

GEO-5010 Research Assignment: Historical Maps Image Registration Using DFT-based Method

Ming-Chieh Hu

July 7, 2025

1 Introduction

The digitization of historical cartographic collections has created new opportunities for spatial analysis and historical research. Platforms like Allmaps¹ have demonstrated the potential of web-based georeferencing tools, yet a significant challenge remains in reconstructing multi-sheet historical maps where individual sheets must be accurately stitched together.

Traditional manual georeferencing approaches present limitations in both accuracy and efficiency. Users select corresponding control points between adjacent map sheets through visual inspection, a process that is either time-intensive or prone to error, particularly with hand-drawn maps exhibiting geometric distortions and varying quality.

This study investigates the applicability of DFT-based image registration techniques, specifically employing the Fourier-Mellin transform, for refining user-selected control points in historical map stitching workflows. The research addresses whether frequency-domain registration methods can effectively handle the unique characteristics of hand-drawn maps, including irregular line work, varying ink densities, and inherent geometric distortions.

This study focuses solely on the research question "Will DFT-based image registration help snapping the feature point selected by human eyes to a more precise location?". Therefore, although there might be other method that will work better in this given scenario e.g., SIFT (Lowe, 2004), RANSAC (Fischler and Bolles, 1981), etc, they're not in the scope of this research.

The structure of this paper is as follows. Section 2 introduces the evaluation and test datasets. Section 3 outlines the theoretical background and the proposed methodology. Section 4 presents the experimental results and analysis. Finally, Section 5 concludes the study and discusses potential directions for future work. All source code and software developed for this project are available in the linked GitHub repository².

2 Dataset

The map dataset selected for this study is assumed to meet the following criteria:

- It represents a large region subdivided into multiple smaller tiles;
- There exist human-identifiable connections (i.e., matching feature points) between adjacent tiles;
- The visual style of the map tiles is as consistent as possible;
- All tiles are of the same scale.

Given the variability and imperfections common in historical maps, a baseline evaluation was first conducted to assess the method's performance on standardized and transformed map patches with known ground truth. This evaluation is described in detail in Section 4.1. For this purpose, two sample datasets—an evaluation set and a test set—were constructed using four tiles from two different map sets.

2.1 Dataset 1: City Atlas Amsterdam

Also known as the Neighborhood Atlas by Loman (Buurtatlas van Loman), this highly detailed atlas was produced and published by Jan Cristiaan Loman Jr. Compiled from official sources at a scale of 1:1,250, it presents the ground plan of all buildings in Amsterdam, including municipal properties, public and private buildings, residences, and warehouses. The maps also feature cadastral annotations and reflect the house numbering system introduced in 1875.

¹<https://editor.allmaps.org>

²GitHub repository: DFT-map-stitching

Although the map set demonstrates a consistent drawing style across tiles, its overall precision is somewhat limited. Variations in line strength, patterns, angles, and even colors can be observed between tiles. Additionally, the atlas employs the Tanaka method—a technique for creating a 3D-like visual effect—by using shadows drawn at a specific angle.

In this study, two tiles from the map set—Buurt A and Buurt B—were selected, as shown in Fig.1. From each tile, ten patch pairs were created to form the evaluation dataset. For each pair, one patch was manually selected, and the other was generated from almost the same location but with a randomly applied rotation and translation. This resulted in a total of 20 evaluation pairs, the samples of them are shown in Fig.2. In addition, ten matching feature points were manually annotated across the two tiles to form the test dataset, as shown in Fig.3.

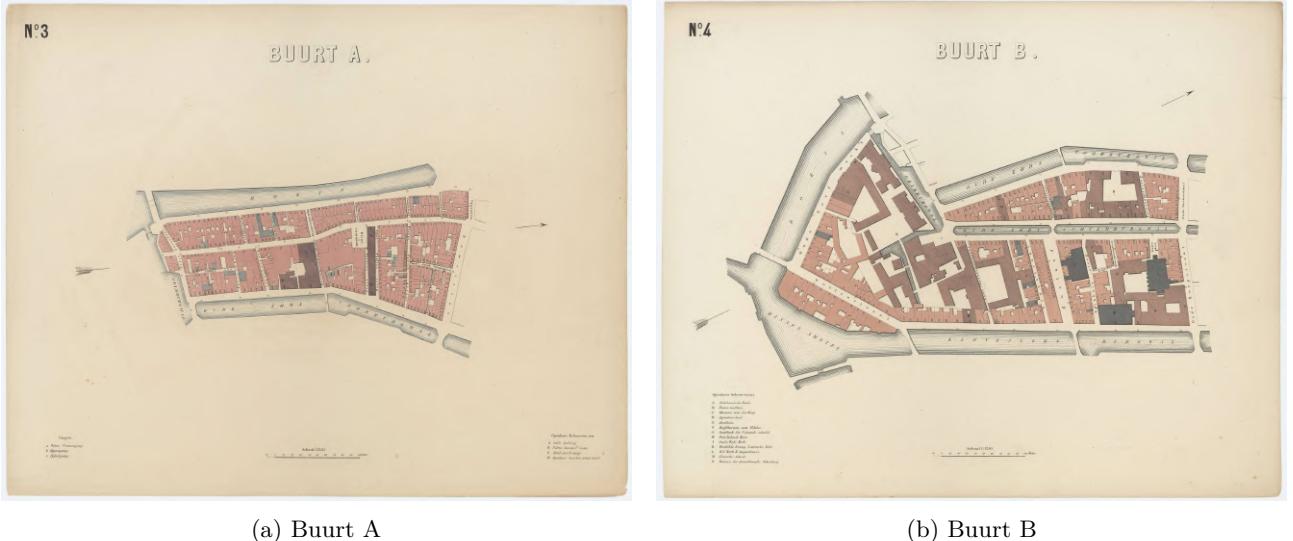


Figure 1: Sampled map tiles from *City Atlas Amsterdam* (Buurt A and B)

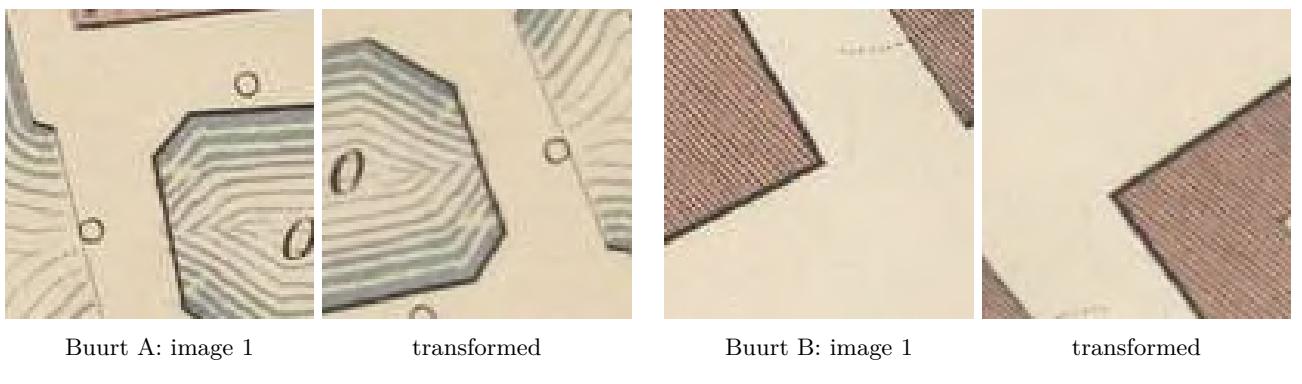


Figure 2: Sample image pairs from selected tiles for evaluation. Each shows a reference image and its transformed target

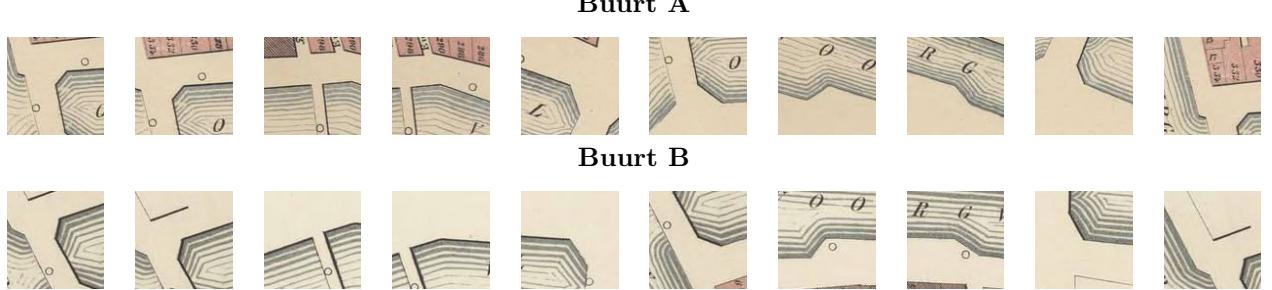


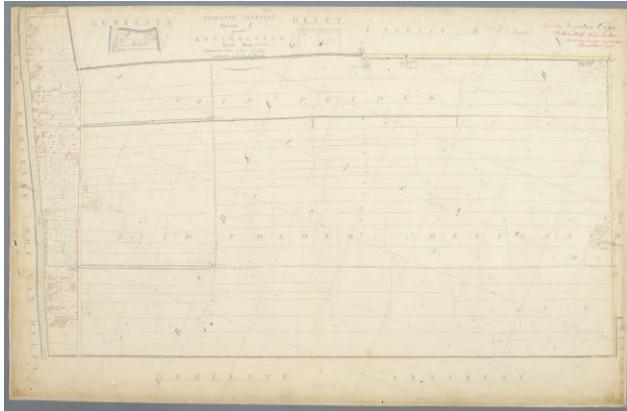
Figure 3: Ten selected patches from two connected tiles in *City Atlas Amsterdam*.

2.2 Dataset 2: Kadaster 1832

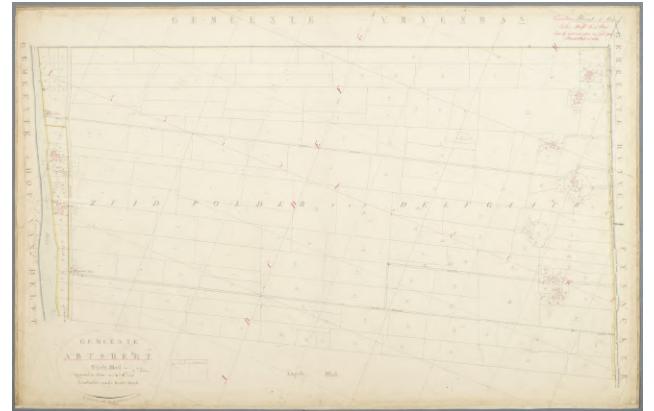
In 1810, the Netherlands was incorporated into France, leading to the introduction of a land tax. To support this system, land ownership needed to be measured, assessed, and officially recorded—marking the beginning of the Kadaster (Dutch Cadastre). The most challenging part of this process was surveying every parcel of land and body of water. The maps were reviewed for accuracy by verifiers, who drew control lines across the map sheets to intersect with parcel boundaries, enabling a systematic check of the measurements.

A notable advantage of this map set is the use of grid lines drawn by the surveyors. These grid lines provide useful spatial references and, in the two tiles selected for this study, they share a consistent orientation (though this is not always the case). While the graphical quality is lower than that of the *City Atlas Amsterdam*, the simpler visual elements may help alignment by reducing visual noise.

Following the same procedure as with the *City Atlas Amsterdam* map set, two tiles from the *Kadaster 1832*—MIN08220C01 and MIN08003A01—were selected, as shown in Fig.4. Examples of the evaluation pairs are illustrated in Fig.5, and the ten test pairs are shown in Fig. 6.



