



Master Photonics for Security Reliability and Safety (PSRS)



POLARIMETRIC IMAGING APPLIED FOR VOID DETECTION IN SILICON WAFERS Internship Report

Presented by
Anahit Hambardzumyan

and defended at
University Jean Monnet

Date: 25 August 2025

Academic Supervisor: Prof. Carlo Riccardi

Host Supervisor: Dr. Davit Hakobyan

Jury Committee: Prof. Nathalie Destouches, Dr. Rajeev Dwivedi

**POLARIMETRIC IMAGING APPLIED FOR VOID
DETECTION IN SILICON WAFERS**

This Master's thesis report contains proprietary and confidential information. The content of this document is intended solely for review and evaluation by the thesis supervisors and the designated jury members mentioned on the cover page. Any disclosure, distribution, or reproduction of this thesis or its contents is prohibited. Unauthorized access to or sharing of the information contained herein is not permitted and may result in legal action.

Contents

1	Wafer Inspection – Context and Outline	6
1.1	Challenges in Semiconductor Wafer Bonding and the Importance of Void Detection	6
1.2	Existing Methods for Void Detection and Limitations	7
1.3	Motivation and Outline	8
2	Polarimetric Imaging Method and Implementation	10
2.1	Polarized Light and Its Interaction with Mechanical Stress	10
2.1.1	Polarization Theory	10
2.1.2	Stokes Vectors for Polarization Analysis	12
2.1.3	Photoelasticity and Stress-Optic Law	14
2.2	Polarimetric Microscope	15
2.2.1	Optical system automatization and Process Workflow	18
2.2.2	Polarimetric Imaging System Characterization	21
2.2.2.1	Modulation Transfer Function	22
2.2.2.2	Characterization of Polarization Analysis	23
3	Stress Modeling and Experimental Evaluation of Voids	25
3.1	Simulation of Void-Induced Mechanical Response	25
3.2	Experimental Validation and Retardance Analysis	26
3.2.1	Benchmarking of the Polarimetric Imaging System	26
3.2.2	Detection of Voids Via Retardance Maps	27
4	Conclusion	33
	Appendices	39

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr. Davit Hakobyan, for giving me the opportunity to carry out my master's internship at Merck Electronics and for his continuous support throughout its duration. With his guidance, I have grown not only in technical expertise but also in communication, presentation, and critical company-oriented thinking—lessons that will stay with me for a lifetime. I am also thankful to all my colleagues at Merck who supported me in different ways along the journey.

I extend my sincere thanks to my academic professors for their guidance and responsiveness during my master's studies, which shaped this important stage of my education.

Finally, I am profoundly grateful to my family and friends for their unwavering encouragement and support.

Abstract

Interfacial voids formed during wafer bonding degrade device performance and yield, yet existing inspection techniques are constrained by resolution, speed, or material transparency [1]. This thesis investigates polarimetric imaging as a non-contact approach to subsurface void detection by exploiting stress-induced birefringence around defects.

We developed an automated polarimetric microscope and, through Stokes formalism, characterized the polarization footprint of voids. Finite-element simulations of representative voids predicted anisotropic stress fields and distinctive retardance signatures that guided interpretation.

On bonded silicon wafers, the measured retardance maps recover every void identified by acoustic C-scan and phase-shift deflectometry and additionally reveal smaller voids.

We discuss practical limits—throughput of scanning and reconstruction, polarization purity, and mechanical stability—and outline mitigations. Overall, the results establish polarimetric imaging as a promising void-detection technique for advanced semiconductor manufacturing, with clear paths to faster acquisition, improved polarization optics, and calibrated stress mapping across materials and bonding schemes.

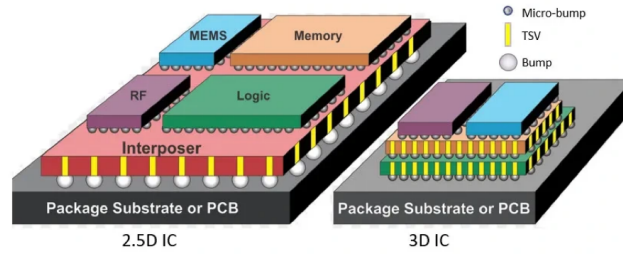


Figure 1. 2.5D vs. 3D packaging technologies [3].

1 Wafer Inspection – Context and Outline

The semiconductor industry has become the backbone of modern information technology, enabling advances in computing, telecommunications, sensing, and consumer electronics. With device dimensions continuing to scale down and packaging technologies evolving toward smaller scale integration, ensuring the structural and functional integrity of semiconductor wafers has become a critical challenge [2]. In particular, wafer-to-wafer and die-to-wafer bonding processes are widely used in advanced device architectures. However, these processes are highly sensitive to contamination, particle defects, trapped gases, and thermal mismatch, which can result in the formation of *voids* at the bonding interface. Voids can compromise electrical performance, reduce device yield, and induce catastrophic failure in later processing steps.

Over the past decades, several inspection techniques have been developed for void detection, including scanning acoustic microscopy, infrared interferometry, and hybrid acousto-optic methods. While these tools have proven effective in specific scenarios, they are limited in spatial resolution, inspection speed, or applicability to opaque and multilayered materials.

1.1 Challenges in Semiconductor Wafer Bonding and the Importance of Void Detection

The growing demand for high-speed devices increases the demand and fabrication for high-performance chips. This tendency gave rise to 2.5D and 3D packaging technologies. The development of these technologies aims size and weight reduction, performance improvement and reduction in power consumption. The schematics in Figure 1 shows the difference of 2.5D and 3D packaging. We can think about the 2.5D packaging as a city of buildings of the same size connected by bridges, while 3D packaging can be compared with a multistory

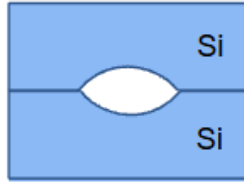


Figure 2. Cross-sectional view of voids in a Si wafer. The void forms a bubble at the interface, leading to deformation in the neighboring layers.

building as it allows vertical stacking of different chips using through-silicon vias (TSV) or other interconnects. 2.5D and 3D packaging have dramatically reduced space usage due to their optimal physical design [3].

The advancements and decrease in size brings its own challenges. The fabrication processes require bonding and thinning, during those processes different types of defects occur in the wafer, such as voids, which are bubbles formed within the wafer due to trapped residual gasses, trapped particles, wafer warpage, etc. [4], [5]. The illustration of voids is presented in the Figure 2.

Even micron-scale voids create local stress, degrading the bond energy and opening leakage paths in both wafer-to-wafer and die-to-wafer bondings [6]. These effects lead to loss of electric yield or delamination during subsequent processing steps [7].

1.2 Existing Methods for Void Detection and Limitations

One of the traditional methods for void inspection is Scanning Acoustic Microscopy (SAM) technique that is based on generating and focusing ultrasonic pulses onto the sample and recording the echo amplitude of the response. The authors report the smallest detectable void sizes of 6–8 μm at a depth of 50 μm within the wafer, at 110–175 MHz. The main drawbacks in SAM technology is the need of immersion in a liquid (typically water), which can be problematic for certain wafers, in addition SAM requires long raster scanning time and is challenging to isolate a specific interface since the response from different layers overlap creating ambiguous interpretation [8].

Another implemented technique is the Infrared interferometry system by Lumetrics. Its a non-contact, fluid free technology, that works by focusing IR low-coherence beam on an interface of wafer, in case the interface contains voids, the regular interference pattern changes yielding three peaks instead of two. By calculating the optical path among these

peaks, one can localize the voids and distinguish them from the surface. The resolution of the IR interferometric system is limited by the IR spot size $50\text{ }\mu\text{m}$. The main challenge of this technology lies in analyzing the complex interferometric signals from multiple surfaces.[9].

Another void detection technology is offered by Onto Innovation, that combines acoustic and optical solutions together. The device is called EchoScan System, which generates picosecond pulsed laser beams and directs them onto the wafer (usually metal or semiconductor layer). After the absorption the interface expands suddenly releasing acoustic wave that travels through the film and reflects at interfaces. Voids reflect differently altering timing or amplitude of returned signal, allowing their detection. This technology has been reported to achieve lateral resolution down to $1\text{ }\mu\text{m}$ and depth resolution of $50\text{--}100\text{ nm}$. However, it is expensive, requires liquid immersion, and is limited by transparency constraints, performing best on opaque or semi-opaque layers such as metals and certain dielectrics [10].

1.3 Motivation and Outline

The challenges described above demonstrate that, despite the availability of various inspection techniques, existing methods offer only partial solutions. Significant advancements are still required, particularly in improving void detectability, inspection speed, and penetration capability. This limitation serves as the primary motivation for our work. This thesis explores polarimetric imaging as a novel method for void detection in bonded semiconductor wafers. This approach leverages the polarization state of light to characterize optical anisotropies introduced by bonding imperfections.

The primary purpose of this work is to establish a proof of concept, demonstrating that polarimetric imaging can reliably detect interfacial voids. Beyond this, the study aims to evaluate the stress distribution around voids and link it to their birefringence signatures, to experimentally validate the imaging system on bonded silicon wafers and benchmark the results against conventional inspection methods such as acoustic C-scans and phase shift deflectometry (PSD), and to assess the advantages, limitations, and prospects of polarimetric imaging for semiconductor void inspection.

A key advantage of this technique is its ability to probe subsurface features without physical contact. More importantly, the detection is based on the optical response to mechanically induced birefringence, which tends to propagate over a broader area than the void itself. This mechanical signature enhances the spatial detectability of defects, thereby enabling the identification of even smaller voids that may otherwise remain undetected by conven-

tional methods.

The thesis is structured as follows:

In Chapter 2 we introduce the fundamental theoretical concepts that support this work. The chapter focuses on the Stokes vector formalism for polarized light, the photoelastic behavior of materials, and the stress-optic law, which serves as a bridge between mechanical stress and the optical response. This theoretical foundation is followed by a detailed description of the polarimetric optical system used in our experiments, with particular attention to the illumination module and the characterization metrics of the imaging system.

In Chapter 3 we present both simulated and experimental results and analyze the mechanical signature generated by voids. In the experimental section, we focus on results obtained from a silicon bonded wafer, outlining the full workflow of the system, and discussing the observed outcomes in detail.

In Chapter 4 we conclude the thesis by summarizing the main contributions and findings. In the second half of it, we discuss recommendations and potential directions for future research.

2 Polarimetric Imaging Method and Implementation

In this chapter, we begin with a brief introduction to the fundamental concepts of polarization. We then describe two methods of polarization analysis commonly used in experiment, explaining their working principles.

In the second part of the chapter, we present the polarimetric microscope bench designed and built for photoelasticity analysis. We then outline the process workflow. Finally, we introduce evaluation metrics offering a preliminary assessment of the system's performance.

2.1 Polarized Light and Its Interaction with Mechanical Stress

Polarization is the extra degree of freedom of a transverse electromagnetic wave and a highly sensitive probe of optical anisotropy. When transparent solids are mechanically stressed, their dielectric response becomes direction dependent-photoelastic: the refractive index splits along orthogonal axes and a wavelength-dependent phase delay - retardance, appears between them. This coupling lets us turn changes in polarization into qualitative and quantitative information about the underlying stress field.

To use polarization for stress metrology we need two ingredients: (i) a clear language to describe polarization states, and (ii) a practical way to measure them. Mathematically, fully polarized fields are handled by Jones vectors/matrices, while partially polarized or spatially varying fields—typical in imaging—are described by the Stokes–Mueller formalism that we adopt here. Experimentally, the Stokes parameters are reconstructed from a small set of intensity measurements using analyzer rotations or a rotating quarter-wave plate.

Within this framework, mechanical stress acts like a weak, linear retarder whose retardance is proportional to the difference of principal stresses and accumulates with optical thickness. In bonded semiconductor stacks, such as in die-to-wafer and wafer-to-wafer bondings, residual fields and local stress concentrators such as micron-scale voids produce measurable birefringence contrast, enabling non-contact inspection of bond quality and stress distribution.

2.1.1 Polarization Theory

Polarization is a unique property of transverse electromagnetic wave, such as light. The polarization of light describes the oscillations of electric field in the direction perpendicular to the propagation direction of light.

