Practical 3: Imaging Using Structured Light

1. Introduction

Optical microscopy is limited by diffraction to ~200 nm lateral resolution (Abbe, 1873). Super Resolution Structured Illumination Microscopy (SR-SIM) overcomes this by projecting known spatial patterns and extracting high-frequency information via frequency mixing (Gustafsson, 2000). This enables e.g. visualization of sub-cellular structures and processes in live cells with minimal phytotoxicity (Wu et al., 2018).

SR-SIM illuminates the sample with phase-shifted cosine patterns. The Moiré effect heterodynes high-frequency sample details down into the passband of the objective's Optical Transfer Function (OTF). Demodulation recovers these side-bands, giving up to 2× lateral resolution gain. Reconstruction methods range from simple summation of weighted bands to Wiener-filter deconvolution for noise suppression (Wiener, 1949; Chen et al., 2006, Chen et al., 2023).

In this coursework, I first validated Microsoft Copilot's (Microsoft Corporation, 2023) ability (Task 1), and had the Copilot generate a basic SR-SIM pipeline (Task 2). I tested Copilot's output via log-magnitude FFT (Nussbaumer et al., 1982), radial Power Spectral Density (PSD) for noise-floor detection, and Fourier Ring Correlation (FRC) for resolution (Task 3) (Banterle et al., 2013). I next developed a custom end-to-end SR-SIM pipeline modelling Gaussian PSF blur, Poisson shot noise, and Gaussian read noise, with five phases and three angles (Task 4) (Chen et al., 2023). I also tested this pipeline visually and quantitatively against widefield and simple-sum reconstructions (Task 5). Further, I performed grid-search and iterative testing to optimise key parameters and obtain statistically significant results.

Copilot's single-frame SR-SIM yielded a 1.35x PSD gain but no FRC improvement. My custom Wiener-filtered pipeline achieved a **1.741x average FRC gain**, approaching theoretical limits of 2x (Gustafsson, 2000). It matched empirical gains of 1.5–1.8x under realistic noise (Nieuwenhuizen et al., 2013).

2. Methods

All algorithms were run, tested and compared on an example image provided in the lectures (but also run with RGB images). This image serves as the ground truth (Figure 1). It was then blurred to simulate how a microscope acquired the image. Then a copilot generated SR-SIM algorithm, and two variations (sum-reconstruction and Wiener-filtered reconstruction) of a more complex algorithm were applied to try to reconstruct the original image and recover high frequency information. These algorithms were then tested using log-magnitude FFT (Nussbaumer et al., 1982) visualisation, radial Power Spectral Density (PSD) analysis for noise-floor detection, and Fourier Ring Correlation (FRC) for resolution assessment (Banterle et al., 2013).

2.1 Dependencies

This project uses the python libraries NumPy, Matplotlib's pyplot module, SciPy's signal.fftconvolve for fast convolution via Fast Fourier Transform and ndimage.fourier_shift for precise frequency-domain image shifts, scikit-image's routines and ImageIO v3 for robust image file reading/writing.

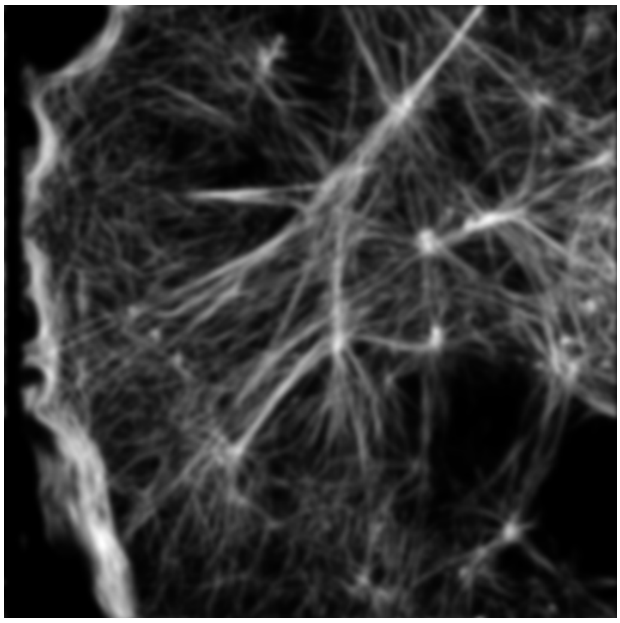

Figure 1. *Used Ground-Truth Image*

2.3 Copilot Verification (Task 1)

I validated Copilot by recreating the first part of the code from the second structured illumination lecture. Copilot was provided with a detailed prompt to load an image, compute its 2D Fourier transform, apply a circular low-pass filter whose radius is controlled by an interactive slider, and then inversely transforms back to show the filtered image. The prompt can be seen in the provided promts.txt file.

2.4 Copilot SR-SIM Retrieval Algorithm (Task 2)

I decided to have Copilot create the simplest possible SR-SIM retrieval algorithm. This was then the starting point to create a more complex, better algorithm in task 4. To create a simple SR-SIM retrieval algorithm I gave Copilot one detailed prompt and then 9 further quick follow up prompts. As suggested in the assignment I divided the task for copilot into the provided sub-steps. modulation, Fourier transform, shift-and-recombine, filtering, inverse transform. I also told copilot to add gaussian noise ($\sigma = 0.01$; models sensor read noise) to the modulated image. I told copilot to write a modular function for each step and a main pipeline function in the end. This pipeline then displays original, modulated, and reconstructed images side by side.