(a) Buurt A



(b) Buurt B

Figure 4: Sampled map tiles from *Kadaster 1832* (tile MIN08220C01 and MIN08003A01)



MIN08220C01: image 1



transformed



MIN08003A01: image 1



transformed

Figure 5: Sample image pairs from selected tiles for evaluation. Each shows a reference image and its transformed target

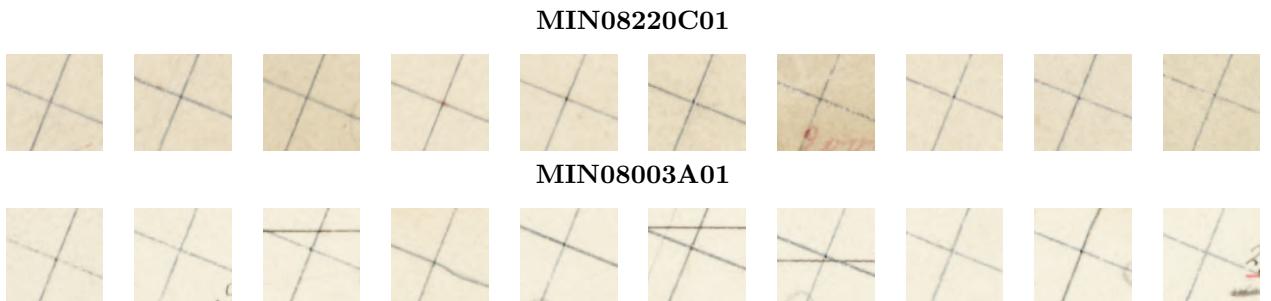


Figure 6: 10 selected patches from 2 connected tiles in *Kadaster 1832*

3 Methodology

The software development and experiments in this work are based on the Python package `imreg_dft`³ developed by Týč and Gohlke (2015). This lightweight implementation of image registration leverages the Fourier-Mellin transform to align two images that differ in scale, rotation, and translation. Additionally, several concepts and components presented here were adapted or modified from the work of Santosh Thoduka⁴, which partially ports the functionality of the `imreg_dft` package.

The method was first described by Reddy and Chatterji (1996). The algorithm estimates the similarity transform (translation, rotation and scale) between two images. The overall procedure is listed below:

1. **DFT to convert images into frequency domain** - compute magnitude spectra of both images
2. **Windowing/filtering of magnitude spectra** - apply window functions (e.g., Hann, Hamming) to reduce border artifacts and high-pass filtering to suppress low-frequency noise
3. **Log-polar transform of magnitude spectra** - convert the magnitude spectra (not original images) to log-polar coordinates where scale becomes vertical translation and rotation becomes horizontal translation
4. **Phase correlation in log-polar space** - estimate scale and rotation parameters as translation offsets
5. **Apply estimated scale/rotation corrections** - transform one image using found parameters
6. **Phase correlation in spatial domain** - estimate translation offset between the corrected images

This method works well when the two input images are nearly identical, differing only by slight translation and rotation. However, it is not suitable for our dataset or use case.

A recent study by Lentz et al. (2025) introduced several image pre-processing techniques to enhance phase correlation, particularly for featureless images. Our task is closely related to theirs—in their case, the goal was cloud image registration—while in ours, we deal with hand-drawn maps. Not only do our images involve translation and rotation, but they also exhibit illumination differences (e.g., background color variation across maps). The challenge is even greater in our scenario: line strokes are inconsistently drawn, and there are often varying levels of detail (LoD) across different tiles.

To address these issues, I implemented the pre-processing methods described in the paper and integrated them into the `imreg_dft` registration pipeline. The resulting workflow is illustrated in Fig. 7 and the implemented methods is detailed in Section 3.4.

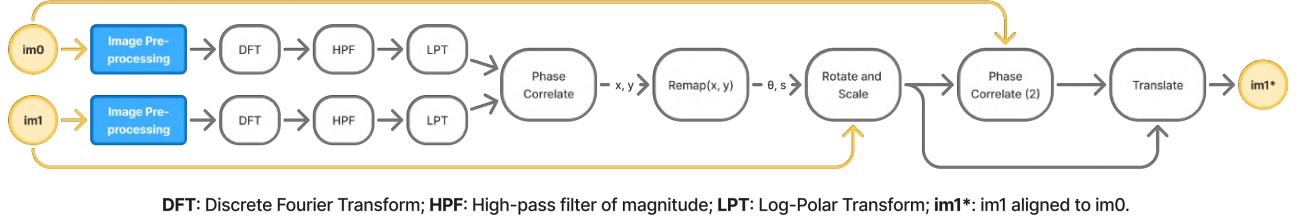


Figure 7: Overall procedure (modified from Santosh Thoduka’s work)

3.1 DFT: Discrete Fourier Transform

By default, the 0-frequency (DC component) in the 2D FFT output is at the top-left corner of the spectrum. When we apply a shift (e.g., using `fftshift` in NumPy or MATLAB), we move the zero-frequency component to the center of the spectrum. It’s often more natural to work with the 0-frequency centered for operations like log-polar transform, rotation estimation, or high-pass filtering.

Frequency effects After shifting the spectrum, higher frequency patterns (more cycles per unit distance) create frequency peaks farther from the center in the frequency domain, while lower frequencies appear closer to the center. The magnitude of these peaks remains similar since we’re using the same amplitude sinusoidal patterns, just with different cycle counts. The effect of changing pattern frequency is displayed in Fig. 8b.

Phase effects As shown in Fig. 8c, we see that changing the phase (horizontal shift) of identical frequency patterns produces virtually identical frequency domain representations. This demonstrates that phase shifts affect where the pattern starts spatially but don’t change the magnitude spectrum - the frequency content strength remains the same regardless of the 0°, 90°, or 180° phase shifts.

³GitHub: `imreg_dft`

⁴Documentation: Image registration based on the Fourier-Mellin transform

Magnitude effects Fig. 8d illustrates how amplitude changes in the spatial domain directly scale the frequency domain magnitudes. Higher amplitude patterns produce proportionally brighter frequency peaks, while lower amplitudes create dimmer peaks at the same frequency locations. This linear relationship between spatial amplitude and frequency magnitude is a key property of the DFT that makes it useful for signal analysis in geomatics applications.

Grid lines on maps Fig. 8a demonstrates how basic geometric features appear in the frequency domain. A vertical line in spatial domain creates a horizontal stripe in the frequency domain, while a horizontal line creates a vertical stripe - this 90-degree relationship is fundamental to DFT. The cross pattern combines both features, showing both horizontal and vertical frequency stripes simultaneously, but with lower individual magnitudes due to the different spatial signal distribution created by having more black pixels.

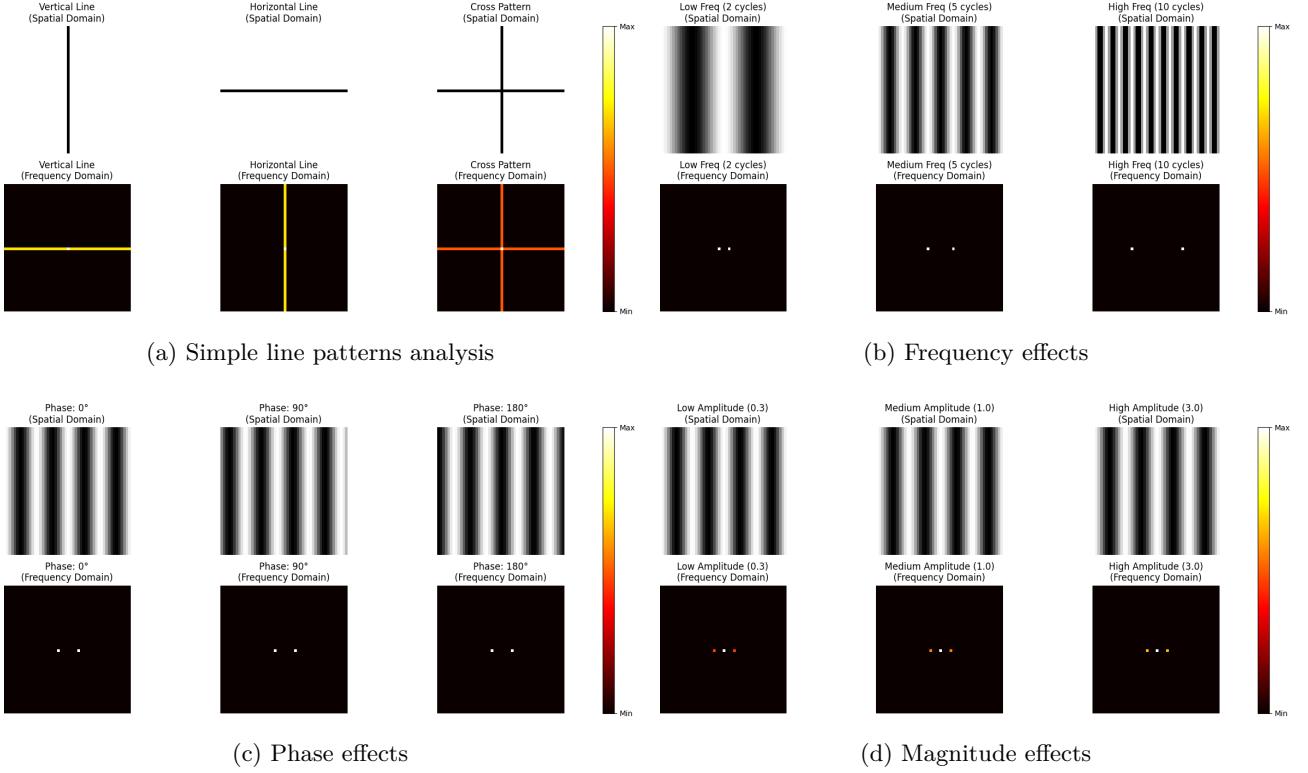
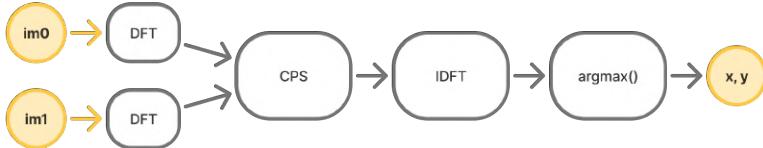


Figure 8: Demonstration of each parameter's effect

3.2 Phase Correlation



IDFT: Inverse Discrete Fourier Transform; **CPS:** Cross Power Spectrum.

Figure 9: Phase correlation procedure (reproduced from Santosh Thoduka's work)

The phase correlation method was introduced by Kuglin (1975). It is based on the Fourier shift theorem, which states that if two images are related by a translation:

$$f_2(x, y) = f_1(x - t_x, y - t_y) \quad (1)$$

then their Fourier transforms have identical magnitudes but differ only by a phase shift:

$$F_2(\xi, \eta) = e^{-j2\pi(\xi t_x + \eta t_y)} F_1(\xi, \eta) \quad (2)$$

where ξ and η are the spatial frequencies in the x and y directions (cycles per unit length) respectively. This phase difference can be isolated using the normalized cross power spectrum (CPS):

$$e^{-j2\pi(\xi t_x + \eta t_y)} = \frac{F_1(\xi, \eta)F_2^*(\xi, \eta)}{|F_1(\xi, \eta)F_2(\xi, \eta)|} \quad (3)$$

where F_2^* is the complex conjugate⁵ of F_2 . The cross power spectrum $F_1(\xi, \eta)F_2^*(\xi, \eta)$ isolates the phase difference between the two signals, while normalizing by the magnitude $|F_1(\xi, \eta)F_2(\xi, \eta)|$ removes amplitude information, preserving only phase relationships. Since $e^{-j2\pi(\xi t_x + \eta t_y)}$ is the Fourier transform of a shifted delta function, the Inverse-DFT (IDFT) produces an impulse at the translation location:

$$\mathcal{F}^{-1}\{e^{-j2\pi(\xi t_x + \eta t_y)}\} = \delta(x - t_x, y - t_y) \quad (4)$$

3.3 Log-polar Transform

Phase correlation is an efficient method for determining the translational offset between pairs of similar images. However, this approach relies on the near absence of rotation and scaling differences between the images, which are typical in real-world scenarios. To recover rotation and scaling differences between two images, we can leverage two geometric properties of the log-polar transform and the translation invariance of the frequency domain.⁶

First, rotation in Cartesian space becomes a translation along the angular coordinate (θ) axis of log-polar space. Second, scaling in Cartesian space becomes a translation along the radial coordinate ($\rho = \ln \sqrt{x^2 + y^2}$) of log-polar space. Finally, translational differences in the spatial domain do not affect the magnitude spectrum in the frequency domain. An illustration of the mapping between Cartesian and log-polar coordinates, taken from Sarvaiya et al. (2009), is shown in Fig. 10.

