ASSIGNMENT 3 - GROUP 20

Daniël Zee
s2063131

Ngoc Xuan Y Nguyen
s4372840

PART 3.1

(1) CALIBRATION STRATEGY FOR CAMIM01–03

Our goal is to convert pixel distances in CamIm01–03 into physical units using the known geometry of a 1mm reticle bar. This strategy rests on three a-priori assumptions:

- **Known spacing:** the reticle has equally spaced markings 1mm apart (`units_between_peaks`=1).
- **Reticle bar alignment:** The reticle bars are aligned to the vertical axis of the image, so no rotation correction is required (`rot_deg`=0).
- **Reticle bar shape:** Reticle bars are shaped like horizontal lines, with width greater than the text in the image.
- **Uniform sampling:** pixels are square and magnification is constant across the three images.

Based on these assumptions, we automate calibration by:

1. Estimating the slowly varying illumination background via a large diameter SE closing (`background_se_size`).
2. Subtracting the background estimation from the image to enhance bar contrast (black-hat transform).
3. Segmenting the reticle from the background using Isodata thresholding.
4. Extract only the reticle bars via a horizontal line SE closing (`line_se_size`).
5. Extract a one-dimensional profile by integrating over the image rows.
6. Detect the peaks in the one-dimensional profile and calculate the distance between them.
7. Filter out outliers and compute the mean pixel separation and converting it to mm/pixel.

(2) IMPLEMENTATION OF THE MEASUREMENT ALGORITHM

The `measurement_pipeline` function in `IAM_A3_Final.ipynb` implements the above in code:

1. **Load and convert to grayscale:**

```
image = dip.ImageRead(filename)
if image.TensorElements()>1:
    image = dip.ColorSpaceManager.Convert(image,"gray")
```

2. **Background removal:**

```
se = dip.SE(background_se_size,"elliptic")
image_bg = dip.Closing(image,se)
image_clean = image - image_bg
image_clean -= dip.Minimum(image_clean)
```

3. **Segmentation and optional rotation:**

```
image_seg = dip.IsodataThreshold(image_clean)
if rot_deg!=0:
    image_seg = dip.Rotation2D(image_seg, angle=np.deg2rad(rot_deg))
```

4. **Reticle bar extraction**

```
image_lines = dip.Closing(image_seg, dip.SE(line_se_size,"rectangular"))
```

5. **Profile extraction & peak detection:**

```
profile = np.array(image_lines).sum(axis=integrate_dim)
peaks,_ = find_peaks(-profile, distance=peak_distance)
```

6. **Outlier rejection & scale computation:**

```
delta_p = peaks[1:] - peaks[:-1]
filtered = reject_outliers(delta_p, m=1)
mean_dp = filtered.mean()
units_per_pixel = units_between_peaks / mean_dp
```

All structuring-element sizes (`background_se_size`, `line_se_size`) and the peak-distance constraint (`peak_distance`) are directly tied to the known bar geometry, ensuring an objective and reproducible measurement of the pixel scale.

(3) Calibration Results and Summary

Table 1 reports the measured mean inter-marking distances (pixels per 1 mm bar) and the derived scales (mm/px) for CamIm01–03. We furthermore show the image results of every processing step in the pipeline.

Table 1: Measured inter-marking distance and derived scale

Image	Pixels(bar)	mm/px
CamIm01.tif	58.55	0.0171
CamIm02.tif	116.67	0.0086
CamIm03.tif	295.67	0.0034

Figure 1: CamIm01 (a) Estimated background illumination (elliptic SE); (b) Background-subtracted (black-hat) image;

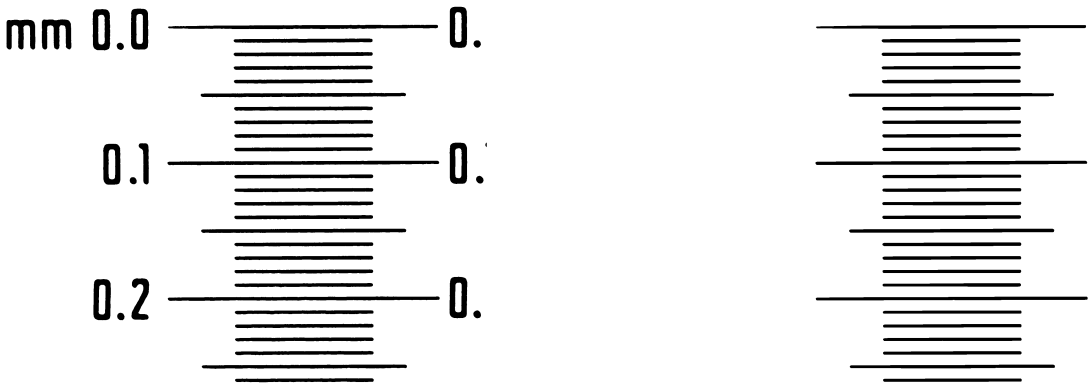

Figure 2: CamIm01 (c) Binary segmentation (Isodata). (d) Reticle bar extraction (line SE)

