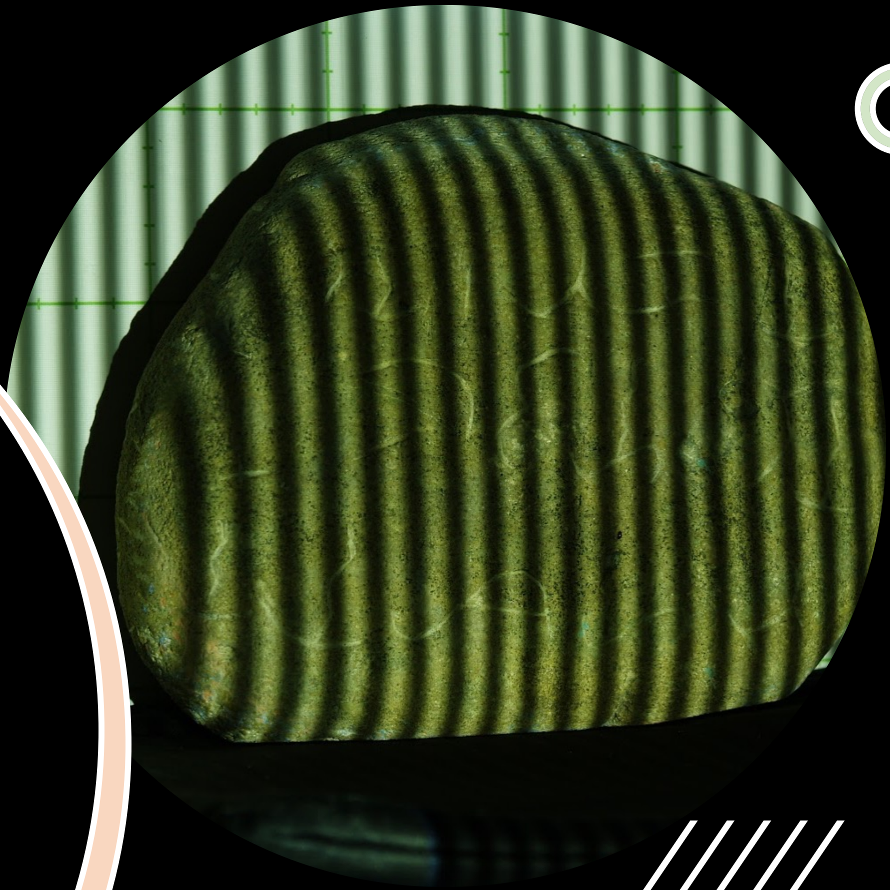
Activity 8: Phase Shift Profilometry



Physics 301
2nd Sem AY 21-22

Submitted by: Mark Jeremy G. Narag
Student Number: 2014-64423
Program: PhD Physics

Objectives

- Derive phase from images projected with phase shifted sinusoids
- Unwrap phase image using quality-guided phase unwrapping algorithm (optional)
- Reconstruct 3D from phase difference of reference and object image phase.

Shape from Structured Light

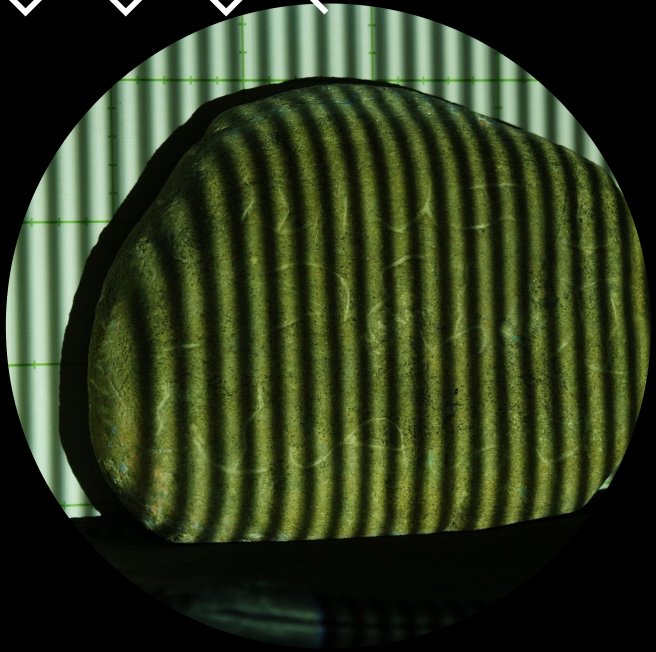


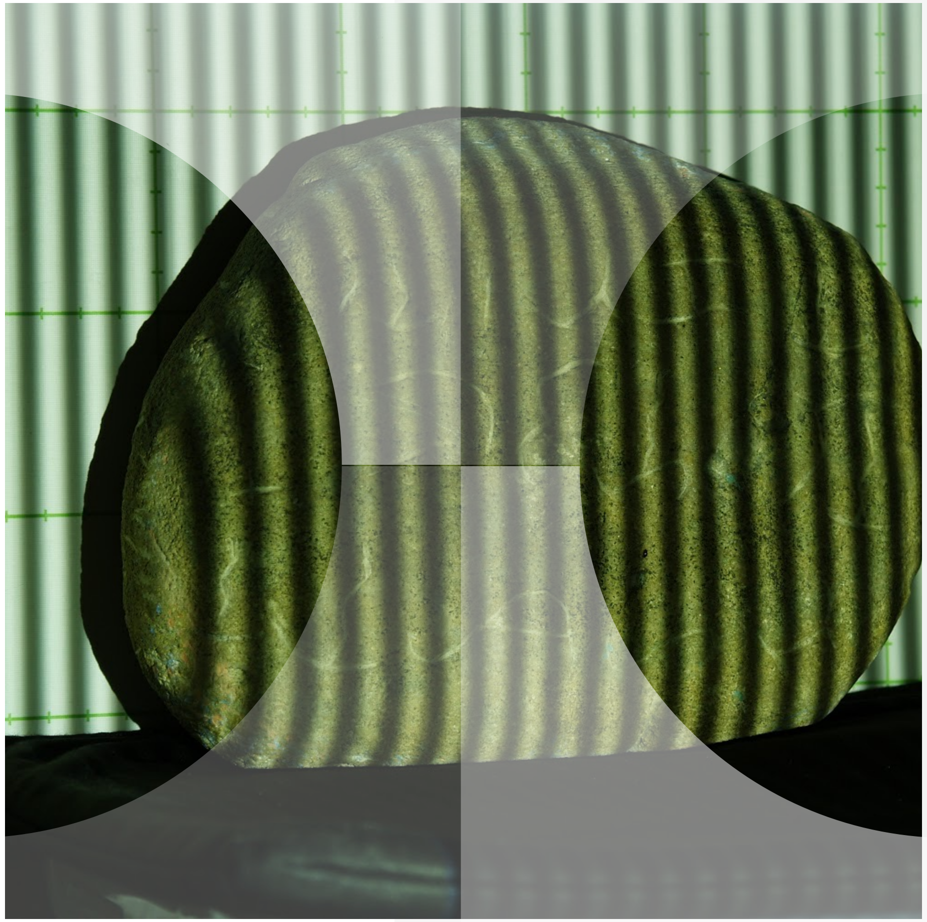
Figure 1. Projected sinusoidal pattern on Ticao stone bearing Baybayin symbols.

Three-dimension (3D) imaging techniques have a wide array of applications in industry, cultural heritage, medicine, and criminal investigation [1,2]. The 3D shape measurement technology seeks to characterize the geometric information or the surface of an object. Of the many forms of technologies in this field, one of these is the structured light branch, which is self descriptive in its non-contact approach to characterizing the surface of an object [3].

Structured light works by illuminating a sinusoidal pattern onto it just like in figure 1. Notice how the rock distorts the projected pattern. These deformations, although 2D, produce a basis for the recreation of the surface of a 3D object.