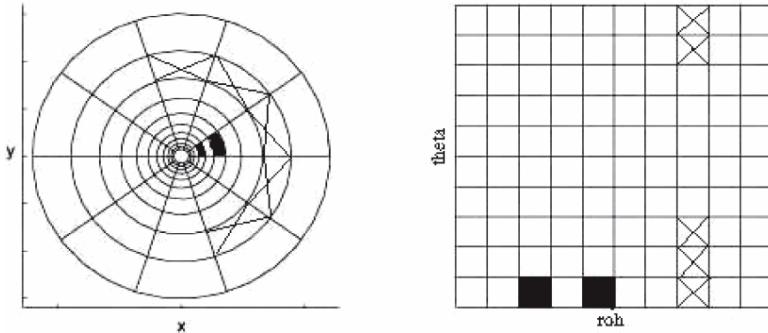


Figure 10: Approximate mapping from Cartesian space to (ρ, θ) space

3.4 Image Pre-processing Methods

As previously mentioned, the image pre-processing techniques employed in this work are based on the methodology proposed by Lentz et al. (2025). This section briefly describes each of the three key steps:

1. Pixel value conjugation
2. Background subtraction
3. Contrast stretching (equivalent to `imadjust()` in Matlab)

These steps are applied sequentially to each image pair as a unified pipeline, rather than being evaluated independently. This combined procedure will be referred to as the *baseline* throughout the remainder of the paper. The overall effect of these pre-processing steps is illustrated in Fig. 11.

The referenced paper also used a Contrast-Limited Adaptive Histogram Equalization (CLAHE, `adapthisteq()` in Matlab) method to further equalize the illumination differences in 2 images. Here I didn't implement this method because the historical maps doesn't really contain much grey area to be re-distributed into bins.

Besides the image pre-processing methods, the authors also went through three more sub-processes before going into phase correlation step, i.e, de-noising blur, image windowing and image padding. All these three processes didn't work well on my case and the outcome of these will be address in Section 4.3.

⁵If $F_2 = (a + bi)$, then $F_2^* = (a - bi)$

⁶This explanation is inspired by the scikit-image example: *Registering images with rotation and scaling*

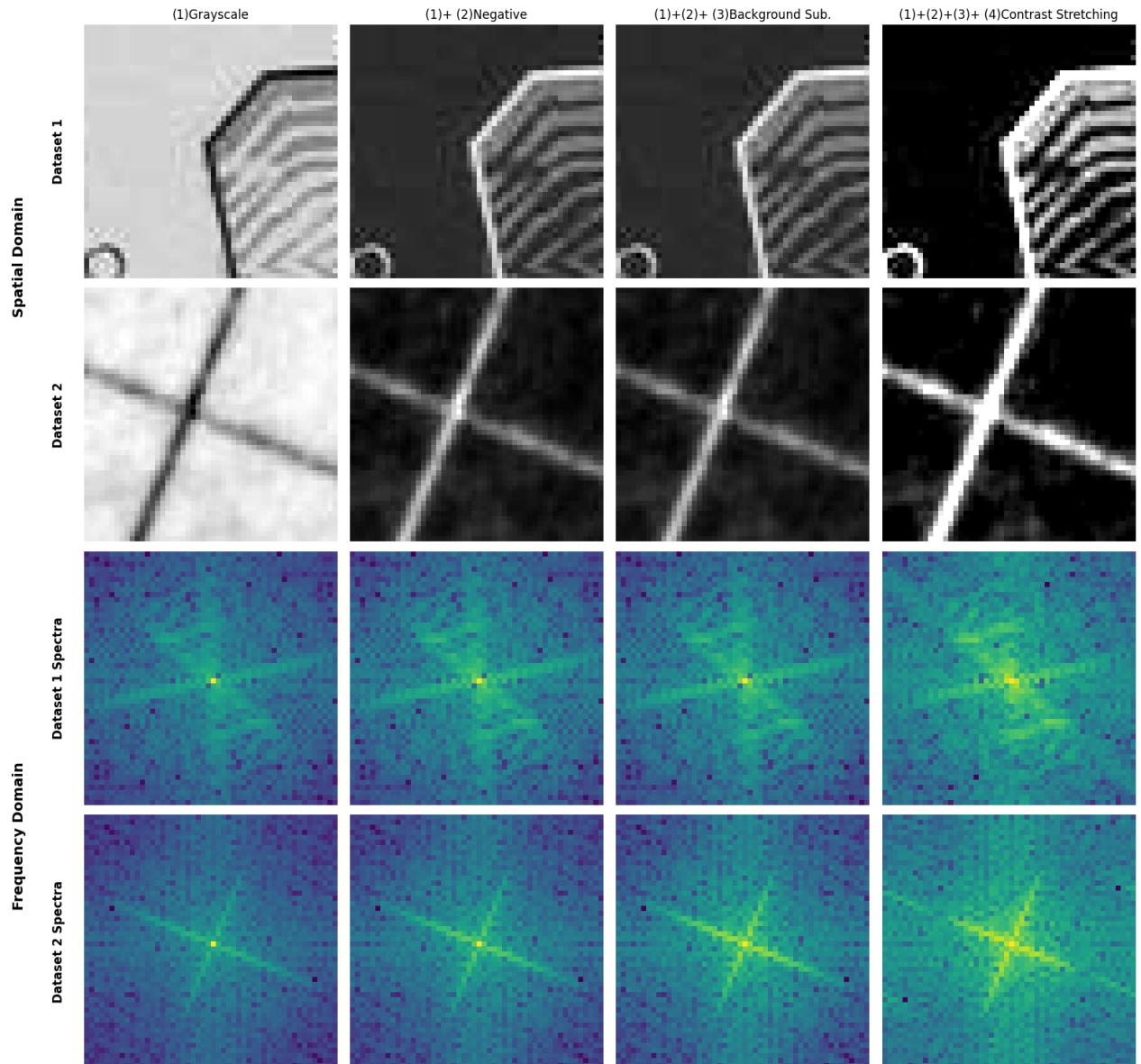


Figure 11: Pre-processing methods applied in the baseline method

3.4.1 Pixel Value Conjugation

I propose a pixel value conjugation step—inviting the image’s black and white values (similar to a negative film effect). This transformation makes subsequent operations more intuitive and allows the use of other standard image processing techniques without modification. Mathematically, this is achieved by applying the transformation $p_n^* = |255 - p_n|$ to each pixel value p_n .

3.4.2 Background Subtraction

The term background illumination (d_0) refers to a uniform pixel offset present throughout an image that exceeds the minimum level of modulation. In the referenced work, a global minimum pixel value is used as d_0 . In contrast, I use two local minima—one from each of the paired map tiles—to normalize the minimum pixel values to exactly zero. This approach is necessary because the background pixel values differ significantly between the two tiles. The difference between proposed method and referenced method is shown in Fig. 12.

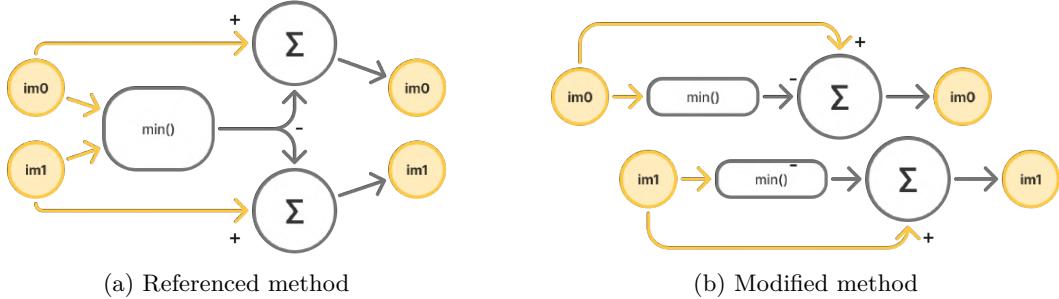


Figure 12: Background subtraction method difference

3.4.3 Contrast stretching

The referenced paper applied a basic image stretching algorithm that maps the lowest 1% of pixel intensities to 0 and the highest 1% to 1. The remaining pixel values are linearly scaled to the [0, 255] range using a linear transformation, resulting in an image with $d_0 = 0$. I re-implemented this approach in Python, making the low and high percentiles configurable parameters for easier adjustment.

4 Experiments

Due to the nature of hand-drawn map datasets, quantitative evaluation of translation or rotation errors is not feasible without ground truth data. Additionally, testing all parameter combinations would be impractical within the scope of this study. Therefore, a strategic and structured evaluation approach is adopted, it consists of 3 steps:

1. Baseline evaluation with controlled dataset and specifically generated ground truth data from the same map tiles, to make sure that this method will work in theory;
2. Experiments on the use case across different sizes of patches, to find out the best size parameter and the effect of size;
3. Experiments on different pre-processing techniques (based on the best-performing patch size from the previous step)

4.1 Baseline Evaluation

The proposed baseline method is first evaluated through a controlled dataset I made from the same 2 pairs of 4 map tiles, but the patch pairs are cropped from the same map tile and applied a random rotation and translation to be sure of that the translation and rotation is controlled and the error can be quantitatively measured.

A total of 40 image patch pairs were used in the evaluation. The results are presented in Tables 1, 2, 3, and 4. The average translation error is 0.827 pixels, and the average rotation error is 0.317 degrees. Note that all translation errors are measured in pixels, and rotation errors in degrees. The maximum translation error observed is 1.469 pixels, and the maximum rotation error is 0.933 degrees. Since all rotation errors remain below 1 degree and all translation errors below 2 pixels, these results indicate that the baseline method performs reliably and with consistent accuracy under controlled conditions.

Table 1: Comparison of predicted and ground truth transformation parameters for Buurt A.