Different polarization states describe different relationships of oscillations and the propaga-

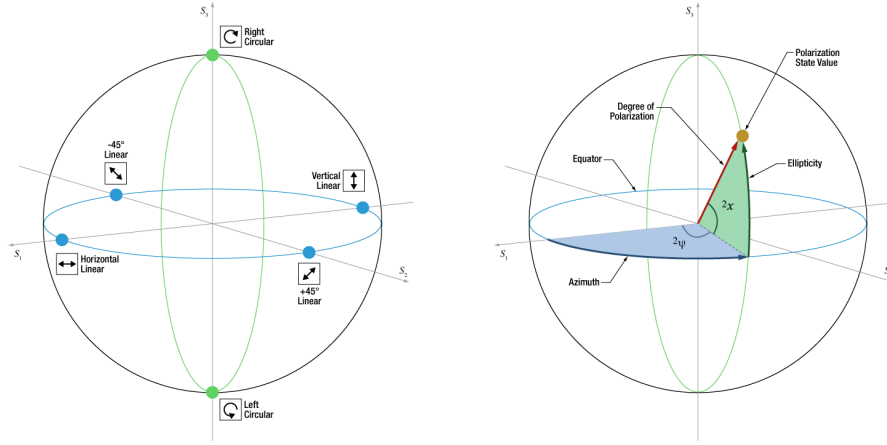


Figure 3. Visualization of polarization states on the Poincaré sphere. The poles correspond to circular polarizations and the equator to linear polarizations. The Poincaré sphere is given by ellipticity angle and azimuth with radius equal to degree of polarization. Adapted from Thorlabs “Using the Poincaré Sphere to Represent the Polarization State” [3].

tion direction. Linearly polarized light fields oscillate along a single direction perpendicular to the direction of propagation, while in circular or elliptic polarization, the fields rotate in a plane as the wave travels.

We can think about linear and circular polarizations as two particular cases of elliptic polarization. In linear polarization, the phase difference between the x and y components of the electric field is 0 or π . The resulting ellipse degenerates into a straight line, while for a circular polarized light the amplitudes of the x and y components are equal and the difference is $\pm \frac{\pi}{2}$ radians.

Right-hand and left-hand circularly polarized light refer to the direction of rotation of the electric field vector as seen by an observer looking into the direction of wave propagation. The electric field in right-hand circular polarization (RCP) rotates clockwise in the plane perpendicular to the propagation, while that in left-hand circular polarization (LCP) rotates counter-clockwise [11].

An intuitive way to visualize all possible polarization states is the Poincaré sphere. Each point on the Poincaré sphere corresponds to a unique polarization state: the poles represent right- and left-circular polarizations, points along the equator correspond to linear polarizations at different orientations, and intermediate points represent elliptical polarizations. This geometric construction provides a unified framework for visualizing the continuous transitions between different polarization states.

The position of a point on the sphere is specified by two angular parameters — the azimuth and the ellipticity angle — together with the radius. The radius is determined by the degree of polarization (DOP), which takes values in the range $(0, 1)$. A fully polarized state lies on the surface of the sphere ($\text{DOP} = 1$), while partially polarized states lie inside the sphere. The concept of the degree of polarization will be discussed in more detail in later chapters.

2.1.2 Stokes Vectors for Polarization Analysis

In optics, their particular interest and application have found the Jones and Stokes formalisms. The Jones formalism is given by two vectors $\vec{E}_x(t)$ and $\vec{E}_y(t)$ and naturally derives from the mathematical phase and amplitude representation of the light, hence it is valid for fully polarized light. In real-world problems we do not always have ideally polarized light, hence Jones matrices can not completely describe the light polarization. Unpolarized or partially polarized light can be described by another mathematical framework called Stokes vectors or Stokes formalism. Stokes vectors can be derived from the wave equations:

$$\begin{aligned} E_x(t) &= E_{0x}(t) \cos[\omega t + \delta_x(t)] \\ E_y(t) &= E_{0y}(t) \cos[\omega t + \delta_y(t)] \end{aligned} \quad (1)$$

Where $E_{0x}(t)$ and $E_{0y}(t)$ are the instantaneous amplitudes, ω is the instantaneous angular frequency and $\delta_x(t)$ and $\delta_y(t)$ are instantaneous phase factors. After some mathematical modifications we end up with the following equation:

$$(E_{0x}^2 + E_{0y}^2)^2 - (E_{0x}^2 - E_{0y}^2)^2 - (2E_{0x}E_{0y} \cos \delta)^2 = (2E_{0x}E_{0y} \sin \delta)^2 \quad (2)$$

We note the magnitudes in parentheses in the following way:

$$\begin{aligned} S_0 &= E_{0x}^2 + E_{0y}^2 \\ S_1 &= E_{0x}^2 - E_{0y}^2 \\ S_2 &= 2E_{0x}E_{0y} \cos \delta \\ S_3 &= 2E_{0x}E_{0y} \sin \delta \end{aligned} \quad (3)$$

Substituting the parenthesis in (2) with notations in (3) we obtain the following:

$$S_0^2 = S_1^2 + S_2^2 + S_3^2. \quad (4)$$

The vector

$$\mathbf{s} = \begin{pmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{pmatrix}$$

is called Stokes vector [12]. The parameter S_0 denotes the total intensity of the light. S_1 represents the dominance of horizontal over vertical polarization, S_2 -the dominance of $+45^\circ$ over -45° polarization, and S_3 quantifies the degree of circular polarization, distinguishing between right- and left-handedness.

In practice, we are interested in the intensity representation of the Stokes parameters because intensity is what can be directly measured in the laboratory using photodetectors or cameras. Obtaining the complete Stokes vector typically requires several independent intensity measurements at different analyzer orientations. Two widely used approaches to acquire these measurements are the classical and the rotating quarter-wave plate methods.

1. The classical method, uses a sequence of a rotating quarter-wave plate and analyzer to obtain the required six intensities [12]. The Stokes vectors are computed from that intensity measurements as shown in Eq. 5

$$\begin{aligned} S_0 &= I_0 + I_{90} \\ S_1 &= I_0 - I_{90} \\ S_2 &= I_{45} - I_{135} \\ S_3 &= I_{\text{RCP}} - I_{\text{LCP}}, \end{aligned} \tag{5}$$

where I_{135} is the same as I_{-45} . We first record I_0 , I_{90} , I_{45} , and I_{135} by rotating only the analyzer to these angles. Next, we measure I_{RCP} and I_{LCP} by inserting the quarter-wave plate and setting it to 0° and 90° , respectively, while keeping the analyzer fixed at 45° .

In the fully automated setup, the quarter-wave plate is not physically removed and reinserted during the first four measurements. Instead, its rotation is synchronized with the analyzer: it is set at -45° relative to the transmission axis of the polarizer and rotated simultaneously with the analyzer. This ensures that the measurement results are unchanged, apart from an overall constant multiplier, which is compensated for in the final computation.

2. The rotating quarter-wave plate (RQWP) method, in which the quarter-wave plate is rotated stepwise in front of a fixed analyzer. The Stokes vectors are obtained from Fourier

analysis of intensity measurements and can be written as follows [13], [14]:

$$\begin{aligned}
S_0 &= \frac{2}{N} \sum_{n=1}^N I_n - \frac{4}{N} \sum_{n=1}^N I_n \cos 4\theta_n, \\
S_1 &= 2 \frac{4}{N} \sum_{n=1}^N I_n \cos 4\theta_n, \\
S_2 &= 2 \frac{4}{N} \sum_{n=1}^N I_n \sin 4\theta_n, \\
S_3 &= \frac{4}{N} \sum_{n=1}^N I_n \sin 2\theta_n,
\end{aligned} \tag{6}$$

where I_n are the measured intensities for n different angles, and θ_n are the corresponding rotation angles of the quarter-wave plate. By the Nyquist–Shannon sampling theorem, a band-limited signal can be perfectly reconstructed if it is sampled at a rate greater than twice its highest frequency. In a rotating-QWP polarimeter, the detected intensity $I(\theta)$ contains harmonics up to the fourth order (terms in 4θ , with $\theta(t) = 2\pi ft$, this corresponds to a temporal frequency $4f$) [13]. To recover the Stokes parameters S_0, S_1, S_2, S_3 without aliasing, it is standard to take $N = 8$ independent, uniformly spaced measurements over 180° , giving the step

$$\Delta\theta = \frac{180^\circ}{8} = 22.5^\circ.$$

A convenient choice is

$$\theta_n \in \{0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ, 90^\circ, 112.5^\circ, 135^\circ, 157.5^\circ\}.$$

We have implemented both methods. In the following chapters, we will discuss in detail the advantages and disadvantages of each, and explain why we chose to proceed with the RQWP method.

2.1.3 Photoelasticity and Stress-Optic Law

Photoelasticity, also known as mechanical birefringence, is a phenomenon in which a material subjected to mechanical stress exhibits birefringent properties, causing the optical properties to vary with the direction of polarization. The essence of photoelasticity lies in the fact that mechanical stress within a material perturbs its refractive index. When light interacts with such a stressed, photoelastic medium, these perturbations manifest as changes

in the polarization state of the transmitted or reflected light. This intrinsic link between mechanical stress as the cause and optical modulation as the effect forms the foundation of photoelastic analysis. To quantify this link, the stress–optic law relates the change in refractive index to the applied stress. Specifically, the stress-induced birefringence produces a phase retardation between orthogonal components of light that is proportional to the principal stress difference $\sigma_1 - \sigma_2$, where σ_1 and σ_2 are the orthogonal principal stresses [11].

For a linear, isotropic photoelastic medium under plane stress, the stress–optic law is

$$\Delta n = C(\sigma_1 - \sigma_2), \quad (7)$$

where $C = C(\lambda)$ is the wavelength-dependent stress–optic coefficient and σ_1, σ_2 are the in-plane principal stresses. The stress-induced birefringence introduces a phase delay between two orthogonal polarization components given by

$$\delta = \frac{2\pi}{\lambda} \Delta n L, \quad (8)$$

with L the *geometric* path length traversed by the ray through the birefringent region (i.e., the physical thickness along the propagation direction). Equivalently, the optical path difference is $\text{OPD} = \Delta n L$, so $\delta = \frac{2\pi}{\lambda} \text{OPD}$. If Δn varies along the path, the phase delay generalizes to $\delta = \frac{2\pi}{\lambda} \int \Delta n(s) ds$ [12].

At bonded interfaces, voids frequently adopt a dome-like profile that concentrates stress around their perimeters. This spatially varying principal-stress difference modulates Δn via (7) and, through (8), produces measurable retardance patterns that can be mapped by polarization-based inspection.

2.2 Polarimetric Microscope

Having established the theoretical framework necessary to understand stress-optic effects, we can now turn to the experimental implementation. In this section we present the optical bench that was designed and assembled to perform the photoelastic measurements. The optical bench consists of two main modules - the illumination module and the detection module (see Figure 4). The illumination module consists of the source, which is a 1310 nm LED source, the power of which can be controlled. The wavelength was chosen in the near-infrared region because it can penetrate the wafer and reach the internal layer interfaces, which are the main region of interest. The next element of the layout is a diffuser that is

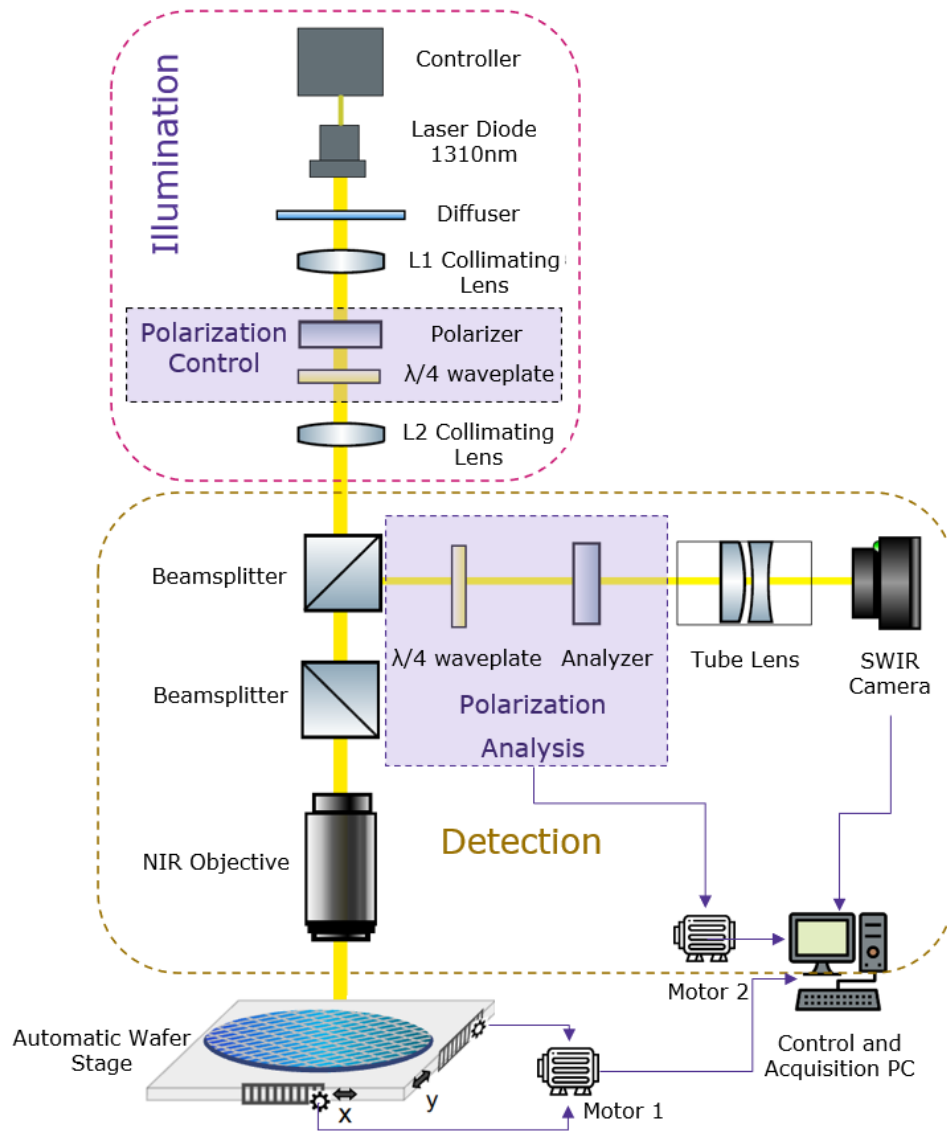


Figure 4. Schematic of the polarimetric microscope system. The setup has two main modules: (i) an illumination module with polarization control and a light source, and (ii) a detection module with automated polarimetric analysis, an XY wafer stage (two degrees of freedom), and a short-wave infrared (SWIR) camera.

