

Wafer defects detection using openCV

Nadav Lehrer

I. INTRODUCTION

In this task, we were asked to identify defects in a silicon wafer on which a circuit was lithographically etched. This task is to be performed by comparing an image of a non-defective section of the wafer (referred to as the reference image or ref-img) with the image of a "near identical" section on a potentially defected wafer (location wise "near identical", not value wise). The test set given consists of 3 wafer sections, of which two sections manifest defects and the remaining section is of a non-defective wafer section.

It is important to note that while we were asked to deliver an algorithm which is as general and robust as possible, the scarce train set will almost surely guaranty an over-fit to the train set. The algorithm is not "learning based" hence no real over-fitting occurs, however the meaning is practically the same – the algorithm is designed to perform well on the given examples however it was not tested on a test set (due to a lack of one), and will probably won't perform with a 100% accuracy in the general case.

II. ASSUMPTIONS

In this section we will introduce the assumption that were laid during the completion of the task. These assumptions were tested and verified to be reasonable.

1. The noise is Gaussian and uncorrelated (white Gaussian noise):

The images received contained high amounts of noise which needed to be removed for the algorithm to perform well. Hence, a few different noise reduction filters (some are low pass filters) were tested, among which is the Gaussian Blur (and a non linear variant of it, the Bilateral Gaussian Filter which preserves edges), and the median filter (another non linear filter). These filters best suit Gaussian noise and salt-and-pepper noise respectively. The filtering results were a clear-cut win for the Gaussian filters, which implies the noise to be Gaussian.

2. The reference image and the inspected image are related by an affine transformation:

As the exercise states, the two images received by the algorithm as an input may not be aligned, hence we may use only an affine transformation estimation. This assumption was tested using a perspective transformation estimation (i.e. estimating the homography transformation), only to produce worse results than the affine transformation. The homography matrix which was estimated also showed extremely small perspective factors (the two left elements of the third row), which imply the transformation is indeed an affine one.

III. SOLUTION

The algorithm presented draws its idea from [1]-[4] in which a similar task is implemented on printed circuit boards (PCB's). The papers cited, all implement a similar image processing pipeline on the two given images, followed by image subtraction to reveal the defects. However, the task given here, presents a more difficult challenge, as detecting defects in silicon wafers is harder due to the nature of the defects (they may occur not only in the etched conducting channels but also on flat surfaces) and due to the small size of inspected regions, which demands the use of high-resolution-high-focal-length cameras, give rise to high amounts of noise. For these reasons, a more sophisticated pipeline is planned which basically consists of two separate pipelines, each is intended for

detection of different sections within the wafer. The two pipelines, referred to as the Separated/Joint pipelines, both receive the reference and inspected images after alignment of the reference image to the perspective of the inspected one. This alignment is done by first detecting features in both images using the ORB detector in OpenCV followed by matching of these features between the two images, and finally an affine transformation matrix estimation using internal OpenCV methods.

The first ("separate") pipeline detects defects in the channels by first removing noise from each of the images, followed by Otsu's Thresholding (OTH) and taking the absolute value of the thresholded images subtraction result. The final result of this stage may contain the thin borders of the channels detected as defects (they appear due to non-perfect homography estimation, interpolation after transformation and noise). To deal with this issue we perform a morphological closing operation with an elliptical 3X3 kernel. A Better approach is to apply two following morphological closings using a vertical 3X1 box kernel and a horizontal 1X3 kernel (to close the one channel borders). However, this filter deals poorly with defects on areas outside the channels, due to the thresholding stage which renders all the area outside the channels as white, thus eliminating a chance of detection in these areas. To deal with this issue, we present a second ("joint") pipeline which produces a mask that detects the defects outside the channels.