Pair	Prediction		Ground Truth		Error	
	(t_x, t_y)	θ	(t_x, t_y)	θ	$\ (t_x, t_y)\ $	θ
1	(4.796, 21.387)	172.557	(5.071, 22.085)	172.067	0.750	0.491
2	(-20.303, -0.418)	-95.695	(-21.168, -1.605)	-96.194	1.469	0.499
3	(-0.731, 22.875)	149.883	(-1.393, 23.087)	149.927	0.695	0.044
4	(2.274, 14.262)	-45.474	(2.402, 12.954)	-45.584	1.314	0.110
5	(-22.364, 21.384)	84.626	(-22.431, 22.808)	85.001	1.425	0.376
6	(9.666, -8.255)	126.066	(9.665, -7.048)	125.957	1.206	0.109
7	(14.803, 18.617)	-133.671	(15.545, 18.044)	-133.935	0.937	0.263
8	(-13.156, -1.914)	39.296	(-13.178, -0.941)	39.236	0.974	0.060
9	(0.188, -4.741)	125.501	(-0.969, -5.258)	125.296	1.268	0.205
10	(21.959, -23.056)	48.515	(21.168, -21.908)	48.965	1.394	0.449

Table 2: Comparison of predicted and ground truth transformation parameters for Buurt B

Pair	Prediction		Ground Truth		Error	
	(t_x, t_y)	θ	(t_x, t_y)	θ	$\ (t_x, t_y)\ $	θ
1	(4.796, 21.387)	172.557	(5.071, 22.085)	172.067	0.750	0.491
2	(-20.303, -0.418)	-95.695	(-21.168, -1.605)	-96.194	1.469	0.499
3	(-0.731, 22.875)	149.883	(-1.393, 23.087)	149.927	0.695	0.044
4	(2.274, 14.262)	-45.474	(2.402, 12.954)	-45.584	1.314	0.110
5	(-22.364, 21.384)	84.626	(-22.431, 22.808)	85.001	1.425	0.376
6	(9.666, -8.255)	126.066	(9.665, -7.048)	125.957	1.206	0.109
7	(14.803, 18.617)	-133.671	(15.545, 18.044)	-133.935	0.937	0.263
8	(-13.156, -1.914)	39.296	(-13.178, -0.941)	39.236	0.974	0.060
9	(0.188, -4.741)	125.501	(-0.969, -5.258)	125.296	1.268	0.205
10	(21.959, -23.056)	48.515	(21.168, -21.908)	48.965	1.394	0.449

Table 3: Comparison of predicted and ground truth transformation parameters for tile MIN08220C01

Pair	Prediction		Ground Truth		Error	
	(t_x, t_y)	θ	(t_x, t_y)	θ	$\ (t_x, t_y)\ $	θ
1	(4.796, 21.387)	172.557	(5.071, 22.085)	172.067	0.750	0.491
2	(-20.303, -0.418)	-95.695	(-21.168, -1.605)	-96.194	1.469	0.499
3	(-0.731, 22.875)	149.883	(-1.393, 23.087)	149.927	0.695	0.044
4	(2.274, 14.262)	-45.474	(2.402, 12.954)	-45.584	1.314	0.110
5	(-22.364, 21.384)	84.626	(-22.431, 22.808)	85.001	1.425	0.376
6	(9.666, -8.255)	126.066	(9.665, -7.048)	125.957	1.206	0.109
7	(14.803, 18.617)	-133.671	(15.545, 18.044)	-133.935	0.937	0.263
8	(-13.156, -1.914)	39.296	(-13.178, -0.941)	39.236	0.974	0.060
9	(0.188, -4.741)	125.501	(-0.969, -5.258)	125.296	1.268	0.205
10	(21.959, -23.056)	48.515	(21.168, -21.908)	48.965	1.394	0.449

Table 4: Comparison of predicted and ground truth transformation parameters for tile MIN08003A01

Pair	Prediction		Ground Truth		Error	
	(t_x, t_y)	θ	(t_x, t_y)	θ	$\ (t_x, t_y)\ $	θ
1	(4.796, 21.387)	172.557	(5.071, 22.085)	172.067	0.750	0.491
2	(-20.303, -0.418)	-95.695	(-21.168, -1.605)	-96.194	1.469	0.499
3	(-0.731, 22.875)	149.883	(-1.393, 23.087)	149.927	0.695	0.044
4	(2.274, 14.262)	-45.474	(2.402, 12.954)	-45.584	1.314	0.110
5	(-22.364, 21.384)	84.626	(-22.431, 22.808)	85.001	1.425	0.376
6	(9.666, -8.255)	126.066	(9.665, -7.048)	125.957	1.206	0.109
7	(14.803, 18.617)	-133.671	(15.545, 18.044)	-133.935	0.937	0.263
8	(-13.156, -1.914)	39.296	(-13.178, -0.941)	39.236	0.974	0.060
9	(0.188, -4.741)	125.501	(-0.969, -5.258)	125.296	1.268	0.205
10	(21.959, -23.056)	48.515	(21.168, -21.908)	48.965	1.394	0.449

4.2 Experiments: Across Different Patch Sizes

To determine the optimal patch size for best performance, three configurations were evaluated: 30×30 , 50×50 , and 100×100 pixels. I didn't choose to go any bigger sizes since the map tiles are usually only connected to the edge and taking numbers like 200×200 pixels can sometimes reach the boundary or simply blank area. Based on the results, the most effective size will be selected for further refinement and parameter tuning. Figures 13, 14, and 15 present the results for Dataset 1 across the three patch sizes, while Figures 16, 17, and 18 show the corresponding results for Dataset 2.

For both datasets, the performance trend follows the pattern: $30\text{px} < 50\text{px} \approx 100\text{px}$, indicating that larger patch sizes generally yield improved registration accuracy and stability. However, despite their relative performance, the results still exhibit noticeable defects and are likely not precise enough for the intended use case. Note that the grid size in each figure corresponds to 5×5 pixels.

The 30×30 pixel patches exhibited the poorest performance, with registration failures and high instability across multiple image pairs. The 50×50 pixel configuration showed substantially improved performance. The 100×100 pixel patches performed unexpectedly well, though some registration errors remained present, often comparable to the 50×50 pixel results. While 30×30 pixel patches performed better on Dataset 2 compared to Dataset 1, noticeable rotation errors persisted. Similar to Dataset 1, both 50×50 and 100×100 pixel configurations demonstrated satisfactory performance with comparable accuracy levels.

This performance pattern can be explained by the fundamental frequency resolution limitations of the DFT. The frequency resolution of the Discrete Fourier Transform is inversely proportional to image size, following $\Delta f = 1/N$ where N is the number of pixels. This results in frequency resolutions of 0.033, 0.02, and 0.01 cycles/pixel for 30×30 , 50×50 , and 100×100 patches respectively. The coarse frequency sampling in 30×30 patches reduces the algorithm's ability to distinguish spatial frequencies in the grid structure, resulting in broader, less distinct correlation peaks and degraded registration performance.

4.3 Experiments on Different Pre-processes

4.3.1 De-noising Blur

Applying Gaussian blur with $\sigma = 1.0$ using `scipy.ndimage.gaussian_filter` to 100×100 pixel patches resulted in visible degraded performance. While the blur reduces noise and drawing artifacts in the hand-drawn maps, it simultaneously attenuates the distinctive features that are essential for accurate phase correlation. This trade-off between noise reduction and feature preservation suggests that for hand-drawn historical maps, the structural information lost through blurring outweighs the benefits of noise suppression. The lines, which are the primary features, require sharp edges to generate distinct correlation peaks in the frequency domain. The results using $\sigma = 1.0$ are shown in Figures 19 and 20.

4.3.2 Image Windowing

A Tukey window with $\alpha = 0.5$ was applied to gradually taper edge pixels to zero, focusing the algorithm on the central region of each patch. This windowing technique is commonly used in phase correlation to mitigate boundary discontinuity that arise from the DFT's periodic assumption. While such filter can be beneficial for large-scale image stitching applications, it proved counterproductive for the current registration task.

Testing with alternative parameters ($\alpha = 0.2$ and $\alpha = 0.8$) confirmed that increasing the windowing strength progressively degraded performance. This degradation occurs because the windowing function attenuates grid intersections near patch boundaries, which often contain critical spatial information for accurate registration. For small patches (100×100 pixels), the loss of boundary features outweighs the reduction in spectral artifacts, making windowing unsuitable for this application. The results using $\alpha = 0.5$ windowing are displayed in Figures 21 and 22.

4.3.3 Image Padding

Following the methodology in Lentz et al. (2025), images were padded with zeros to three times their original size. Additional padding ratios of $2\times$ and $1.5\times$ were also evaluated, but neither configuration improved performance.

While increased padding theoretically allows the algorithm to detect larger translation vectors by expanding the search space, this proves counterproductive for the current application. The use case focuses on refining user-selected control points in historical map tiles, where the expected translations are relatively small. Excessive padding creates additional opportunities for false correlation peaks at large displacements. Since the initial control point selection provides approximate alignment already, the registration algorithm should prioritize precision within a limited search radius rather than robustness to large translations. The results with 0-padding to $3\times$ their original size are displayed in Figures 23 and 24 (note that these images are cropped again to the original size).