positioned right after the source to break the spatial coherence that we have in the beam, in order to avoid interference effects. It is important to carefully adjust the rotation frequency of the diffuser by controlling the input current, because very high angular frequencies cause vibrations of the optical table and the wafer stage resulting image vibrations.

Lens L1 is placed immediately after the light source to collimate the diverging beam. Lens L2 is positioned such that it images the source onto the condenser aperture plane, thereby

achieving a variation of simplified Köhler illumination.

Köhler illumination achieves homogeneous specimen illumination by using two distinct sets of conjugate planes: the illumination planes, where the light source, the condenser aperture diaphragm and the objective rear aperture are optically conjugate, and the field planes, where the field diaphragm, the specimen and the intermediate image plane are optically conjugate. This separation ensures that the image of the light source does not appear in the specimen plane allowing independent control over the illuminated field size and the aperture [15].

In the illumination module, we use a polarizer and a quarter-wave plate, which together allow us to generate circularly polarized light. First, the linear polarizer is positioned to provide linearly polarized light. Then, the fast axis of the quarter-wave plate is set at a -45° angle relative to the transmission axis of the linear polarizer, producing left-hand circularly polarized light.

The detection module contains two identical beam splitters positioned inversely to compensate any parasitic phase shift in polarization that brings the anti-reflection coating of the first beam splitter. Cube beam splitter has been chosen over plate beam splitter, because the later has more effect on the incoming wave polarization. Since the system is in reflection mode the objective acts both as a condenser and a collecting lens of magnification $20\times$.

The objective works in pair with the tube lens, allowing to benefit from infinity-corrected system, meaning that the light collected from the specimen is collimated and only focuses after it passes through the tube lens. Infinity-corrected system allows the insertion of different optical elements in the path that require normal incidence. The beam is focused on the sample. The sample holder is an 300 mm wafer stage attached to the Motor 1, such that it is possible to control its position in both x and y directions. The beam reflected from the sample passes through the polarization analysis block, which consists of a quarter-wave plate and an analyzer (linear polarizer). Both elements are mounted on Motor 2, which automates their rotation independently and provides precise control over their orientation. At the initial position, the quarter-wave plate and analyzer are aligned relative to the input polarization control unit, defining the parallel- and cross-polarized reference configurations. This calibration provides a zero point for subsequent measurements. The purpose of this analysis block is to evaluate the sample's retardance by probing the reflected beam under different polarization states. Beyond the parallel and cross-polarized cases, additional orientations of the quarter-wave plate and analyzer can be explored to perform a more

comprehensive polarization analysis.

The imaging system employs a short-wave infrared (SWIR) camera with a spatial resolution of 640×512 pixels, operating at a wavelength of 1310 nm. We adopt a camera-based, full-field acquisition to achieve high throughput, in contrast to point-by-point methods such as acoustic scanning microscopy. Since the detector is a pixelated sensor array, each pixel can be treated as an independent point detector, with its own corresponding Stokes vector of size $(4, 1)$. Consequently, the full image can be represented as a Stokes matrix of size $(4, 640, 512)$. The camera allows adjustment of the exposure time, which should be synchronized with the angular frequency of the diffuser in order to ensure proper temporal averaging of the unwanted rotating grain pattern. Vignetting primarily limits the field of view and can be mitigated by careful optical alignment.

2.2.1 Optical system automatization and Process Workflow

To ensure precise measurements across angles and positions, we developed a fully automated system that serves as a proof-of-concept for autonomous wafer scanning. As mentioned in the previous section, both the wafer stage and the polarization analysis block in the optical bench are fully automated. Given the small field of view (FOV), we therefore adopted a scanning approach to map larger areas of the wafer surface. The scan is performed in a serpentine pattern, moving sequentially through each (x_i, y_i) position on the wafer, as illustrated in Figure 5. After acquiring all the images, the complete map is generated by “stitching” the individual frames together in the correct order. In the classical method, six images are captured at each position, whereas in the RQWP method, eight images are required.

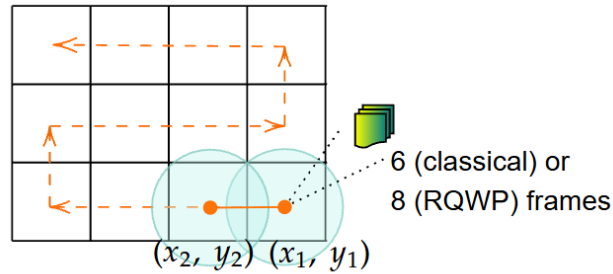


Figure 5. Serpentine scanning pattern for image acquisition

Because the setup includes two automated subsystems—the wafer stage and the polarization analysis unit—it is natural to ask how best to sequence their movements. Two strate-

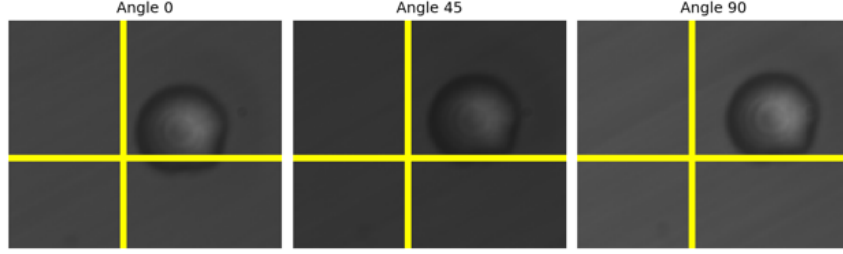


Figure 6. The rotation of the images due to the analyzer rotation, the features are slightly shifting up and right as we change the linear polarizer angle from 0° to 90° . The yellow cross-lines serve as a visual reference.

gies were considered:

1.Per-position configuration method – At each (x_i, y_i) position, cycle through all QWP/analyzer configurations before moving to the next coordinate.

2.Full-area pass method – Set a given configuration of the quarter-wave plate (QWP) and analyzer, scan the entire surface, then change to the next configuration and repeat until all required configurations (six or eight) are completed.

To determine which strategy is more efficient, we performed a benchmark scan over a $2\text{ mm} \times 2\text{ mm}$ region of the wafer and measured the total running time for each approach. The results are summarized in Table 1.

Table 1. Scanning configurations and acquisition time for a $2\text{ mm} \times 2\text{ mm}$ area.

	Time (sec)	Num of Positions	Acquired images
Per-position config			
Classical	1125	30	180
RQWP	974	30	240
Full area pass			
Classical	225	30	180
RQWP	276	30	240

In the classical method, we observed that images acquired from the same position but with different analyzer configurations showed a slight rotation of identical features, as illustrated in Figure 6. These images were taken from a small ASE-Die sample ($12\text{ mm} \times 8\text{ mm}$), which was used as a test prior to receiving the wafer samples. This effect is attributed to a slight tilt of the beam incident on the analyzer, which requires extremely precise alignment to eliminate.

Although the classical method offered a slight advantage in acquisition time, the observed

image rotation artifacts made it unsuitable for our application. For this reason, we proceeded with the full-area pass RQWP method, which provided stable and consistent image alignment across all polarization configurations despite the increase in total measurement time.

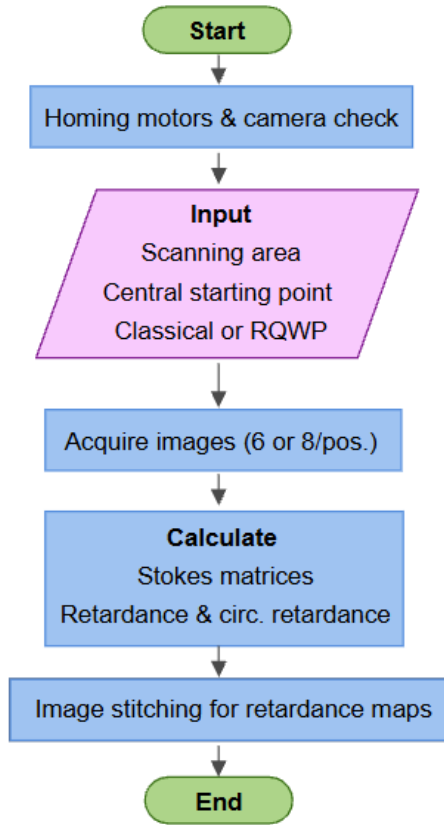


Figure 7. Workflow of the measurement process, from system initialization to retardance map generation.

In the Figure 7 we present the process workflow in a flowchart. For that, we have developed Python scripts supported by the modules required to run the camera and the motors. Although in our current measurements we use the RQWP approach, we have retained the classical method in the workflow to allow for future cross-validation between the two methods.

2.2.2 Polarimetric Imaging System Characterization

Once the optical bench assembly was complete, we performed imaging system characterization before passing to the actual experiments. In this part of the thesis we present the metrics we have used to evaluate our imaging system and discuss the theoretical expectations with laboratory measurements.

As a starting point, we made a rough estimate of the resolution limit using the theoretical value given by the Rayleigh criterion, based on the optical specifications provided in the manufacturer's datasheet.

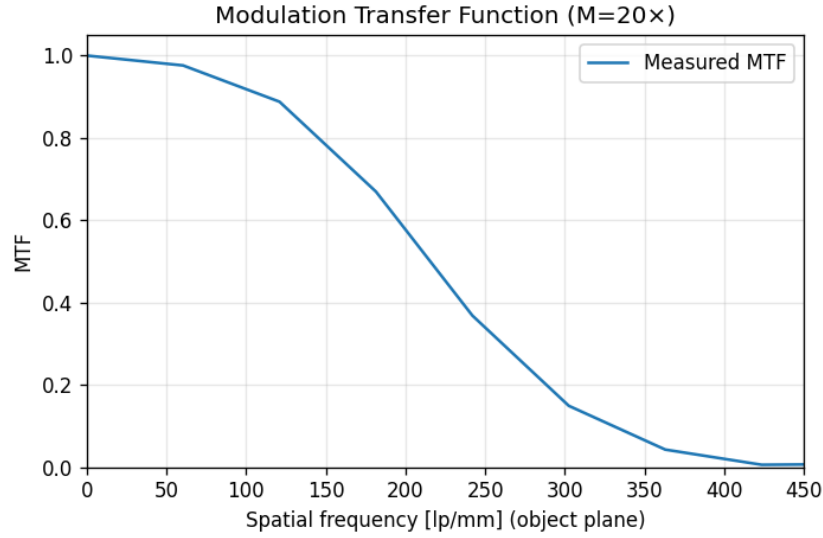


Figure 8. Modulation transfer function calculated with the slanted edge method in the object plane using an objective of 20 $\times$ magnification.

For our 20 $\times$ Mitutoyo Plan Apo NIR objective (NA 0.40, WD 20 mm, Edmund Optics, stock #46–404), the theoretical resolution limit calculated from the Rayleigh criterion is 1.99 μm . We then measured the field of view (FOV) of the imaging system with the same objective, obtaining a horizontal FOV of 0.50 mm and a vertical FOV of 0.40 mm. From these values, the corresponding pixel size was determined to be 0.78 μm .