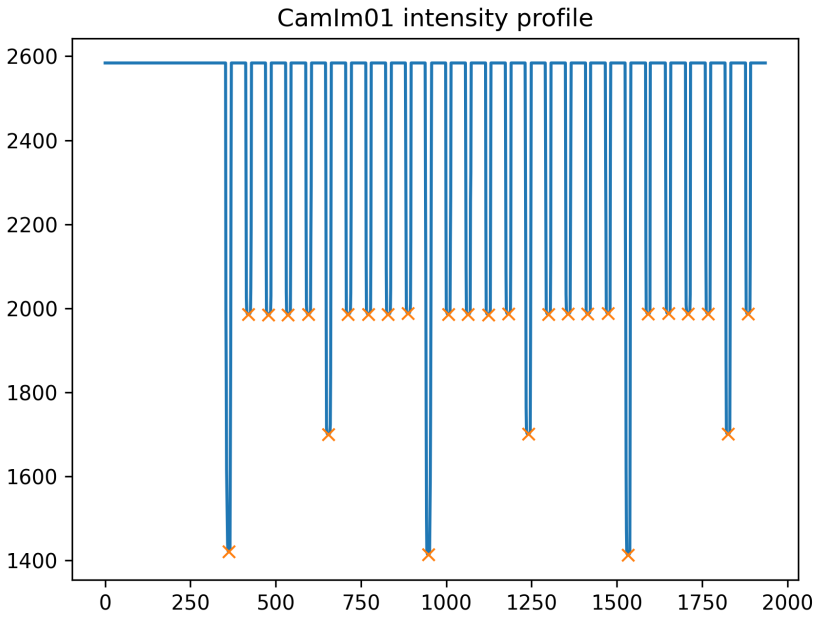

Figure 3: CamIm01 intensity profile: x-axis is pixel column index; y-axis is summed foreground intensity. Orange crosses mark detected minima (bar edges) used to compute mean spacing $\overline{\Delta p} = 58.55$px.

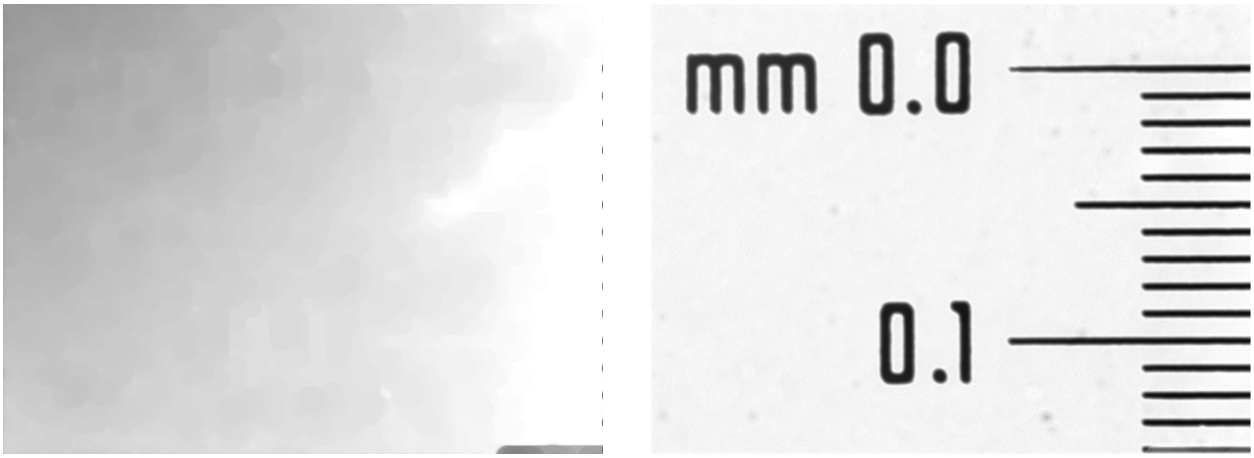

Figure 4: CamIm02 (a) Estimated background illumination (elliptic SE); (b) Background-subtracted (black-hat) image;