And using Phase Shift Profilometry, we can estimate the phases of the intensity patterns for all pixels on the image. We can also 3D this using Fourier Transform Profilometry.

[1] Sansoni, G.; Trebeschi, M.; Docchio, F. State-of-the-art and application of 3D imaging sensors in industry, cultural heritage, medicine, and criminal investigation. Sensors 2009, 9, 568–601.
[2] Blais F. A review of 20 years of range sensors development. J. Electronic Imaging. 2004;13:231–240.
[3] Zuo, C., Feng, S., Huang, L., Tao, T., Yin, W., Chen Q. Phase shifting algorithms for fringe projection profilometry: A review. Science Direct, Optica and Lasers in Engineering, Volume 109, Page 23-59. October 2018.



Phase Shift Profilometry

We can divide PSP into three main steps:

1. **Phase wrapping**
   i. Projecting 4 intensity functions
   ii. Subtracting like terms
   iii. Solving phase

2. **Phase unwrapping**

3. **Phase-to-height conversion**
   i. Estimating the reference
   ii. Getting phase difference
   iii. 3D reconstruction

Step 1: Phase wrapping, projecting four intensity functions

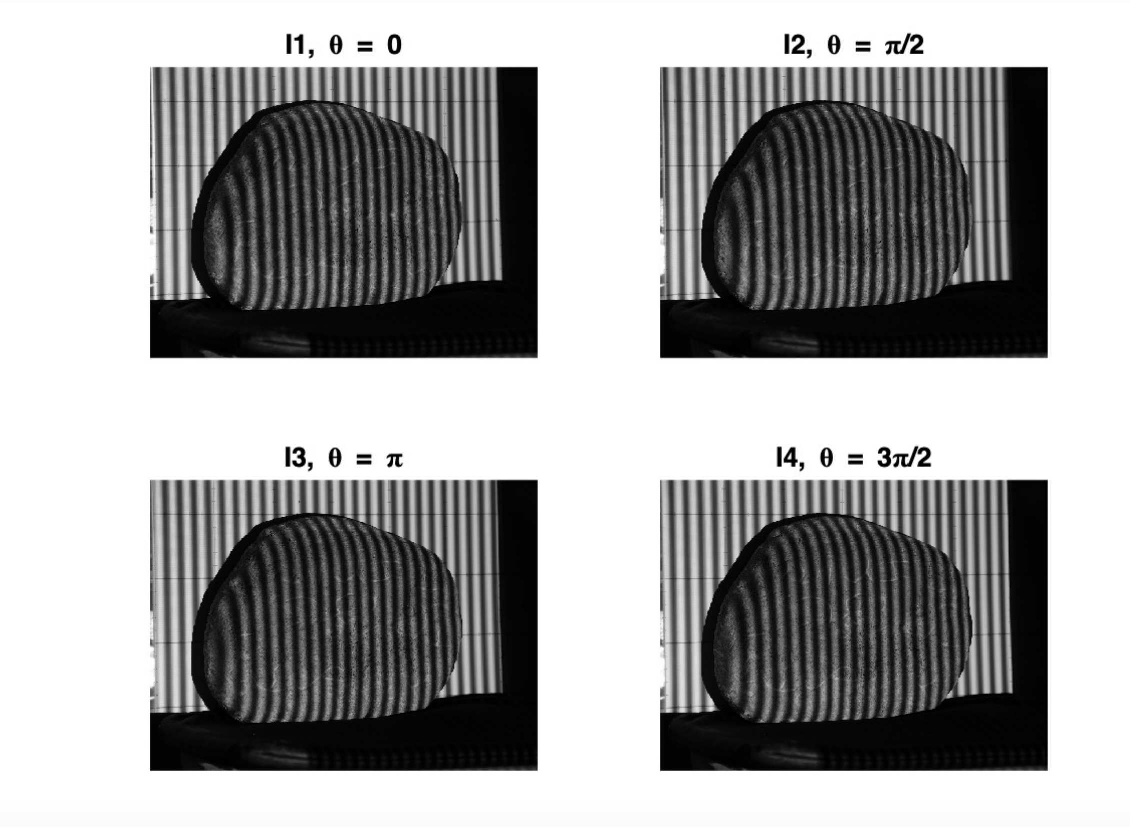
In this activity We have 4 phase shifted sinusoidal patterns (I1,I2,I3,I4) as shown in the matlab code. We can express these as intensity functions given by:

$$I_1(x,y) = I_o(x,y) + I_{mod}(x,y)cos(\phi(x,y))$$
$$I_2(x,y) = I_o(x,y) + I_{mod}(x,y)cos(\phi(x,y) + \frac{\pi}{2})$$
$$I_3(x,y) = I_o(x,y) + I_{mod}(x,y)cos(\phi(x,y) + \pi)$$
$$I_4(x,y) = I_o(x,y) + I_{mod}(x,y)cos(\phi(x,y) + \frac{3\pi}{2})$$

Where $I_o(x,y)$is the average background intensity, $I_{mod}(x,y)$is the intensity value of the fringe pattern and is the phase value. Similarly, we can express the above equations as

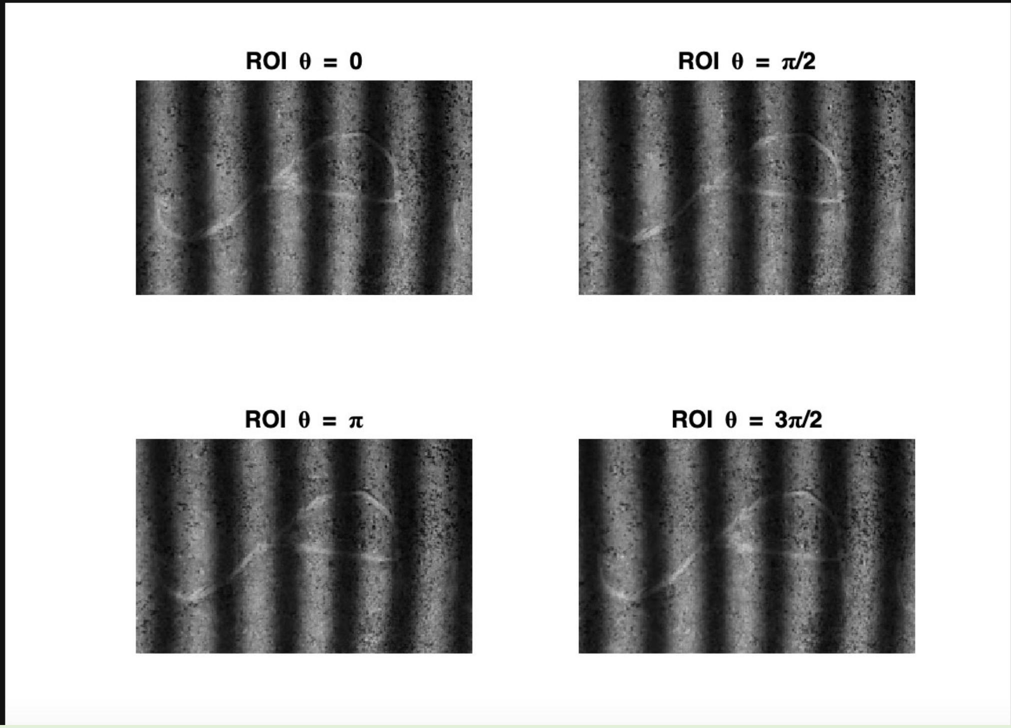
$$I_1(x,y) = I_o(x,y) + I_{mod}(x,y)cos(\phi(x,y))$$
$$I_2(x,y) = I_o(x,y) - I_{mod}(x,y)sin(\phi(x,y))$$
$$I_3(x,y) = I_o(x,y) - I_{mod}(x,y)cos(\phi(x,y))$$
$$I_4(x,y) = I_o(x,y) + I_{mod}(x,y)sin(\phi(x,y))$$

```
%loading the 4 images of different intensity functions
I1 = rgb2gray(imread('picture1.jpeg'));
I2 = rgb2gray(imread('picture2.jpeg'));
I3 = rgb2gray(imread('picture3.jpeg'));
I4 = rgb2gray(imread('picture4.jpeg'));
figure(1);
subplot(2,2,1); imshow(I1); title('I1, \theta = 0');
subplot(2,2,2); imshow(I2); title('I2, \theta = \pi/2');
subplot(2,2,3); imshow(I3); title('I3, \theta = \pi');
subplot(2,2,4); imshow(I4); title('I4, \theta = 3\pi/2');
```