2.2.2.1 Modulation Transfer Function The modulation transfer function (MTF) describes how effectively an optical system can reproduce (or “transfer”) different levels of detail from the object to the image. It is defined as the ratio of image contrast to object contrast as a function of spatial frequency, providing a quantitative measure of the system’s resolution and contrast performance [16].

To evaluate the MTF of our imaging system experimental measurements were performed using the slanted-edge method [17].

In this approach, a high-contrast edge is imaged with a slight tilt relative to the pixel grid. The tilt allows the edge profile to be reconstructed at a resolution finer than the camera pixels. From this edge profile (edge-spread function), we obtain the line-spread function by differentiation, and its Fourier transform gives the MTF, which describes how image contrast is preserved across spatial frequencies.

The measured curve is normalized at low frequency and expressed in line pairs per millimetre in the object plane, using the known magnification (20×) and camera sampling (0.78 μm/pixel), which sets the Nyquist limit at 423 lp/mm. A small amount of smoothing and windowing is applied to reduce noise.

This single-image technique is efficient, robust, and consistent with results from more elaborate test patterns, which is why it forms the basis of the ISO 12233 standard. It provides the system MTF, which includes both the optics and the camera sensor. Careful choice of edge angle, averaging, and region of interest ensures good signal-to-noise ratio and minimizes aliasing [17].

2.2.2.2 Characterization of Polarization Analysis Although we use a polarization control unit to generate circularly polarized light, various elements along the optical path can alter the polarization state and partially depolarize the beam, leaving a fraction of the light unpolarized or elliptically polarized. To estimate the purity of the polarization state, we used the *degree of polarization (DoP)* as the primary metric to measure the amount of polarized light in the beam. The DoP quantifies how polarized a light beam is and is defined as:

$$DoP = \frac{\sqrt{S_1^2 + S_2^2 + S_3^2}}{S_0} \quad (9)$$

[18].

Alternatively, the DoP can also be expressed in terms of intensity measurements as:

$$DoP = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} \quad (10)$$

This formulation allows practical measurement using a powermeter in combination with a linear polarizer acting as an analyzer. The analyzer used in our experiments has an extinction ratio of $1 : 10^7$, as reported by the manufacturer, ensuring that the measurement itself introduces negligible depolarization.

We measured the DoP at two locations in the optical system:

1. **Immediately before the sample**, where the DoP was found to be approximately 0.95.
2. **After reflection from the beam splitter**, right before entering the polarization analysis unit, where the DoP dropped to 0.72.

The significant reduction in DoP is likely due to imperfections in the beam splitters. However, other contributing factors may include surface roughness and coating imperfections of the objective, which can induce depolarization, multiple reflections and scattering inside the beam splitter, leading to mixing of polarization states and misalignment of the optical path or finite extinction ratio of the polarizer used in measurement.

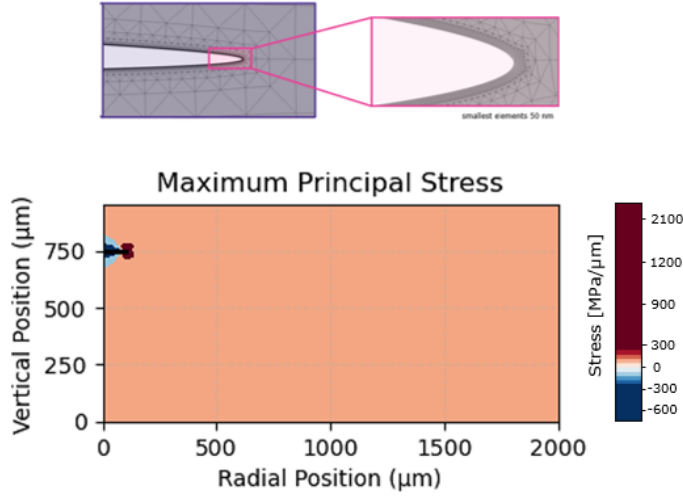


Figure 9. Computational model of voids: finite element mesh and stress distributions around a void in the silicon wafer with element size down to 50 nm and cross-sectional maximum principal stress distribution

3 Stress Modeling and Experimental Evaluation of Voids

In this section, we present the software modeling of voids and the resulting mechanical stress they generate in their vicinity. We then show the maximum principal stress distribution and the shear stress distribution obtained from the model. Following this, we present the results from our polarimetric imaging experiments and discuss the observed retardance patterns around voids within the wafer.

3.1 Simulation of Void-Induced Mechanical Response

The simulation was performed, in collaboration with the mechanical team, using a finite-element model of a representative void within the wafer. The void depth was $200\ \mu\text{m}$, with a radius of $100\ \mu\text{m}$, and the material was silicon (Si). To accurately capture the high stress gradients near the void tip, the mesh was locally refined, with the smallest elements of 50 nm (see Figure 9).

The maximum principal stress map indicates where tensile stresses are most intense, with the highest values occurring at the void tip. This region is critical because high tensile principal stress increases the likelihood of crack initiation and subsequent propagation. The mechanical stress effect, visible in the red–blueish regions of the maximum principal stress image, extends laterally from approximately $50\ \mu\text{m}$ to $150\ \mu\text{m}$ around the void.

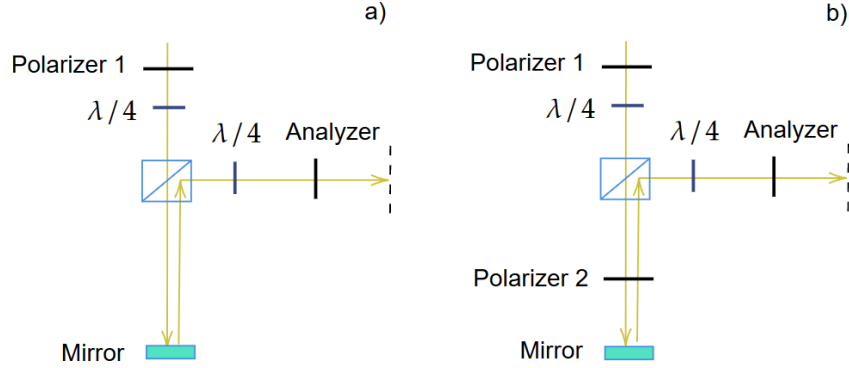


Figure 10. Two configurations for validation of the detection module; (a) Mirror configuration to validate the detection of circular polarized light; (b) Mirror-polarizer configuration to validate the detection of linear polarized light.

By analyzing the maximum principal stress distribution, the study highlights how void geometry directly affects local mechanical integrity.

3.2 Experimental Validation and Retardance Analysis

In this section, we present the experimental validation of the polarimetric imaging system and its application to retardance mapping of bonded silicon wafers. The validation process aims to confirm that the system accurately measures known polarization states before applying it to real samples. Following the validation, we demonstrate how the system can detect structural features such as bonding voids by analyzing both linear and circular retardance distributions across the wafer surface.

3.2.1 Benchmarking of the Polarimetric Imaging System

We configured the polarization state from the polarization control module to produce left circular polarization. Consequently, the reflected beam from the mirror is expected to be right circularly polarized. To validate this, we computed the Stokes parameters using a mirror in place of the actual sample. This configuration with mirror is shown in the Figure 10 (a). In the Figure 11 we show the Stokes matrices obtained for that configuration, the mean value μ is displayed for each S_i parameter. The mean values of the S_1 and S_2 are close to zero, indicating that the linear polarization components in the horizontal-vertical and $\pm 45^\circ$ orientations are negligible. In contrast, the mean value of the S_3 parameter reaches approximately 0.66, confirming the right-handedness of the circularly polarized light.

The mirror test confirmed that the light is predominantly left circularly polarized before

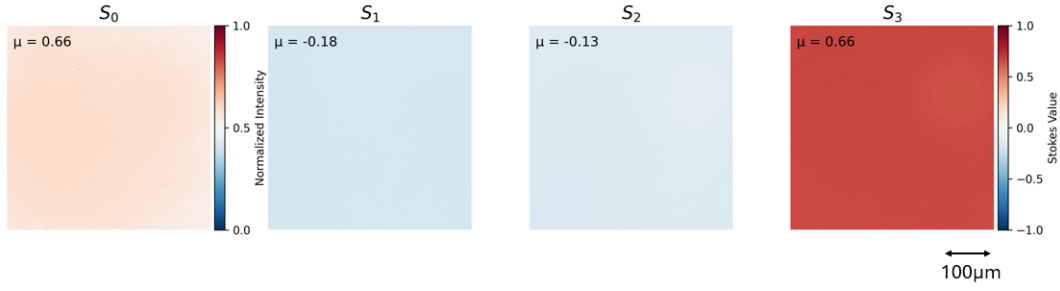


Figure 11. Stokes parameters calculated for the mirror, the mean values of S_1 and S_2 are almost 0 and for S_3 $\mu = 0.66$, meaning that the light is mostly right circular polarized (S_1 , S_2 and S_3 values range from -1 to 1).

reaching the sample. However, this alone does not guarantee that the polarimetric microscope measures all polarization states accurately—particularly linear ones.

To address this, we performed an additional experiment prior to placing the Si wafer under the microscope. In this configuration, a second polarizer (P2) was inserted just before the mirror in the optical path as shown in the Figure 12 (b), enabling precise control of the polarization state of the reflected beam. Since the mirror itself does not exhibit photoelasticity, circular polarization should only undergo a handedness reversal upon reflection, while linear polarization states should remain unchanged apart from this reversal. By adjusting the orientation of P2 relative to the system reference (zero position), we generated well-defined polarization states at the output: horizontal, vertical, 45° , and 135° linear polarizations.

Figure 12 shows the measured Stokes parameter maps for each P2 configuration, with the mean value μ indicated in each case. Nonzero S_3 values were observed for some linear polarization configurations (particularly horizontal and 45°), which are not ideal and likely arise from residual phase retardance in the optical path. Potential sources include slight birefringence of the beam splitter and objective, as well as imperfect alignment of P2 or the analyzer. Such effects can introduce a small circular polarization component into an otherwise purely linear state, leading to the detected nonzero S_3 . Among these, the dominant contribution is attributed to the beam splitter, which is not polarization-preserving in reflection mode at 45° .

3.2.2 Detection of Voids Via Retardance Maps

We define linear retardance as follows:

$$R = \arctan \frac{S_2}{S_1} \quad (11)$$

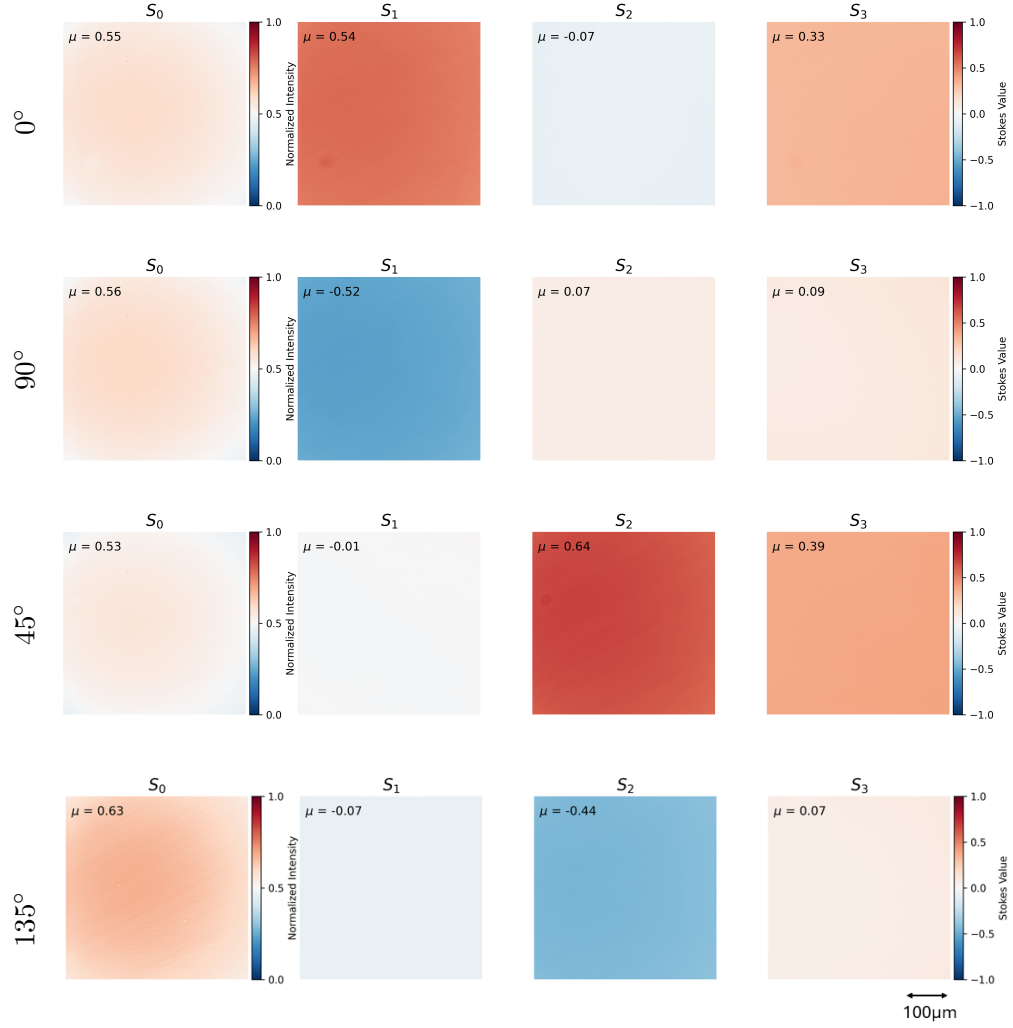


Figure 12. Stokes parameter maps (S_0, S_1, S_2, S_3) obtained for different input linear polarization states (0° (horizontal), 90° (vertical), $45^\circ, 135^\circ$) using the mirror-polarizer configuration. Mean values (μ) for each parameter are indicated.