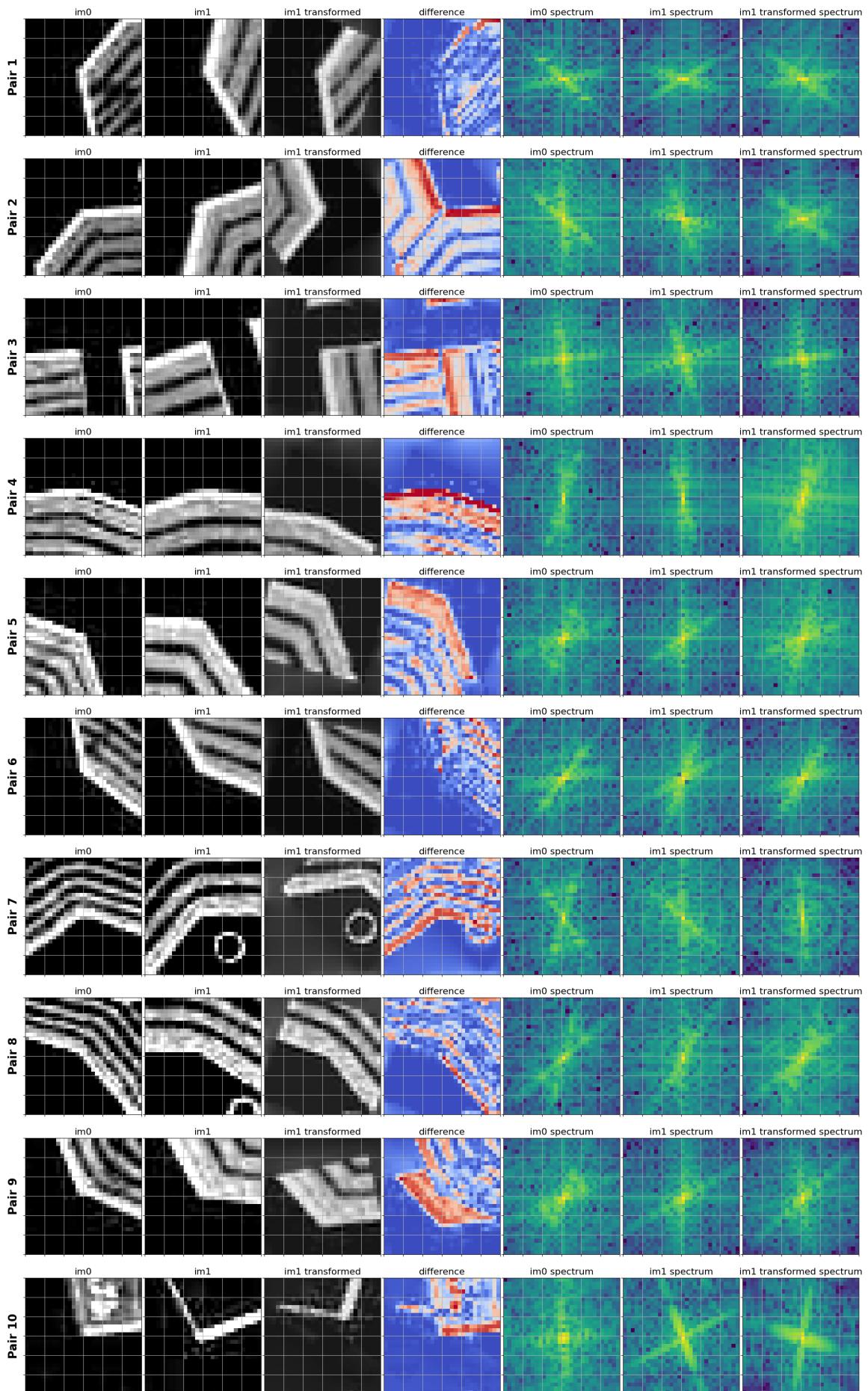


Figure 13: Test result: dataset 1, 30×30 pixels patch

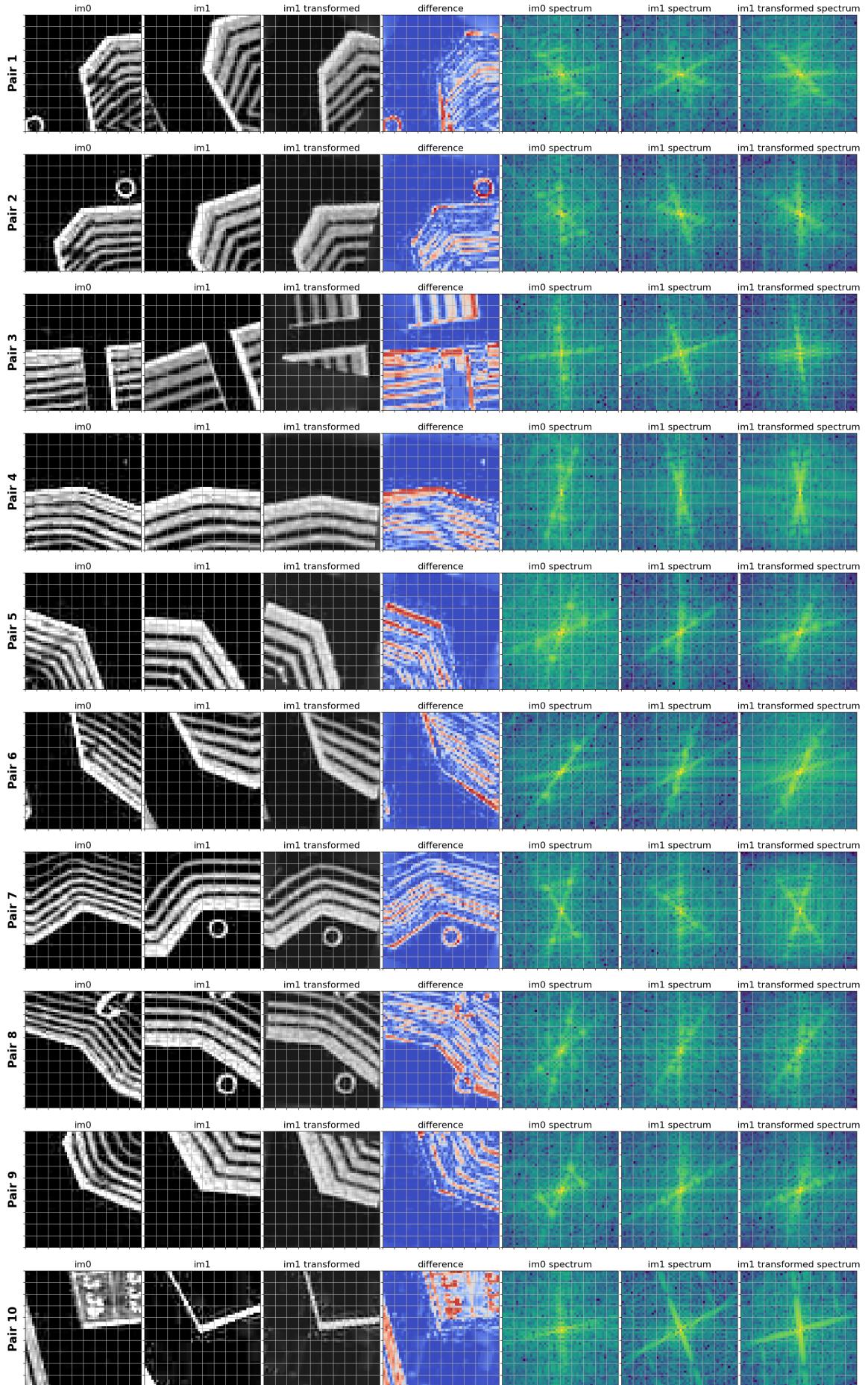


Figure 14: Test result: dataset 1, 50×50 pixels patch

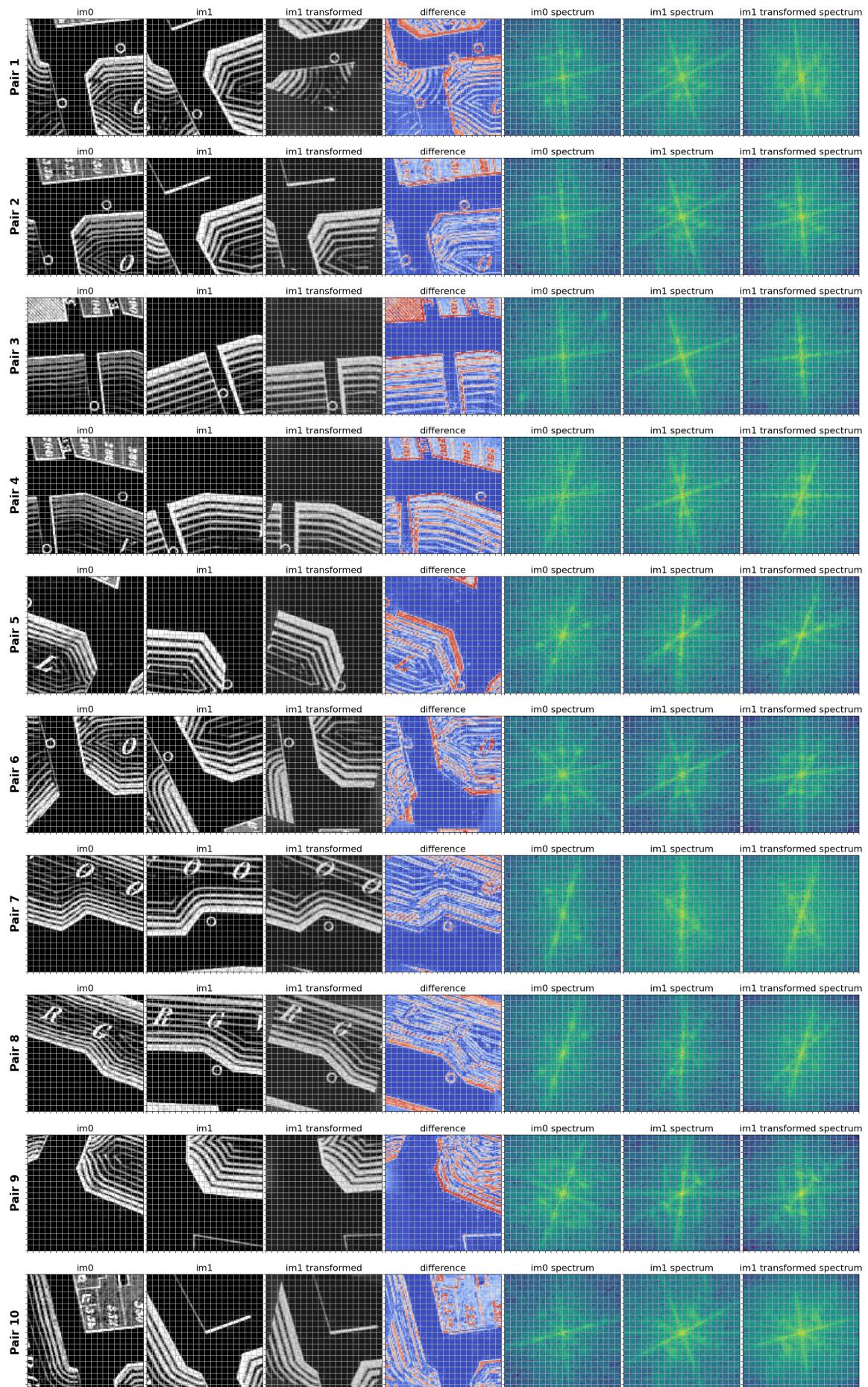


Figure 15: Test result: dataset 1, 100×100 pixels patch

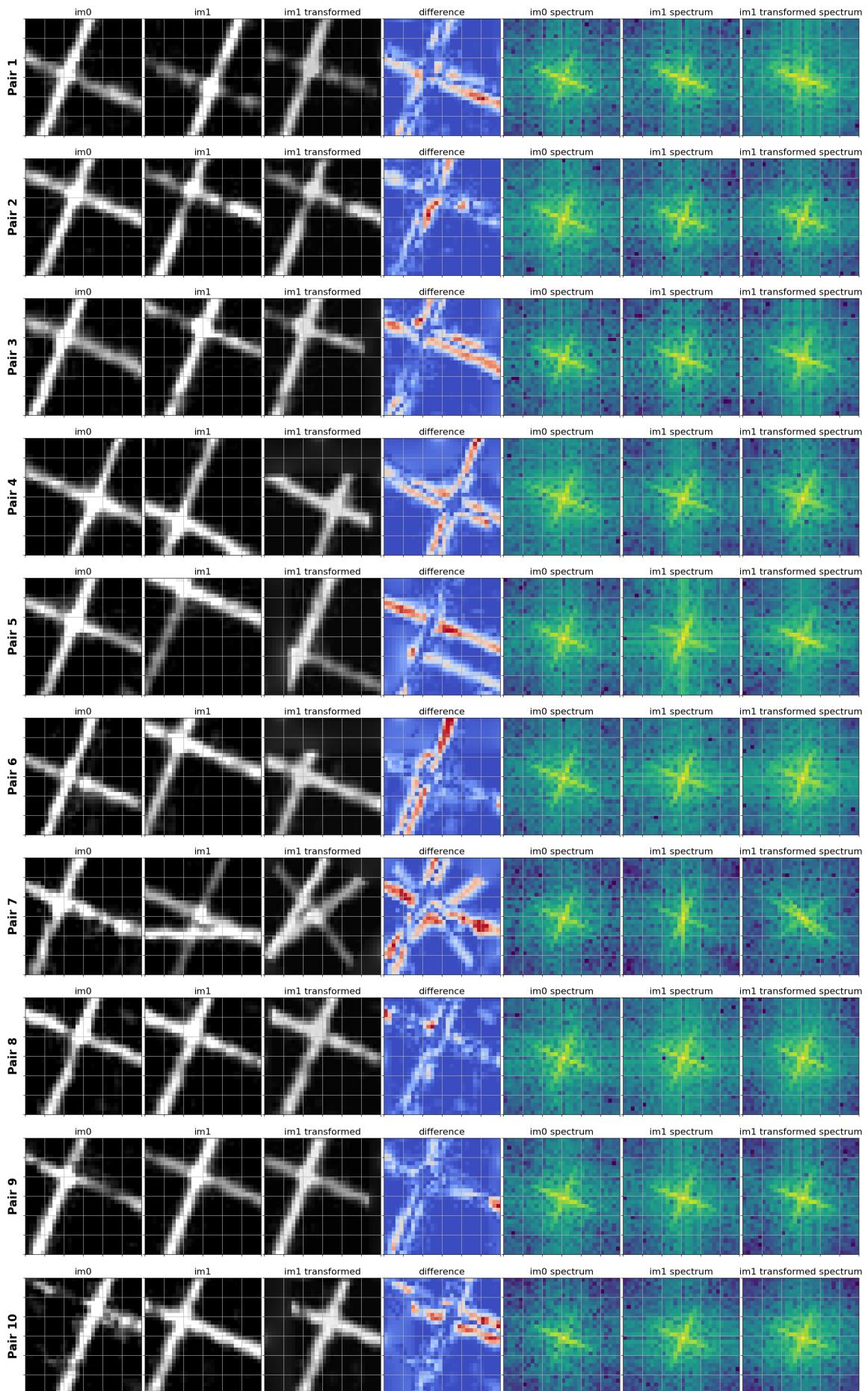


Figure 16: Test result: dataset 2, 30×30 pixels patch

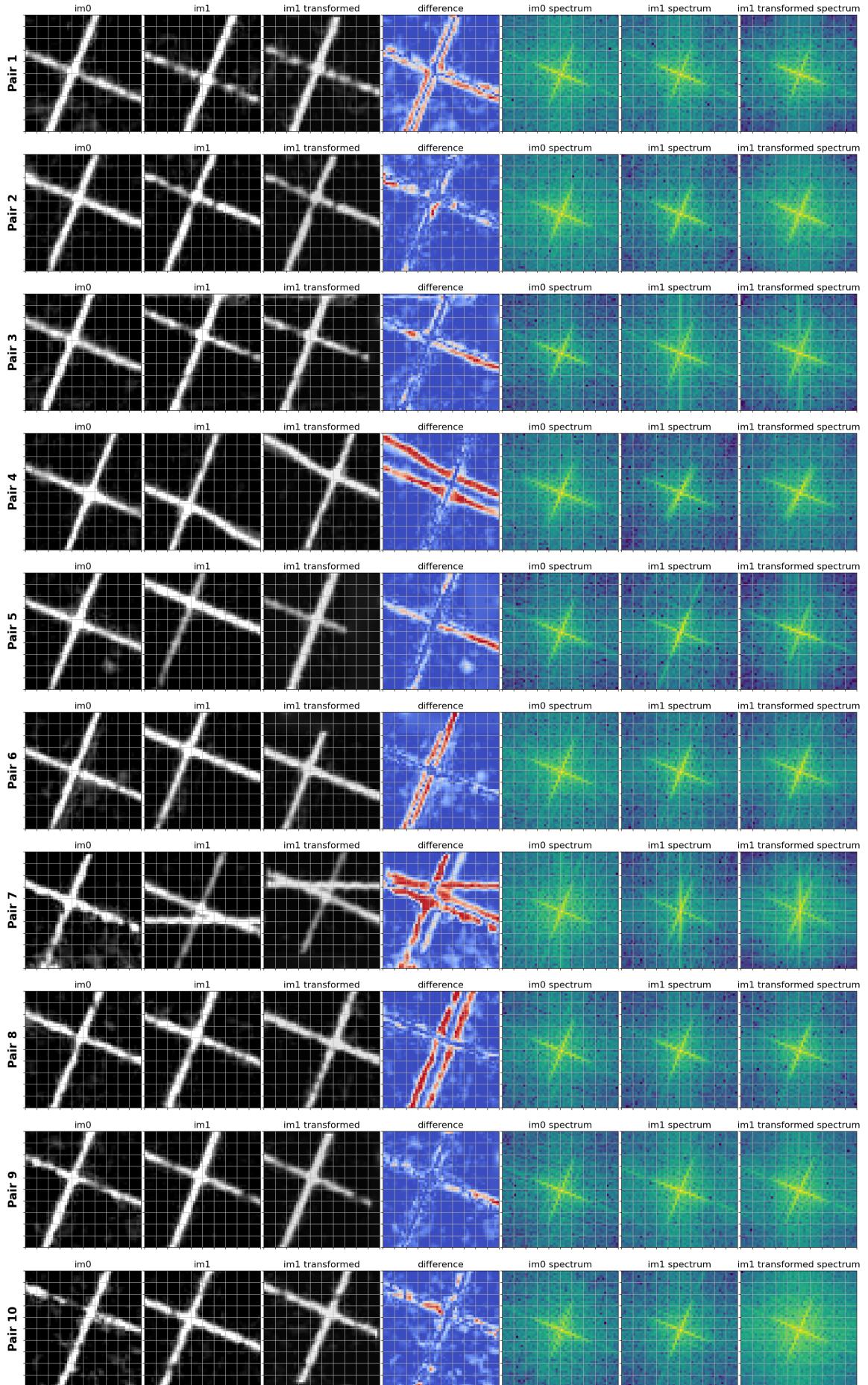


Figure 17: Test result: dataset 2, 50×50 pixels patch

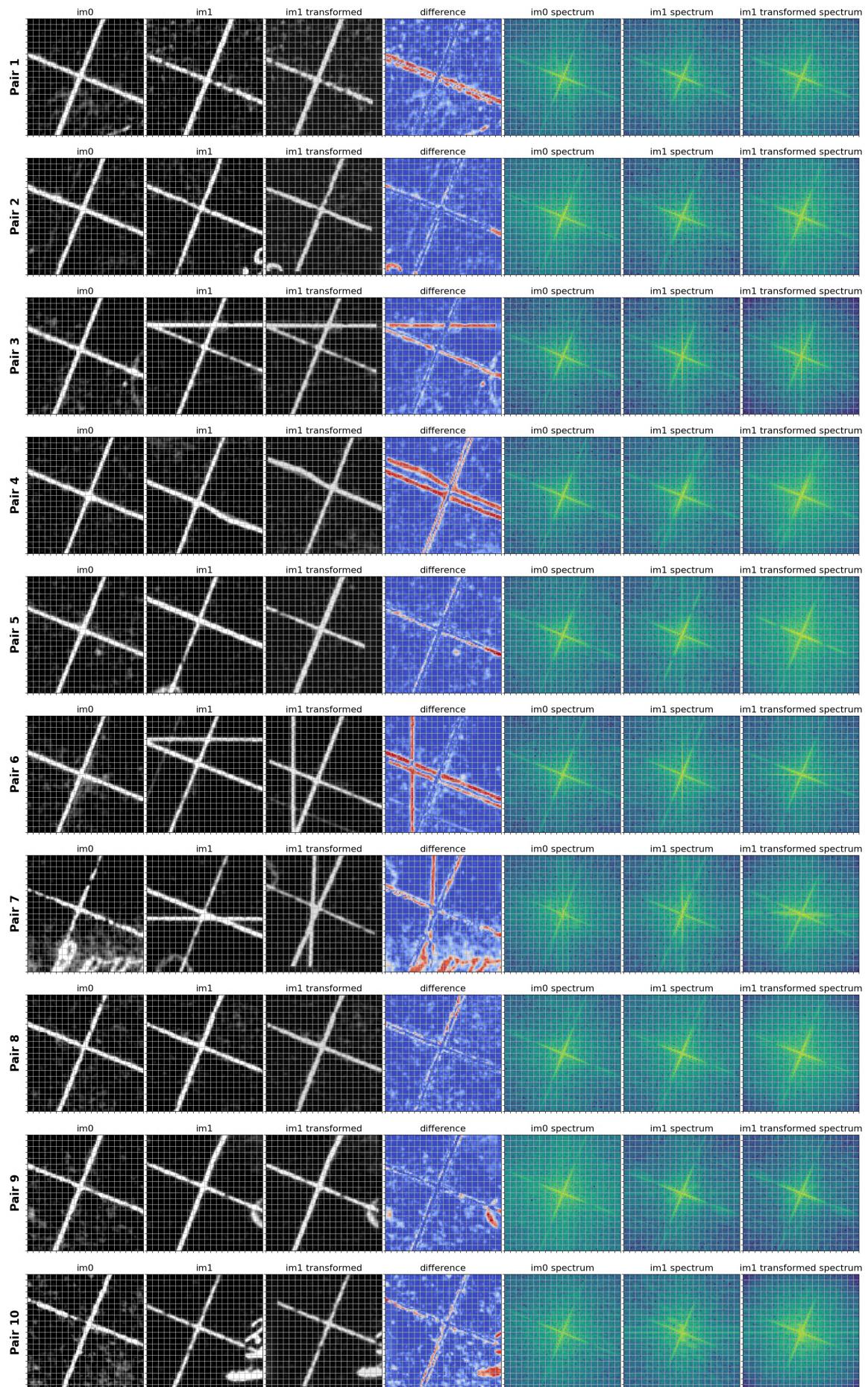


Figure 18: Test result: dataset 2, 100×100 pixels patch

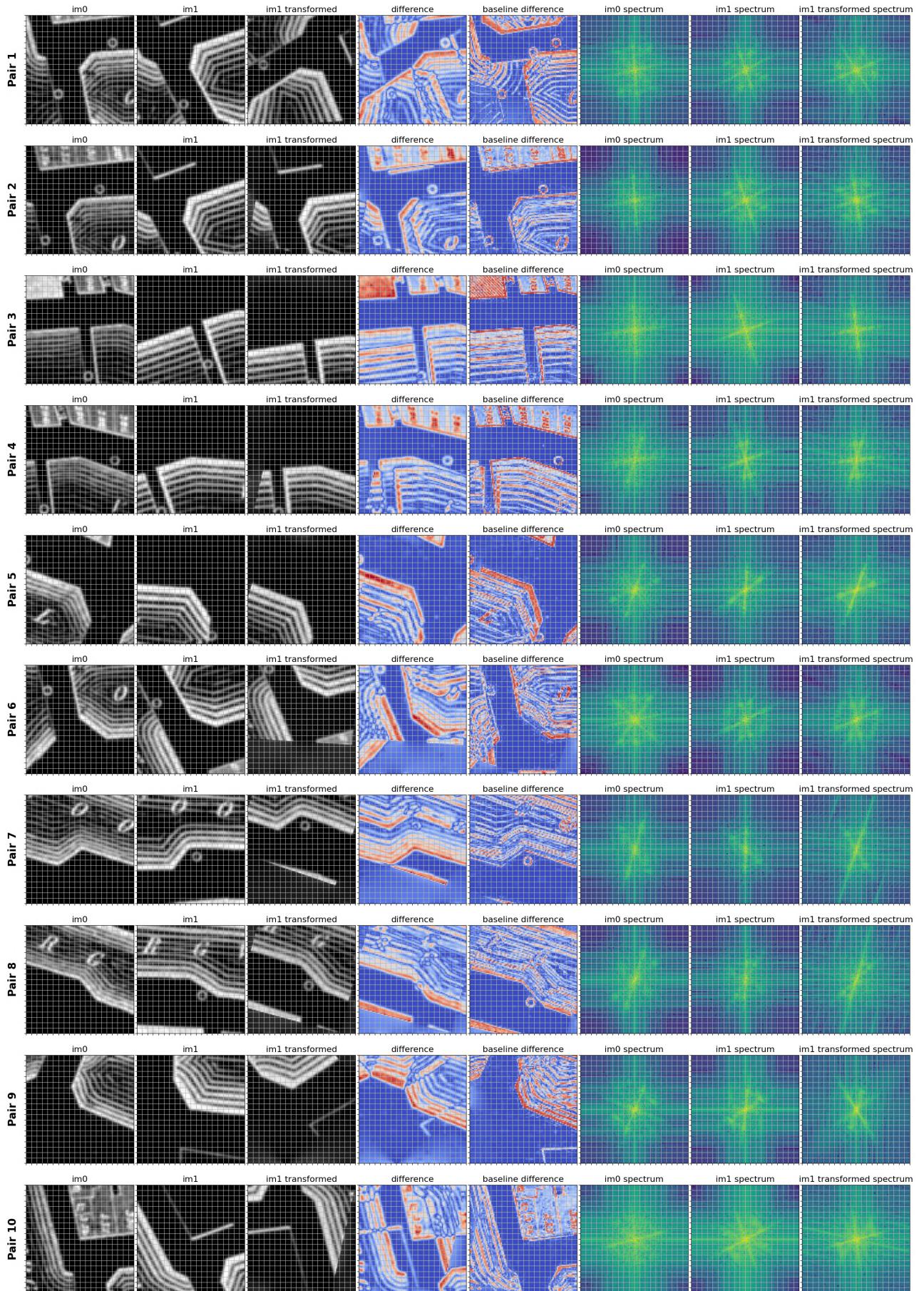


Figure 19: Test result: dataset 1 with gaussian blur effect applied

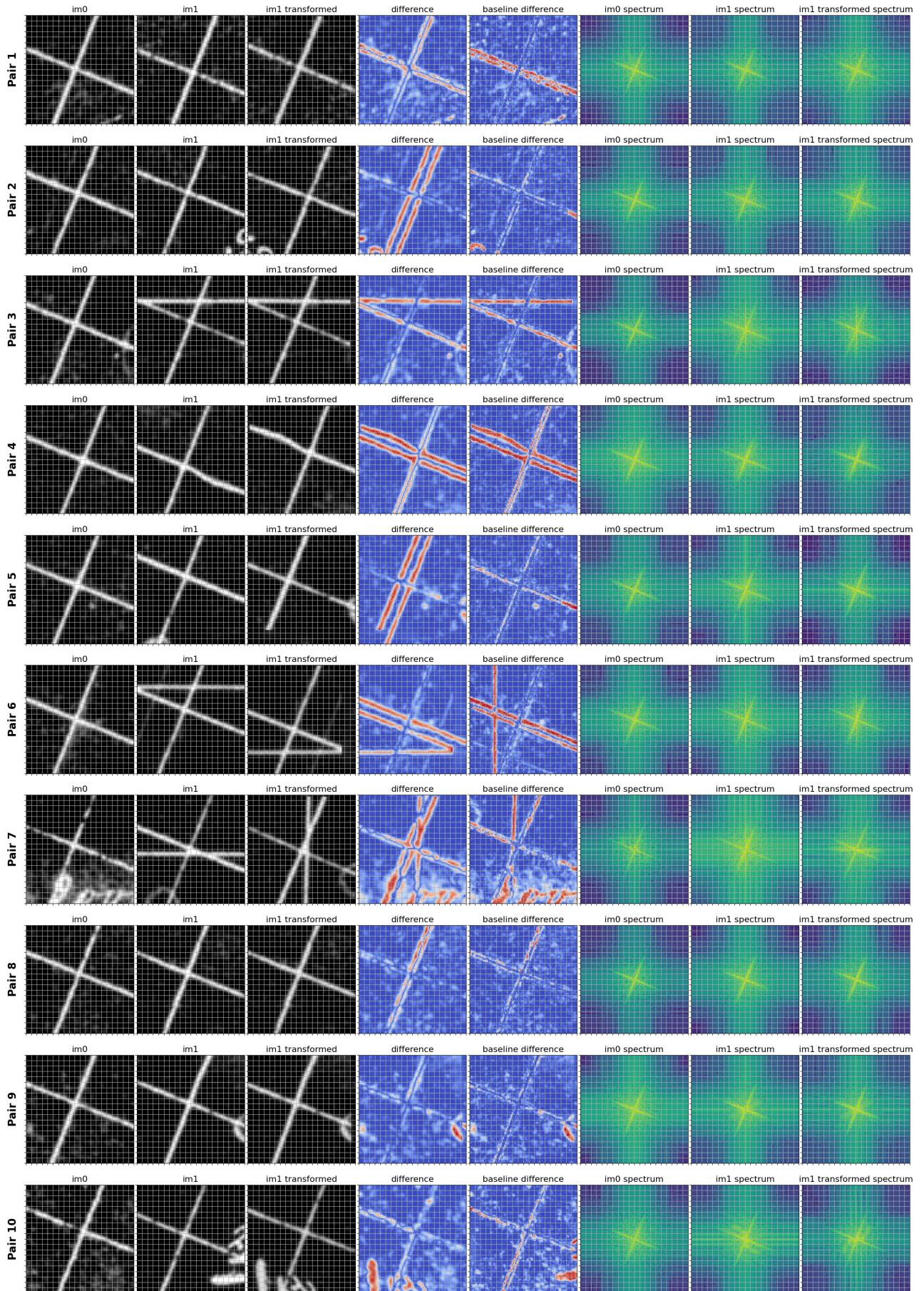


Figure 20: Test result: dataset 2 with gaussian blur effect applied

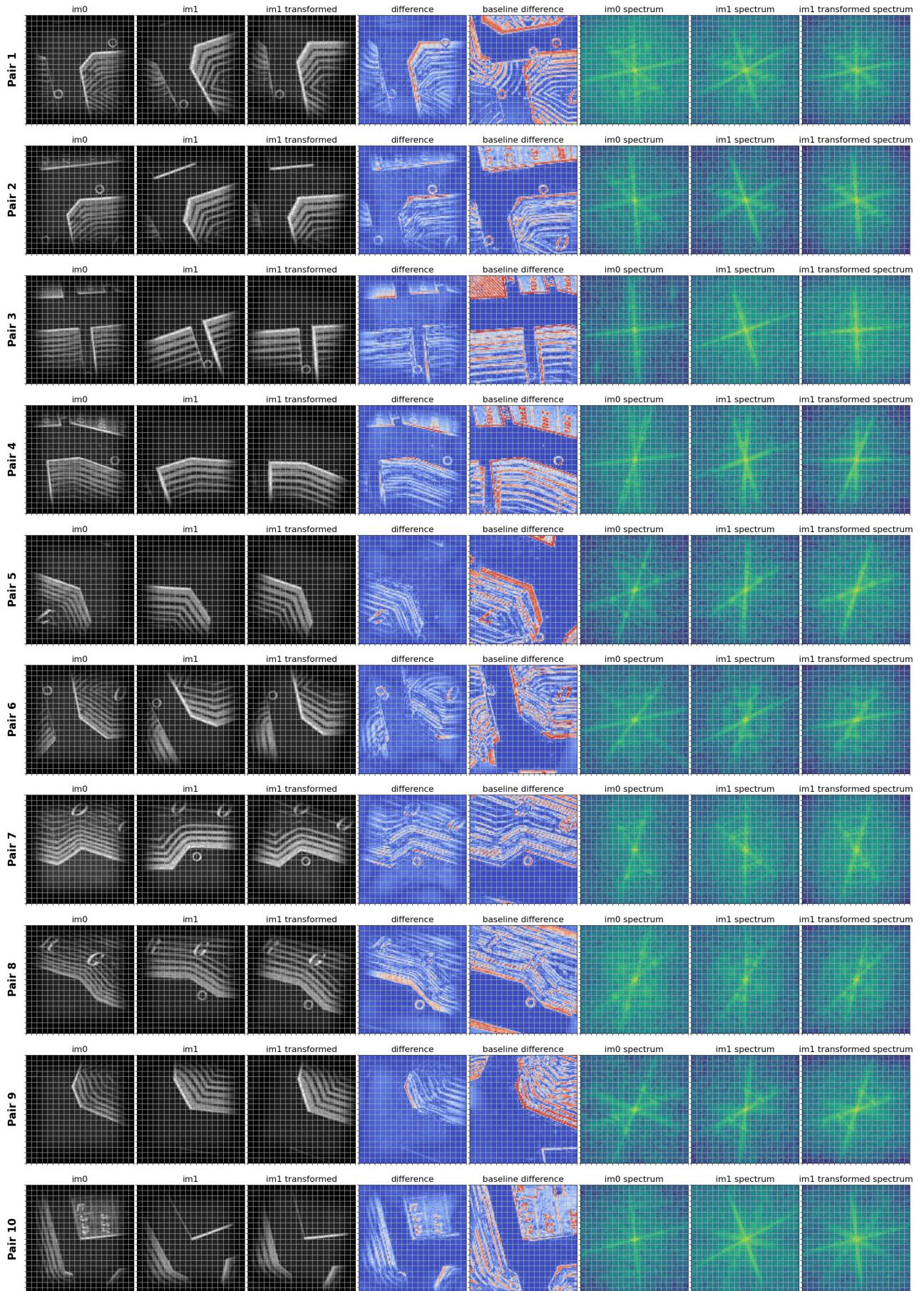


Figure 21: Test result: dataset 1 with tukey window applied

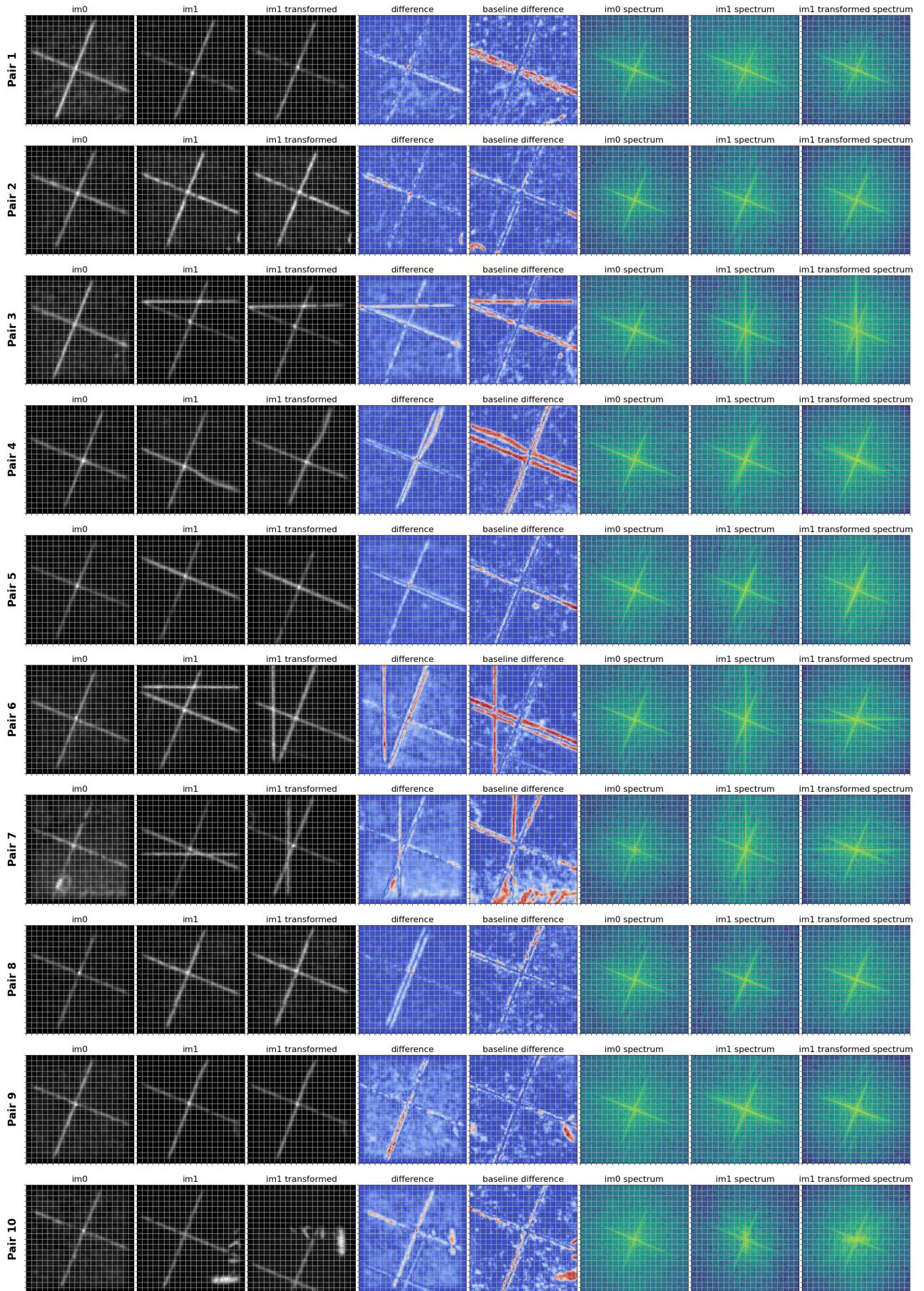


Figure 22: Test result: dataset 2 with tukey window applied

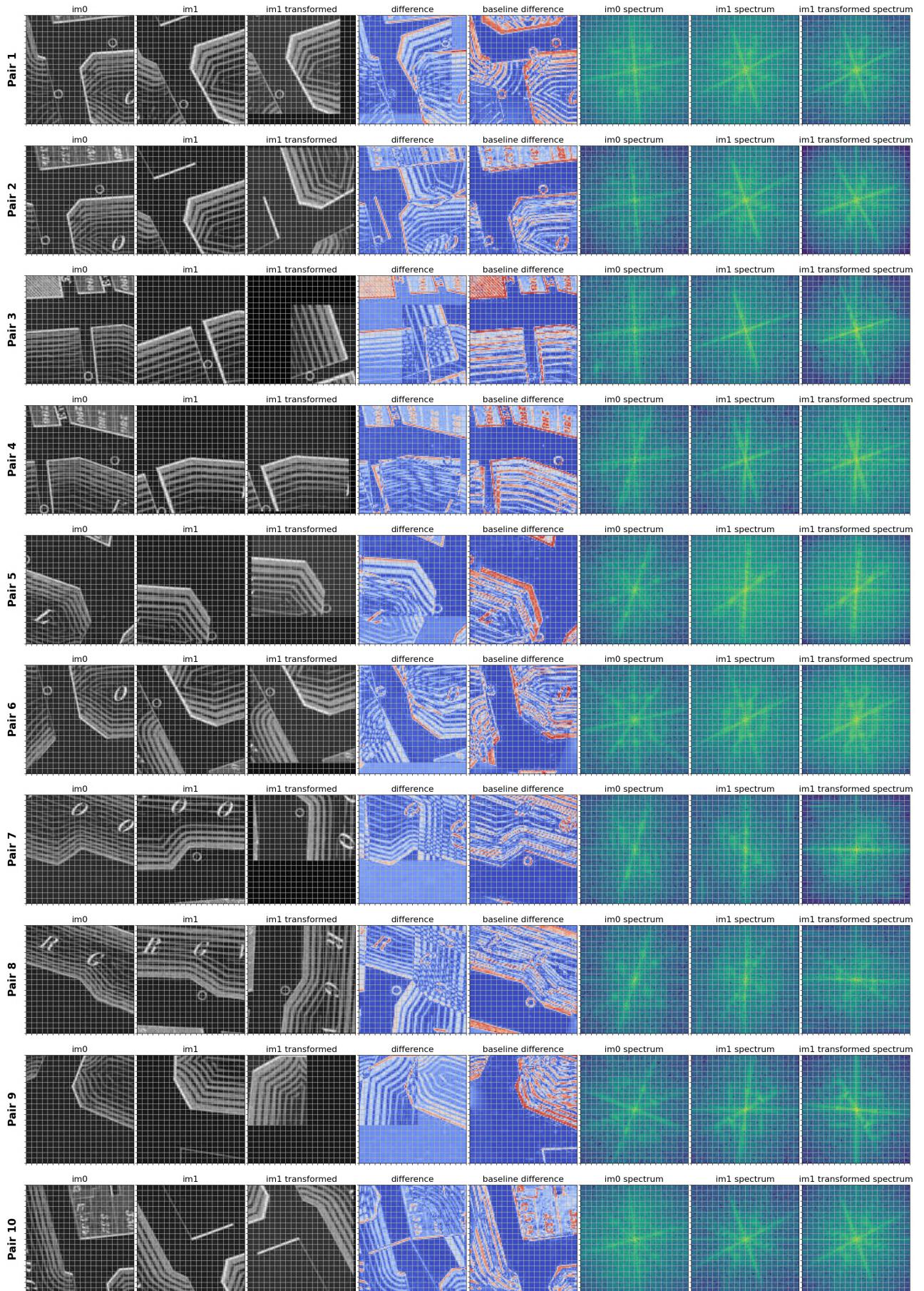


Figure 23: Test result: dataset 1 with 0-padding to 300×300 px

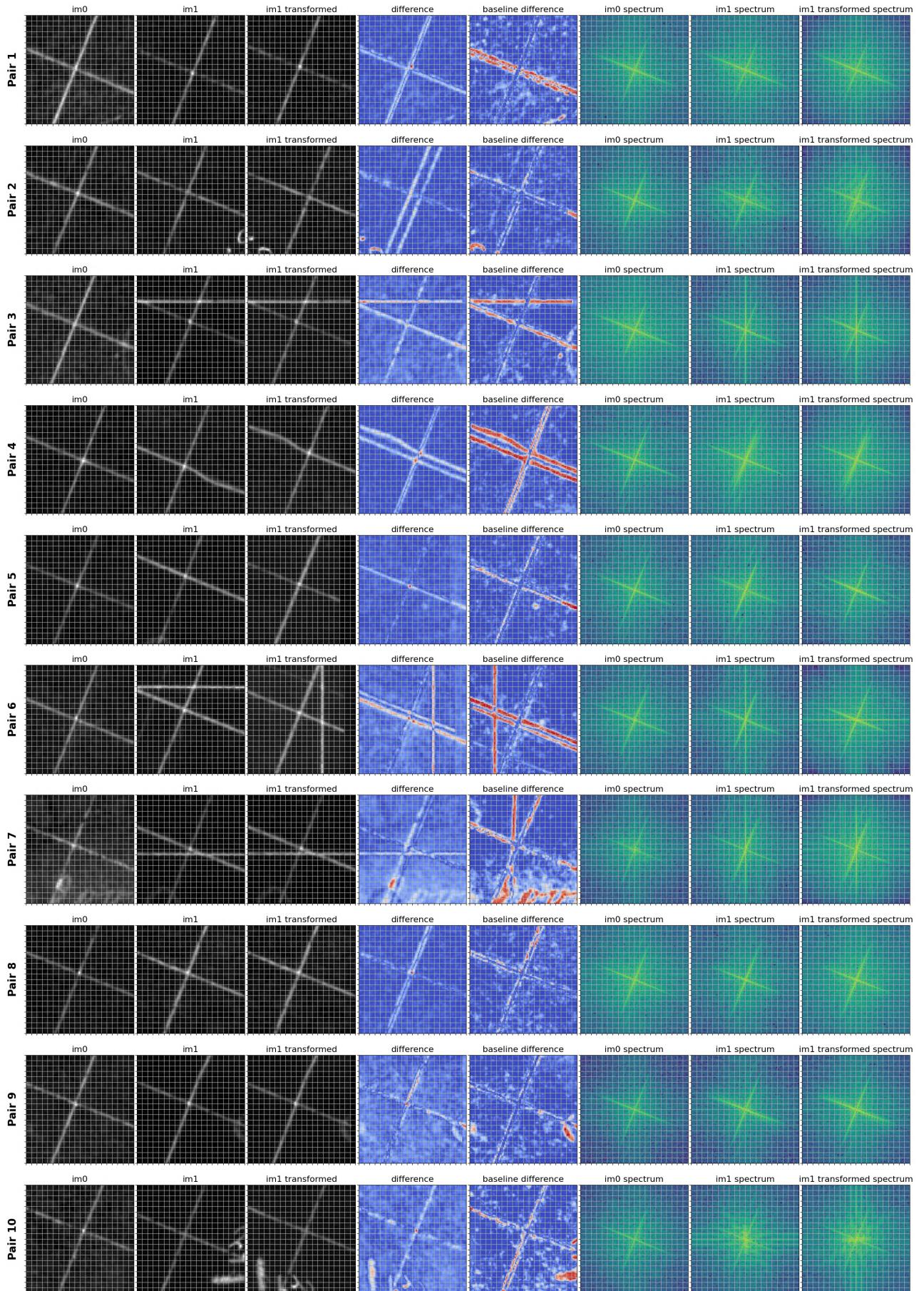


Figure 24: Test result: dataset 2 with 0-padding to 300×300 px

5 Conclusion

This study investigated the application of Fourier-Mellin Transform-based image registration for aligning hand-drawn historical map patches. Through systematic evaluation across controlled and real-world datasets, several key findings emerged regarding the method’s performance and limitations.