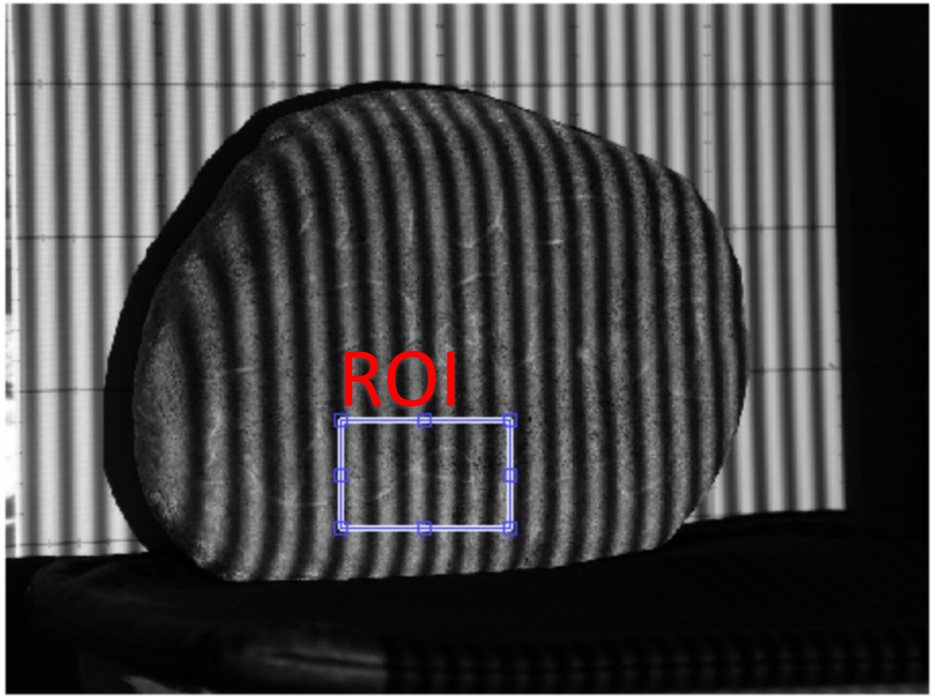


Step 1: Phase wrapping, projecting four intensity functions

```
%crop region of interest
[J1, RECT] = imcrop(I1);
J2 = imcrop(I2, RECT);
J3 = imcrop(I3, RECT);
J4 = imcrop(I4, RECT);
figure(2);
subplot(2,2,1); imshow(J1); title('ROI \theta = 0');
subplot(2,2,2); imshow(J2); title('ROI \theta = \pi/2');
subplot(2,2,3); imshow(J3); title('ROI \theta = \pi');
subplot(2,2,4); imshow(J4); title('ROI \theta = 3\pi/2');
```



Since our object of interest (Ticao rock) is tricky to work with, it is better to select a region of interest (ROI) to 3D instead of the whole rock. We will see the effect of this later.



Step 1: Phase wrapping, subtracting like terms and solving the phase

Subtracting like terms we get

$$I_4(x,y) - I_2(x,y) = 2I_{mod}(x,y)sin(\phi(x,y))$$
$$I_1(x,y) - I_3(x,y) = 2I_{mod}(x,y)cos(\phi(x,y))$$

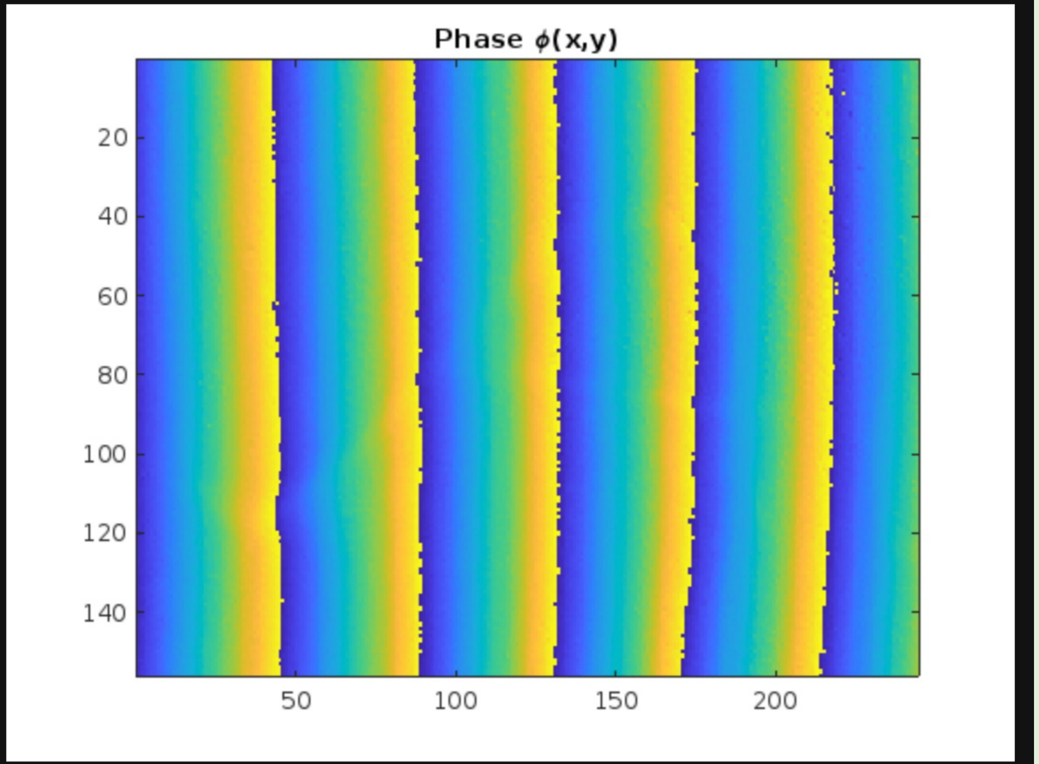
as imposed in line 22 and 23.

We can now solve the phase:

$$\phi(x,y) = \tan^{-1}\left(\frac{I_4(x,y) - I_2(x,y)}{I_1(x,y) - I_3(x,y)}\right).$$

as imposed in line 24.

```
21  %calculating the phase
22  Num = double(J4) - double(J2);
23  Den = double(J1) - double(J3);
24  PHI = atan2(Num, Den);
25  figure(3); imagesc(PHI); title('Phase \phi(x,y)');
```



