The projection onto 1D increases accuracy. For semiconductor images, estimates of the translation between subimages in different tiles are more accurate when the objective function fits the curves obtained by the summation of rows and columns separately, than when it fits the 2D surfaces.

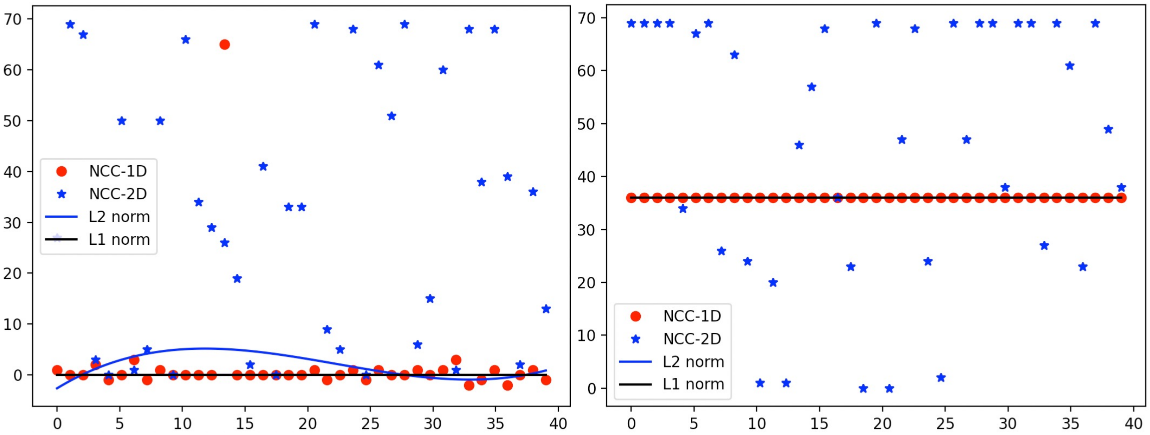

Figure 4.19: This figure illustrates two findings: (1) NCC-1D estimates (red dots) are much better than NCC-2D estimates (blue stars), and (2) outlier rejection is necessary and can be accomplished by fitting a polynomial to the NCC-1D estimates using an ℓ^1 norm (black line) versus an ℓ^2 norm (blue line). For the definition of the norms, please see Appendix E. In both plots, the horizontal axis shows the subimage number within the horizontal seam (blue dotted outline) in the images in Figure 4.21. The left plot shows the estimated correction in the x-direction, while the right shows that for the y-direction. In both cases, NCC-1D estimates are clearly more consistent than NCC-2D estimates for well over half of the subimages. Estimating the y component of the polynomial estimate for the non-linear correction with either norm yields similar results, i.e., the black and blue lines are indistinguishable. On the other hand, the x component (left plot) varies significantly, even though there is only one significant outlier.

Figure 4.19 illustrates this in the case of NCC estimates using the projections (red circles) and not (blue stars). The standard deviations for the x components were 10.46 and 40.63, respectively, which represents a factor of 4 advantage for the projection method. When considering the y component, which contains no outliers in the projection case, the improvement is numerically infinite due to the limits of floating-point arithmetic. We use NCC for the comparison because, as we will see, it is the most accurate.

Projection methods are faster. SSD-1D is, by far, the fastest method for small subimages. NCC-1D is still at least 50× faster than NCC-2D. FFT-1D is also faster than FFT-2D, which is itself fast, but, unfortunately, not accurate. Figure 4.23 shows runtimes for representative subimage sizes. One can formulate SSD, NNC and FFT in $1D$ as follows,

$$FFt(P) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi tP}dt, \text{(1D Fast Fourier Transform)} \tag{4.2.3}$$

$$SSD(P, P') = \frac{1}{2}||P - P'||_{\ell^2}^2, \text{(1D Sum of Squared Differences)} \tag{4.2.4}$$

$$NCC(P, P') = \frac{\sum P * P'}{\sqrt{\sum P^2}\sqrt{\sum P'^2}}, \text{(1D Normalized Cross Correlation)}, \tag{4.2.5}$$

where P and P' are the projections of the images.

NCC-1D is the most accurate algorithm. For semiconductor images, in general, SSD-1D and NCC-1D are more accurate than FFT-1D or FFT-2D, when aligning subimages, as illustrated in Figure 4.21. For images containing brightness variations (probably due to charging), NCC is more robust than SSD; see Figure 4.22. For images containing the types of repeated patterns common in semiconductor images, SSD-1D and NCC-1D are both highly accurate, while FFT-1D was less accurate, see Figure 4.20.

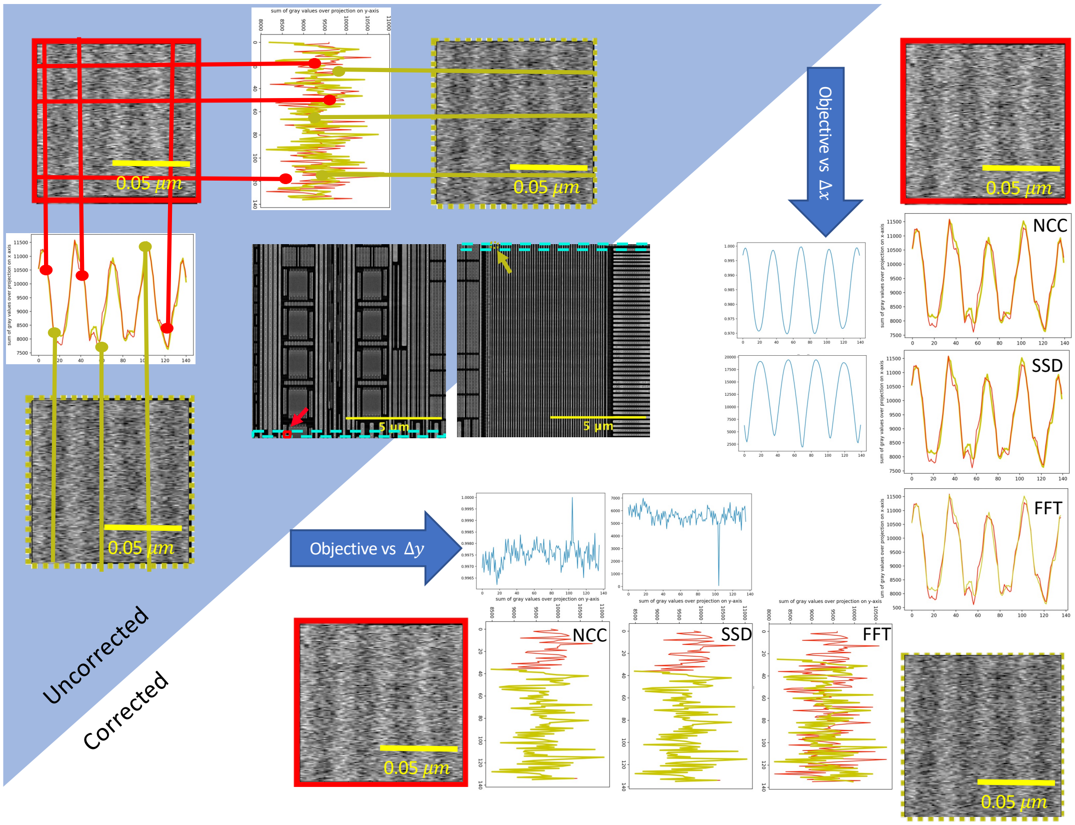

Figure 4.20: A comparison of 1D NCC, SSD and FFT methods on repeated patterns. The images in the shaded blue upper-left triangle are the uncorrected projections along rows (vertical lines) and columns (horizontal lines). The position of the subimages with repeated patterns outlined in red and yellow are shown in the full images (center). The images in the unshaded lower-right triangle illustrate the correction, with the subimage outlined in yellow being translated down approximately 40 pixels. Between the red and yellow images are three plots of the projections after correction using NCC, SSD and FFT fitting. The y-corrections are shown along the bottom of the figure, while the x-corrections are shown along the right of the figure. The FFT is a direct method, but the NCC and SSD are calculated by maximizing (respectively minimizing) an objective. Those objectives are shown above and to the left of the corrected projections. It is also noticeable that even in the presence of repeated patterns, the minima on the SSD objectives are sharper than the NCC maxima, which would normally indicate greater confidence in the estimate. If the SSD estimate were insensitive to variations in brightness, it might be the better estimate.

Polynomial fitting robustly estimates non-linear distortions. Although translation estimates between subimages in overlapping margins depend on local rigid-body transformations and may fail for some subimages, fitting an overall nonlinear deformation to the collection of these local translations recovers non-linear deformations. In Figure 4.24, the individual $x-$ and $y-$translations are fit to a polynomial using ℓ^1 and ℓ^2 norms. The ℓ^1 norm effectively rejects the influence of outliers, as shown by composite RGB images (also in Figure 4.24). For the definition of the norms, please see Appendix E.

One of the greatest advantages of this method is that the grain patterns can be used for alignment. Surprisingly, the grain pattern created by the manufacturing process can be easily distinguished from noise and therefore used for matching in cases where image features are absent. Figure 4.18 shows one example.

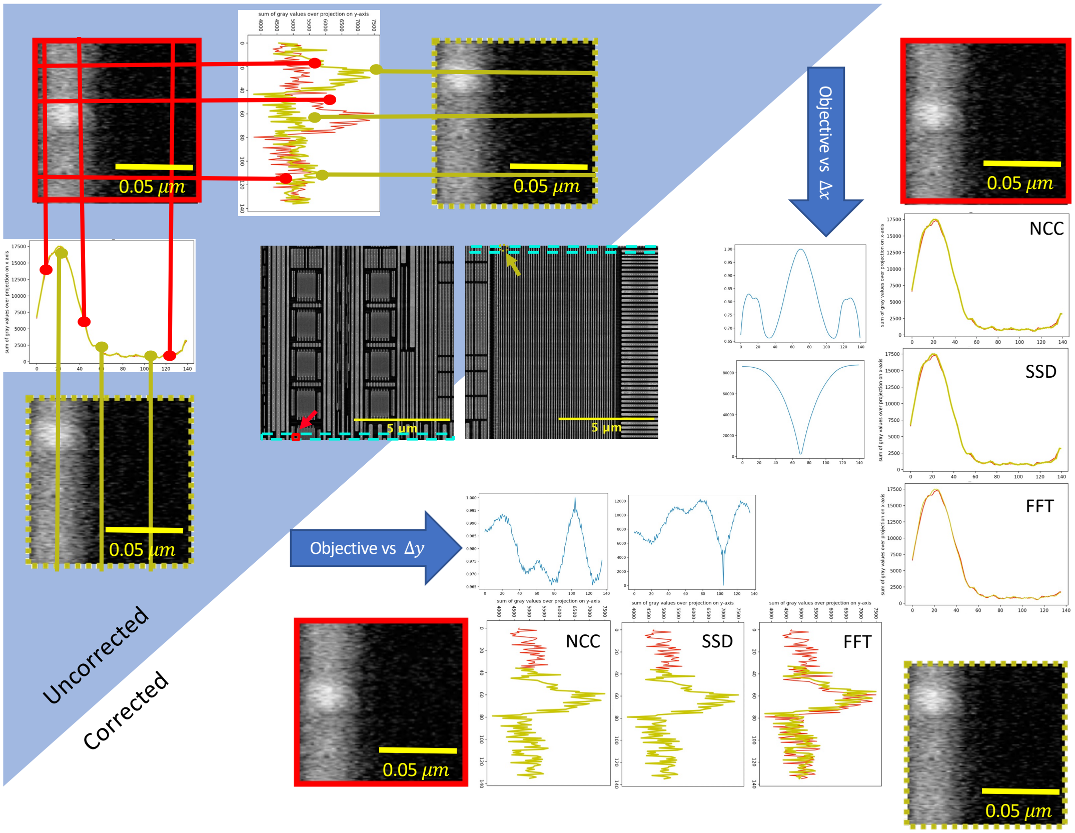

Figure 4.21: A comparison of 1D NCC, SSD and FFT methods. The images in the shaded blue upper-left triangle are the uncorrected projections along rows (vertical lines) and columns (horizontal lines). The position of the subimages outlined in red and yellow are shown in the full images (center). The images in the unshaded lower-right triangle illustrate the correction, with the subimage outlined in yellow being shifted down, which approximates 40 pixels. Between the red and yellow images are three plots of the projections *after* correction using NCC, SSD and FFT fitting. The y-corrections are shown along the bottom of the figure, while the x-corrections are shown along the right of the figure. The FFT is a direct method, but the NCC and SSD are calculated by maximizing (respectively minimizing) an objective. Those objectives are shown above and to the left of the corrected projections. Notice that the minima on the SSD objectives are sharper than the NCC maxima, which would normally indicate greater confidence in the estimate. If the SSD estimate were insensitive to variations in brightness, it might be the better estimate.

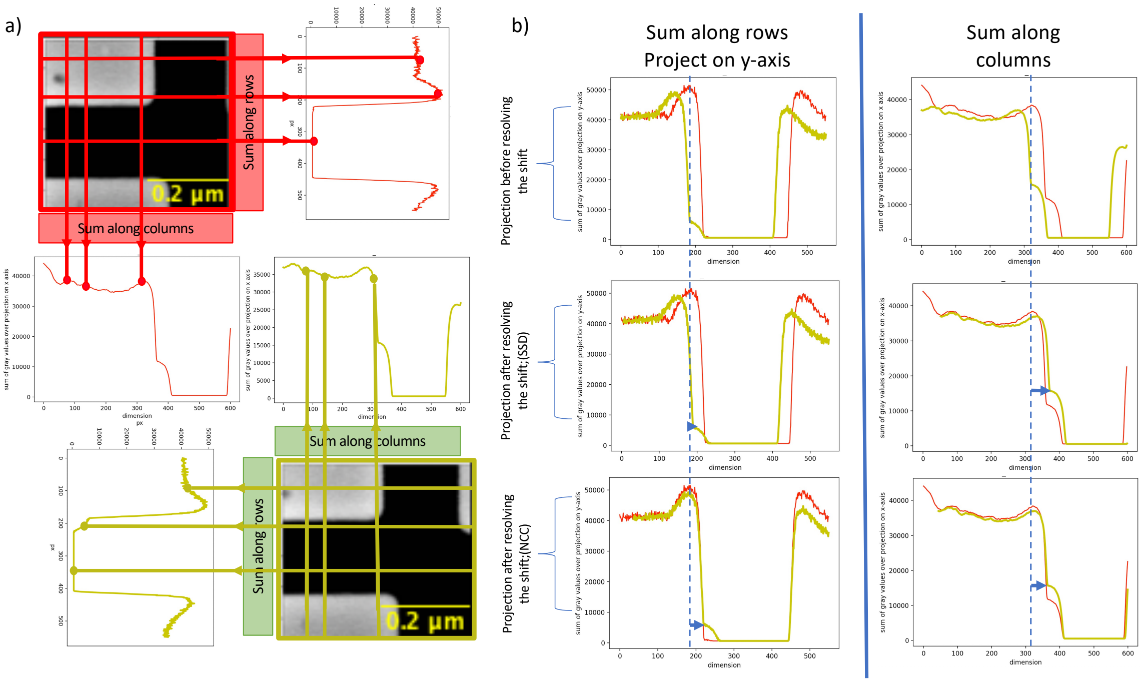

Figure 4.22: An example of the projection method, using the magnified subimages from Figure 3.1. Subfigure a) is a schematic illustrating summation along the rows and columns of the images. Horizontal red and green lines represent summation over the rows (projection on the y-axis). Vertical lines represent summation over columns (projection on the x-axis). Subfigure b) shows the projection result before and after estimating the translation between the (red and green) subimages images. The upper plots show the uncorrected projections, the middle plots present the correction based on SSD, and the lower plots the correction based on NCC. The relative sizes of the corrections are represented by blue arrows based on the reference position indicated by the dotted blue line. The NCC estimates are visibly better than those based on SSD in this example. This pattern of superior performance for NCC was only observed in the presence of charging. Although charging is rare in our data sets, it cannot be eliminated.