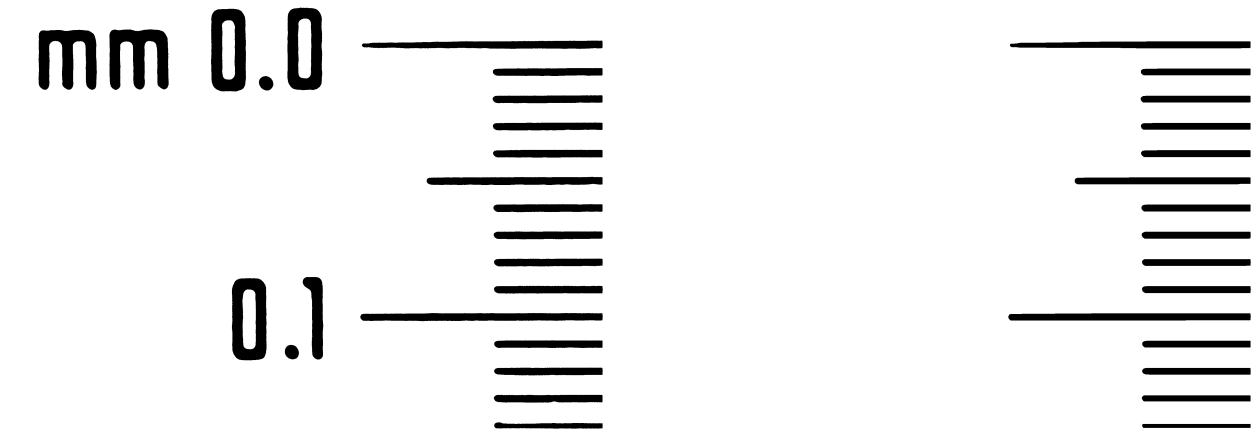

Figure 5: CamIm02 (c) Binary segmentation (Isodata). (d) Reticle bar extraction (line SE)

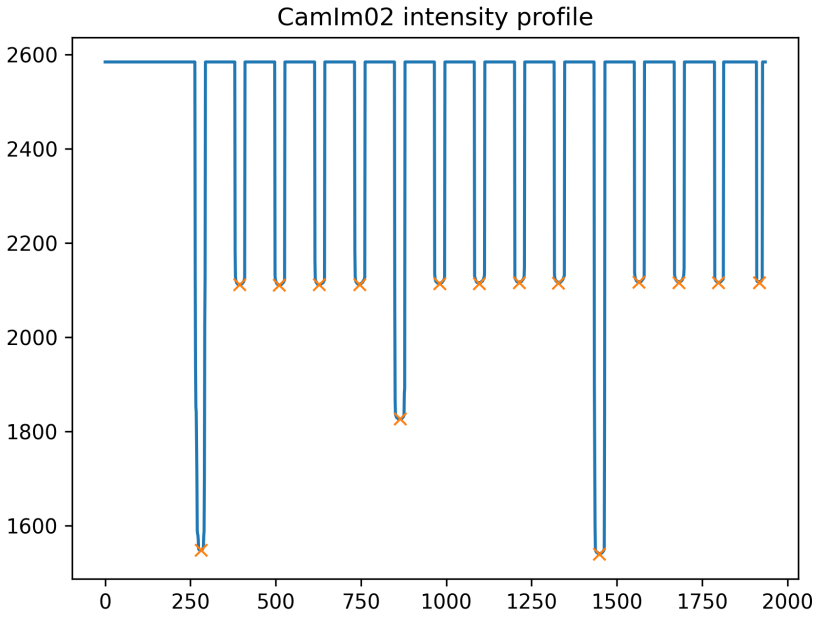

Figure 6: CamIm02 intensity profile: x-axis is pixel column index; y-axis is summed foreground intensity. Orange crosses mark detected minima (bar edges) used to compute mean spacing $\overline{\Delta p} = 116.67$px.

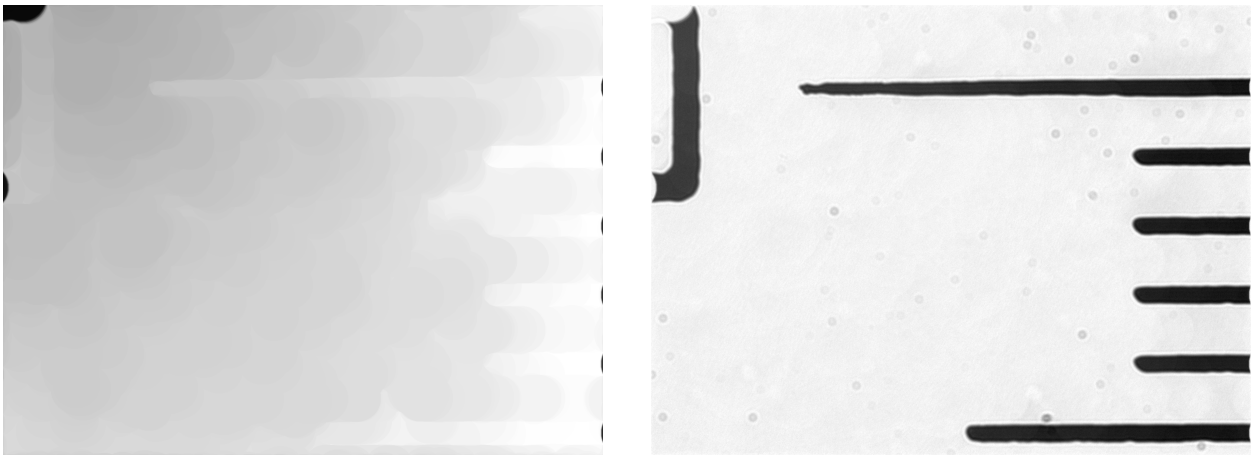

Figure 7: CamIm03 (a) Estimated background illumination (elliptic SE); (b) Background-subtracted (black-hat) image;