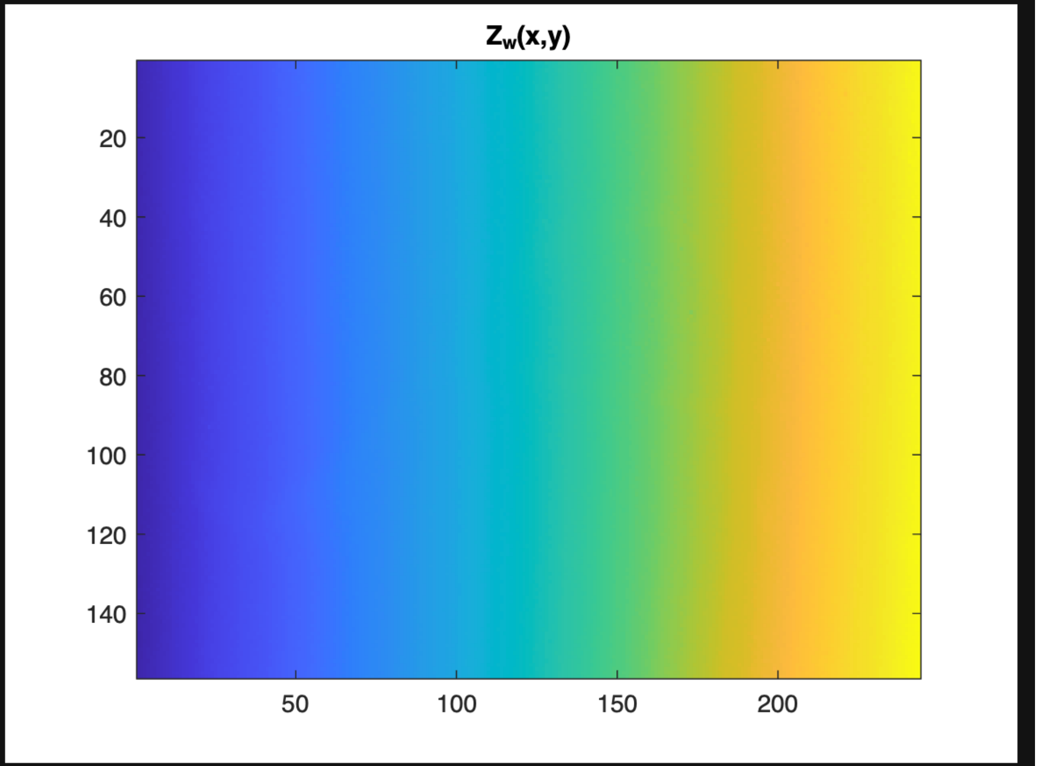
Step 2: Phase unwrapping

Phase unwrapping means adding integer amounts of 2π to pixels that have jumped from $-\pi$ to π. We will use the built-in unwrapping function in Matlab [4]. We have several phase unwrapping techniques such as quality-guided phase unwrapping. Matlab code for this is available here: [5]. *We will try this on the latter part.*

We still cannot see the 3D here but if we display the mesh of Z_w, it will look like it is riding on a ramp.

To get the 3D object, we need to remove this ramp. First, we need to estimate it (next step).

```
%unwrapping the phase
Z_w = unwrap(PHI,[],2);
figure(4); imagesc(Z_w); title('Z_{w}(x,y)');
```



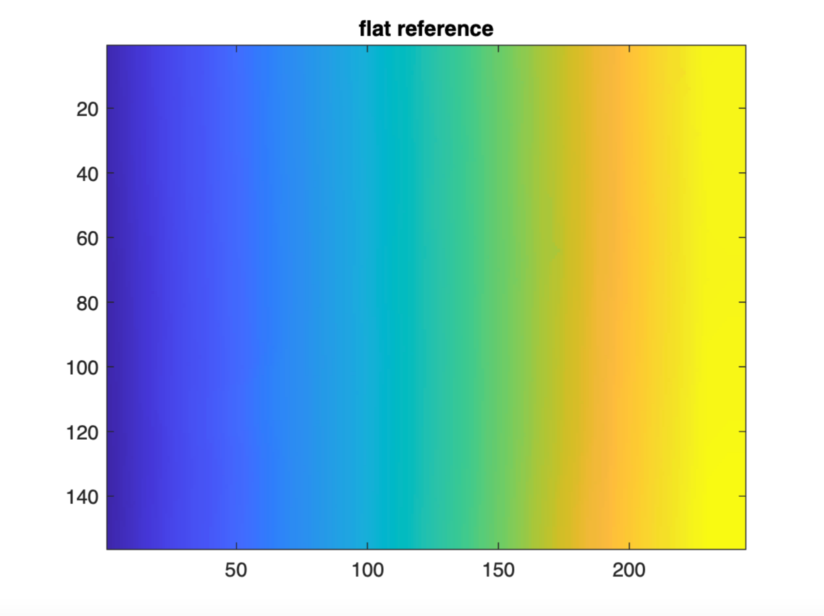
[4] Soriano, M. Physics 301 Activity Manual, Phase Shift Profilometry, 2022.
[5] Carey Smith (2022). QualityGuidedUnwrap2D_r1 (https://www.mathworks.com/matlabcentral/fileexchange/29499-qualityguidedunwrap2d_r1), MATLAB Central File Exchange. Retrieved June 19, 2022.

Step 3: Phase-to-height conversion, *estimating the reference and getting the phase difference*

Here, we remove the ramp by estimating the background. We can do this by creating a **structuring element**, a disk shape with radius 15, as implemented in Line 30. Then we can perform morphological opening as implemented in Line 31.

Experimentally, we can estimate the background by capturing an image (structured light) without the object. Unwrapping this background image will already give us the ramp.

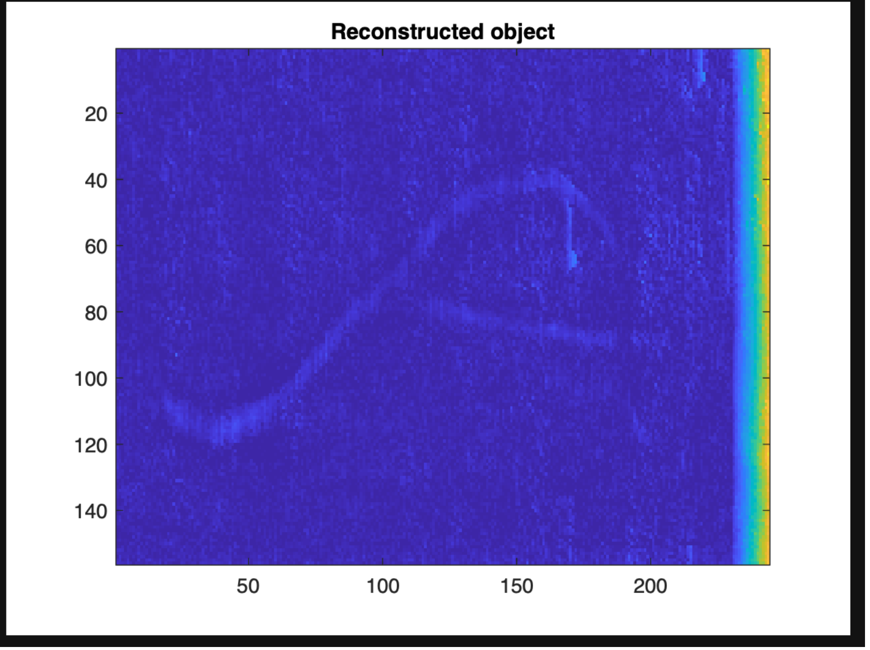
```
29  %estimating the reference
30  se = strel('disk',15);
31  background = imopen(Z_w,se);
32  figure(5); imagesc(background); title('flat reference');
```



Once we get the reference/background, we can reconstruct the object by just getting the phase difference of Z_w and the background, as implemented in line 34.

Finally, here is our reconstructed object. We can clearly see the Baybayin symbol carved on the Ticao stone!!