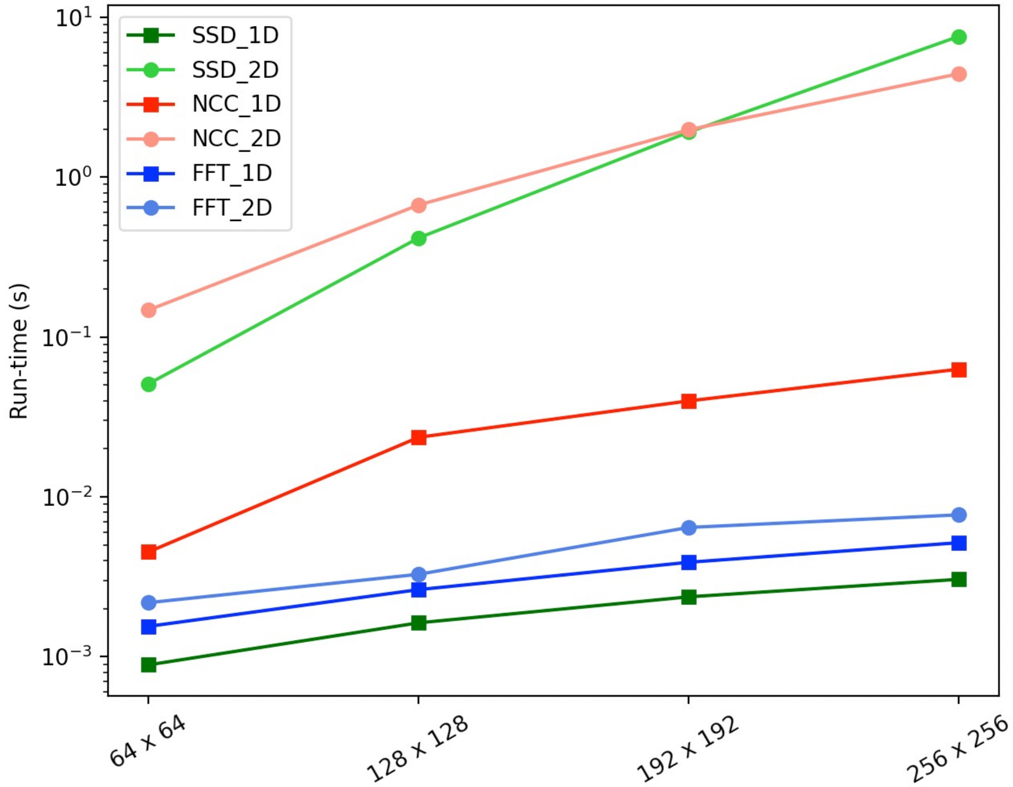

Figure 4.23: Using a semi-log plot, we observe that all methods incorporating projection onto 1D are faster than their corresponding 2D method. For SSD and NCC, the difference is significant. SSD-1D is the fastest, followed by the FFT methods. These three methods are less accurate, at least in the presence of charging. Fortunately, although slower than SSD-1D, NCC-1D is still 50× faster than NCC-2D. The timing was calculated for four representative image sizes (horizontal axis) to show that the real-world performance is consistent with the theoretical and computational complexity and consistent across subimage sizes.

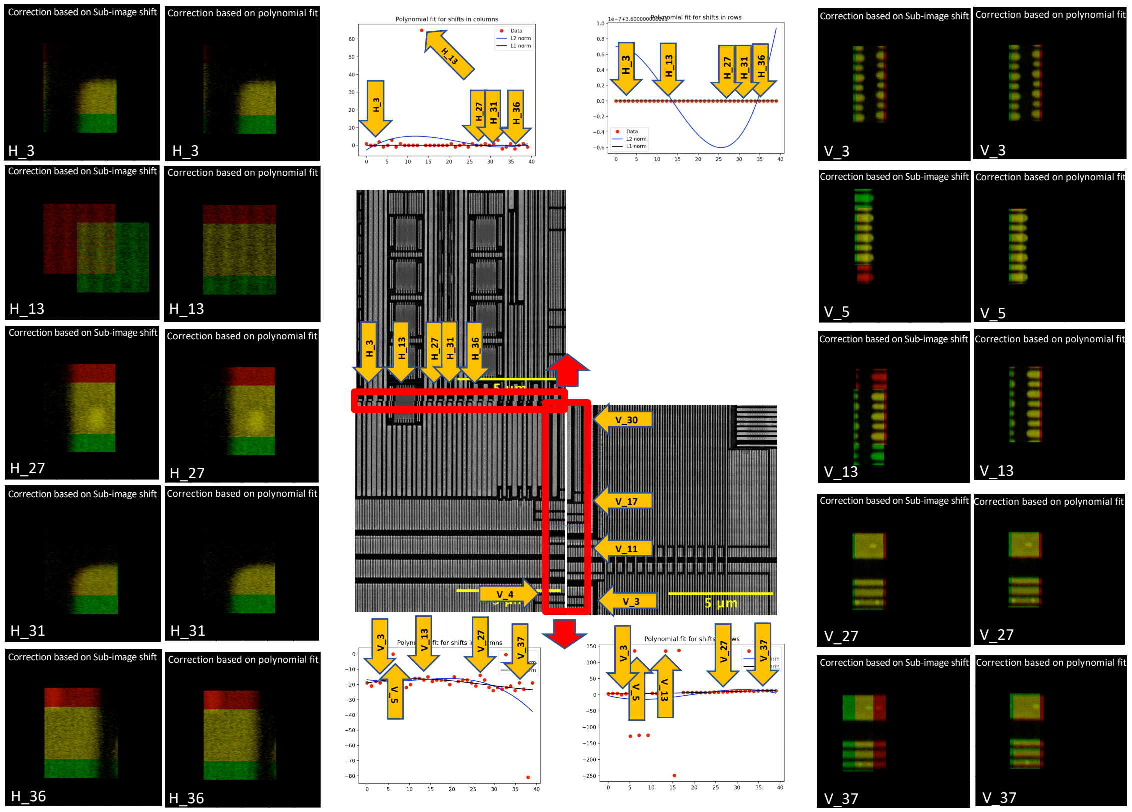

Figure 4.24: A comparison of stitching along two seams, outlined in red (centre). Representative subimages are labelled with yellow arrows indicating their positions in the overlaps. Subimages in the horizontal (respectively vertical) overlap start with “h” (respectively “v”). The same yellow arrows are used in the plots of displacements (top center, $x-$translation on the left and $y-$translation on the right) and (bottom center, similarly). The left two columns of images are from the horizontal seam, and the right two from the vertical seam. In each pair of columns, the images are labelled to show their position in the seam using the same system as the images and plots. Each image is a composite, using the red and green channels for the two images being aligned. In a perfect alignment, all overlapping pixels will be yellow (or black). The left image in each pair is uncorrected, and the right one is corrected using the NCC-1D correction. Notice that h13 and v5 appear to be outliers in the plots, and the composite uncorrected images show large areas of red and green pixels and smaller numbers of yellow pixels.

We compared six methods, conventional 2D and novel 1D versions of SSD, NCC and FFT, with respect to computational cost, accuracy and reliability in the presence of artifacts (e.g., brightness changes due to charging). Although FFT methods can, in theory, detect subpixel translations and benefit from the high efficiency of the FFT itself (e.g., asymptotic computational complexity $\mathcal{O}(n^2 \log(n))$ for FFT-2D versus $\mathcal{O}(n^4)$ complexity for SSD-2D and NCC-2D, for a $n \times n$ subimage), in practice, they had the lowest accuracy, especially in the presence of repeated patterns, so we do not recommend them for stitching semiconductor images.

On the other hand, SSD and NCC successfully handled repeated patterns; see Figure 4.20. Between SSD and NCC, SSD-1D was faster than NCC-1D, but SSD methods may become inaccurate in the presence of charging. In theory, the 1D-projectional methods should be much faster, with computational complexity reduced from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^2)$, but in practice, the observed acceleration was greater for SSD-1D than for NCC-1D, see Figure 4.23. We would still recommend NCC-1D based on its insensitivity to charging.

We also found that after using any of these methods to estimate local translations, global non-linear transformations could be detected by fitting those translations to a polynomial. Since even the most robust NCC-1D estimates still produce outliers, see Figure 4.19, we recommend using ℓ^1 estimates over ℓ^2 estimates, i.e., fitting using a linear program rather than quadratic optimization.

Finally, we were able to observe a clear distinction between noise and grain patterns, which becomes clearer after projecting onto 1D. This can be explained by the reduction of noise (by a factor of $1/\sqrt{n}$), which results from adding up rows (or columns) of pixel values.

In this chapter, we introduced two novel methods to solve $2D$ stitching. One of the methods is based on finding reliable landmarks using “numerical landmarks”, and the other is about reducing the computational complexity of matching by projecting onto each axis and using $1D$ matching. In addition to the reduced complexity, projection makes apparent so-called “grain patterns” which can also be used for matching. We analyzed the accuracy and efficiency of the methods and tested them on multiple data sets obtained from our industrial partner. Once we solve the mosaic in $2D$, our next goal is to stack the $2D$ layers and reconstruct the $3D$ connectivity of the sample, which is discussed in the following chapter.

Chapter 5

3D Reconstruction and Volume Stitching

Recent advances in modern microscopy enable three-dimensional imaging of large specimens at high resolution. Three-dimensional reconstruction is the process of creating three-dimensional models from a set of multiple images. In materials science, this is done by stacking a set of consecutive layers. These layers are formed by taking the image of the sample surface, cutting and removing that surface, polishing it, and taking the image of this new layer. Delayering is done using an ion beam. Focused ion beam (FIB) is a technique in materials science, the biological field, and specifically semiconductors for analysis and deposition of materials. While SEM uses electrons, FIB uses ions for imaging. Ions are produced by a liquid metal ion source that uses metal and heated it to the liquid state to form ions, especially gallium ion source. [50]. There are also instruments developed recently using plasma beams of gas ions, such as xenon [13]. Unlike an electron microscope, FIB is destructive to the specimen. When the high-energy gallium ions hit the sample, they sputter atoms from the surface.

Gallium atoms will also be implanted into the top surface, and the surface will be made non-crystalline. Due to sputtering, the FIB is used to modify materials at the micro- and nanoscale. FIB tools are designed to etch surfaces. Using FIB, atoms in one layer are milled without disrupting atoms in the following layer or without any residual disruptions above the surface. Currently, because of the sputter, FIB is milling the surface at the sub-micrometre length scale [33].

$3D$ $X-$ray tomography and $3D$ Serial Sectioning by FIB-SEM are destructive automated methods that have different resolutions in the $3D$ reconstruction of the specimen without destroying the sample. Each pixel in $X-$ray CT images indicates attenuation properties of a specific material volume. In this regard, the presence of different substances results in representing an average of their properties in the obtained images, which is known as a partial-volume effect. This results in the resolution limitations of $X-$ray CT and could make all material boundaries blurred [25]. So, this conventional $3D$ imaging technique has its own drawbacks as it is not of sufficient resolution. Although $3D$ FIB-SEM could work, it is not time efficient [54].

Generally, we take $2D$ images of $3D$ scenes while reconstructing them in $3D$ is the reverse process, See Figure 5.1 and Figure 5.2.

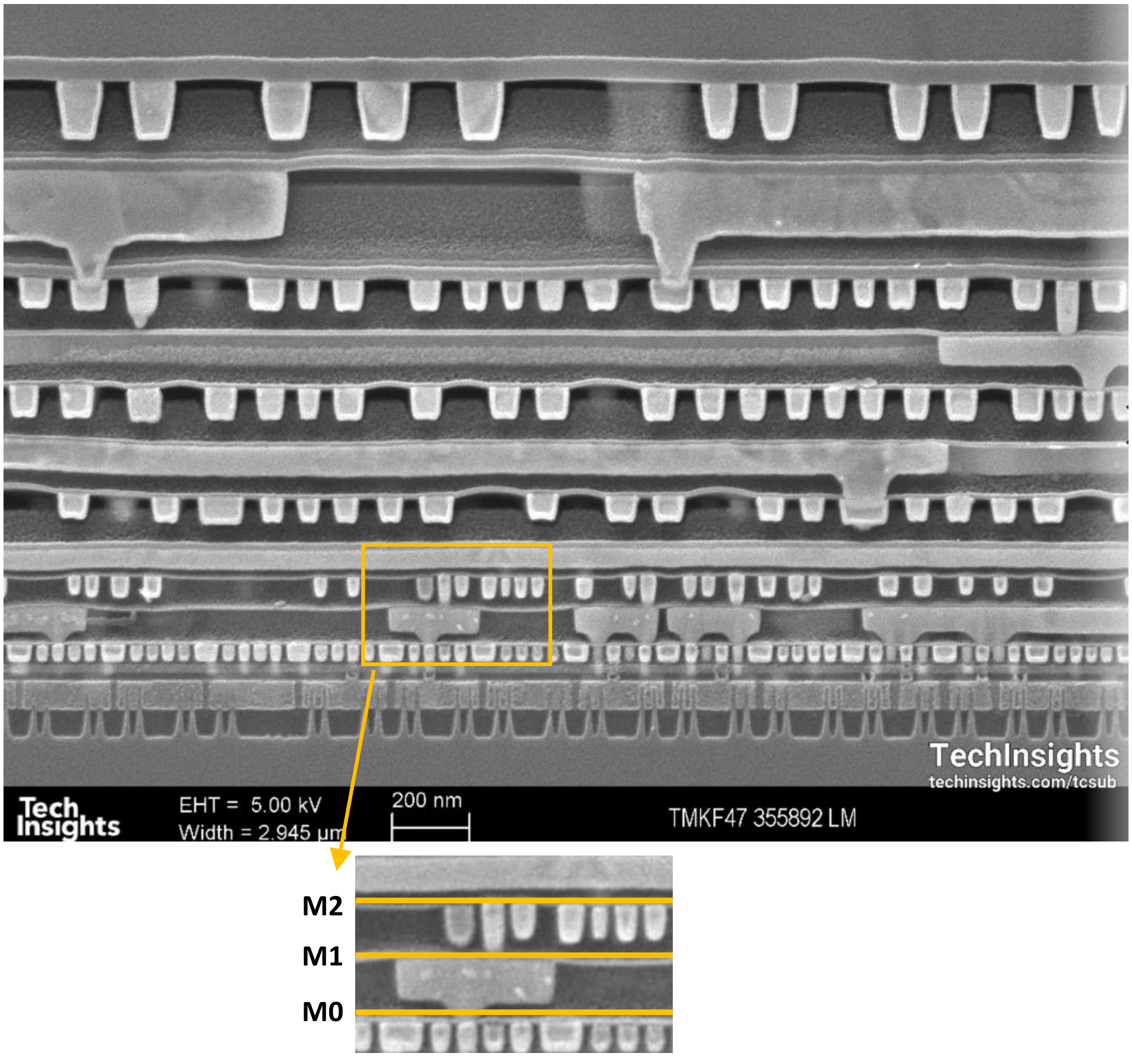

Figure 5.1: Cross-section of semiconductor; vias connect layers. One can cut and polish the sample from the orange lines on the magnified image to take an image of the surface of each layer. Without cutting, M_2 can be captured by even low voltage, and for being able to see M_1 and M_0, higher voltages are needed.

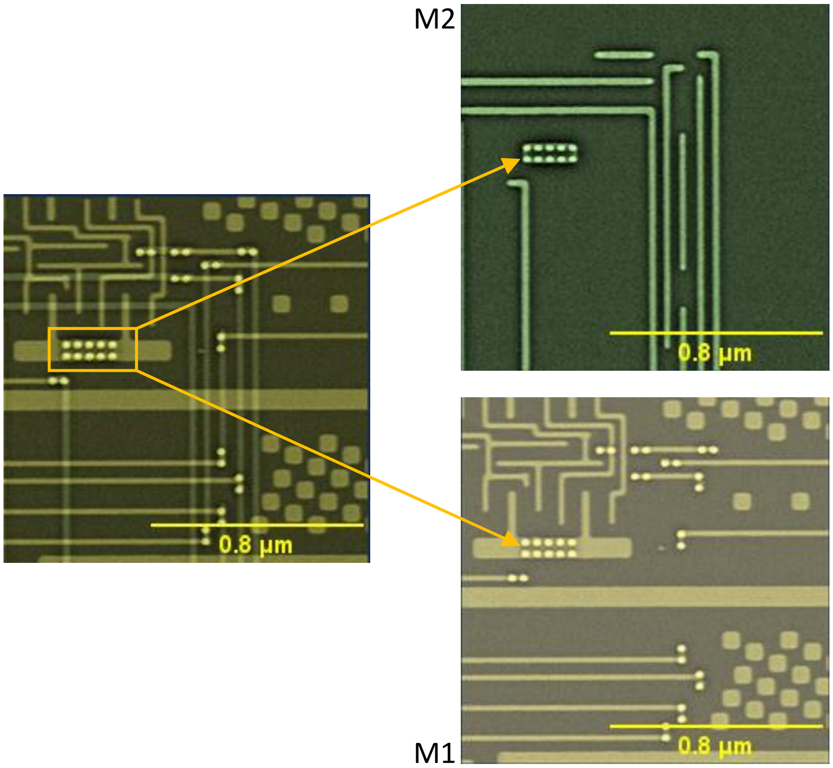

Figure 5.2: Plane view of two layers (SEM images) on top of one another. M_2 is on top of M_1, the yellow circles on M_1 show vias. As the via gets connected upward, it should connect to either a via or wire in M_2. The yellow circles on M_2 show a feasible solution for this connection.

Vias are always connected vertically, and they are only connected to wires and vias; see Figure 5.3. So, there will be a feasible connection area on top of each via in the top layer. The important thing is that when we cut and polish and see via on the surface of a layer, we are sure that this via should be connected to a via or wire in the latest removed layer.