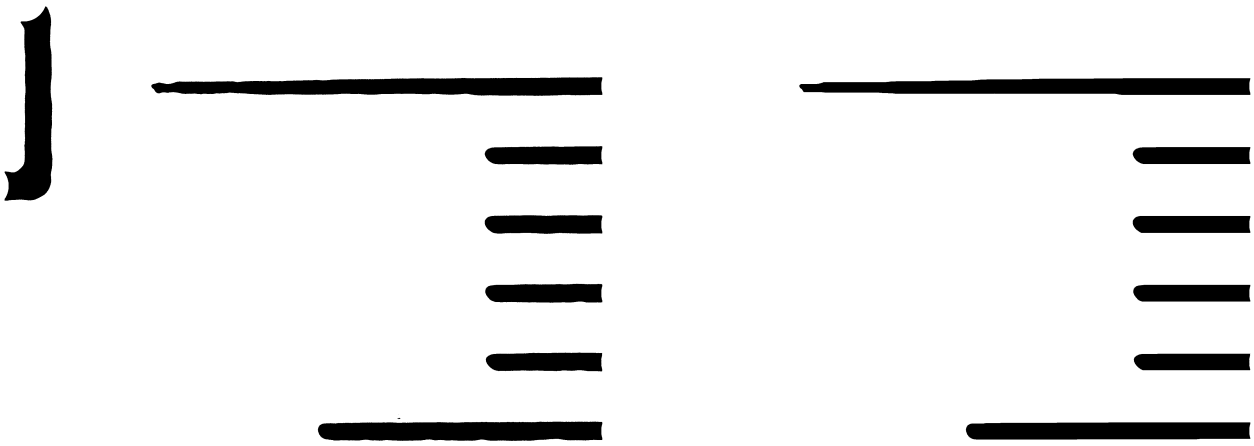

Figure 8: CamIm03 (c) Binary segmentation (Isodata). (d) Reticle bar extraction (line SE)

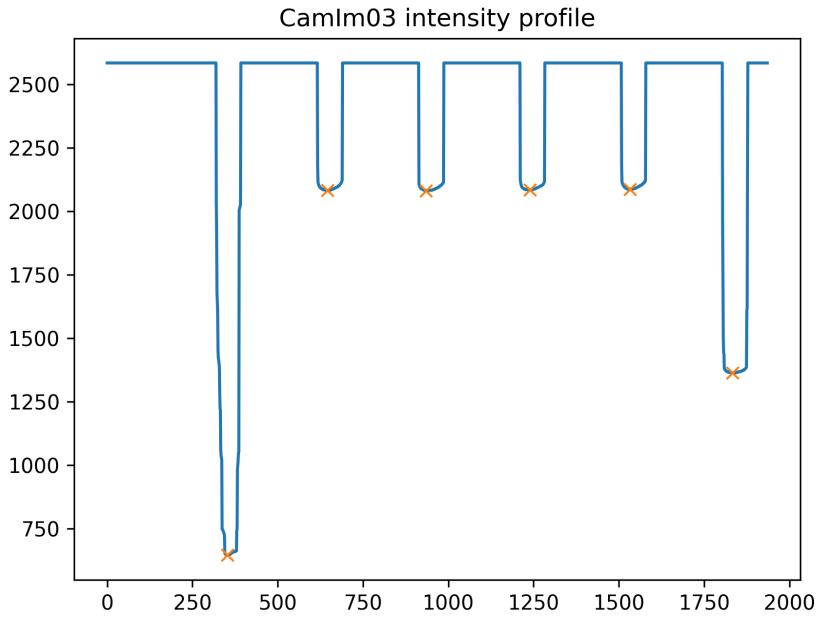

Figure 9: CamIm03 intensity profile: x-axis is pixel column index; y-axis is summed foreground intensity. Orange crosses mark detected minima (bar edges) used to compute mean spacing $\overline{\Delta p} = 295.67$px.

(4) Derived Lens Magnifications

Using CamIm03 (100×) as the reference, we compute the effective magnification for CamIm01 and CamIm02 by

$$\text{mag}_i = \frac{\overline{\Delta p}_{\text{ref}}}{\overline{\Delta p}_i} \times 100,$$

where $\overline{\Delta p}_{\text{ref}} = 295.67$px is the mean peak distance in CamIm03 and $\overline{\Delta p}_i$ the corresponding value in image i. The results are summarized in Table 2.

Table 2: Computed magnification factors relative to the 100× reference (CamIm03).

