

Photogrammetry and Image Fusion Techniques for High-Quality Colour 3D  
Reconstruction of MEMS Structures

by

Walid Hisham Abdelmalek

Submitted in partial fulfilment of the requirements  
for the degree of Master of Applied Science

at

Dalhousie University  
Halifax, Nova Scotia  
November 2023

Dalhousie University is located in Mi'kma'ki, the  
ancestral and unceded territory of the Mi'kmaq.  
We are all Treaty people.

© Copyright by Walid Hisham Abdelmalek, 2023

## Table of Contents

<b>List of Tables</b> .....	iv
<b>List of Figures</b> .....	v
<b>Abstract</b> .....	x
1. Introduction .....	1
1.1. Problem Definition .....	1
1.2. Literature Review .....	6
1.2.1. Registration .....	6
1.2.2. Focus Measure .....	7
1.2.3. Image Fusion .....	10
2. MEMS .....	13
2.1. Introduction .....	13
2.2. MUMPS .....	13
3. Microscopes .....	19
3.1. Microscope Fundamentals .....	19
3.2. Magnification Problem .....	23
3.3. Objectives & C-mounts .....	24
4. Experimental Setup: Table Microscope .....	25
4.1. Camera .....	25
4.2. Illumination .....	25
4.3. Tabletop microscope .....	27
4.4. Zaber XYZ Motion Stage .....	33
4.5. Microscope interference .....	34
4.6. Image Acquisition MATLAB .....	35
5. 3D Imaging Processing .....	38
5.1. Registration .....	38
5.2. Focus Algorithm .....	41
5.3. Depth Map .....	44
5.4. Image Fusion .....	51
5.5. Smoothing .....	55
6. Optimization .....	60
6.1. Different Focus Measure Algorithms .....	60

6.2. Guidelines for smoothing .....	70
6.3. Guidelines for the number of images .....	74
7. Gallery .....	76
8. Conclusion.....	96
References .....	98
Appendix A-1 Image Registration .....	101
Appendix A-2 Focus Measure .....	103
Appendix A-3 Smoothing and Fusion .....	106

## List of Tables

Table 1 Objective comparisons: fixed Beam Splitter, Doubler, and 160 mm Tube Length. .....	31
Table 2 Effect of tube length on WD & FOV for 25X objective. ....	32
Table 3 Vertical noise analysis for 3 focus measure algorithms. ....	67
Table 4 Measured heights of POLY1 objects. ....	82



## List of Figures

Figure 1 A MEMS microscopic image of a bearing with an electrostatic motor (chip = IMUDTRM1).....	2
Figure 2 2D MEMS image showing a rack, slidable and slightly tilted (chip = IMUDTRM1).....	3
Figure 3 A perspective view of a MEMS rack from the previous Figure showcasing the focus bar.....	4
Figure 4 Scanning Electron Microscope image of a MEMS actuator with all areas in focus (Image: C. Arthur, Dalhousie MEMS Lab, chip = IMUDTRM1).....	5
Figure 5 Brenner Focus Measure: Left, a sharply defined artificial image; Right, the resulting Brenner focus measure image.....	9
Figure 6 Brenner Focus Measure: Left, an artificial image with a blurred edge; Right, the resulting Brenner focus measure image.....	9
Figure 7 PolyMUMPs Layers.....	14
Figure 8 Simple MUMPs cantilever. Top: pre oxide removal. Bottom: after oxide removal.....	15
Figure 9 Simple MUMPs cantilever with dimples. Top: pre oxide removal. Bottom: after oxide removal.....	16
Figure 10 MUMPs POLY2 hasp (rounded corners not shown). Top: pre oxide removal. Bottom: after oxide removal.....	17
Figure 11 SEM photo of MUMPs hasp zoom in from Figure 3 earlier (Image: C. Arthur, Dalhousie MEMS Lab, chip = IMUDTRM1).....	18
Figure 12 Fixed-Finite tube length of a typical microscope.....	20
Figure 13: Thin Lens Equation.....	20
Figure 14: Circle of Confusion.....	22
Figure 15 Perspective view of a specimen showing focus bar.....	23
Figure 16 Left: photo of objective. Right: photo of 40 mm long C mount threaded tube.....	24
Figure 17 In-line illumination.....	26
Figure 18 Off-axis illumination.....	26
Figure 19 Top view schematic of 90-degree (yellow) and 45-degree (green) setups.....	28
Figure 20 Light paths of 90-degree (yellow) and 45-degree (green) setups.....	28

Figure 21 Top view and perspective photos of table microscope. The table holes are spaced 1” apart. ....	29
Figure 22 Sample 90-degree photo of bonding pads and a thermal actuator (chip = IMUDTVT2). ....	30
Figure 23 Sample 45-degree photo of bonding pads, note narrow focus bar. ....	30
Figure 24 Zaber XYZ stage, table holes are 1” apart. XYZ letters refer to camera axes. ....	33
Figure 25 25x microscope objectives 90-degrees and 45-degree interference. ....	34
Figure 26 Vertically offset microscopes as seen from above and from 45-degree LED. Table holes are 1” apart. ....	35
Figure 27 A 90-degree 2D Microscopic image of a MEMS actuator (chip = IMUDTRM1). ....	36
Figure 28 Sequence of 45-degree images of MEMS actuator, showing rising focus bars. ....	36
Figure 29 90-degree 2D image of a MEMS plate with perforations (chip = IMUDVT2). ....	37
Figure 30 Pair of 45-degree images of perforated MEMS plate showcasing focus bars at different locations. ....	37
Figure 31 Importance of Image registration: Left: fused unregistered images, Right: fused registered images (chip = IMUDTVT2). ....	40
Figure 32 Zoom-in of the Figure above: Left is unregistered, Right is registered ....	40
Figure 33 2D image of MEMS grid actuator (chip = IMUDTPH1). ....	41
Figure 34 Left: 1 sample 45-degree image, Right: sample stack of 9 45-degree images. Z axis is the image number. XY axes are % of the image. ....	42
Figure 35 Left: 1 sample 45-degree perspective view RGB image. Right: resultant focus measure image, highlighting in-focus areas (greyscale false colour, yellow is in focus). ....	43
Figure 36 Left: 45-degree image stack, Right: focus stack (false colour). Z axis is the image number. XY axes are % of the image. ....	44
Figure 37 A 2D image of MEMS bearing with an electrostatic motor along with a gear rack (chip = IMUDTRM1). ....	45
Figure 38 Finding Focus peaks. Top: sample red pixel on grayscale focus image. Bottom left: drilling through focus stack (false colour). Bottom right: focus curve for sample pixel. ....	46

Figure 39 Focus peak map: maximum focus across all images. False colour: yellow is a high focus peak, and dark blue is low.....	47
Figure 40 Top: background sample red pixel on grayscale focus image. Bottom left: drilling through focus stack (false colour). Bottom right: peak-less focus curve for sample pixel. ....	48
Figure 41 2D image of MEMS chip grid actuator (chip = IMUDTPH1). ....	50
Figure 42 False colour depth map of the device from the previous Figure showing progressive colour bars, as well as multicoloured areas / 'blobs'.....	51
Figure 43 Result of image fusing, note poor image quality and artifacts in background areas. The red box is shown magnified in the next Figure. ....	53
Figure 44 Zoomed-in view from the previous Figure. Top: Fused Image, Bottom: Depth map. Note that multicolour patches in the depth map lead to artifacts in the fused image. ....	54
Figure 45 A 2D image of a MEMS chip spur gears (chip = IMUDTRM1). ....	55
Figure 46 Top: Focus peaks in comparison to Bottom: focus SNR. ....	56
Figure 47 Depth maps (left images) at different smoothing levels (numbers at far left) with corresponding fused images (right images). ....	58
Figure 48 Zoomed-in top left corner of previous images at different smoothing levels. .	59
Figure 49 BREN focusing algorithm: Top = staircase graph and Bottom = image with 100 smoothing.....	61
Figure 50 GDER focusing algorithm: Top = staircase graph and Bottom = image with 100 smoothing.....	62
Figure 51 GLVM focusing algorithm: Top = staircase graph and Bottom = image with 100 smoothing.....	63
Figure 52 GRA3 focusing algorithm: Top = staircase graph and Bottom = image with 100 smoothing.....	64
Figure 53 GRAT focusing algorithm: Top = staircase graph and Bottom = image with 100 smoothing.....	65
Figure 54 HELM focusing algorithm: Top = staircase graph and Bottom = image with 100 smoothing.....	66

Figure 55 Comparisons of wire to MEMS transition between different focus measures at 100 smoothing.....	68
Figure 56 Comparisons of background detail between different focus measures at 100 smoothing.....	69
Figure 57 2D image of a MEMS electrostatic actuator (chip = IMUDTRM1). ....	71
Figure 58 Zoom-in of the previous Figure. Effect of excess smoothing. ....	72
Figure 59 Excessive smoothing: Top: depth map & 3D image. Bottom: zoom-in on the lower left corner. (Smoothing = 1000). ....	73
Figure 60 Effect of the number of captured images on 3D image quality.....	75
Figure 61 2D overhead image of MEMS Grid Actuator (chip = IMUDTPH1). ....	76
Figure 62 3D image of MEMS Grid Actuator from the previous Figure. ....	77
Figure 63 2D overhead image of MEMS Thermal Actuators (chip = IMUDTPH1).....	78
Figure 64 3D image of MEMS Thermal Actuators from the previous Figure. ....	79
Figure 65 2D image of MEMS irregular gold-coated plate on modified Grid Actuator (chip = IMUDTCV4). ....	80
Figure 66 3D image of MEMS irregular gold-coated plate from the previous Figure. ....	81
Figure 67 Calculating the heights of POLY1 features.....	82
Figure 68 2D image of MEMS electrostatic motor (chip = IMUDTRMC).....	83
Figure 69 3D image of MEMS device shown in the previous Figure. ....	84
Figure 70 Closer 2D image of another MEMS electrostatic motor (chip = IMUDTRMC). ....	85
Figure 71 3D image of MEMS device shown in the previous Figure. ....	86
Figure 72 2D image of MEMS showing 2 wires and 3 bonding pads (chip = IMUDTEW1).....	87
Figure 73 3D image of MEMS wires shown in the previous Figure. ....	88
Figure 74 2D image showing 1 bonding pad with wire and 1 pad with unwired soldered ball (chip = IMUDTVT2). ....	89
Figure 75 3D image of MEMS wire and solder ball shown in the previous Figure. ....	90
Figure 76 2D image showing 3 pads with unwired soldered balls (chip = IMUDTVT2). ....	91
Figure 77 3D image of MEMS solder balls shown in the previous Figure. ....	92
Figure 78 2D overhead image of MEMS Thermal Actuators (chip = IMUDTCV4). ....	93

Figure 79 3D image of MEMS Thermal Actuators from the previous Figure. .... 94  
Figure 80 Comparison of SEM image from Section 2 with 3D image..... 95

## **Abstract**

Obtaining a three-dimensional image of a MEMS (Micro Electro Mechanical Systems) chip can be challenging due to its small size, necessitating work at a micrometer scale. In this work, photogrammetry will be used to capture three-dimensional images of MEMS. To construct these 3D images, multiple images are taken from a perspective view to include the height and three-dimensional details. When capturing images from a perspective view through a microscope, only a portion of the image will be in focus. To produce an all-in-focus 3D image, three imaging techniques will be employed. Image registration will be necessary to ensure alignment since vibration can cause pixels to shift in each frame. Subsequently, focus measurement techniques will be utilized to determine which parts of the image are in focus and which are not. This information will be used to generate a depth map indicating which image has the best focus. A final fused image in which all pixels are in focus will be formed from the depth map, showcasing all the intricate details of the MEMS chip. In this study, we will examine the effects of smoothing, focus measure operators, and the number of images required to make an adequate image.

## LIST OF ABBREVIATIONS USED

1D	1-dimensional
2D	2-dimensional
3D	3-dimensional
BREN	Brenner's
CoC	Circle of Confusion
DOF	Depth of Field
$f$	Focal Length
FOV	Field of View
GDER	Gaussian Derivative
GLVM	Modified Gray Level Variance
GRA3	3D Laplacian
GRAT	Threshold Gradient
HELM	Helmi's Mean Method
$i$	Image Distance
LED	Light Emitting Diode
LWD	Long Working Distance
MEMS	Micro Electro Mechanical Systems
MUMPs	Multi-User MEMS Process
$o$	Object Distance
OXIDE	Silicon Dioxide
POLY	Polysilicon
PSG	PhosphoSilicate Glass
SEM	Scanning Electron Microscopy
SNR	Signal to Noise Ratio
WD	Working Distance

## 1. Introduction

### 1.1. Problem Definition

Micro electro mechanical systems (MEMS) are miniaturized devices that serve a wide variety of applications. These MEMS devices contain components (electronic and moving parts) ranging between 1 to 100 micrometers. MEMS chips are distinct from typical electrical microchips due to the inclusion of mechanical components such as moving parts and actuators. MEMS devices are fabricated using a micromachining technique called surface etching. This process allows for the precise fabrication of intricate structures.

When comparing macro-objects to MEMS, imaging MEMS is much harder. Macro objects are large and do not require significant magnification during imaging. MEMS are distinct from macro-objects in that it is difficult to obtain three-dimensional information as they are imaged under high magnification. When looking at MEMS chips they seem flat and feature poor. Unlike purely electrical microchips the ability to image MEMS components in 3D would provide valuable insights into the structure and functionality of MEMS devices and assist in validating the designs of MEMS chips. By using three-dimensional imaging techniques, we aim to better identify the 3D features on the microchip.

However, imaging a MEMS chip is more challenging due to the impact of magnification on the level of detail that can be observed. Additionally, MEMS imaging faces the additional difficulty of a shallow depth of field (described later), further complicating the imaging process when compared to macro-objects.

There are a variety of existing ways that imaging a microelectromechanical system can be done and this includes conventional optical microscopy, scanning electron microscopy (SEM), laser scanning and three-dimensional optical profiling. These methods have various limitations that can be examined.

Conventional optical microscopy is the most common method used for imaging MEMS chips, but it has certain drawbacks. Conventional optical microscopy is typically limited to viewing the specimen positioned perpendicular to the microscope, providing an



overhead 2D view of the specimen. For example, Figure 1 shows a portion of a MEMS microchip that shows a bearing with an electrostatic motor (chip = IMUDTRM1). This 2D view hinders the ability to differentiate between raised and non-raised areas on the chip. Additionally, conventional optical microscopy is limited by its shallow depth of field, this means that objects near or far away from the objective become out of focus, which restricts its imaging capabilities. Thus, the taller part of MEMS which can be a few microns in height can be slightly out of focus.

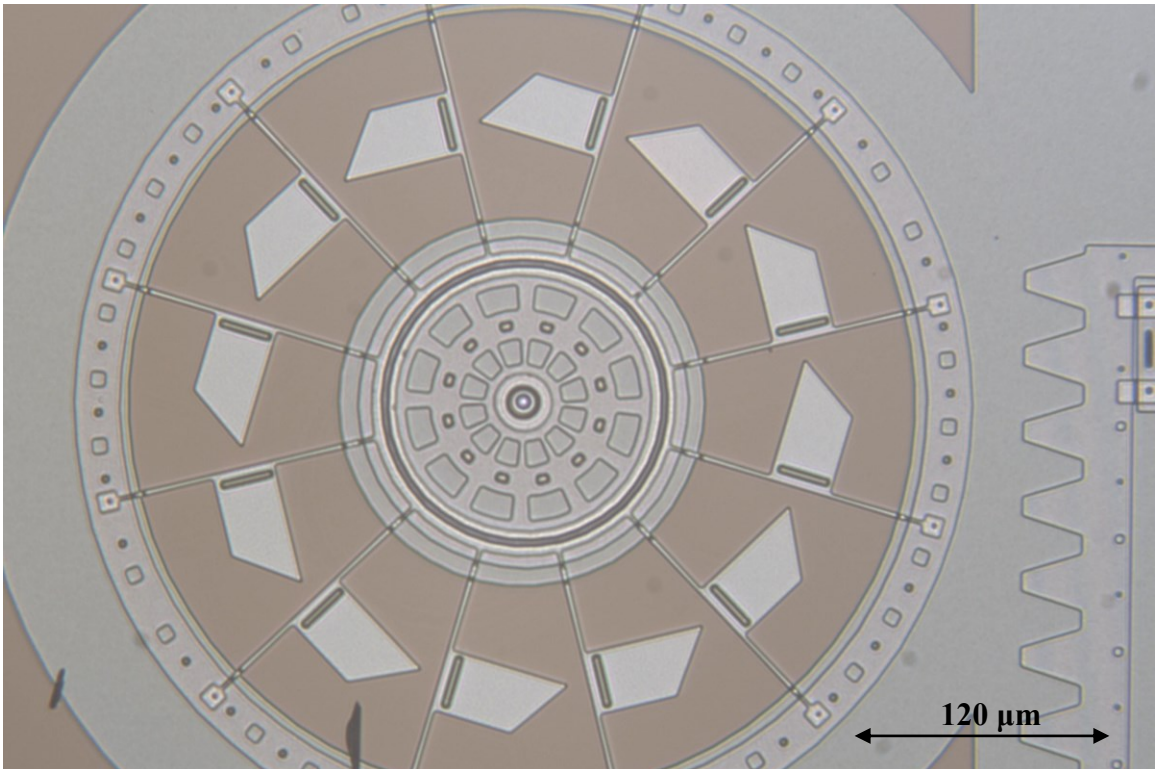


Figure 1 A MEMS microscopic image of a bearing with an electrostatic motor (chip = IMUDTRM1).

When examining a MEMS chip from a conventional view, the microstructures on the chip are not easily discernible. However, by switching to a perspective view, we gain the ability to identify the raised and non-raised parts of the chip. Conceptually, positioning the microscope at a 45-degree angle to the specimen would produce a 3D perspective view. However, viewing a specimen from a perspective angle using an optical microscope introduces large focus problems. For example, Figure 2 shows a 2D image of a slidable MEMS rack. Figure 3 shows the same device imaged at 45 degrees,

note only a narrow bar of the image is in focus. This is because when observing the specimen at an angle the far and near parts can be 10 to 100's of microns out of focus. Only a narrow portion of the image will be in focus at any given time, which provides us with a limited depth of field.

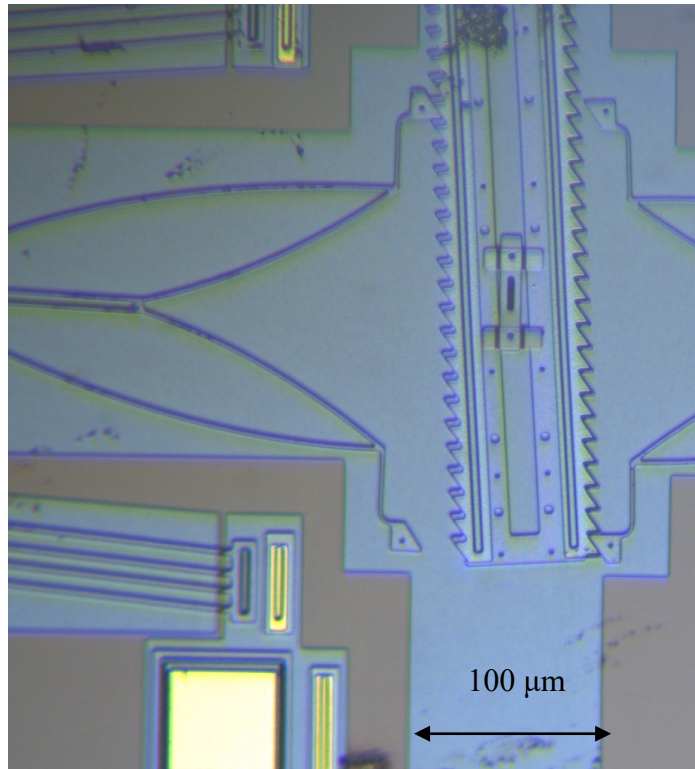


Figure 2 2D MEMS image showing a rack, slidable and slightly tilted (chip = IMUDTRM1).

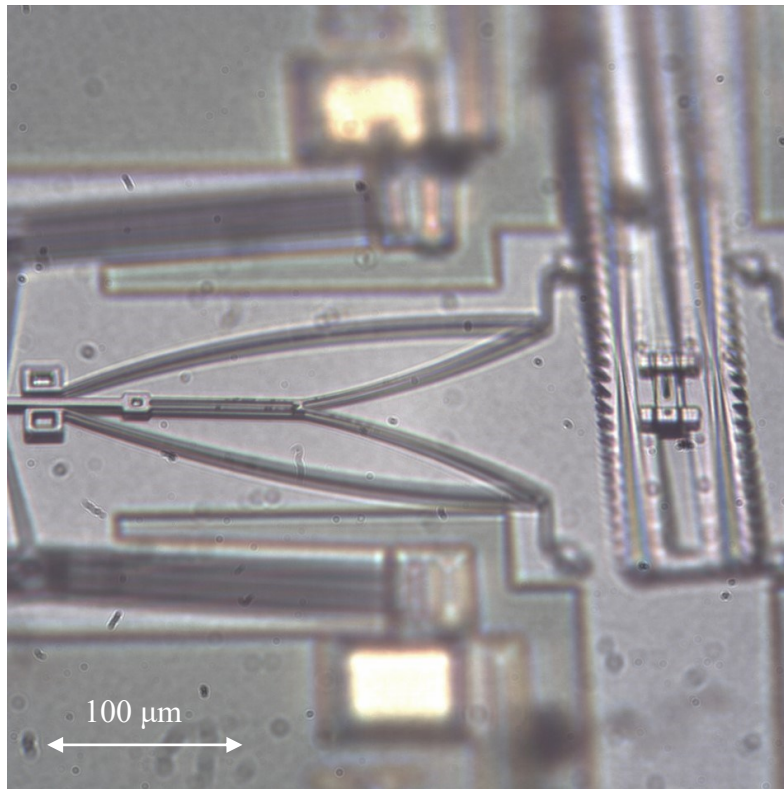


Figure 3 A perspective view of a MEMS rack from the previous Figure showcasing the focus bar.

Scanning Electron Microscopy (SEM) is a method of obtaining three-dimensional images of microchips. SEMs utilize a beam of electrons that travel toward the MEMS chip and provide a higher resolution and larger depth of field than other methods. The resulting image produced by the Scanning Electron Microscope is grayscale. SEM can have both overhead and perspective views and SEM images often look very 3-dimensional. For example, Figure 4 shows a MEMS thermally actuated pusher. However, SEMs are very expensive, and samples must be imaged within a vacuum, thus SEM are not commonly found in most MEMS labs.

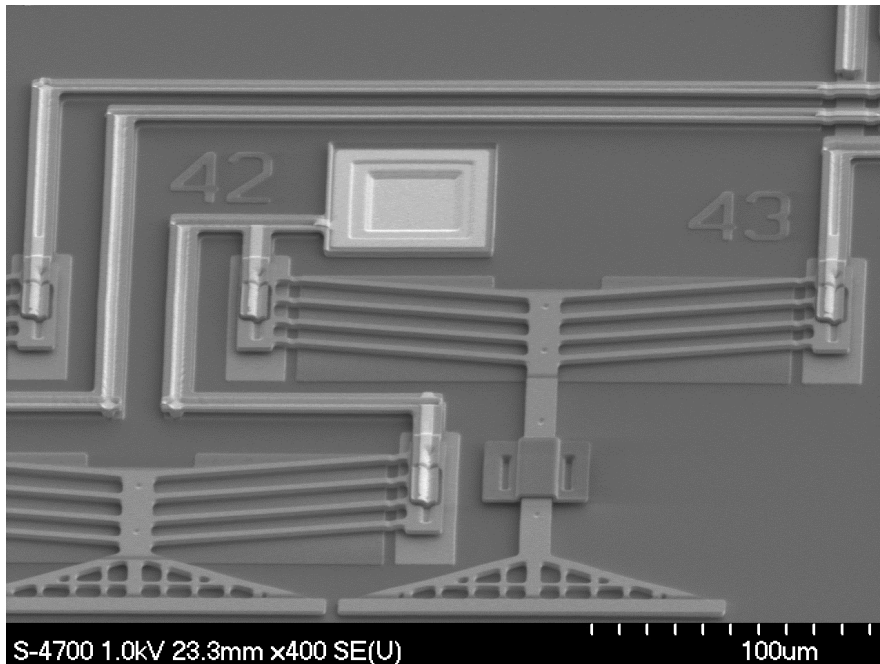


Figure 4 Scanning Electron Microscope image of a MEMS actuator with all areas in focus (Image: C. Arthur, Dalhousie MEMS Lab, chip = IMUDTRM1).

An alternative method that might potentially be employed to acquire three-dimensional information is the utilization of three-dimensional laser scanners. The scanners operate by employing a laser probe that sweeps across the MEMS chip. However, these scanners come with unavoidable drawbacks when applied to MEMS chips. Typically, their accuracy is limited to the millimeter range rather than the micrometer range. Moreover, scanning MEMS chips presents additional challenges due to their reflective nature, which further complicates the scanning process.

A promising method is photogrammetry. Photogrammetry is a low-cost method to obtain macro-scale three-dimensional images of an object without needing any physical contact with the object. Photogrammetry requires that many high-quality images of the object be taken from different viewpoints, software is used to combine the high-quality images into a three-dimensional image.

The goal of this study is to apply photogrammetry techniques to construct a colour 3D MEMS image from multiple 2D images taken with a conventional optical microscope. One method of reconstructing an image is by using image fusion. Focus

stacking determines which parts of multiple images are in focus, once this is known image fusion can be used to combine the in-focus components to create an image with a three-dimensional perspective.

## 1.2.Literature Review

Image Fusion [1] is a method of creating a combined or fused image from multiple different images. This can be done in a variety of ways. Image fusion methods are often used in creating images that have an extended depth of field: partially focused images can be combined into one image where all areas are in focus. To create a fused image, a stack or sequence of original images will need to be taken with a change in focus between the images.

There are 3 steps to image fusion: Registration, Focus Measure & Fusion.

### 1.2.1. Registration

Image registration is a required process in generating a perspective view image when dealing with multiple images. It is not uncommon for consecutive frames to exhibit micromovements, which can result from various factors such as the resolution of the stage moving the microchip or slight shifts in camera position. These movements, although small, have the potential to introduce undesired artifacts during the image fusion process. To address this, an effective image registration algorithm is necessary to align the multiple images to a common reference frame. By employing robust image registration techniques, the final perspective view image can be generated with improved quality and reduced artifacts.

Yatsenko [2] developed a MATLAB function for image registration that measures the shift using cross-correlation in the frequency domain. This function can provide sub-pixel precision and returns the shift values. Cross-correlation can be employed to achieve accurate alignment and minimize distortions caused by shifts between frames. We utilized this method to enhance our image registration process (see later section 5.1), but there are other registration techniques.

For example, Reddy [3] proposes four primary methods of image registration: correlation method, Fast Fourier transform method, feature-based method, and graph

theoretic method. As another example, Zhang [4] suggested the use of affine transformation as a method of registration. Affine transformation is a geometric technique that maps points from one coordinate system to another to register the images within the stack and align them in the same scene. Zitova [5] provides a survey of different image registration techniques.

### 1.2.2. Focus Measure

When obtaining images from a perspective view, it is imperative to consider whether this view has a shallow depth of field. Only a limited portion of the image may be in focus and sharp, and focus measurement becomes a requirement when performing image fusion. Focus measure is the quantitative measure of the sharpness or focus in different areas of an image to determine which areas are in focus and which areas are out of focus. There are a range of algorithms that employ different methods of focus measure with varying results. Pertuz [6] provides a good summary of the different techniques. Different techniques that have been explored include: Gradient-based, Laplacian-based, Wavelet-based, Statistics-based, and CT-based methods [6]. The section below describes some of these methods.

One of the earliest methods (1976), is Brenner's focus measure [7]. Brenner's focus measure is computationally efficient compared to other possible focus measure algorithms. It relies on the use of the differential of the grayscale image to evaluate the level of focus (see Eq. 1).

$$\phi = \sum_{(i,j)} |I(i,j) - I(i+2,j)|^2 \quad \text{Eq. 1}$$

To calculate Brenner's focus measure, at each pixel in an image,  $I$ , the algorithm compares the current pixel to adjacent horizontal pixels (typically  $i$  &  $i+2$  are compared). Brenner's is in effect calculating the x gradient but can also be modified to calculate the y gradient or total gradient. A higher Brenner's value indicates greater sharpness in the pixel.

The focus measure value is obtained by summing the squared differences in Eq. 1 over a neighborhood of pixels: e.g., 5x5 or 10x10 pixels. The user may select the size of the neighbourhood based on the image. The main contributors to the focus measure values are edges and features. A smaller neighborhood is advantageous when an image contains many features, as it allows for a finer resolution. A larger neighborhood is beneficial when an image contains a relatively feature-poor background, as it helps ensure all neighbourhoods include some features. A medium-sized neighborhood (ranges from 7-14 pixels) is often used to produce accurate focus numbers within the image.

Figure 5 Left displays an artificial image sample of 7x7 pixels. In this example, the image exhibits a sharp edge. Figure 5 Right demonstrates the calculated Brenner focus measure (3x3 neighbourhood). When Brenner is calculated, the edges produce large focus measures. The remaining portions of the sharp image go to zero due to the absence of any changes.

Figure 6 Left depicts the sample image after artificial blurring, accompanied by the resulting focus measure in Figure 6 Right. With the introduction of blurring, the sharp edge is no longer visible. This alteration leads to a broader and lower valued distribution of focus measure values (see Figure 6 Right).

Sharp edges play a vital role in focus measurement, as they yield higher focus values, signifying areas that are more in focus. Conversely, areas that are blurred or edge-poor result in lower focus values, indicating potential out-of-focus regions. Thus distinct sharp edges are indispensable for accurately reconstructing the final image.



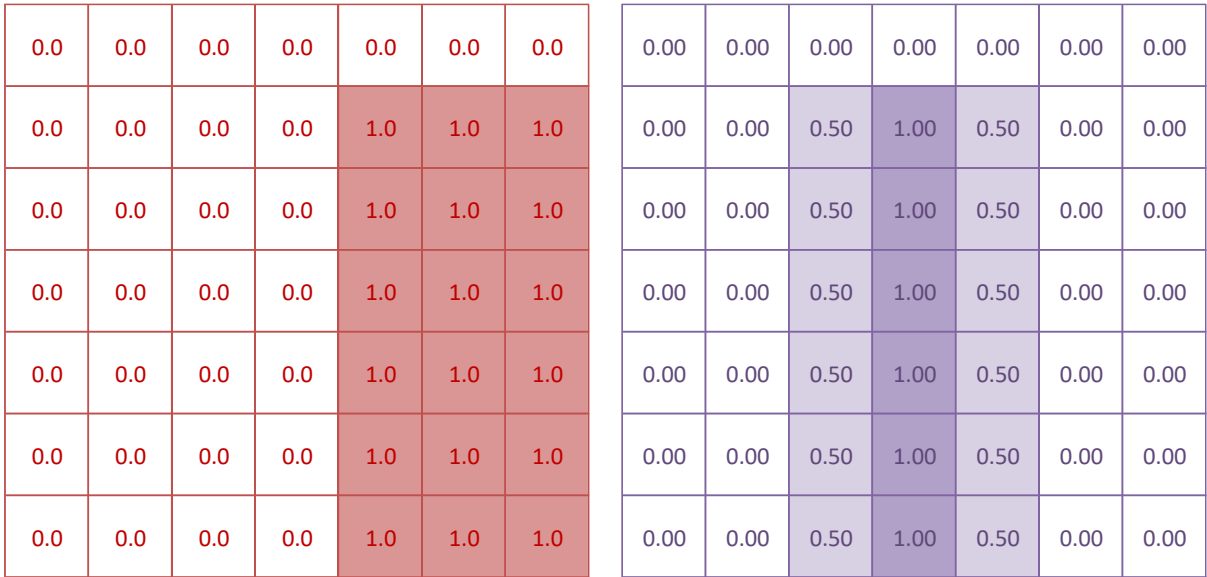


Figure 5 Brenner Focus Measure: Left, a sharply defined artificial image; Right, the resulting Brenner focus measure image.

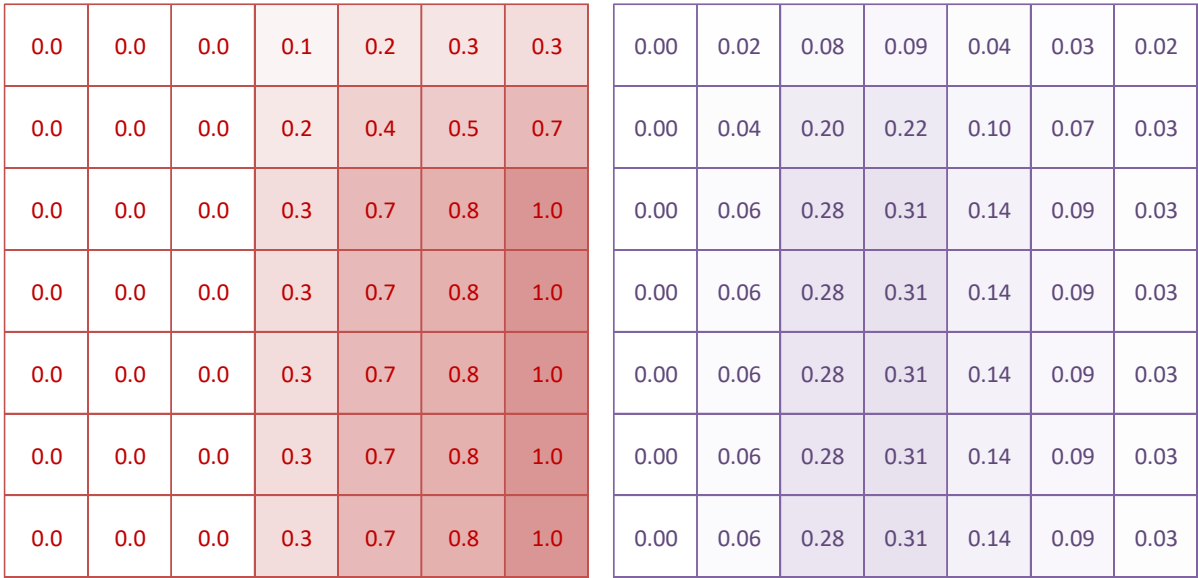


Figure 6 Brenner Focus Measure: Left, an artificial image with a blurred edge; Right, the resulting Brenner focus measure image.



### 1.2.3. Image Fusion

Image fusion is the process of combining multiple images to create a single all-in-focus image. There are various methods for creating the final image, including techniques like combining pixels from different images using a weighted sum or selecting specific pixels from different images and merging them into a final composite image. There were three broad historical uses for image fusion: 1) Cell imaging and cytology, 2) Computer vision and depth maps and 3) Macro photography.

#### 1) Cell imaging and cytology:

Cell imaging in 3D can help identify bacteria and microorganisms. However, at the highest magnifications, only a narrow Z slice of the cell is in proper focus. There are various ways to obtain a full-depth in-focus image. In fact, the earlier mentioned 1976 Brenner algorithm [7] was used in cytology. To identify the in-focus components, a stack of images was captured with changing focus, and then a focus measure was applied to determine which part of which image was in focus.

Shah et al. [8] and Osibote [9] both concentrated on tuberculosis detection through the application of focus stacking and focus measure techniques to identify the best-focused image. They introduced various methods and arrived at conclusions regarding the performance of different focus measure functions in detecting the optimally focused image within the stack. Osibote [9] utilized a linear actuator to control the microscope's focus level, capturing and storing images at fixed intervals along one axis. Various focus measure algorithms were employed to determine which parts of the image should be fused. Shah [8] used the best average focus approach to create an all-in-focus image for aiding tuberculosis diagnosis.

#### 2) Computer vision and depth maps:

Computer vision and depth maps play a crucial role in determining object dimensions and converting 2D images to 3D volumes. A depth map is a reconstruction of not only the object's X and Y position but also its Z position as inferred from focus measurements. In this case, the objects in the image are macro-sized, not micro, but may

be at substantially different depths and thus at different foci. Clear depth perception allows for discerning fine details and reducing ambiguity in object shape and position.

Pertuz 2013 [6] provides a good summary of the different focus measure techniques to determine Shape-from-focus (SFF). Pertuz 2022 [10] aimed to enhance the limited depth of field provided by typical cameras. They tested and compared various fusion techniques for creating an all-in-focus image. To achieve this, they explored various focus measurement and image fusion algorithms to determine the most effective methods. Pertuz also developed focus algorithms to identify the best-focused areas and used Shape-from-focus to reconstruct all-in-focus images.

Fedorov [11] aimed to construct a multi-focus image from multiple images. The images taken all had different focus settings, so when the final image was created, it was in focus throughout. In their program, all the images were pre-registered, and focus estimation was performed before the construction of a multi-focus image. They created the image by utilizing the focus values to create tiles and select the best tiles from the images, which were then stitched together.

Choi [12] focused on enhancing image fusion efficiency to generate an all-in-focus image. They achieved this by employing a strategy of working with smaller sets of images to reduce the required time for image capture and processing. Their approach entails scanning the image at a lower resolution and generating a depth map using shape-from-focus techniques. Subsequently, the depth map was utilized to select the optimal image sets by adjusting the lens position.

### 3) Macro photography:

In macro photography the camera is placed very close to a small, typically millimeter scale, object. This produces a highly magnified but limited depth of field image. In macro photography, achieving focus across all parts of the image can be challenging. To address this challenge, photographers must capture multiple images and employ focus stacking techniques to attempt to achieve comprehensive focus. There exist several commercially available software packages for macro photography: e.g., Helicon Focus [13] and CombineZM [14].

Olkowicz [13] utilized image fusion software for imaging rock roughness; Helicon Focus software was used for focus stacking of the images. Clark [15] used a camera microscope combination and Helicon Focus software to image electronic components. As another example, Bercovici [14] used the CombineZM software to measure focus and combine to create an image that has all areas in focus.

So far, all the methods we have explored have not been specifically applied to the objects we intend to examine: MEMS. Very few studies have used image fusion study for MEMS, the closest known study was by Xie [16]. Xie [16] investigated the 3D surface reconstruction of microparts including 1 large MEMS gear. The images they captured were taken from directly above the object. Through image fusion, they combined 2D pixels and depth data to generate an all-in-focus image. Finally, Xie employed a height map/depth map to reconstruct a 3D representation of the micropart [16]. However, Xie [16] examined very large MEMS objects: the object they investigated measured 300 to 400  $\mu\text{m}$  in height. They also employed a coarse Z resolution of 5  $\mu\text{m}$ . Our goal is to work at an even smaller micro-scale MEMS objects. Furthermore, Xie utilized a vertical view without considering perspective views. Finally, Xie conducted tests using grayscale images instead of coloured images [16].

The goal of this thesis is to generate a colour perspective view image that depicts the 3D features of micron-scale MEMS devices. By combining various methods from the above discussed papers, such as registration (Yatsenko), focus measurement, and image fusion (Pertuz), we aim to select the best-focused areas to produce an all-in-focus image of MEMS devices.

## 2. MEMS

### 2.1.Introduction

MEMS devices have many uses such as: accelerometers, pressure sensors, and projection systems. The most common MEMS devices are typically designed to operate as accelerometers. The MEMS devices measure the acceleration forces which are converted into electrical signals. Accelerometers are utilized in various devices such as automobiles, cell phones, and many others.