2.5 Testing Copilot Algorithm (Task 3)

To test the copilot algorithm, as well as for all other downstream tasks, I implemented a unified image-loader (setup_and_load_image). The functions converts all images (also RGB) to single-channel (grayscale) floating-point arrays in [0, 1] for algorithmic simplicity. This ensures all subsequent processing receives a consistent, bounded input.

Before numerically testing the copilot's algorithm I first plotted the log-magnitude FFT images for visual evaluation of the algorithm. This gives an overall idea of the functioning of the algorithm. It also shows potential problems with noise. Then I chose to conduct radial Power Spectral Density (PSD) analysis for noise-floor detection with Fourier Ring Correlation (FRC) for resolution assessment. FRC is to the gold-standard test in SR-SIM studies (Chan et al., 2023).

The function radial_psd(img) centers the FFT of the zero-mean image, computes the squared magnitude (power spectrum), bins power by integer radius, and returns average power per radius to produce an isotropic PSD curve. In k_cut_from_psd(psd, outer_frac=0.1, r_min=2), I estimated noise floor from the highest 10% of frequencies (outer_frac) and ignored the central radii < 2 to avoid low-frequency bias, returning the smallest radius where power falls below the noise estimate. I implemented frc(img1, img2), computing cross-power spectrum numerator ($F1 \cdot F2^*$) and normalizing by individual power spectra binned by radius to yield the FRC curve, and frc_cutoff(img1, img2, threshold=0.2071) to find the first frequency index where FRC drops below the 1/2-bit threshold (~0.2071), the standard resolution cutoff. I added plot_frc_curve(...) to plot normalised spatial frequency (cycles/pixel, Nyquist = 1) and overlay the threshold line.

The function test_copilot_sim_reconstruct runs the entire testing pipeline. It loads the ground truth, generates a random grating, ran one SIM reconstruction, and displays FFTs of ground truth, modulated, and reconstructed images with log-magnitude. Then it computes radial PSD cut-offs (k_mod, k_rec) and their ratio, simulates two independent noisy acquisitions (A & B), computes FRC cut-offs for raw vs reconstructed, prints the resolution gain, plots FRC curves, and returned a dictionary of key metrics.

2.6 Custom SR-SIM Implementation (Task 4)

I developed an end-to-end SR-SIM pipeline that models realistic imaging physics and harmonic demodulation. The pipeline simulates optical blur and noise sources. I selected modulation parameters (depth 0.8, 3 angles, 5 phases) for high spectral coverage.

The function rescale(img) shifts image minimum to zero and divides by dynamic range + ε for display and PSD consistency. In define_psf(shape, σ_px), I built a centered 2D Gaussian PSF (Point-Spread Function) (Claxton et al., 2007) that normalises total energy to unity (photon count conservation). In simulate_widefield(img, psf, gaussian_std=0.01, photons=1e4), I convolved the image with PSF via fftconvolve, replaced NaNs, scaled by photon count λ (Poisson shot noise), added Gaussian read noise, and clipped to [0, 1] to simulate widefield microscope acquisition.

Generate_sim_patterns creates cosine gratings $1+\text{modulation}*\cos(2\pi(kx*x+ky*y)+\phi)$ across multiple angles and phase shifts to

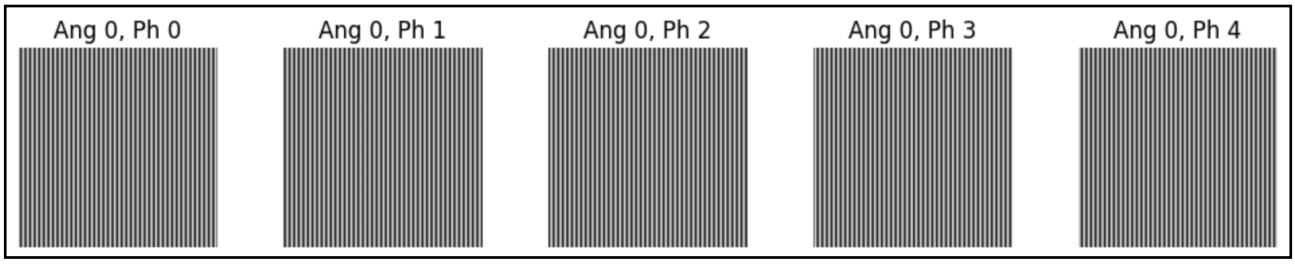

Figure 2. *Selected Illumination Patterns*

encode high-frequency information (Figure 2). In simulate_sim_raw, I applied each pattern, blurred, and added noise to form a 3D stack of raw SIM frames.

The function preprocess_sim_frames(frames) computes the background (global minimum) and dynamic range for normalisation. I implemented mirror_pad using reflect padding with a 50% padding fraction (see Figure 3). This mirror extension reduces FFT ringing artefacts in the images.

In reconstruct_fft_components_roll, I computed FFTs of each frame, extracted DC (0th order) and ±1 side-bands via phase-weighted sums over phases, calculated integer shifts from k-vector components, and rolled side-bands to baseband for recombination.