5.1 Key Findings

The baseline evaluation with controlled ground truth data demonstrated that the method achieves high accuracy under ideal conditions. All errors remained below 2 pixels for translation and 1 degree for rotation, confirming the theoretical viability of the approach for hand-drawn map registration.

However, real-world performance revealed significant challenges. The patch size experiments showed a clear performance hierarchy: $30\text{px} < 50\text{px} \approx 100\text{px}$, with smaller patches suffering from insufficient frequency resolution. The smaller-size patches showed more frequent registration failures due to coarse sampling that hindered spatial frequency discrimination.

Experiments on other pre-processing methods revealed that common enhancement techniques proved counterproductive for this specific application. Gaussian blur degraded performance by attenuating the sharp grid features essential for correlation peaks. Tukey windowing similarly reduced accuracy by suppressing critical boundary information in small patches. Standard zero-padding also proved not optimal, as larger search spaces introduced false correlation peaks inappropriate for control point refinement tasks.

5.2 Limitations and Challenges

Despite theoretical success, the method exhibits notable limitations for practical deployment. Visual assessment revealed that even the best-performing configurations (50×50 and 100×100 patches) still contained noticeable errors of both rotation and translation. The hand-drawn nature of historical maps introduces irregularities that violate the periodic assumptions underlying DFT-based methods, leading to reduced correlation peak distinctiveness.