In a comparison between MEMS actuators and their macro counterparts, Bell [17] discovered that MEMS actuators outperformed the larger macro-sized ones. Specifically, MEMS actuators exhibited superior resolution and a higher maximum frequency, highlighting their enhanced capabilities in these aspects. This is why MEMS devices are commonly utilized as accelerometers and pressure sensors. Their superior accuracy and small form factor make them well-suited for these applications [17].

### 2.2.MUMPS

Multi-User MEMS Process (MUMPs) is a cost-effective method used for prototyping MEMS chips. MUMPs allows multiple users to share a silicon wafer thus reducing the cost and the increasing accessibility to MEMS technology. PolyMUMPs is a variant of MUMPs this method utilizes a three-layer polysilicon surface micromachining method [18].

The PolyMUMPs process begins with a 525 um thick silicon wafer (see Figure 7). PolyMUMPS includes several thin layers of different materials. Starting from the bottom the layers are: NITRIDE, POLY0, OXIDE1, POLY1, OXIDE2, POLY2, and METAL. A 0.6 um layer of silicon NITRIDE layer is deposited onto the silicon wafer, this acts as an insulating layer to prevent unwanted electrical interactions. This helps ensure that the MEMS chip has proper functionality and the needed reliability [18].

METAL	0.5 $\mu\text{m}$
POLY2	1.5 $\mu\text{m}$
OXIDE2	0.75 $\mu\text{m}$
POLY1	2 $\mu\text{m}$
OXIDE1	2 $\mu\text{m}$
POLY0	0.5 $\mu\text{m}$
NITRIDE	0.6 $\mu\text{m}$
SILICON WAFER	

Figure 7 PolyMUMPs Layers.

After the deposition of the silicon NITRIDE layer, a fixed 0.5  $\mu\text{m}$  layer of polysilicon is uniformly deposited onto the wafer. This layer is referred to as POLY0. Photolithography, a chemical etching process, is employed to pattern this layer. Photolithography starts with the application of a photoresist layer (a few  $\mu\text{m}$ ) onto the wafer. Photoresist is a light-sensitive material that weakens upon exposure to light. A mask containing the desired pattern is placed over the photoresist and then exposed to UV light to remove the resist regions that will be etched out from the POLY0 layer. This step defines the areas where the microstructures will be formed. Subsequently, the POLY0 layer is etched using a reactive ion etch (RIE) system, which removes only

exposed regions of POLY0. The photoresist is then removed with an etchant. Note: in all the following layers each layer is patterned similarly using photolithography [18].

Following this, a 2  $\mu\text{m}$  phosphosilicate glass (PSG) layer is deposited. This layer will serve as a sacrificial layer (similar to a support layer in 3D printing) and is known as the first oxide OXIDE1. The PSG layer will be removed via etching at the end of the PolyMUMPs process to free structural layers. OXIDE1 can be photolithographically patterned in several ways described below.

The next polysilicon structural layer is then deposited, becoming the first movable structural layer referred to as POLY1. This layer has a thickness of 2  $\mu\text{m}$ . This layer is patterned as desired using photolithography.

Whether the POLY1 is movable depends on how the OXIDE1 was etched. Where no pattern was made in OXIDE1, it is present and the POLY1 deposited on the OXIDE1 is free to move once the OXIDE1 is removed. However, there is a patterning layer called ANCHOR1 where the OXIDE1 is etched away. This removes the support layer and when the POLY1 is deposited, it directly connects POLY1 to POLY0. Thus, OXIDE1 controls the vertical anchoring of POLY1.

Figure 8 shows a schematic of a MUMPs cantilever in POLY1.

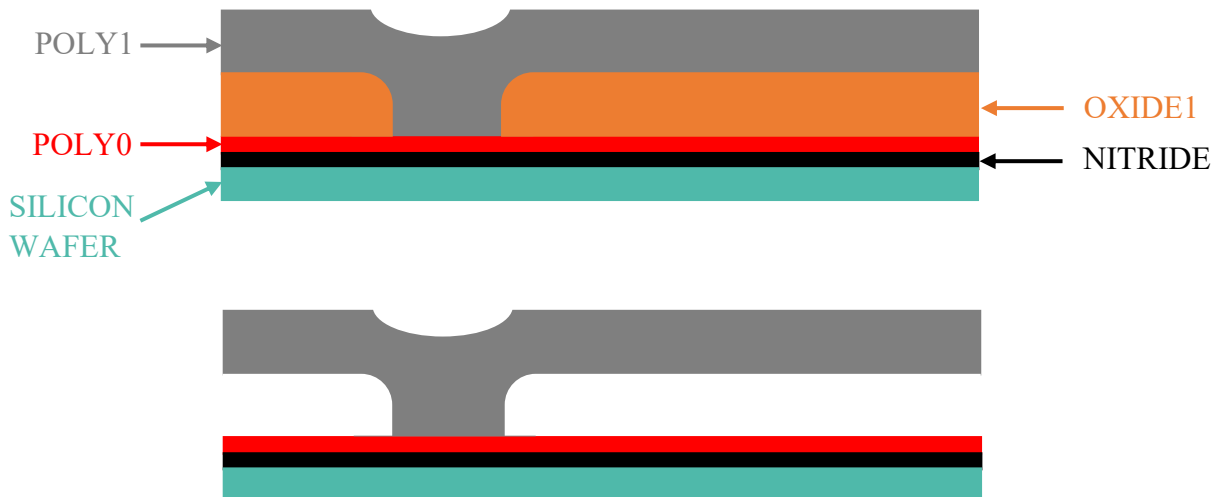


Figure 8 Simple MUMPs cantilever. Top: pre oxide removal. Bottom: after oxide removal.

There is a further option for OXIDE1. Long cantilevered objects can sag and if they contact with the underlying layers can adhere via surface forces known as stiction, especially if there is any water present. To counteract this, it is possible to only ‘half etch’ the OXIDE1 using the DIMPLE layer. Dimples are lithographically patterned and RIE etched holes, 0.75  $\mu\text{m}$  deep, in the OXIDE1 layer. This produces bumps in the POLY1 layer that reduce contact area and help reduce the influence of adhesion forces experienced at the microscale level [18] (see Figure 9).

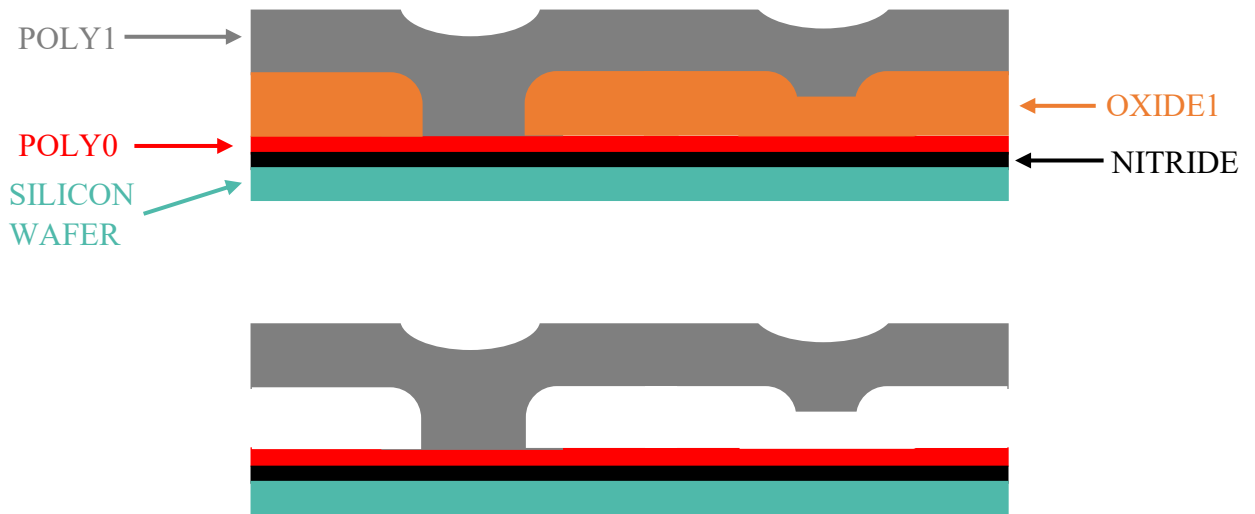


Figure 9 Simple MUMPs cantilever with dimples. Top: pre oxide removal. Bottom: after oxide removal.

After the POLY1 layer, a second sacrificial layer of PSG (0.75  $\mu\text{m}$  thick) is then deposited, known as the second oxide layer OXIDE2, which will be removed via etching during the final stages of production. The OXIDE2 can be photolithographically patterned in several ways described below.

The final polysilicon structural layer, known as POLY2, is deposited with a thickness of 1.5  $\mu\text{m}$ . This layer is patterned as desired using photolithography. Whether the POLY2 is movable depends on how the OXIDE2 was etched. Where no pattern was made in OXIDE1, it is present and the POLY2 deposited on the OXIDE2 is free to move

once the OXIDE2 is removed. However, there are 2 patterning layers called POLY1-POLY2-VIA and ANCHOR2 where the OXIDE2 is etched away. Where POLY1-POLY2-VIA exists, it removes the OXIDE2 support layer and when the POLY2 is deposited, it directly connects POLY2 to POLY1. Where ANCHOR2 exists, it removes OXIDE1 and OXIDE2 support layers and when the ANCHOR2 is deposited, it directly connects POLY2 to POLY0. Thus, OXIDE2 controls the vertical anchoring of POLY2.

Figure 10 shows a schematic of a MEMS hasp in POLY2, compared to the SEM photo in Figure 11.

The final layer in the PolyMUMPs process is a 0.5  $\mu\text{m}$  metal layer called METAL, which serves as electrical routing. Subsequently, the chips are diced, and they undergo a process of immersion in a solution to release the oxide layers. To minimize stiction, which refers to friction within the MEMS chips, they are immersed in DI (de-ionized) water followed by alcohol.

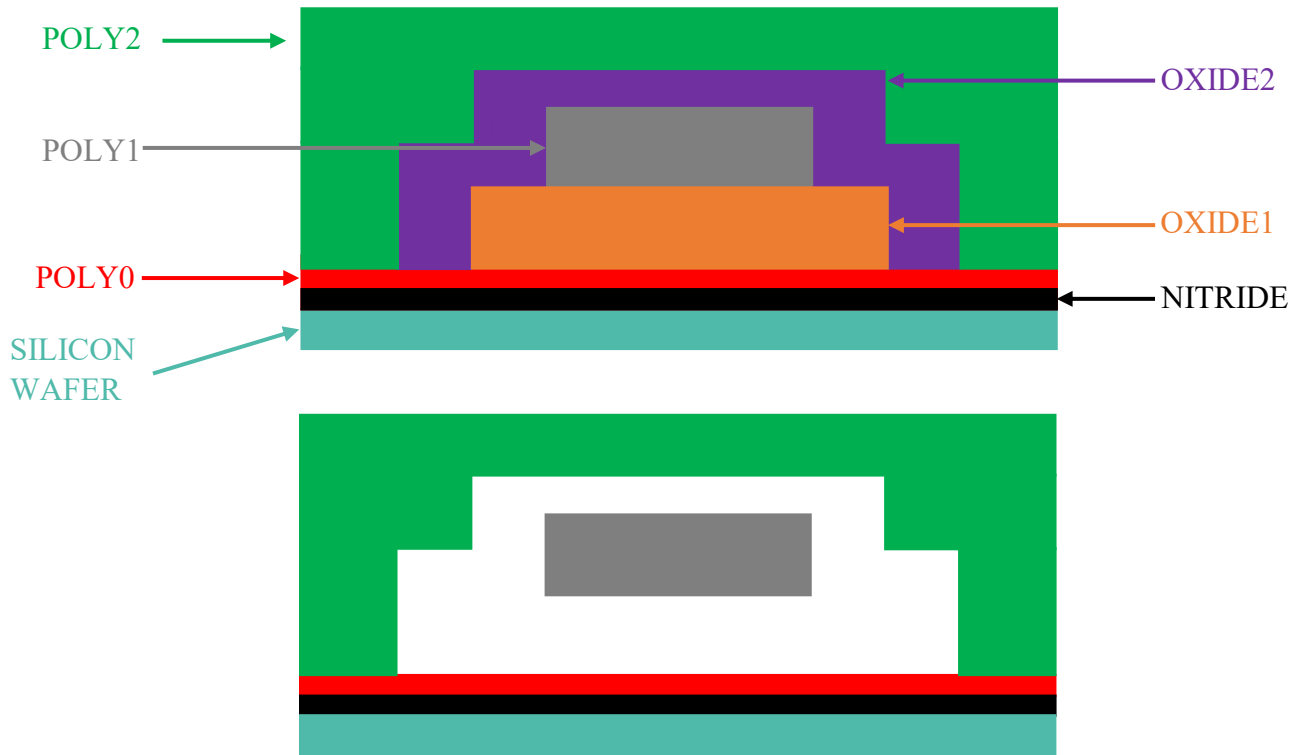


Figure 10 MUMPs POLY2 hasp (rounded corners not shown). Top: pre oxide removal. Bottom: after oxide removal.



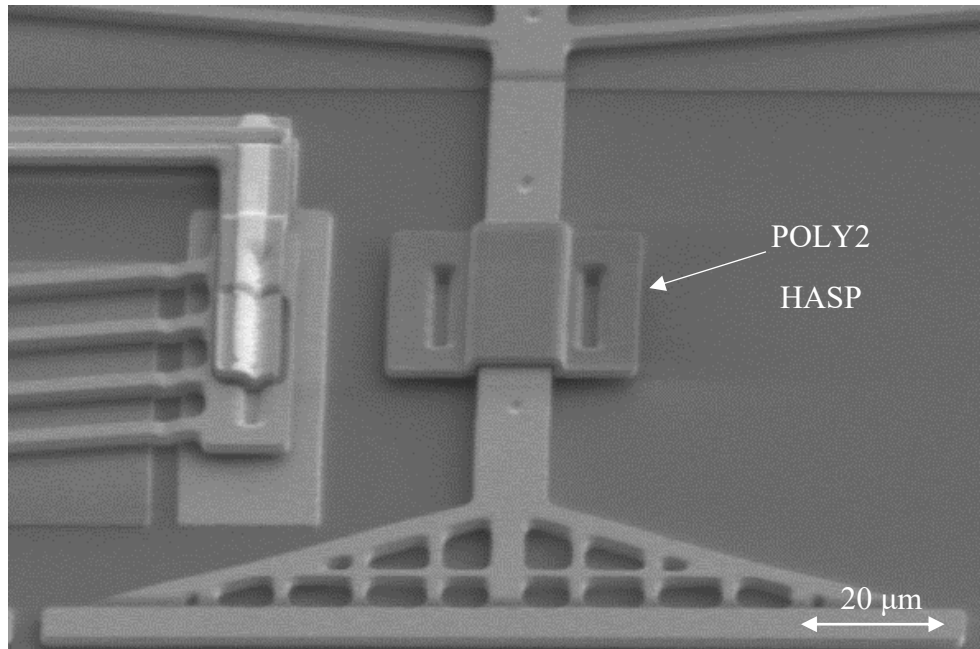


Figure 11 SEM photo of MUMPs hasp zoom in from Figure 3 earlier (Image: C. Arthur, Dalhousie MEMS Lab, chip = IMUDTRM1).

### 3. Microscopes

#### 3.1. Microscope Fundamentals

Microscopes grant us the ability to observe things that would otherwise be invisible to the naked eye. The smallest objects visible to the human eye have a diameter of approximately 75 microns [19]. However, with the aid of optical microscopes, we can explore the micro world. Depending on the type of microscope used, it is even possible to observe objects on an even smaller scale.

In a typical compound optical microscope, there are two lenses: one is the objective lens, which is closer to the specimen, and the other is the eyepiece. An objective lens is comprised of multiple lens elements that collaborate to create a magnified image of the specimen. The eyepiece lens is similarly constructed. Objectives and eyepieces are converging lenses that take incoming rays from an object and converge them to a focal point, resulting in the creation of a real image. A real image is one that can be projected onto a surface, viewed by an observer, or imaged by a camera.

In the most common variant of optical microscopes the objective and eyepiece are separated by a fixed finite tube length, the standard mechanical tube length is 160 mm, see Figure 12. Changing the mechanical tube length of a fixed finite tube will affect several parameters: the magnification, the field of view and the working distance, which is the distance from the end of the objective to the specimen.

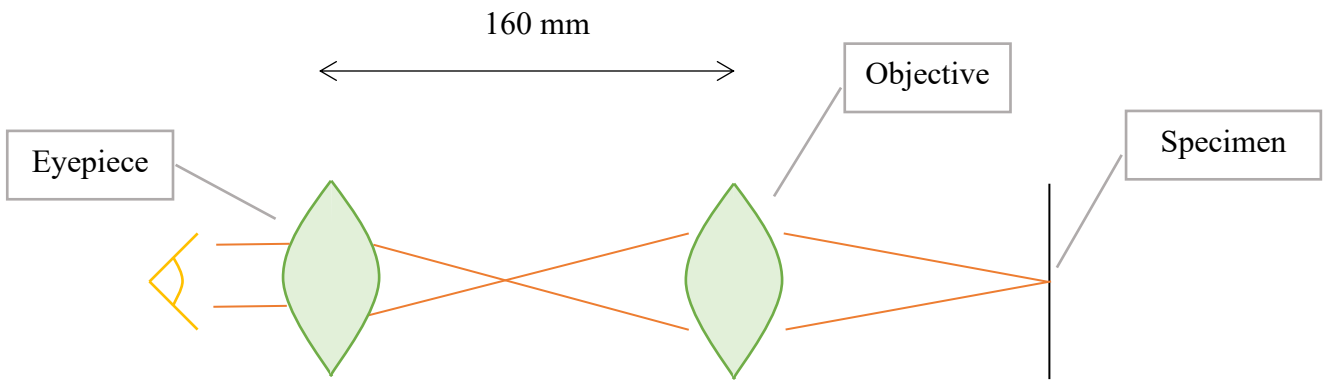


Figure 12 Fixed-Finite tube length of a typical microscope.

The thin lens equation (see Eq. 2 [20]) describes the relationship between the image distance ( $i$ ), object distance ( $o$ ), and the focal length ( $f$ ) in a converging lens (see Figure 13). For example: if  $o = f$ ,  $i = f$ , but if  $o = f/3$ ,  $i = 3/2 f$ .

	$\frac{1}{o} + \frac{1}{i} = \frac{1}{f}$	Eq. 2
--	---	-------

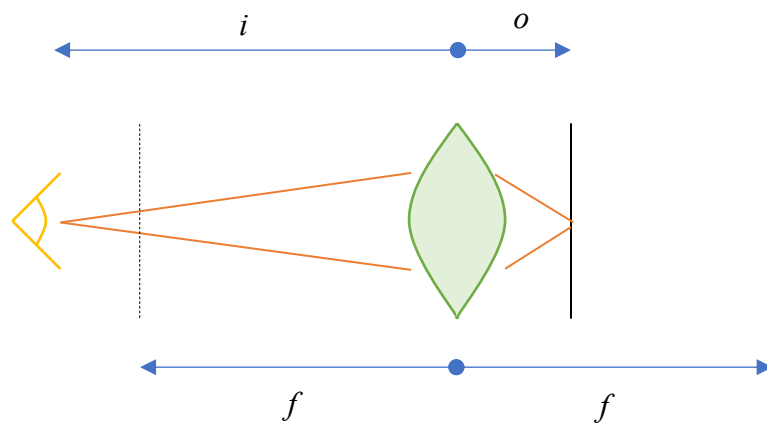


Figure 13: Thin Lens Equation.

The field of view (FOV) refers to the area that a microscope can observe. The size of the field of view decreases with increasing magnification. It can be determined experimentally by using an object of known size as the specimen. Eq. 3 demonstrates how to calculate the FOV (in x and y) for a camera [21].

$FOV(\mu m) = \frac{\text{Known Object Size } (\mu m)}{\text{Object Pixels}} \times \text{Camera Pixels}$	Eq. 3
---	-------

The Circle of Confusion (CoC) occurs when an object is not perfectly in focus and point sources become blurred circles of light (see Figure 14). From the lens equation, given a focal length,  $f$ , and an image distance,  $i$ , there is only 1 distance,  $o$ , where the image will be in focus. When an object is too close or too far from the subject, the blur spot becomes larger. Note, even when the lens is perfectly focused, we still get a small dot due to the optical limitations of lenses, but this dot is submicron.

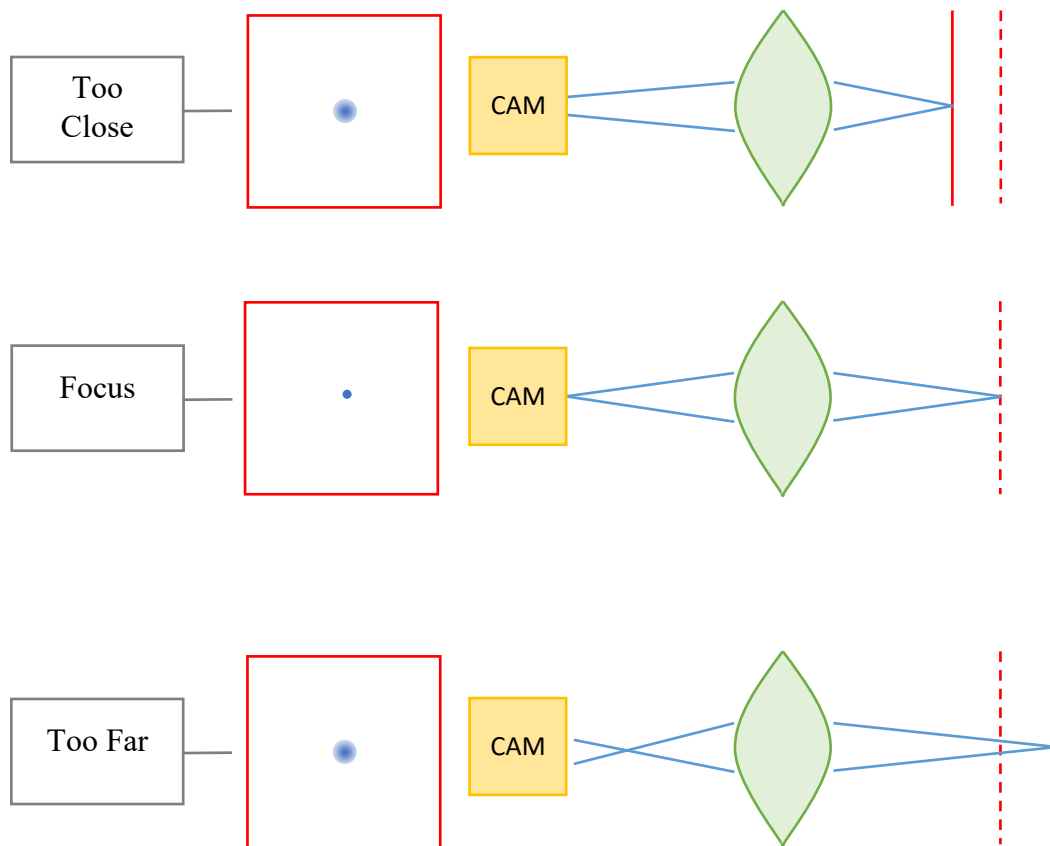


Figure 14: Circle of Confusion.

Depth of Field (DOF) is an important aspect that describes the distance between the nearest and farthest planes that appear in focus in the image. The range of distances where the circle of confusion is perceived as sharp is known as the depth of field. If a microscope has a large depth of field, then there is more leeway in the positioning of the camera and a larger working distance range. As magnification increases, the depth of field decreases hence micro-photography has a shallow depth of field. As the object's distance is decreased, the depth of field also decreases, hence up-close macro photography has a shallow depth of field [22].

### 3.2. Magnification Problem

Increasing the magnification in a microscope has several effects on the image. Firstly, the brightness of the image is inversely proportional to the square of the magnification,  $M$ , meaning that as the magnification increases, the brightness of the image decreases. This can be managed by using a brighter light source. Secondly, higher magnification also results in a reduction in the field of view, limiting the area visible in the microscope. Thirdly, image vibration is exacerbated as the magnification of the image increases. Image vibration is typically micro-sized, it can be caused by various factors, such as building vibrations, user coupled movements, and minor mechanical disturbances.

Finally, as the magnification increases the depth of field decreases, making it harder to keep multiple depths of a specimen in focus. When viewing specimens at an angle the issue is far worse. Consider a microscope with a FOV of 100 microns and a DOF of 3 microns. When viewed at 45 degrees, the required depth of field is 70 microns, far more than the available DOF. The resultant image has a narrow strip referred to as the "focus bar" (see Figure 15). Only the section of the image within this focused area will appear sharp and distinct. Any other part of the image outside this focus bar will be blurred, making it crucial to place the focus bar over the region of interest to accurately capture the desired details.

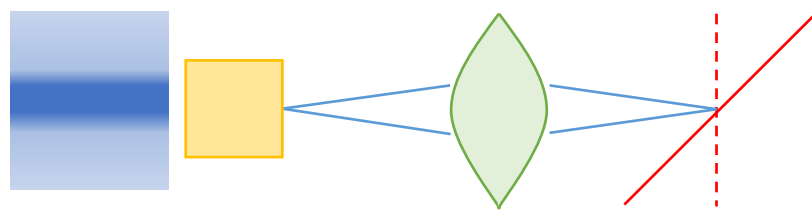


Figure 15 Perspective view of a specimen showing focus bar.

### 3.3.Objectives & C-mounts

The rated magnification value of the objective lens (indicated on the lens, see Figure 16) is calculated for the standardized tube length of 160 mm. However, by adjusting the tube length the magnification of the objective lens can be changed, subsequently affecting both the field of view FOV and the depth of field DOF. These alterations allow us to manipulate the microscope's overall magnification and focus characteristics accordingly. C-mounts serve as a standard lens mount with a diameter of 1 inch and a 32 TPI thread. C-mount adapters are available to connect to objective threads (0.79 inch diameter and 36 TPI) and other optical components such as cameras. C-mounts provide the flexibility to customize the microscope's geometry to meet specific observation requirements. With the addition of C-mounts, we can increase the magnification at the expense of reducing the field of view and depth of field.



Figure 16 Left: photo of objective. Right: photo of 40 mm long C mount threaded tube.

The working distance is the distance between the microscope's objective end and the specimen. Common microscope objective working distances are on the order of millimeters. Typically, the working distance decreases as the objective magnification increases. Long working distance (LWD) is desired for the microscope this allows for greater room for positioning of equipment close to the specimen without interfering with the objective lens. Long working distance objectives are available but cost more than the common objectives as they are more specialized [23].

## 4. Experimental Setup: Table Microscope

### 4.1. Camera

Rather than using an eyepiece, a digital camera can be used to record the microscope images. The camera has no lens and is mounted using the standard C-mount: the microscope acts as the camera lens. The camera does not provide any extra magnification. In fact, magnification has no strict meaning when using a microscope camera, as the image can be made larger or smaller simply by displaying it on a larger or smaller screen. Better measures are either the FOV in microns or the resolution in microns/pixel. To calculate resolution, one can examine a specimen with a known size and measure the number of pixels it occupies on the screen. By multiplying by the camera pixels, the field of view of the microscope can be determined.

A digital camera connected to the microscope can be controlled through a computer program such as MATLAB, offering various functionalities for adjustments. In MATLAB, users can modify the camera's exposure, gain, and colour balance settings. Exposure refers to the brightness and darkness of the image captured by the camera. Gain, on the other hand, amplifies the signal received from the camera. Lastly, the colour balance feature allows users to adjust the colours of the captured image, providing customization options for better visualization and analysis.

### 4.2. Illumination

MEMS chips are highly reflective and closely resemble mirrors. When imaging a MEMS chip from directly above, it is required that the light source approaches the specimen perpendicularly. An in-line illumination setup utilizes a beam splitter, which splits the light into two separate beams (see Figure 17). The light source enters the beam splitter from the side and half of the beam is discarded, the other half is reflected down to the specimen. The reflection from the specimen re-enters the beam splitter from below and half is discarded, the other half reaches up to the camera and captures the image for display. As a maximum of one quarter of the light reaches the camera, a bright light source is required.



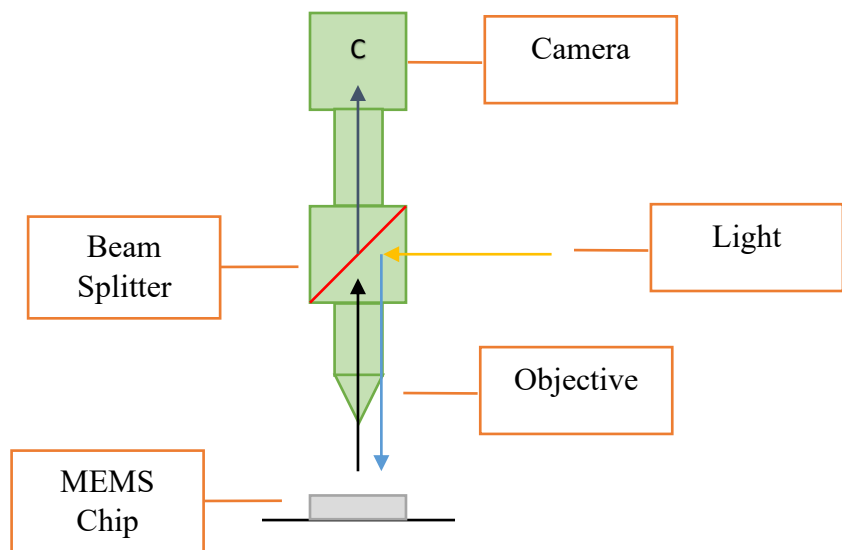


Figure 17 In-line illumination.

When imaging off-axis, positioning the lighting at the angle from the camera is essential. Failure to achieve proper lighting alignment may lead to a black image or the presence of artifacts that can significantly interfere with the image quality. When illuminating off-axis, the in-line illumination technique will not work effectively as the illumination fails to reflect to the objective. The illumination must be placed at the same angle with respect to the specimen as the objective. For example, if the objective is at 45 degrees, then so must the illumination be, in this case, it is 90 degrees from the objective (see Figure 18). This setup is referred to as the "perspective view" because the camera is now positioned to provide 3D information.

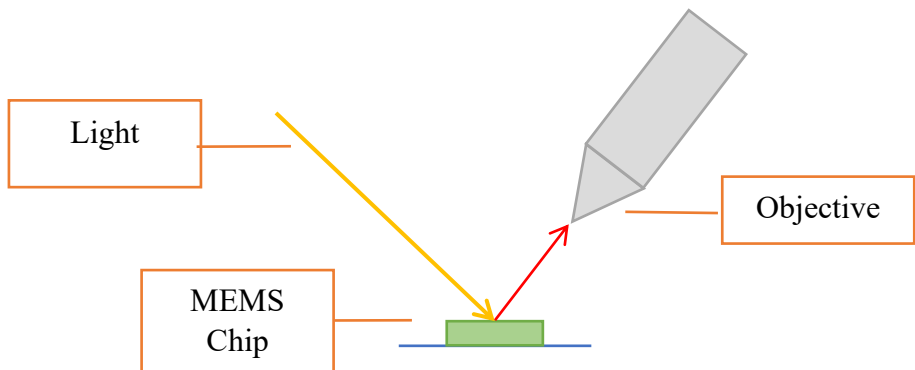


Figure 18 Off-axis illumination.

### 4.3. Tabletop microscope

The decision was made to create a bespoke horizontal microscope setup instead of using a standard vertical setup. The setup is shown in Figure 19, Figure 20 and Figure 21. There were various reasons for choosing this setup, including countering the effects of gravity, and achieving easier access to the MEMS chip by reorienting it so that the user can see it along with the microscopes.

For our setup, we chose a 2-microscope configuration: a 90-degree microscope captures a head-on image of the MEMS chip, while the other 45-degree microscope is used to capture the perspective view. To achieve this, we built two separate microscopes. The 90-degree microscope designated for capturing head-on images features a beam splitter and a 90-degree LED for in-line illumination of the chip. On the other hand, to capture the perspective view, an off-axis lighting system is required to ensure the best possible lighting conditions. A 45-degree LED light source is focused into a spot by a stand-alone microscope objective mounted on a manual XYZ stage. The stage is used to accurately position the off-axis light spot. Each microscope has a separate tube assembly and separate digital cameras. Figure 22 shows a sample of a portion of a MEMS chip showing a thermal actuator. This photo was taken with the 90-degree setup, while Figure 23 shows a sample photo taken with the 45-degree setup. Note that the 2 cameras and the sample have different XYZ coordinate frames.

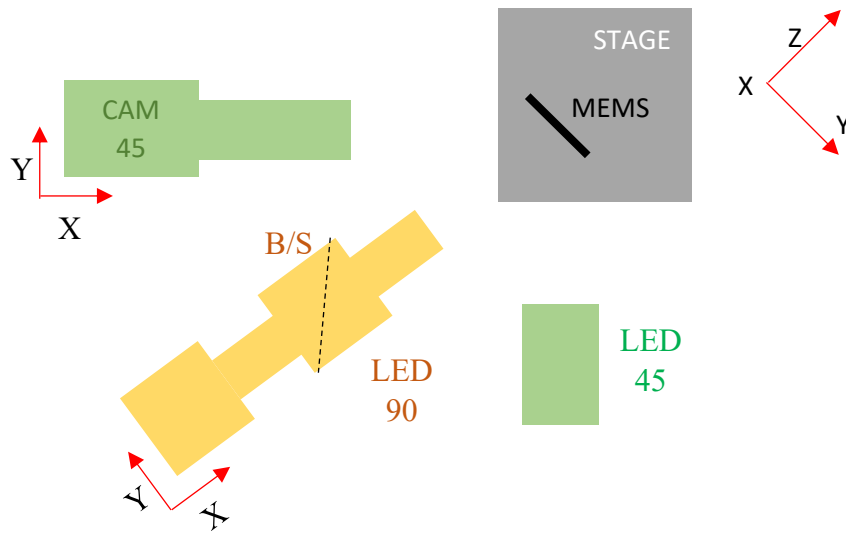


Figure 19 Top view schematic of 90-degree (yellow) and 45-degree (green) setups.

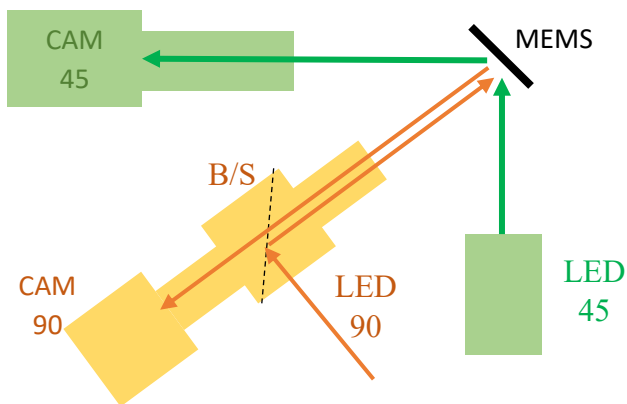


Figure 20 Light paths of 90-degree (yellow) and 45-degree (green) setups.

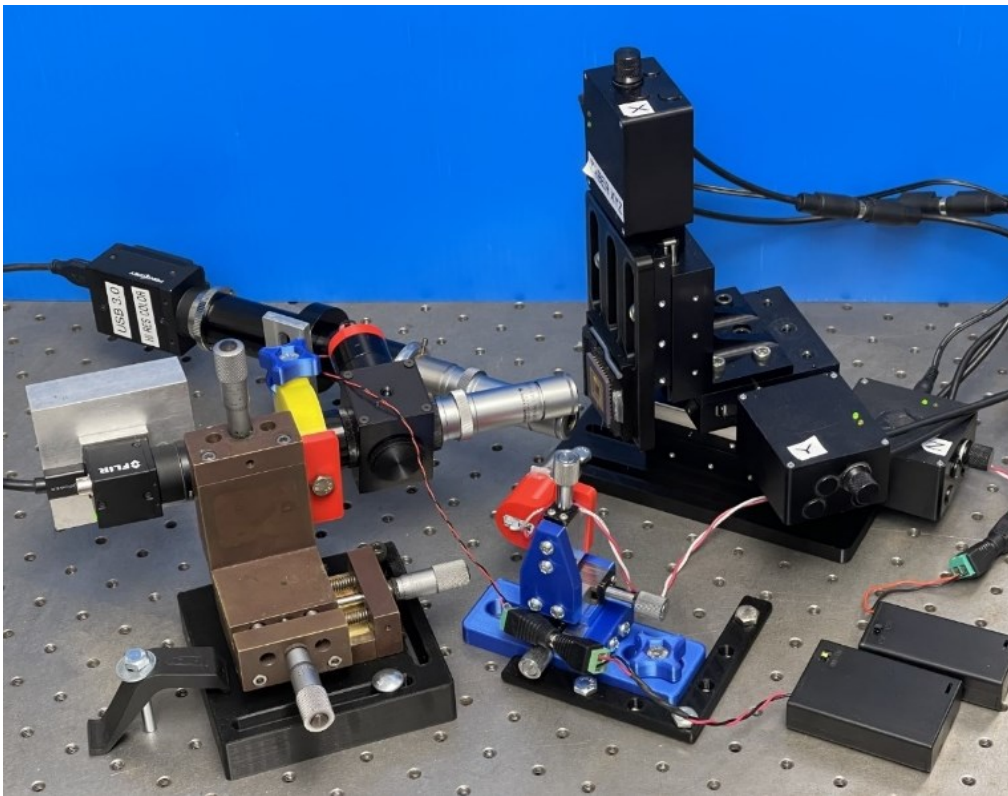
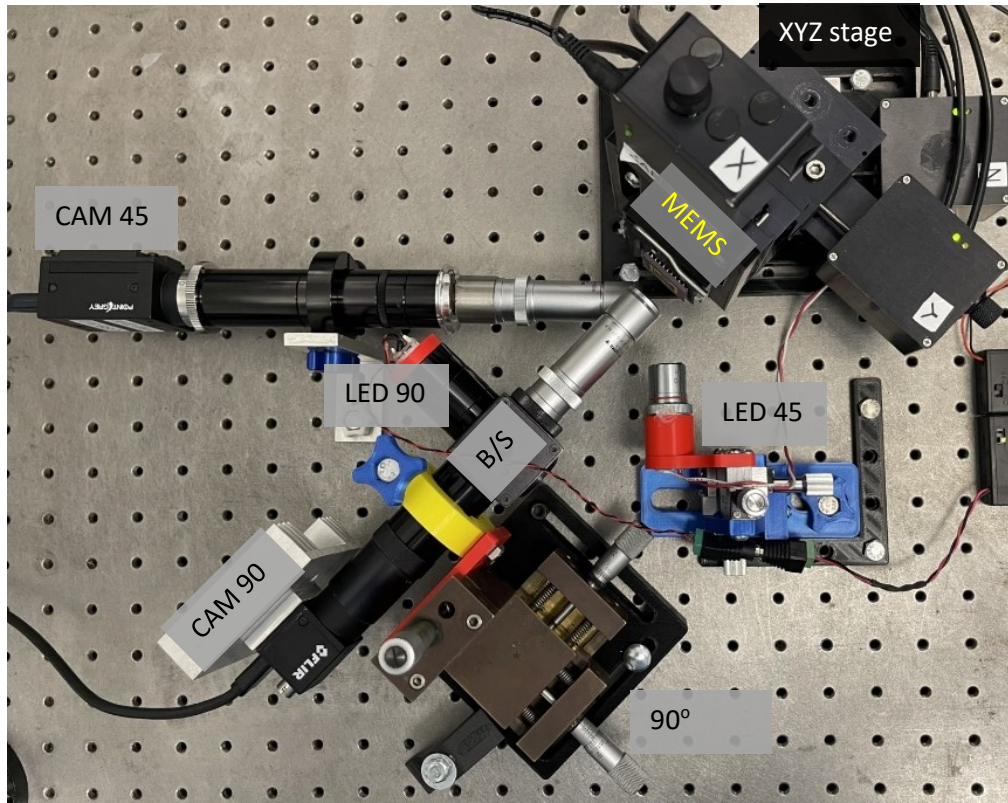


Figure 21 Top view and perspective photos of table microscope. The table holes are spaced 1" apart.