In the first variation of the algorithm, the function simple_sim_reconstruct then sums DC and side-band components weighted by 1/(2·modulation) (normalising contrast), average over angles (improving SNR), and inverse FFT real part for the reconstructed image. The padded frames were then cropped back to the original size.

The second variation of the algorithm conductions Wiener-filtered reconstruction (sim_reconstruct_wiener). I computed the Optical Transfer Function (OTF) as the FFT of the PSF, applied the Wiener filter $H^* / (|H|^2 + \varepsilon)$ (H^*: complex conjugate; ε: regularisation to prevent noise amplification), rolled filters for side-bands, normalised by total weight to maintain brightness consistency, and performed inverse FFT real part. I was not able to reconstruct padded images using the Wiener filter. These were flipped and full of noise artefacts. After many unsuccessful attempts to fix this issue, I resorted to using unpadded data for the Wiener-filtered reconstruction.

2.7 Testing Custom SR-SIM Implementation (Task 5)

I integrated my custom reconstruction into a single pipeline (sim_pipeline) that provides the same visual and quantitative comparisons as in 2.5 with a widefield baseline.

The pipeline displays ground truth, widefield, SR-SIM sum, and SR-SIM Wiener images alongside their log-magnitude FFTs. I computed radial PSD cut-offs for each and printed noise-floor crossing gains. I generated two independent half-stacks for FRC analysis, printed and plotted FRC cut-off gains relative to widefield, and returned a dictionary of images and metrics.

These test results vary from run to run as they are subject to the random noise used to simulate the widefield image. Thus, I conducted statistically significant testing next. I performed repeated pipeline runs across independent noise generations using the iterative_testing function, which gave statistically more significant average performance metrics.c The functions collects

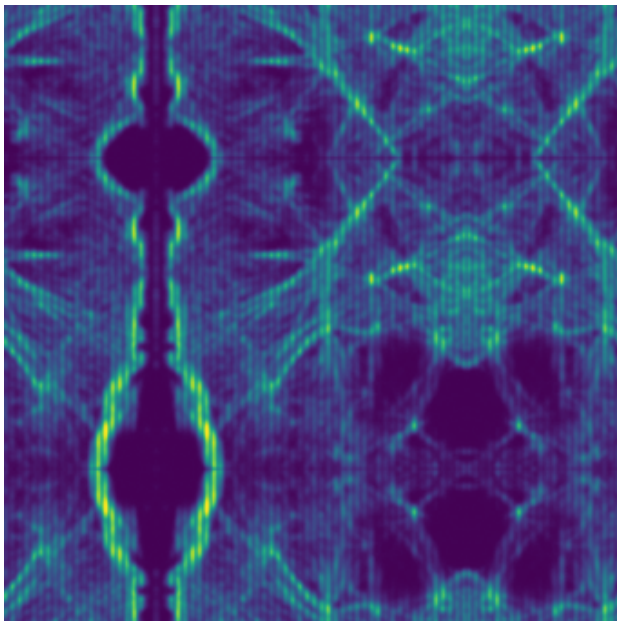

Figure 3. *Example Padded Frame*

lists of radial PSD cut-offs and FRC cut-offs for both sum and Wiener reconstructions, and stacks the full FRC curves. After converting lists to NumPy arrays, it computes average resolution gains relative to the widefield baseline. It plots bar charts of average noise-floor and FRC 1/2-bit gains. It also plots average FRC curves.

2.8 Parameter Fine-Tuning

I then amended the iterative_testing function with the grid_search_testing function to fine-tune the otherwise arbitrary hyper-parameter choices (sigma_px, modulation, num_angles, num_phases, gaussian_std, photons, eps). This function iterates the iterative_testing function over a search space of possible parameter combinations to find the best possible parameters. It optimises for the highest Wiener-filtered reconstructed average FRC 1/2-bit cutoff gain. FRC 1/2-bit cutoff gain was choses as it measures the true resolution increase. Wiener-reconstruction algorithm was chosen as it consistently outperforms the sum-reconstruction algorithm. The function was run over a defined search space (see 3.5) with 25 iterative_testing iterations per combinations.

3. Results

3.1 Copilot Verification

Copilot successfully reproduced the first part of the code discussed in the lecture. The produced code loads an image, computes its 2D Fourier transform, applies a circular low-pass filter whose radius is controlled by an interactive slider, and then inversely transforms back to show the filtered image. It did so with only one prompt and one attempt.

3.2 Copilot SR-SIM Retrieval Algorithm

The copilot created a SR-Sim retrieval algorithm as described in methods using 10 prompts. The image output of said function can be seen in Figure 4. They show the original, modulated, and reconstructed example image.

3.3 Testing Copilot Algorithm

Figure 5 displays the log-magnitude FFTs. The ground-truth image exhibits the strongest spectral content, the modulated frame the weakest, and the reconstructed image approximately doubles the modulation's magnitude. The radial PSD analysis indicates that the modulated frame's noise-floor crossing occurs at k = 79, while the reconstruction extends to k = 107, representing a 1.35× gain. This results is subject to slight variation due to random noise. Independent-stack FRC 1/2-bit cut-offs for both the modulated frames (A vs B) and the reconstructed frames (A vs B) fall at 1/362 px, yielding no net resolution gain. This results remains unchanged even after many iterations of the experiment. The corresponding FRC curves are plotted in Figure 6.

