1 Method Description

The method consists out of 5 consecutive blocks, each described in a separate subsection below.

1.1 Alignment

The goal of this block is to find a translation that geometrically aligns the reference image with respect to the inspection image (A more complete solution would also compensate rotations and scaling). First, each image is denoised using a small 4×4 median filter. Note that this filter may remove defects from the inspection image, however this is fine as the filter output is used only for alignment. Next, we calculate the x and y gradients for each image. The optimal translation is then found by selecting the peak in the sum of the squared cross correlations of the two gradient images. The reason we chose to apply the cross correlation on the gradients, and not on the images directly, is that in this block color matching had not been applied yet, and the gradients, which are less sensitive to such mismatches, gave superior performance. However, the gradients are sensitive to noise, and therefore the median filter was first applied. Both the gradient calculation and the cross correlation, are efficiently implemented in the frequency domain.

1.2 Reference Image Denoising

It is apparent that even for perfectly aligned and color-matched reference and inspection images, there are differences between them that originate from noise, and not defects. Such noise is characterized by its spatial whiteness (as opposed to the defects pixels which seem to be clustered). The goal of this step is to denoise the images, such that the noise variance of the difference between the images would decrease. The denoised version of the inspection image is used for the color matching block, however it is not used for the later blocks, as it can eliminate thin defects. The denoising method used is known as total variation denoising [1], which essentially attempts to find a piece-wise constant image that well approximates the given image, using a variant on LASSO:

$$\text{Denoise}(I) := \arg\min_{I'} \|I' - I\|_2^2 + \lambda \left(\|\nabla_x I'\|_1 + \|\nabla_y I'\|_1 \right)$$

While the piece-wise constant prior is often used for natural images, it seems especially suited to these images. A common issue in regularization based denoisers is choosing the regularization factor λ. Here, it is chosen automatically by choosing the value that minimizes the error between the denoised reference and the inspection images, as measured by the median absolute difference. The implementation I used was downloaded from here.

1.3 Color Matching

The goal of this step is to compensate illumination mismatch between the two images using an affine transformation. The scale and offset were found using RANSAC (motivated by my discussion with Asaf in the first interview), using subsets of size 100, maximal distance of 1/4. Another output of this block is an initial, rough, defected-pixels detection, which correspond to pixels that do not agree with the linear model (outliers).

1.4 Difference Map Calculation

The aligned, colored-matched and denoised images are now subtracted from each other, with the resulting image denoted by D_1. Pixels with a high difference (in absolute values), are more likely to correspond to defects. However, there are high differences in other pixels as well, from two main causes. The first, is the presence of white noise in the inspection image. This issue is addressed in the next block. The second cause is mismatch in value in places with high gradient. To mitigate this effect, we define D_2 as follows:

$$D_2 = D_1^2 \exp\left(-\left(\frac{G}{\sigma}\right)^2 \right)$$

where G is the magnitudes of the gradient of the reference image. The value of σ controls the attenuation of pixels with high gradient. It is chosen automatically as 10 times the median of G. Note that this "soft" approach allows defects to be detected on edges as well, but they are required to be more significant compared to non on edge defects.

1.5 Clustering

D_2 is thresholded, by a threshold value that is calculated automatically using the rough outlier detection from the color matching block. Then, we look for clusters of pixels that are above the threshold. The goal of this clustering is to filter out false positives by their spatial sparseness. The clustering algorithm I used is DBSCAN [2]. Clusters are discarded according to two tests: first, if the number of pixels they contain is less than 10. Second, if the summation of the pixel values (according to D_2) within the cluster is smaller than 0.3.

2 Results and Illustrations

The results are displayed in figures 1, 2 and 3. The dynamic range of each displayed image was chosen such that the highest pixel value is white, and the lowest pixel value is black.

References

[1] Antonin Chambolle, Vicent Caselles, Daniel Cremers, Matteo Novaga, and Thomas Pock. An introduction to total variation for image analysis. *Theoretical foundations and numerical methods for sparse recovery*, 9(263-340):227, 2010.