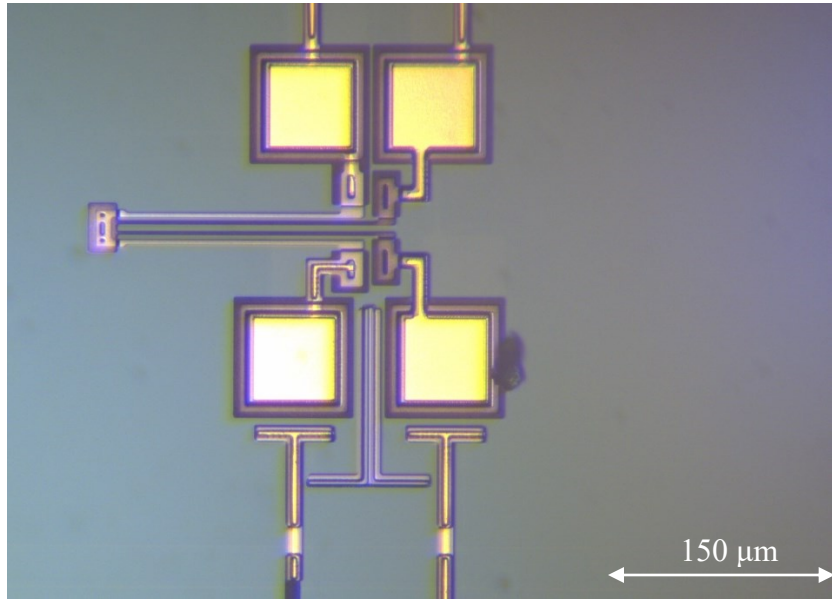


Figure 22 Sample 90-degree photo of bonding pads and a thermal actuator (chip = IMUDTVT2).

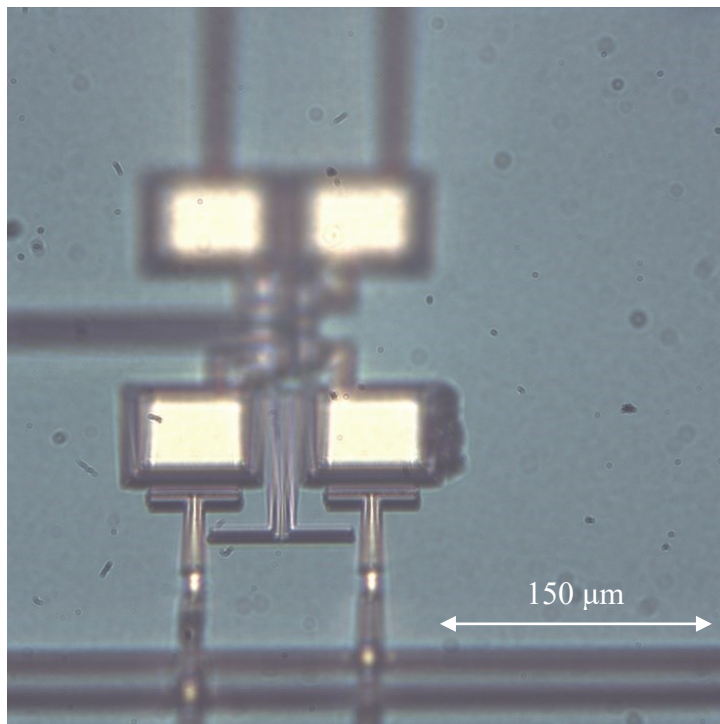


Figure 23 Sample 45-degree photo of bonding pads, note narrow focus bar.

As we chose not to use a standardized microscope setup, we gained the flexibility to customize our microscopes using C-mounts, beam splitters, and other components, as well as the ability to employ various illumination techniques. For example, a Doubler lens is a C-mount attachment that increases magnification, enabling us to view a more magnified image of the chip. It is mounted just before the camera. However, when using a doubler, the working distance is reduced, necessitating the objective to be closer to the specimen.

By customizing the microscope, we decided to test multiple combinations with different objectives and tube lengths, both including and excluding a doubler. This approach allowed us to examine the effects on the working distance and the field of view, helping us determine which combination provided the best results for our needs. Specially we wished to have a small FOV (~400 microns) to view fine detail, but we also wished for a large WD (>10 mm) to allow illumination from the side. In Table 1 the effect of objective on FOV & WD was measured on 90-degree images. Images were taken of a MEMS chip, all capturing the same feature with a known actual size for reference. Throughout this test, the Beam Splitter, Doubler, and a tube length of 160 mm were kept constant. Only the objectives were changed, leading to variations in WD & FOV. The diameter is the physical diameter of the objective & angle is the subtended cone angle of the objective, these values will be used later.

Objective	WD (mm)	FOV (um)	Diam (mm)	Angle (deg)
4x	17	1301	38	58
6x	17	859	36	56
10x	7	528	38	108
8x (LWD)	16	1243	42	66
25x (LWD)	13	423	42	78

Table 1 Objective comparisons: fixed Beam Splitter, Doubler, and 160 mm Tube Length.

From the table, the FOV & WD decreased as we used larger objective magnification see 4x to 6x to 10x. Note that the 8X & 25X objectives had relatively large WD & FOV. The 8X & 25X objectives were long working distance objectives, while the 4x, 6x & 10x were standard objectives.

In Table 2 the effect of tube length on FOV & WD was measured, we tested only the 25x LWD objective. During this test, we utilized a beam splitter but excluded the doubler from the setup. We also varied the tube length to observe its impact on the working distance and field of view. The working distance remained relatively constant in both cases, with the only noticeable change being the field of view, which decreased as the tube length was extended. This finding suggests that longer tube lengths provide more magnification with little WD penalty.

Tube Length (mm))	WD (mm)	FOV (um)
135	10	844
150	9	778
165	11	716
200	11	649

Table 2 Effect of tube length on WD & FOV for 25X objective.



#### 4.4.Zaber XYZ Motion Stage

To ensure the utmost accuracy and precision in our microscope setup, it was essential to have a stage that could facilitate exact focus adjustments. A Zaber motorized stage was chosen to meet the requirements (see Figure 24). The T-LS28 Zaber model stage offers movement in the X, Y, and Z axes, providing a range of 28 mm in each direction, and a resolution of 0.1 microns. The stage has a maximum speed of 4 mm/s. The Zaber system operates on a 12V power supply. These features make it well-suited for examining MEMS chips on the micron scale [24].



Figure 24 Zaber XYZ stage, table holes are 1” apart. XYZ letters refer to camera axes.

Zaber stages have manual joystick controls and can also be controlled through a computer, either using the Zaber application or MATLAB. Note that the physical controls on the Zaber stages do not offer the same level of precision as when controlled through a computer. When controlling through MATLAB, the user can send precise movements to the Zaber stage and specify when to execute them, allowing for process automation.



#### 4.5. Microscope interference.

Using the setup for the 25x objective shown in Table 1 it was discovered that both the 90-degree and 45-degree microscopes had working distances of approximately 13 mm and the subtended angles of the objectives were 78 degrees and 78 degrees (see Table 1). We realized that using them simultaneously would result in interference, the objectives of both microscopes would have to occupy the same space above the specimen as shown in Figure 25.

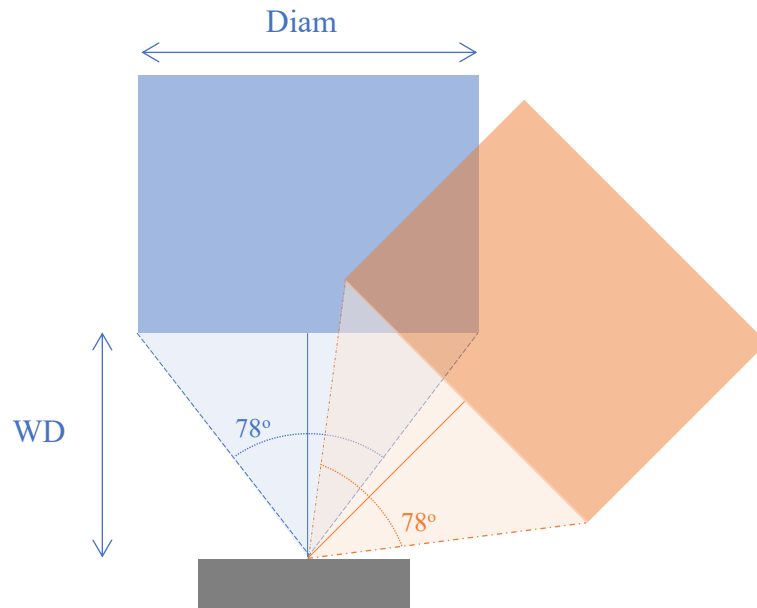


Figure 25 25x microscope objectives 90-degrees and 45-degree interference.

It is impossible to use both microscopes simultaneously, but we wish to image the same sample using both microscopes. To address this issue, we employed the Zaber stage holding the MEMS chip. The microscopes were vertically offset, with the 45-degree microscope 22 mm lower than the 90-degree microscope, thus avoiding physical interference (see Figure 26) The MEMS chip can be quickly imaged with the 90-degree microscope and then using the Zaber the chip is lowered by the vertical offset to use the 45-degree microscope. This setup enabled us to precisely match the location of the MEMS chip being imaged with both microscopes, ensuring that we observed the same scene with each microscope.

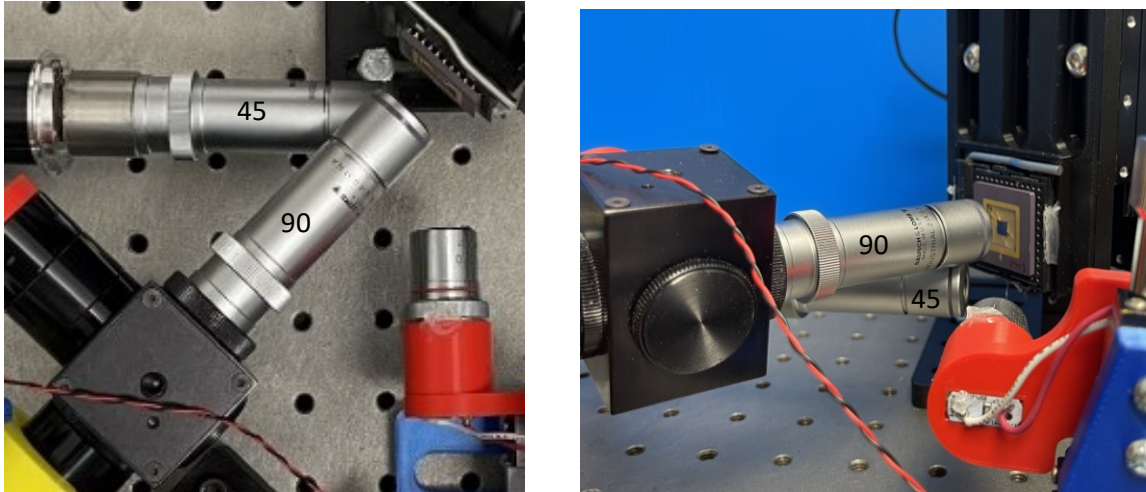


Figure 26 Vertically offset microscopes as seen from above and from 45-degree LED. Table holes are 1” apart.

#### 4.6. Image Acquisition MATLAB

A MATLAB script was written to capture images. This code controls the Zaber stage and both cameras and enables image acquisition of the MEMS chip. The process begins by capturing a 90-degree image. Afterwards, the Zaber stage, housing the MEMS chip, is moved down to align with the 45-degree perspective view of the microscope camera to capture a sequence of images.

For example, Figure 27 shows a 90-degree image of a displacement actuator, which moves the gear when the actuators are heated. Figure 28 shows three sample 45-degree images, note focus bars. As another example, Figure 29 shows a 90-degree image of a large perforated plate, and Figure 30 shows two sample images of the same plate at a 45-degree angle. For the perspective view portion, a *for* loop is employed to instruct the Zaber stage to make small steps to move the focus bar along the image. After each step in the loop, an image is taken to capture multiple images with different focal points. This process ensures that a portion of each image taken has appropriate focus.

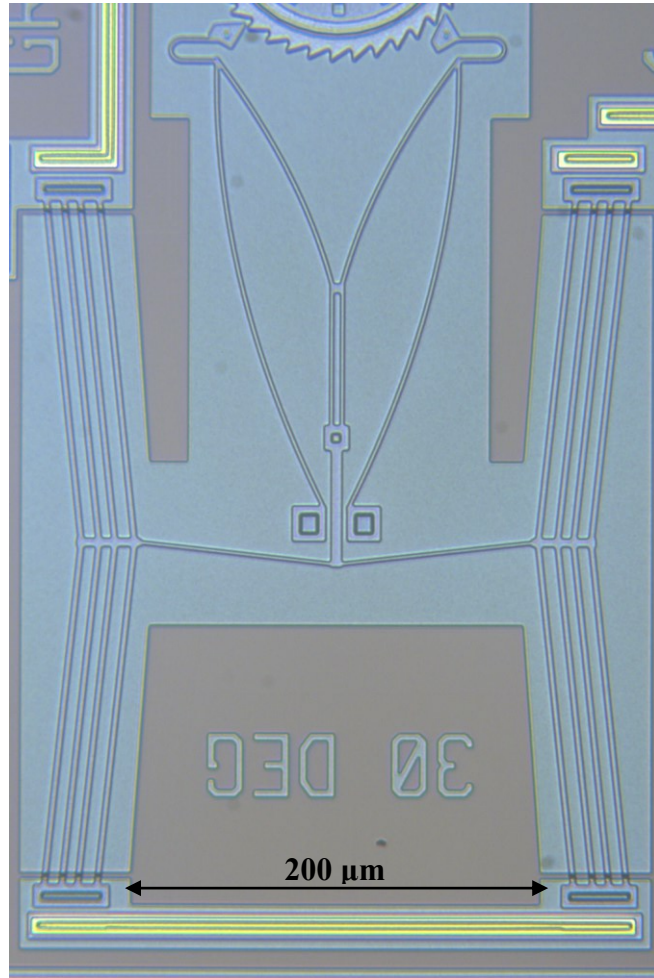


Figure 27 A 90-degree 2D Microscopic image of a MEMS actuator (chip = IMUDTRM1).

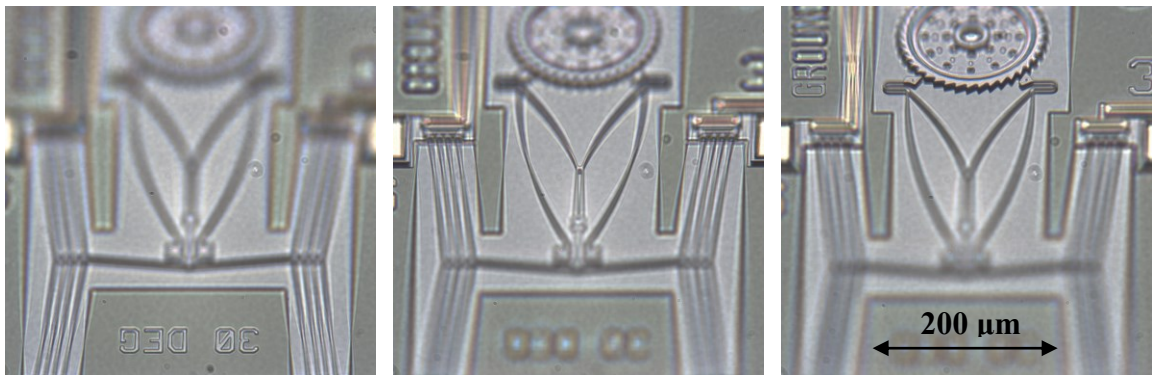


Figure 28 Sequence of 45-degree images of MEMS actuator, showing rising focus bars.

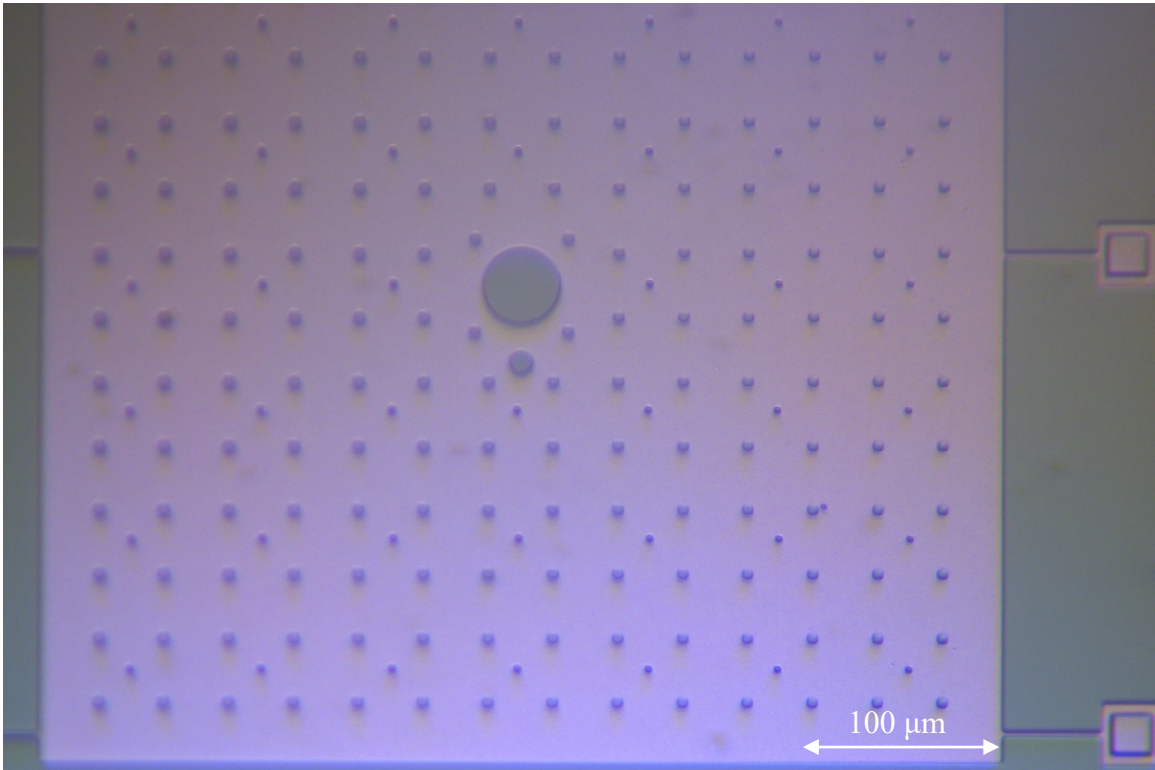


Figure 29 90-degree 2D image of a MEMS plate with perforations (chip = IMUDVT2).

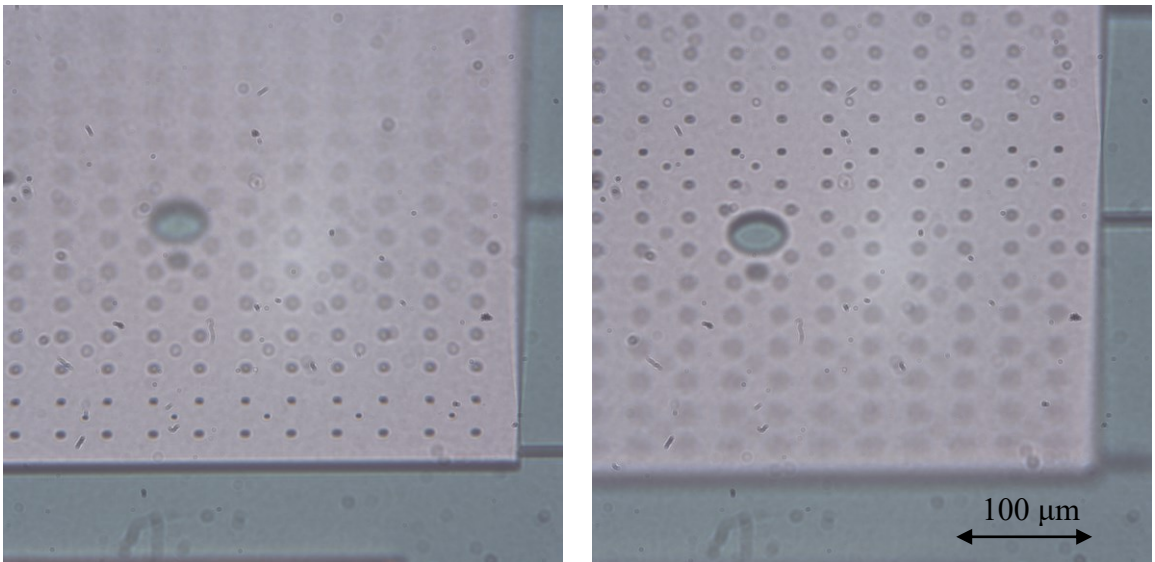


Figure 30 Pair of 45-degree images of perforated MEMS plate showcasing focus bars at different locations.



## 5. 3D Imaging Processing

### 5.1.Registration

Image registration plays a crucial role in generating a fused image with minimal noise or artifacts resulting from misalignment. If image registration is not first carried out, the resultant images may be unusable, as there is always a source of vibration that misaligns the images. When working in a micrometer scale, even small movements can make a significant difference in the result compared to the macro world.

To execute image registration, a separate program is used. A dedicated directory named "REG" is created to store the newly registered images. Each image is read and stored within a stack. Two vectors are initialized: one to store the horizontal (x) shifts and another for the vertical (y) shifts.

A *for* loop begins to iterate through all the images in the stack. During each iteration, the images are converted to grayscale. These grayscale images are then passed through a Matlab add-on function called "measureShift" [2]. This algorithm was written by Yatsenko [2] and is available as a Matlab add-on. This function works using the current image and the one before it, which allows for comparative analysis that gives shift values in both the x and y directions. The "measureShift" function uses cross-correlation techniques to determine the shift at the whole-pixel level. The image is then adjusted using the whole-pixel shift calculated earlier. A phase measurement is then done, which enables the calculation of subpixel shifts within the image. These shift values are then returned to the main script, providing information about the alignment adjustments.

The values resulting from the iteration are stored in the previously initialized arrays. The current image is then designated as the preceding image within the iterative *for* loop. This sequential arrangement ensures that the comparison of image shifts takes place in the same order as the images in the stack. This sequential comparison approach is necessary due to the presence of focus bars in the images.

When comparing with just the first image, the initial and final images differ significantly because the focus bar in the first image is fully out of focus.

Cumulative shifts are calculated independently for both the x and y directions. This involves subtracting the minimum shift value from the maximum shift value for each respective direction. This procedure quantifies the overall translation of the images along both the horizontal and vertical axes. Typical values of cumulative shifts are in the range of 0 to 30 pixels in x and/or y.

These computed shift values serve as the foundation for generating an initial empty canvas. This empty canvas is bigger than the image: the dimensions of this canvas are established by adding the dimensions of the first image to the calculated cumulative shifts. This preparatory step is essential as it ensures accurate alignment of all images on the canvas, positioning them correctly for subsequent cropping.

In a subsequent iterative loop, the images are placed onto the canvas, utilizing their shift values to ensure precise placement. Following this alignment phase, any excess white space or bars resulting from image placement are cropped out. This cropping step produces final images that are: 1) registered to each other 2) exclude edges that are only present in edge images. These registered images are then written to disk in the previously mentioned REG directory.

In the following section, the fusing algorithm that combines the images will be explained in detail. However, to demonstrate the necessity of pre-registering images Figure 31 shows two fully fused images: one fused without image registration, while the other first registered and then fused. There are significant differences as the unregistered image exhibits noticeable shifts and skew and the wires appear chopped and are not straight, unlike the registered image (see zoomed-in Figure 32).

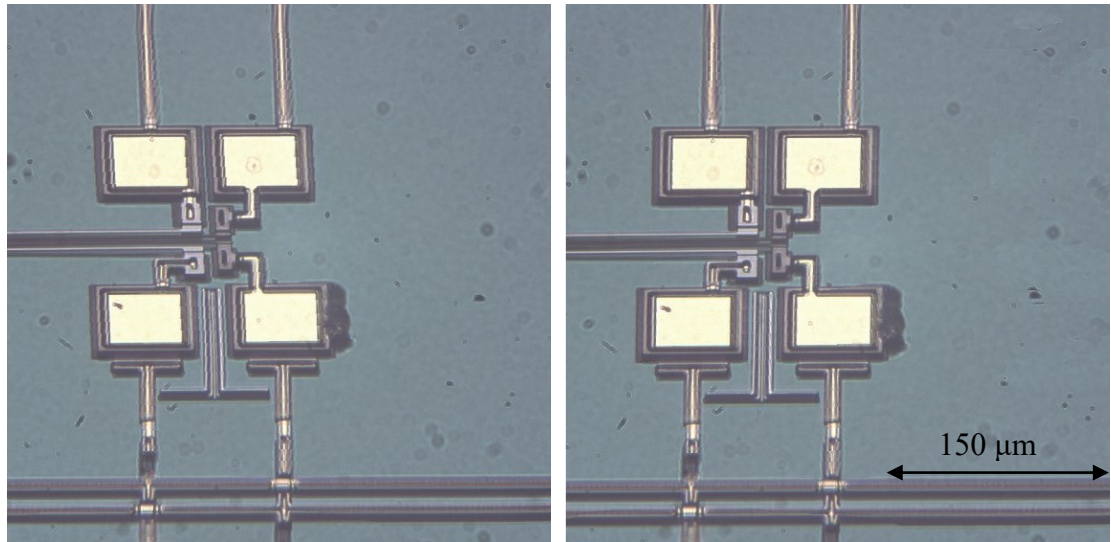


Figure 31 Importance of Image registration: Left: fused unregistered images, Right: fused registered images (chip = IMUDTVT2).

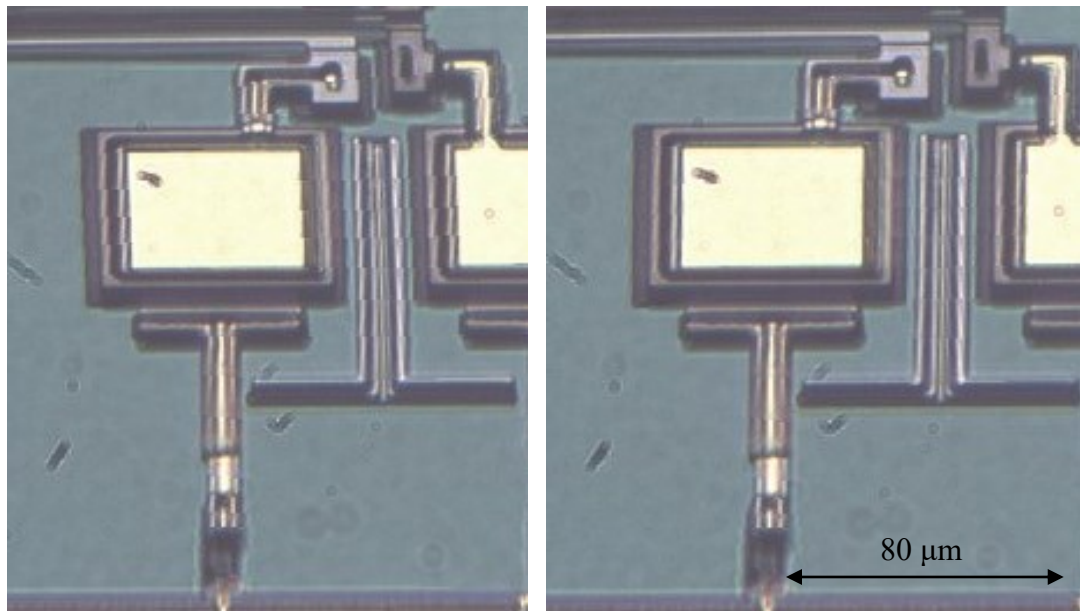


Figure 32 Zoom-in of the Figure above: Left is unregistered, Right is registered

## 5.2.Focus Algorithm

Once the sequence of 45-degree images has been captured and registered, the focus measure of each pixel needs to be determined. To achieve this, a *for* loop is used to iterate through each picture. Within the loop, the 45-degree images are read and then stored in a stack. In the following example, the device is a MEMS grid actuator as shown in a 90-degree image in Figure 33. An image stack of 45-degree images is stored in Matlab as a 4-dimensional array (see Figure 34). The first two dimensions in the array represent the height Y and width X of the images, the third dimension Z is used to stack multiple images, and the fourth dimension holds the RGB colour data.

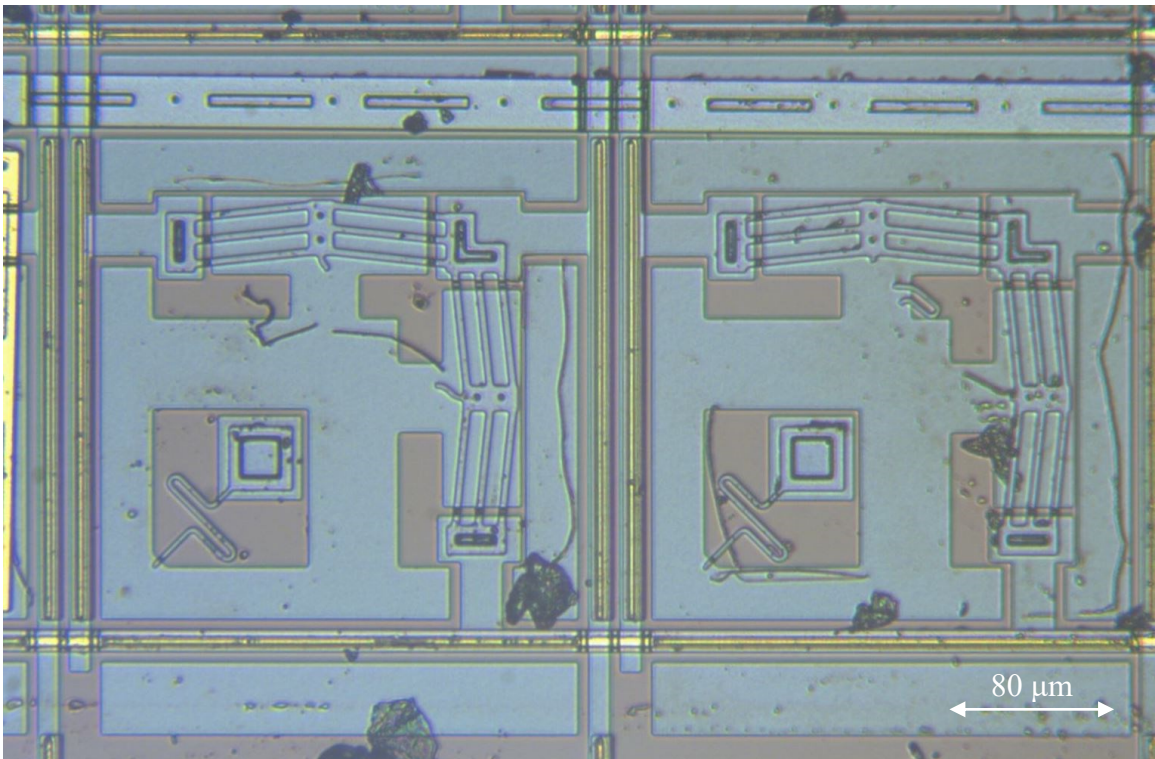


Figure 33 2D image of MEMS grid actuator (chip = IMUDTPH1).



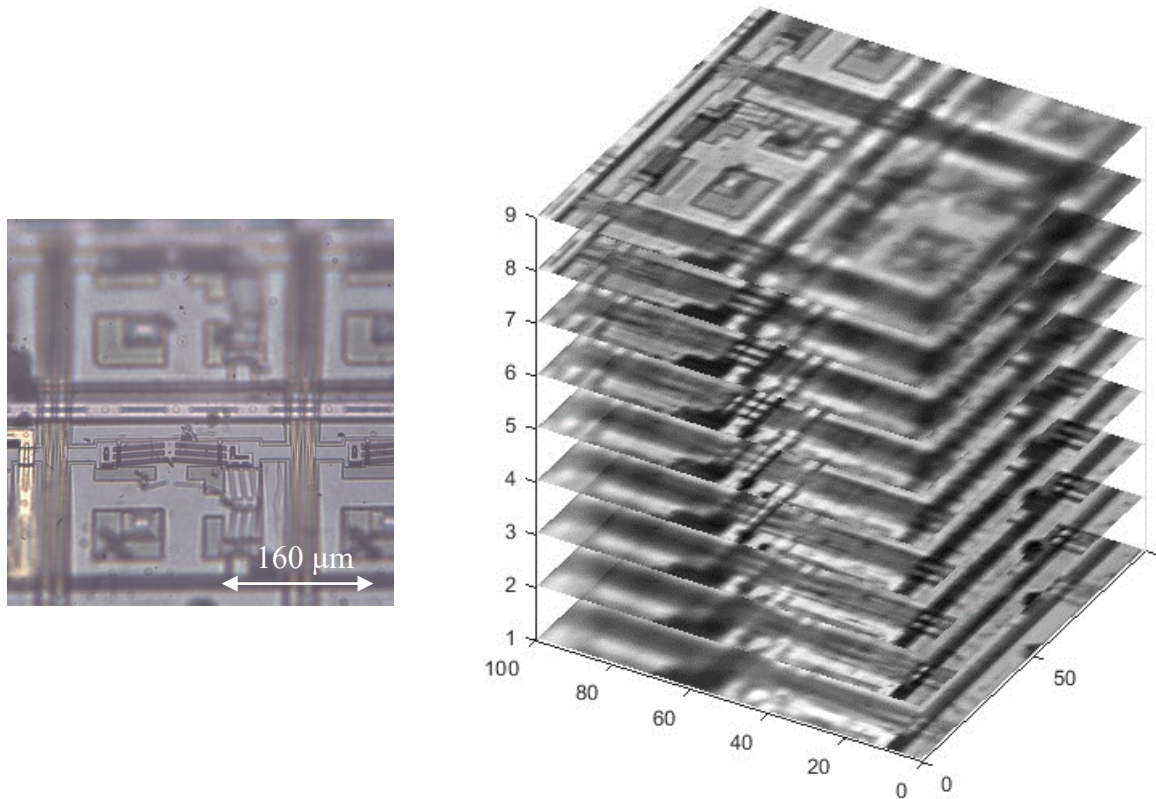


Figure 34 Left: 1 sample 45-degree image, Right: sample stack of 9 45-degree images. Z axis is the image number. XY axes are % of the image.

Once all the images are stored in the stack, another *for* loop is initiated. Within this loop, a Matlab function called "squeeze" is employed to extract an image from the stack. Squeeze collapses the extra dimensions to leave a 2D RGB image. The extracted image is then converted to grayscale and processed through the Matlab focus algorithm known as "focusmeasure".

This algorithm was written by Pertuz [25] and is available as a Matlab add-on. This algorithm allows the user to choose focus measure methods such as BREN and others to assess the sharpness and clarity of the image. The focus algorithm evaluates the focus value of each pixel by considering its neighboring pixels. The user defined number of neighbors is used to specify the radius of nearby pixels being analyzed. The algorithm returns a grayscale focus image of the same size as the original image. Figure 35 shows a sample image and the resultant focus image. The focus image is a greyscale image, displayed in yellow/blue false colour for clarity.

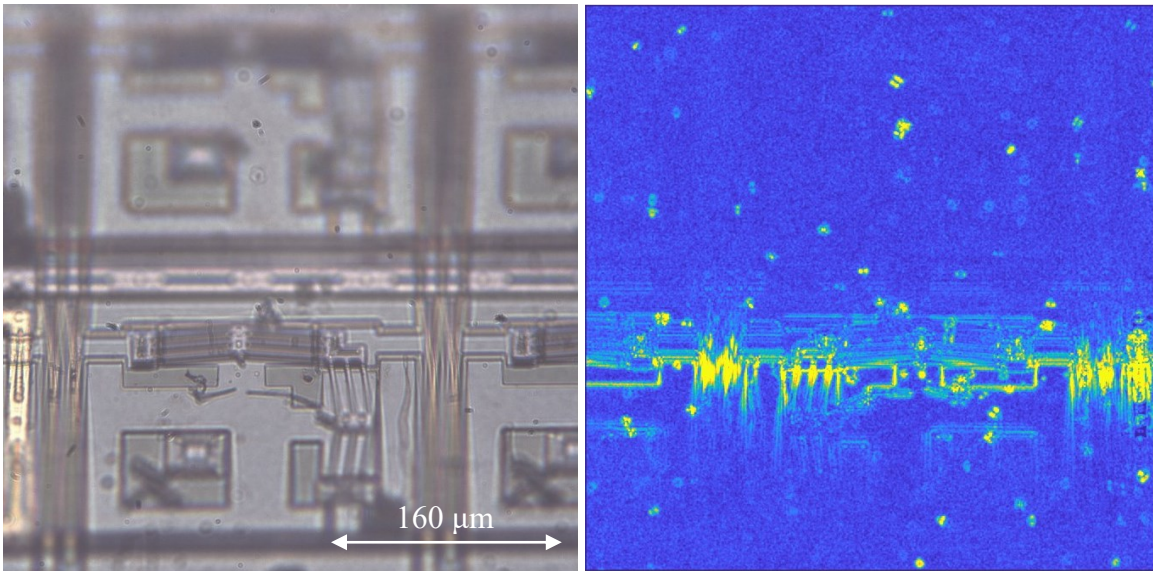


Figure 35 Left: 1 sample 45-degree perspective view RGB image. Right: resultant focus measure image, highlighting in-focus areas (greyscale false colour, yellow is in focus).

The focus images are trimmed to remove any erroneous data from the edges due to incomplete neighbourhoods. The trimming width is chosen to be half the number of neighbours. Pixel values in this rectangular frame are set to zero. The resulting trimmed focus images ensure that the data at the edges is excluded from further analysis or processing.

Because values from the focus measure algorithm can be tiny (e.g.  $10^{-4}$  or large e.g.,  $> 1$ ) for ease of handling the script proceeds to normalize the values to between 0 and 1. First, the minimum value of the focus images is determined, excluding any zeros. Then, this minimum value is subtracted from all elements in the images, effectively setting the minimum value to 0. Subsequently, the maximum value in the focus images is identified, and all elements are divided by this maximum value. This normalization process ensures that all focus images are rescaled to a range between 0 and 1, facilitating further analysis and processing of the focus images.

The resulting greyscale focus images are stored in a focus stack, represented as a 3-dimensional array containing the dimensions of height  $Y$ , width  $X$ , and the third dimension  $Z$  is used to stack multiple focus images together (see Figure 36).