3.4 Custom SR-SIM Implementation

The image outputs of my SR-Sim retrieval algorithm as described in methods can be seen in Figure 7. (sigma_px=3, modulation=0.7, photons=50000, eps=0.1). They show the original, simulated widefield, sum-reconstructed, and Wiener-reconstructed example image.

3.5 Testing Custom SR-SIM Implementation

Figure 8 displays the log-magnitude FFTs for the original, simulated widefield, sum-reconstructed, and Wiener-reconstructed example image (sigma_px=3, modulation=0.7, photons=50000, eps=0.1). The sum reconstructed SIM shows only minimal improvements over the small widefield FFT. The Wiener reconstructed SIM FFT is slightly larger as that of the ground truth image. This may be due to noise and the lack of padding.

A grid search over σ_px: [1.5, 2.0], modulation: [0.5, 0.7, 0.9], and ε: [0.01, 0.1] identified the optimal parameters (σ_px = 2.0, modulation = 0.7, num_angles = 3, num_phases = 5, gaussian_std = 0.01, photons = 5×10^4, ε = 0.01) for the Wiener-reconstructed SF-SIM algorithm. It achieved a Wiener FRC gain of **1.741×**. Figure 9 reports average noise-floor crossing gains of 1.00x (widefield), 1.21x (SR-SIM Sum), and 1.41x (SR-SIM Wiener). Figure 9 also reports average FRC 1/2-bit cutoff gains of 1.00x (widefield), 0.90x (SR-SIM Sum), and 1.74x (SR-SIM Wiener).

The average FRC curves in Figure 10 show that the Wiener reconstruction (orange) maintains near-perfect correlation at low spatial frequencies but then plunges sharply between normalised frequencies 0.4–0.5. It crosses the 1/2-bit threshold (~0.207) at ≈0.45. In contrast, the simple-sum