[2] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

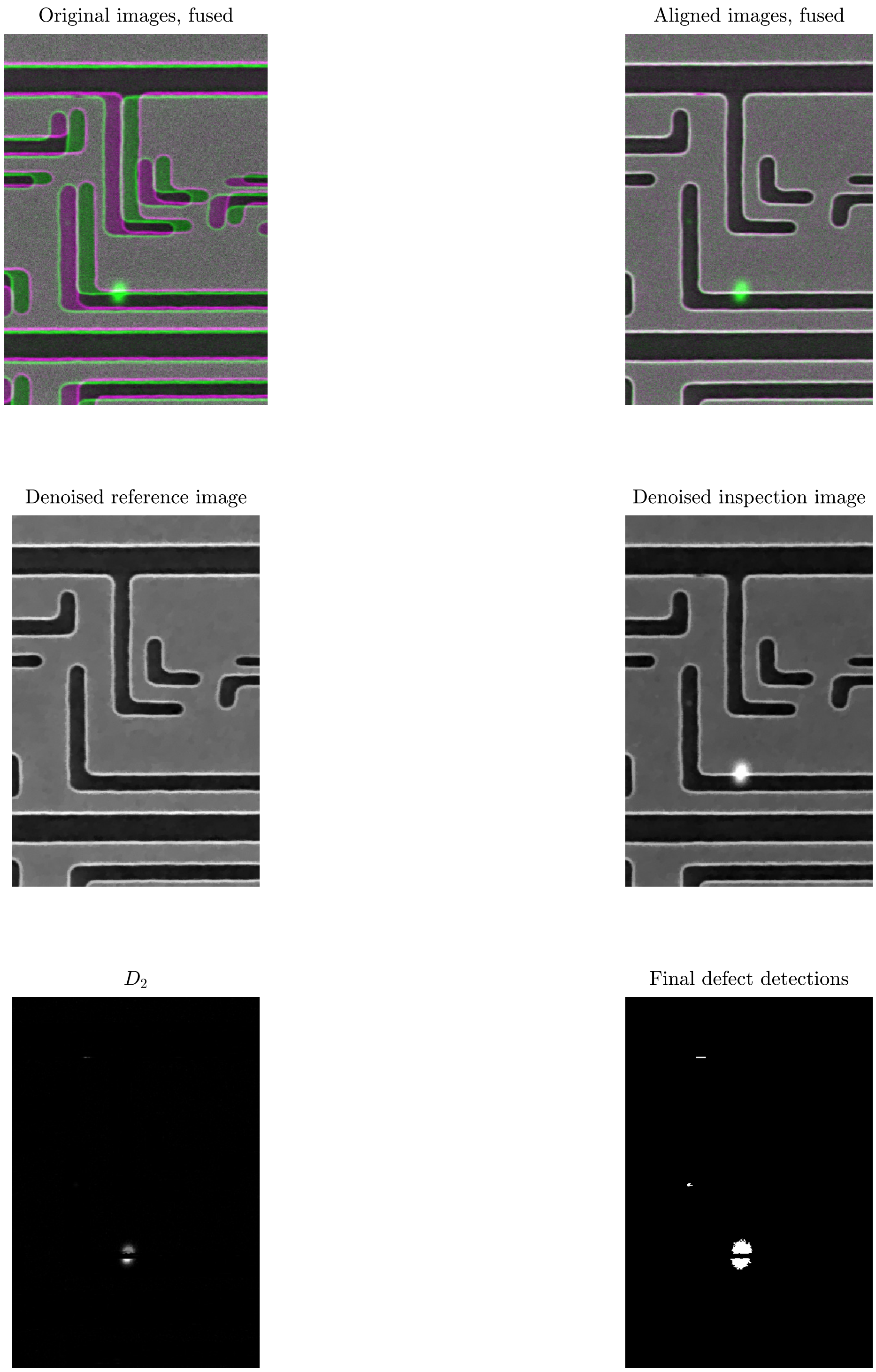

Figure 1: Final and intermediate results of case 1.