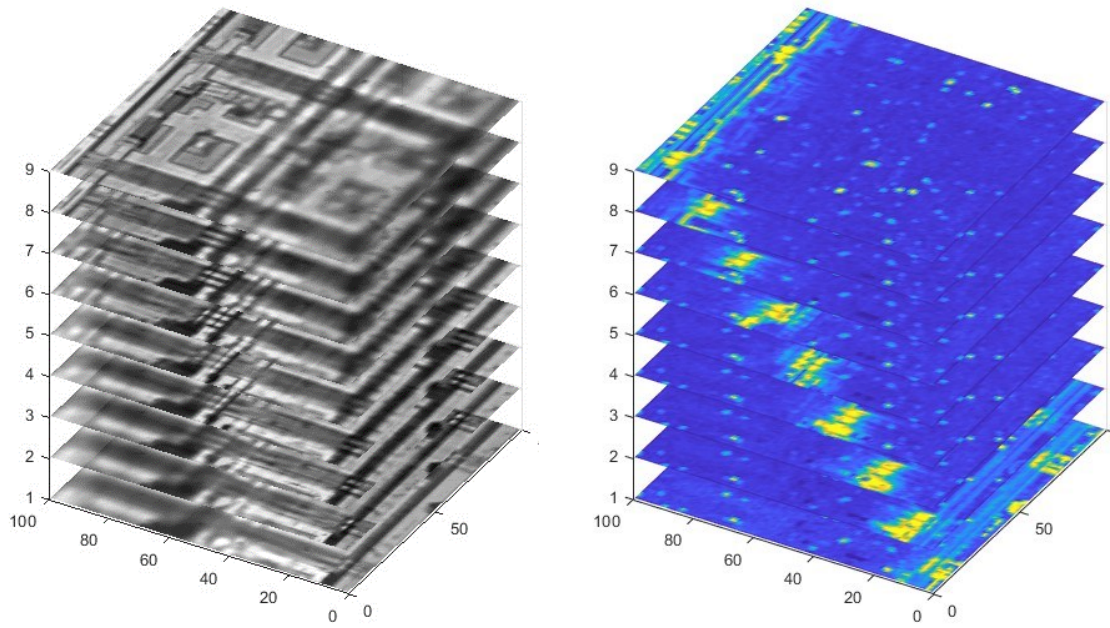


Figure 36 Left: 45-degree image stack, Right: focus stack (false colour). Z axis is the image number. XY axes are % of the image.

### 5.3. Depth Map

A depth map is an image that provides information to the user, detailing which points on the surface are closer to the viewpoint. In our case, the desired depth map shows which image is in the best focus at a certain pixel. This information is critical to determine during image fusion, as it guides us in selecting the appropriate images for the fusion process.

Before generating a depth map, it is essential to ascertain the maximum focus at each pixel. This data serves as an indicator to identify the image with the most suitable focus for a specific pixel. The process to determine the maximum focus per pixel is called "drilling." During this procedure, we examine the corresponding pixel in each image within the stack and identify the image with the highest focus.



Figure 37 shows the 90-degree 2D overhead view of a MEMS bearing with an electrostatic motor along with a gear rack. Figure 38 Top shows the focus image for 1 45-degree view of the device in Figure 37. Superimposed on this a selected pixel is depicted in red. We drill into the focus stack at this pixel location (Figure 38 Bottom Left) recording the focus values for each image as shown in the focus curve (Figure 38 Bottom Right). This analysis enables us to determine which image has the maximum focus value (image 8 in this case).

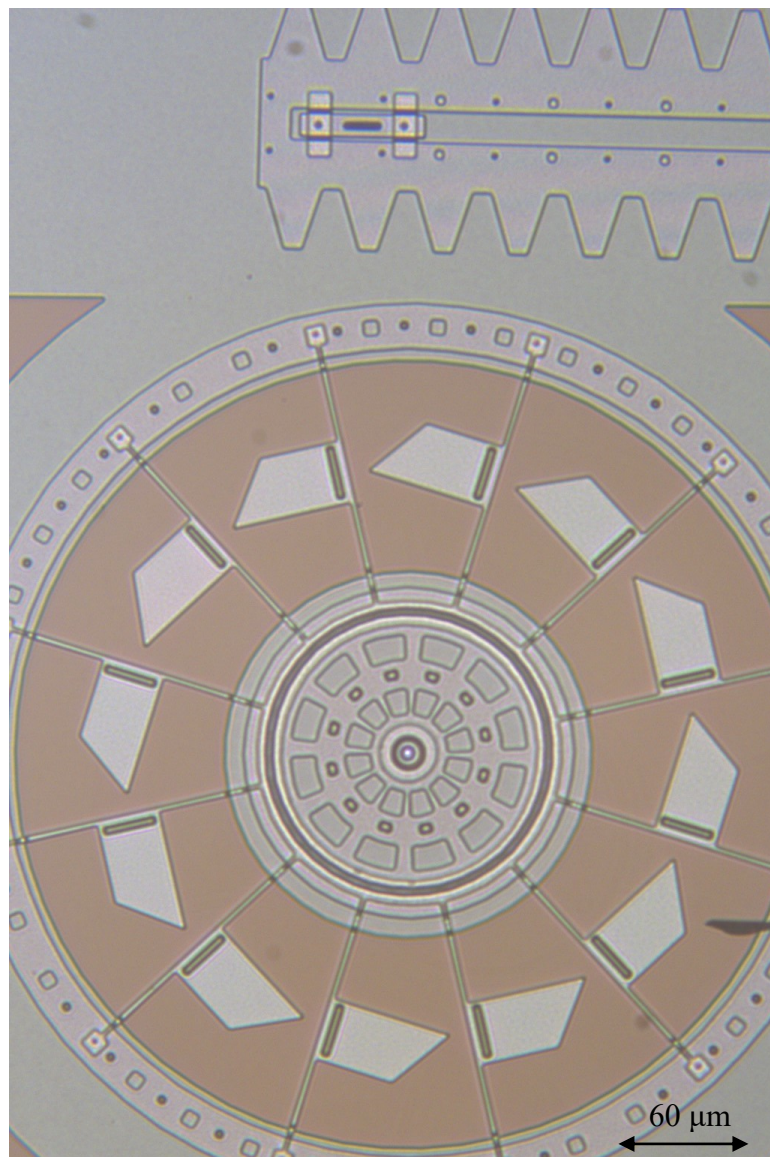


Figure 37 A 2D image of MEMS bearing with an electrostatic motor along with a gear rack (chip = IMUDTRM1).

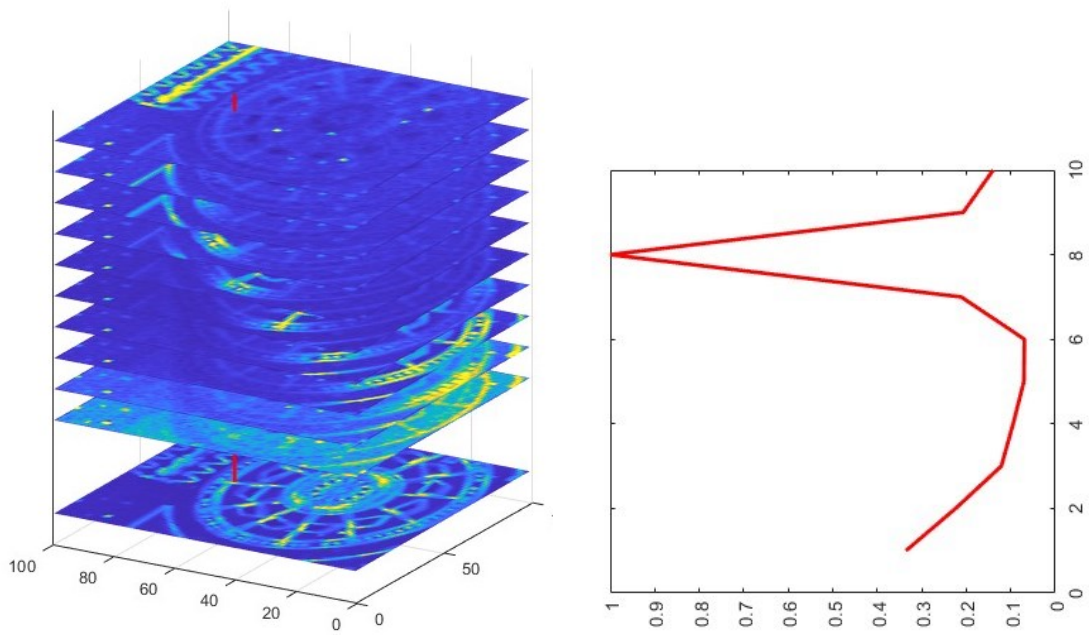
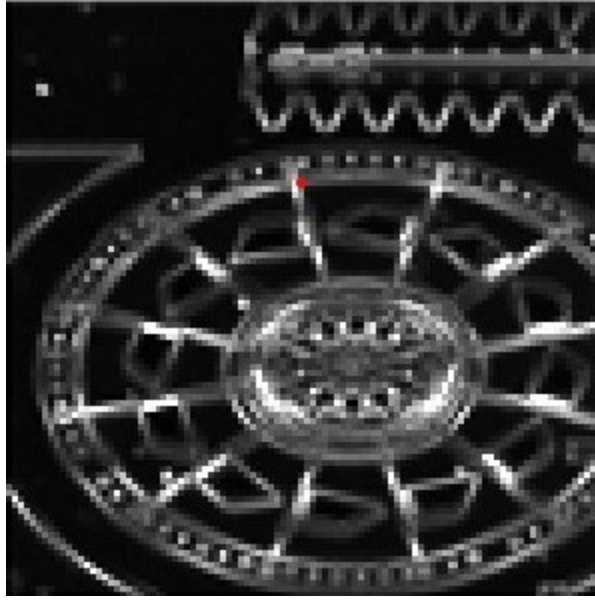


Figure 38 Finding Focus peaks. Top: sample red pixel on grayscale focus image. Bottom left: drilling through focus stack (false colour). Bottom right: focus curve for sample pixel.

By performing the drilling process at each pixel, we can generate an image known as a peak map (see Figure 39). The peak map captures the highest focus achieved per pixel during the drilling process and visualizes it within an image. Pixels that exhibit greater focus compared to others appear in shades of yellow, indicating sharper and more pronounced features within the image. In contrast, blue pixels represent lower focus values in comparison, signifying relatively less emphasis or prominence.

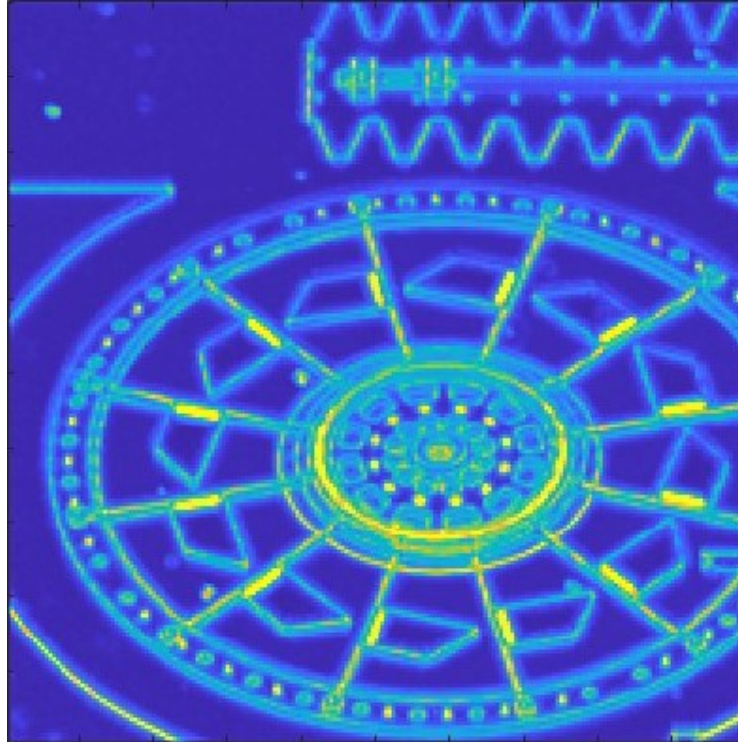


Figure 39 Focus peak map: maximum focus across all images. False colour: yellow is a high focus peak, and dark blue is low.

However, there is an issue with the background areas devoid of features (dark blue in Figure 39). Figure 40 shows a sample pixel chosen from the background with a low peak focus. In the absence of a distinct peak in the focus curve (Bottom right), the selected image could potentially be from any location within the image stack. As a result, when drilling through the focus stack flat areas have no distinct peak, which will have to be addressed later.

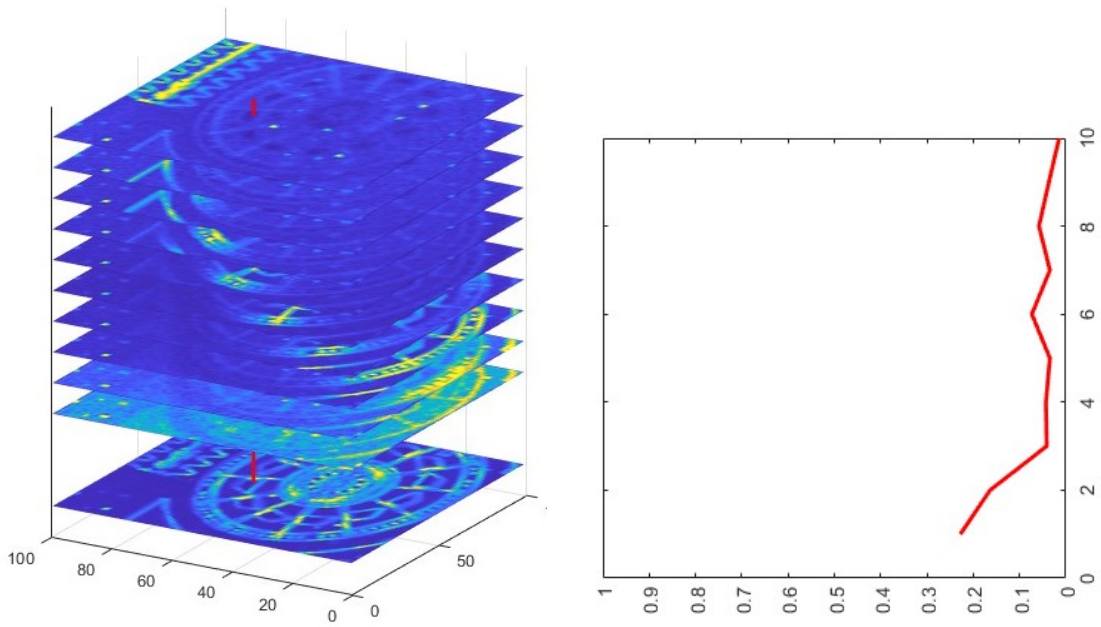
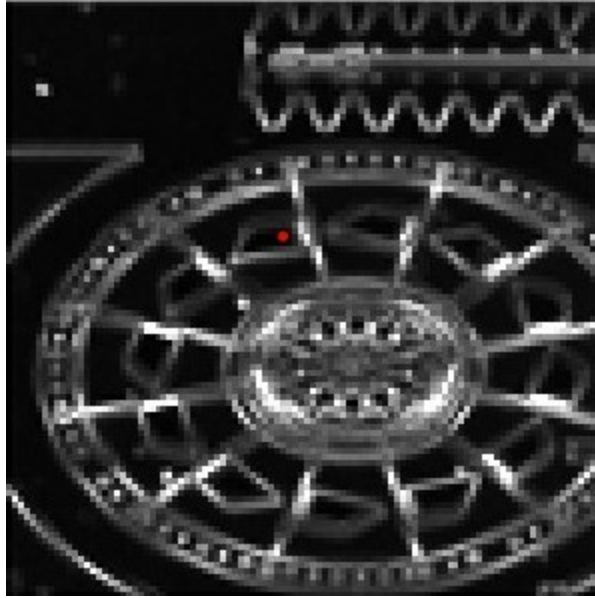


Figure 40 Top: background sample red pixel on grayscale focus image. Bottom left: drilling through focus stack (false colour). Bottom right: peak-less focus curve for sample pixel.

The focus peak map can be used to generate the depth map. At each pixel, we record not only the peak focus but also which image produced that peak focus. In Matlab, the focus peaks and depth map can be simultaneously generated using the function:

$$[\text{focus\_peaks}, \text{depth\_map}] = \text{max}(\text{focus\_stack}, [], 3); \quad \text{Eq. 4}$$

which records both the focus peak values and location (depth map) of the maximum of the focus stack. Note that the depth map and peak map fulfill separate roles. Whereas the peak map contains focus values (decimals  $0 \rightarrow 1$ ), conversely the depth map contains the image number (integers  $1 \rightarrow N$ ) associated with the highest focus value attained.

As an example of the process to generate a depth map Figure 41 shows the 2D overhead view of a MEMS grid actuator. Using Eq. 4 a false colour depth map is generated where each pixel's colour corresponds to the image number, see Figure 42. In this case, 45 perspective images were taken. A Matlab "jet" colormap was used: blue corresponds to the first image (1), green to mid images, and red to the last (45) image. As images are captured sequentially at a 45-degree angle, the higher parts of the image are further away. The observed colour progressive transition from blue to green to red is as expected and consistent, resulting in discernible bars. However, certain areas exhibit discrepancies that do not form clear bars but mottled multicoloured 'blobs'. These will be discussed later.



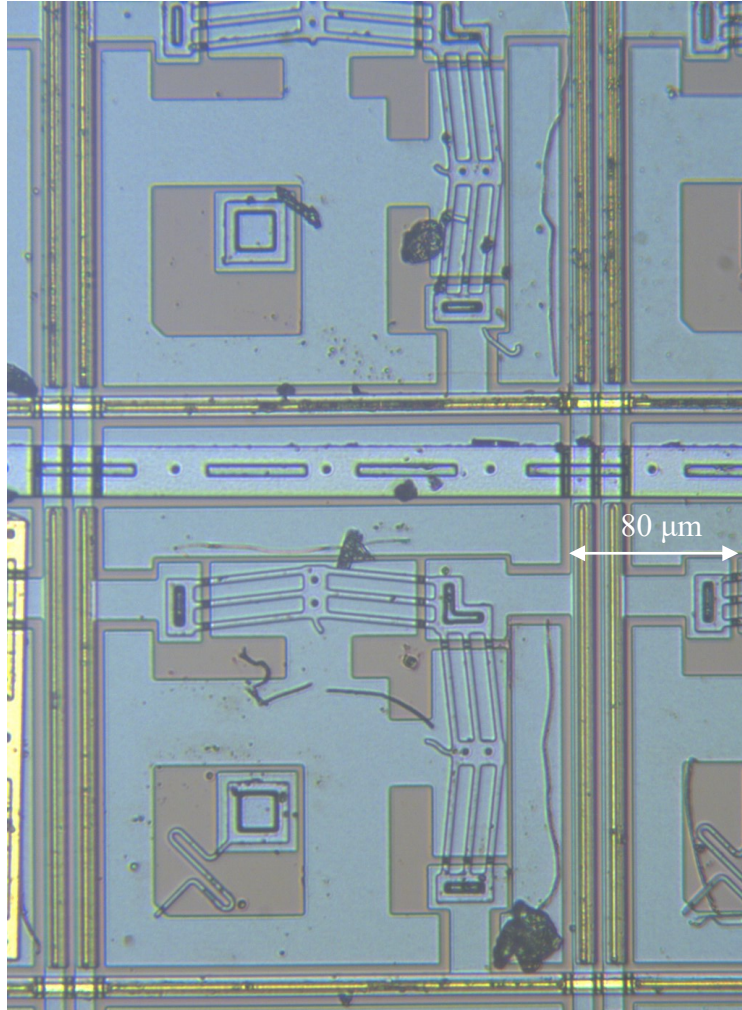


Figure 41 2D image of MEMS chip grid actuator (chip = IMUDTPH1).

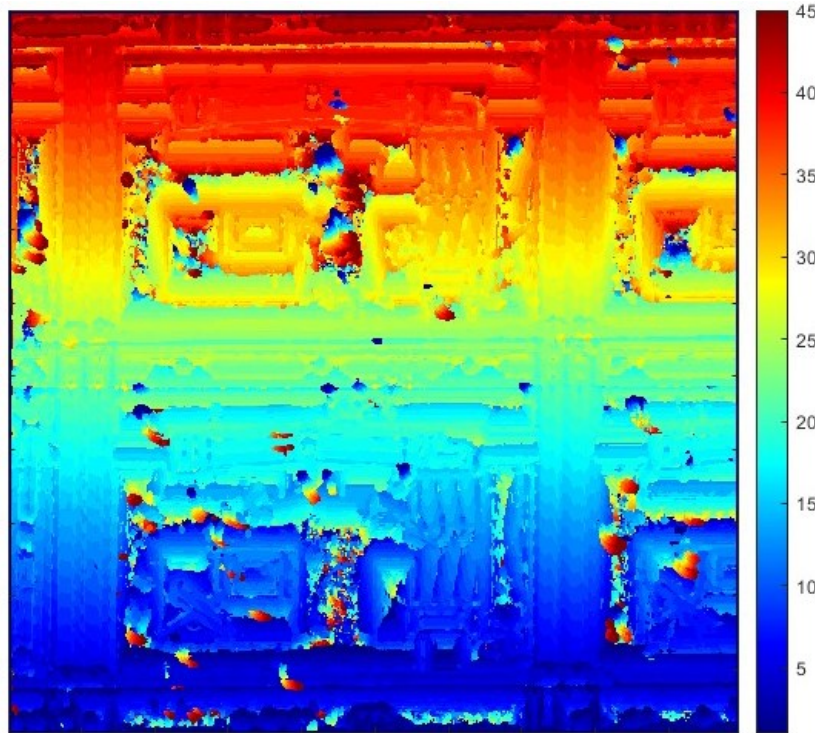


Figure 42 False colour depth map of the device from the previous Figure showing progressive colour bars, as well as multicoloured areas / 'blobs'.

#### 5.4. Image Fusion

To calculate the fused image, we start by initializing a blank image. Then, we use a *for* loop to iterate through all the images in the stack. For each image,  $i$ , we wish to keep only those pixels that are focus peaks and discard all others. A Boolean image called `depth_test` is created and filled with ones. Next, we erase `depth_test` pixels where the depth map is not  $i$ . To do this a Matlab logical masking operation is used:

```
depth_test (depth_map  $\neq$  i) = 0
```

(Note: in Matlab “`==`” is the “not equal” operator.)

This sets all non-peak `depth_test` pixels to 0. We then elementwise multiply (`.` `*`) the image  $i$  with the resulting `depth_test`, which acts as a mask that zeroes out the pixels that won't be used. The product is added to the blank image:

```
result_image = result_image + image.* depth_test
```

By iterating through the entire stack, we create an in-focus image consisting only of the peak focus pixels taken from the images specified in the depth map.

The above algorithm works for grayscale images but not for colour images. However, by simply using the "repmat" function, a 3-colour version of the resulting depth test is created. Again, we multiply the colour image with the colour version of the resulting depth test, which acts as a mask that zeroes out the pixels that won't be used.

When image fusion is performed by utilizing the maximum focus at each pixel (see Figure 43), an image is indeed formed. However, the overall quality is compromised, and parts of the background appear to be out of focus and excessively noisy, making the image inadequate. In Figure 44, a close-up of the background is presented. Where smooth bars are present in the depth map, the fused image is clear. However, in the multicoloured 'blobs' or patches in the depth map, the pixels are extracted haphazardly from different image numbers originating from various depths within the stack. Extracting pixels from different sections of the stack leads to a blend of both focused and unfocused areas, resulting in a mixed-focus outcome.

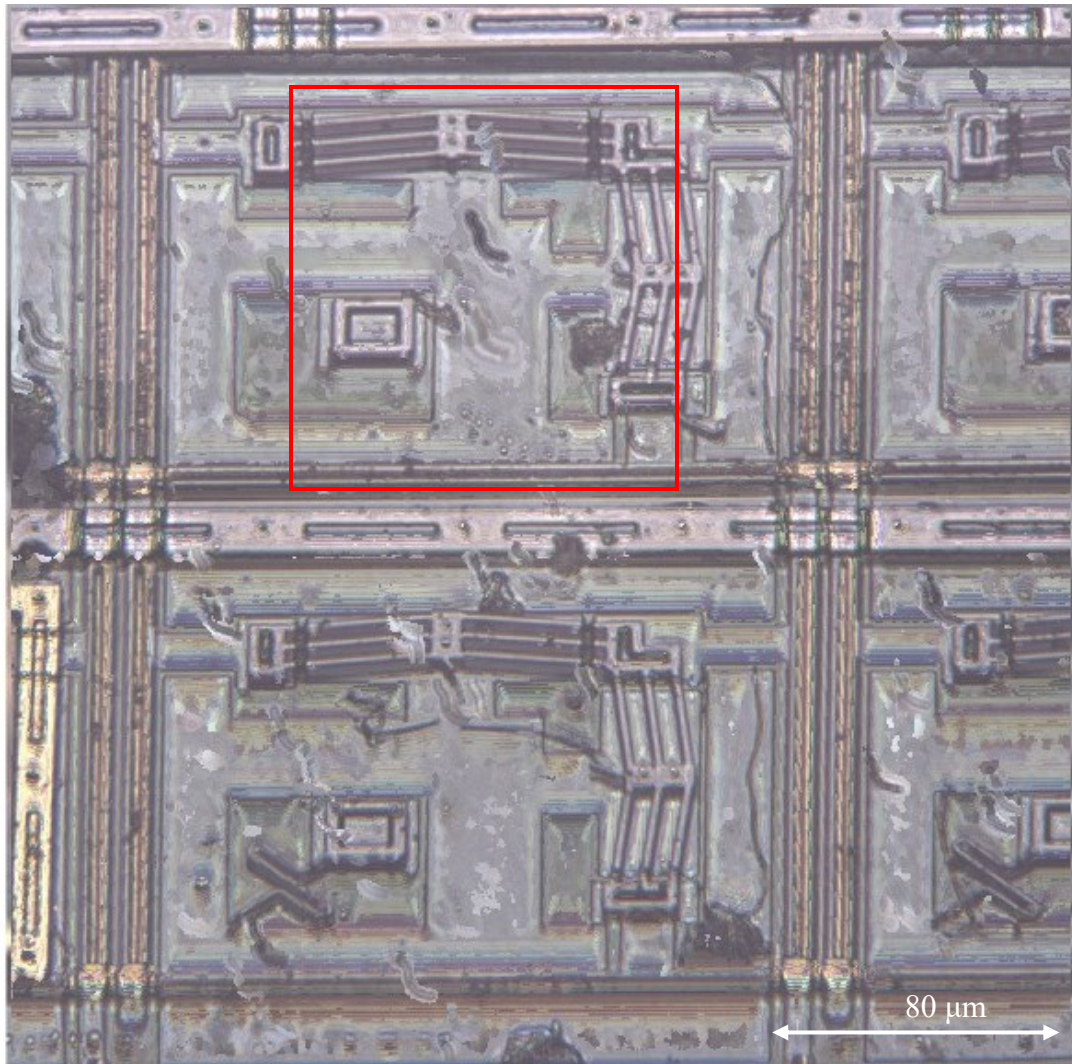


Figure 43 Result of image fusing, note poor image quality and artifacts in background areas. The red box is shown magnified in the next Figure.

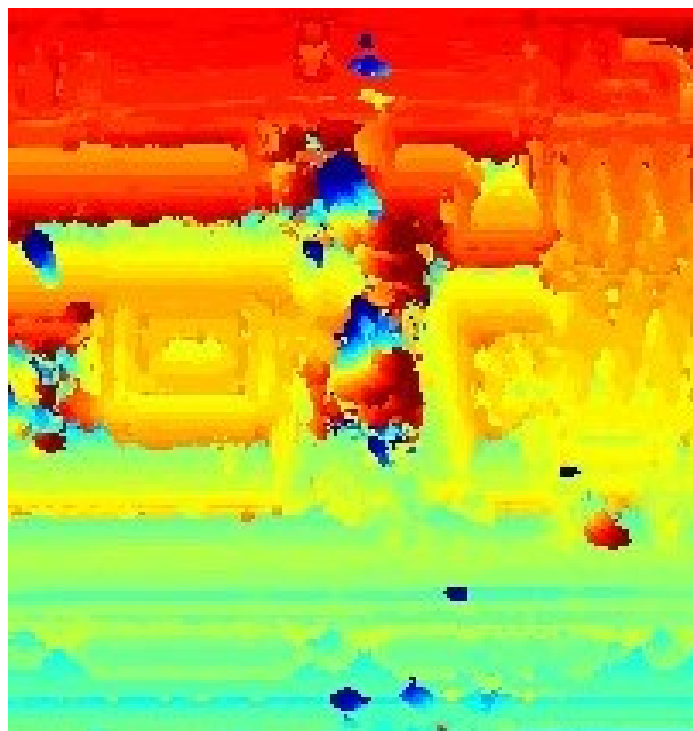
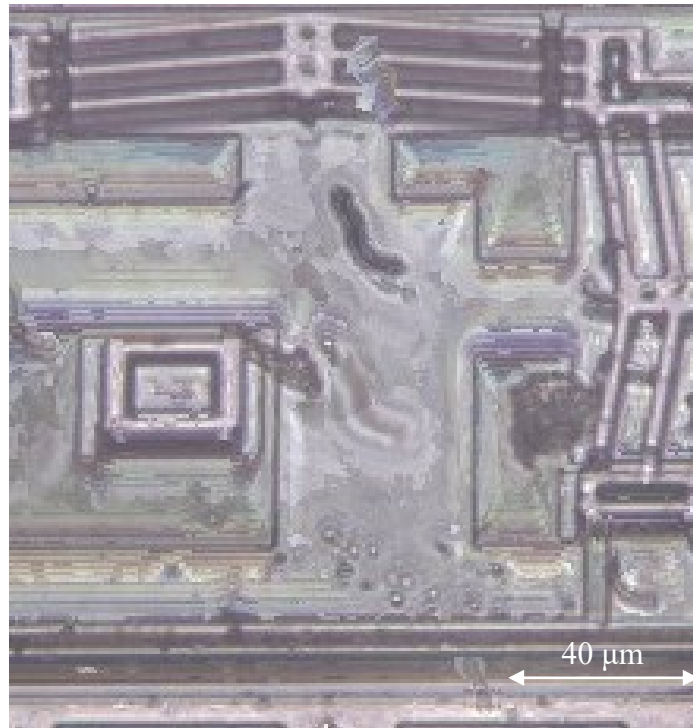


Figure 44 Zoomed-in view from the previous Figure. Top: Fused Image, Bottom: Depth map. Note that multicolour patches in the depth map lead to artifacts in the fused image.



Why do the background areas devoid of features exhibit such a high level of noise? Upon closer examination of one of these areas, it becomes evident that the focus never reaches a pronounced peak in comparison to the edges or features. Previously Figure 40 showed a sample pixel chosen from the background with a low peak focus. As stated earlier, in the absence of a distinct peak in the focus curve, the selected image could erroneously be chosen from an incorrect (and out of focus) depth location within the image stack.

### 5.5.Smoothing

The signal-to-noise ratio (SNR) is employed to assess the quality and clarity of an image. SNR is calculated by dividing the focus peaks by the mean focus values at each pixel ( $\text{max. drill value} / \text{average drill value}$ ). For example, Figure 45 shows the 2D overhead view of a portion of a MEMS chip of gears.

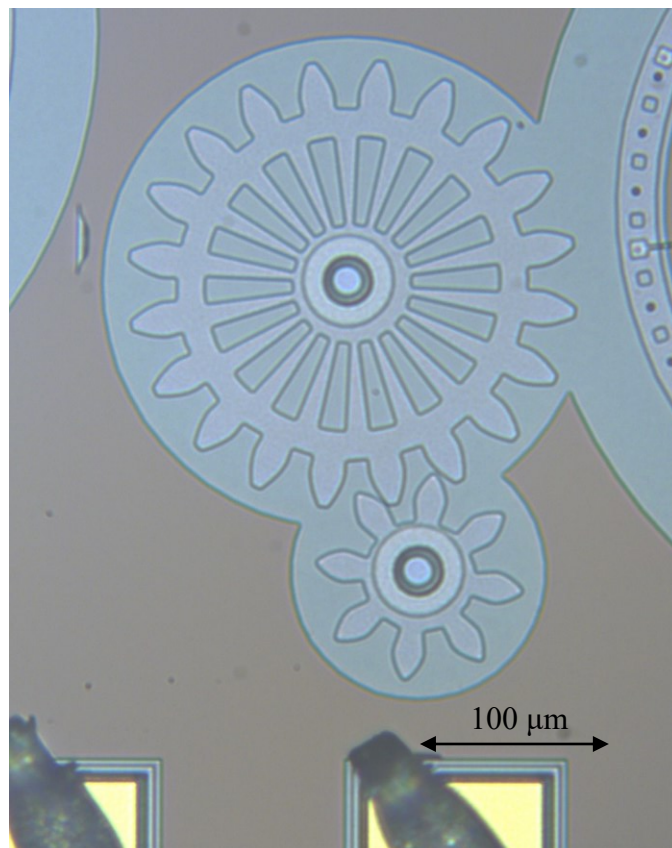


Figure 45 A 2D image of a MEMS chip spur gears (chip = IMUDTRM1).

Figure 46 top shows the peak focus image (0→1 range), while Figure 46 bottom shows the corresponding and similar SNR image (1→7+ range). When the SNR drops much below 2 (see colour bars), the noise becomes nearly indistinguishable, as the peak amplitude in that region closely resembles the noise level and thus the incorrect image number may be assigned to the depth map as shown previously in Figure 44. A higher SNR corresponds to a more prominent peak and a clearer and better-defined outcome.

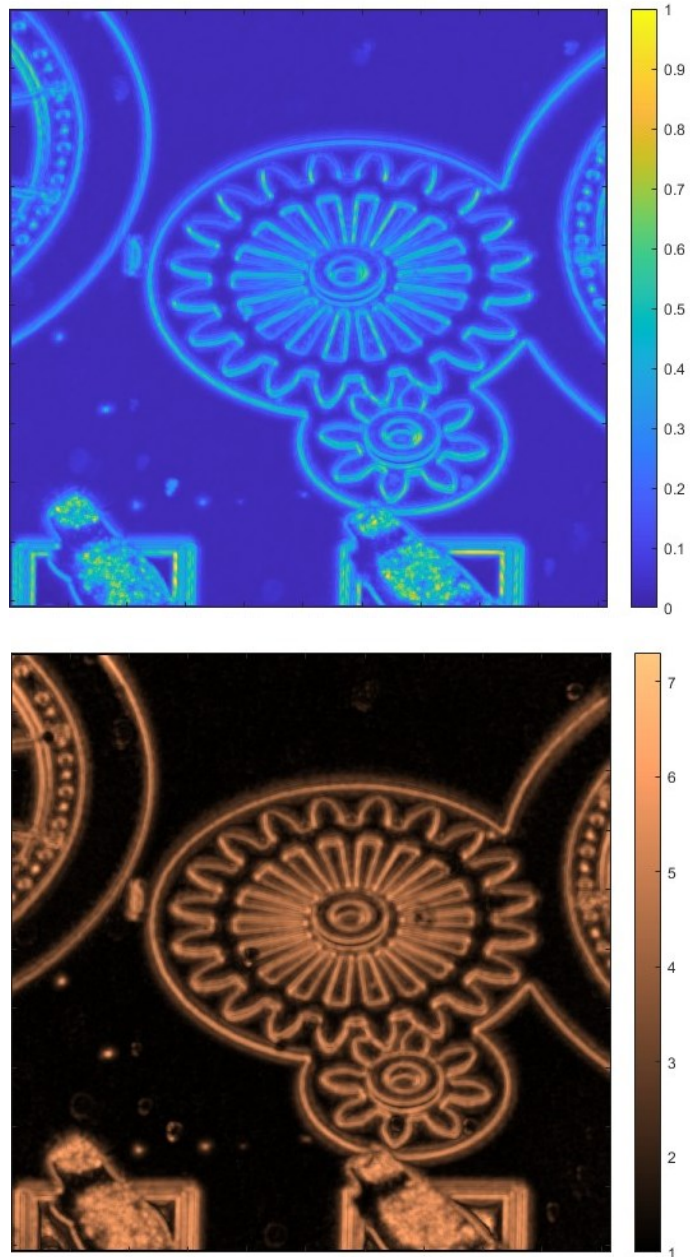


Figure 46 Top: Focus peaks in comparison to Bottom: focus SNR.

The goal of smoothing is to get a similar SNR across the image indicating clear focus peaks and thus a clearer image. Smoothing the image helps us observe the true peaks. This is achieved by averaging the noise and identifying the peak. This is necessary because featureless areas lack strong points of interest. Without a prominent peak, a random image with the ‘highest’ focus value is chosen, but it could in fact come from an arbitrary depth within the stack. This leads to images with artifacts and a lack of clarity. Images with extensive flat areas require more smoothing compared to images with many features.

A Gaussian filter smooths images by computing a weighted average of adjacent pixels, consequently minimizing the impact of noise on the pixels relative to the focus peak. The smoothing area was chosen to be a rectangle, as opposed to a square. A 2:1 aspect ratio rectangle was selected because we observe consistent focus bars in the horizontal direction, rather than the vertical direction. By capturing more focus pixels horizontally, we are examining focus pixels with a similar focus. The size of the rectangular area where smoothing is applied is user-defined: this will vary for each image depending on the features of the image.

Figure 47 displays the resultant 3D image of the device previously shown in Figure 27 with different levels of smoothing. Note that these images are zoomed in to show details. As the level of smoothing is increased, you can observe that the depth map becomes more consistent, with fewer spots or ‘blobs’. Additionally, as the smoothing levels increase, fewer features are distinguishable in the depth map. This is due to the peaks being lowered and the low areas being raised because of the smoothing process.

Figure 48 shows the top left corner of Figure 47 at a higher level of zoom. Note that without smoothing there are irregular patches in the flat surfaces that progressively disappear with increased smoothing.