Image	Computed magnification
CamIm01.tif	19.8×
CamIm02.tif	39.5×
CamIm03.tif	100.0×

The computed values for CamIm01 and CamIm02 (19.8× and 39.5×) closely match the nominal microscope objectives of 20× and 40×, differing by less than 1%. This agreement validates our calibration pipeline and confirms that pixel-to-physical scaling is correctly extracted from the reticle bar measurements.

Part 3.2

(5) ICS Header Interpretation

ICS (Image Cytometry Standard) is a file format for storing multi-dimensional image data along with rich metadata. The '.ics' header for `scale-img.ics` contains the following key–value pairs:

`ics_version:` Version of the ICS format (1.0).

`filename:` Base name of the image data ("scale-img").

`layout parameters:` Number of image parameters or dimensions (3).

`layout order:` Order of indexing: bits, then x, then y.

`layout sizes:` Length along each dimension: 32 bits per pixel, 256 pixels in x, 256 pixels in y.

`layout coordinates:` Coordinate system ("video").

`layout significant_bits:` Number of meaningful bits (32).

`representation format:` Data type format ("real" = floating point).

`representation sign:` Signedness ("signed").

`representation byte_order:` Byte order (4-3-2-1, i.e. big-endian to little-endian mapping).

`representation SCIL_TYPE:` Internal SCIL image type identifier ("f2d" = 32-bit float).

Each field corresponds directly to metadata required for correctly interpreting and reading the ICS image: versioning, dimensionality and ordering, coordinate conventions, bit-depth, and numeric data representation.

(6) TIFF vs. ICS Numeric Comparison

We compared each pixel in the original TIFF (`scale-img.tif`) and the reloaded ICS+IDS pair (`scale-img.ics` + `.ids`) and found:

Table 3: Pixel-value comparison between TIFF and ICS

Statistic	Value
Mean absolute difference	0.0000
Std. absolute difference	0.0000
Max absolute difference	0.0000

These zero-difference metrics indicate that every pixel value in the ICS image matches the TIFF exactly, confirming a bit-perfect, lossless conversion. Although the original TIFF stores 8-bit integers, promoting to 32-bit floats in the ICS format preserves all values without rounding error.

The file sizes are:

Table 4: File sizes of TIFF and ICS+IDS

Format	Size (bytes)
TIFF	65 769
ICS + IDS	262 416

The ICS+IDS representation requires roughly four times more storage, owing to its separate index file and 32-bit floating-point data layout, despite maintaining identical pixel values.

PART 3.3

(7) PREPROCESSING AND PEAK DETECTION FOR `SCALE-IMG.ICS`

To robustly detect the reticle bar edges, we selected a large elliptical structuring element for background estimation (`background_se_size`=10) and a narrow rectangular SE for reticle bar extraction (`line_se_size`=(8,1)). The rationale is:

- **Elliptic SE (size=10):** Captures the slowly-varying illumination field without encroaching on the bar edge width.
- **Line SE (8×1):** Filter out the vertical tick marks while leaving the horizontal tick marks in place.

Figure 10 - 11 illustrates the effect of these choices:

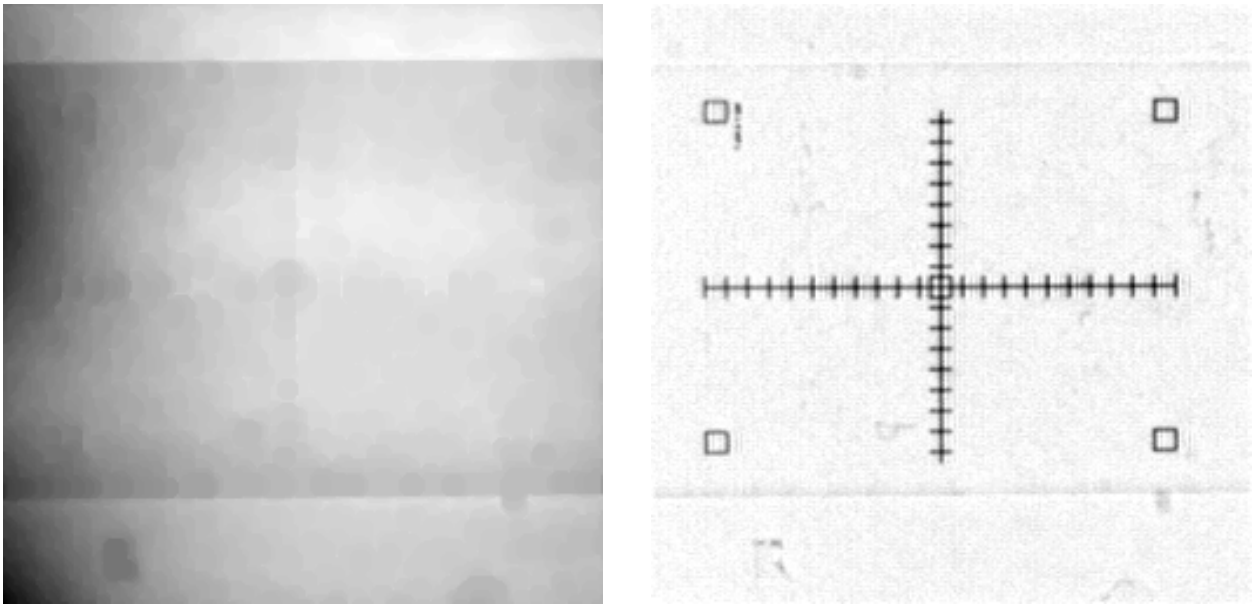

Figure 10: (a) Estimated background (elliptic SE); (b) Background-subtracted (black-hat) image.