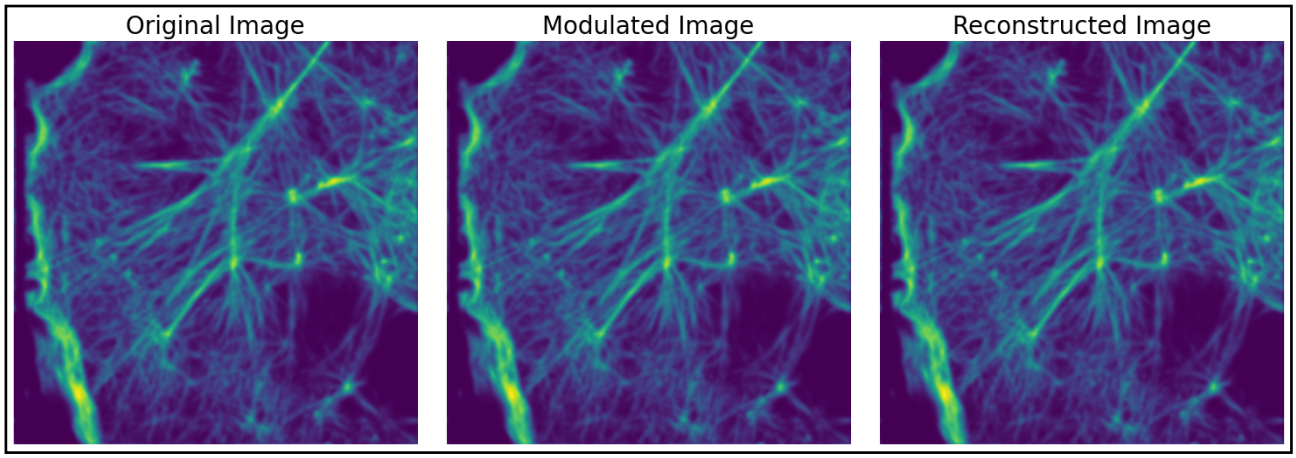

Figure 4. *Images from Copilot ST-SIM Reconstruction Algorithm*

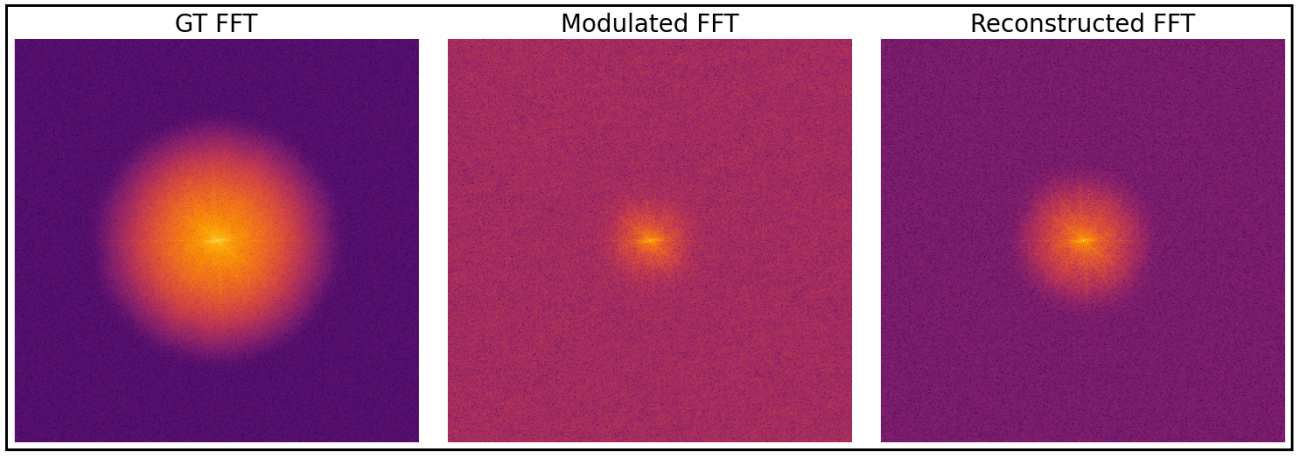

Figure 5. *FFT from Copilot ST-SIM Reconstruction Algorithm*

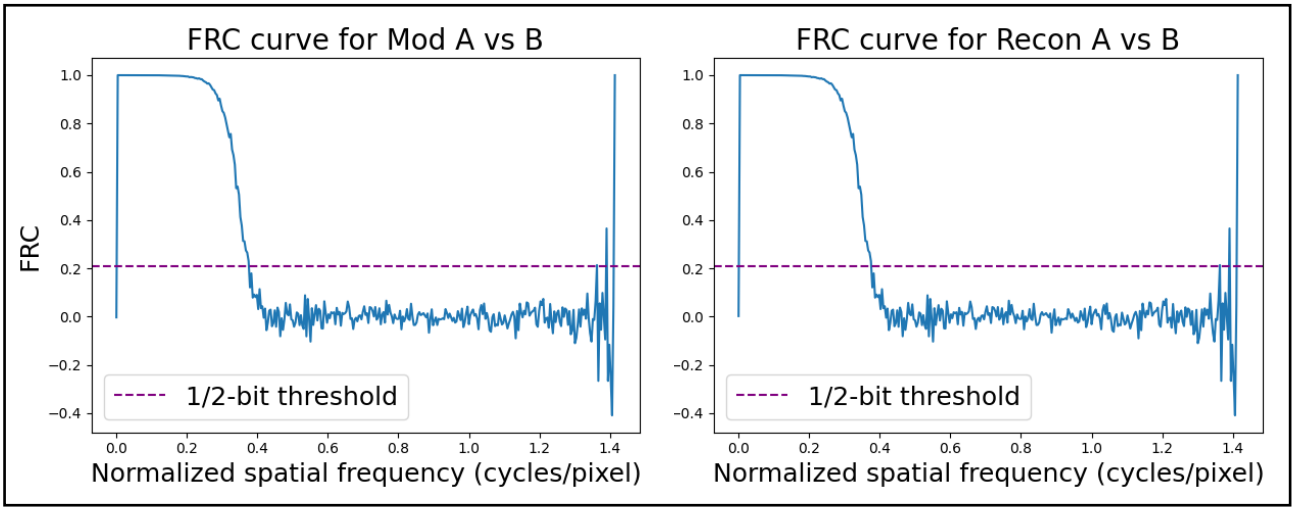

Figure 6. *FRC curves from Copilot ST-SIM Reconstruction Algorithm for two independent modulated and reconstructed images each.*

reconstruction (blue) decays more gradually and only falls below the threshold at ≈1.0. Thus, although Wiener filtering yields very high fidelity at low frequencies, the sum method preserves correlation out to higher normalised frequencies.