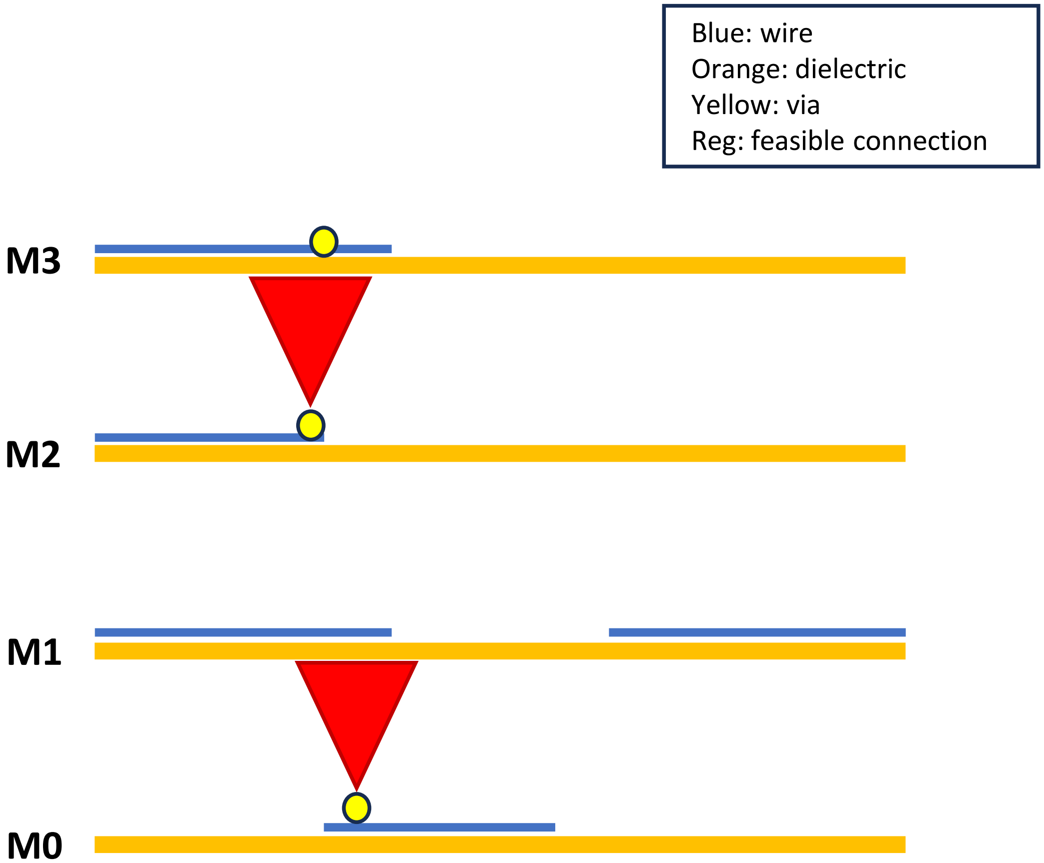

Figure 5.3: Schematic view of possible connections between layers

Different solutions and tools are available for automatic 3D reconstruction of large images. The typical strategy in 2D, which can be used in 3D as well, is to perform a pairwise registration through Normalised Cross-Correlation (NCC) [44], which provides an image similarity score.

Similar to other state-of-the-art tools, the $3D$ reconstruction process has to perform the following main steps:

1. find the relative position between each pair of adjacent stacks (in fact, since in our case tiles are stacked in $z-$axis and parallel along the $x-$axis direction, there might be small rotations, and this means to find the displacement between them);

2. find a globally optimal placement of stacks in the final image space;

3. combine all stacks into a single 3D image and substitutes overlapping regions with a blended version of them.

Our main focus in this dissertation is on finding an optimum registration technique that can be used in aligning the stack of the layers. By applying segmentation algorithms on the plan-view imaging of the alignment process for consecutive layers in a metallization stack, one can extract vias and wires based on what is visible in each image layer. Extracting features is known as feature detection. Then, a map between vias and wires in consecutive layers can be defined and used to align consecutive layers and build a 3D reconstruction of the semiconductor.

There are some attempts in the literature to solve the 3D reconstruction problem, which will be discussed in the following section.

5.1 Registration Methods

There are lots of efforts in the literature to present a fully automatic and robust 3D image registration method for the reconstruction of detailed 3D images, which will deal with complex deformation problems for different types of microscopic images. Image registration is done by either minimizing the differences between consecutive layers or by maximizing the correlation (mutual information) between them. The optimization method and objective functions are the same as what is discussed for registration in $2D$ in Chapter 3.

In 2015, a promising method for $3D$ registration was proposed in [52]. This fully automatic method does 3D registration for all data and consistently produces solid 3D reconstructed objects with fewer discontinuities and deformations than benchmark methods.

In 2019, Kajihara T. et al. proposed a registration method for three-dimensional (3D) reconstruction from biological serial section images [23]. They had a novel registration method using blending rigid transformation to handle non-rigid distortions. Secondly, they proposed a method of selecting target images based on a blending criterion to avoid tissue discontinuity. The last problem in biological samples is to scale change of tissue, so they proposed a scale adjustment method using the tissue area before and after registration.

There are also some applications in medical science for image registration by minimizing a loss function. For instance, in [14], Cao X. et al. proposed a method to register the images from different modalities using loss function minimization. In this way, they measured similarities between different modalities like MRI and CT, which are different medical imaging modalities.

In the following section, one approach based on graph theory is proposed for aligning corresponding features.

5.2 Minimum Spanning Tree (MST)

In this method, a weighted undirected graph of tiles whose edges represent displacements is considered. Weights are computed as the inverse of displacement reliability measures. Reliability measurement is calculated using NCC, yielding $+\infty$ for very unreliable ones. NCC is searching for the maximum correlation between consecutive layers. The smaller the NCC is, the bigger its inverse would be, which increases the weight. These prevent the tree from connecting stitchable tiles (marked as blue squares) from passing through the non-stitchable ones (marked as yellow circles); see Figure 5.4.

If different reliability measures are available, minimum spanning trees are obtained for each direction separately. The marked tile in the figure is a non-stitchable tile that is excluded from any MST path traversing stitchable tiles[9].

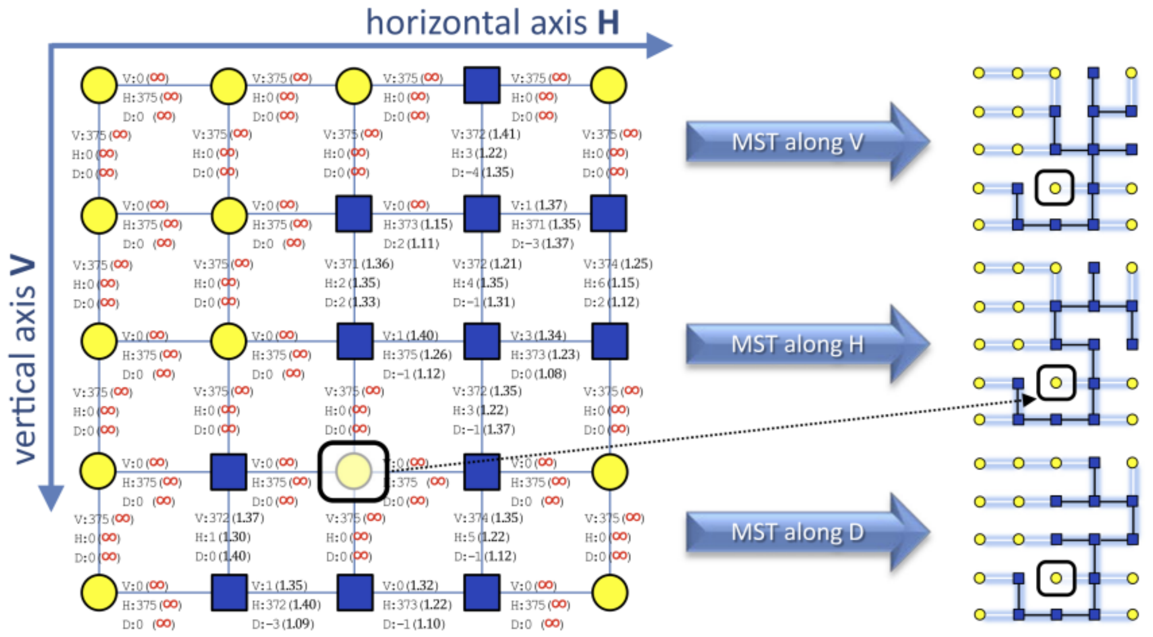
horizontal axis H
vertical axis V
V:0 (∞) H:375 (∞) D:0 (∞)
V:375 (∞) H:0 (∞) D:0 (∞)
V:375 (∞) H:0 (∞) D:0 (∞)
V:375 (∞) H:0 (∞) D:0 (∞)
V:375 (∞) H:0 (∞) D:0 (∞)
V:375 (∞) H:0 (∞) D:0 (∞)
V:375 (∞) H:0 (∞) D:0 (∞)
V:372 (1.41) H:3 (1.22) D:−4 (1.35)
V:375 (∞) H:0 (∞) D:0 (∞)
V:0 (∞) H:375 (∞) D:0 (∞)
V:0 (∞) H:375 (∞) D:0 (∞)
V:0 (∞) H:373 (1.15) D:2 (1.11)
V:1 (1.37) H:371 (1.35) D:−3 (1.37)
V:375 (∞) H:0 (∞) D:0 (∞)
V:375 (∞) H:0 (∞) D:0 (∞)
V:371 (1.36) H:2 (1.35) D:2 (1.33)
V:372 (1.21) H:4 (1.35) D:−1 (1.31)
V:374 (1.25) H:6 (1.15) D:2 (1.12)
V:0 (∞) H:375 (∞) D:0 (∞)
V:0 (∞) H:375 (∞) D:0 (∞)
V:1 (1.40) H:375 (1.26) D:−1 (1.12)
V:3 (1.34) H:373 (1.23) D:0 (1.08)
V:375 (∞) H:0 (∞) D:0 (∞)
V:375 (∞) H:0 (∞) D:0 (∞)
V:375 (∞) H:0 (∞) D:0 (∞)
V:372 (1.35) H:3 (1.22) D:−1 (1.37)
V:375 (∞) H:0 (∞) D:0 (∞)
V:0 (∞) H:375 (∞) D:0 (∞)
V:0 (∞) H:375 (∞) D:0 (∞)
V:0 (∞) H:375 (∞) D:0 (∞)
V:0 (∞) H:375 (∞) D:0 (∞)
V:375 (∞) H:0 (∞) D:0 (∞)
V:372 (1.37) H:1 (1.30) D:0 (1.40)
V:375 (∞) H:0 (∞) D:0 (∞)
V:374 (1.35) H:5 (1.22) D:−1 (1.12)
V:375 (∞) H:0 (∞) D:0 (∞)
V:0 (∞) H:375 (∞) D:0 (∞)
V:1 (1.35) H:372 (1.40) D:−3 (1.09)
V:0 (1.32) H:373 (1.22) D:−1 (1.10)
V:0 (∞) H:375 (∞) D:0 (∞)
MST along V
MST along H
MST along D

Figure 5.4: Minimum Spanning Tree (MST between two slices), H is for MST along the horizontal axis, V is for MST along the vertical axis, and D is for along depth. The weights along each direction are noted beside the nodes. Based on the weights, three MST can be obtained from the graph for each direction. As an example of a non-stitchable node, one of the yellow nodes is marked in the image.

One challenge for this method in our case is that nonlinear deformations in the horizontal plane make it impossible to stitch a tile based on one or two neighbours.

5.3 In Plane Feature detection

Semiconductor metallization images mostly consist of three levels of grey values, light gray, gray, and dark gray. Having an algorithm to detect these values and classify them is helpful. One method is to acquire a histogram of the images. In a plain image without considering noise, we will have three bumps for these three values between 0 and 255. By fitting Gaussians to these bumps, the threshold could be found, and each Gaussian will become the representative for one species.

The Gaussian fitting problem is an optimization problem which any optimization algorithms and solvers could solve. The optimization problem is as follows:

$$\min_{\mu_i,\sigma_i,w_i} \Sigma_{i\in 100 \text{ bins}}(\Sigma_{j=1}^{4}\text{bins}(\mu_j,\sigma_j,i) - H(i))^2, \tag{5.3.1}$$

where H is the histogram of the image.

After solving this optimization with IPOPT called from Pyomo in Python, the fitted Gaussians were obtained in Figure 5.5.

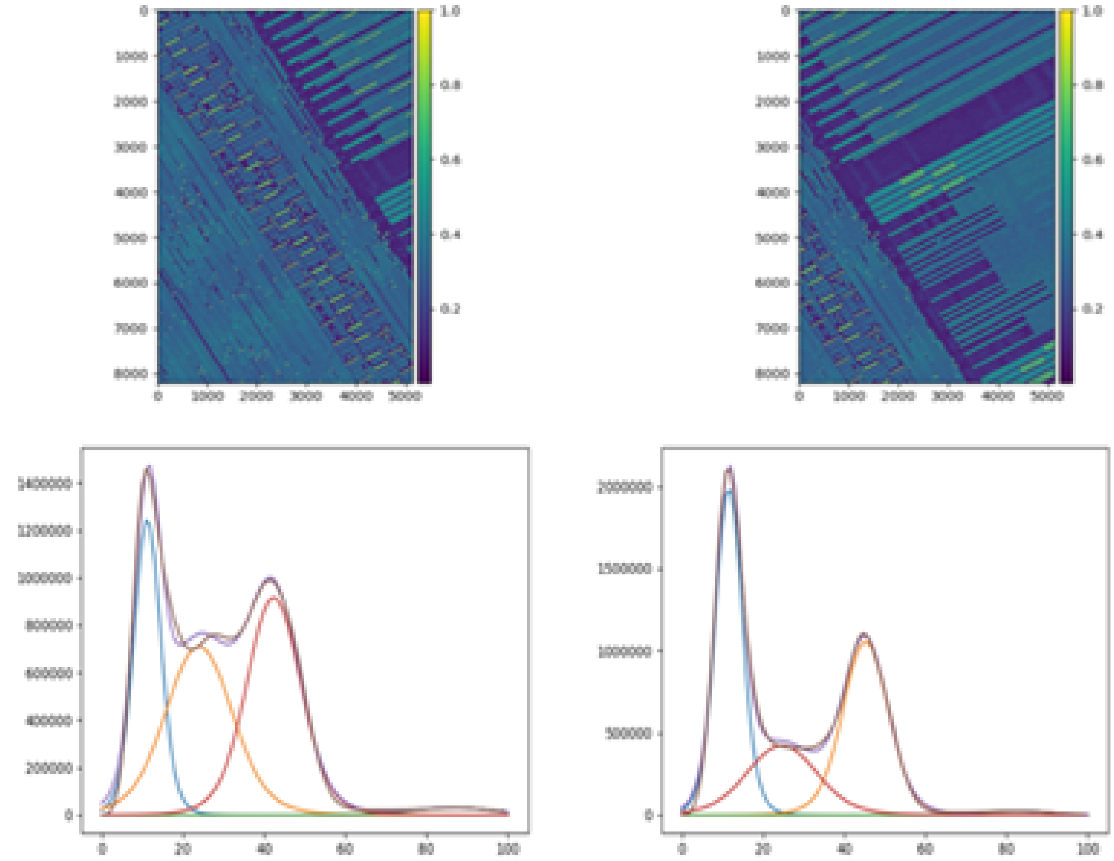

Figure 5.5: Four Gaussians (blue, orange, red, greed) are fit to the smooth histogram (brown) of a semiconductor image. The purple colour shows the histogram of the sum of the fitted Gaussians.

This approach could be used in segmenting images, which are needed for our novel approach in $3D$ stitching. One aspect of our proposed solution is to reduce the computational complexity to reduce our calculation. This way, our calculation is more focused on necessary features in the image and does not consider all pixel values.

An object can be viewed from multiple perspectives using three-dimensional (3D) modelling, which consists of data acquisition, 3D reconstruction and surface reconstruction. Among all, 3D reconstruction is the important one, which can be categorized into three types, i.e. statistical models, discriminative learning models and generative learning models [35]. Statistical models use mathematical theories to extract the spatial and geometric features of 3D data. In this approach, a corresponding matching is performed between multiple layers of 3D data to search for the maximum region likelihood and compute the best match. Through data-driven training, discriminative learning models compute affine transformations with data inference based on spatial coherence of 3D data. On the other hand, generative models use raw 3D data directly to learn latent representations of input 3D data, which are then used later to generate the output samples. In [35], a detailed comparison of the three types of 3D reconstruction techniques is reviewed.

In conclusion, most of the methods in the literature are based on pixel-wise or landmark-based image registration. In Chapter 6, we will propose a novel method which matches via positions to metalization pixels to align consecutive $2D$ images obtained from semiconductor $z-$stack layers. Matching isolated positions to pixel values is novel where all other methods match positions to positions (if using landmarks) or pixel values to pixel values.

Chapter 6

Novel Approach to Volume Stitching

$3D$ reconstruction is about using 2D images to analyze and interpret micro-object surface properties. The anatomic surface, which is obtained from the 3D shapes of SEM micrographs, gives quantitative measurements and informative visualization, which can benefit various branches of science, like biology and materials science. In this research, we developed an algorithm to reconstruct the 3D structure of the sample using the 2D SEM images of the layers of the sample, see Figure 6.1.