Panel (a) shows that the chosen SE size removes uneven shading while preserving bar contrast; panel (b) demonstrates high-contrast bar edges in the black-hat image, ready for thresholding. Subsequent segmentation (c) and line-SE closing produce a clean binary mask of the bars (d), confirming that these parameters effectively isolate the features of interest.

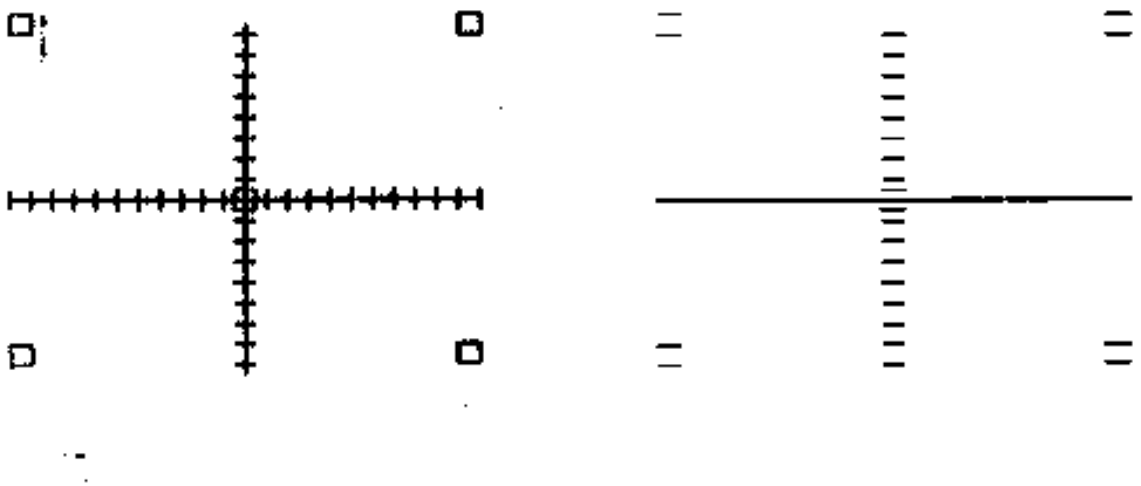

Figure 11: (c) Binary segmentation (Isodata); (d) Reticle bar extraction (line SE)

(8) PIXEL SIZE SUMMARY

Applying this parameter set yields the following calibration:

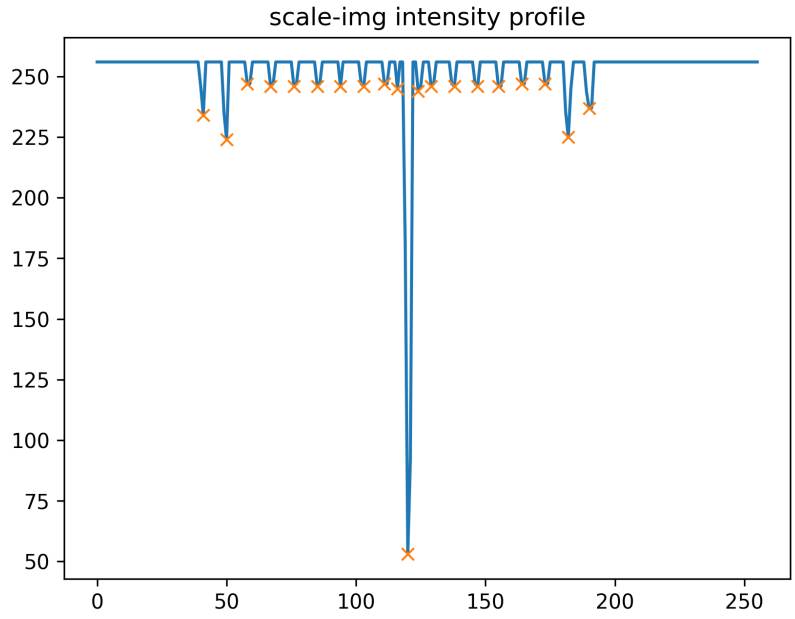

Figure 12: scale-img intensity profile: x-axis is pixel column index; y-axis is summed foreground intensity. Orange crosses mark detected minima (bar edges).

Figure 12 shows the intensity profile over the rows of the reticle bar extraction image. We can clearly see the contributions of the large middle line and the bars in the image corners. As these only effect the magnitude of the peaks, and not their positions, they do not effect the calculated distance between peaks.