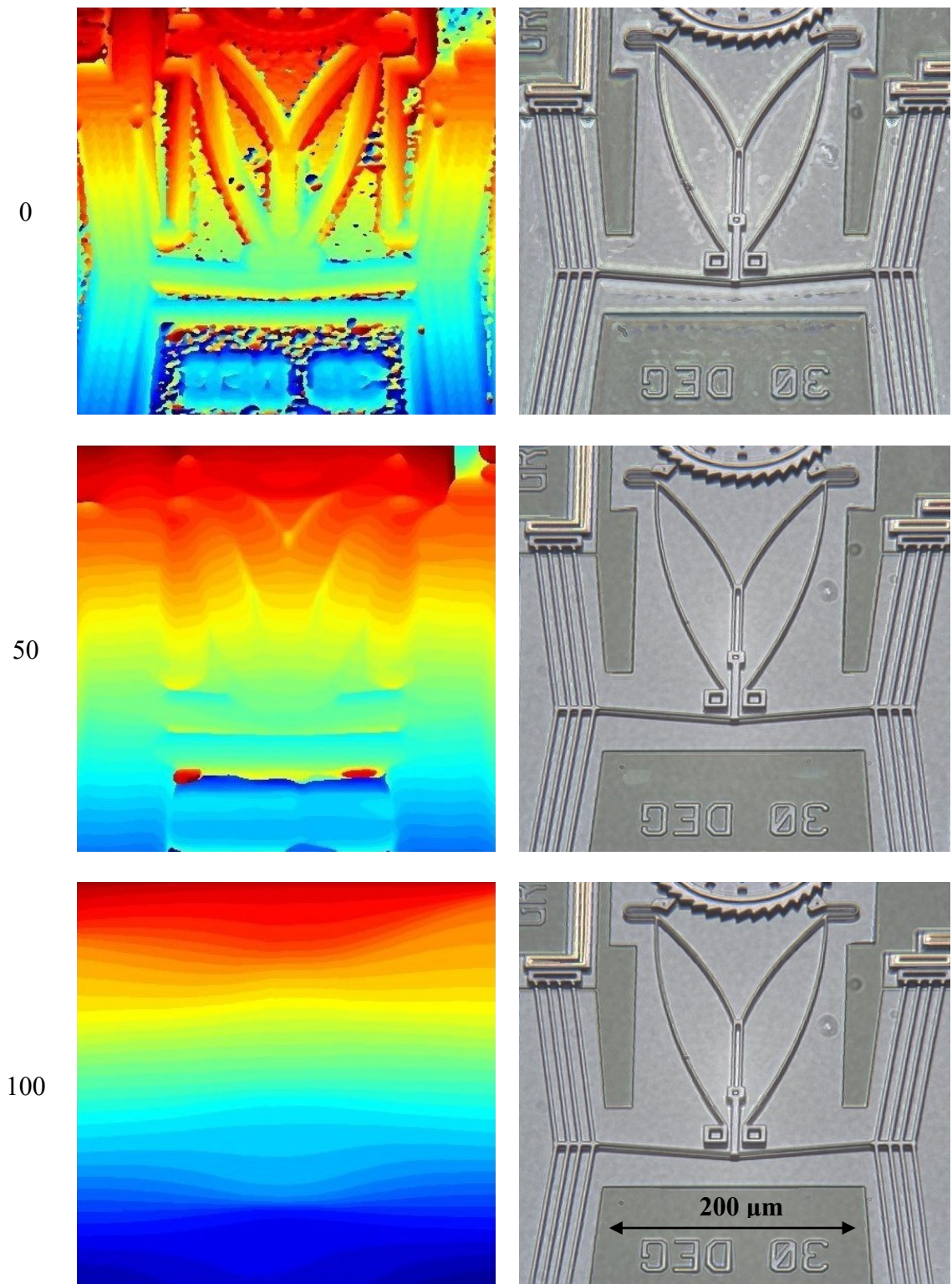


Figure 47 Depth maps (left images) at different smoothing levels (numbers at far left) with corresponding fused images (right images).

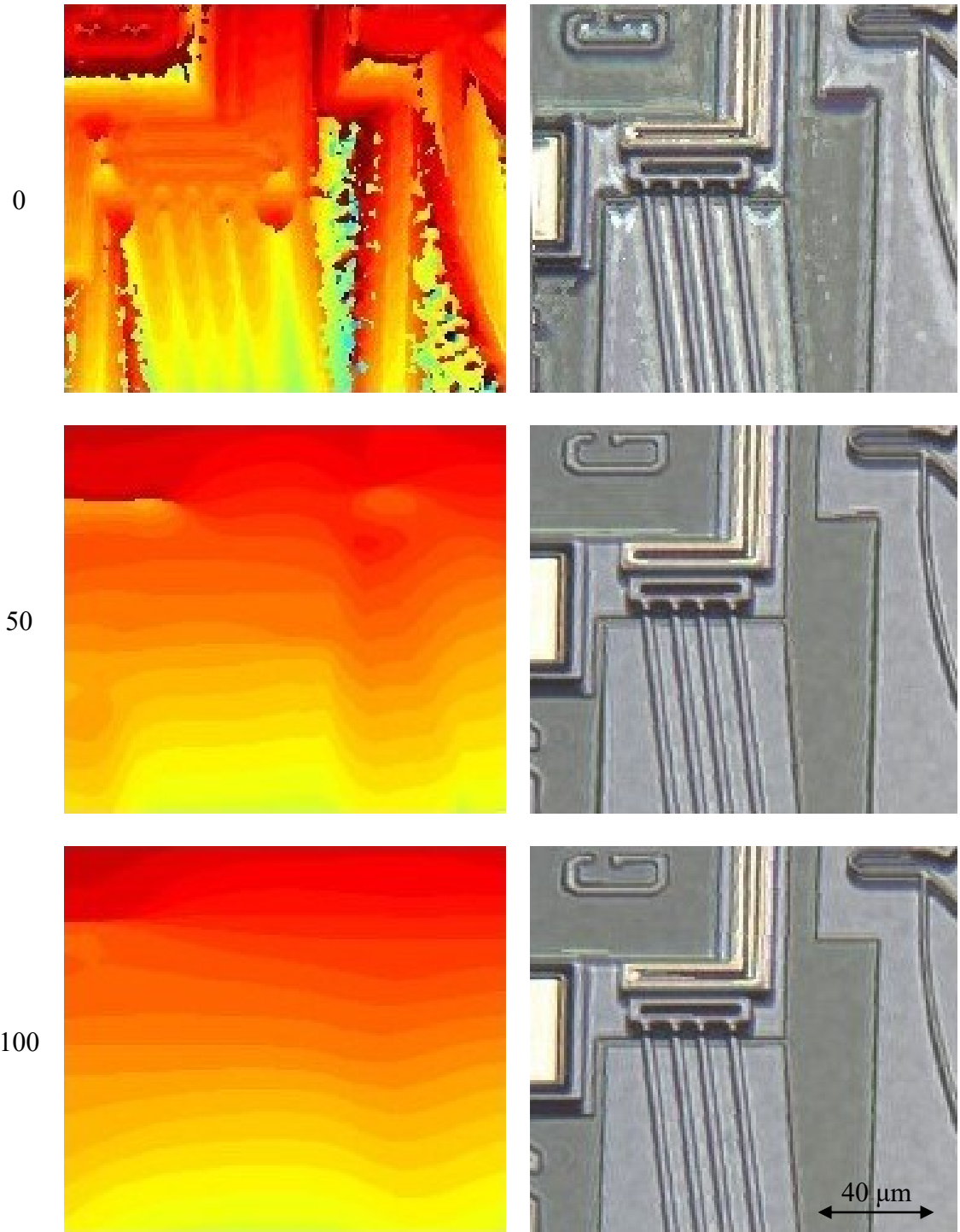


Figure 48 Zoomed-in top left corner of previous images at different smoothing levels.

## 6. Optimization

### 6.1. Different Focus Measure Algorithms

There are a variety of focus measures offered in the focusmeasure package we used [25]. Typically, focus measure algorithms excel in specific uses, meaning not all of them will deliver satisfactory outcomes. We experimented with a variety of focus measure algorithms including Brenner's (BREN) [7, 6], Gaussian derivative (GDER), Modified Gray level variance (GLVM), Threshold Gradient (GRAT), 3D Laplacian (GRA3) and Hemli's mean method (HELM) [26].

Looking at the highly smoothed previous images, the focus bars are approximately horizontal, indicating that the horizontal portions are in the same focus. As a diagnostic tool only, we consider an extreme form of smoothing where all depth pixels in a row are averaged. Row smoothing is not a good smoothing algorithm to use in practice as it introduces large horizontal image artifacts called 'tears' or 'rips'. It averages pixels from opposite sides of the long row and introduces errors.

The advantage of row smoothing from a diagnostic viewpoint is that the 2D depth maps are converted to 1D 'staircase' graphs (see Figure 49). The staircase graphs show the corresponding image number with the highest focus across the entire row, creating a visual representation of how focus changes vertically across the image stack.

The uniformity of the staircase may vary it will not be a perfect staircase. However, the staircase graph should resemble a regular staircase if the focus measure performs well. In certain images, protruding features can lead to small spikes or dips in the staircase graph, thereby affecting the calculated average focus value. Typically, everything within the focus bar area is expected to be in focus; however, large protruding elements indicate an issue with the focus algorithm.

The following sequence of Figures (Figure 49, Figure 50, Figure 51, Figure 52, Figure 53, and Figure 54) illustrate how various focus measurement algorithms influence the resultant fused image. A discussion follows the Figures.

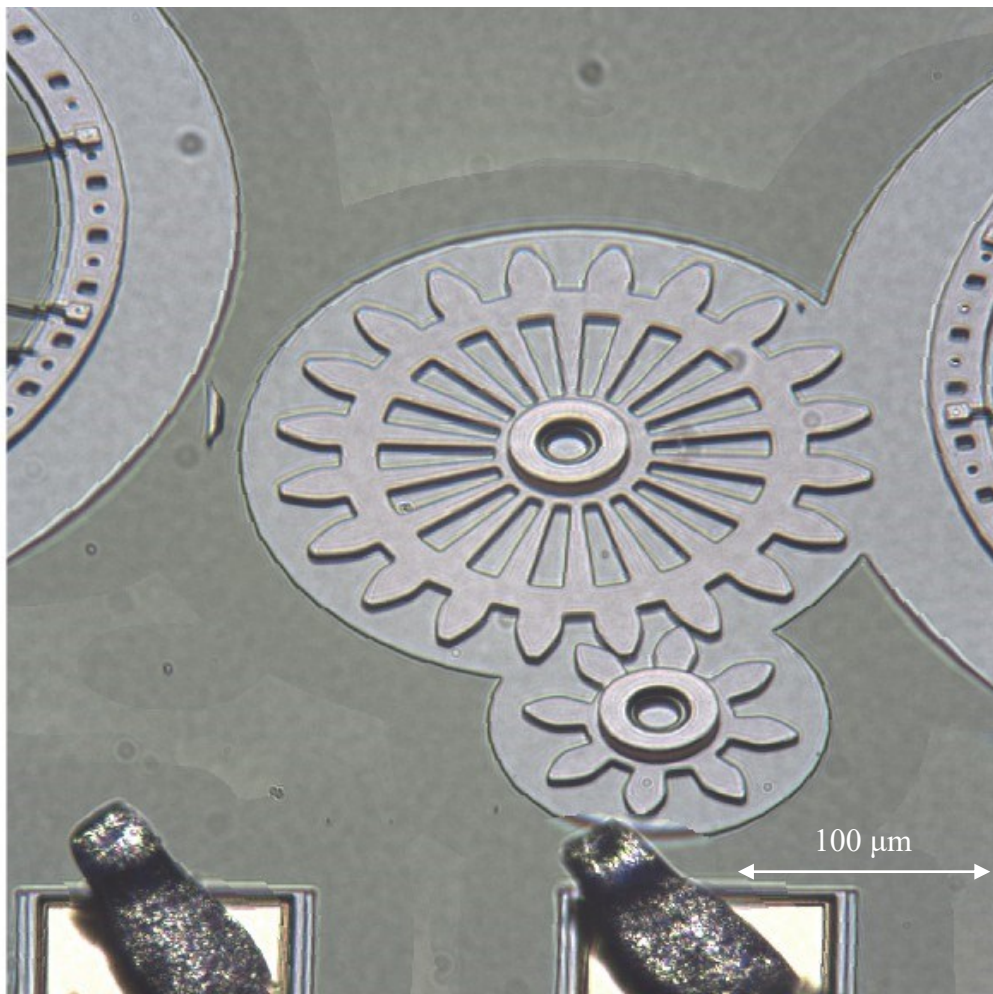
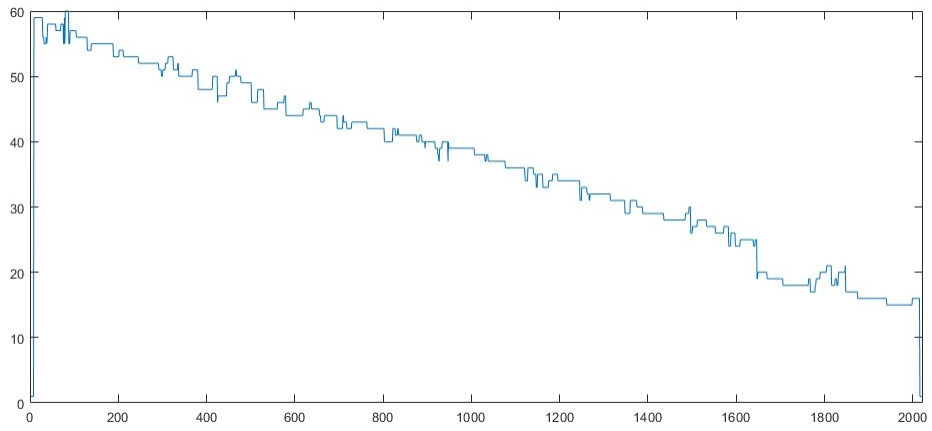


Figure 49 BREN focusing algorithm: Top = staircase graph and Bottom = image with 100 smoothing.



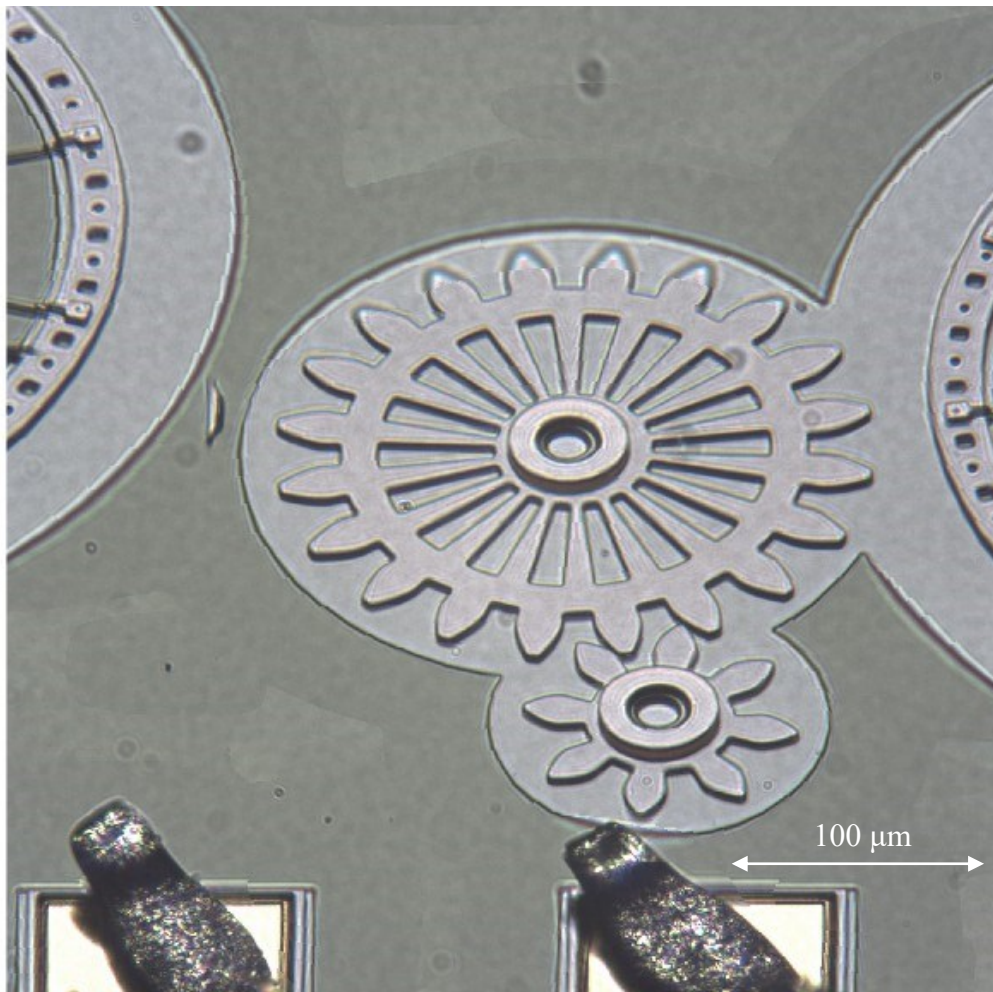
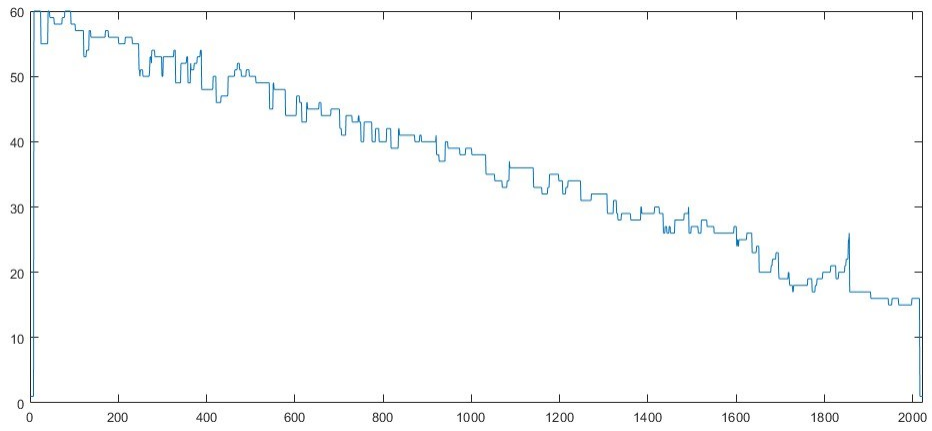


Figure 50 GDER focusing algorithm: Top = staircase graph and Bottom = image with 100 smoothing.

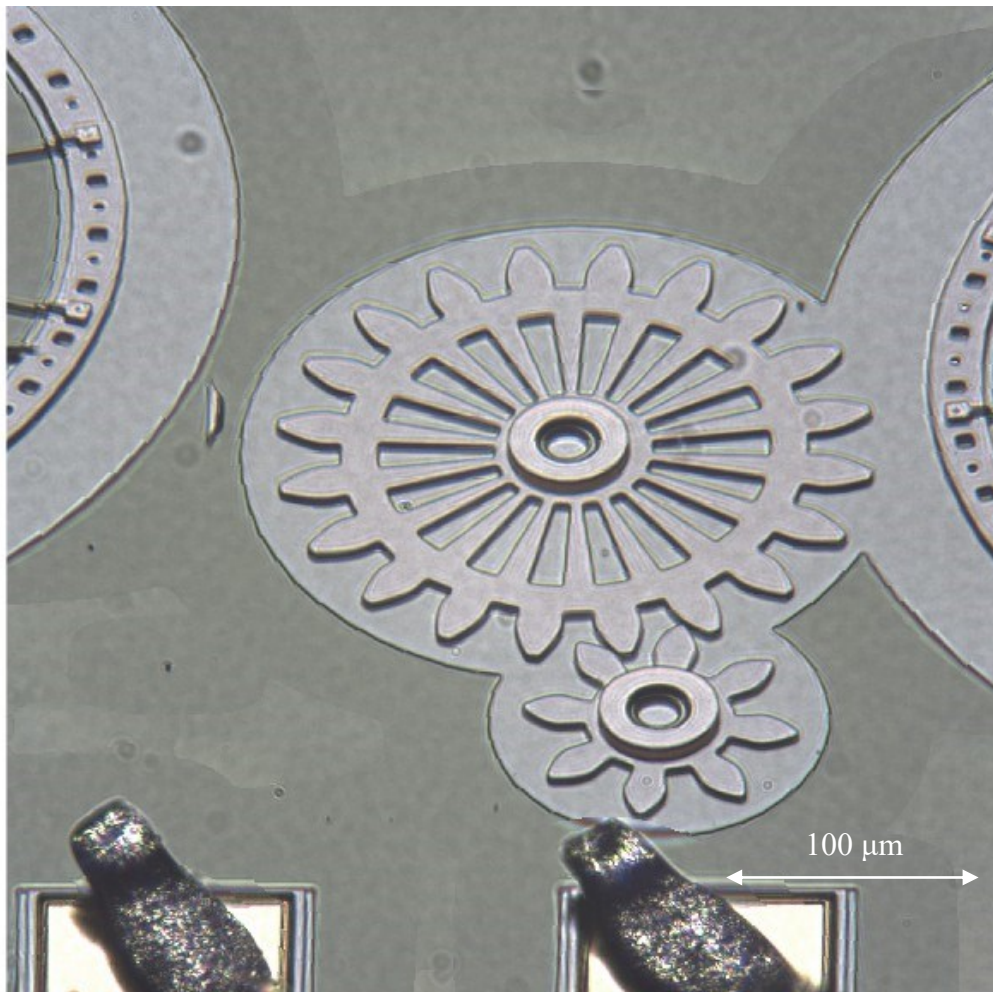
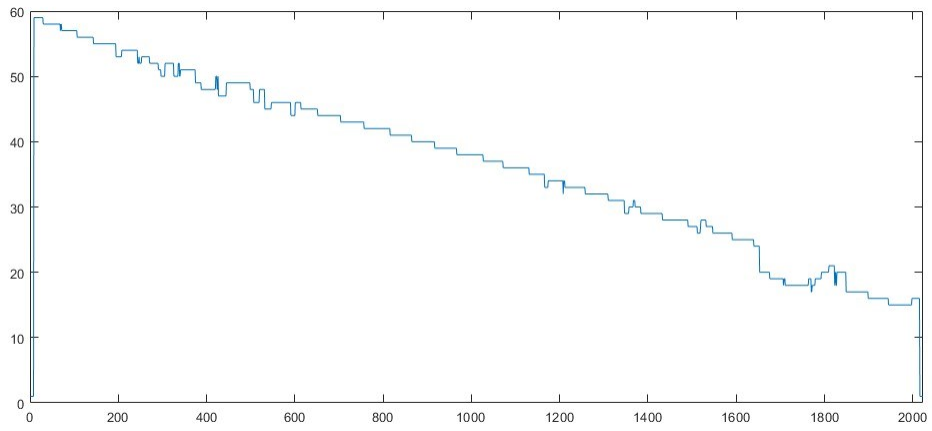


Figure 51 GLVM focusing algorithm: Top = staircase graph and Bottom = image with 100 smoothing.

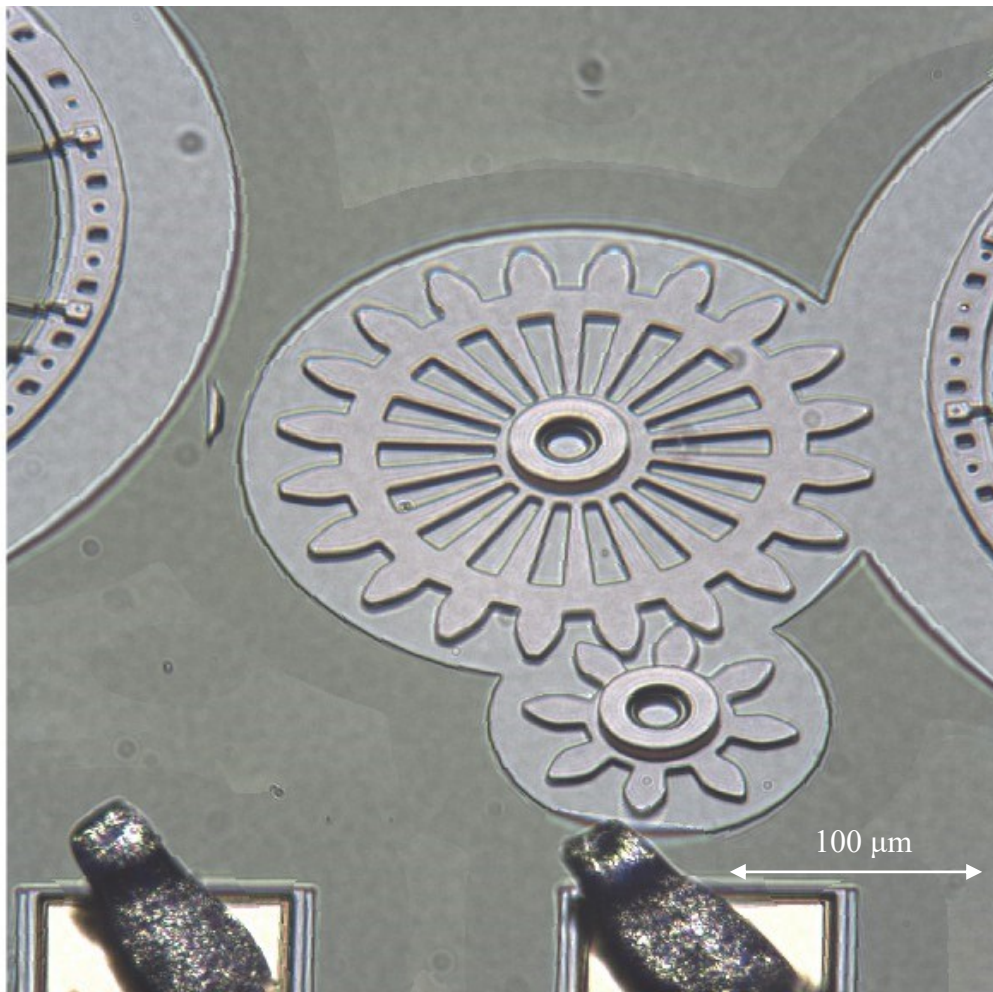
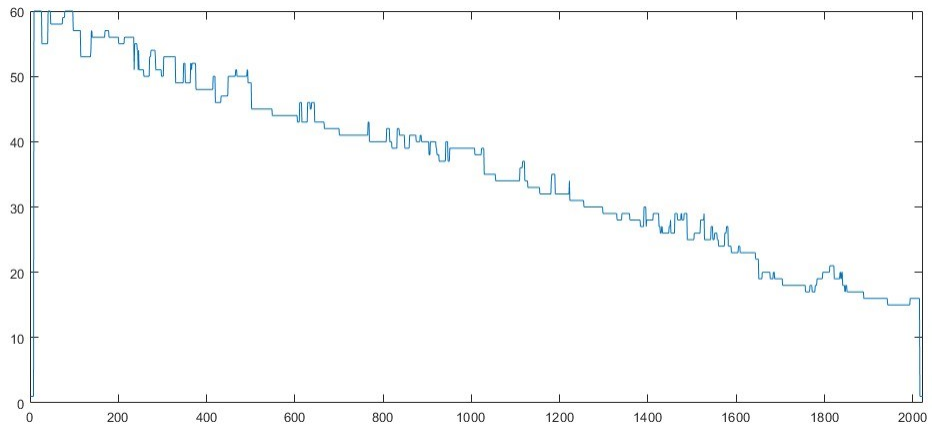


Figure 52 GRA3 focusing algorithm: Top = staircase graph and Bottom = image with 100 smoothing.



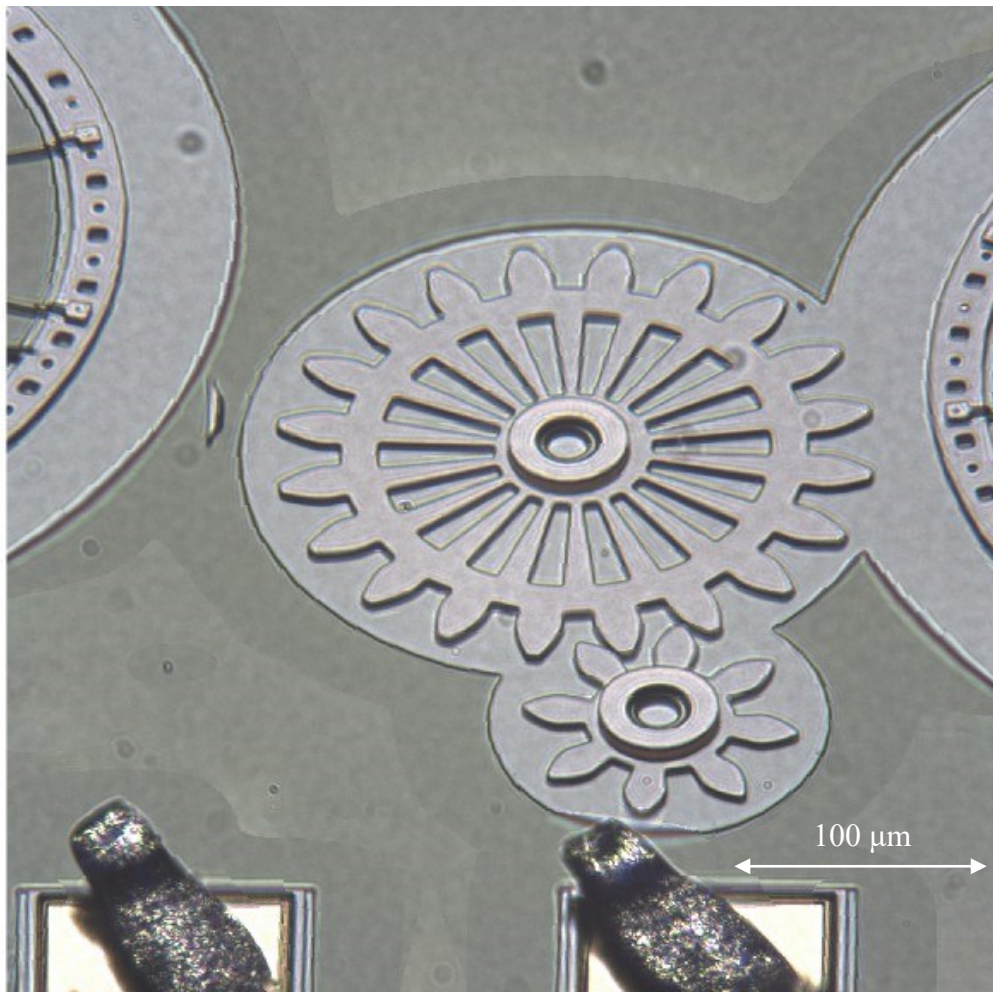
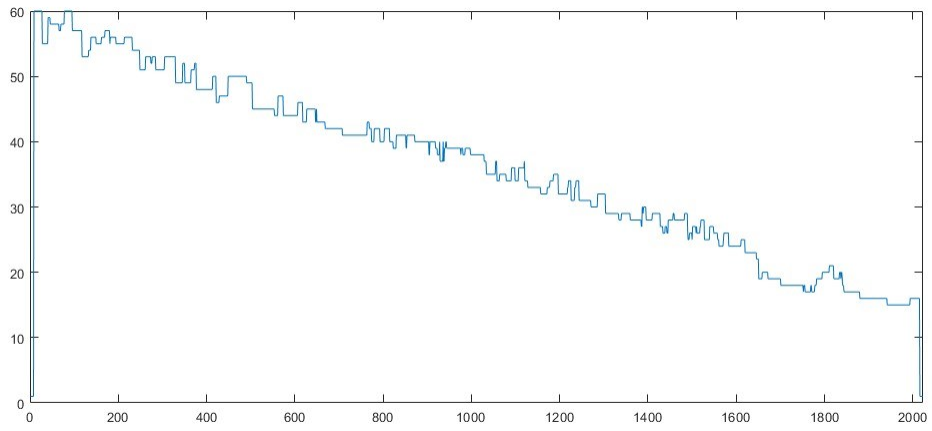


Figure 53 GRAT focusing algorithm: Top = staircase graph and Bottom = image with 100 smoothing.



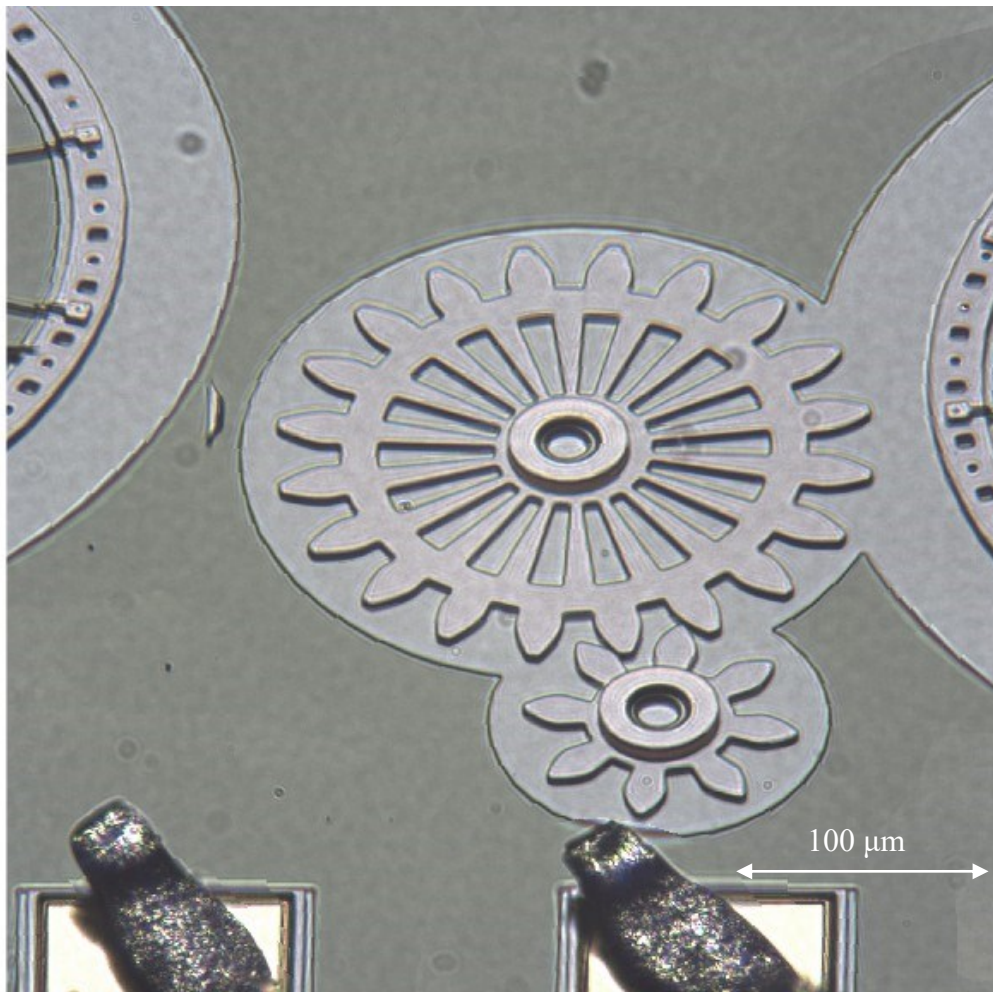
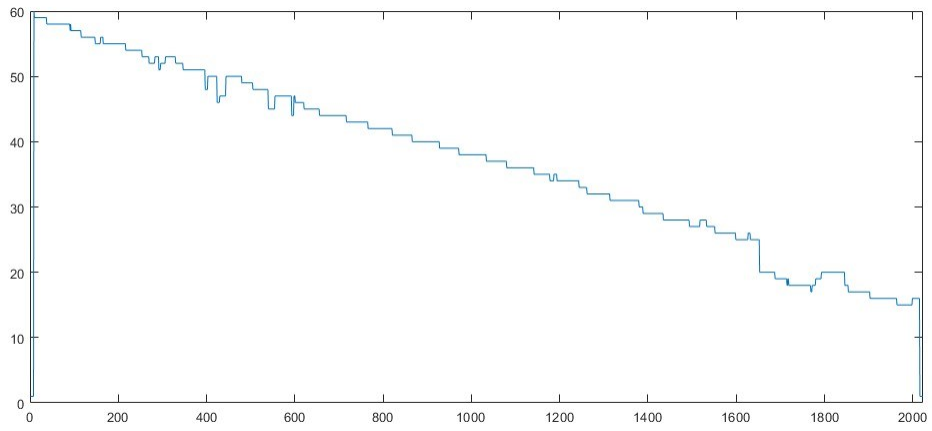


Figure 54 HELM focusing algorithm: Top = staircase graph and Bottom = image with 100 smoothing.

To compare the different models we examine 3 commonly used models: BREN, GDER & HELM. For each method, we examine the pixel to pixel variation. In an ideal staircase, there would be a flat step separated by the occasional jumps.

Table 3 displays the pixel to pixel vertical displacement for the three different focus measure algorithms. These displacements represent the amount of noise experienced in the graph. If the graph isn't comparable to a staircase, the sum values will be substantially higher, indicating a significant amount of noise within the focus algorithm. Upon reviewing the table, it becomes clear that HELM is the best option, as it has the lowest values for both the absolute sum and the squared sum. This suggests that HELM is the most similar to a staircase in terms of vertical displacement.

	$\sum Z_i-Z_{i-1} $	$\sum(Z_i-Z_{i-1})^2$
BREN	209	453
GDER	290	762
HELM	92	162

Table 3 Vertical noise analysis for 3 focus measure algorithms.

Each focus measure handles various aspects of the MEMS chip differently. Figure 55 gives a side-by-side comparison of the transition between a foreground wire and the MEMS background. In this case, there is a large jump in the focus position above the wire. Note how BREN has a very large area that is out of focus compared to others while, GDER performs the best in this regard, with minimal portions of the image out of focus, and the second-best is HELM. The rest of them appear blurry.

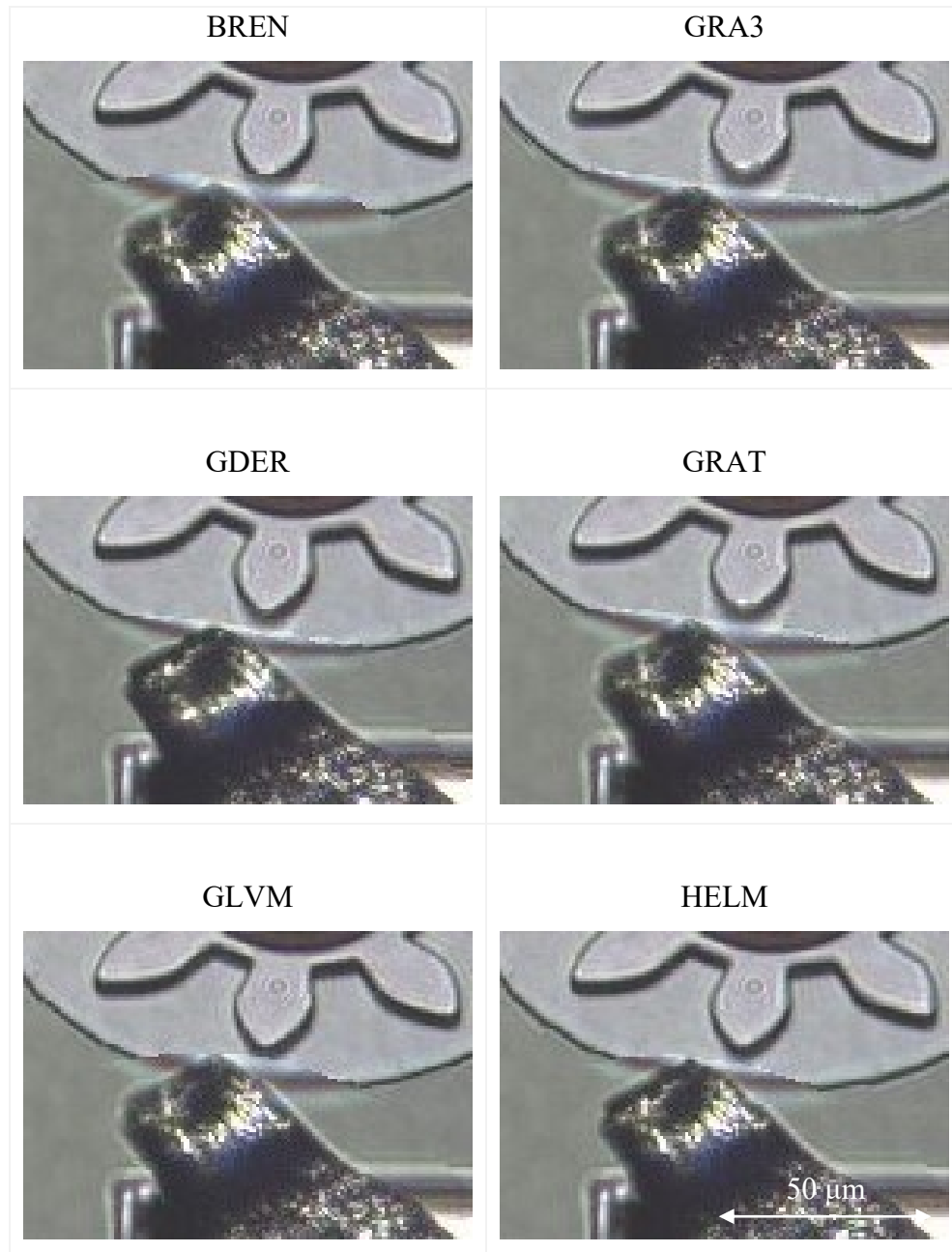


Figure 55 Comparisons of wire to MEMS transition between different focus measures at 100 smoothing.