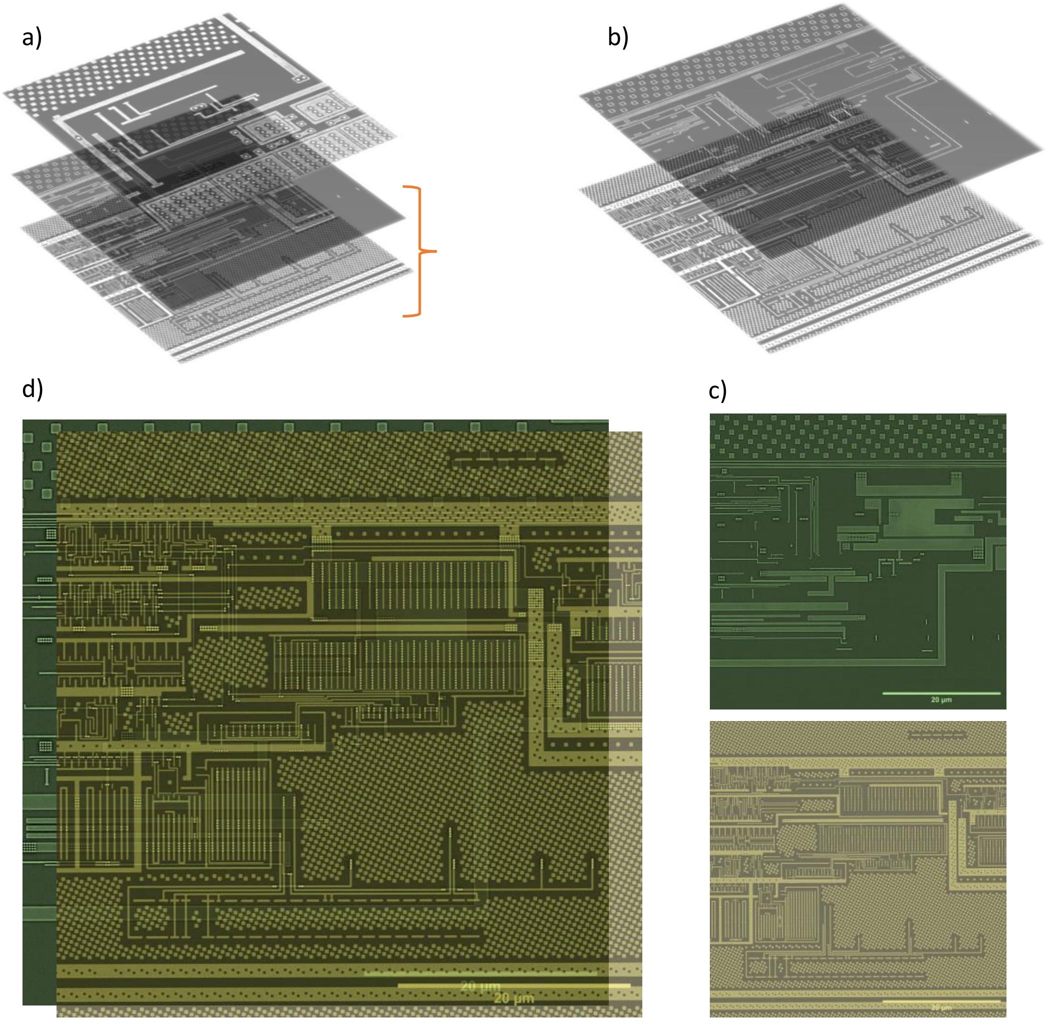

Figure 6.1: Stack of multiple layers of a semiconductor sample is shown. a) shows the stack of 3 layers, b) shows the two selected layers, c) is the plane view of the selected layers, and d) shows moving the M2 (yellow) layer on top of the M1 (green) layer to find the match.

The images have two major components: metal and dielectric (e.g. silicon), and three visible structures: dielectric without metal, metal wires under a layer of dielectric, and metal vias (through-plane wires connecting different layers), see Figure 6.2. Vias always run vertically, connecting to either a via or a wire in the upper layer. The challenge, as mentioned in chapter 5 and Figure 5.3, is that above each via, there are several possible connecting points (either via or wire). In most cases, like Figure 5.2, wires are close together, so connecting them can be tricky.

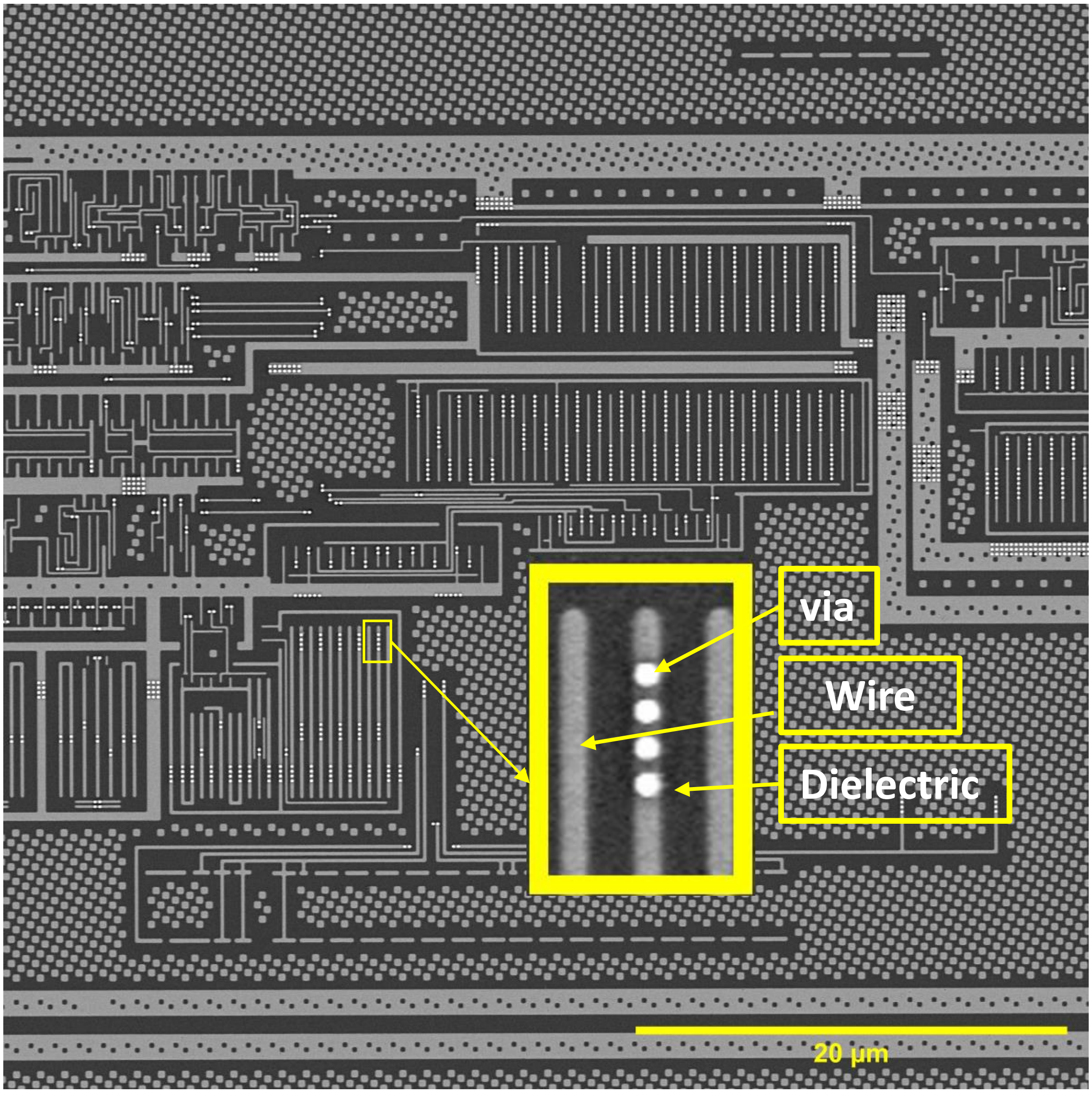

Figure 6.2: Three visible structures in semiconductor images

In the following, we describe our algorithm using the two layers from the stack of layers, which is shown in Figure 6.1. Figure 6.3 shows the two layers $M1$ (bottom layer) and $M2$ (top layer).

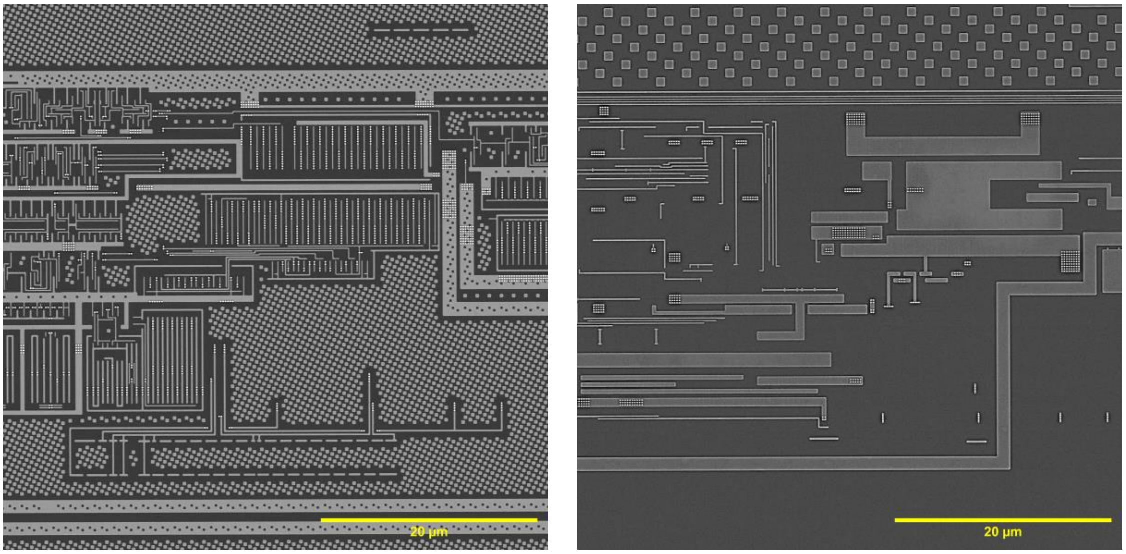

Figure 6.3: Left shows $M1$ (bottom layer), and right shows $M2$ (top layer) from a set of layers that are imaged from semiconductor metallization layers using SEM.

Considering computational complexity and time, one can use parallelization. We divide the layers into small subimages by putting a 4 × 4 grid on each layer, see Figure 6.4. So, each subimage in $M1$ is going to get connected to its corresponding subimage in $M2$.

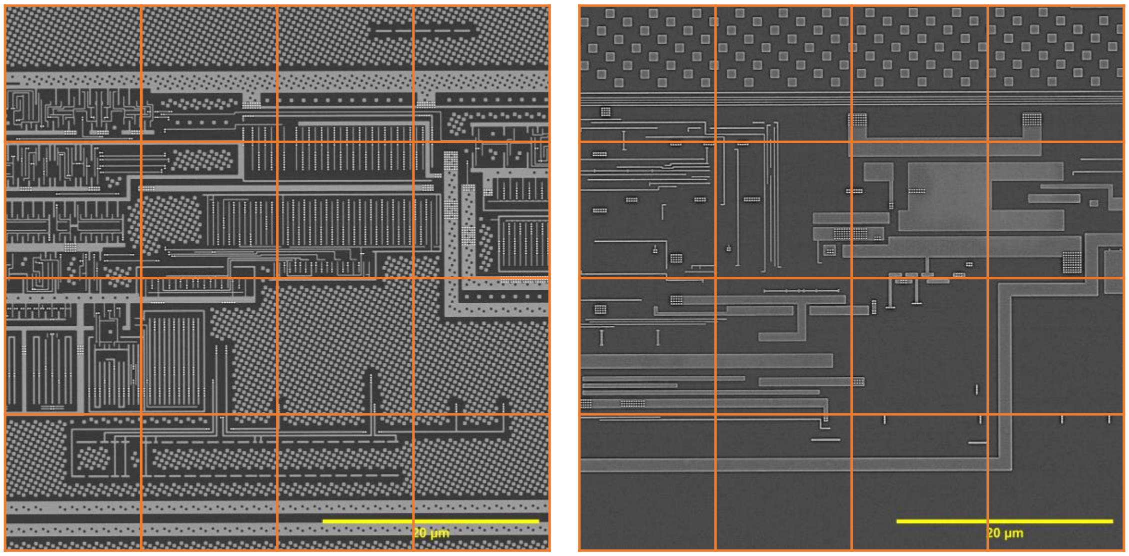

Figure 6.4: 4 × 4 grid on the layers in Figure 6.3, which are picked from the stack of layers.

As mentioned, vias are connecting vertically, i.e., when a via in $M1$ layer is detected, it is going to connect to either via or wire in $M2$ layer. This clarifies that vias connect to the metal part of the top layer and not dielectrics. Based on this knowledge, one can segment metals in the top layer as both via and wire are metal. Figure 6.5 shows the metal segmentation of the $M2$ layer with a 4 × 4 grid for parallelization. This segmentation considers all wires and vias as metal, i.e., vias from the $M1$ layer connect to the metal segmentation in the $M2$ layer.

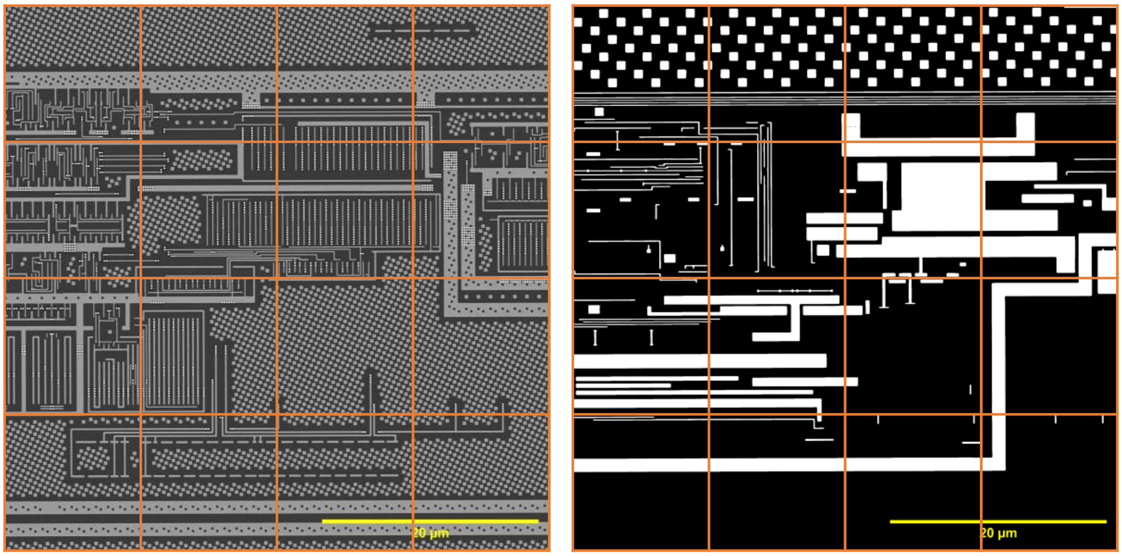

Figure 6.5: Metal segmentation of the $M2$ layer with a 4×4 grid.

Connecting the blocks may seem easy by looking for the metals on top of each via and connecting them, but there are challenges here. In subimage alignment, one can choose the yellow sublayers marked in Figure 6.6.

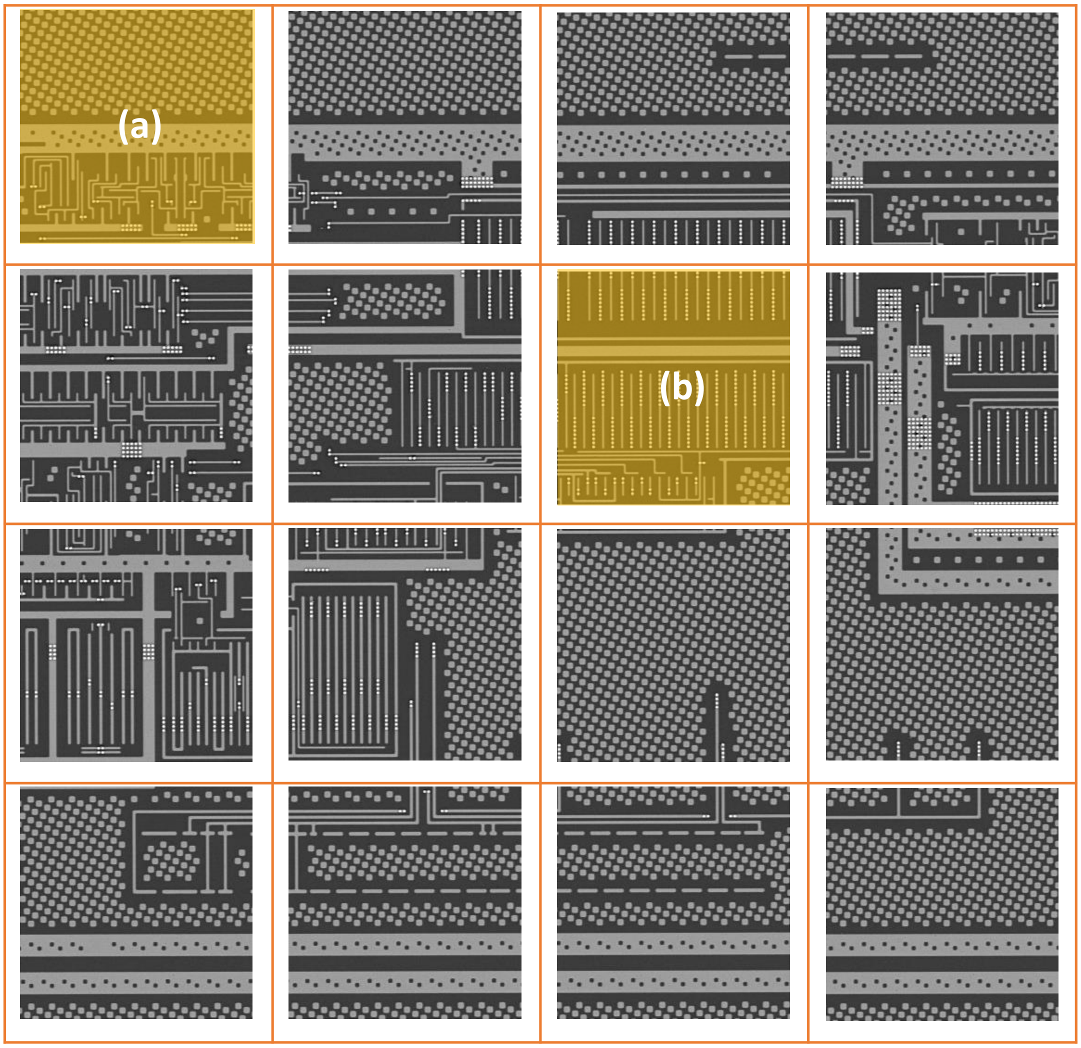

Figure 6.6: (a) and (b) are two selected sublayers from the grid on the $M1$ layer, which its vias are going to get connected vertically to the $M2$ layer.