The second ("joint") pipeline first subtracts and takes the absolute value of the two images (after alignment), only to send this result to a further refinement method, where we perform a non-local means denoising (NLMDN) [5] followed by OTH. The result of the stages so far, produces a mask which indeed contains the defects located outside the channels, however it includes vast areas of channel borders detected as defects (due to the same reasons stated in the previous paragraph). To face this issue, we keep track of all the pixels which are to be ignored. These pixels include the channels (and a margin around them) obtained by looping through the dilated OTH ref-img from the first pipeline and saving the white conducting-channel pixels into a hash-set for a later O(1) time access. To be completely sure we detect defects that are located solely on the intersection region (common surface region) of the reference and inspected image, we also ignore the pixels of the non-intersecting area, which are obtained easily in the image alignment method. Finally, we perform a morphological closing operation with a 2X2 ellipse kernel to reduce noise. This leads to a final assumption: the defects outside the channels can be detected only if they fit this 2X2 kernel, otherwise they are treated as noise.

Finally, the two masks are combined and morphologically closed to produce the desired output.

IV. RESULTS

In this section we will present the results on the given data set including some intermediate interesting result. First, we will start by presenting the algorithm described above, via a flow chart/block diagram, and proceed by showing the results.

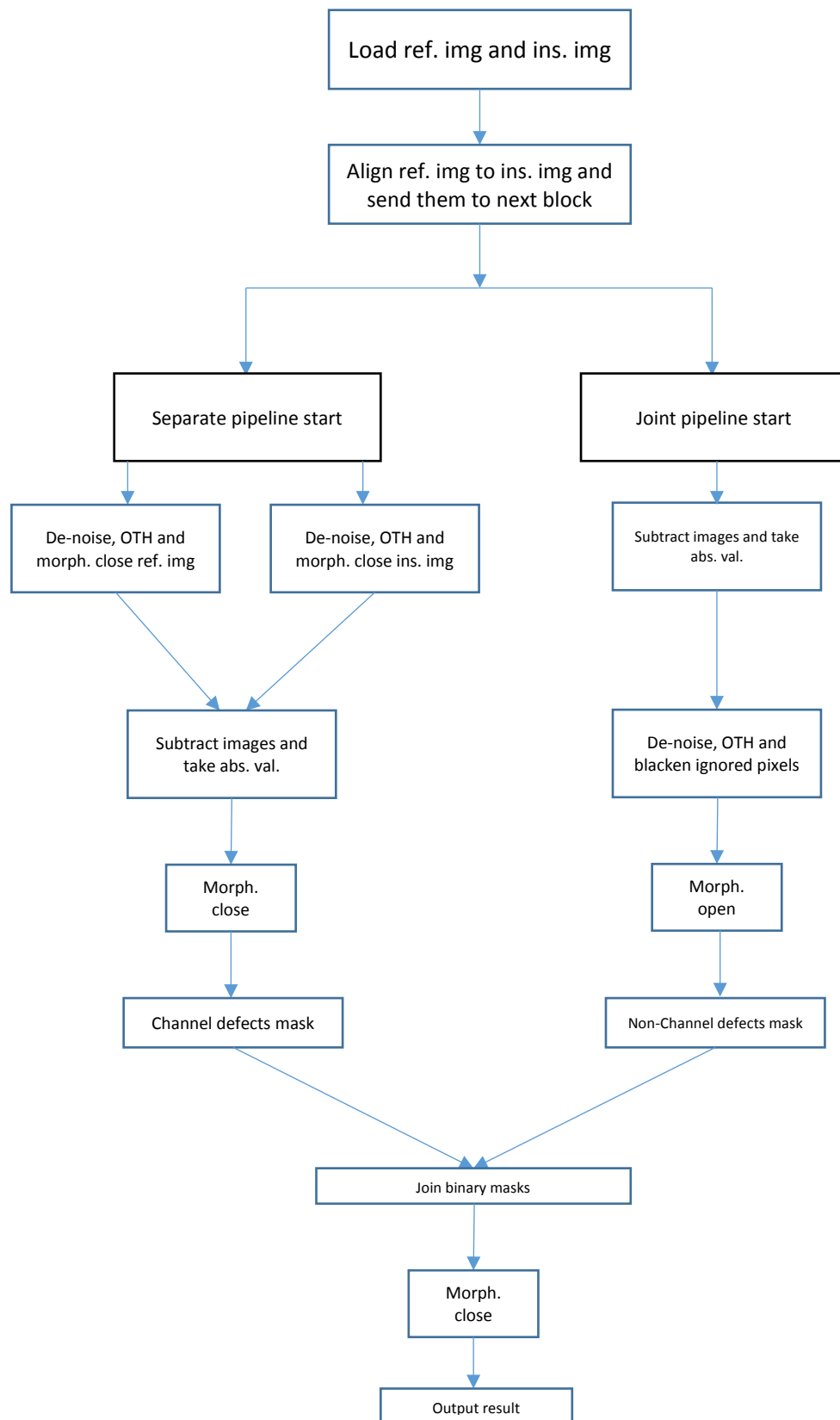


figure 1. The algorithm depicted by a block diagram

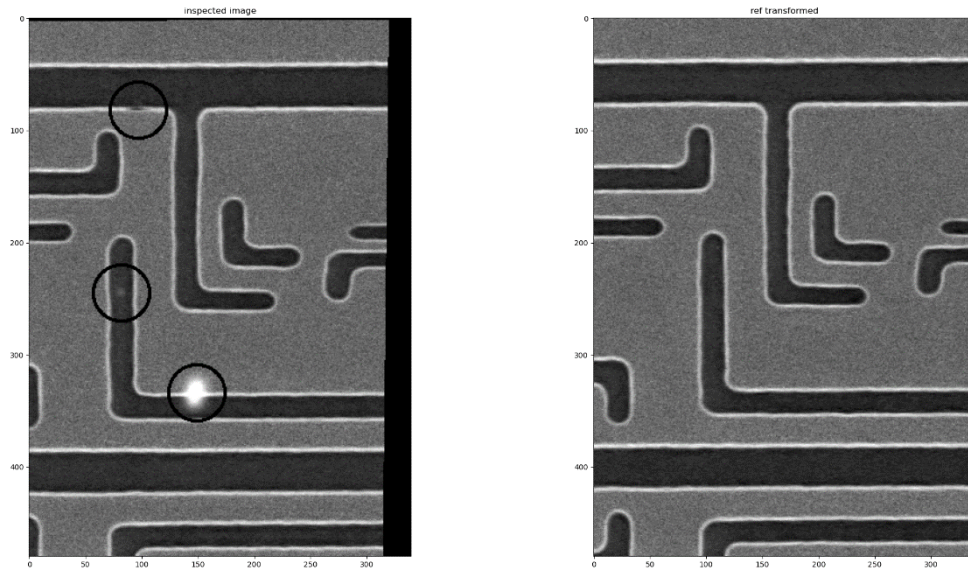


figure 2: Case 1 – on the left we see the inspected image after alignment to the perspective of the reference image (on the right). We are also able to see the defects we intend to detect, circled on the inspected image.

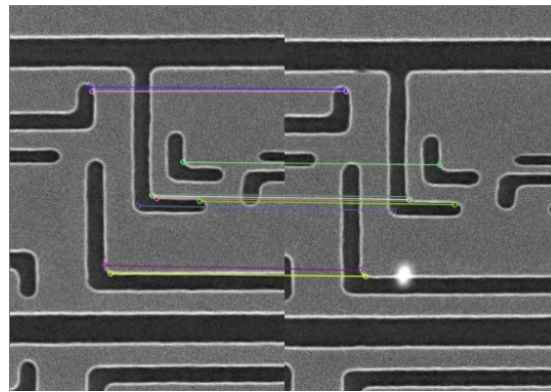


figure 3: Case 1- the feature matches of the orb detector on the first case example.