In Figure 56, a side-by-side comparison of fine background details reveals that most methods perform poorly in maintaining focus with abrupt 'contour' lines. However, HELM stands out as an exception, demonstrating consistent focus and preserving details effectively, unlike the others. Notably, GDER fares quite poorly in this regard, with a significant portion appearing out of focus.

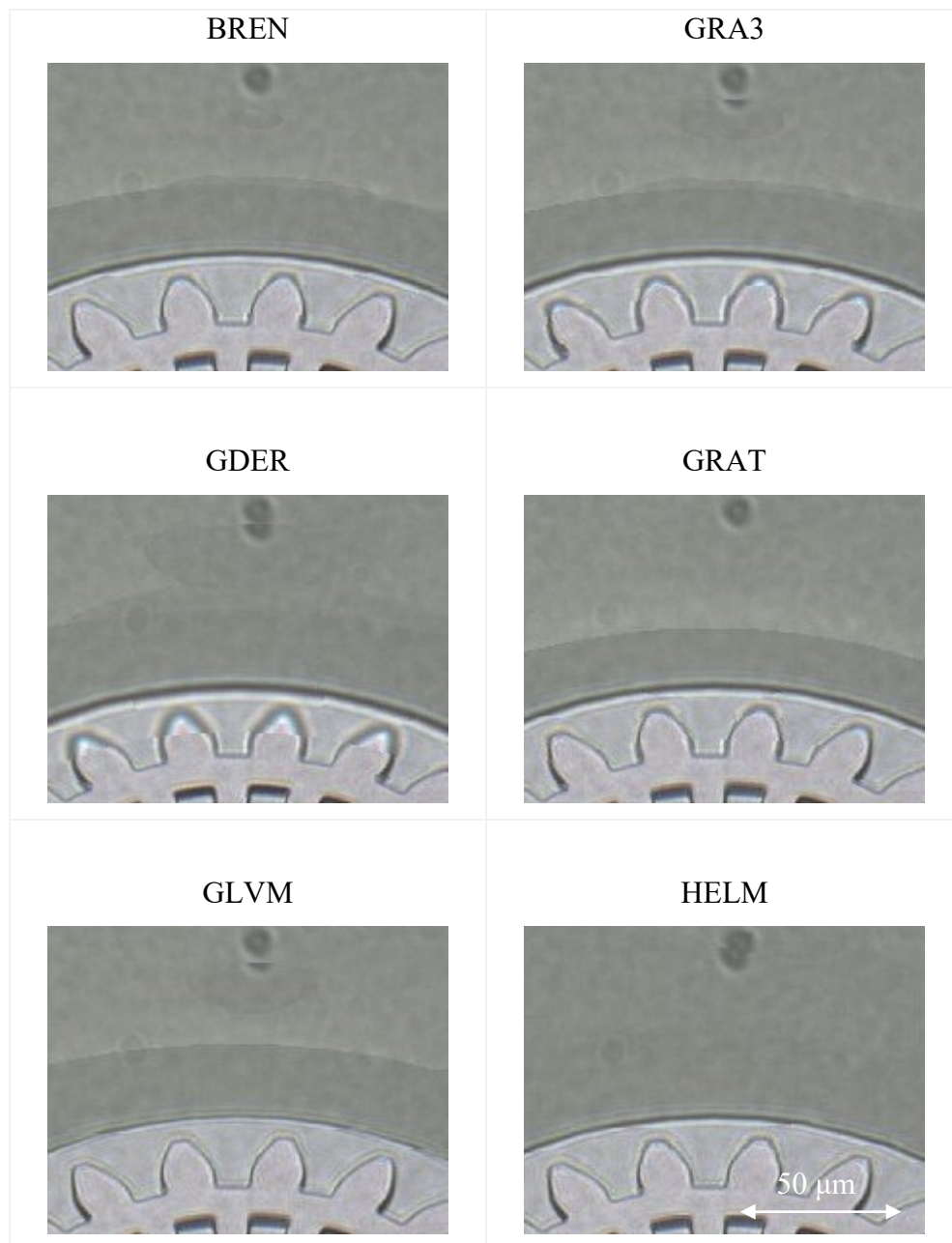


Figure 56 Comparisons of background detail between different focus measures at 100 smoothing.

When selecting the best focus measure method, it must effectively maintain focus in various scenarios, including raised features and flat areas. Evaluating all the methods collectively, it becomes evident that HELM performs the best among them. It excels in capturing fine background details and maintains relatively good performance when there is a transition in features between the foreground and background.

## 6.2.Guidelines for smoothing

Excessive smoothing can lead to problems in the final image. Without any smoothing applied, you'll observe a distribution of high peaks along the edges and numerous low peaks in the background areas. When smoothing is introduced, the high peaks are diminished, and the lower areas are elevated. However, if too much smoothing is applied, the peaks may be reduced to a point where they adversely affect the image quality. This can result in pixels being taken from the wrong regions, compromising the image.

Figure 57 shows a 2D view of a MEMS electrostatic actuator. Figure 58 shows the resultant 3D images of a zoomed-in region near the top. Too much smoothing (see Figure 58 Right) can cause portions of the image to blur and introduce jagged breaks in lines. This becomes evident when examining the wire, which is no longer straight. Furthermore, an increase in smoothing leads to a loss of shadows and features. This effect is noticeable in the anchors, as they appear three-dimensional when smoothing is set at 100, but they become flat as smoothing is increased.

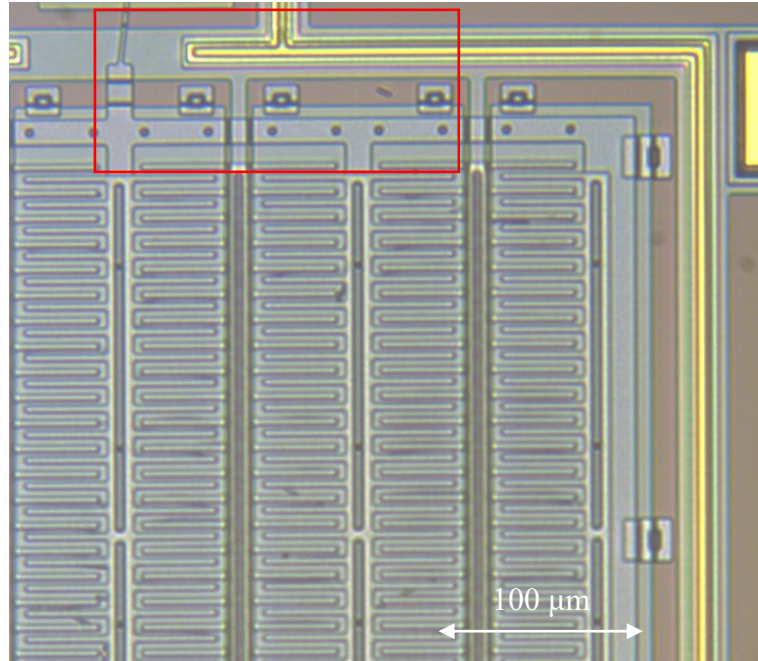


Figure 57 2D image of a MEMS electrostatic actuator (chip = IMUDTRM1).

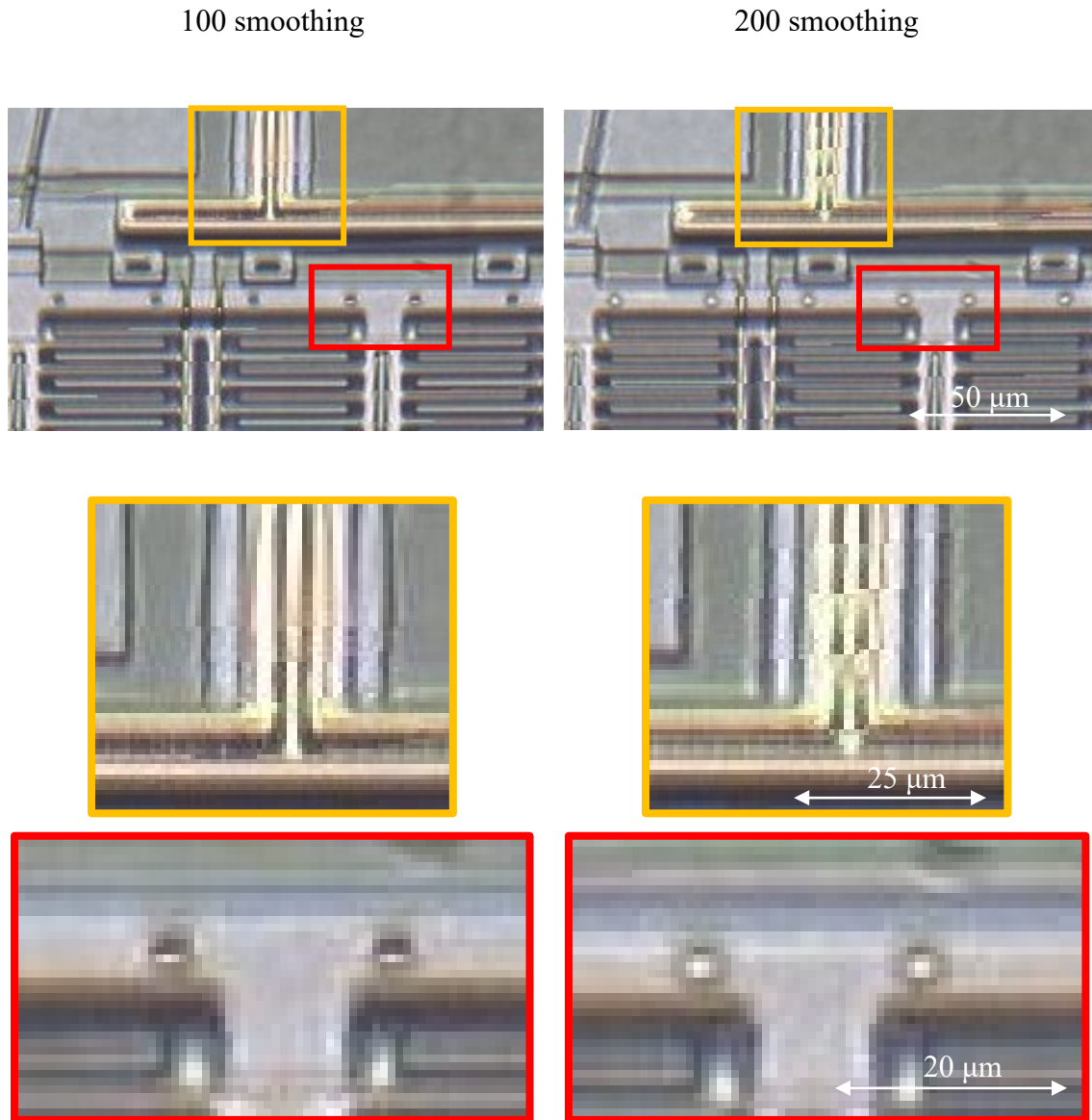


Figure 58 Zoom-in of the previous Figure. Effect of excess smoothing.

In Figure 59, the further impact of excessive smoothing (1000) is evident. By increasing the smoothing, the transition between pixels becomes less smooth, as can be seen in the depth map. Additionally, areas in the resultant image appear choppy and out of focus, lacking subtle changes.

The amount of smoothing required for an image fluctuates greatly, making it difficult to determine precisely. However, a rough estimate can be obtained based on the features present in the image. If there is a significant amount of background and only a



few features, a substantial amount of smoothing will be necessary, as numerous blobs will appear in the image. Conversely, if there are many features, roughly 100 units of smoothing will suffice to create a focused image.

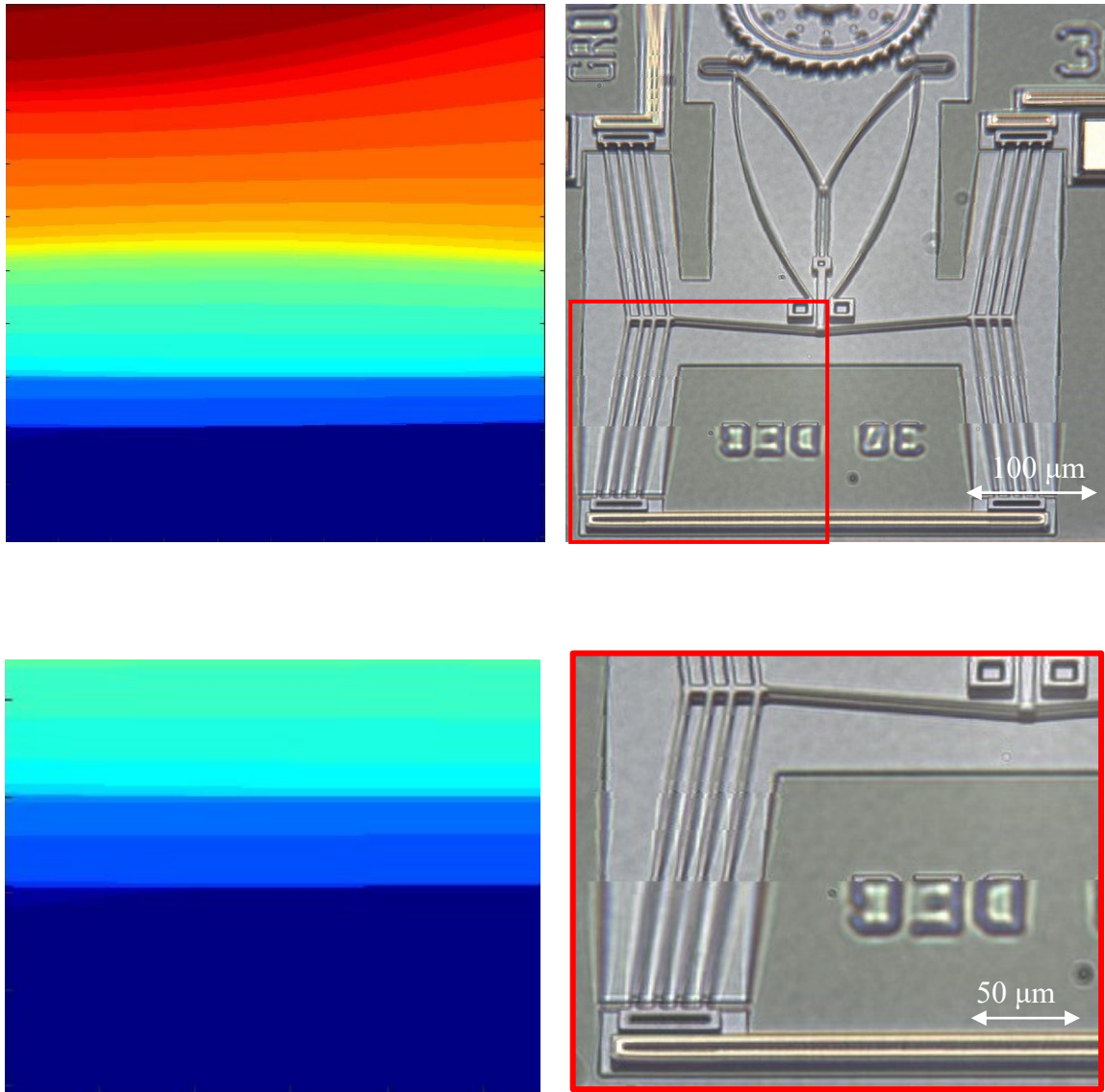


Figure 59 Excessive smoothing: Top: depth map & 3D image. Bottom: zoom-in on the lower left corner. (Smoothing = 1000).



### 6.3.Guidelines for the number of images

By capturing and analyzing fewer images, the time required to fuse them decreases. However, this approach introduces certain issues. When employing a reduced number of images, more blurred spots may appear within the resultant images due to the lack of information available for generating a focused portion. It's important to strike a balance between the number of images taken and the quality of the fused output to ensure clear and sharp results.

Ideally, hundreds of images could be taken to meticulously select the finest pixel for each corresponding pixel. Regrettably, adopting this approach would consume an extensive amount of time and prove to be inefficient. In Figure 60, the number of captured images was increased to observe its impact on the final image and determine the optimal quantity of pictures necessary for generating a fully focused image. All the images were aligned and fused using 100 units of smoothing. It is evident across all the images that certain portions are out of focus, and lines appear distorted due to substantial gaps in the image. However, by increasing the quantity of captured photos, the image quality improves significantly, resulting in enhanced sharpness and overall focus.

A minimum of 25 images is recommended to achieve a good-quality result. The number of figures required depends on the image quality desired, but you should capture many images with minimal jumps to create the best possible image. Based on this a number of images in the range of 25 - 60 is recommended.

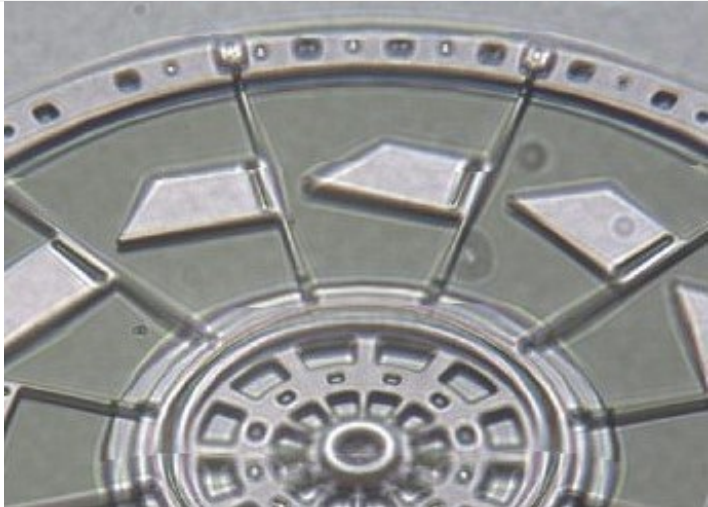
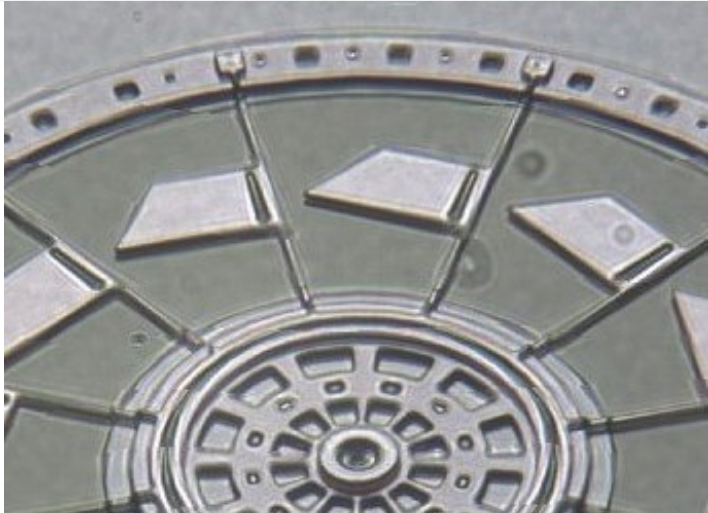
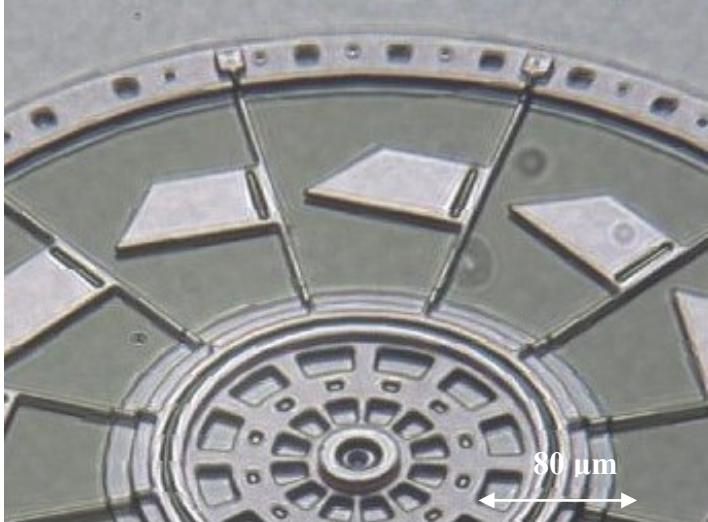
<p>10 images</p>	
<p>25 images</p>	
<p>50 images</p>	

Figure 60 Effect of the number of captured images on 3D image quality.

## 7. Gallery

The following section displays a gallery of approximately half a dozen different MEMS devices. It starts with the presentation of 2D images, followed by their respective 3D images. Among the images included are those of actuators, which convert energy into motion by heating and expanding elements. Circular electrostatic motors are also featured. Images of wire bonding pads are also included. Finally, as the goal of this thesis was to create 3D MEMS images, the last images include a comparison of a 3D image from this work with a SEM image.

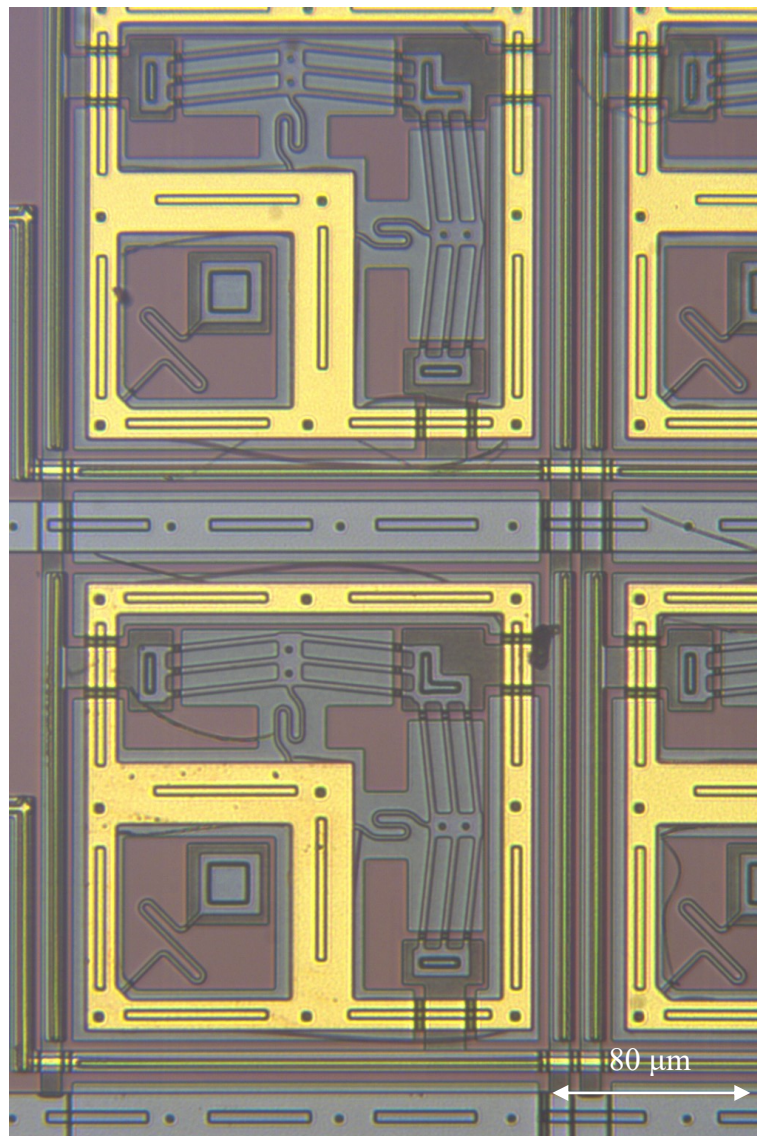


Figure 61 2D overhead image of MEMS Grid Actuator (chip = IMUDTPH1).



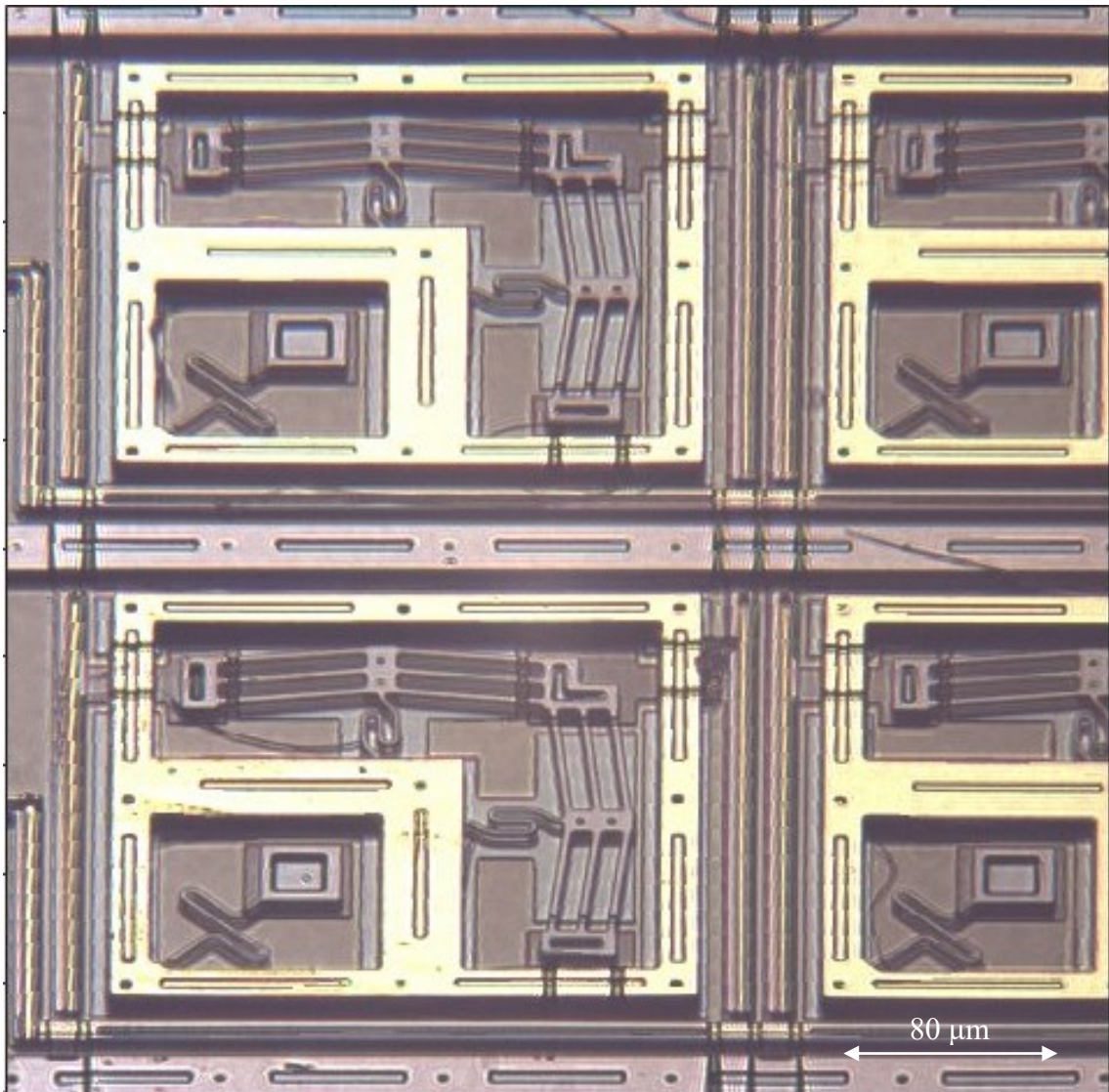


Figure 62 3D image of MEMS Grid Actuator from the previous Figure.

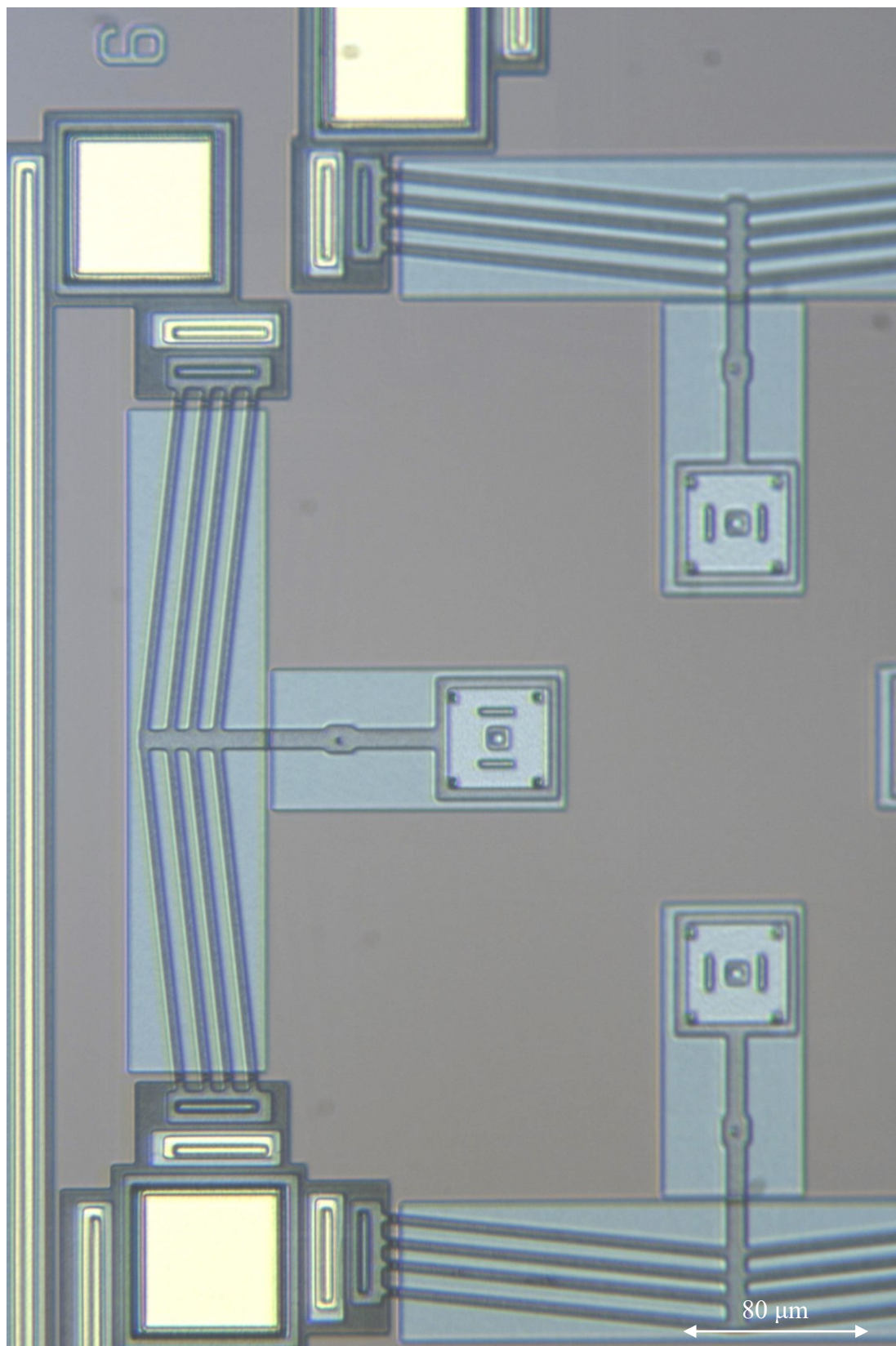


Figure 63 2D overhead image of MEMS Thermal Actuators (chip = IMUDTPH1).

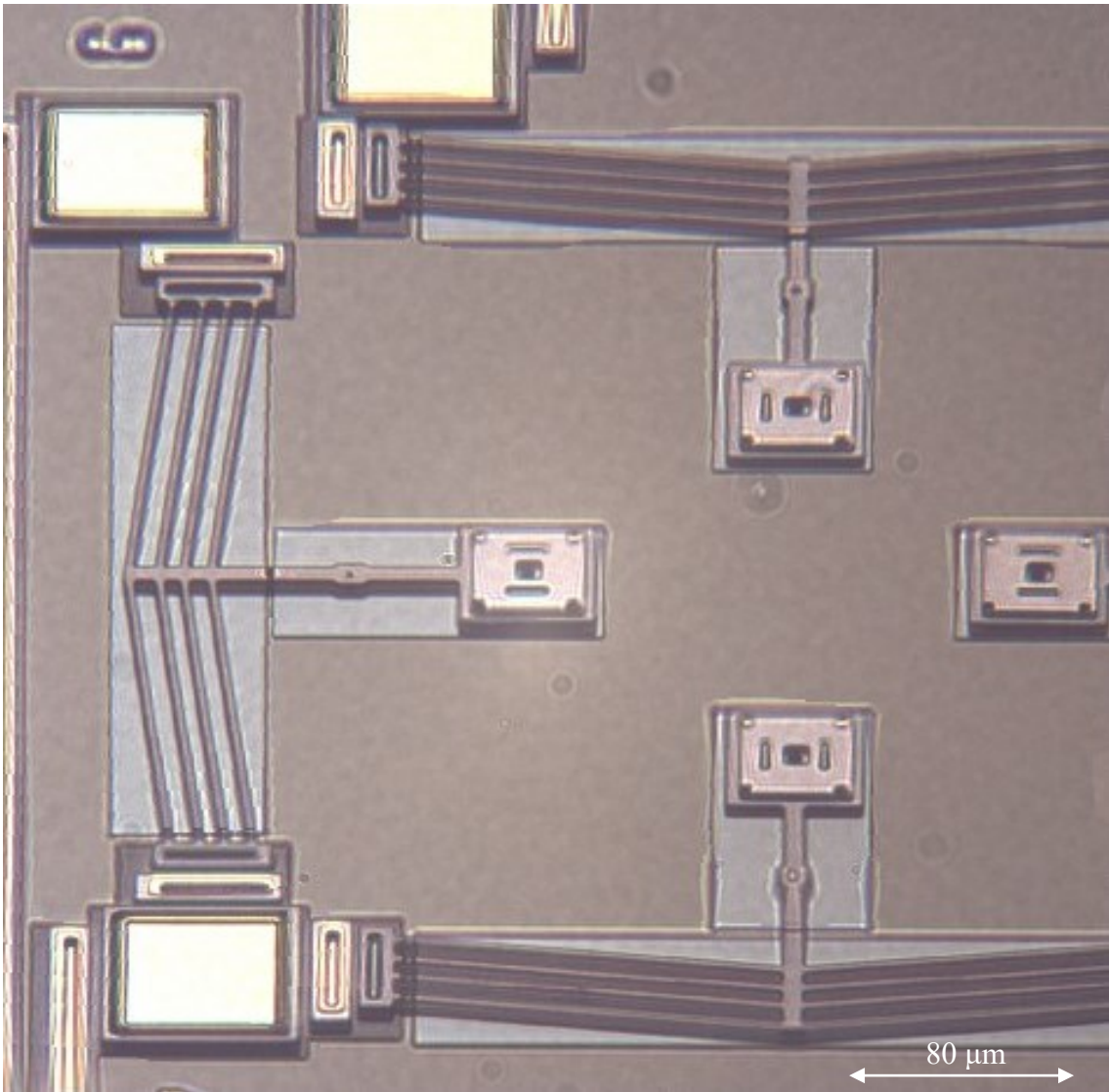


Figure 64 3D image of MEMS Thermal Actuators from the previous Figure.



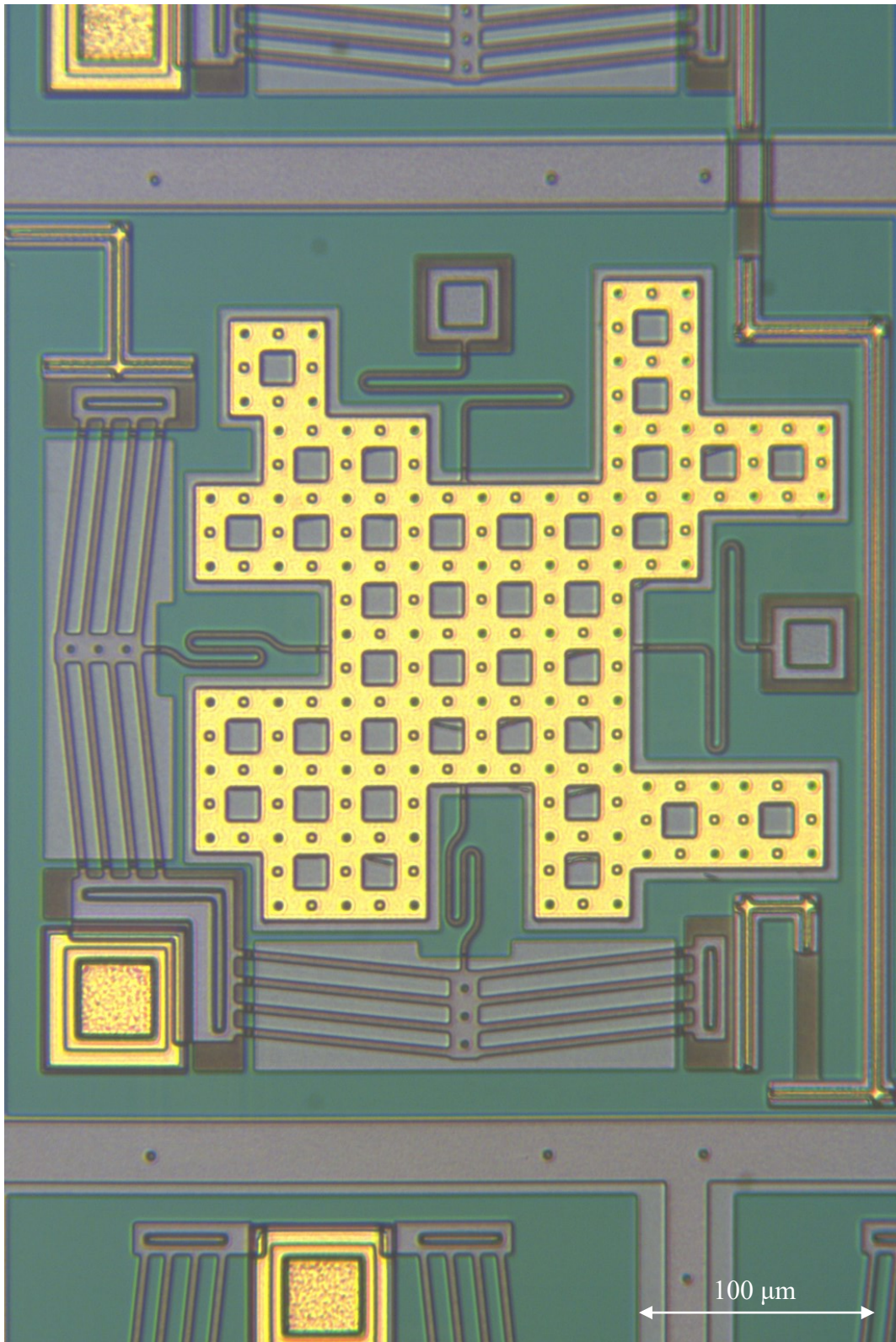


Figure 65 2D image of MEMS irregular gold-coated plate on modified Grid Actuator (chip = IMUDTCV4).