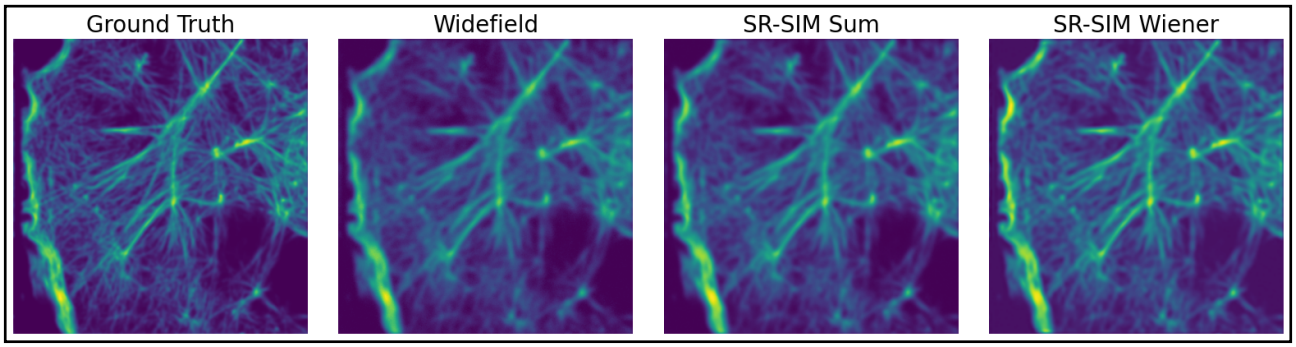

Figure 7. *Images from Custom ST-SIM Implementation*

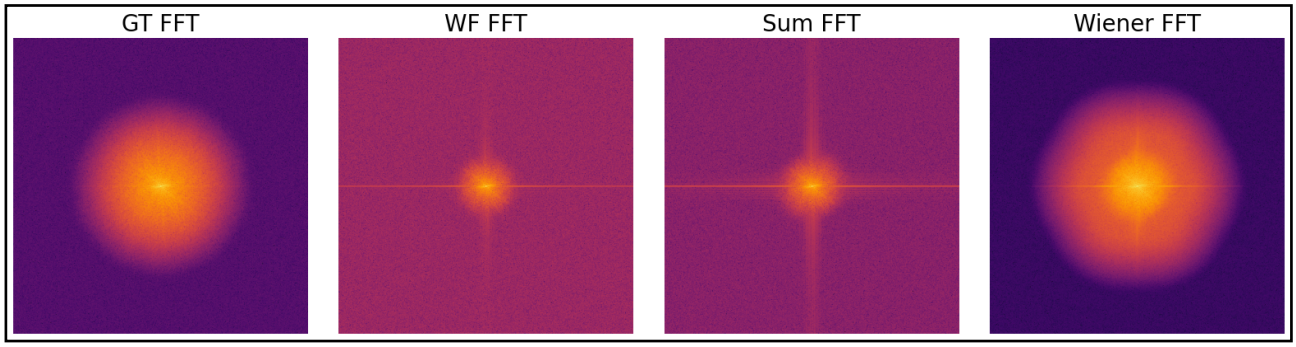

Figure 8. *FFT from Custom ST-SIM Implementation*

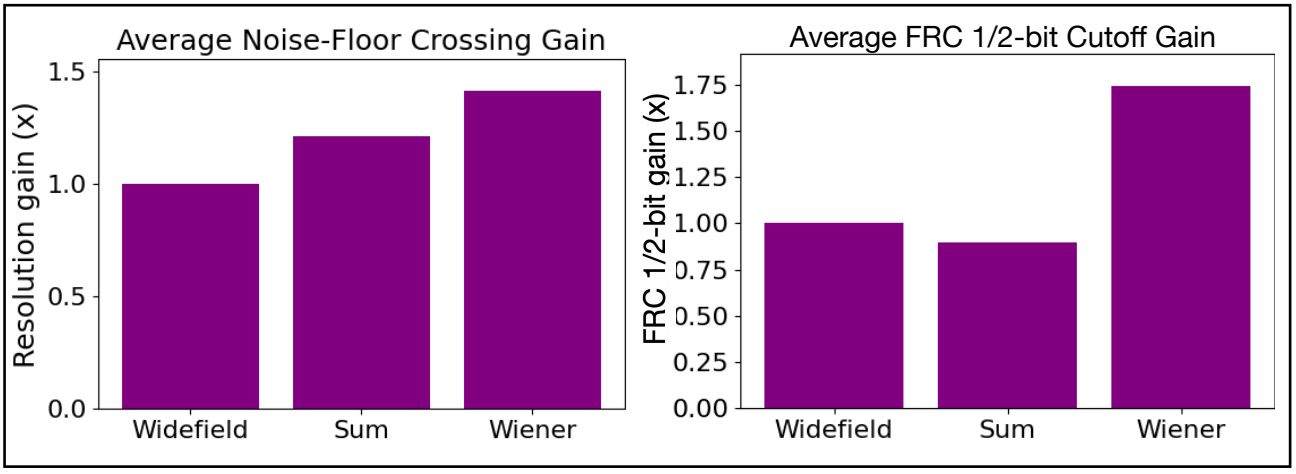

Figure 9. *Average Noise-Floor Crossing Gain and Average FRC 1/2-bit Cutoff Gain of sum-reconstruction and Winer-reconstruction against simulated widefield image(s) with best found hyper parameters.*