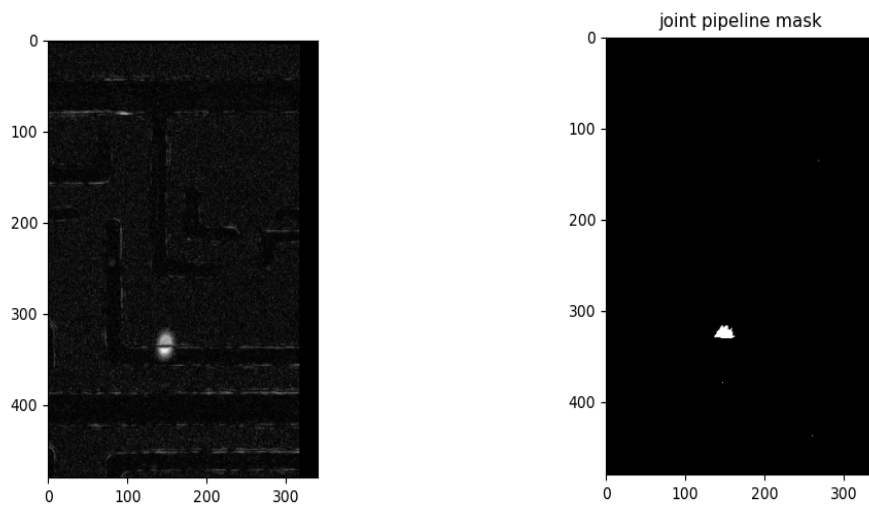


figure 4: Case 1- on the left we can see the absolute value of the subtraction-after-alignment result and the final mask yielded from it on the right.

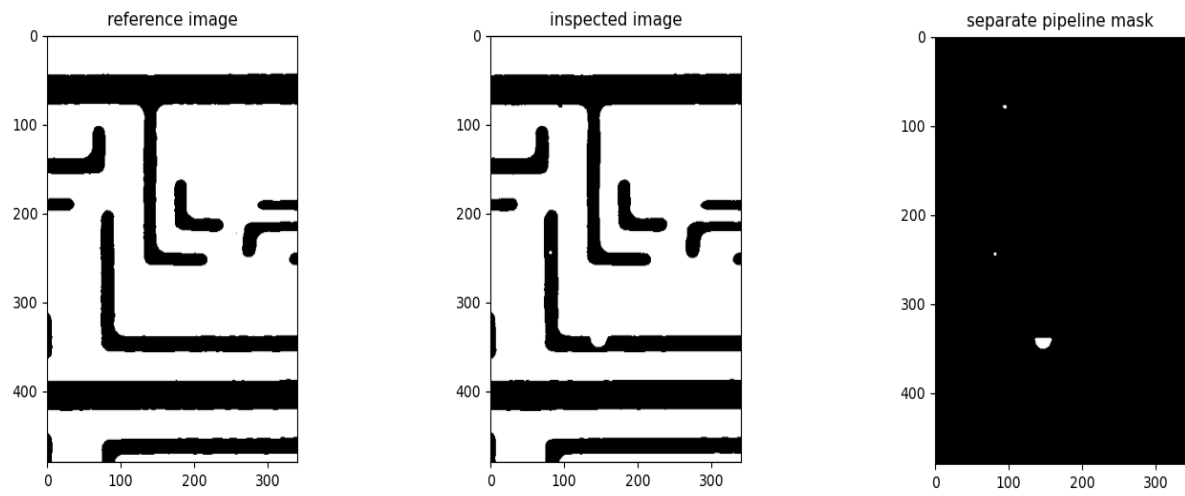


figure 5: Case 1 – here we can see the results from the separate pipeline in which we get the mask of channel defects (on the right) from the