Table 5: Derived pixel-to-unit conversion for `scale-img.ics`

Image	Pixels per 1 unit bar	Units per pixel
scale-img.ics	8.73	0.1145

Table 5 summarizes the mean inter-mark spacing (8.73px) and the resulting scale (0.1145 units/px).

PART 3.4

(9) CALIBRATION OF CAMIM04

Applying the identical pipeline to `CamIm04.tif` yielded:

Approach and differences to 3.3.7

- **Color → Grayscale:** CamIm04 is 3-channel sRGB, so we first converted to single-channel float, whereas `scale-img.ics` was already scalar.
- **Elliptic SE (size=15):** A larger diameter is needed to estimate the background illumination, as the image if of a higher resolution.
- **Image tilt (1 degree):** As the original image is not well aligned to the horizontal axis, we tilt the image to be better aligned.
- **Line SE (1×32):** The size of the line SE is increased because the reticle bars use more pixels. The line SE is changed to a vertical line due to the changed orientation of the reticle bars compared to the previous images.

Key intermediates Figures 13-15 show the intermediate results of the processing steps.

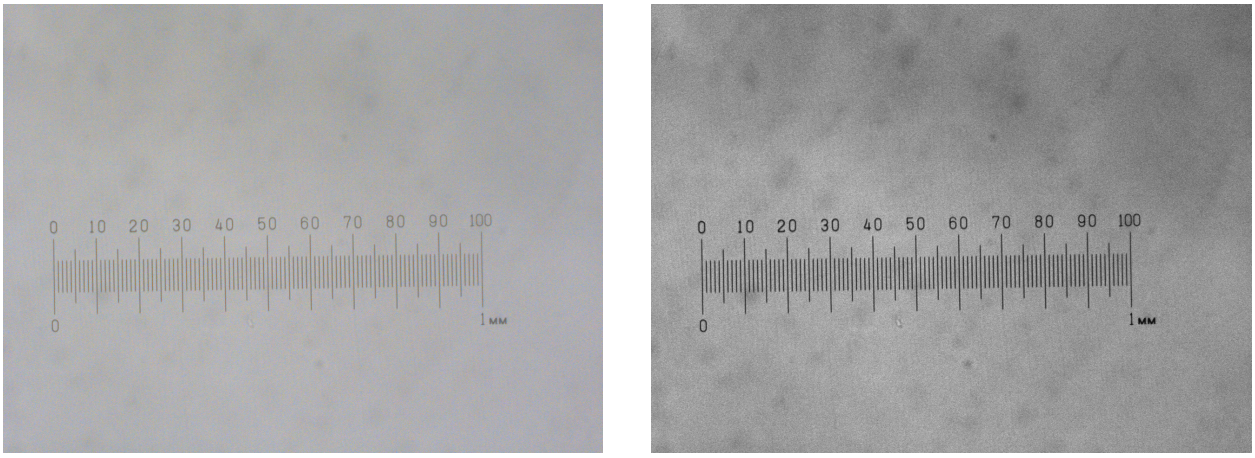

Figure 13: (a) Raw color image; (b) converted to grayscale (SFLOAT).

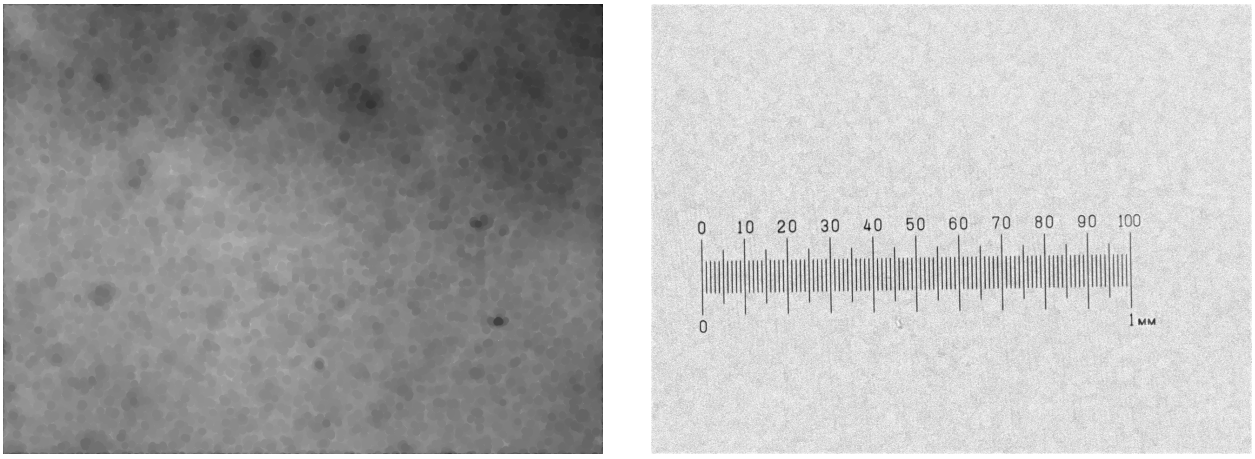

Figure 14: (c) Estimated background (elliptic SE size=10); (d) black-hat cleaned.