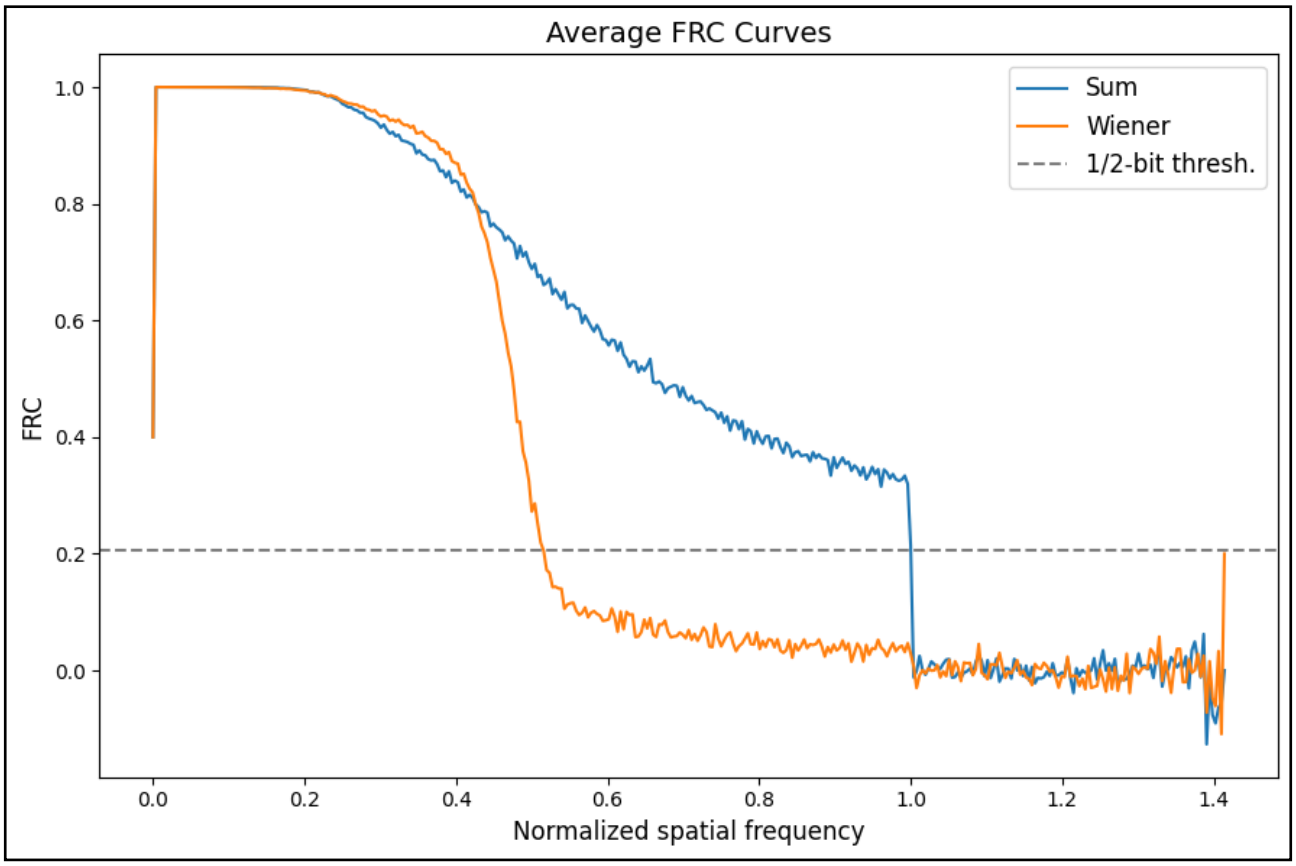

Figure 10. *Averaged FRC curves of sum-reconstruction and Winer-reconstruction with best found hyper parameters.*

4. Discussion

4.1 Prompting (Task 6)

Generally more detailed prompts improve output quality (White et al., 2023). My detailed instructions for copilot worked. This is also true for brainstorming prompts that I used in later sections. In general, the more constraints I put on Copilot, the more precisely the output aligns with my expectations. This is especially true for coding, where I specified the exact functions to write for the initial SR-SIM algorithm.

At a certain depth into the project Copilot struggled and often failed to provide good ideas for optimisation and fixes for issues. Copilot also sometimes contradicted itself. E.g. after suggesting one fix to my Wiener-reconstruction padding issues, it later named exactly this part of the code as the reason for bad performance. Therefore, while Copilot is useful for brainstorming and saving time, it is limited to aid in cutting edge SIM projects.

The first most useful prompt I used was a brief project description with “Give me an extensive, detailed numbered list of all the things that have to be done, the steps that need to be taken. Each should be explained very briefly in simple terms. Format as hierarchical numbered list."

A second useful prompt was a project description with “Below is my code so far. Give me simple things I can tweak that will make the reconstruction SR SIM higher resolution. I want to recover more information.” Not all of Copilot’s ideas being actually useful, induced much work. But still it gave much good feedback that in the end was essential to my algorithm.

Third, I personally found that asking for explanations is not only useful for me the user, but also improves the Copilot’s performance. That is why I always ask it to explain an error before given solutions. this introduces a reasoning step and forces the LLM to think through the problem actively.

4.2 Copilot SR-SIM Retrieval Algorithm

Copilot delivered a modular SR-SIM pipeline in ten prompts, encapsulating modulation, FFT, shift-and-recombine, filtering, and inverse

FFT stages. This structure mirrors the canonical algorithm for doubling lateral resolution in SIM (Gustafsson, 2000). However, the one-frame implementation lacked multi-angle, multi-phase data and thus did not achieve any real resolution improvement (much less achieve the theoretical 2x). Furthermore, the modulated images did not realistically simulate a widefield image.

4.3 Custom SR-SIM Implementation

My pipeline models realistic imaging physics (Gaussian PSF blur, Poisson shot noise, and Gaussian read noise). It then applies harmonic demodulation with five phases and three angles (Chen et al., 2023). This approach aligns with simulation studies that demonstrate improved resolution and contrast when realistic noise is included (Nieuwenhuizen et al., 2013). A remaining limitation is the idealised Gaussian PSF. Measuring an experimental PSF may further enhance accuracy.

While I employed padding before sum reconstruction (Figure 3.), I was not able to reconstruct padded images using the Wiener filter. These were always flipped and full of odd artefacts. After many unsuccessful attempts to fix this, issue resorted to using unpadded data for the Wiener reconstruction. This may explain why Figure 8 shows so much high frequency noise in the Wiener-reconstructed image. Conducting the wiener-reconstruction with padded images may improve resolution.

Finally, implementing subpixel shifts via Fourier shift theorem (Balci et al., 2006) rather than integer rolls would reduce interpolation error.

4.4 Testing Custom SR-SIM Implementation

Sum reconstruction preserved correlation out to normalised frequency ≈1.0, while Wiener filtering peaked at low frequencies and fell sharply beyond ≈0.45 (Figure 10). This behavior illustrates an expected tradeoff in Wiener deconvolution between noise suppression and high-frequency restoration (Wiener, 1949, Chen et al., 2006).