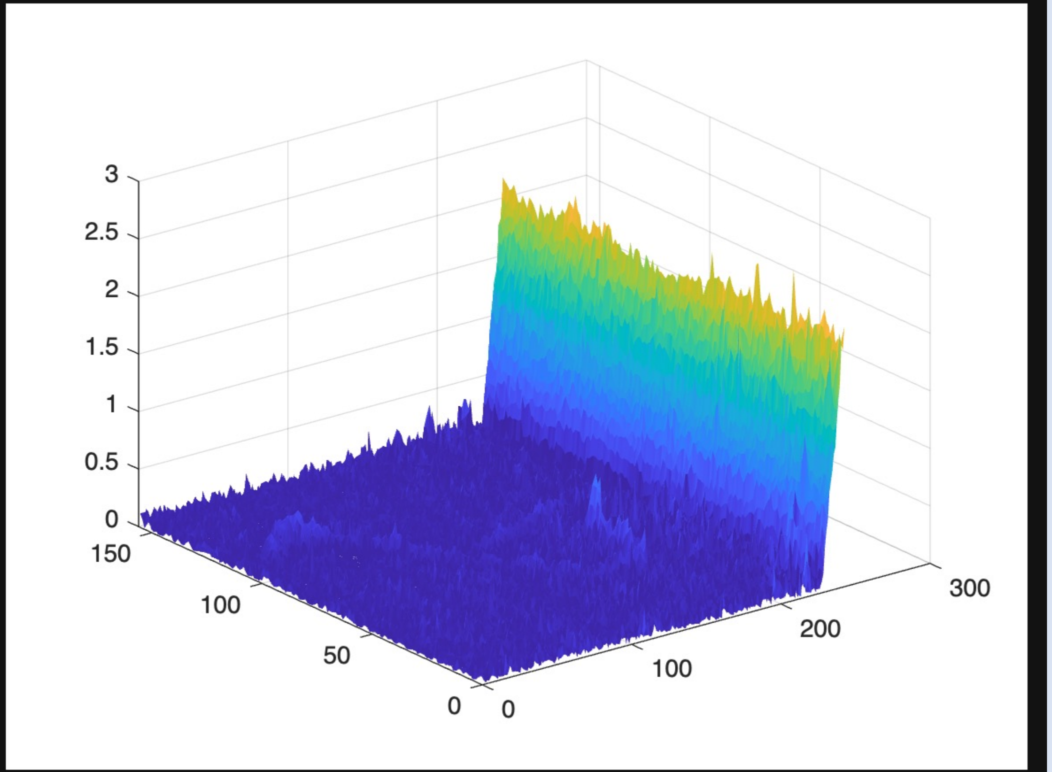
```
33  %getting phase difference
34  Obj = Z_w-background;
35  figure(6); imagesc(Obj); title('Reconstructed object');
```



Step 3: Phase-to-height conversion, 3D view

Here is the 3D version of the reconstructed object:

```
36 figure(7);
37 s = surf(Obj);
38 s.EdgeColor = 'none';
```

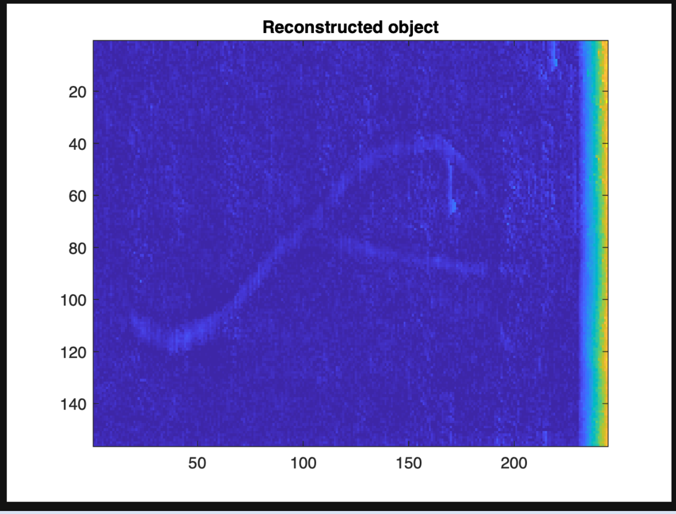


There is a big jump in height on the right side so let's crop that area:

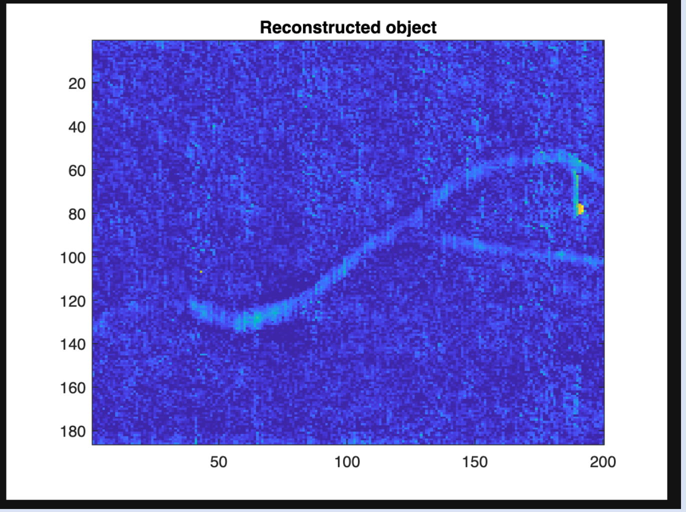
```
37 s = surf(Obj(:,1:200));
```

Let's do this too on line 35.

```
33 %getting phase difference
34 Obj = Z_w-background;
35 figure(6); imagesc(Obj); title('Reconstructed object');
```

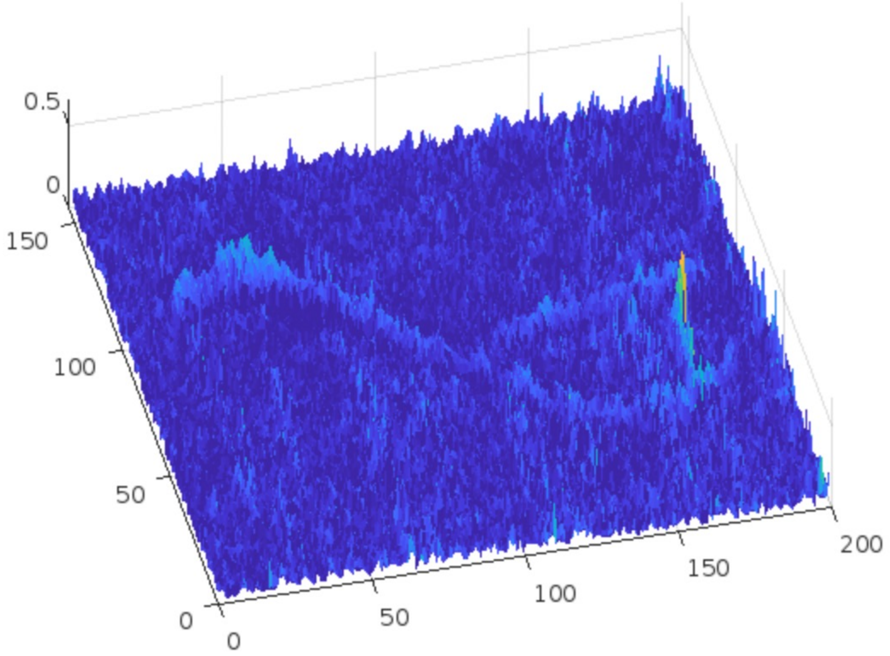


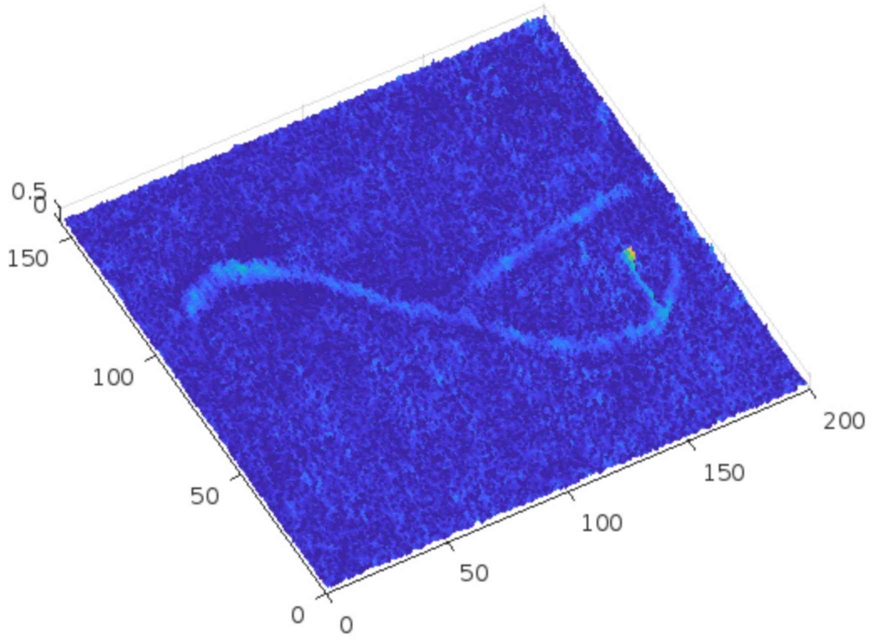

```
33 %getting phase difference
34 Obj = Z_w-background;
35 figure(6); imagesc(Obj(:,1:200)); title('Reconstructed object');
```