We find the correlation between layers not based on the intensity or pixel values but based on the important features in layers, which makes our approach not intensity-wise but feature-wise. In $M1$ vias are the through-plane wires connecting different layers. Knowing vias are connecting vertically, the only necessary feature in $M1$ is via, which in Figure 6.7(1) the $M1$ segmentation based on via is shown for subimage

(a) in Figure 6.6. Segmentation is done by putting a threshold on the M1 layer. Vias are the brightest feature in semiconductor SEM images, as the amount of metal in those coordinates is higher than in other parts of the imaged layer. So, the threshold is near to the brightest gray value in the layer.

Each via has the size of approximately 10×10 pixels. Instead of using all via pixels, which increases the calculation time to find the correlation between the $M1$ and $M2$ layers, we segment the vias and find their centroids using contours. Contours can be explained simply as a curve joining all the continuous points (along the boundary), having the same colour or intensity. It is also known as a blob, which is a group of connected pixels in an image that shares some common property. The contours are a useful tool for shape analysis and object detection and recognition; see Figure 6.7(2). Step (3) in Figure 6.7 shows the centroid of each blob is found to reduce the number of coordinates which are used in finding the maximum correlation between the via layer ($M1$) and the metal layer ($M2$).

Having the centroids' coordinates, instead of calculating the correlation for all pixels, only those coordinates are considered in the calculation. By finding the maximum correlation and its position, which is equal to the shift in the x and y directions, the two layers are aligned. Each layer is assigned to one channel of RGB, and the aligned layers are shown in Figure 6.7 (4). The whole process is summarized in Algorithm 4

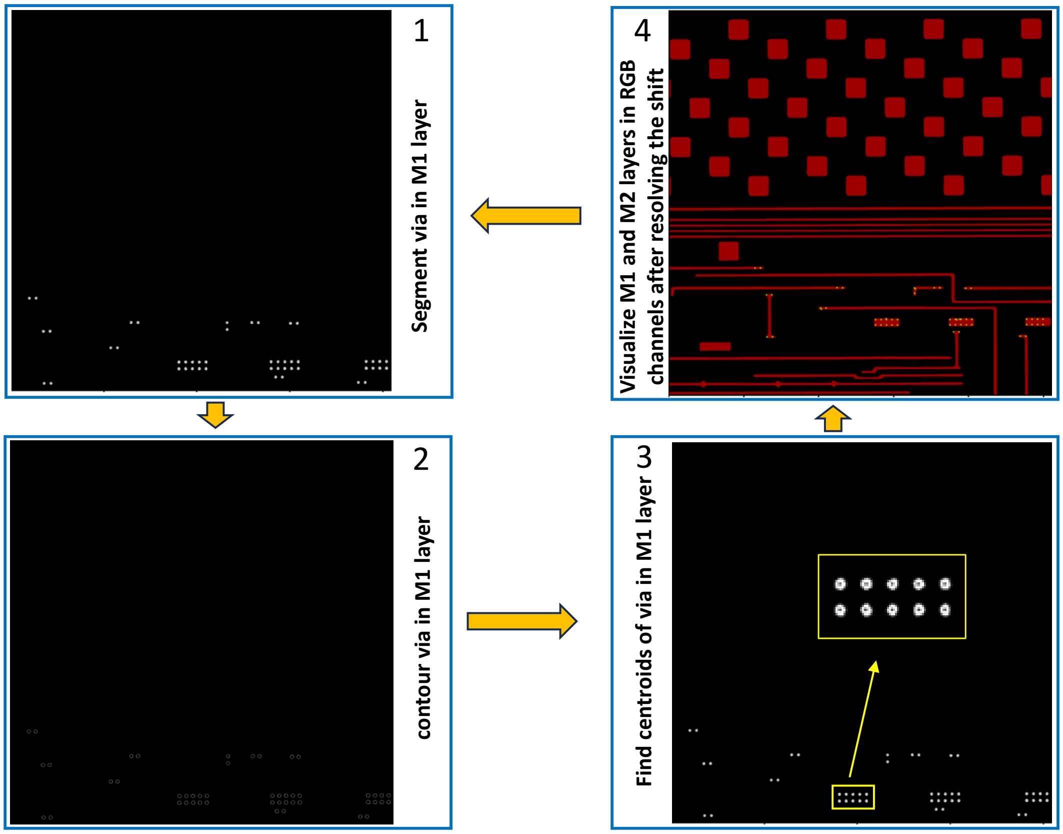
1
Segment via in M1 layer
2
contour via in M1 layer
3
Find centroids of via in M1 layer
4
Visualize M1 and M2 layers in RGB channels after resolving the shift

Figure 6.7: Feature detection steps to find the input coordinates for finding the correlation. (1) Via segmentation of the subimage (a) in Figure 6.6. The threshold is set as 250. The vias are the brightest features in the layers as the metal density in those coordinates is higher, so they are shown brighter in SEM images. Gray values above 250 detect vias in our data set, (2) Contours of vias after thresholding in step (1), (3) Centroids of the contours are plotted. The yellow box magnified a group of vias to clarify how the centroids, (4) aligned subimages which, are taken from $M1$ and $M2$ layers. Each layer is assigned to one RGB channel after resolving the shift, so the connection is shown.

We are dealing with big data sets. It is not only a matter of working with big images with a high amount of data, but a high number of layers which need to be stacked. We tried to refine the algorithm by lowering the computational complexity as well as improving the accuracy.

Algorithm 4 Aligning metalization layers

```
Require: $M_1$                                  ▷ First layer, source of vias
Require: $M_2$                                  ▷ Second layer, wires receiving vias
Require: $t_1$                                  ▷ Threshold to Segment via in M1 layer
Require: $t_2$                                  ▷ Threshold to Segment metal in M2 layer
Require: $\sigma$                               ▷ smoothing parameter
Require: $g$                                    ▷ Grid size g × g
  $V \leftarrow \text{Segment}(M_1, t_1)$       ▷ Via segmemtation M1
  $C \leftarrow \text{contour}(V)$              ▷ Contours of vias
  $P \leftarrow \text{centroid}(C)$             ▷ Set of centroids of vias
  $\mathcal{P} \leftarrow \text{divide}(P, g)$  ▷ $g^2$ subsets of via centroids
  $S \leftarrow \text{Segment}(M_2, t_2)$       ▷ Metal segmemtation M2
  $S' \leftarrow \text{smooth}(S, \sigma)$      ▷ Gaussian filter
  for $P \in \mathcal{P}$ do                    ▷ Iterate through grid of subimages
      $\Delta_P \leftarrow \text{argmax}_{\delta_x,\delta_y} \Sigma_{(x,y)\in P} S'(x+\delta_x, y+\delta_y)$   ▷ Maximize vias under wires
  end for
  return $\Delta$
```

To reduce the computational complexity, the proposed problem needs some preprocessing to reduce the amount of data to accelerate our calculation and make our computation efficient. The images are reduced to vias and then to the centroids of the vias as single points. As a result, we compare a set of points against a set of images. Also, to preserve the image features on the borders (vias and wires), we used the information of the neighbouring pixels. In this regard, the Fourier transform of the M2 layer is taken and a Gaussian filter is applied. Using a blurred version of the metal layer, we make the objective function smoother, considering neighbouring pixels and making it less likely to find the match to be on the edge of the metallization.

Another criterion is to increase the accuracy. As the size of the subimages or the number of centroids is not enough to give us the best match, we can see several local maxima, which leads to a wrong solution for the NCC. Original images are $8K \times 8K$, and each subimage's size after putting a grid is $2K \times 2K$. In the subimage (a) in

Figure 6.6, there are 14 coordinates of centroids which are recorded in a data file. In this regard, we reduce our correlation calculation on a block of size $2K$ to 14 points. The coordinates are recorded in a data file to only use those coordinates for calculating the correlation and not the rest of the pixel coordinates. In some blocks, the best match can happen in different positions, and that causes several local maxima and may lead to a wrong alignment, see Figure 6.8. This can be seen as an outlier in our $x-$ and $y-$ coordinate lists in the Excel sheet.

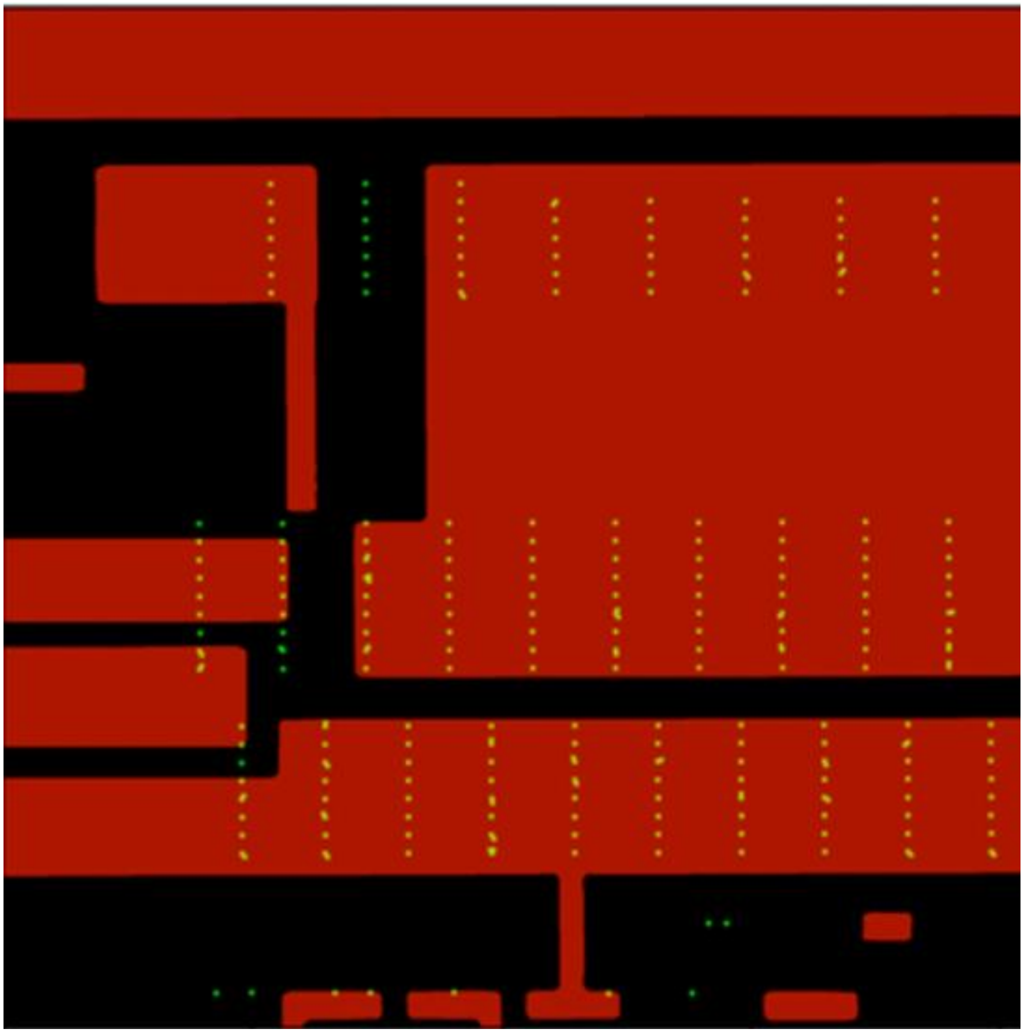

Figure 6.8: Aligned subimages from $M1$ and $M2$ layers in RGB channels. The shift value appeared to be an outlier in the correlation calculation.

To solve this problem, we use two methods: (1) increases the size of the search area in the Metal layer ($M2$). In this method, we move the block from the $M1$ layer of size 2k on an enlarged block from the $M2$ layer of size $3K$ by adding $\frac{1}{4}$ of the size of the neighbouring blocks to the current search area, (2) initializing our search for max correlation on the median of the list or correlations and again applying NCC around the median for each sublayer. We chose the median as the outliers are so large that other statistical center measurements would be affected by the outliers.

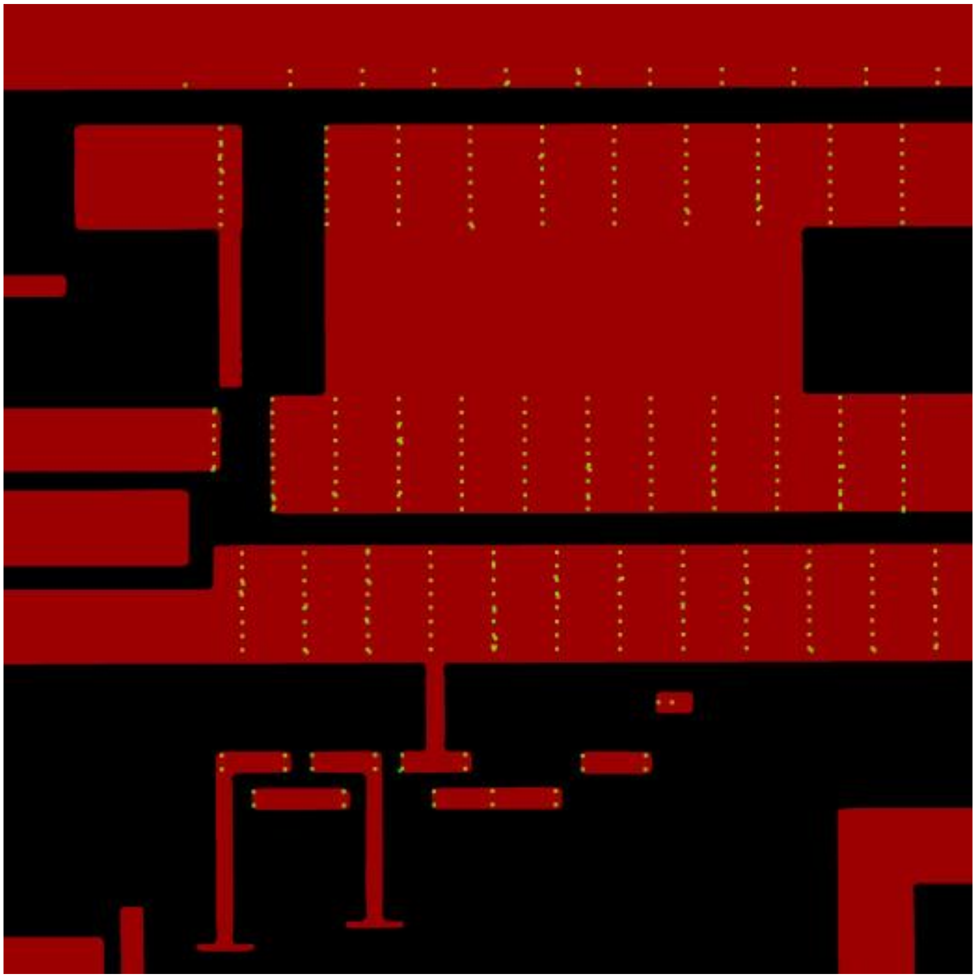

Figure 6.9: Resolve the shift with higher accuracy after increasing the search area. This approach is most applicable when there is a drift or change in the position of the beam of sample from layer to layer.

It is obvious that without using the statistical center measurements, we can find the shift values mostly correct by increasing the search area in the metal layer. In this specific layer, the last row of blocks does not contain any via or a very small number of them, so they produce outliers as expected. Figure 6.10, Figure 6.11, Figure 6.12, and Figure 6.13 are showing the steps which are shown in Figure 6.7 for the whole grid after increasing the search area in the $M2$ layer.