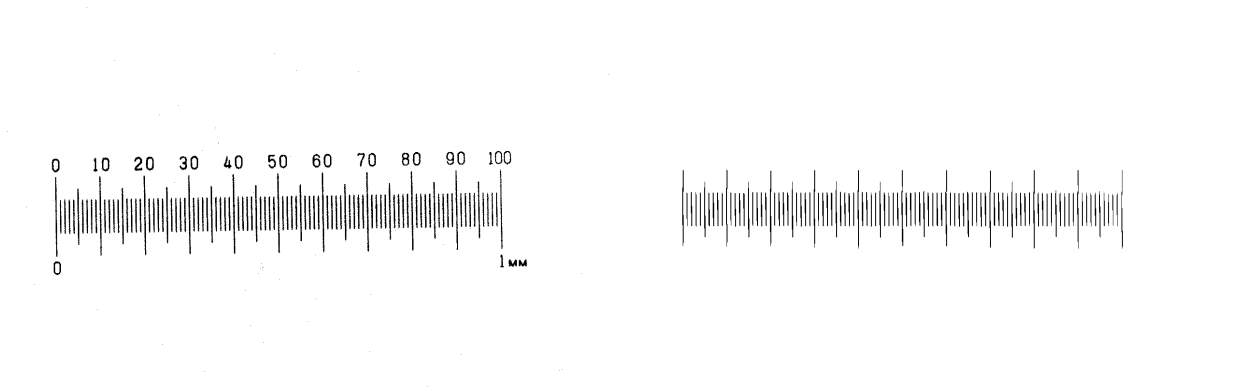

Figure 15: (e) Binary segmentation (Isodata); (f) Rotation followed by reticle bar extraction (line SE)

(10) COMPUTED OBJECTIVE MAGNIFICATION

Profile extraction and peak detection Figure 16 shows the intensity profile over the columns of the reticle bar extraction image. We integrate over the columns instead of the rows because we now have vertical reticle bars.

Numeric results

- Mean inter-peak spacing: $\overline{\Delta p} = 9.00$px
- Pixel scale: $1\,\mathrm{mm}/9.00\,\mathrm{px} = 0.1111\,\mathrm{mm/px}$

Table 6: CamIm04 pixel calibration

Image	Pixels(bar)	mm/px
CamIm04.tif	9.00	0.1111

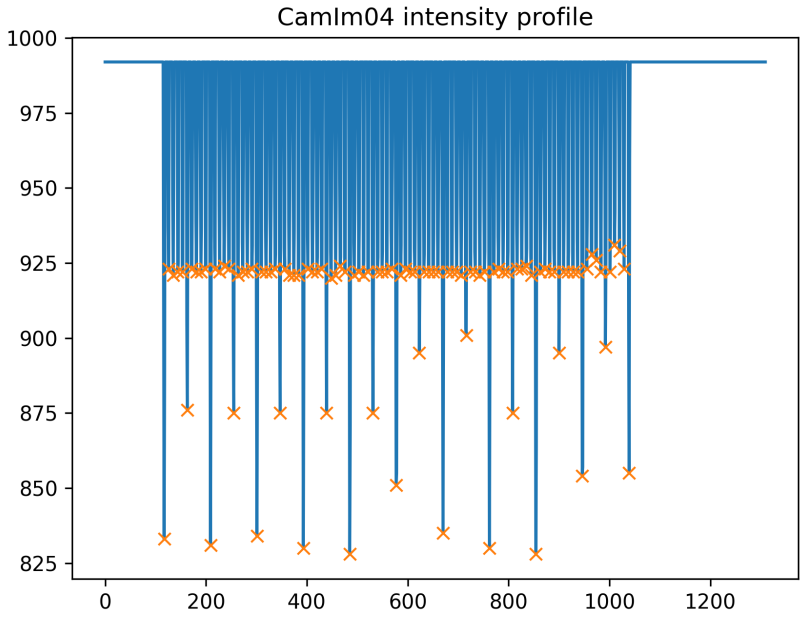

Figure 16: CamIm04 intensity profile: x-axis is pixel column index; y-axis is summed foreground intensity. Orange crosses mark detected minima (bar edges).

Using CamIm03 (100× reference, $\overline{\Delta p}_{\text{ref}} = 295.67$px):

$$M = \frac{\overline{\Delta p}_{\text{ref}}}{\overline{\Delta p}_{\text{CamIm04}}} \times 100 = \frac{295.67}{9.00} \times 100 \approx 3.0 \times .$$

Table 7: Computed magnification for CamIm04

Image	Computed mag (×)
CamIm04.tif	3.0×