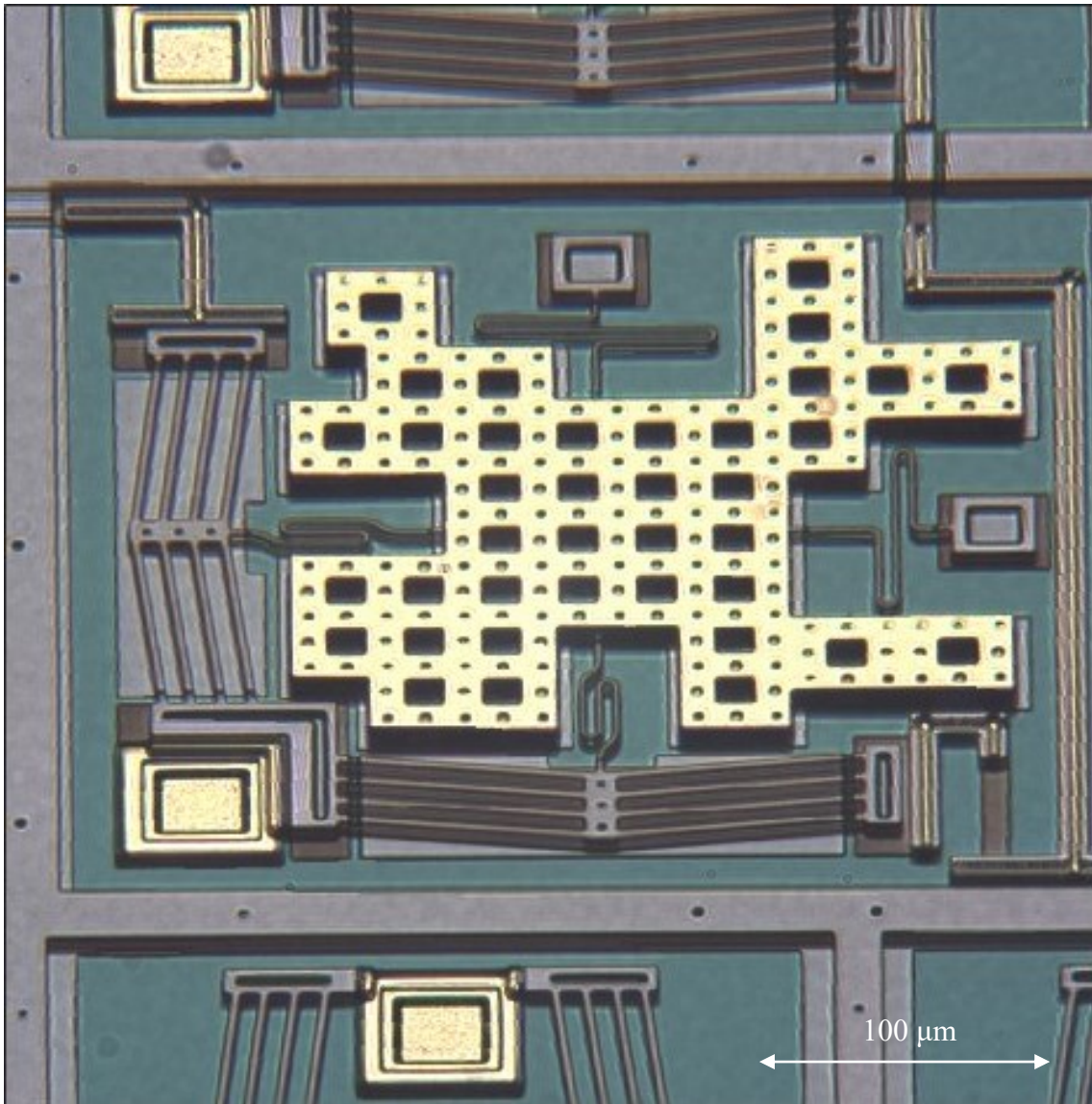


Figure 66 3D image of MEMS irregular gold-coated plate from the previous Figure.

In Figure 67, the methodology for calculating the height of the features is presented. To begin, the microns per pixel of an image must be determined based on a known object. Using this microns-per-pixel value, the pixel count of the shadow can be determined. Since the shadow portion of the image appears larger than the actual height, a multiplication factor is applied: the cosine of 45 degrees = 0.707



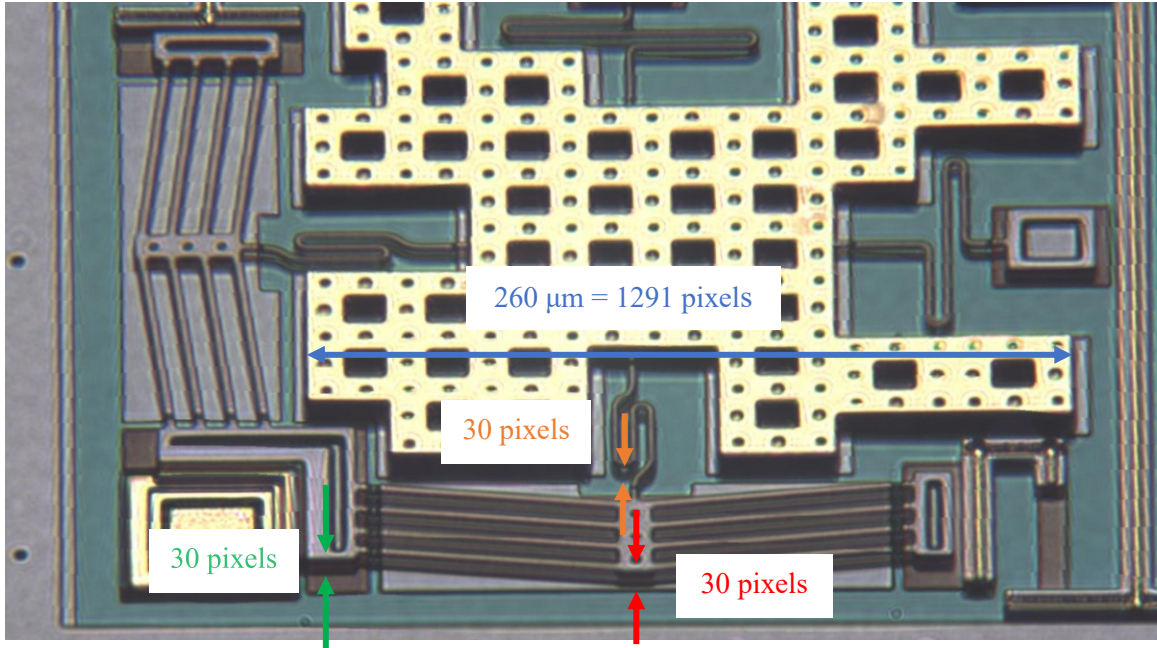


Figure 67 Calculating the heights of POLY1 features.

The image scale is obtained by measuring a known length in X. In this case, a 260  $\mu\text{m}$  feature was 1291 pixels long, giving 0.20  $\mu\text{m}/\text{pixel}$  in X. Note that due to the 45 degree angle, this corresponds to a 0.14  $\mu\text{m}/\text{pixel}$  in Z.

Pixel measurements of the POLY1 height were taken at three different locations, each with an expected height of 4  $\mu\text{m}$ . The resulting measurements at each location were consistently  $4.24 \pm 0.14 \mu\text{m}$ , as detailed in Table 4.

	Pixels	$\mu\text{m}$
Paperclip	$30 \pm 1$	$4.24 \pm 0.14$
Chevron Anchor	$30 \pm 1$	$4.24 \pm 0.14$
Chevron Center	$30 \pm 1$	$4.24 \pm 0.14$

Table 4 Measured heights of POLY1 objects.

To assess the accuracy of the method, we compared the measured values with the nominal MUMPs heights of POLY1 above the surface. According to MUMPs handbook [18], the layer height of OXIDE1 was  $2.0 \pm 0.25 \mu\text{m}$ , while POLY1, was  $2.0 \pm 0.15 \mu\text{m}$ , giving a combined height =  $4.0 \pm 0.28 \mu\text{m}$ . This suggests that our error falls within an acceptable range.

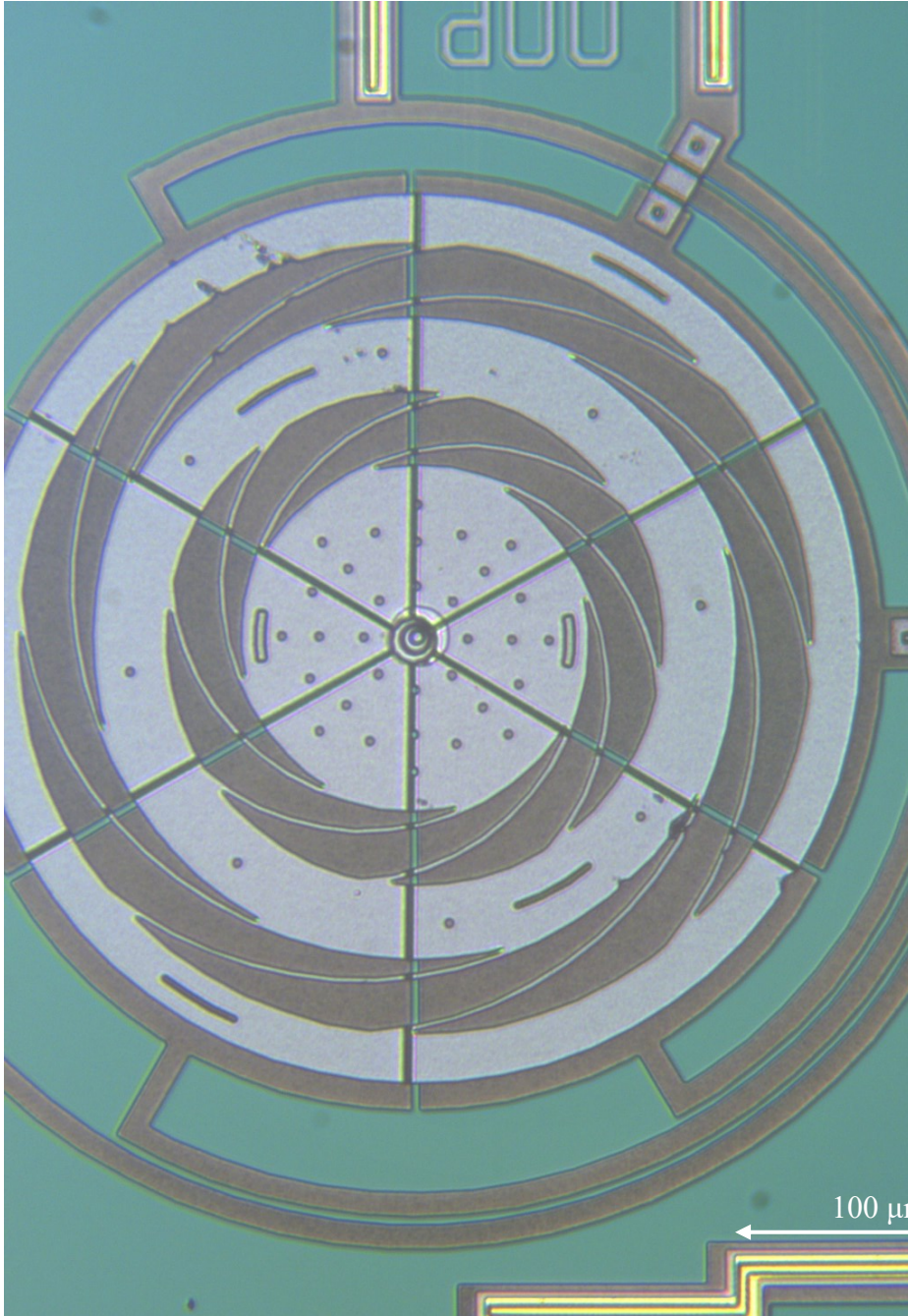


Figure 68 2D image of MEMS electrostatic motor (chip = IMUDTRMC).

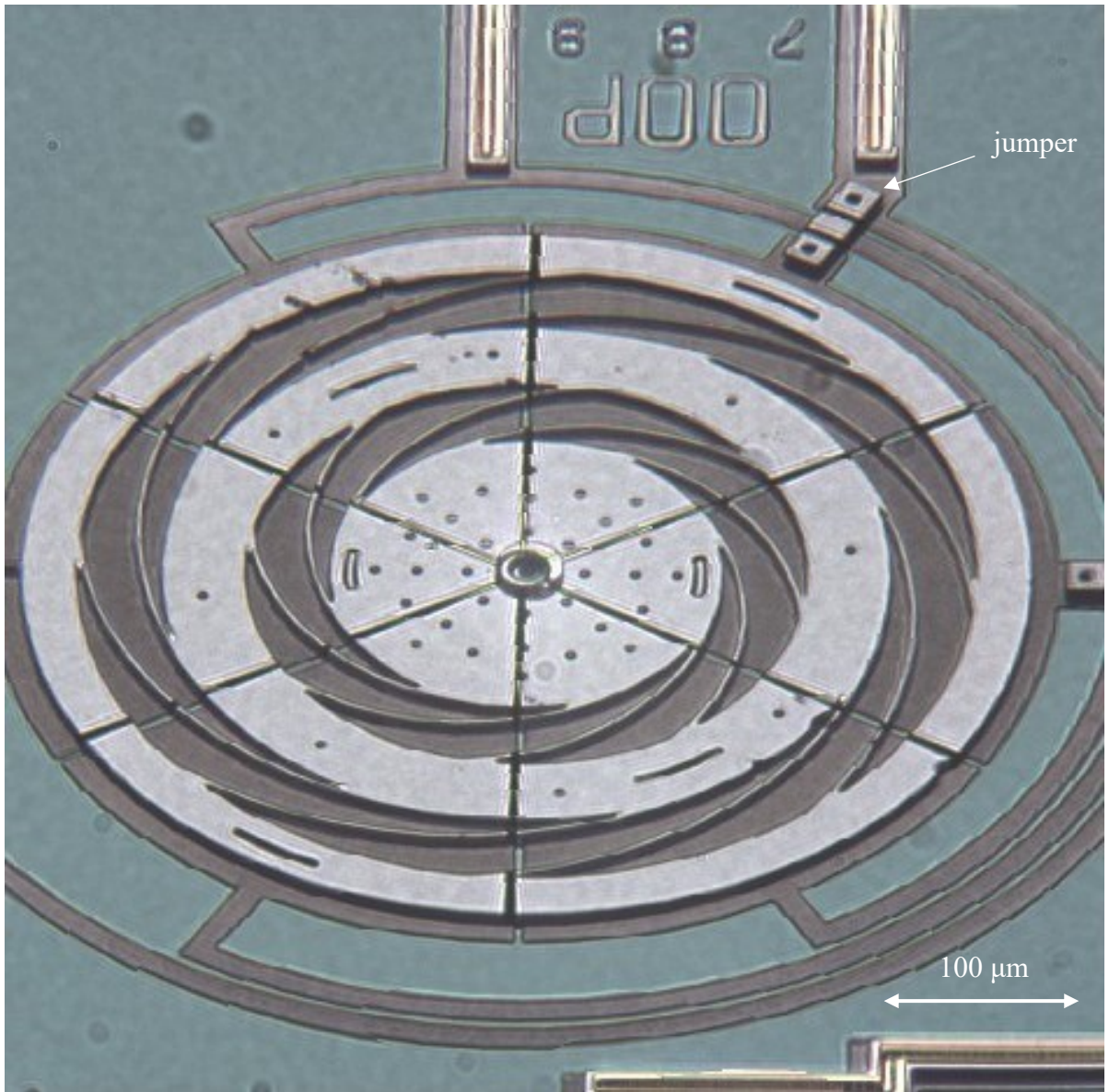


Figure 69 3D image of MEMS device shown in the previous Figure.



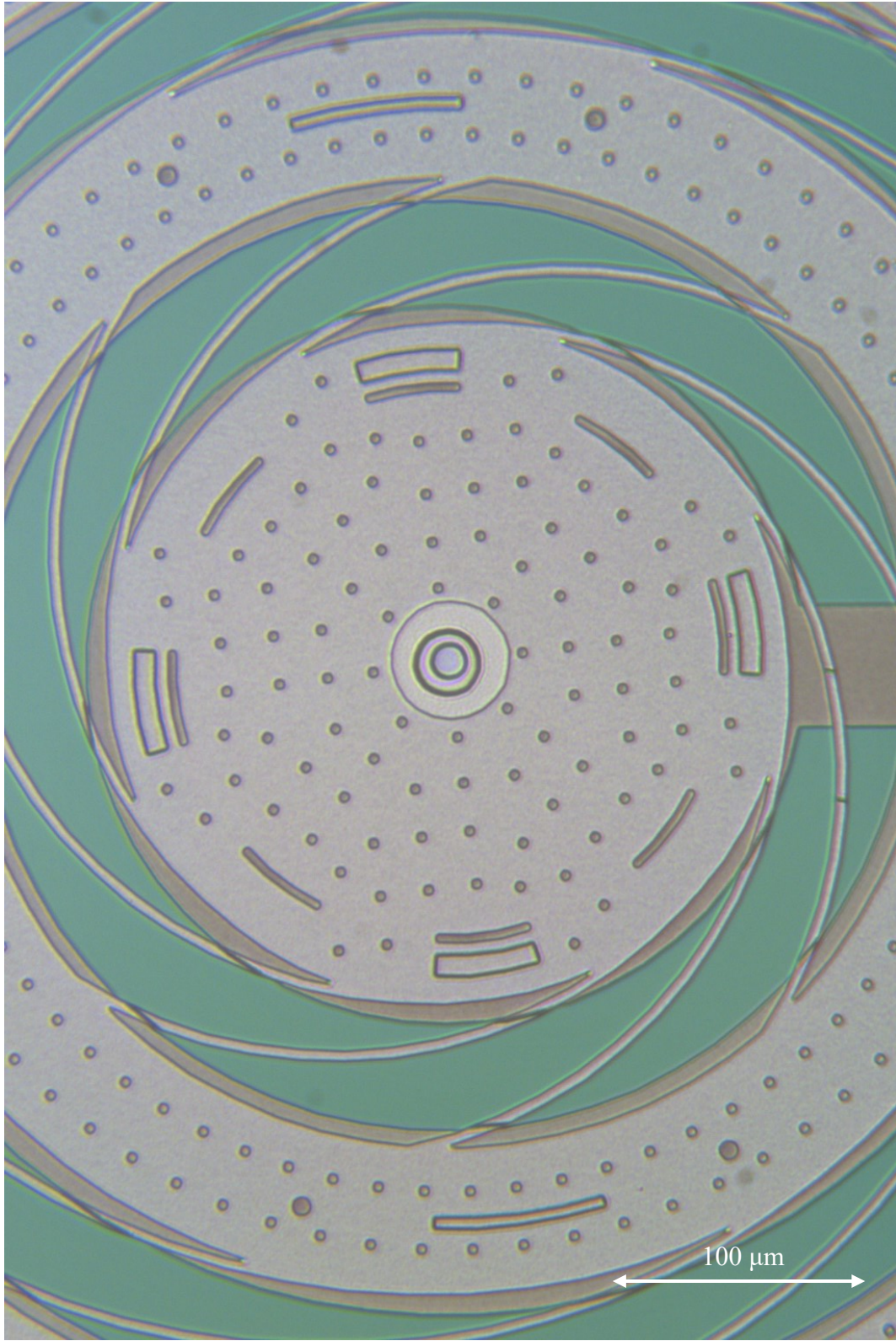


Figure 70 Closer 2D image of another MEMS electrostatic motor (chip = IMUDTRMC).

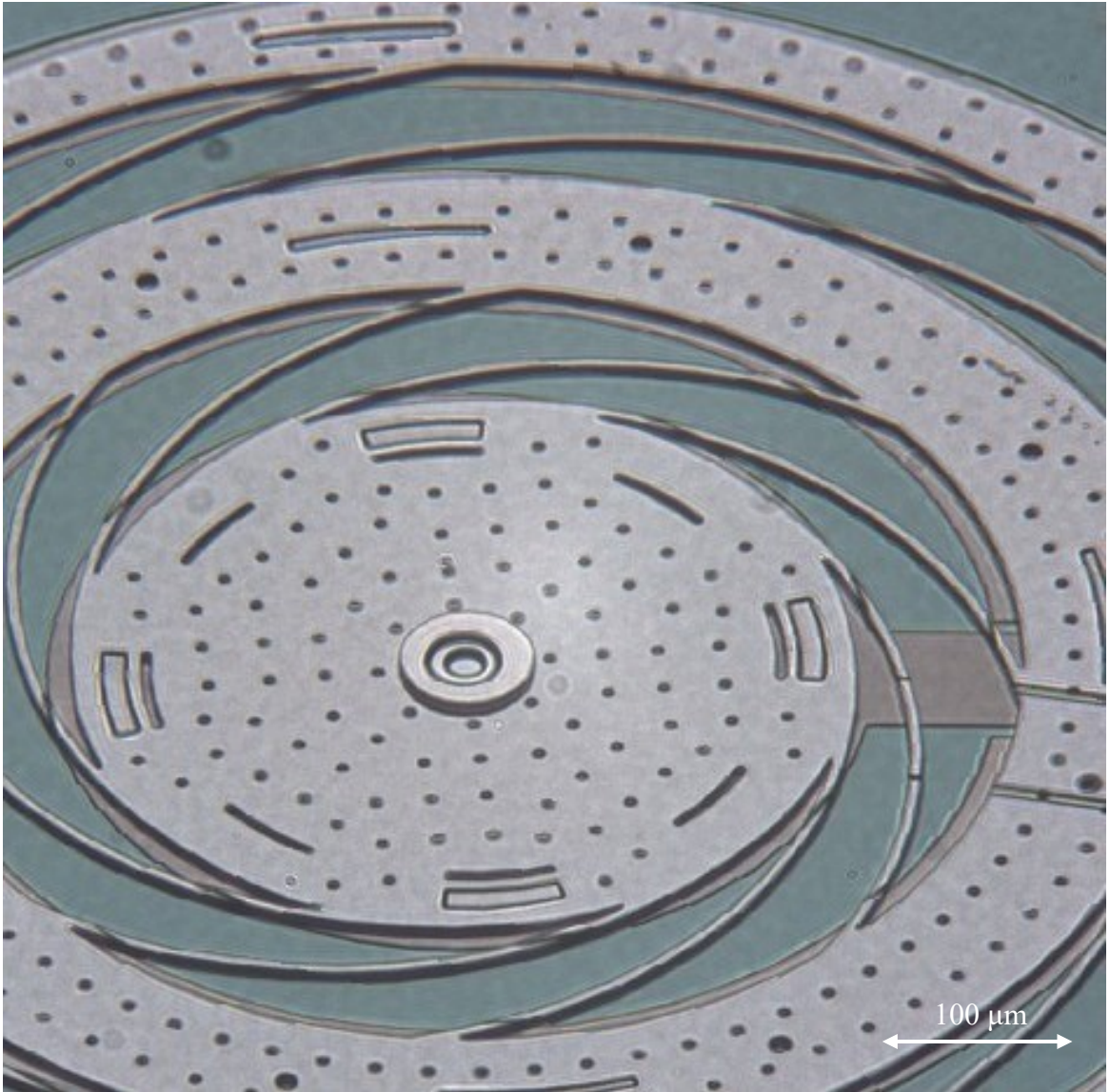


Figure 71 3D image of MEMS device shown in the previous Figure.



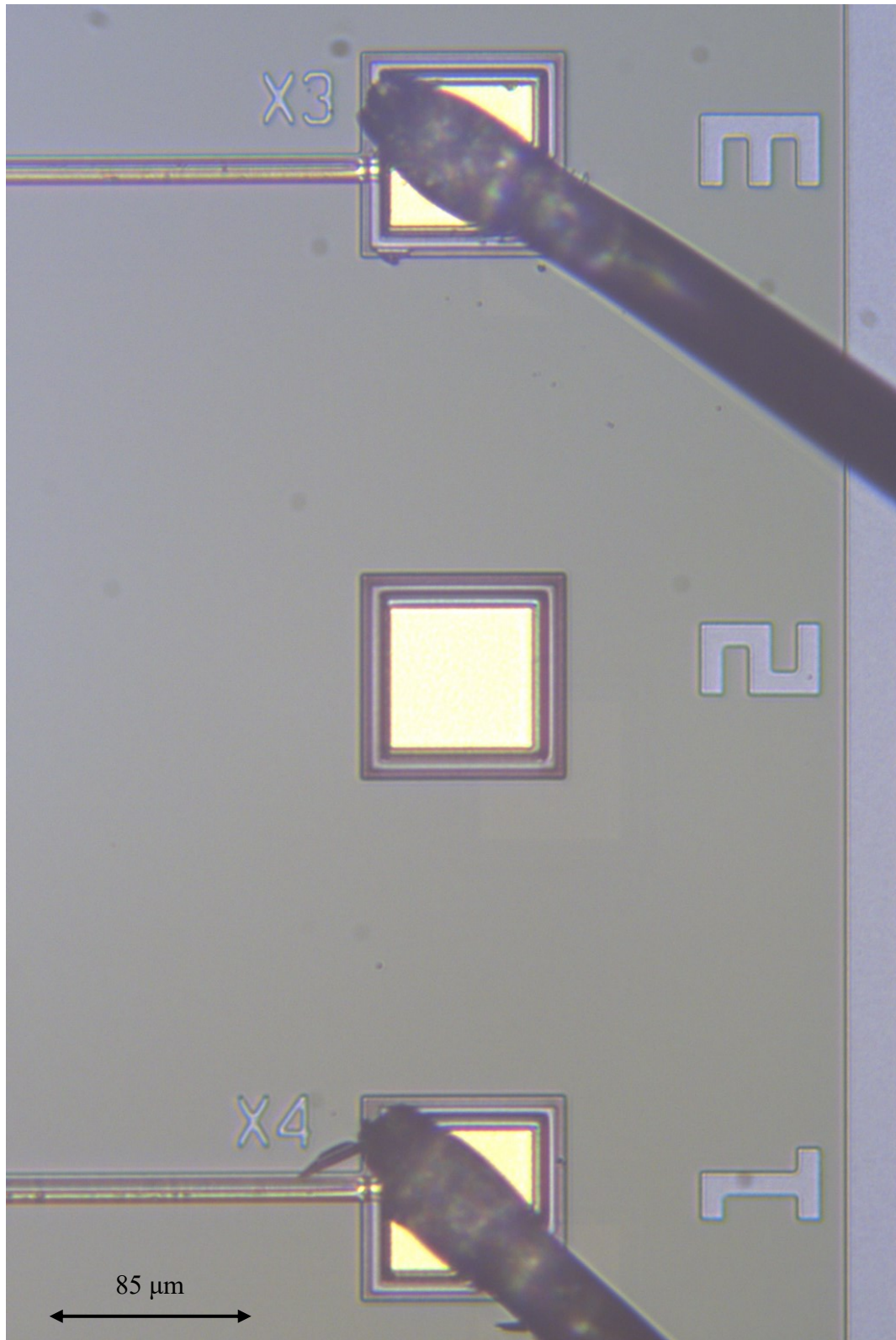


Figure 72 2D image of MEMS showing 2 wires and 3 bonding pads (chip = IMUDTEW1).

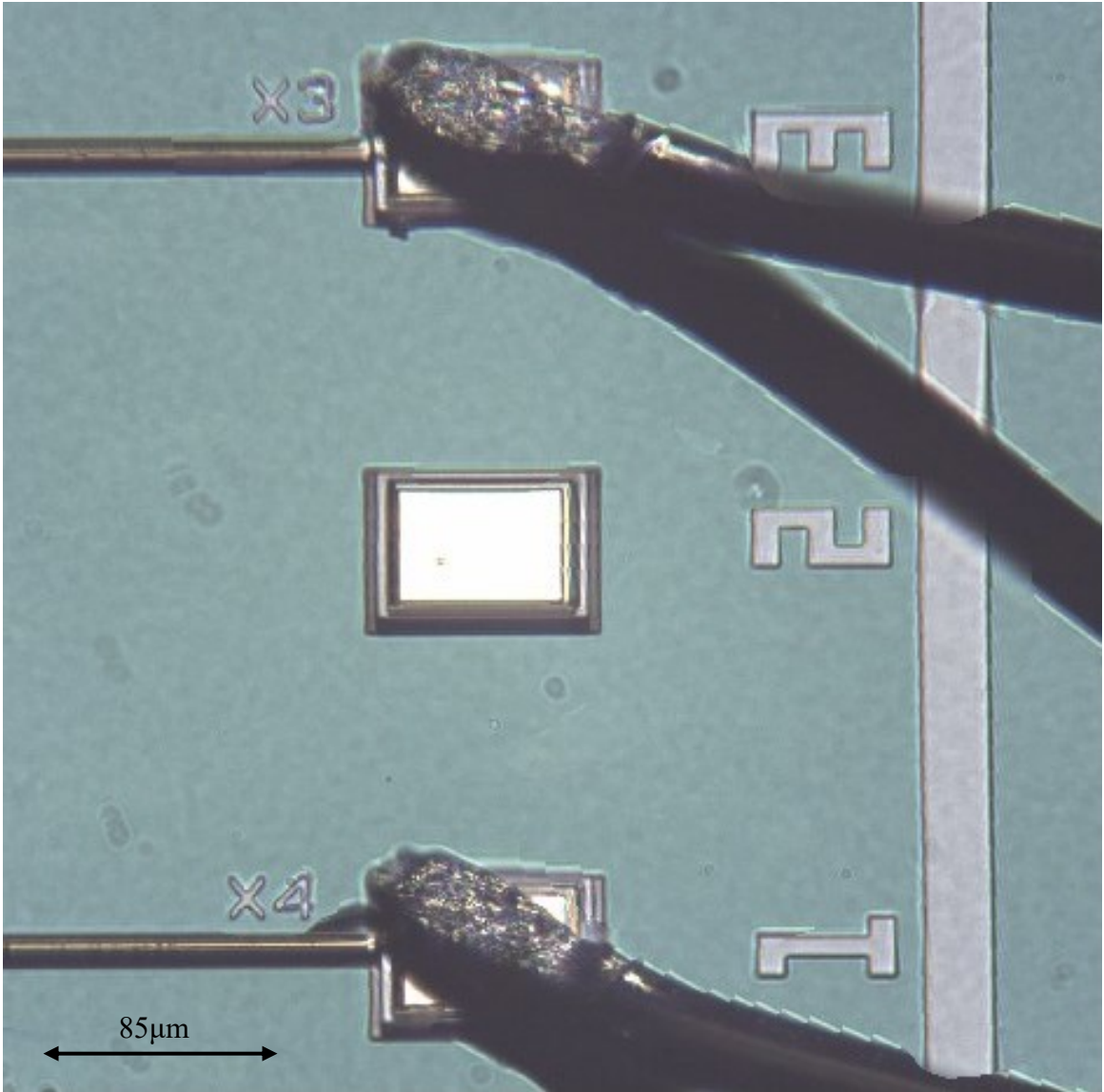


Figure 73 3D image of MEMS wires shown in the previous Figure.

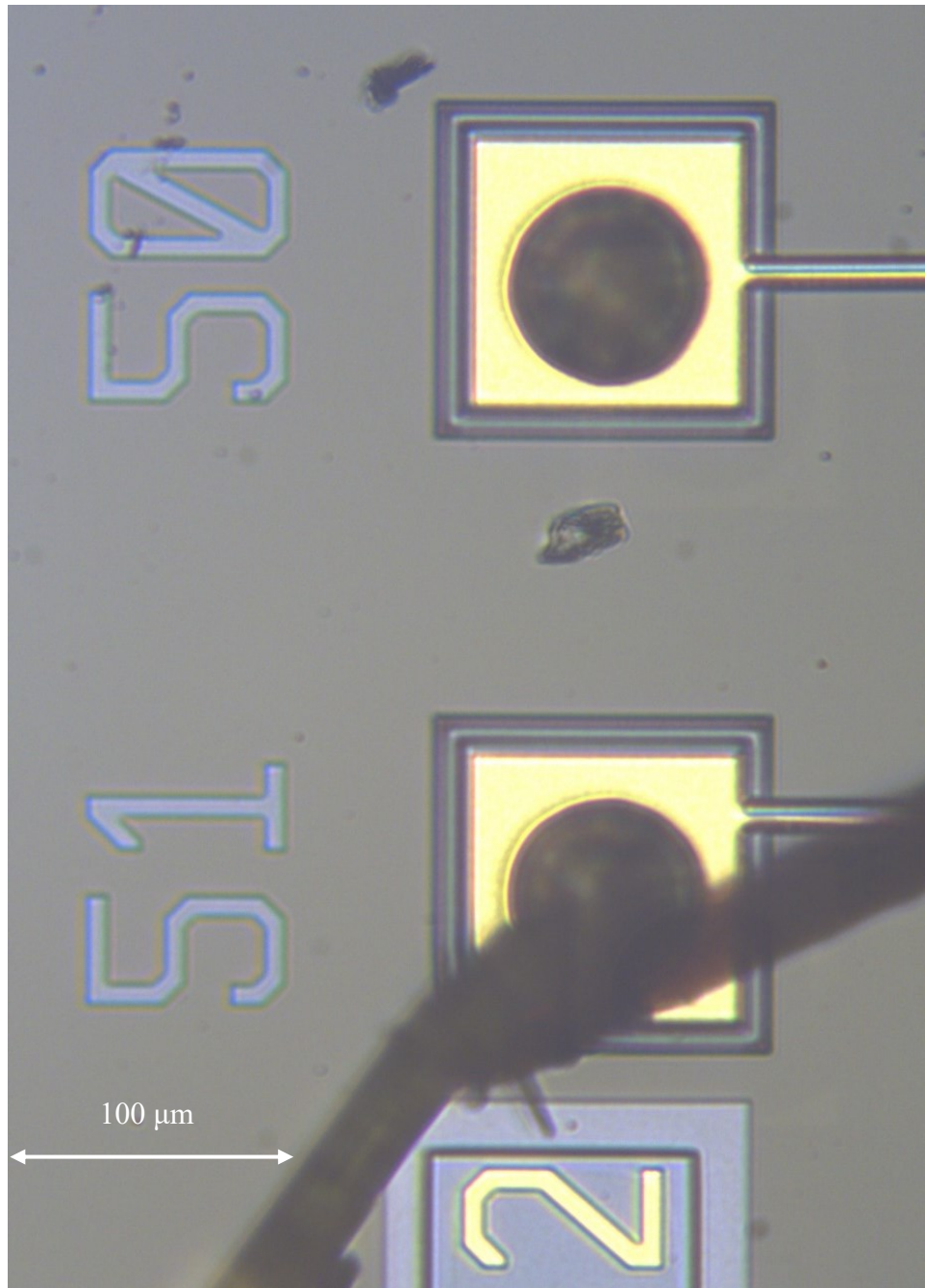


Figure 74 2D image showing 1 bonding pad with wire and 1 pad with unwired soldered ball (chip = IMUDTVT2).



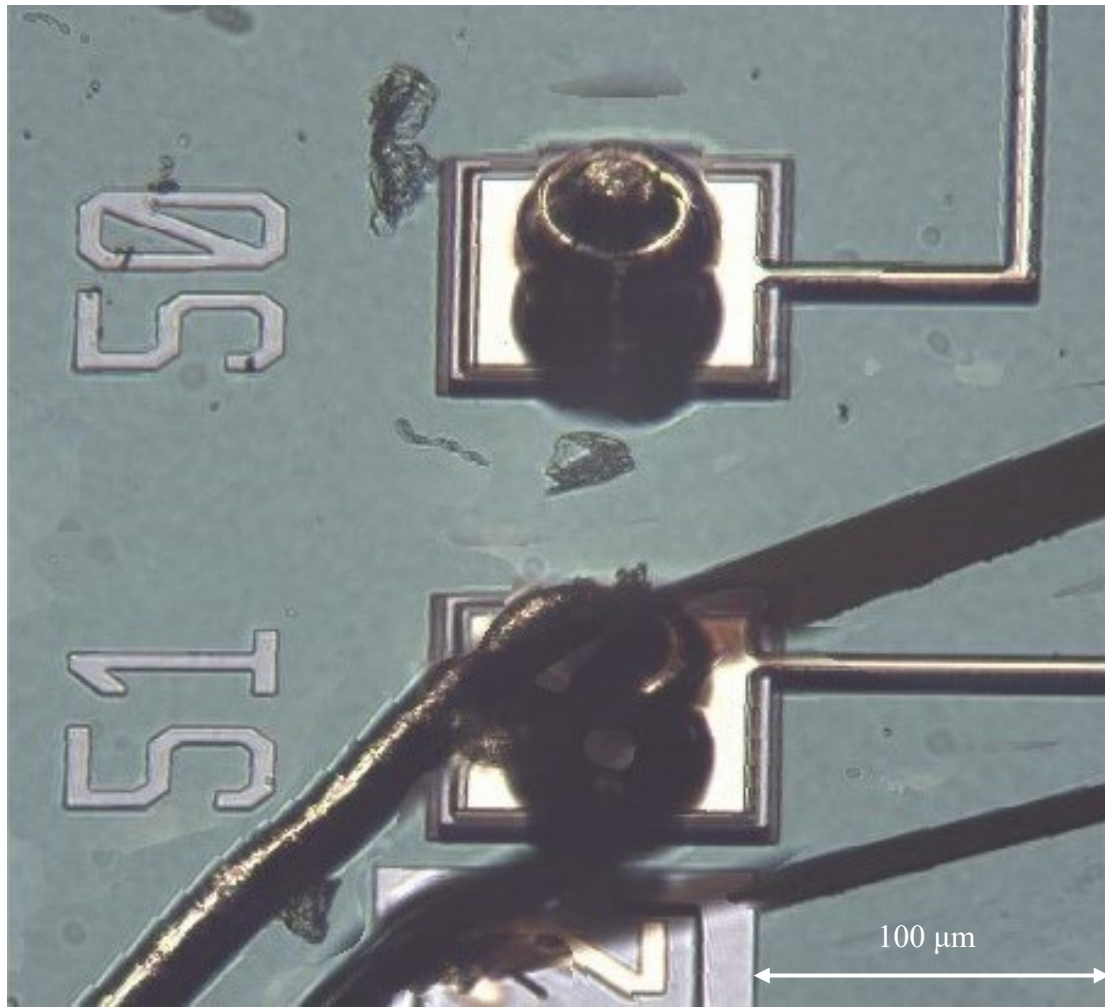


Figure 75 3D image of MEMS wire and solder ball shown in the previous Figure.