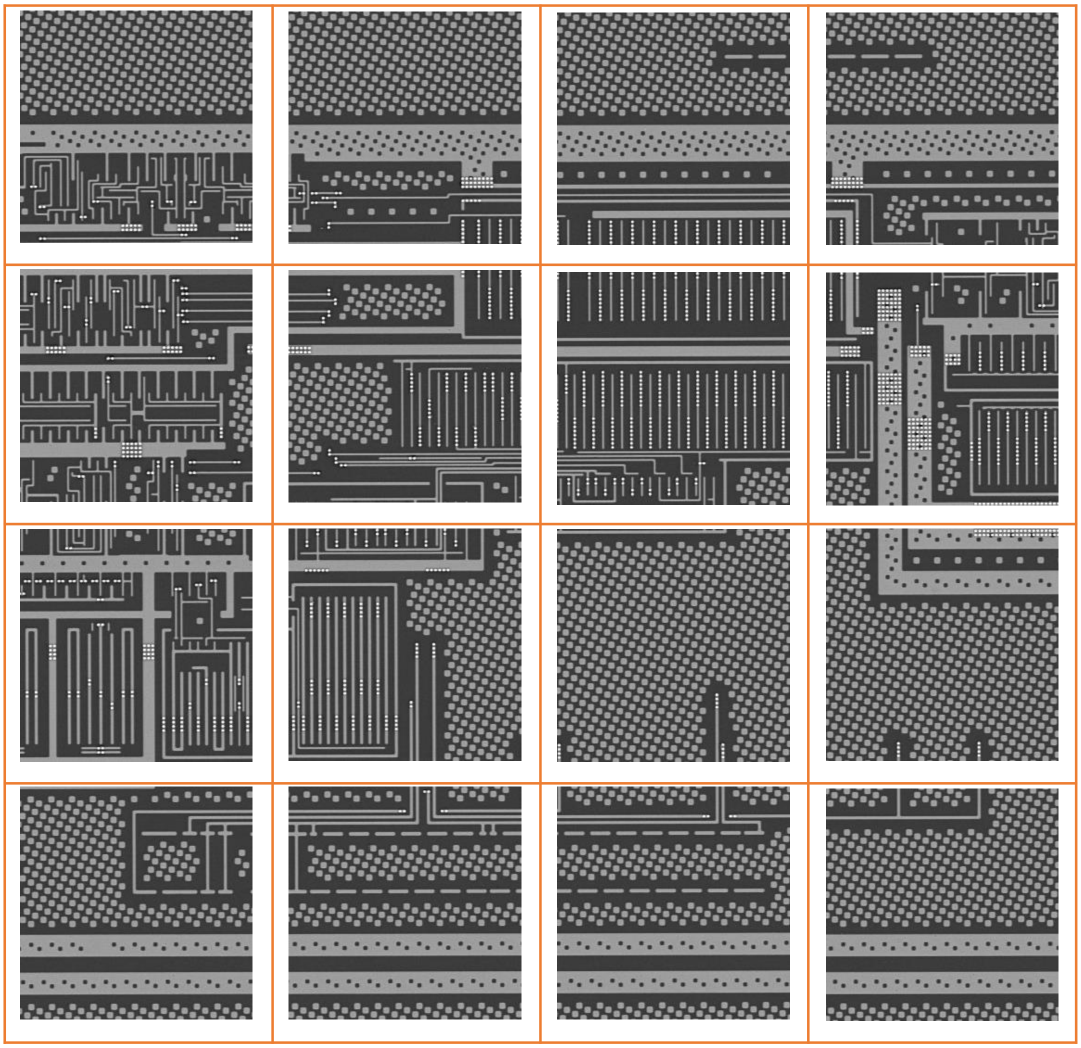

Figure 6.10: Whole grid in $M1$ layer.

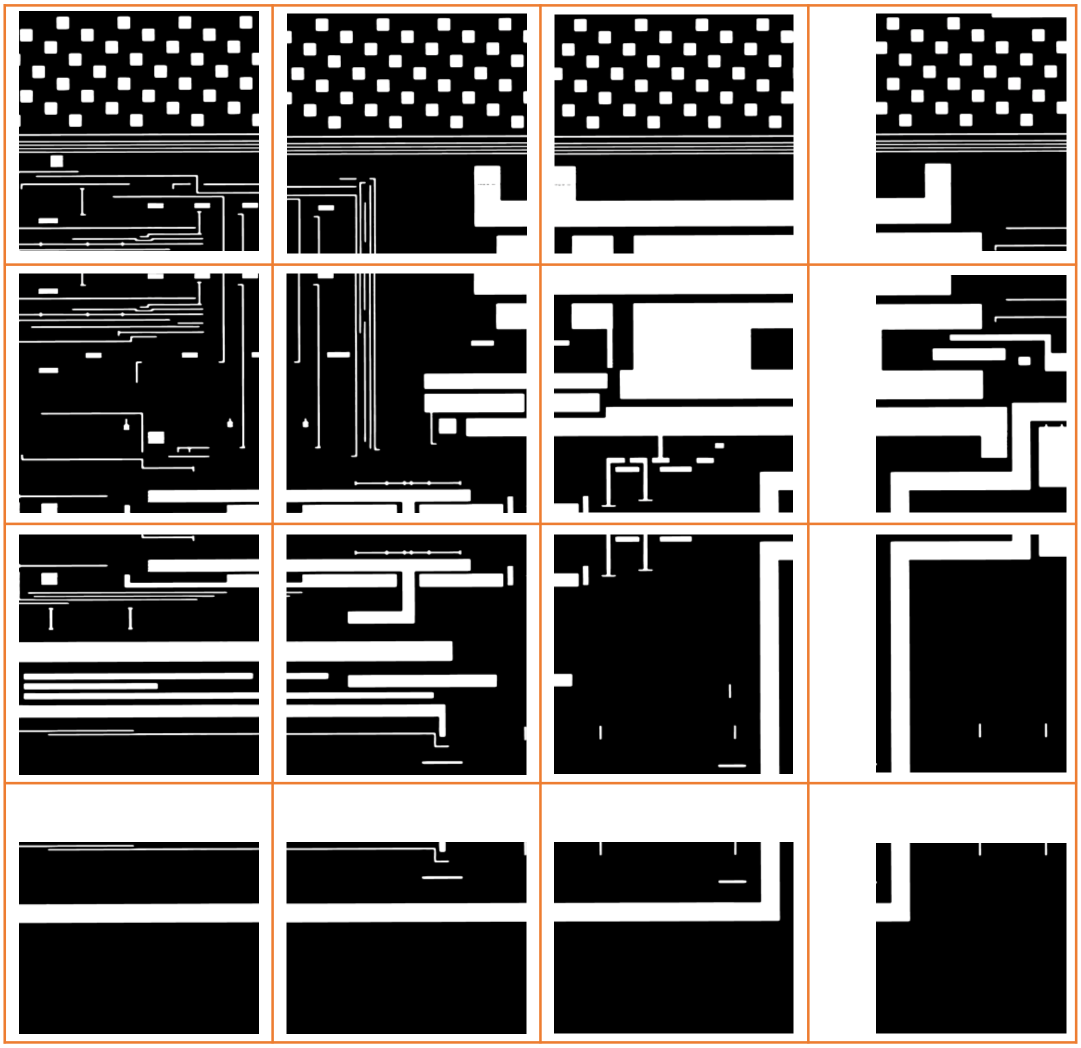

Figure 6.11: Whole grid in $M2$ layer after metal segmentation.

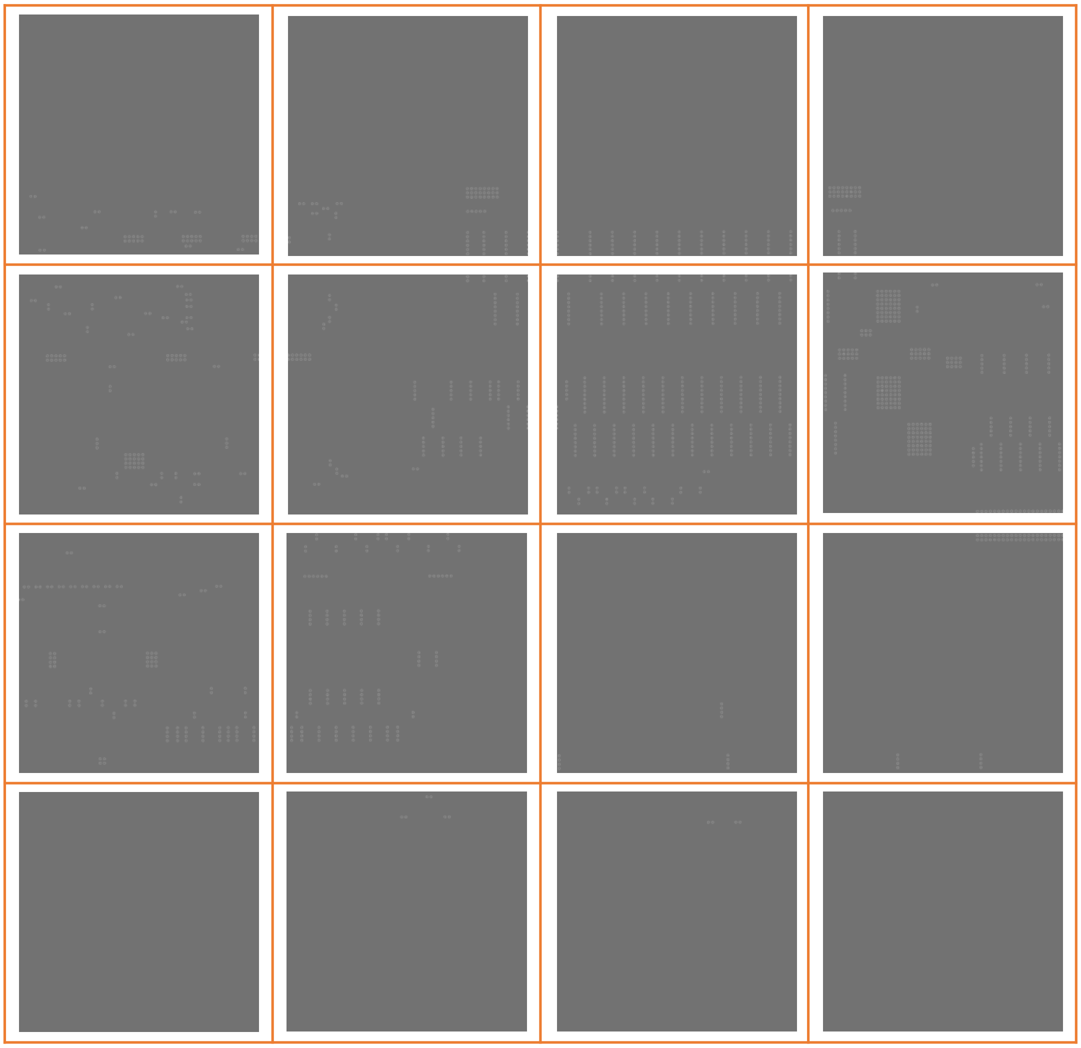

Figure 6.12: Contours of segmented vias in $M1$ layer. The contrast is changed to gray for the sake of better visualization.

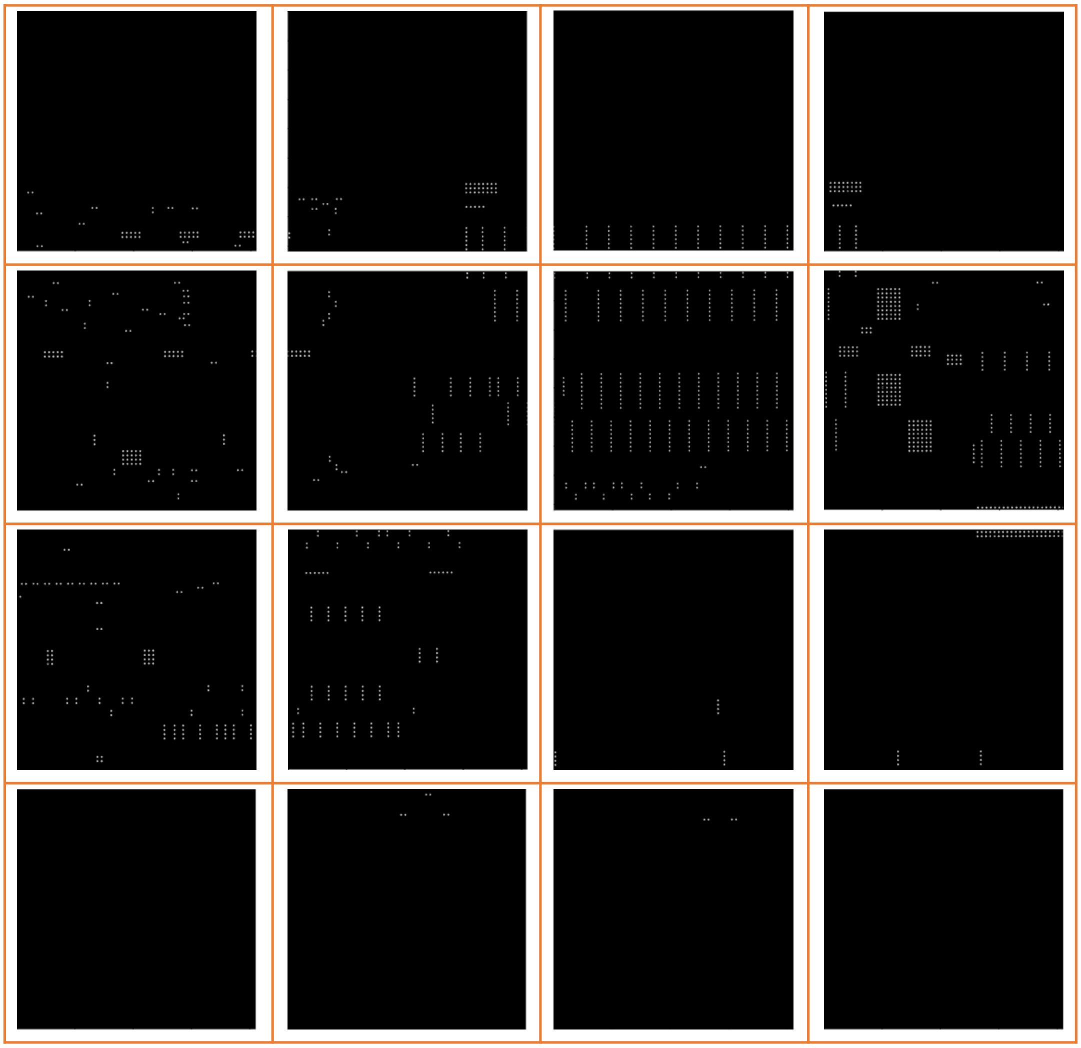

Figure 6.13: Centroids of the contours which are found for $M1$ blobs.

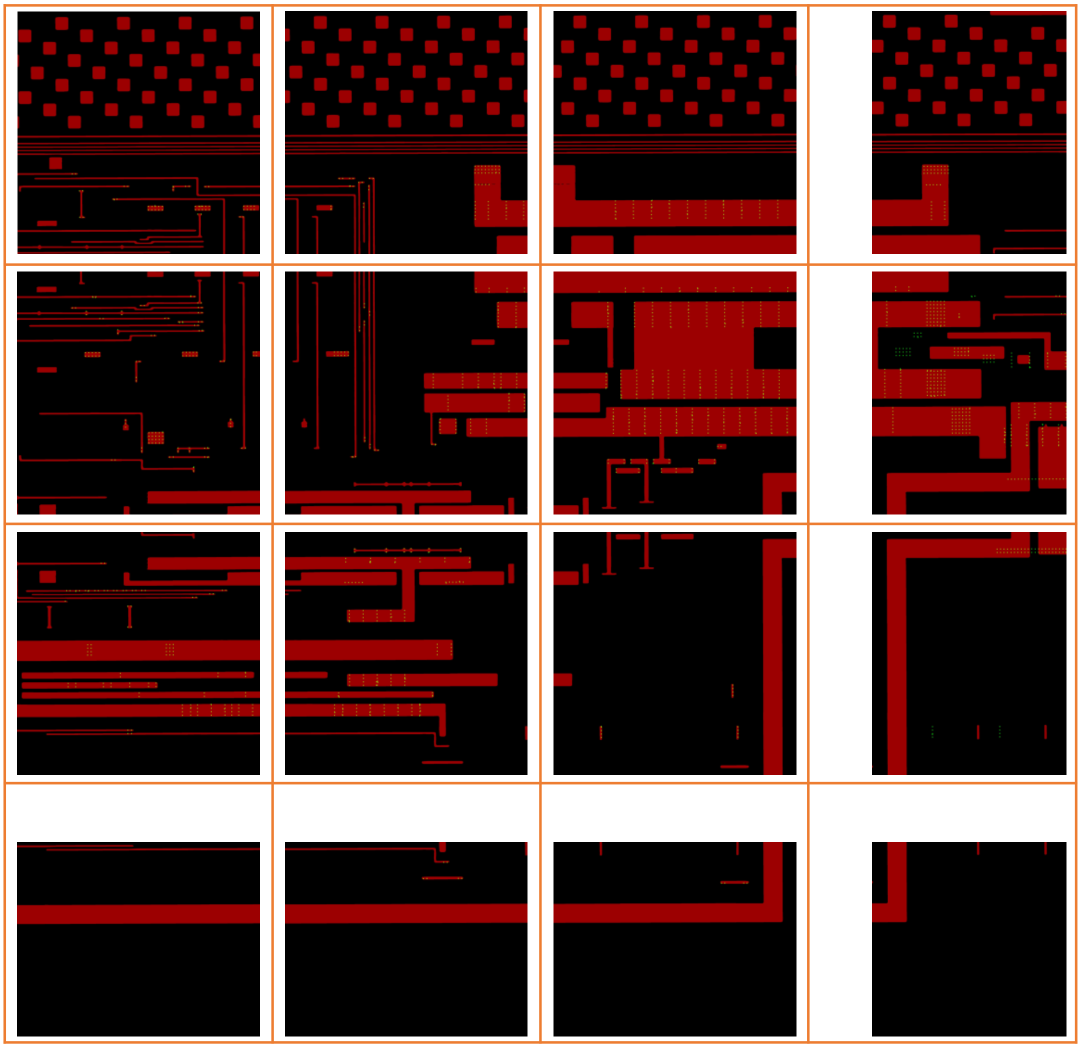

Figure 6.14: Resolved shift for $M1$ and $M2$ layers in RGB channels.