The absence of ground truth data for real historical maps hindered quantitative evaluation, limiting the ability to establish objective success thresholds. Additionally, the method’s sensitivity to drawing artifacts and local distortions suggests limited robustness for diverse historical map types.

5.3 Future Work

Several promising directions emerge from this study’s findings that could address the current limitations of DFT-based registration for hand-drawn historical maps.

First, a line-fitting approach could exploit the inherent grid structure or edges more directly. By applying line detection algorithms (such as Hough transforms) to identify grid lines, computing their intersection points, and pairing the most distinctive intersections between image patches, this method could bypass the frequency-domain limitations observed in small patches.

Second, implementing a RANSAC-like optimization algorithm could provide a more exhaustive search strategy. By randomly sampling rotation and translation parameters and evaluating the differences, this approach could overcome the issues that affect phase correlation in the presence of noise and artifacts. This stochastic search method would be particularly valuable for handling the unpredictable variations in hand-drawn maps.

Third, the scalability to full map tiles remains unexplored. Rather than processing small patches, direct registration of entire map tiles could provide richer spatial context and enhance the frequency resolution that was insufficient in this study. Full-tile registration could offer a more seamless and practical user experience.

Additionally, developing quantitative evaluation metrics through synthetic ground truth generation or manual annotation would enable more rigorous performance assessment.

References

- Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.
- Kuglin, C. D. (1975). The phase correlation image alignment method. In *IEEE Int. Conf. on Cybernetics and Society, 1975*, pages 163–165.
- Lentz, J., Sevil, H. E., and Fries, D. (2025). Image preprocessing to enhance phase correlation of featureless images. *Scientific Reports*, 15(1):10287.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110.
- Reddy, B. S. and Chatterji, B. N. (1996). An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE transactions on image processing*, 5(8):1266–1271.
- Sarvaiya, J. N., Patnaik, S., and Bombaywala, S. (2009). Image registration using log-polar transform and phase correlation. In *TENCON 2009 - 2009 IEEE Region 10 Conference*, pages 1–5.
- Týč, M. and Gohlke, C. (2011–2015). imreg_dft: Image registration using discrete fourier transform. https://github.com/matejak/imreg_dft. Brno University of Technology, Brno, Czech Republic and Laboratory for Fluorescence Dynamics, University of California, Irvine.