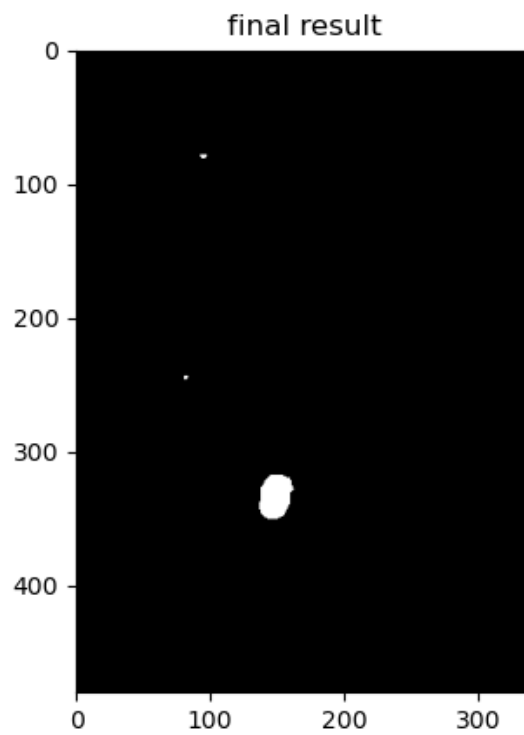


figure 6: Case 1 – the final result yielded by joining the two rightmost images of the last two figures (the joint and separate pipelines results).

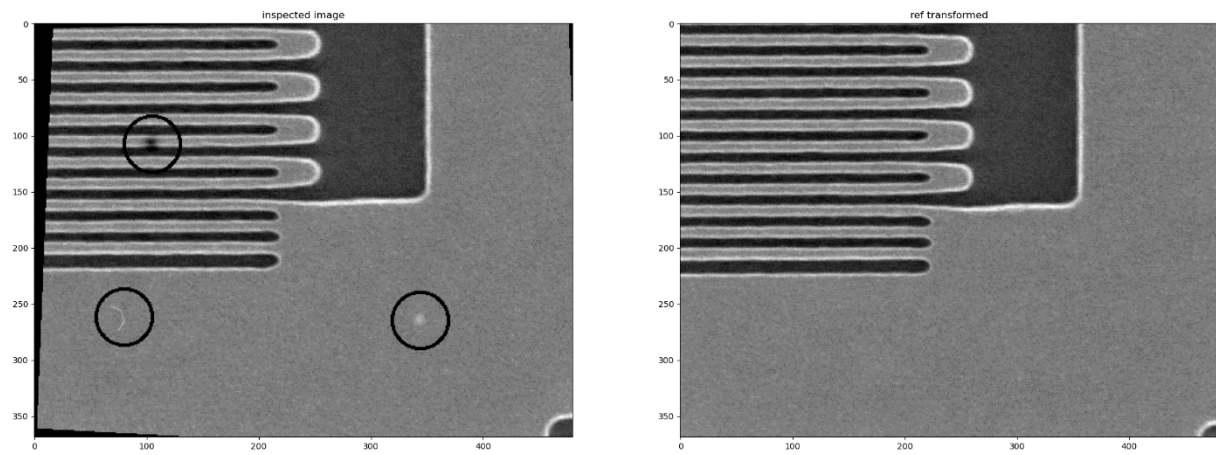


figure 7: Case 2 – on the left we see the inspected image after alignment to the perspective of the reference image (on the right). We are also able to see the defects we intend to detect, circled on the inspected image.