In this specific layer, the last row of blocks does not contain any via or a very small amount of them, so they produce outliers as expected; see Figure 6.15. Such outliers can get eliminated using the ℓ^1 norm. Using ℓ^1 and ℓ^2 norms, a polynomial is fitted to the shift values. For the definition of the norms, please see Appendix E. Based on the results, the ℓ^1 norm acts better in the presence of outliers; see Figure 6.16.

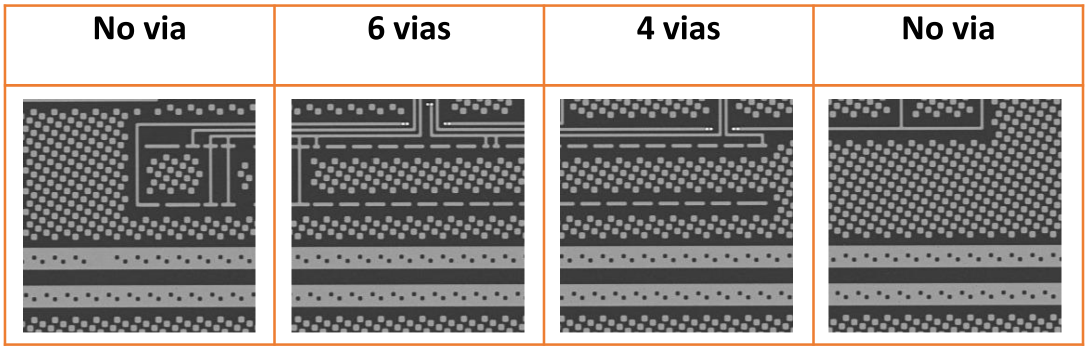

Figure 6.15: Examples of subimages from $M1$ layer which produce outliers.

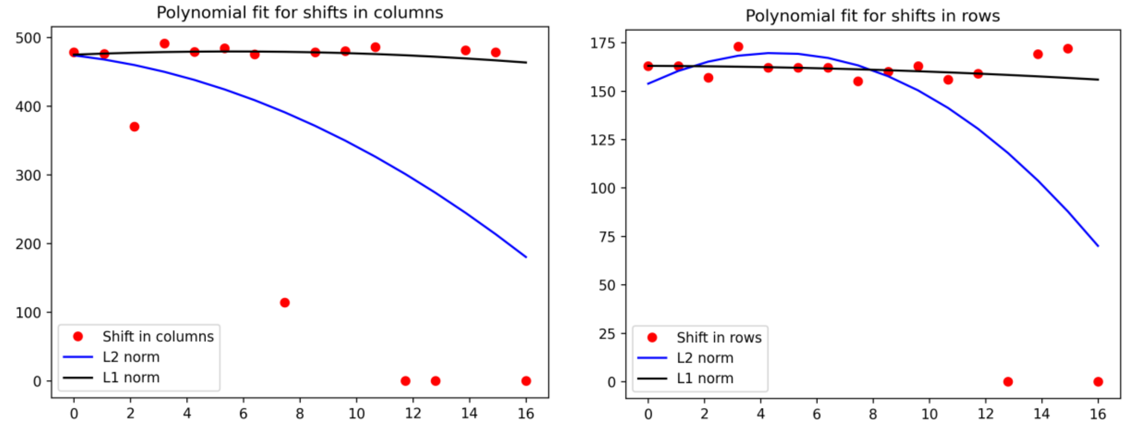

Figure 6.16: Fitted parabolas using ℓ^1 and ℓ^2 norms. Even in the presence of outliers, ℓ^1 fitted without getting affected by outliers. In both, $x-$axis is the number of the boxes in the 4×4 grid. $y-$axis shows the shift in columns in the left plot and the shift in the rows in the right plot.

In simple cases, the alignment should be done without making the search focus or expanding the search area in the metal layer. Using this simple alignment in the presented data set, the outliers are evident for the grid of 4×4; see Figure 6.8. Visualizing the maximum correlation using heat maps can show how local maximums appear in finding the correlation. Figure 6.17 are the heat maps of sliding each subimage from $M1$ on its corresponding subimage from $M2$ to find the maxima (expanding the search area in $M2$ layer is applied). As can be seen, in some cases, several local maxima were obtained, or the match happened in different places due to having wide wires or a very small number of vias to be matched with the metal layer.

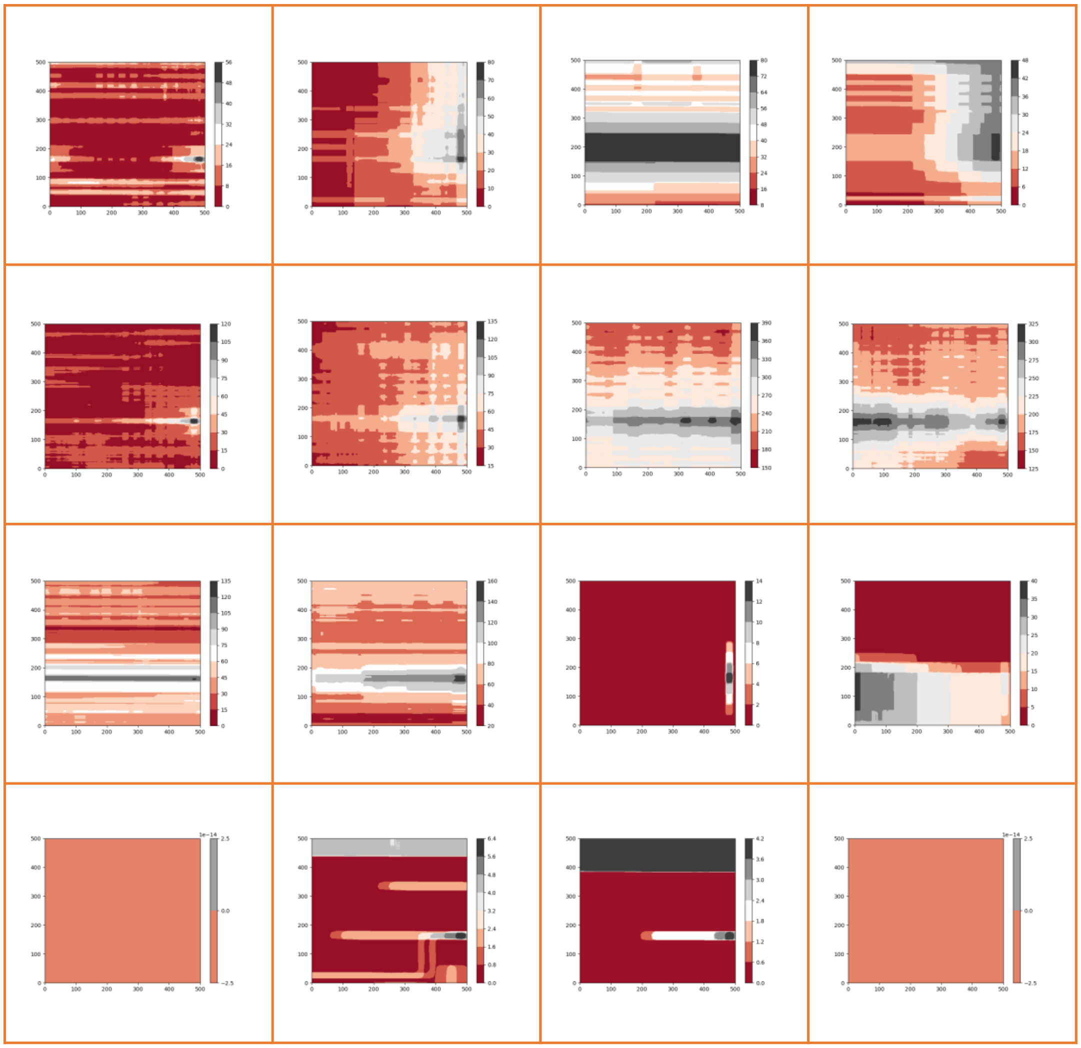

Figure 6.17: Heat maps of sliding vias extracted from the $M1$ subimages on the corresponding $M2$ metallization subimages. The search area for each subimage on $M2$ is expanded by $\frac{1}{4}$ of the size of subimages in positive direction in both $x-$ and $y-$ axes.

Expanding the search area helps us to preserve the features which are falling off of the edge. This also leads to a nearly accurate estimation of the shifts. By making our search focus on the median position of maxima for all subimages, one can look around the median on each heat map to find the maxima; in this way, we eliminate many local maxima that lead to incorrect alignment of the layers.

To recap, our novel alignment algorithm for reconstructing the $3D$ structure of semiconductor metallization layers from $2D$ SEM images reduces both the computational complexity and the time of the computation while achieving accuracy. Our method is feature-wise instead of pixel- or intensity-wise, so the correlation computation is reduced to the detected via centroids instead of whole image pixels. In this way, less data is involved in our calculation while we achieve accuracy by increasing our search area and eliminating local maxima using statistical central measurements.

Chapter 7

Conclusion

This Ph.D. focused on two related projects involving stitching in $2D$ and $3D$ using SEM semiconductor data sets we received from an industrial partner. Due to the volume of data, automatic stitching is very important, but given the ultimate goal of reconstructing circuits, even small errors must be avoided in stitching since they could lead to connections being broken. In addition to rigid-body motion (mostly translation), which we would expect due to the mechanical movement of the stage, we also observed non-linear distortions which cannot be explained by the mechanical movement, but must be associated to variations in the electronic components of the SEM. We were able to show that even this non-linear distortion can be resolved so that tiles can be accurately stitched together. To do this, we registered multiple subimages (typically under 10% of the width of the image), looking only for translations, and then fit those translations to an overall polynomial distortion. We were able to show that multiple approaches to fitting the distortion are feasible, and in this thesis, we compare several methods from the point of view of accuracy and computational efficiency.

Matching individual subimages may exhibit large errors. It is necessary to reduce the impact of these outliers. Fortunately, it is not necessary to use pre-processing since polynomial fitting to the shift values in the overlap area using the ℓ^1 norm (Manhattan distance) was observed to be insensitive to outliers. This was not the case for the Cartesian distance (ℓ^2).

Chapter 1 explains the motivation behind our research by introducing Moore's law and reverse engineering. Chapter 2 explains the SEM and its configuration and image formation. The $2D$ and $3D$ stitching are discussed in Chapters 3 to 6. In conclusion, our research questions are answered using our novel methods and results.

7.1 Research Questions

1. ***Is it possible to identify "numerical landmarks", that is, image regions which correlate to themselves much more strongly than they correlate with neighbouring regions, without having a predefined set of image features to match to? If so, how can they best be determined?***

   Yes, in Chapter 4.1, we define a modified Harris response function, based on NCC rather than SSD, and based on the self-correlation of a large neighbourhood of pixels, resulting in an ACS. Using the $2D$ histogram (x and y axes represent minimum and maximum eigenvalues) and visualizing the agreement between fitted paraboloids and the ACS, we identify the region with reliable candidates for numerical landmarks, which corresponds to positive values for the easy-to-calculate modified Harris response function. In addition to identifying the numerical landmarks, we test them by defining a greedy algorithm for finding a distributed set of landmarks spanning the overlap region, and using that set to register adjacent tiles.

2. ***What approaches reduce the computational complexity of stitching? Can dividing the overlap area into smaller subimages lead to effective parallelization?***

   We compared three common methods (FFT, SSD, NCC) in image registration, which is the main core of image stitching, both as conventional $2D$ methods and our novel method of reducing the dimension to $1D$ by projection (summing over rows and columns). We first compared the accuracy, with and without artefacts (e.g., brightness changes due to charging), and found NCC methods were more accurate overall, and the least susceptible to artefacts. The least accurate methods were based on the FFT, even though, in theory, FFT methods should be able to detect sub-pixel translations.

   The 1D-projectional methods reduce the computational complexity from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^2)$, for a $n \times n$ subimage. While SSD-1D is faster than NCC-1D, SSD methods were observed to be not accurate in the presence of artefacts, so we do not recommend them. We even found that the projection methods revealed undetected grain patterns which could be used for registration.

   Dividing the overlap between tiles into subimages and processing them independently could be done in parallel, and fortunately, we found that doing so resulted in accurate registration.

3. ***How can we align $2D$ stitched images from multiple layers without common features to reconstruct circuits?***

$3D$ datasets are big because it is not only a matter of working with big images with a high amount of data but a high number of layers that need to get stacked. We define a method that lowers computational complexity and improves accuracy. Preprocessing is needed to reduce the amount of data to speed up our computations. We know that the vias are going to get connected upward to the wires or vias of the above layer. So, the key features in $3D$ reconstruction are via and wires. The images are reduced to vias and then to the centroids of the vias as single points. In this way, we compare a set of points against a set of images rather than comparing the whole pixels in consecutive layers. In addition, we used neighbouring pixels to ensure that we are considering the borders of the image components (vias and wires) by taking the Fourier transform of the M2 layer and applying a Gaussian filter to it.

Parallelization is achieved by dividing the layer surface into smaller subimages and aligning corresponding subimages. As a final result, we have shift values for each of the subimages, and we can fit a polynomial to each layer obtained shift values. This polynomial is used for the general alignment. As we are reducing the inputs of the correlation function to a set of points rather than all pixels in each layer, we are able to expand our search area to neighbouring subimages and detect even nonlinear drifts that might fall off the edge of the subimages at a low cost. We also use statistical central measurement (median) to eliminate the outliers that occur due to various local maximums in correlation results. This helps us have our search for maximum correlation focused around the median of the subimages' shift values.

7.2 Future Work

Image stitching has wide applications. High-resolution images are essential in many scientific research areas, like cancer research, biology, material science, and semiconductors. Our study data sets were all semiconductor images. The algorithms in this thesis should be tested on biological tissues since the need to examine biological structures in three dimensions also leads to very large data sets. Biological samples are sensitive to increasing the voltage of SEM to gain better resolution or signal-to-noise and creating fiducials before imaging to use it as a landmark for stitching is also not recommended. Numerical landmarks and $1D$ projection in $2D$ stitching and the idea of reducing our computation to the set of points (centroids of specific features) could also be valuable in biological studies. Dealing with these problems mathematically, reduces the damage to the sample, while it is an aid to the hardware system of SEM.

Appendix A

Convex Optimization

A convex optimization problem is one of the form

$$\min f_0(x) \tag{A.0.1}$$

Subject to:

$$f_i(x) \leq b_i, i = 1, ..., m$$

where the functions $f_0, ..., f_m : R^n \rightarrow R$ are convex, i.e., satisfy $f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)$ for all $x, y \in R^n$ and all $\alpha, \beta \in R$ with $\alpha + \beta = 1$, $\alpha \geq 0$, $\beta \geq 0$. The least-squares problem, and linear programming problem, are both special cases of the general convex optimization problem [8].

Appendix B

Local Extremum

The extremum (or extreme value) of a function is the point at which its maximum or minimum value is obtained in some interval. The local extremum (or relative extremum) of a function is the point at which its maximum or minimum value is obtained in some interval which contains the point.

Appendix C

Coordinate Search

Coordinate search begins with an initial guess in the parameter space, at which point the score function is evaluated. To get a better score than the initial guess, a small environmental search is initiated based on this location. In this method, the search algorithm selects one point from each parameter axis to determine a better score value. In the optimizer configuration section, you can define the search distance from the current base location. If a better score value is found at a certain point during the local environmental search, this point is selected as the base location for the next environmental search. The search distance for an algorithm is reduced, by a user-defined factor, if better results cannot be found in the given environment around the current base location. Typically, this reduction is one-half.

Appendix D

Exhaustive Search

When a problem includes searching for an element with a specific property, it is known as a brute force solution, usually involving permutations, combinations, or subsets of a set. A solution space is a space in which all possible answers can be enumerated systematically. When conducting an exhaustive search for discrete problems where no efficient solution method is known, it may be necessary to test each possibility sequentially until the solution is determined. Exhaustive searching, direct searching, or brute force searching are all terms used to describe such an exhaustive examination of all possibilities. Finding the integral pixel shift between two images is an example of a discrete problem.