The linear retardance R provides a measure of the phase difference between two orthogonal linear polarization components, this metric is useful for revealing stress-induced birefringence.

To generate the retardance map, the measured Stokes parameter images were stitched together from multiple field-of-view acquisitions. Each tile was processed individually to compute R according to Eq. 11. However, this per-tile computation results in variations in the absolute retardance scale between tiles, requiring careful selection of color map limits to preserve feature visibility across the stitched map.

To illustrate these considerations, we applied the method to a bonded silicon wafer sample. The wafer consists of two silicon wafers joined together, each having thickness of $775\ \mu\text{m}$, and the beam was focused on the bonding interface, because voids occur in this region. Acoustic C-scan and PSD images of the wafer were used as references to locate the scanned area and compare the results. Figure 13 presents the C-scan and PSD images in the first row, alongside the same area measured with our system. Panel (c) shows the stitched intensity map, and panel (d) shows the corresponding linear retardance map.

The retardance map is shown as a false-color image computed separately for each tile. As a result, a global minimum and maximum could not be applied consistently across the full stitched map. Furthermore, selecting optimal colormap limits was challenging: in background regions the values vary little, whereas in areas with voids they can span the full arctan range. Setting the colormap to the full range from $-\pi$ to $+\pi$ causes small features to become less visible. We therefore chose to present two versions of the map: one optimized for small-scale features and another for larger-scale structures.

In Figure 13, the intensity and retardance maps allow us to identify all voids previously reported by the C-scan and PSD measurements. These voids range from approximately 0.9 mm to 1.6 mm in diameter. The retardance map reveals a distinct polarimetric footprint: in the lower-right quadrant of each void, the retardance values are higher (appearing in red), while in the upper-left quadrant they are lower (appearing in dark blue). In contrast, the lower-left and upper-right regions display intermediate retardance values (light blue), indicating a transition between the two extremes.

This asymmetric retardance pattern likely reflects the local stress field and birefringence orientation around each void. The higher values in the lower-right quadrant and lower values in the upper-left quadrant suggest a preferential alignment of the stress-induced optical axis, possibly influenced by the void's geometry and the surrounding material's elastic

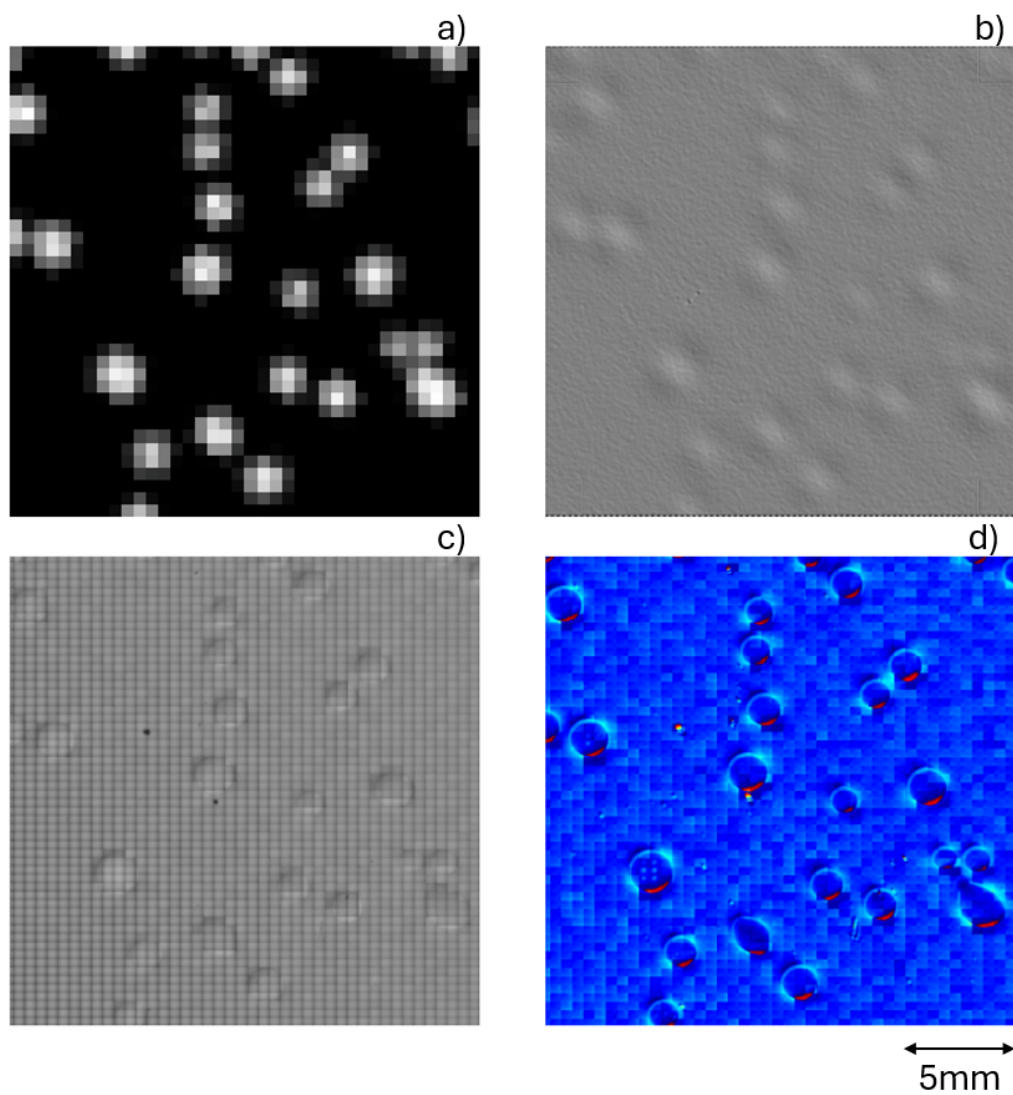


Figure 13. Computational model of wafer voids with corresponding measurements: (a) C-scan crop (with inset); (b) power spectral density (PSD) map; (c) intensity map; (d) retardance map.

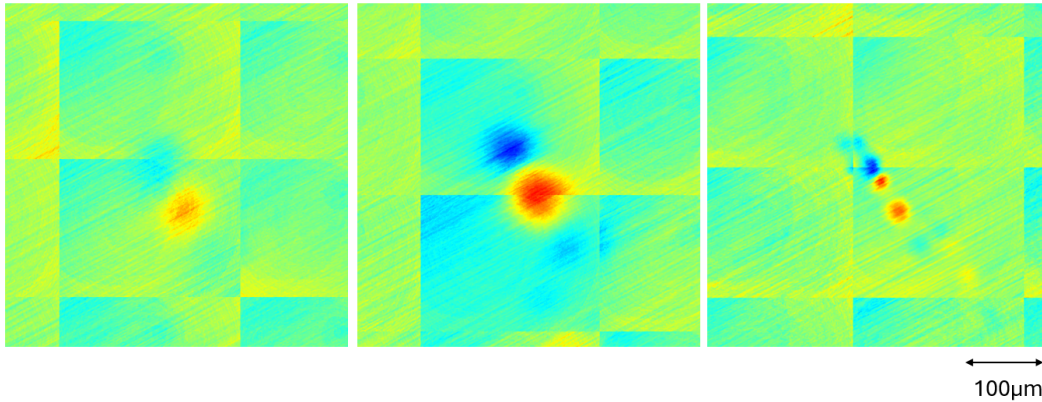


Figure 14. The retardance map adjusted for the smaller voids, the features range from 20 – 150 μm . and have different retardance values

properties. The intermediate (light blue) regions in the lower-left and upper-right areas may correspond to zones where the principal stress directions transition, producing retardance values closer to the mean.

As mentioned previously, smaller features require careful adjustment of the colormap to become visible. In Figure 14, we highlight the areas where such features were observed using this adapted scaling. Notably, these features are absent in both the C-scan and PSD maps, meanwhile our approach was able to detect them. They exhibit a diagonal pattern of alternating high and low retardance values, running from the upper-left to the lower-right corner—resembling the footprint observed for larger voids. While it is challenging to provide a precise estimate of the void size to which these features may be related, we can quantify the size of the observed features themselves, which ranges from approximately 20 – 150 μm .

Recalling the void simulations presented in Section 3.1, we find that our experimental observations are consistent with the simulated stress fields. In particular, the measured retardance maps exhibit blue-to-red transitions that closely resemble the compression and expansion patterns associated with the maximum principal stress. While it is challenging to identify voids of precisely 100 μm in radius for a direct quantitative comparison with the simulated case, our ongoing investigations aim to further refine this correspondence and evaluate the stress distribution around experimentally observed voids with greater accuracy.

During some of the measurements, we noticed faint ghost features in the retardance maps,

caused by a small mechanical shift of just a few micrometers in the wafer position during acquisition. With the thinner test wafer we had used earlier, this issue did not appear. However, once we switched to the thicker double-bonded wafer, its added height made the sample slightly less stable in the holder, which introduced the shift. Looking back, we can also see a trace of the same effect in our scan, though it was less visible there since the wafer positioning on the stage was better.

Due to limited wafer availability, we could not repeat the experiment with a more robust holding setup. We did explore image-processing methods to correct the misalignment, though these were complicated by the presence of vignetting in the images. Still, the experience underlines how sensitive the acquisition is to small mechanical details, and points us toward improvements for future measurements.

4 Conclusion

This thesis explored the use of polarimetric imaging as a non-contact technique for detecting interfacial voids in bonded semiconductor wafers. To this end, we developed a reflection-mode polarimetric microscope operating in NIR range, equipped with automated polarization control and a serpentine scanning stage. Alongside the instrument design, we established a practical workflow for estimating the Stokes parameters and producing retardance maps over millimetre-scale areas. A careful system characterization—covering field of view, resolution, modulation transfer function, and degree of polarization—together with controlled polarization tests provided a solid experimental foundation for measurements on bonded silicon wafers.

Finite-element simulations of a representative void reproduced the expected localization and anisotropy of the associated stress field. Through the stress–optic relation, these simulations motivated the search for asymmetric retardance patterns in experiment. The measured retardance maps indeed revealed quadrant-dependent features consistent with a preferential orientation of the stress-induced optical axis in the vicinity of voids. Across stitched fields of view, every void identified by acoustic C-scan and PSD references was recovered in our polarimetric reconstructions, and we additionally observed smaller features that exhibited a similar signature.

A full-area-pass strategy combined with the rotating quarter-wave-plate method provided more stable alignment at the cost of modestly longer runtimes compared to that with the classical method, and was ultimately adopted for wafer measurements.

The study also brought to light several practical limitations. Mechanical drift of the stage during scanning led to ghosting near wafer edges; vignetting complicated image corrections; and tile-wise retardance scaling made it difficult to construct globally consistent mosaics without losing small-scale features. In the polarization path, a drop in the degree of polarization between the source and analyzer suggested residual birefringence and scattering from beam splitters and mirrors. Finally, the overall throughput of the scanning and stitching pipeline remains below what would be required for high-volume inspection.

Even with these constraints, the results show that polarimetric imaging can serve as a valuable complementary tool for subsurface inspection in semiconductor manufacturing. By exploiting stress-induced birefringence that extends beyond the physical footprint of voids, the method not only enhances overall defect visibility but also enables the detection of

much smaller features than those resolved by conventional techniques. This provides access to much smaller voids and fine stress patterns that were not captured with acoustic or interferometric methods.