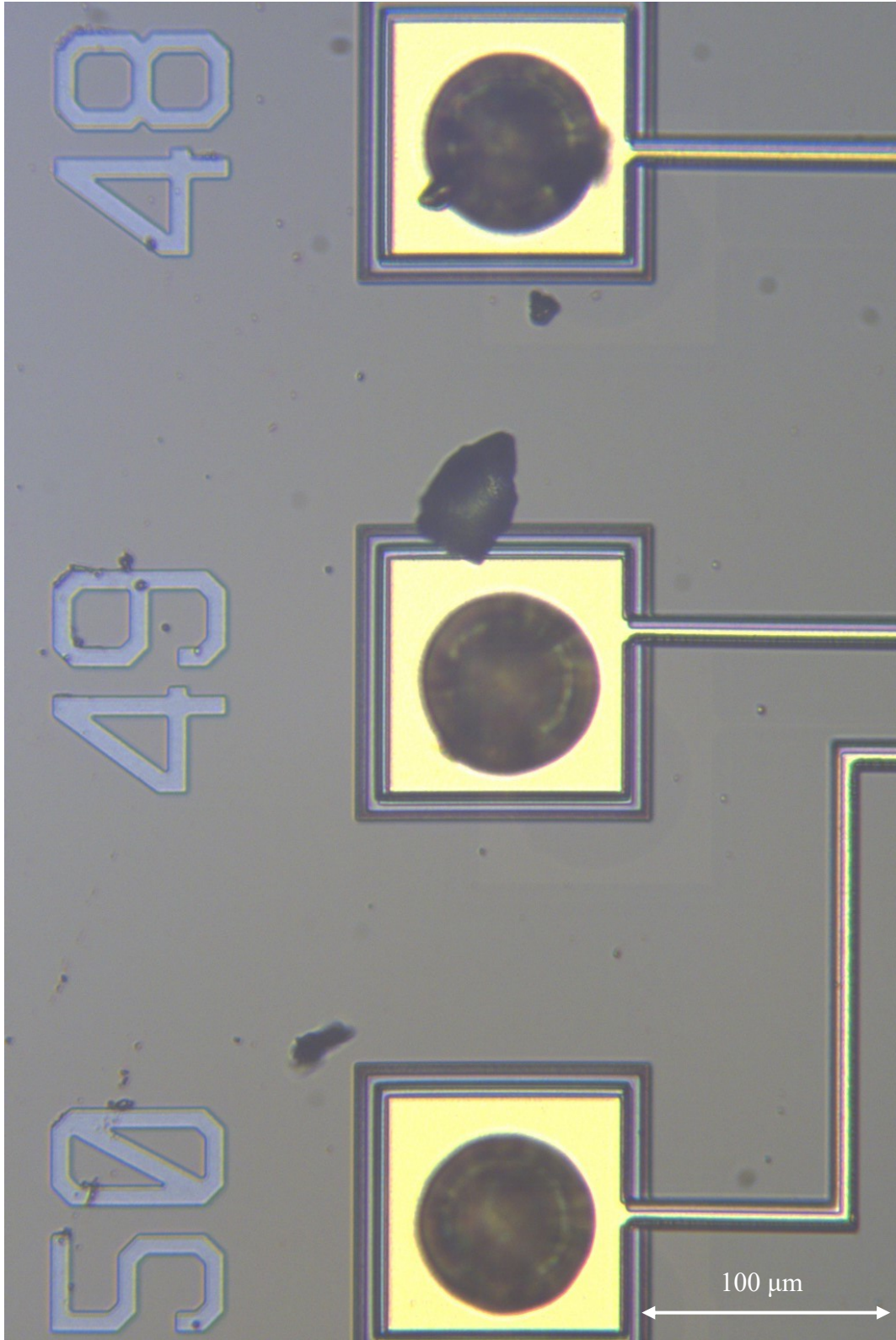


Figure 76 2D image showing 3 pads with unwired soldered balls (chip = IMUDTVT2).

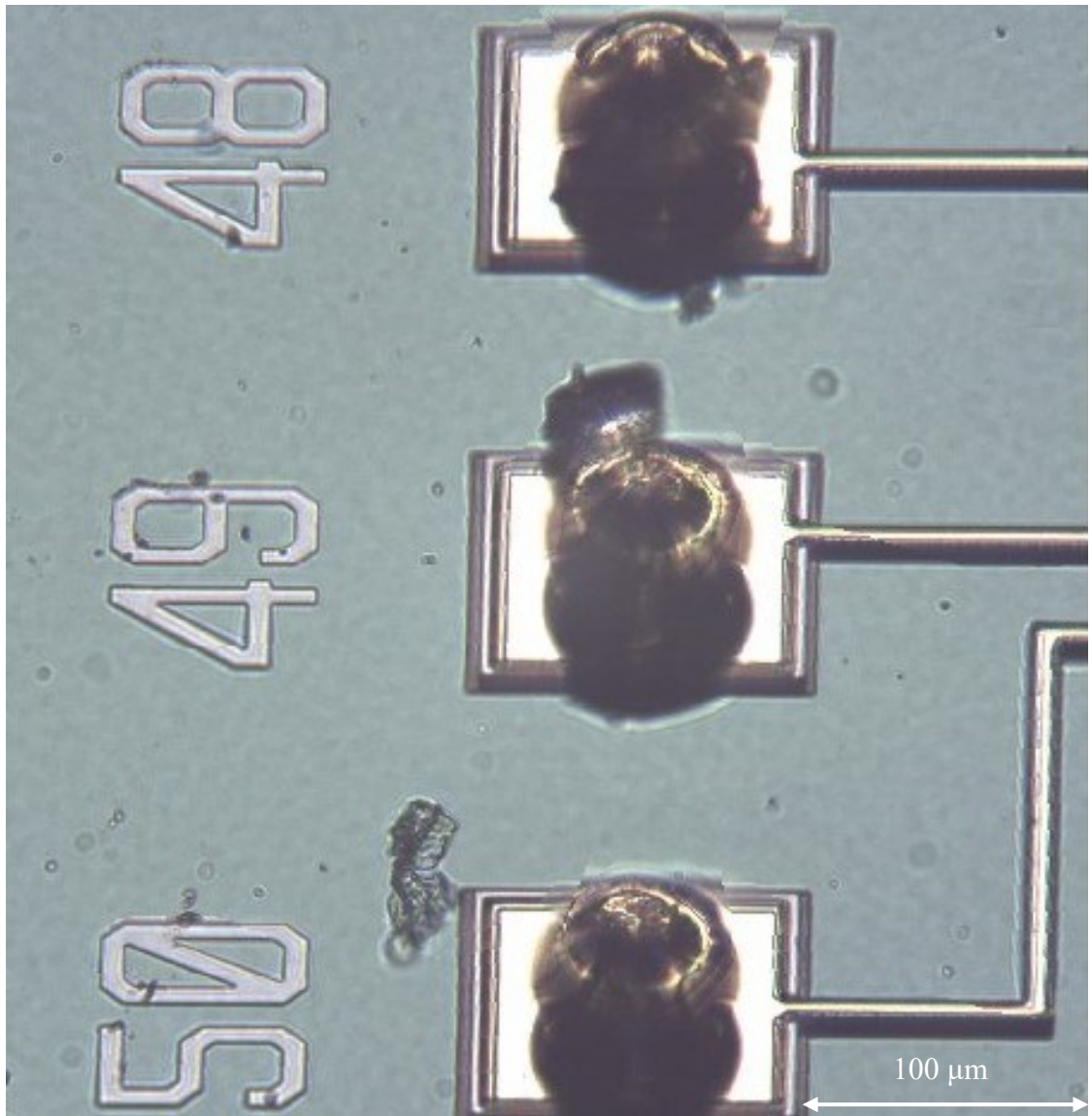


Figure 77 3D image of MEMS solder balls shown in the previous Figure.



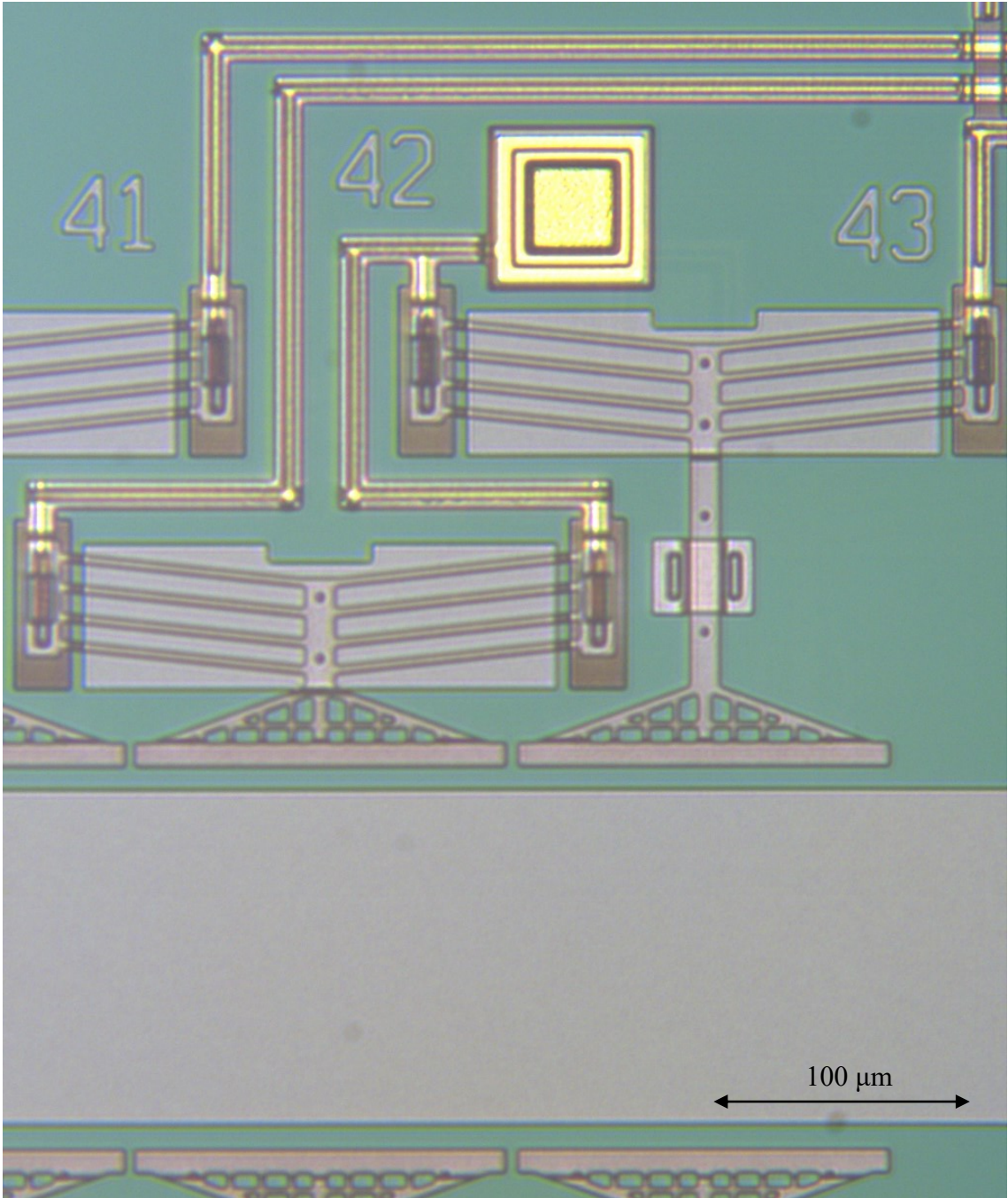


Figure 78 2D overhead image of MEMS Thermal Actuators (chip = IMUDTCV4).

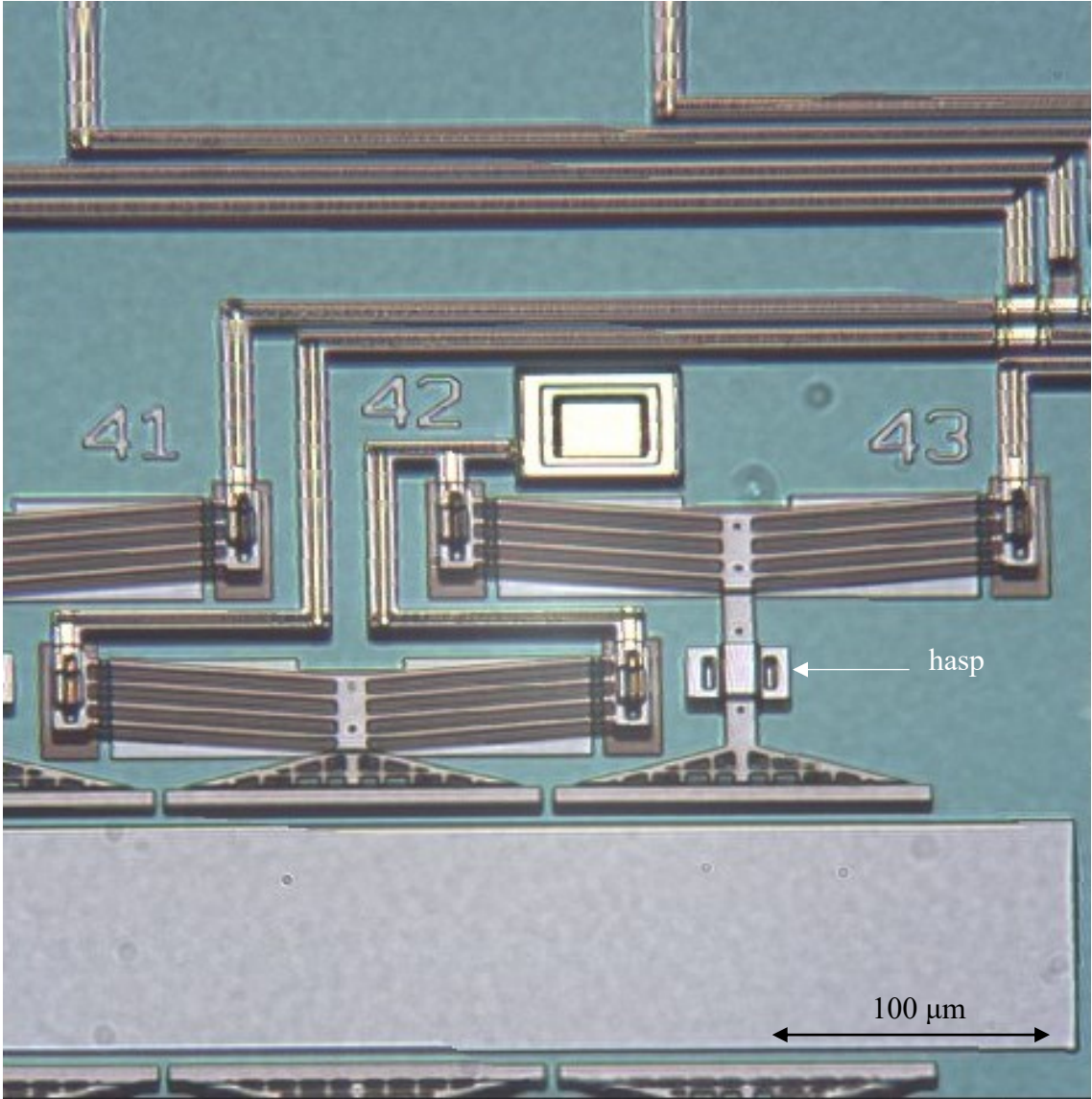


Figure 79 3D image of MEMS Thermal Actuators from the previous Figure.

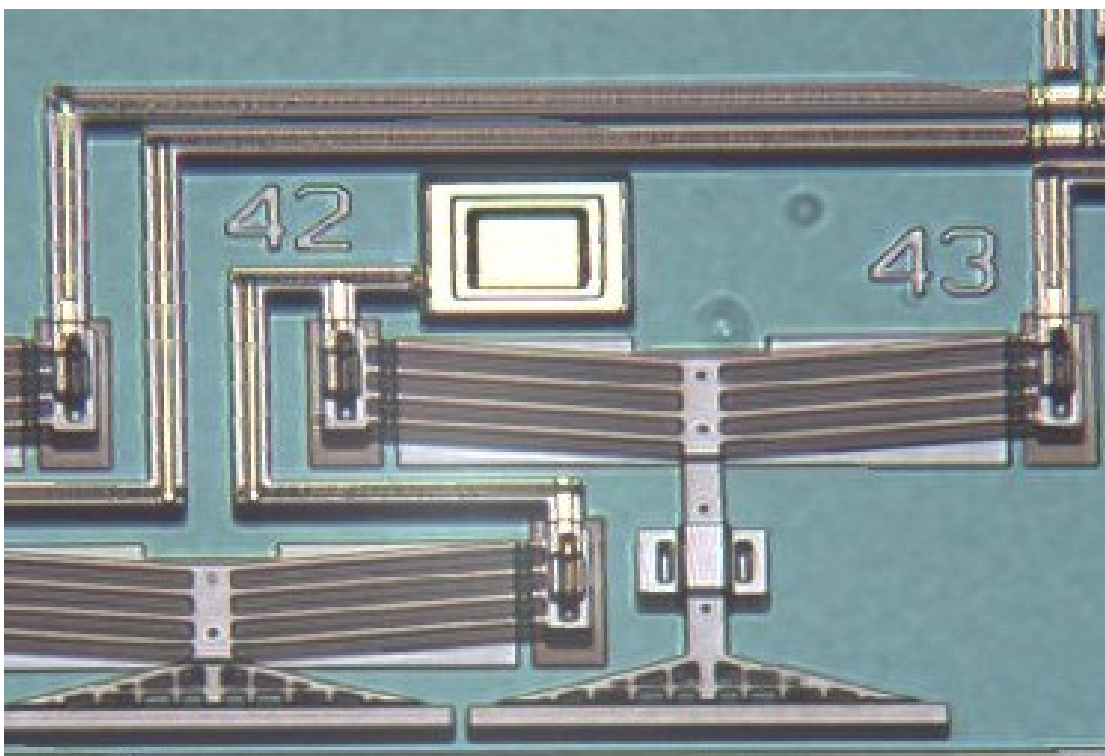
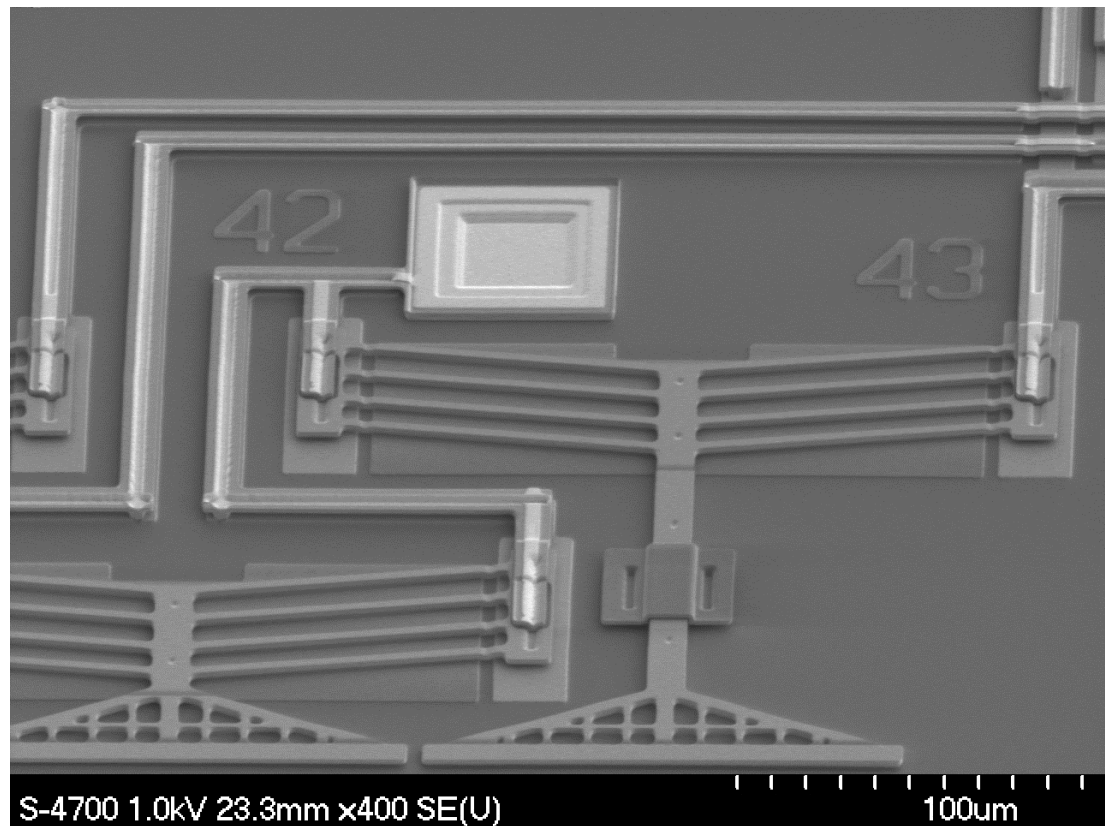


Figure 80 Comparison of SEM image from Section 2 with 3D image.

## 8. Conclusion

In this thesis, our research centred around an advanced imaging technique capable of generating high-quality, colour 3D-like images. Specifically, our objective was to produce detailed 3D representations of MEMS microchips, highlighting their intricate features. This endeavour aimed to facilitate the identification of these features, thereby aiding future developments in MEMS technology. While alternative methods for generating 3D-like images exist, they often come with substantial costs and complications and lack colour, as is the case with Scanning Electron Microscopy (SEM).

To capture 3D-like images, we adopted an existing non-MEMS approach. We acquired images from a perspective view by positioning a custom microscope at an angle relative to the MEMS chip. This setup featured a two-dimensional camera microscope that, through multiple perspective view images, provided three-dimensionality. The MEMS chip was mounted on a Zaber XYZ translation stage, which allowed movement between the cameras and allowed for adjustments to the focus position of the camera. These adjustments were necessary due to the shallow depth of field inherent in microscopy.

Our imaging process involved capturing a series of 30-50 images with varying focus points. We processed these images through multiple algorithms to register them, determine the focus for each pixel, and then refine the depth map. A depth map is a diagram/image that indicates for each pixel which source image had the highest focus value. Following this procedure, we fused selected pixels from the depth map together to create a final image.

To enhance image quality, we conducted a variety of tests, including adjustments to the number of images captured, variations in the smoothing level, and the utilization of different focus measure algorithms. We determined that the optimal approach involved capturing as many images as possible with minimal intervals between them. This allowed us to effectively capture the focus information across the entire image. While taking too many images could significantly extend the time required, we struck a balance by typically capturing 30 to 50 images. The smoothing level was adapted according to the



image content; when most of the image exhibited consistent features, more smoothing was applied, whereas images with abundant features required minimal smoothing. Among the various focus measure algorithms tested, it was evident that Helmi's algorithm outperformed all others.

## References

- [1] F. Sadjadi, "Comparative Image Fusion Analysis," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, San Diego, 2006.
- [2] D. Yatsenko, "measureShift," MATLAB Central File Exchange, 2016.
- [3] B. Reddy and B. Chatterji, "An FFT-based technique for translation, rotation, and scale-invariant image registration," *IEEE Transactions on Image Processing*, vol. 5, no. 8, pp. 1266-1271, 1996.
- [4] C. Zhang, J. Bastian, C. Shen, A. van den Hengel and T. Shen, "Extended depth-of-field via focus stacking and graph cuts," in *2013 IEEE International Conference on Image Processing*, Melbourne, 2013.
- [5] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977-1000, 2003.
- [6] S. Pertuz, D. Puig and M. A. Garcia, "Analysis of focus measure operators for shape-from-focus," *Pattern Recognition*, vol. 46, no. 5, pp. 1415-1432, 2013.
- [7] J. Brenner, B. Dew, J. Horton, T. King, P. Neurath and W. Selles, "An automated microscope for cytologic research a preliminary evaluation," *The journal of histochemistry and cytochemistry*, vol. 24, no. 1, pp. 100-111, 1976.
- [8] M. I. Shah, S. Mishra, M. Sarkar and C. Rout, "Identification of robust focus measure functions for the automated capturing of focused images from Ziehl-Neelsen stained sputum smear microscopy slide," *Cytometry A.*, vol. 91, no. 8, pp. 800-809, 2017.
- [9] O. A. Osibote, R. Dendere, S. Krishnan and T. S. Douglas, "Automated focusing in bright-field microscopy for tuberculosis detection," *J Microsc.*, vol. 240, no. 2, pp. 155-163, 2010.
- [10] S. Pertuz, D. Puig, M. A. Garcia and A. Fusiello, "Generation of All-in-Focus Images by Noise-Robust," *IEEE Transactions on Image Processing*, vol. 22, no. 3, pp. 1242-1251, 2013.
- [11] D. Fedorov, B. Sumengen and B. S. Manjunath, "Multi-Focus Imaging using Local Focus Estimation and Mosaicking," in *2006 International Conference on Image Processing*, Atlanta, 2006.

- [12] D. Choi, A. Pazyzbekova, W. Zhou and P. v. Beek, "Improved image selection for focus stacking in digital photography," in *2017 IEEE International Conference on Image Processing (ICIP)*, Beijing, 2017.
- [13] M. Olkowicz, M. Dabrowski and A. Pluymakers, "Focus stacking photogrammetry for micro-scale roughness reconstruction: a methodological study," *The Photogrammetric Record*, vol. 34, no. 165, pp. 11-35, 2019.
- [14] A. Bercovici, A. Hadley and U. Villanueva-Amadoz, "Improving depth of field resolution for palynological microscopy," *Palaeontologia Electronica*, vol. 12, no. 2, 2009.
- [15] D. Clark and B. Brown, "A Rapid Image Acquisition Method for Focus Stacking in Microscopy," *Microscopy Today*, vol. 23, no. 4, pp. 18-25, 2015.
- [16] H. Xie, W. Rong, L. Sun and W. Chen, "Image Fusion and 3-D Surface Reconstruction of Microparts Using Complex Valued Wavelet Transforms," in *2006 International Conference on Image Processing*, Atlanta, 2006.
- [17] D. J. Bell, T. J. Lu, N. Fleck and S. M. Spearing, "MEMS actuators and sensors: observations on their performance and selection for purpose," *Journal of Micromechanics and Microengineering*, vol. 15, no. 7, pp. 153-164, 2005.
- [18] A. Cowen, B. Hardy, R. Mahadevan and S. Wilcenski, "PolyMUMPs design handbook," *MEMSCAP Inc.*, vol. 13, 2011.
- [19] A. Maier, S. Steidl, V. Christlein and J. Hornegger, *Medical Imaging Systems: An Introductory Guide*, Cham: Springer, 2018.
- [20] A. Maier, S. Steidl, V. Christlein and J. Hornegger, *Medical Imaging Systems An Introductory Guide*, Cham: Springer, 2018.
- [21] G. Hollows and N. James, "Understanding Focal Length and Field of View," Edmund Optics, [Online]. Available: [https://www.edmundoptics.com/knowledge-center/application-notes/imaging/understanding-focal-length-and-field-of-view/?gclid=Cj0KCQjwldKmBhCCARIsAP-0rfwCO25zHi8\\_8lSi1Y2V8jhMKRAqMlxix9omYQXtCBCHOUrJOZepvIcaAvNEEALw\\_wcB](https://www.edmundoptics.com/knowledge-center/application-notes/imaging/understanding-focal-length-and-field-of-view/?gclid=Cj0KCQjwldKmBhCCARIsAP-0rfwCO25zHi8_8lSi1Y2V8jhMKRAqMlxix9omYQXtCBCHOUrJOZepvIcaAvNEEALw_wcB). [Accessed 10 August 2023].
- [22] K. R. Spring and M. W. Davidson, "Depth of Field and Depth of Focus," Nikon MicroscopyU, [Online]. Available: <https://www.microscopyu.com/microscopy-basics/depth-of-field-and-depth-of-focus>. [Accessed 26 July 2023].
- [23] J. C. Long, K. R. Spring and M. W. Davidson, "Objective Working Distance," Nikon Microscopy U, [Online]. Available:

<https://www.microscopyu.com/tutorials/workingdistance>. [Accessed 9 August 2023].

- [24] Zaber Technologies Inc, "Zaber T-Series Positioning Products User's Manual," 14 July 2006. [Online]. Available: <https://www.zaber.com/documents/ZaberT-SeriesProductsUsersManual2.xx.pdf>. [Accessed 27 July 2023].
- [25] S. Pertuz, "Focus Measure," MATLAB Central File Exchange, 2023.
- [26] F. S. Helmi and S. Scherer, "Adaptive shape from focus with an error estimation in light microscopy," in *ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat., Pula, 2001*.
- [27] N. Farahani, A. Braun, D. Jutt, T. Huffman, N. Reder, Z. Liu, Y. Yagi and L. Pantanowitz, "Three-dimensional Imaging and Scanning: Current and Future Applications for Pathology," *J Pathol Inform*, vol. 8, 2017.
- [28] A. Santos, C. Ortiz De Solórzano, J. J. Vaquero, J. M. Peña, N. Malpica and F. Del Pozo, "Evaluation of autofocus functions in molecular cytogenetic analysis," *Journal of microscopy*, vol. 188, no. 3, pp. 264-272, 1997.

## Appendix A-1 Image Registration

```
% Create image stacks
firstImage = imread(fileNames{1});
imageWidth = size(firstImage, 2);
imageHeight = size(firstImage, 1);

% Initialize the image stack to save images
imageStack = zeros(imageHeight, imageWidth, numImagesToProcess, 3);

% Initialize arrays to store shift values between images
xShiftTable = zeros(1, numImagesToProcess);
yShiftTable = zeros(1, numImagesToProcess);

% Initialize the reference image
referenceImage = imread(fileNames{1});
referenceImage = rgb2gray(referenceImage);
previousImage = referenceImage;

currentImageIndex = 1;
disp(['Measuring ', num2str(numImagesToProcess), ' images...']);

% Measure shifts and save images to the stack
for i = 1:stepSize:totalNumFiles
    fprintf(['.']);

    % Read the current image from file
    currentImage = imread(fileNames{i});
    currentImage = im2double(currentImage);

    % Save the current image to the image stack
    imageStack(:, :, currentImageIndex, :) = currentImage;

    % Convert images to grayscale
    currentImage = rgb2gray(currentImage);

    % Use MeasureShift to calculate shift between each image
    [xShiftDiff, yShiftDiff] = measureShift(fft2(currentImage) .*
conj(fft2(previousImage)));
    xShiftTable(currentImageIndex) = round(xShiftDiff, 1);
    yShiftTable(currentImageIndex) = round(yShiftDiff, 1);

    % Set the current image as the previous image for the next iteration
    previousImage = currentImage;

    currentImageIndex = currentImageIndex + 1;
end
```

```

end
fprintf('\n');

% Calculate cumulative shift values
for i = 2:numImagesToProcess
    xShiftTable(i) = xShiftTable(i) + xShiftTable(i - 1);
    yShiftTable(i) = yShiftTable(i) + yShiftTable(i - 1);
end

% Calculate shift deltas
xShiftDelta = max(xShiftTable) - min(xShiftTable);
yShiftDelta = max(yShiftTable) - min(yShiftTable);

% Calculate the dimensions of the registered images
outputWidth = round(imageWidth + abs(xShiftDelta), 0);
outputHeight = round(imageHeight + abs(yShiftDelta), 0);

% Display a message indicating the registration process
disp(['Registering ', num2str(numImagesToProcess), ' images...']);

% Initialize a canvas for the registered image
registeredImage = zeros(outputHeight, outputWidth, 3);

% Process and save registered images
for i = 1:numImagesToProcess
    fprintf(['.']);

    % Get the current image from the stack
    currentImage = squeeze(imageStack(:, :, i, :));

    % Create a blank canvas for the shifted image
    registeredImage = zeros(outputHeight, outputWidth, 3);

    xStart = abs(round(xShiftTable(i), 0));
    yStart = abs(round(yShiftTable(i), 0));

    disp(['i = ', num2str(i), ', xStart = ', num2str(xStart), ', yStart = ', num2str(yStart)]);

    % Place the shifted image onto the canvas
    registeredImage(yStart + 1:imageHeight + yStart, xStart + 1:imageWidth + xStart, :) =
currentImage;

    % Save the registered image to the "REG" folder
    imwrite(registeredImage, fullfile('REG', fileNames{i}));
end
fprintf('\n');

```



## Appendix A-2 Focus Measure

```
% Read the first image in the list to determine the image dimensions
first_image = imread(file_names(1, :));
x_size = size(first_image, 2);
y_size = size(first_image, 1);

% Create 3D arrays to store image data and focus measures
image_stack = zeros(y_size, x_size, num_pics, 3);
focus_stack = zeros(y_size, x_size, num_pics);

%-----
% Start page numbers
j = 1;

% Loop through the images, skipping some based on the interval
for i = 1:step_pics:num_files
    % Read the ith image
    ith_image = imread(file_names(i, :));

    % Convert the image to double format to avoid rounding errors
    ith_image = im2double(ith_image);

    % Store the image data in the image stack
    image_stack(:, :, j, :) = ith_image;
    j = j + 1;

    % Display progress
    fprintf(['#', num2str(i), ' ']);
    if mod((i + 1) / step_pics, 10) == 0
        fprintf('\n');
    end
end

% Loop through each image to calculate focus measures
for i = 1:num_pics
    % Get the ith page (image)
    ith_image = squeeze(image_stack(:, :, i, :));

    % Calculate focus measure using the 'HELM' method
    neighbours = 14;
    focus_im = focusmeasure(im2gray(ith_image), 'HELM', neighbours);

    % Store the focus measure in the focus stack
    focus_stack(:, :, i) = focus_im;
end
```

```

% Display progress
fprintf([num2str(i), ', ']);
if mod((i + 1) / step_pics, 10) == 0
    fprintf('\n');
end
end

%-----
% Erase any erroneous data at edges
trim_edges = 1;

if trim_edges
    edges = ceil(neighbours / 2) + 1;

    % Set focus values at edges to zero
    for i = 1:edges
        focus_stack(i, :, :) = 0;           % upper edge
        focus_stack(y_size + 1 - i, :, :) = 0; % lower edge
        focus_stack(:, i, :) = 0;         % left edge
        focus_stack(:, x_size + 1 - i, :) = 0; % right edge
    end
end

%-----
% Normalize the focus images

% Find the minimum focus value (excluding zeros at edges)
focus_min = min(focus_stack(focus_stack > 0));

% Subtract the minimum focus value to avoid negative values
focus_stack = focus_stack - focus_min;

% Set any negative values to zero
focus_stack(focus_stack < 0) = 0;

% Find the maximum focus value in the entire stack
focus_max = max(focus_stack, [], 'all');

% Normalize the focus stack by dividing by the maximum focus value
focus_stack = focus_stack / focus_max;

%-----
% Find the maximum focus of all images and collapse the stack
disp(' ');
disp('Calculating focus peaks...');

```

```

% Get initial focus peaks and depth map
[focus_peaks, depth_map_1] = max(focus_stack, [], 3);

% Calculate the mean focus value and signal-to-noise ratio (SNR)
focus_mean = mean(focus_stack, 3);

% Calculate SNR by dividing focus peaks by focus mean
focus_snr = focus_peaks ./ focus_mean;

% Clean up any NaN values (divide by zero) and set them to zero
focus_snr(isnan(focus_snr)) = 0;

% Create a figure to display the focus peaks
fig_focus = figure('Color', [.8 .8 .8], 'Name', 'FOCUS');
set(gcf, 'units', 'inches', 'position', [14, 1, 7, 6])

% Display the focus peaks using a colormap
imagesc(focus_peaks)
colormap parula
colorbar

% Set a threshold for focus values
f_threshold = 0;

% Store the original focus stack
focus_stack_old = focus_stack;

```

## Appendix A-3 Smoothing and Fusion

```
%----- SMOOTHING LOOP -----
f_smooth = 1;

% Display the current smoothing range
disp(['Current smoothing range = ', num2str(f_smooth)]);

while f_smooth >= 0
    t_1 = tic;

    % Smooth the focus stack using Gaussian filtering
    focus_stack = imgaussfilt3(focus_stack_old, [f_smooth/2, f_smooth, 2]);

    % Calculate the depth map
    disp('Calculating depth map...');

    % Find the depth at each pixel
    [focus_peaks, depth_map] = max(focus_stack, [], 3);

    show_snr = 1;
    if show_snr
        % Calculate SNR for the smoothed focus stack
        focus_mean = mean(focus_stack, 3);
        focus_snr = focus_peaks ./ focus_mean;

        % Clean up any NaN values (divide by zero) and set them to zero
        focus_snr(isnan(focus_snr)) = 0;

        % Create a figure to display the focus SNR
        fig_focus_snr = figure('Color', [.8 .8 .8], 'Name', 'FOCUS SNR');
        set(gcf, 'units', 'inches', 'position', [14, 1, 7, 6])

        % Display the focus SNR using a colormap
        imagesc(focus_snr)
        colormap cool
        colorbar
        ax = gca;
        ax.CLim = [1, max(focus_snr, [], 'all')];
    end

    % Close previous depth and depth smooth figures if they exist
    if exist('fig_depth_1', 'var')
        close(2);
    end
    if exist('fig_depth_smooth', 'var')
```

```

    close(3);
end

% Create a figure to display the depth map before smoothing
fig_depth_1 = figure('Color', [.8 .8 .8], 'Name', 'DEPTH 1');
set(gcf, 'units', 'inches', 'position', [7, 3, 7, 6])

% Display the depth map
imagesc(depth_map_1)
axis off;
colormap jet

% Create a figure to display the smoothed depth map
fig_depth_smooth = figure('Color', [.8 .8 .8], 'Name', 'DEPTH SMOOTH');
set(gcf, 'units', 'inches', 'position', [13.5, 3, 7, 6])

% Display the smoothed depth map
imagesc(depth_map)
axis off;
colormap jet

%-----
% Calculate the result image
disp('Calculating result image...');

% Initialize the result image
result_image = zeros(y_size, x_size, 3);

% Loop through each image and calculate the result image
for i = 1:num_pics
    % Get the ith image
    image = squeeze(image_stack(:, :, i, :));

    % Create a binary mask based on depth_map
    depth_test = ones(y_size, x_size);
    depth_test(depth_map ~= i) = 0;

    % Convert the binary mask to an RGB format
    depth_test_rgb = repmat(depth_test, [1, 1, 3]);

    % Multiply the image by the depth mask
    add_image = image .* depth_test_rgb;

    % Add the result to the result_image
    result_image = result_image + add_image;
end

```

```

% Close the previous result figure if it exists
if exist('fig_result', 'var')
    close(4);
end

% Create a figure to display the result image
fig_result = figure('Color', [.8 .8 .8], 'Name', 'RESULT');
set(gcf, 'units', 'inches', 'position', [0.1, 3, 7, 6])

% Display the result image
imshow(result_image, 'InitialMagnification', 'fit')

% Prompt the user for new smoothing or stop
t_2 = toc(t_1);
disp(['Elapsed time = ', num2str(round(t_2, 2)), ' s']);

f_smooth = input(['Enter new smoothing? (Enter to stop)... ']);

% If the user does not provide input, set f_smooth to -1 to stop
if isempty(f_smooth) == 1
    f_smooth = -1;
end

end

%-----
% Display the first focus stack image using a colormap
imagesc(squeeze(focus_stack(:, :, 1)))
colormap parula
colorbar

disp(' ');
display('Done. ');
disp(' ');

```