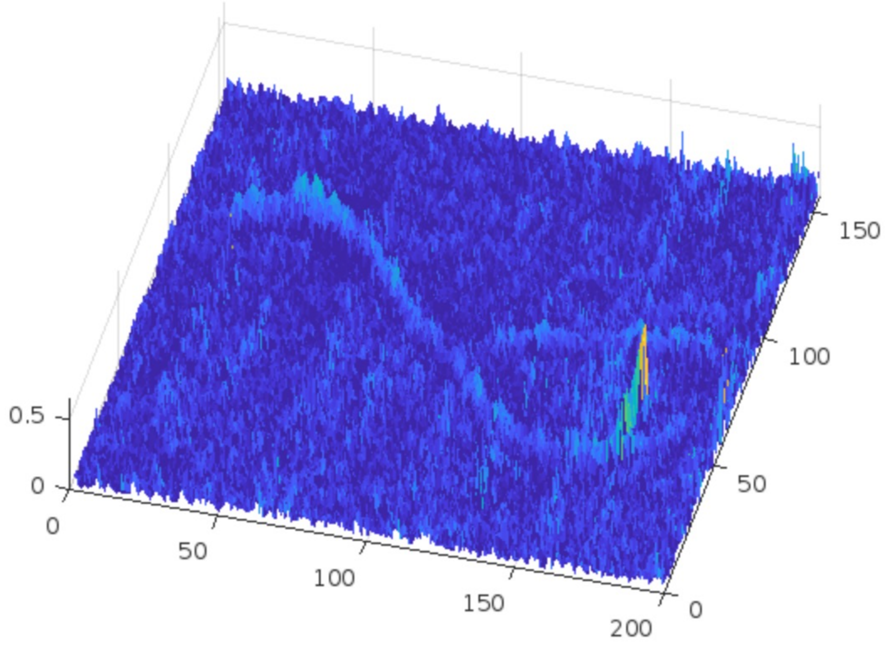
Cropping the big jump to focus on the Baybayin symbol

Step 3: Phase-to-height conversion, 3D view

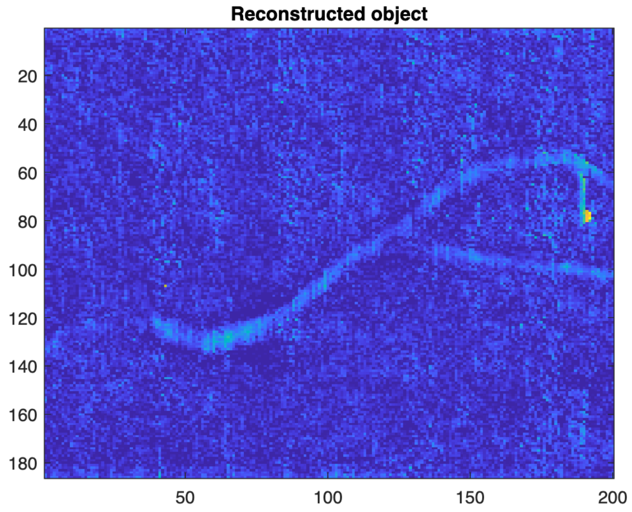




Here are different 3D perspective of the reconstructed object



Additional Step 4: Improving 3D reconstructed image using FFT

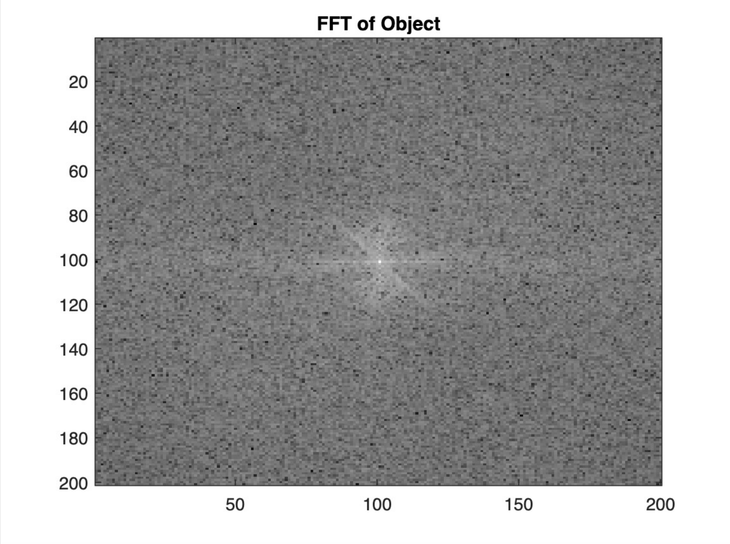


We can observe apparent horizontal lines from our reconstructed object. We can get rid of this by applying Fourier Transform (line 40):

```
39  %applying FFT to improve reconstructed image
40  Obj_fft = fftshift(fft2(Obj(:,1:200)));
41  size_fft = size(Obj_fft);
42  C=ones(size_fft);
43  for (y=round(size_fft(:,2)/2)+5 : size_fft(:,2)) %right mask
44      for (x= round(size_fft(:,1)/2)-5: round(size_fft(:,1)/2)+5)
45          C(x,y)=0;
46      end
47  end
48  for (y= 1:round(size_fft(:,2)/2)-5) %left mask
49      for (x= round(size_fft(:,1)/2)-5: round(size_fft(:,1)/2)+5)
50          C(x,y)=0;
51      end
52  end
53  Obj_fft_filtered = Obj_fft.*C;
```

Here is the FFT of our reconstructed object:

```
54  figure(8);
55  imagesc(log(abs(Obj_fft))); colormap(gray); title('FFT of Object');
```

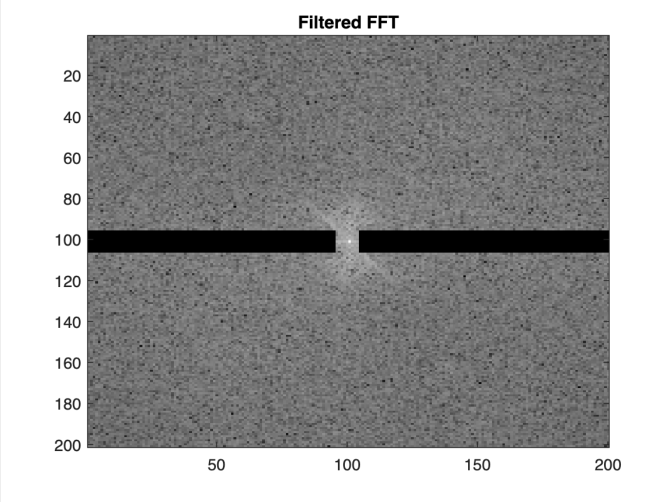


We filter the FFT by applying a horizontal mask on the fft.

The mask are created in lines 42-52.