Future directions. This study should be regarded as a proof of concept rather than a finalized inspection solution. The results demonstrate that reflection-mode polarimetric imaging can reliably reveal voids, including features down to a few tens of micrometres, but the current implementation remains a laboratory prototype with clear room for improvement. Several targeted improvements could strengthen the approach and improve scalability: (i) mechanical refinements such as stiffer, closed-loop stages and improved vibration isolation; (ii) more uniform illumination and detection through flat-field calibration, vignetting correction, and a consistent retardance scaling strategy for mosaics; (iii) optimization of the polarization path with low-birefringence components, improved coatings, and fine alignment to preserve a high degree of polarization; (iv) calibrated, quantitative stress mapping by linking retardance directly to material constants and thickness; (v) faster acquisition and reconstruction enabled by multi-threaded control and GPU-based computation; and (vi) broader validation across wafer types and bonding schemes, including multi-wavelength operation and joint analyses with SAM and PSD.

In summary, this work establishes both the feasibility and the promise of reflection-mode polarimetric imaging for wafer-void inspection. It lays out an instrument architecture and workflow that successfully recover voids across a range of sizes, while also identifying the technical refinements needed for an industrially viable, high-speed implementation. Beyond its immediate application, this study highlights the potential of polarimetric imaging to enrich the broader toolkit of semiconductor inspection technologies.

References

- [1] M. Inês Silva, E. Malitckii, T. G. Santos, and P. Vilaça, "Review of conventional and advanced non-destructive testing techniques for detection and characterization of small-scale defects," *Progress in Materials Science*, vol. 138, p. 101 155, 2023, issn: 0079-6425. DOI: <https://doi.org/10.1016/j.pmatsci.2023.101155>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0079642523000877>.
- [2] J.-Q. Lu, "3-d hyperintegration and packaging technologies for micro-nano systems," *Proceedings of the IEEE*, vol. 97, pp. 18–30, Feb. 2009. DOI: 10.1109/JPROC.2008.2007458.
- [3] Techovedas. "What is 2d, 2.5d & 3d packaging of integrated chips?" Accessed 2025-08-19. (Nov. 23, 2023), [Online]. Available: <https://techovedas.com/what-is-2d-2-5d-3d-packaging-of-integrated-chips/>.
- [4] Y.-S. Kim, T. Nguyen, and S.-H. Choa, "Enhancement of the bond strength and reduction of wafer edge voids in hybrid bonding," *Micromachines*, vol. 13, p. 537, Mar. 2022. DOI: 10.3390/mi13040537.
- [5] C. Wang, Y. Liu, and T. Suga, "A comparative study: Void formation in silicon wafer direct bonding by oxygen plasma activation with and without fluorine," *ECS Journal of Solid State Science and Technology*, vol. 6, no. 1, P7, Dec. 2016. DOI: 10.1149/2.0311612jss. [Online]. Available: <https://dx.doi.org/10.1149/2.0311612jss>.
- [6] Y. Guo, C. Zong, and T. Britton, "Development of local plasticity around voids during tensile deformation," *Materials Science and Engineering: A*, vol. 814, p. 141 227, 2021, issn: 0921-5093. DOI: <https://doi.org/10.1016/j.msea.2021.141227>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921509321004962>.
- [7] F. Nagano, S. Iacovo, A. Phommahaxay, *et al.*, "Void formation mechanism related to particles during wafer-to-wafer direct bonding," *ECS Journal of Solid State Science and Technology*, vol. 11, Jun. 2022. DOI: 10.1149/2162-8777/ac7662.
- [8] C. Chen, J. Slabbekoorn, J. Bogdanowicz, *et al.*, "Detection of bonding voids in multi-tier stacks with scanning acoustic microscope," in *2023 IEEE 25th Electronics Packaging Technology Conference (EPTC)*, 2023, pp. 320–325. DOI: 10.1109/EPTC59621.2023.10457746.

- [9] N. Poduje, W. Kerr, and K. T. Turner, "Inspection of bonded interfaces using scanning infrared interferometry," *ECS Transactions*, vol. 33, no. 4, p. 537, Oct. 2010. DOI: 10.1149/1.3483545. [Online]. Available: <https://dx.doi.org/10.1149/1.3483545>.
- [10] O. Innovation, *Echoscan™ void detection system*, Accessed: 2025-08-06, 2025. [Online]. Available: <https://ontoinnovation.com/products/echo-scan-void-detection>.
- [11] E. Hecht, *Optics*, 5th ed. Pearson, 2017, ISBN: 978-0133977226.
- [12] D. H. Goldstein, *Polarized Light*, 3rd ed. CRC Press, 2011, ISBN: 978-1439849040.
- [13] B. Schaefer, E. Collett, R. Smyth, D. Barrett, and B. Fraher, "The rotating quarter-waveplate method for measuring the stokes polarization parameters," *American Journal of Physics*, vol. 75, no. 2, pp. 165–170, 2007. DOI: 10.1119/1.2386162. [Online]. Available: <https://pol3he.sites.wm.edu/wp-content/uploads/sites/50/2020/01/measuring-Stokes-parameters.pdf>.
- [14] T. Kihara, "Measurement method of stokes parameters using a quarter-wave plate with phase difference errors," *Applied Optics*, vol. 50, pp. 2582–2587, Jun. 2011. DOI: 10.1364/AO.50.002582.
- [15] Z. Microscopy. "Microscopy basics | köhler illumination." Accessed: 2025-08-13. (n.d.), [Online]. Available: <https://zeiss-campus.magnet.fsu.edu/articles/basics/kohler.html>.
- [16] J. W. Goodman, *Introduction to Fourier Optics*, 3rd ed. Roberts and Company Publishers, 2005, ISBN: 978-0974707723.
- [17] X. Zhang, T. Kashti, D. Kella, *et al.*, "Measuring the modulation transfer function of image capture devices: What do the numbers really mean?," vol. 8293, pp. 6–, Jan. 2012. DOI: 10.1117/12.912989.
- [18] B. DeBoo, J. Sasian, and R. Chipman, "Degree of polarization surfaces and maps for analysis of depolarization," *Opt. Express*, vol. 12, no. 20, pp. 4941–4958, Oct. 2004. DOI: 10.1364/OPEX.12.004941. [Online]. Available: <https://opg.optica.org/oe/abstract.cfm?URI=oe-12-20-4941>.

Statement of Non-Plagiarism

I hereby declare that the research and work presented in this report are entirely my own, conducted in strict adherence to the academic regulations and ethical standards of the EMJMD PSRS. All external sources and contributions have been duly acknowledged and referenced. I confirm that this report has not been submitted, either in part or in full, for any other degree or qualification at any academic institution.

Anahit Hambardzumyan

Statement of AI Tools Usage

I hereby declare that generative AI tools, including large language models, were employed in the preparation of this thesis solely for ancillary purposes such as improving clarity of expression, assisting in the refinement of language, and providing structural suggestions. The substantive intellectual contributions, including research design, data analysis, simulations, interpretation, and conclusions, are entirely my own, and I critically reviewed and edited all AI-generated text.

Date and signature of the author:



25/08/2025

Supervisor Approval

I, the undersigned, **Davit Hakobyan**, supervisor of **Anahit Hambardzumyan**, a student of the Erasmus Mundus Joint Master Degree in Photonics for Security, Reliability, and Safety (EMJMD PSRS), during his internship at **UNITY SC**, hereby certify that I approve the content of his research internship report entitled *"Polarimetric Imaging Applied For Void Detection In Silicon Wafers"*.



Davit Hakobyan
Supervisor

Date: Grenoble, 25/08/2025

Appendices

Listing 1. Control and Acquisition

```
import sys
import thorlabs_apr as apr
import time
import cv2
import numpy as np
from pathlib import Path
# from datetime import datetime
from tqdm import tqdm
import json

sys.path.append("utils\\")
import motors_functions as mtf
from controller import XYStageHandler, Camera

# Acquisition parameters
SETTLE_DELAY = 0.2
CLASSICAL = False
EXPOSURE_TIME = 20000 #4000 #11000
output_root = Path("output")
subfolder = "NCAP_scan_20x"
session_folder = "6_junction_letters_20over20"
BASE_FOLDER = output_root / subfolder / session_folder
BASE_FOLDER.mkdir(parents=True, exist_ok=True)

MARGIN = (10, 10) # (x, y) in mm
FOV = (0.5, 0.4)
STEP_OVER = 10 # % overlap between consecutive images
TARGET = (45.800, 16.8)
x_range = np.arange(TARGET[0] - MARGIN[0], TARGET[0] + MARGIN[0], FOV[0] * (1
    - STEP_OVER))
y_range = np.arange(TARGET[1] - MARGIN[1], TARGET[1] + MARGIN[1], FOV[1] *
    (1 - STEP_OVER))

config = {
    "MARGIN": MARGIN,
    "FOV": FOV,
    "STEP_OVER": STEP_OVER,
```

```

    "TARGET": TARGET,
    "RUNNING_TIME": None # to be filled later
}

print("x_range:", x_range)
print("y_range:", y_range)

if __name__ == "__main__":
    XY = XYStageHandler()
    assert XY.simulation_mode == False
    cam = Camera(exposure_time=EXPOSURE_TIME)
    cam.verify_acquisition()

    SN_motor_1 = 90877483 # QWP
    SN_motor_2 = 90877482 # Analyzer
    motor_1 = apt.Motor(SN_motor_1)
    motor_2 = apt.Motor(SN_motor_2)
    mtf.home()

    # Generate serpentine scan sequence
    sequence = []
    for i, y in enumerate(y_range):
        x_order = x_range[::-1] if i % 2 else x_range
        for x in x_order:
            sequence.append([x, y])
    num_of_pos = len(sequence)

    # Move to first position
    x_prev, y_prev = sequence[0]
    XY.move_to(x_goal=x_prev)
    XY.move_to(y_goal=y_prev)

    # Start acquisition
    image_counter = 1
    qwp_angles = np.arange(0, 180, 22.5)
    output_path = BASE_FOLDER
    output_path.mkdir(parents=True, exist_ok=True)
    init_angle = -45

    start_time = time.time()
    if CLASSICAL:
        x_prev, y_prev = None, None

```

```

# Define all steps as tuples: (polarizer angle, qwp mode, label
  prefix)
# qwp mode: "relative" for init_angle + angle, "fixed" for
  init_angle only
steps = [
    (0,      "relative", "0"),
    (45,     "relative", "45"),
    (90,     "relative", "90"),
    (135,    "relative", "135"),
    (135,    "fixed",    "cross"),
    (45,     "fixed",    "co"),
]

for pol_angle, qwp_mode, label_prefix in steps:
    mtf.abs_pos_pol(pol_angle)
    if qwp_mode == "relative":
        mtf.abs_pos_qwp(init_angle + pol_angle)
    else:
        mtf.abs_pos_qwp(init_angle)

    for x, y in tqdm(sequence):
        if y != y_prev:
            XY.move_to_fast(y_goal=y)
            y_prev = y
        if x != x_prev:
            XY.move_to_fast(x_goal=x)
            x_prev = x
            XY.move_to_fast(x_goal=x)
        time.sleep(SETTLE_DELAY)
        image = cam.acquire_frame()
        filename = output_path / f"I_{label_prefix}_{image_counter}
            }.png"
        cv2.imwrite(str(filename), image)

        image_counter += 1
        if image_counter > num_of_pos:
            image_counter = 1
    end_time = time.time()
else:
    mtf.abs_pos_pol(0)
    for angle in qwp_angles:

```

```

mtf.abs_pos_qwp(init_angle + angle)
for x, y in tqdm(sequence, desc=f"Angle {angle:.1f}°"):
    # Move if needed
    if y != y_prev:
        XY.move_to_fast(y_goal=y)
        y_prev = y
    if x != x_prev:
        XY.move_to_fast(x_goal=x)
        x_prev = x

    time.sleep(SETTLE_DELAY)
    image = cam.acquire_frame()
    filename = output_path / f"I_{angle}_{image_counter}.png"
    cv2.imwrite(str(filename), image)

    image_counter += 1
    if image_counter > num_of_pos:
        image_counter = 1

end_time = time.time()

print(f"Number of Positions: {num_of_pos}")
print(f"Total Images Acquired: {num_of_pos * len(qwp_angles)}")

elapsed = end_time - start_time
print(f"\nAcquisition completed in {elapsed:.2f} seconds")

cam.close()
mtf.home()
XY.move_to(x_goal=TARGET[0])
XY.move_to(y_goal=TARGET[1])

config["RUNNING_TIME"] = "%.2f" % elapsed

config_path = BASE_FOLDER / "config.json"
with open(config_path, "w") as f:
    json.dump(config, f, indent=4)