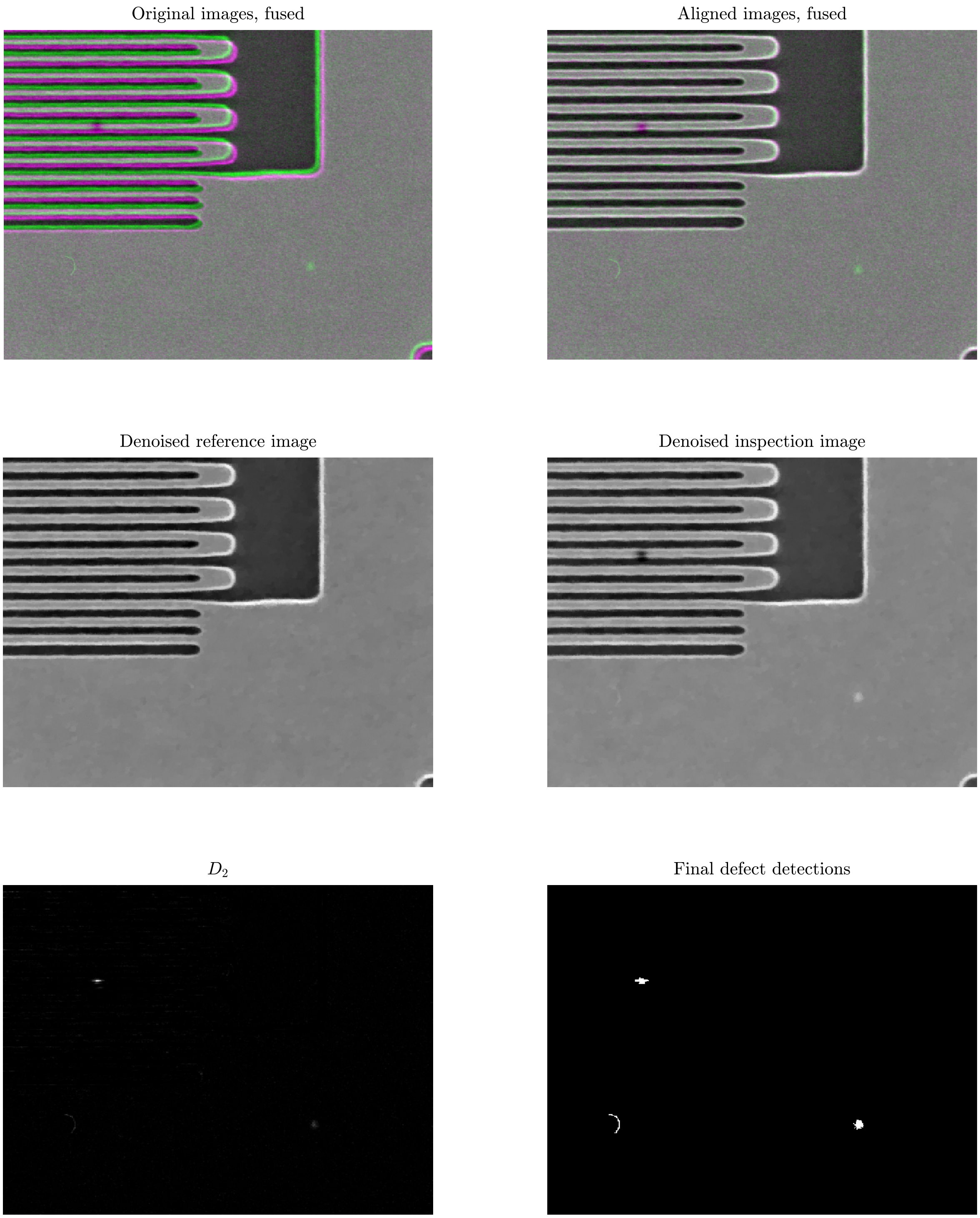

Figure 2: Final and intermediate results of case 2.