Grid search over key parameters (σ_px, modulation, angles, phases, σ_noise, photon count, ε) optimised the Wiener FRC gain to an average of 1.741x (Figure 9). A 1.741x FRC gain under Wiener-filtered SR-SIM approaches the theoretical twofold resolution enhancement predicted for ideal three-angle, multiple-phase SIM systems (Gustafsson, 2000). Empirical studies report practical gains of 1.5–1.8x when noise and optical aberrations are included (Nieuwenhuizen et al., 2013). Thus, my pipeline's 1.741x improvement aligns with the upper range of experimental performance. This speaks to the quality and precision of my algorithm.

The hyperparameter grid search was narrow and limited by compute time. In a future project expanding ranges and adopting adaptive search methods could yield greater gains. Setting the iterative search on more than 25 iterations per combinations would have further increased the statistical significance of the findings. Finally, validating on experimental SIM data will test real-world performance.

5. Conclusion

This project began by validating Microsoft Copilot's (Microsoft Corporation, 2023) ability to generate interactive Fourier-mask code. I then guided Copilot to produce a basic single-frame SR-SIM pipeline (Chen et al., 2023). I quantified its performance via log-magnitude FFT (Nussbaumer et al., 1982), radial PSD, and FRC metrics (Banterle et al., 2013) . Next, I developed a more complex SR-SIM implementation that models Gaussian PSF blur, Poisson shot noise, and five-phase three-angle harmonic demodulation. I compared simple-sum and Wiener-filter (Wiener, 1949; Chen et al., 2006, Chen et al., 2023) reconstructions against a simulated widefield baseline. Then, I performed iterative testing to obtain statistically significant results. Finally, I conducted a grid search to optimise key parameters for maximal FRC gain.

Copilot's single-frame SR-SIM achieved a 1.35x increase in PSD cutoff but no FRC resolution gain. My custom sum reconstruction extended correlation to normalised frequency ≈1.0. The Wiener filter produced near-ideal low-frequency fidelity but dropped sharply beyond ≈0.45 (Figure 10). Grid search optimisation yielded a Wiener-filtered FRC gain of 1.741x (Figure 9), which approaches the theoretical two-fold improvement (Gustafsson, 2000). It also matches empirical values under realistic noise (Nieuwenhuizen et al., 2013).

These results demonstrate that accurate noise modelling and Wiener deconvolution can recover high-frequency details in SR-SIM. Remaining limitations include the lack of padding for Wiener reconstruction and a too narrow hyperparameter exploration. Future work should implement subpixel side-band shifts (Balci et al., 2006) to push performance closer to the diffraction-unlimited limit. Finally, the developed pipeline should be validated on experimental SIM data to test real-world performance.

References

1. Gustafsson, M. G. (2000). Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy. *Journal of microscopy*, *198*(2), 82-87.

2. Wu, Y., & Shroff, H. (2018). Faster, sharper, and deeper: structured illumination microscopy for biological imaging. *Nature methods*, *15*(12), 1011-1019.

3. Wiener, N. (1964). Extrapolation, interpolation, and smoothing of stationary time series. The MIT press.

4. Chen, J., Benesty, J., Huang, Y., & Doclo, S. (2006). New insights into the noise reduction Wiener filter. *IEEE Transactions on audio, speech, and language processing*, *14*(4), 1218-1234.

5. Nussbaumer, H. J., & Nussbaumer, H. J. (1982). *The fast Fourier transform* (pp. 80-111). Springer Berlin Heidelberg.

6. Chen, X., Zhong, S., Hou, Y., Cao, R., Wang, W., Li, D., ... & Xi, P. (2023). Superresolution structured illumination microscopy reconstruction algorithms: a review. *Light: Science & Applications*, *12*(1), 172

7. Microsoft Corporation. (2023). Microsoft Copilot [Computer software]. Retrieved May 2, 2025, from https://copilot.microsoft.com

8. Banterle, N., Bui, K. H., Lemke, E. A., & Beck, M. (2013). Fourier ring correlation as a resolution criterion for super-resolution microscopy. *Journal of structural biology*, *183*(3), 363-367.

9. Nieuwenhuizen, R. P., Lidke, K. A., Bates, M., Puig, D. L., Grünwald, D., Stallinga, S., & Rieger, B. (2013). Measuring image resolution in optical nanoscopy. *Nature methods*, *10*(6), 557-562.

10. Abbe, E. (1873). Beiträge zur Theorie des Mikroskops und der mikroskopischen Wahrnehmung. *Archiv für mikroskopische Anatomie*, *9*(1), 413-468.

11. Heintzmann, R., & Cremer, C. G. (1999, January). Laterally modulated excitation microscopy: improvement of resolution by using a diffraction grating. In *Optical biopsies and microscopic techniques III* (Vol. 3568, pp. 185-196). SPIE.

12. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., ... & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

13. Claxton, C. D., & Staunton, R. C. (2007). Measurement of the point-spread function of a noisy imaging system. *Journal of the Optical Society of America A*, *25*(1), 159-170.

14. Balci, M., & Foroosh, H. (2006). Subpixel estimation of shifts directly in the Fourier domain. *IEEE Transactions on Image Processing*, *15*(7), 1965-1972.

15. Heintzmann, R., & Huser, T. (2017). Super-resolution structured illumination microscopy. *Chemical reviews*, *117*(23), 13890-13908.