```

Listing 2. Helper function helper.py for the algebraic calculations

```

import cv2
import time

```

```

import numpy as np
from matplotlib import pyplot as plt
from pathlib import Path
from mpl_toolkits.axes_grid1 import make_axes_locatable
import motors_functions as mtf
motors_functions = None

QWP_SETTLE_TIME = 1.0 # seconds

def acquire_with_analyzer(camera, I, base_path, image_counter = 0):
    print("\n Acquiring images with Classical method...")
    base_path.mkdir(parents=True, exist_ok=True)
    init_angle = -45
    for angle in [0, 45, 90, 135]:
        mtf.abs_pos_pol(angle)
        mtf.abs_pos_qwp(init_angle + angle)
        time.sleep(QWP_SETTLE_TIME)
        image = camera.acquire_frame()
        I["0"][str(angle)] = image / 2
        cv2.imwrite(str(base_path / f"I_{angle}_{image_counter}.png"),
                    image)
    print(" Done with linear polarizations.\n")
    return I

def acquire_with_qwp(camera, I, base_path, image_counter = 0):
    init_angle = -45
    mtf.abs_pos_qwp(init_angle) # Set QWP to fixed retardance angle
    for angle in [135, 45]: # Order optimized
        mtf.abs_pos_pol(angle)
        image = camera.acquire_frame()
        I["pi/2"][f"{angle}°"] = image
        if angle == 45:
            cv2.imwrite(str(base_path / f"I_co_{image_counter}.png"),
                        image)
        else:
            cv2.imwrite(str(base_path / f"I_cross_{image_counter}.png"),
                        image)
    print(" Done with QWP acquisition.\n")
    return I

```

```

def acquire_classical_images(camera, base_path, image_counter = 0 ):
    I = {"0":{ }, "pi/2":{ }}
    I = acquire_with_analyzer(camera, I, base_path, image_counter)
    I = acquire_with_qwp(camera, I, base_path, image_counter)
    return I

def acquire_rqwp_images(camera, output_path, qwp_angles, analyzer_angle =0
    , image_counter = 0):

    """
    Rotate QWP through given angles and acquire images with a fixed
    analyzer.
    """
    output_path.mkdir(parents=True, exist_ok=True)
    I = {}

    mtf.abs_pos_pol(analyzer_angle)

    init_angle = -45
    print("\n Acquiring images with Rotating Quarter Waveplate method..."
        )
    for angle in qwp_angles:
        mtf.abs_pos_qwp(init_angle + angle)
        time.sleep(QWP_SETTLE_TIME) # Allow time for mtf to settle
        image = camera.acquire_frame()
        I[angle] = image
        cv2.imwrite(str(output_path / f"I_{angle}_{image_counter}.png"),
            image)

    print(" Done with QWP acquisition.\n")
    return I

def compute_stokes_from_rqwp(images_by_angle, stokes_path = None, display
    = True, send_home = True):
    """
    Computes S0, S1, S2, S3 from intensity measurements across QWP angles
    using the Fourier decomposition method.

    Parameters:
    - images_by_angle: dict {angle (deg): 2D numpy array image}

    Returns:

```

```

- s0, s1, s2, s3: 2D numpy arrays
"""

print("  Computing Stokes parameters...")
eps = 1e-8
angles_deg = sorted(images_by_angle.keys())
theta = np.radians(np.array(angles_deg)) # shape: (N,)
N = len(theta)
image_stack = np.array([images_by_angle[angle].astype(np.float32) for
    angle in angles_deg]) # shape: (N, H, W)
image_stack = np.squeeze(image_stack) # To convert form (N, 512, 640,
    1) to (N, 512, 640)
normalized_image_stack = image_stack
# Compute sine and cosine terms (shape: (N,))
sin2 = np.sin(2 * theta)[:, np.newaxis, np.newaxis] # (N,1,1) for
    broadcasting
cos4 = np.cos(4 * theta)[:, np.newaxis, np.newaxis]
sin4 = np.sin(4 * theta)[:, np.newaxis, np.newaxis]

# Vectorized sum over all images
I_sum = np.sum(normalized_image_stack, axis=0)
    # shape: (H, W)
I_sin2 = np.sum(normalized_image_stack * sin2, axis=0)
I_cos4 = np.sum(normalized_image_stack * cos4, axis=0)
I_sin4 = np.sum(normalized_image_stack * sin4, axis=0)

# Coefficient calculation
A = (2 / N) * I_sum
B = (4 / N) * I_sin2
C = (4 / N) * I_cos4
D = (4 / N) * I_sin4

# Stokes parameters
s0 = A - C
s1 = 2 * C
s2 = 2 * D
s3 = B
s1 /= s0 + eps
s2 /= s0 + eps
s3 /= s0 + eps

S = [s0, s1, s2, s3]

```

```

    if stokes_path:
        for i in range(len(S)):
            cv2.imwrite(str(stokes_path / f"s_{i}.png"), S[i])

    print(" Stokes images saved.\n")
    print(" Rotating the mtfors back to initial state.\n")
    if send_home:
        mtf.home()
    if display:
        display_stokes(S, stokes_path)
    return S

def compute_and_save_stokes(I, stokes_path=None, full=False, display=True,
    send_home=True, analyze=False):
    print(" Computing Stokes parameters...")
    if not is_flattened(I):
        I = flatten_intensity_dict(I)
    s_0 = I[0] + I[90]
    s_1 = (I[0] - I[90]) / (s_0 + np.finfo(float).eps)
    s_2 = (I[45] - I[135]) / (I[45] + I[135] + np.finfo(float).eps)

    S = [s_0, s_1, s_2]

    if full:
        s_3 = (I["co"].astype(np.float32) - I["cross"].astype(np.float32))
            / \
            (np.clip(I["co"] + I["cross"], 0, 255).astype(np.float32) +
             np.finfo(float).eps)
        S.append(s_3)

    if stokes_path:
        for i, s in enumerate(S):
            cv2.imwrite(str(stokes_path / f"s_{i}.png"), s)

    print(" Stokes images saved.\n")
    print(" Rotating the mtfors back to initial state.\n")

    if send_home:
        mtf.home()
    if display:
        display_stokes(S, stokes_path)

```

```

    if analyze:
        analyze_stokes_std(S)

    return S

def display_stokes(S, stokes_path):
    print("  Displaying Stokes images...\n")

    DoP = calculate_DoP(S)

    fig, axs = plt.subplots(2, 2, figsize=(12, 10))
    titles = [
        "s - Total Light Intensity",
        "s - Horizontal vs Vertical Polarization",
        "s - Diagonal Polarization ( $\pm 45^\circ$ )",
        "s - Circular Polarization (Right/Left)"
    ]

    S[0] = S[0]/np.max(S[0])
    print(np.max(S[0]))
    for i, ax in enumerate(axs.flat):
        # Set vmin and vmax
        if i == 0:
            vmin, vmax = np.min(S[i]), np.max(S[i]) # Autoscale s0
        else:
            vmin, vmax = -1, 1 # Normalized for -s1s3

        # Plot the image
        im = ax.imshow(S[i], cmap='viridis', vmin=vmin, vmax=vmax)

        # Add scale bar on s

        ax.text(40, 450, "200  $\mu\text{m}$ ", fontsize=8)
        ax.annotate(text='', xy=(276, 470), xytext=(20, 470),
                    arrowprops=dict(arrowstyle='<->'))

        # Set main title + subtitle
        mean_value = np.mean(S[i])
        ax.set_title(f"{titles[i]}\nmean = {mean_value:.2f}", fontsize=14)
        ax.axis('off')

    # Colorbar

```

```

        divider = make_axes_locatable(ax)
        cax = divider.append_axes("right", size="5%", pad=0.05)
        cbar = fig.colorbar(im, cax=cax)
        if i == 0:
            cbar.set_label("Intensity")
        else:
            cbar.set_label("Polarization Strength")

plt.tight_layout(rect=[0, 0.03, 1, 0.92]) # Reserve space for
suptitle
fig.suptitle(f"Degree of Polarization: {DoP:.2f}", fontsize=14) #
Must come after tight_layout
plt.savefig(str(stokes_path / "stokes.png"), dpi=300, bbox_inches='
tight')
plt.show()

def analyze_stokes_lines(S, stokes_index, random_row, random_col):
    """
    Analyze a random horizontal and vertical line from Stokes vector data.

    Parameters:
        S (np.ndarray): Stokes vectors, shape (width, height, 4)
        stokes_index (int): Which Stokes parameter to use (0, 1, 2, or 3)
        seed (int, optional): Random seed for reproducibility.

    Returns:
        dict: Contains standard deviations and line indices.
    """

    horizontal_line = S[random_row, :, :]
    vertical_line = S[:, random_col, :]

    std_horizontal = np.std(horizontal_line)
    std_vertical = np.std(vertical_line)
    fig, axs = plt.subplots(2, 1, figsize=(10, 6))

    axs[0].plot(horizontal_line)
    axs[0].set_title(f'Horizontal Line at Row {random_row} - Stokes S{
        stokes_index}')
    axs[0].set_xlabel('Pixel Position')

```

```

    axs[0].set_ylabel('Intensity')

    axs[1].plot(vertical_line)
    axs[1].set_title(f'Vertical Line at Column {random_col} - Stokes S{
        stokes_index}')
    axs[1].set_xlabel('Pixel Position')
    axs[1].set_ylabel('Intensity')

    plt.tight_layout()
    plt.show()

    return {
        'std_horizontal': std_horizontal,
        'std_vertical': std_vertical,
        'row_index': random_row,
        'col_index': random_col
    }

def calculate_DoP(S):
    return np.mean(np.sqrt(S[1]**2 + S[2]**2 + S[3]**2))

def analyze_stokes_std(S, seed = None):
    if seed is not None:
        np.random.seed(seed)
        np.random.seed(seed)
    width, height, _ = S[0].shape
    random_row = np.random.randint(-1, width - 1)
    random_col = np.random.randint(-1, height - 1)

    [analyze_stokes_lines(S[i], i, random_row, random_col ) for i in range
        (len(S))]
    return

def flatten_intensity_dict(I_old):
    I_new = {
        0: I_old["0"]["0"],
        45: I_old["0"]["45"],
        90: I_old["0"]["90"],
        135: I_old["0"]["135"],
        "co": I_old["pi/2"]["45°"],
        "cross": I_old["pi/2"]["135°"]
    }

```

```

    }
    return I_new

def is_flattened(I):
    return all(isinstance(k, (int, str)) and not isinstance(v, dict) for k
                  , v in I.items())

```

Listing 3. Stitching and Retardance map generation

```

import sys
from pathlib import Path
import numpy as np
import cv2
import glob
import os
import json
import matplotlib.pyplot as plt
from scipy.ndimage import median_filter
from mpl_toolkits.axes_grid1 import make_axes_locatable
import matplotlib.font_manager as fm

# Local imports
sys.path.append(str(Path(__file__).parent / "utils"))
from determine_cell_number import run
import helper

BASE_DIR = Path(r"6_junction_letters_20x20_v2_all_data\6
                _junction_letters_20x20_v2")
OUTPUT_DIR = Path(r"6_junction_letters_20x20_v2_all_data\
                  section_for_experiments")
# Load from same BASE_FOLDER
config_path = Path(BASE_DIR) / "config.json"

if config_path.exists():
    with open(config_path, "r") as f:
        config = json.load(f)
        MARGIN = tuple(config["MARGIN"])
        FOV = tuple(config["FOV"])
        STEP_OVER = config["STEP_OVER"]
        TARGET = tuple(config["TARGET"])
else: # This else is going to get out once we are consistent with the
      updates of control_and_acquisiton

```

```

    MARGIN = (10, 10)          # (x, y) in mm
    FOV = (0.5, 0.4)          # (hfov, vfov) in mm
    STEP_OVER = 0.10
    TARGET = (30, 30)

# --- Configuration ---
CLASSICAL = False
INDIVIDUAL = True

SECTION = False # If True, it will propose to take a section , not the
                # entire image

if SECTION:
    #SELECT_AREA: If True it will do a stitching selection of area from
    #scratch, otherwise it will use the previous one
    SELECT_AREA = input("Do you want to select a new area? (yes/no): ").
        strip().lower() == "yes"
    sample_image_path = r'6_junction_letters_20x20_v2_all_data\6
        _junction_letters_20x20_v2_all_individual\stitched_Retardance.png'

ANGLES = np.arange(0, 180, 22.5) if not CLASSICAL else [0, 45, 90, 135, "
    cross", "co"]
THRESH_PIX = 6000
OVERLAP_PIX = (int(0.1 * 640), int(0.1 * 512))
FONT_PROPS = fm.FontProperties(size=10)
WAVELENGTH_NM = 1310
PHOTOELASTIC_CONST = 2.5e-12
EPS = 1e-10

# Compute scan ranges
x_steps = np.arange(TARGET[0] - MARGIN[0], TARGET[0] + MARGIN[0], FOV[0] *
    (1 - STEP_OVER))
y_steps = np.arange(TARGET[1] - MARGIN[1], TARGET[1] + MARGIN[1], FOV[1] *
    (1 - STEP_OVER))

# Ensure output directory exists
OUTPUT_DIR.mkdir(parents=True, exist_ok=True)

# --- Utility Functions ---
def sigmoid_adjust(img, cutoff=0.6, gain=10):
    norm = (img - img.min()) / (img.max() - img.min() + EPS)