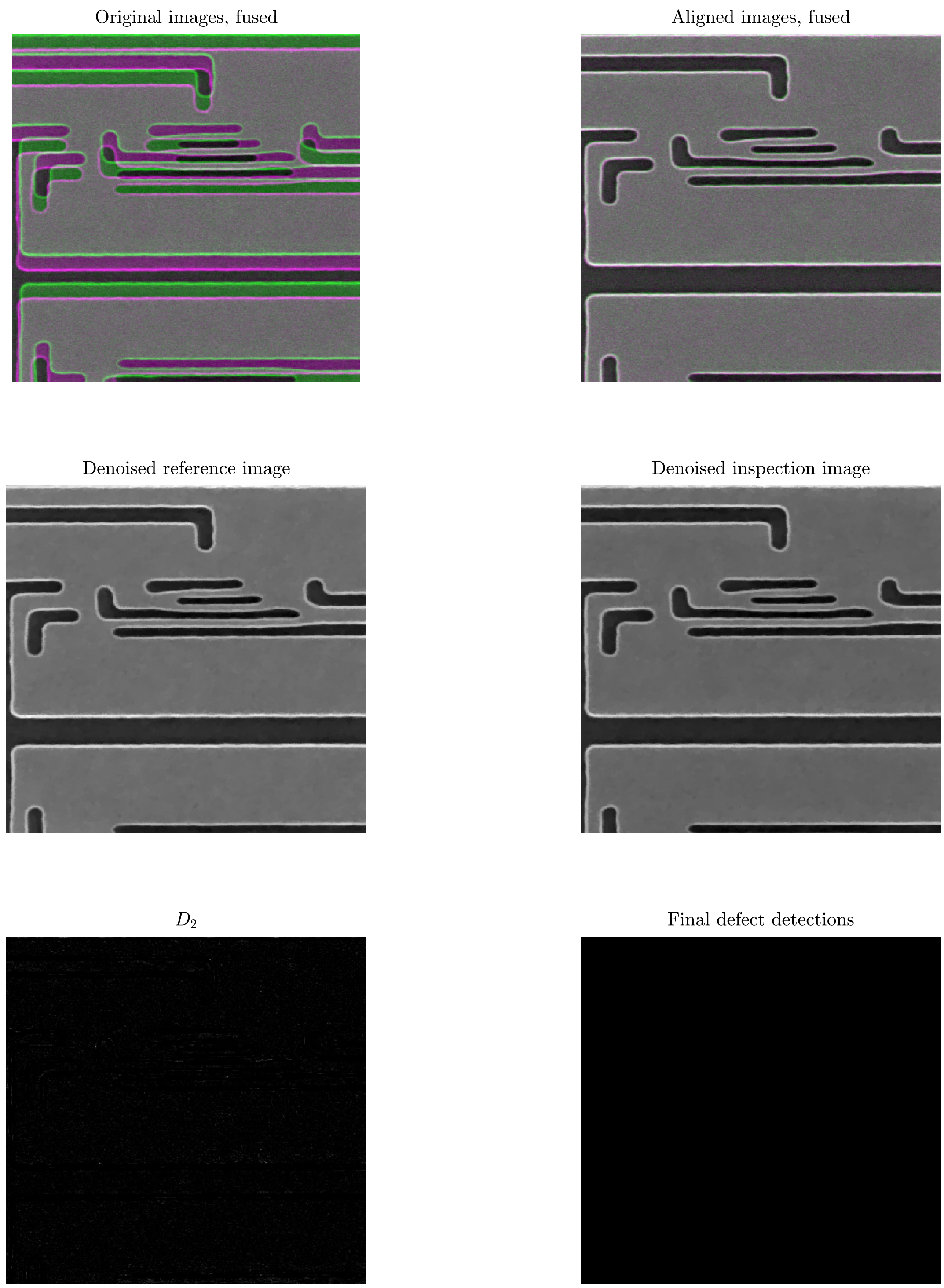

Figure 3: Final and intermediate results of case 3.