Appendix E

Norms

Norm is about measuring the distance. Different norms can be used based on the choice of algorithm to calculate the distance between two points.

E.1 L1 Norm

This norm, known as the Taxicab norm, is about summing up the distances one would take in different horizontal and vertical directions to reach the destination point. It is like taking a taxi or walking through streets to move from one place to another on the map, see Figure E.2. The L1 norm is formulated as follows:

$$||u - v||_1 = |u_x - v_x| + |u_y - v_y| \tag{E.1.1}$$

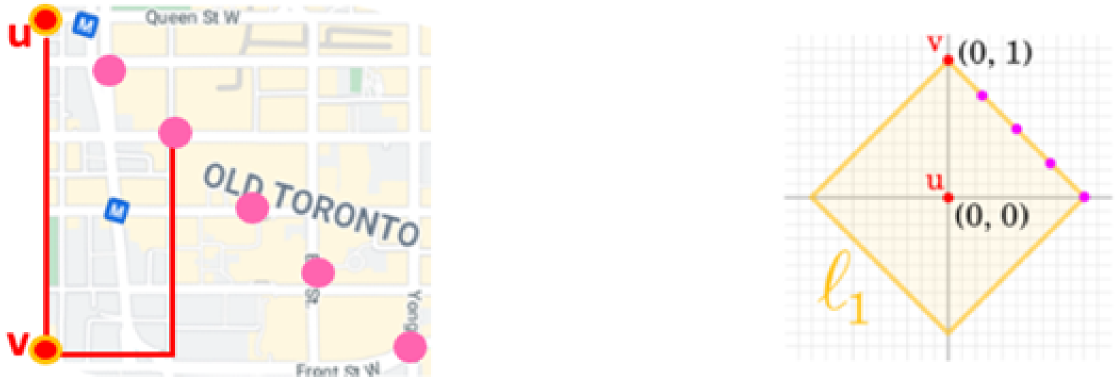

Figure E.1: L1 Norm (Taxicab Norm or Manhatan distance) visualization. To travel from v to any of the pink points standing for destination u, one needs to move through streets and pass blocks. So, the movements in $x-$ and $y-$ are adding up based on the path which is used to reach any of the pink points that can be our destination u. The left figure shows the map view example, and the right one is the ℓ^1 norm in the mathematical coordinate system.

E.2 L2 Norm

This Norm is about finding the shortest distance between two points. In terms of the geometrical aspect, it is like taking a plane to go from one point to another on the map, see Figure E.1. L2 norm is formulated as follows:

$$||u - v||_2 = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2} \tag{E.2.1}$$

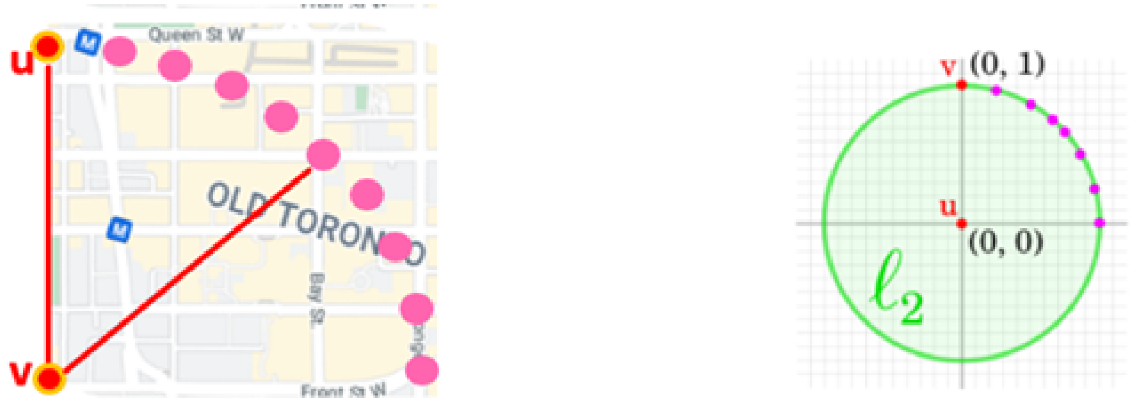

Figure E.2: L2 Norm visualization, which is the shortest distance between two points. The left figure shows the map view example, and the right one is the ℓ^1 norm in the mathematical coordinate system.

E.3 L∞ Norm

This Norm is about finding the maximum distance in both directions of x and y. In terms of the geometrical aspect, it is like taking the maximum straight distance in both directions of x and y to reach the destination, see Figure E.3. L∞ norm is formulated as follows:

$$||u - v||_\infty = \max\left((u_x - v_x), (u_y - v_y)\right) \tag{E.3.1}$$

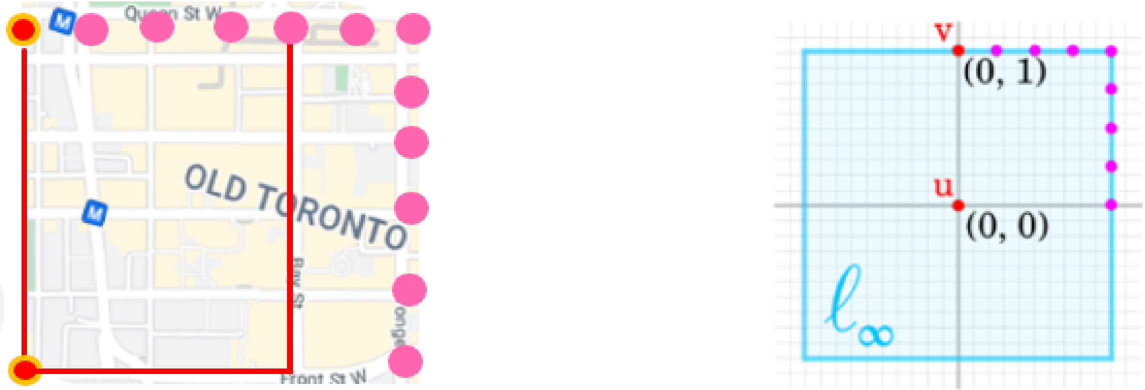

Figure E.3: L∞ Norm visualization, which is the maximum straight distance in both $x-$ and $y-$ directions to reach the destination. The left figure shows the map view example, and the right one is the ℓ^1 norm in the mathematical coordinate system.

Bibliography

[1] K. Akhtar, S. A. Khan, S. B. Khan, and A. M. Asiri. Scanning electron microscopy: Principle and applications in nanomaterials characterization. In *Handbook of Materials Characterization*, pages 113–145. Springer, 2018.

[2] C. Anand, Y. Sartipi, N. Khoonkari, and C. Schankula. *CCEM 2020 - Image Processing and Optimization.* Apr 2020. URL `https://www.researchgate.net/publication/341078491_CCEM_2020_-_Image_Processing_and_Optimization`.

[3] M. Ball and D. McCartney. The measurement of atomic number and composition in an sem using backscattered detectors. *Journal of Microscopy*, 124(1):57–68, 1981.

[4] R. A. Beezer. A first course in linear algebra. 2021.

[5] B. J. Bella. Imaging distortion correction in panorama stitching in sem. Master's thesis, Masaryk University, 2017.

[6] H. Bethe. *Handbook of Physics.* Springer, Berlin Heidelberg New York, 1933.

[7] U. Bhosle, S. Chaudhuri, and S. Dutta Roy. A fast method for image mosaicing using geometric hashing. *IETE Journal of Research*, 48(3-4):317–324, 2002.

[8] S. Boyd, S. P. Boyd, and L. Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[9] A. Bria and G. Iannello. Terastitcher-a tool for fast automatic 3d-stitching of teravoxel-sized microscopy images. *BMC bioinformatics*, 13(1):316, 2012.

[10] L. G. Brown. A survey of image registration techniques. *ACM computing surveys (CSUR)*, 24(4):325–376, 1992.

[11] M. Brown, D. G. Lowe, et al. Recognising Panoramas. In *ICCV*, volume 3, page 1218, 2003. doi: https://doi.org/10.1109/ICCV.2003.1238630.

[12] J. Buckman. Use of automated image acquisition and stitching in scanning electron microscopy: Imaging of large scale areas of materials at high resolution. *Microsc. Anal*, 28(1):13–15, 2014.

[13] T. Burnett, R. Kelley, B. Winiarski, L. Contreras, M. Daly, A. Gholinia, M. Burke, and P. Withers. Large volume serial section tomography by xe plasma fib dual beam microscopy. *Ultramicroscopy*, 161:119–129, 2016.

[14] X. Cao, J. Yang, L. Wang, Z. Xue, Q. Wang, and D. Shen. Deep learning based inter-modality image registration supervised by intra-modality similarity. In *Machine Learning in Medical Imaging: 9th International Workshop, MLMI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings 9*, pages 55–63. Springer, 2018.

[15] J. Chalfoun, M. Majurski, T. Blattner, K. Bhadriraju, W. Keyrouz, P. Bajcsy, and M. Brady. MIST: Accurate and Scalable Microscopy Image Stitching Tool

with Stage Modeling and Error Minimization. *Scientific reports*, 7(1):1–10, 2017. doi: https://doi.org/10.1038/s41598-017-04567-y.

[16] D. Clarke and P. Stuart. An anomalous contrast effect in the scanning electron microscope. *Journal of Physics E: Scientific Instruments*, 3(9):705, 1970.

[17] L. Erasmus, D. Hurter, M. Naudé, H. Kritzinger, and S. Acho. A short overview of mri artefacts. *SA Journal of Radiology*, 8(2), 2004.

[18] A. Flynn, A. Green, G. Boxer, R. Pedley, and R. Begent. A comparison of image registration techniques for the correlation of radiolabelled antibody distribution with tumour morphology. *Physics in Medicine & Biology*, 44(7):N151, 1999.

[19] J. I. Goldstein, D. E. Newbury, J. R. Michael, N. W. Ritchie, J. H. J. Scott, and D. C. Joy. *Scanning electron microscopy and X-ray microanalysis*. Springer, 2017.

[20] S. Gupta and V. Kapoor. *Fundamentals of mathematical statistics*. Sultan Chand & Sons, 2020.

[21] C. Harris, M. Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[22] M. Hisham, S. N. Yaakob, R. Raof, A. A. Nazren, and N. Wafi. Template matching using sum of squared difference and normalized cross correlation. In *2015 IEEE student conference on research and development (SCOReD)*, pages 100–104. IEEE, 2015.

[23] T. Kajihara, T. Funatomi, H. Makishima, T. Aoto, H. Kubo, S. Yamada, and

Y. Mukaigawa. Non-rigid registration of serial section images by blending transforms for 3d reconstruction. *Pattern Recognition*, 96:106956, 2019.

[24] K. Kanaya and S. Okayama. Penetration and energy-loss theory of electrons in solid targets. *Journal of Physics D: Applied Physics*, 5(1):43, 1972.

[25] R. A. Ketcham and W. D. Carlson. Acquisition, optimization and interpretation of x-ray computed tomographic imagery: applications to the geosciences. *Computers & Geosciences*, 27(4):381–400, 2001.

[26] K. G. Larkin, M. A. Oldfield, and H. Klemm. Fast fourier method for the accurate rotation of sampled images. *Optics communications*, 139(1-3):99–106, 1997. doi: https://doi.org/10.1016/s0030-4018(97)00097-7.

[27] C.-C. Lin, F. Stellari, L. Gignac, P. Song, and J. Bruley. Automatic registering and stitching of TEM/STEM image mosaics. In *ISTFA 2013*, pages 152–158. ASM International, 2013.

[28] B. Ma, T. Zimmermann, M. Rohde, S. Winkelbach, F. He, W. Lindenmaier, and K. E. Dittmar. Use of autostitch for automatic stitching of microscope images. *Micron*, 38(5):492–499, 2007.

[29] B. Madhukar and R. Narendra. Lanczos resampling for the digital processing of remotely sensed images. In *Proceedings of International Conference on VLSI, Communication, Advanced Devices, Signals & Systems and Networking (VCASAN-2013)*, pages 403–411. Springer, 2013. doi: https://doi.org/10.1007/978-81-322-1524-0_48.

[30] N. Marturi, S. Dembélé, and N. Piat. Fast image drift compensation in scanning electron microscope using image registration. In *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 807–812. IEEE, 2013.

[31] D. P. Mitchell and A. N. Netravali. Reconstruction filters in computer-graphics. *ACM Siggraph Computer Graphics*, 22(4):221–228, 1988.

[32] J. Modersitzki. *Numerical Methods for Image Registration*. OUP Oxford, 2003. doi: https://doi.org/10.1093/acprof:oso/9780198528418.001.0001.

[33] P. Nowakowski, M. Ray, and P. Fischione. Advanced tools and techniques for delayering and cross-sectioning semiconductor devices. In *ISTFA 2017*, pages 592–596. ASM International, 2017.

[34] C. Pawlowicz, B. Trindade, and M. Green. The role of cloud computing in a modern reverse engineering workflow at the 5nm node and beyond. In *ISTFA 2021*, pages 163–171. ASM International, 2021. doi: https://doi.org/10.31399/asm.cp.istfa2021p0163.

[35] J. T. S. Phang, K. H. Lim, and R. C. W. Chiong. A review of three dimensional reconstruction techniques. *Multimedia Tools and Applications*, 80(12):17879–17891, 2021.

[36] J. Piňos, Š. Mikmeková, and L. Frank. About the information depth of backscattered electron imaging. *Journal of microscopy*, 266(3):335–342, 2017.

[37] S. Rahangdale, L. Voortman, and P. Kruit. Real-time multi-beam sem image

stitching (conference presentation). In *Real-Time Image and Video Processing 2018*, volume 10670, page 106700E. SPIE, 2018.

[38] V. Rankov, R. J. Locke, R. J. Edens, P. R. Barber, and B. Vojnovic. An algorithm for image stitching and blending. In *Three-Dimensional and Multidimensional Microscopy: Image Acquisition and Processing XII*, volume 5701, pages 190–199. International Society for Optics and Photonics, 2005.

[39] Y. R. Rao, N. Prathapani, and E. Nagabhooshanam. Application of normalized cross correlation to image registration. *International Journal of Research in Engineering and Technology*, 3(5):12–16, 2014.

[40] M. Riordan, L. Hoddeson, and C. Herring. The invention of the transistor. *Reviews of Modern Physics*, 71(2):S336, 1999.

[41] J. L. Rodgers and W. A. Nicewander. Thirteen ways to look at the correlation coefficient. *American statistician*, pages 59–66, 1988. doi: https://doi.org/10.2307/2685263.

[42] Y. Sartipi. 3d surface reconstruction and 2d segmentation of scanning electron microscopy images. 2019.

[43] Y. Sartipi. 3d Surface Reconstruction and 2D Segmentation of Scanning Electron Microscopy Images. Master's thesis, McMaster University, 2022.

[44] J. N. Sarvaiya, S. Patnaik, and S. Bombaywala. Image registration by template matching using normalized cross-correlation. In *2009 international conference on advances in computing, control, and telecommunication technologies*, pages 819–822. IEEE, 2009.

[45] M. Schmidt. Least squares optimization with l1-norm regularization. *CS542B Project Report*, 504:195–221, 2005.

[46] R. Simon and B. Williams. Secondary-electron emission. *IEEE Transactions on Nuclear Science*, 15(3):167–170, 1968.

[47] E. J. Sternglass. Backscattering of kilovolt electrons from solids. *Physical Review*, 95(2):345, 1954.

[48] M. A. Sutton, J. J. Orteu, and H. Schreier. *Image correlation for shape, motion and deformation measurements: basic concepts, theory and applications*. Springer Science & Business Media, 2009.

[49] T. Svoboda, J. Kybic, and V. Hlavac. *Image processing, analysis & and machine vision-a MATLAB companion*. Thomson Learning, 2007.

[50] L. Swanson. Liquid metal ion sources: Mechanism and applications. *Nuclear Instruments and Methods in Physics Research*, 218(1-3):347–353, 1983.

[51] G. Vino and A. D. Sappa. Revisiting harris corner detector algorithm: a gradual thresholding approach. In *International Conference Image Analysis and Recognition*, pages 354–363. Springer, 2013.

[52] C.-W. Wang, E. Budiman Gosno, and Y.-S. Li. Fully automatic and robust 3d registration of serial-section microscopic images. *Scientific reports*, 5(1):15051, 2015.

[53] E. W. Weisstein. Taylor series. *https://mathworld. wolfram. com/*, 2004.

[54] C. S. Xu, K. J. Hayworth, Z. Lu, P. Grob, A. M. Hassan, J. G. García-Cerdán, K. K. Niyogi, E. Nogales, R. J. Weinberg, and H. F. Hess. Enhanced fib-sem systems for large-volume 3d imaging. *elife*, 6:e25916, 2017.

[55] J. Yi, X. He, Y. Sun, and Y. Li. Electron beam-physical vapor deposition of sic/sio2 high emissivity thin film. *Applied surface science*, 253(9):4361–4366, 2007.

[56] W. Zhou, R. Apkarian, Z. L. Wang, and D. Joy. Fundamentals of scanning electron microscopy (sem). In *Scanning microscopy for nanotechnology*, pages 1–40. Springer, 2006.