```

```

        return 1 - 1 / (np.exp(gain * (norm - cutoff)) + 1)

def add_colorbar(im, ax):
    cax = make_axes_locatable(ax).append_axes("right", size="5%", pad
        =0.05)
    plt.colorbar(im, cax=cax)

def load_gray(path):
    return cv2.imread(str(path), cv2.IMREAD_GRAYSCALE)

def calculate_retardance(S1, S2):
    return np.arctan2(S2, S1 + EPS)

def calculate_ellipticity(S1, S2, S3):
    return np.arctan2(S3, np.hypot(S1, S2) + EPS)

def get_params(image_path, scale_percent = 5):
    img = cv2.imread(image_path)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) # Convert BGR to RGB for
        matplotlib
    H, W, _ = img.shape
    grid_rows = len(y_steps)
    grid_cols = len(x_steps)

    width = int(img.shape[1] * scale_percent / 100)
    height = int(img.shape[0] * scale_percent / 100)
    img = cv2.resize(img, (width, height), interpolation=cv2.INTER_AREA)

    img_height, img_width = img.shape[0], img.shape[1]
    cell_width = img_width / grid_cols
    cell_height = img_height / grid_rows
    clicked_cells = []
    return grid_rows, grid_cols, cell_width, cell_height, clicked_cells

# --- Image Saving ---
def save_panel(data, title, cmap, output_path, with_colorbar=False):
    arr = np.nan_to_num(data.astype(np.float32))
    if title in ("S1", "S2"):

```

```

    vmin, vmax = -1, 1
elif title == "Retardance":
    vmin_global = -2.8
    vmax_global = -1
    local_vmin, local_vmax = np.percentile(arr, [2, 98])
    vmin = min(local_vmin, vmin_global)
    vmax = max(local_vmax, vmax_global)
elif title == "Circ. Retardance":
    vmax = arr.max() * 1.1
    vmin = .5
else:
    vmin, vmax = arr.min(), arr.max()
norm = np.clip((arr - vmin) / (vmax - vmin + EPS), 0, 1)
img = (norm * 255).astype(np.uint8)
if cmap == 'gray':
    img = cv2.cvtColor(img[0], cv2.COLOR_GRAY2BGR)
    # exit()
else:
    img = cv2.applyColorMap(img, getattr(cv2, f'COLORMAP_{cmap.upper()}'))
cv2.imwrite(str(output_path), img)

if with_colorbar:
    fig, ax = plt.subplots(figsize=(4, 4))
    im = ax.imshow(arr, cmap=cmap, vmin=vmin, vmax=vmax)
    ax.set_title(title, fontproperties=FONT_PROPS)
    ax.axis('off')
    add_colorbar(im, ax)
    fig.savefig(str(output_path.with_name(output_path.stem + '_cb.png')
    )), dpi=100, bbox_inches='tight')
    plt.close(fig)

def save_images(results, index=None):
    titles = ["Original", "S1", "S2", "S3", "Retardance", "Circ. Retardance"]
    cmaps = ['gray', 'jet', 'jet', 'jet', 'jet', 'jet']
    for data, title, cmap in zip(results, titles, cmaps):
        fname = f"I_{index}_{title.replace(' ', '_')}.png" if INDIVIDUAL
            else f"Frame_{index:02d}_{title.replace(' ', '_')}.png"
        out = OUTPUT_DIR / fname if INDIVIDUAL else BASE_DIR / fname
        save_panel(data, title, cmap, out, with_colorbar=False)

```

```

# --- Processing Functions ---

def process_single(idx):
    roi = np.stack([load_gray(BASE_DIR / f"I_{angle}_{idx}.png") for angle
                     in ANGLES])
    intens = [median_filter(img.astype(float), size=6) for img in roi]
    M = helper.compute_and_save_stokes if CLASSICAL else helper.
        compute_stokes_from_rqwp
    S0, S1, S2, S3 = M(dict(zip(ANGLES, intens + intens[:2])), display=
        False, send_home=False)
    S3[0, 0], S3[-1, -1] = 1, -1
    ret = calculate_retardance(S1, S2)
    circ = calculate_ellipticity(S1, S2, S3)
    save_images([roi, S1, S2, S3, ret, circ], index=idx)

def divide_and_conquer():
    sample = load_gray(BASE_DIR / f"stitched_{ANGLES[0]}.png")
    H, W = sample.shape
    h_splits = int(np.ceil(H / THRESH_PIX))
    w_splits = int(np.ceil(W / THRESH_PIX))
    ret_full = np.zeros_like(sample, float)
    circ_full = ret_full.copy()
    S1_full = ret_full.copy(); S2_full = ret_full.copy(); S3_full =
        ret_full.copy()

    for i in range(h_splits):
        for j in range(w_splits):
            y0, y1 = i * H//h_splits, (i+1)*H//h_splits if i < h_splits-1
                else H
            x0, x1 = j * W//w_splits, (j+1)*W//w_splits if j < w_splits-1
                else W
            tile = np.stack([load_gray(BASE_DIR / f"stitched_{a}.png")[y0:
                y1, x0:x1] for a in ANGLES])
            intens = [median_filter(t.astype(float), size=6) for t in tile
                ]
            S0, S1, S2, S3 = (helper.compute_and_save_stokes if CLASSICAL
                else helper.compute_stokes_from_rqwp)(dict(zip(ANGLES,
                    intens + intens[:2])), display=False, send_home=False)
            S3[0,0], S3[-1,-1] = 1, -1
            ret_full[y0:y1, x0:x1] = calculate_retardance(S1, S2)

```

```

        circ_full[y0:y1, x0:x1] = calculate_ellipticity(S1, S2, S3)
        S1_full[y0:y1, x0:x1] = S1; S2_full[y0:y1, x0:x1] = S2;
        S3_full[y0:y1, x0:x1] = S3

    save_images([sample, S1_full, S2_full, S3_full, ret_full, circ_full],
                index=1)

# --- Stitching ---
def stitch(base, x_steps, y_steps, overlap, title, use_matrix=False,
           matrix=None):
    rows = []

    if SECTION and use_matrix and matrix is not None:
        h, w = matrix.shape    # ← rows, cols

        for r in range(h):
            bottom = h - 1 - r
            rev    = (bottom % 2 == 0)
            cols   = matrix[bottom, :-1] if rev else matrix[bottom]
            imgs   = []
            for i, idx in enumerate(cols):
                fname = base / f"I_{idx}_{title}.png"
                if not fname.exists():
                    img = np.zeros((512,640,3), dtype=np.uint8)
                else:
                    img = cv2.imread(str(fname))
                if i > 0:
                    img = img[:, overlap[1]:]    # crop horizontal overlap
                imgs.append(img)
            row_img = np.hstack(imgs)
            if r > 0:
                row_img = row_img[overlap[0]:, :] # crop vertical overlap
            rows.append(row_img)
        return np.vstack(rows)
    else:
        cols = x_steps.size
        rows_n = y_steps.size
        for r in range(rows_n):
            actual_r = rows_n - 1 - r
            rev = (actual_r % 2 == 1)
            imgs = []
            for c in range(cols):

```

```

        idx = actual_r * cols + (cols - 1 - c if rev else c) + 1
        fname = base / f"I_{idx}_{title.replace(' ', '_')}.png"
        img = cv2.imread(str(fname)) if fname.exists() else np.
            zeros((512, 640, 3), dtype=np.uint8)
        if c > 0:
            img = img[:, overlap[1]:]
            imgs.append(img)
        row_img = np.hstack(imgs)
        if r > 0:
            row_img = row_img[overlap[0]:, :]
            rows.append(row_img)
        return np.vstack(rows)

# --- Main Entry ---
def main(x_steps, y_steps):
    if SECTION:
        grid_rows, grid_cols, cell_width, cell_height, clicked_cells =
            get_params(sample_image_path)

        if SELECT_AREA:
            img = cv2.imread(sample_image_path)
            img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) # Convert BGR to
                RGB for matplotlib
            width = int(img.shape[1] * 5 / 100)
            height = int(img.shape[0] * 5 / 100)
            img = cv2.resize(img, (width, height), interpolation=cv2.
                INTER_AREA)
            idxs = run(img, grid_rows, grid_cols, cell_width, cell_height
                , clicked_cells) if SELECT_AREA else None
            h, w = map(int, (input("Cells Vertical: "), input("Cells
                Horizontal: ")))
            mat = np.array(idxs).reshape((h, w))
            np.save('frame_index_data.npy', mat)
        else:
            mat = np.load('frame_index_data.npy')
            print("Loaded indices:", mat)
            idxs = mat.flatten().tolist()
            x_steps, y_steps = mat.shape
            print(idxs)

    if INDIVIDUAL:
        pattern = os.path.join(BASE_DIR, "I_0.0_*.png")

```

```

if SECTION:
    total_images = len(idxs)
    for i in idxs: process_single(i)
else:
    total_images = len(glob.glob(pattern))
    idxs = np.arange(start=1, step=1, stop=total_images+1)
    for i in (idxs if len(idxs)>1 else range(1, total_images)):
        process_single(i)
if total_images == 0:
    print(f"No image found in the base folder {BASE_DIR}.")

for title in ("Original", "Retardance", "Circ. Retardance"):
    if SECTION:
        img = stitch(OUTPUT_DIR, x_steps, y_steps, OVERLAP_PX, title,
                      use_matrix=True, matrix=mat)
    else:
        img = stitch(OUTPUT_DIR, x_steps, y_steps, OVERLAP_PX, title,
                      use_matrix=False, matrix=None)
    out = OUTPUT_DIR / f"stitched_{title.replace(' ', '_')}.png"
    cv2.imwrite(str(out), img)

if __name__ == '__main__':
    main(x_steps, y_steps)

```

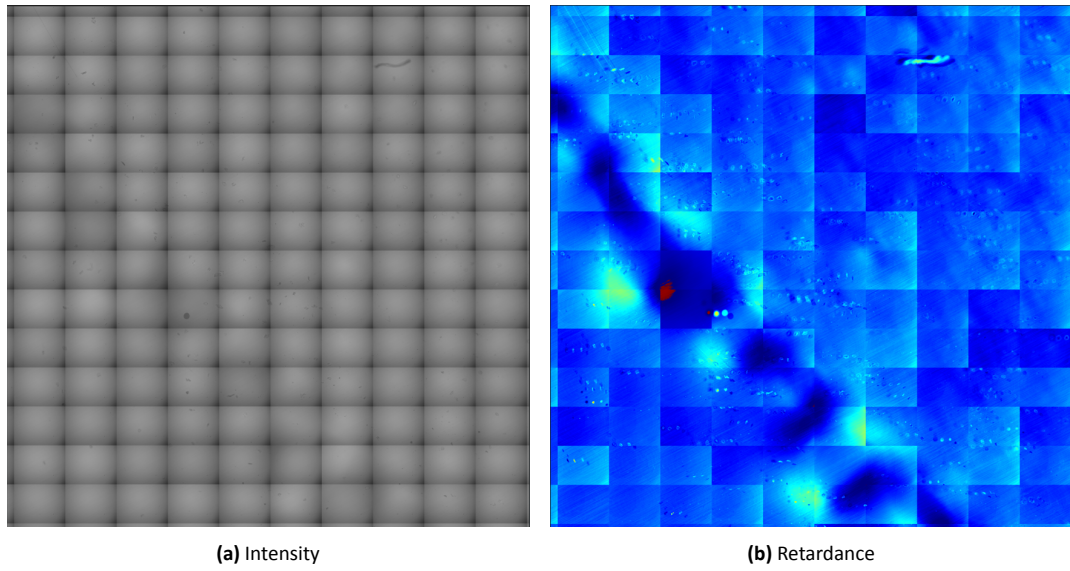


Figure 15. Intensity and retardance images for the edge of the wafer. The wafer shift on the stage causes ghost features on the retardance map.