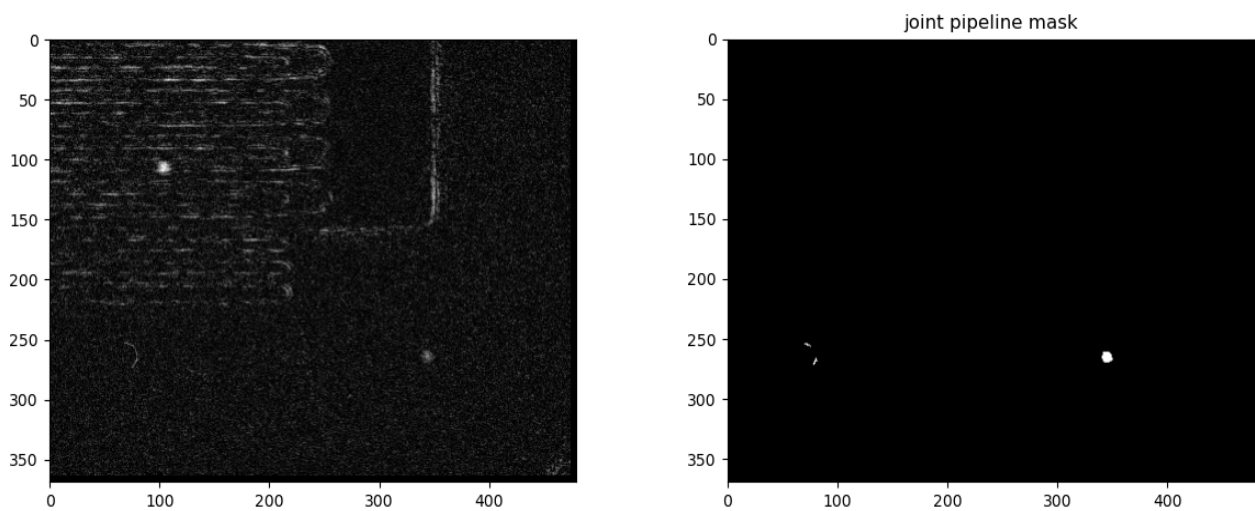


figure 8: Case 2 - on the left we can see the absolute value of the subtraction-after-alignment result and the final mask yielded from it on the right.

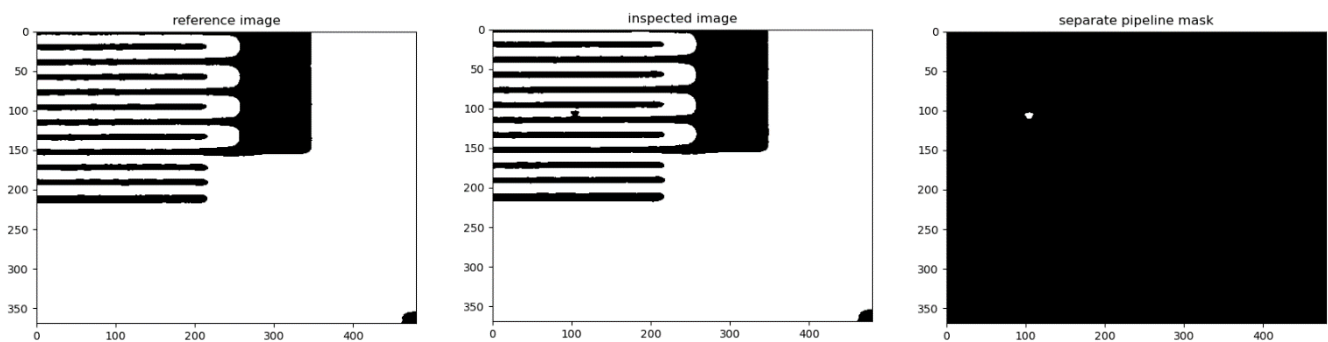


figure 9: Case 2 – here we can see the results from the separate pipeline in which we get the mask of channel defects (on the right) from the

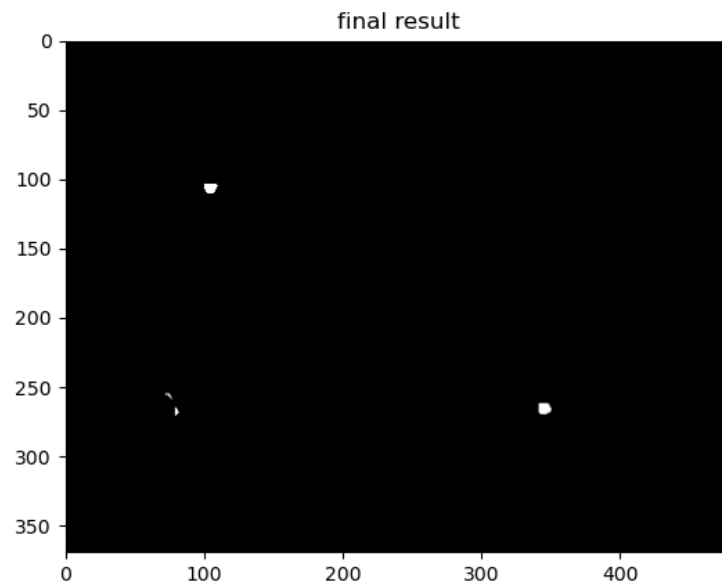


figure 10: Case 2 - the final result yielded by joining the two rightmost images of the last two figures (the joint and separate pipelines results).