We then multiply this mask to the FFT of our reconstructed image (line 53). Then we got this:

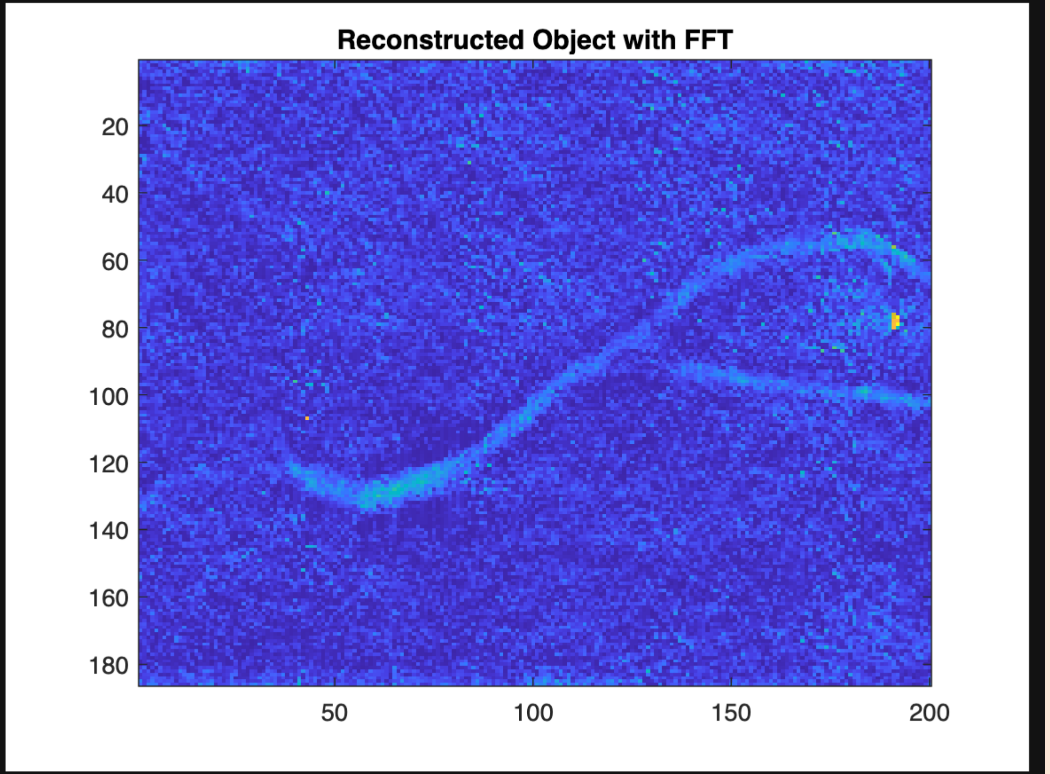
```
56  figure(9);
57  imagesc(log(abs(Obj_fft_filtered))); colormap(gray);  title('Filtered FFT');
```

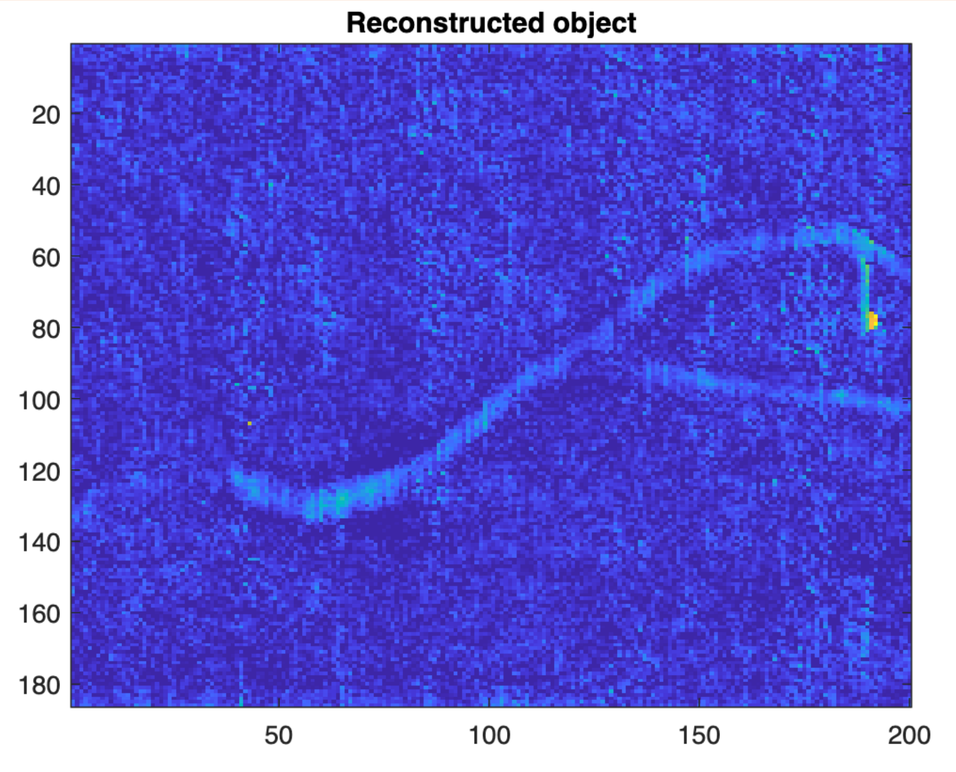


Additional Step 4: Improving 3D reconstructed image using FFT

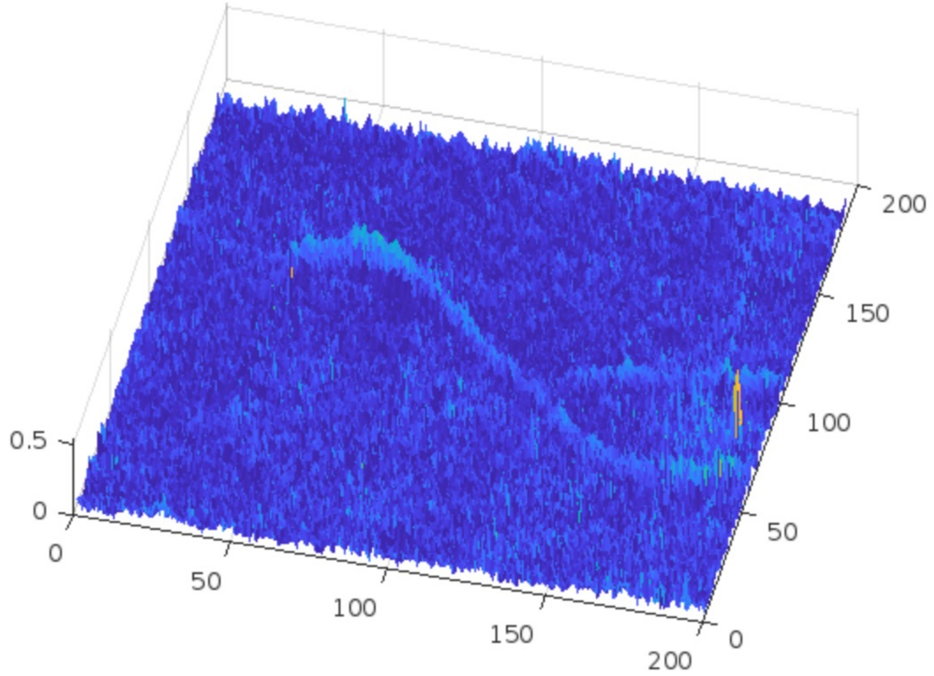
Lastly, applying inverse fft on to the filtered fft (line 58), we then obtain our reconstructed object (left). Compared to our original reconstructed object (right), we can see that we were able to eliminate the apparent horizontal lines, thus producing better reconstruction.