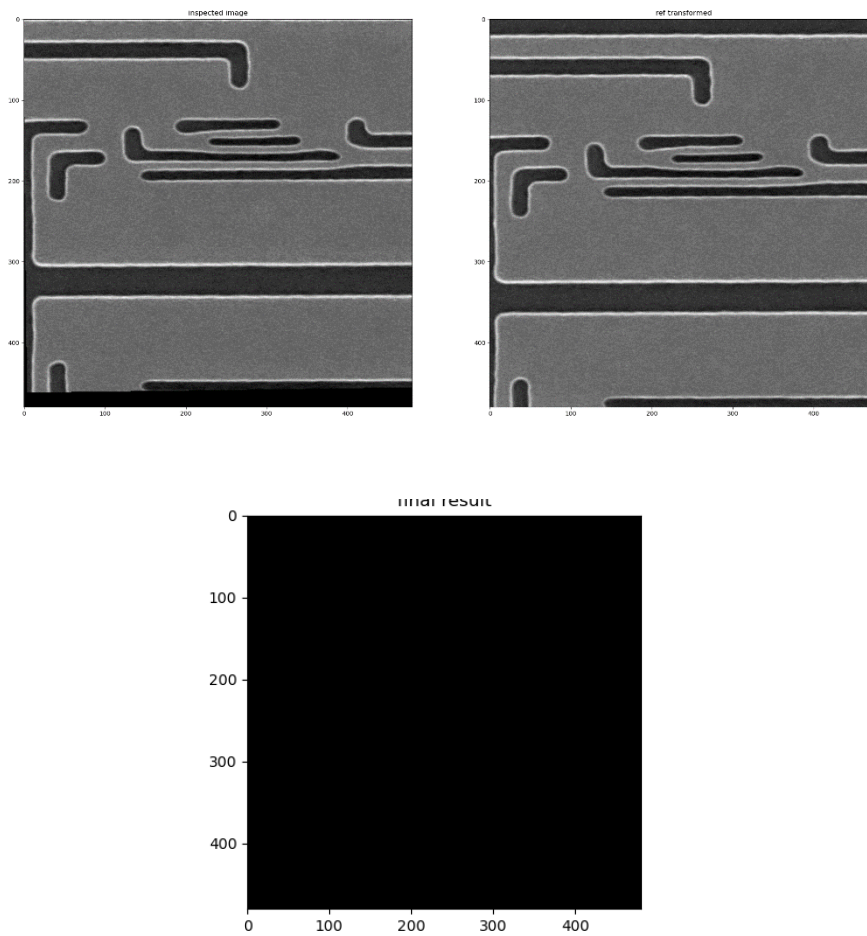


figure 11: Case 3, from the top left and clockwise: 1. the inspected image after alignment to the reference image perspective. 2. the reference image. 3. the combined final result which is the output of the algorithm. This test case had no defects hence no use in showing the intermediate results.

V. REFERENCES

- [1] Ibrahim, Ismail, et al. "A printed circuit board inspection system with defect classification capability." *Int. J. Innovative Manage. Inf. Prod* 3.1 (2012).
- [2] Chauhan, Ajay Pal Singh, and Sharat Chandra Bhardwaj. "Detection of bare PCB defects by image subtraction method using machine vision." *Proceedings of the World Congress on Engineering*. Vol. 2. 2011.
- [3] Ce, Win. "PCB defect detection USING OPENCV with image subtraction method." *2017 International Conference on Information Management and Technology (ICIMTech)*. IEEE, 2017.
- [4] Nayak, Jithendra PR, et al. "PCB Fault detection using Image processing." *IOP Conf. Series: Materials Science and Engineering*. Vol. 225. 2017.
- [5] Buades, Antoni, Bartomeu Coll, and Jean-Michel Morel. "Non-local means denoising." *Image Processing On Line* 1 (2011): 208-212.