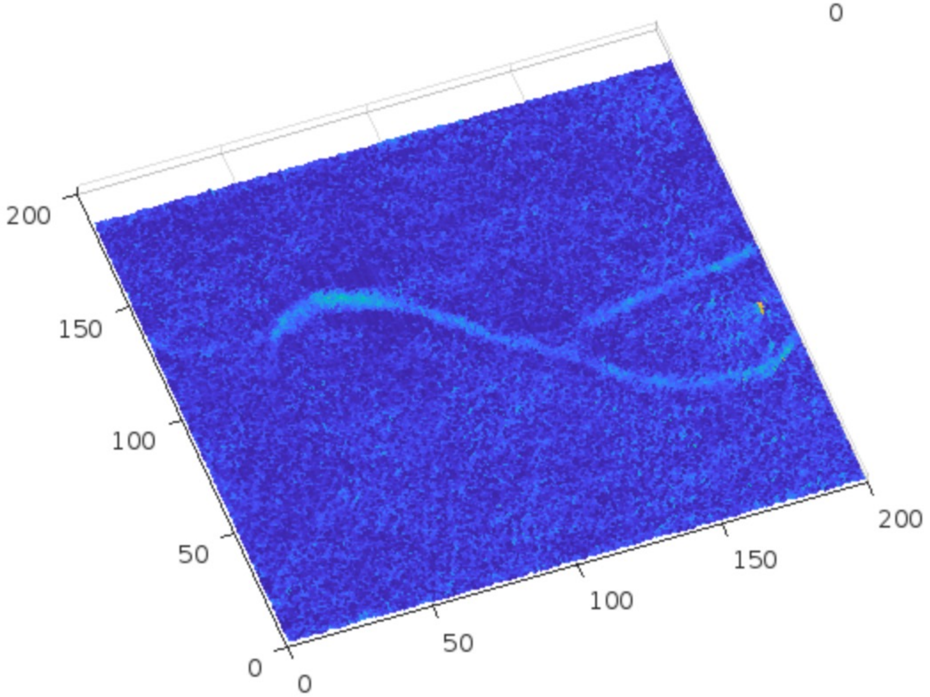
```
58  Obj_fft_filtered_inverse = ifft2(ifftshift(Obj_fft_filtered));
59  figure(10);
60  imagesc(double(abs(Obj_fft_filtered_inverse)));
61  title('Reconstructed Object with FFT');
```



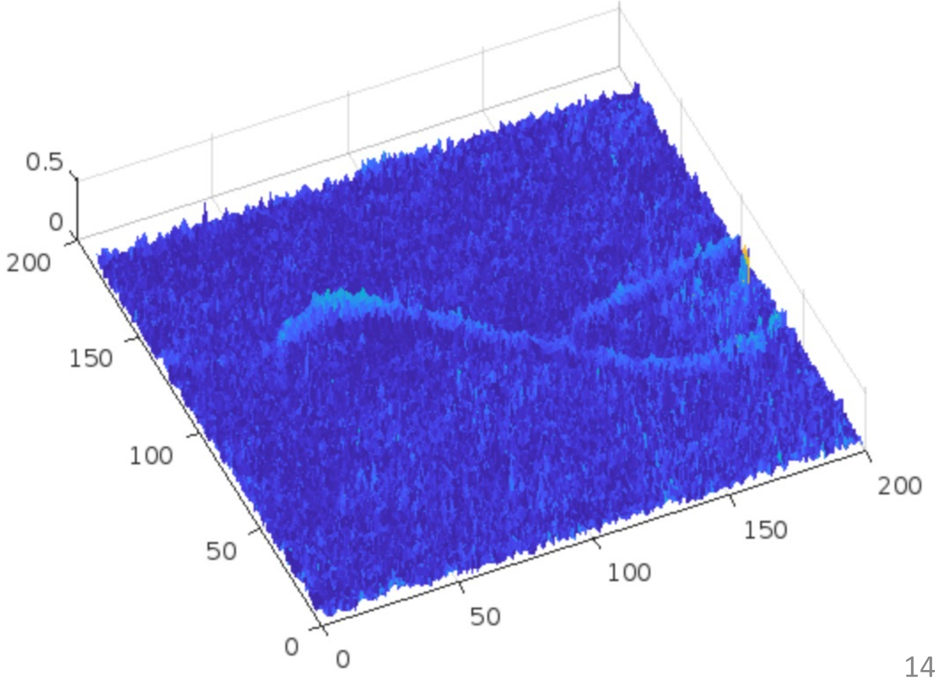


Additional Step 4: Improving 3D reconstructed image using FFT

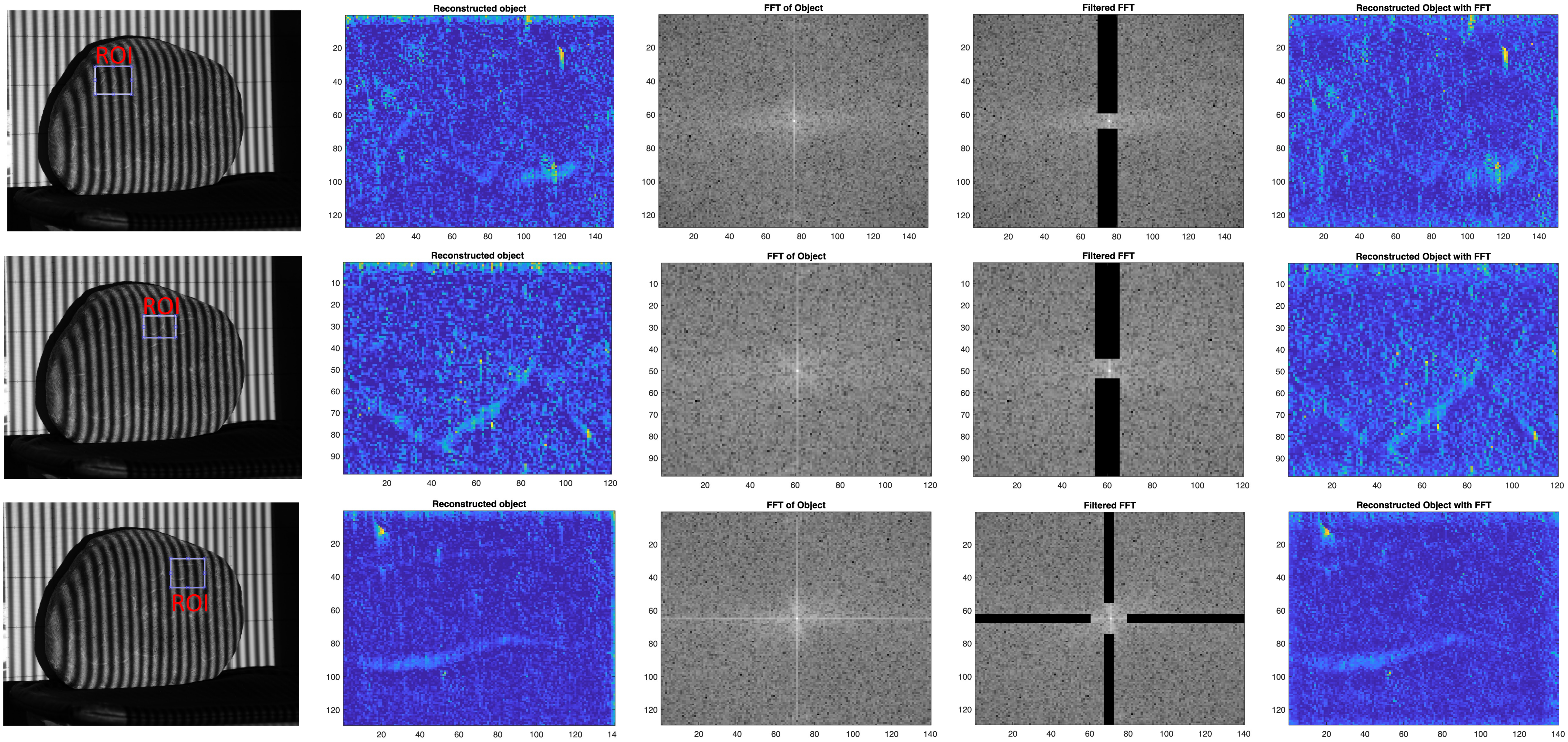




Here are different 3D perspective of the reconstructed object with FFT

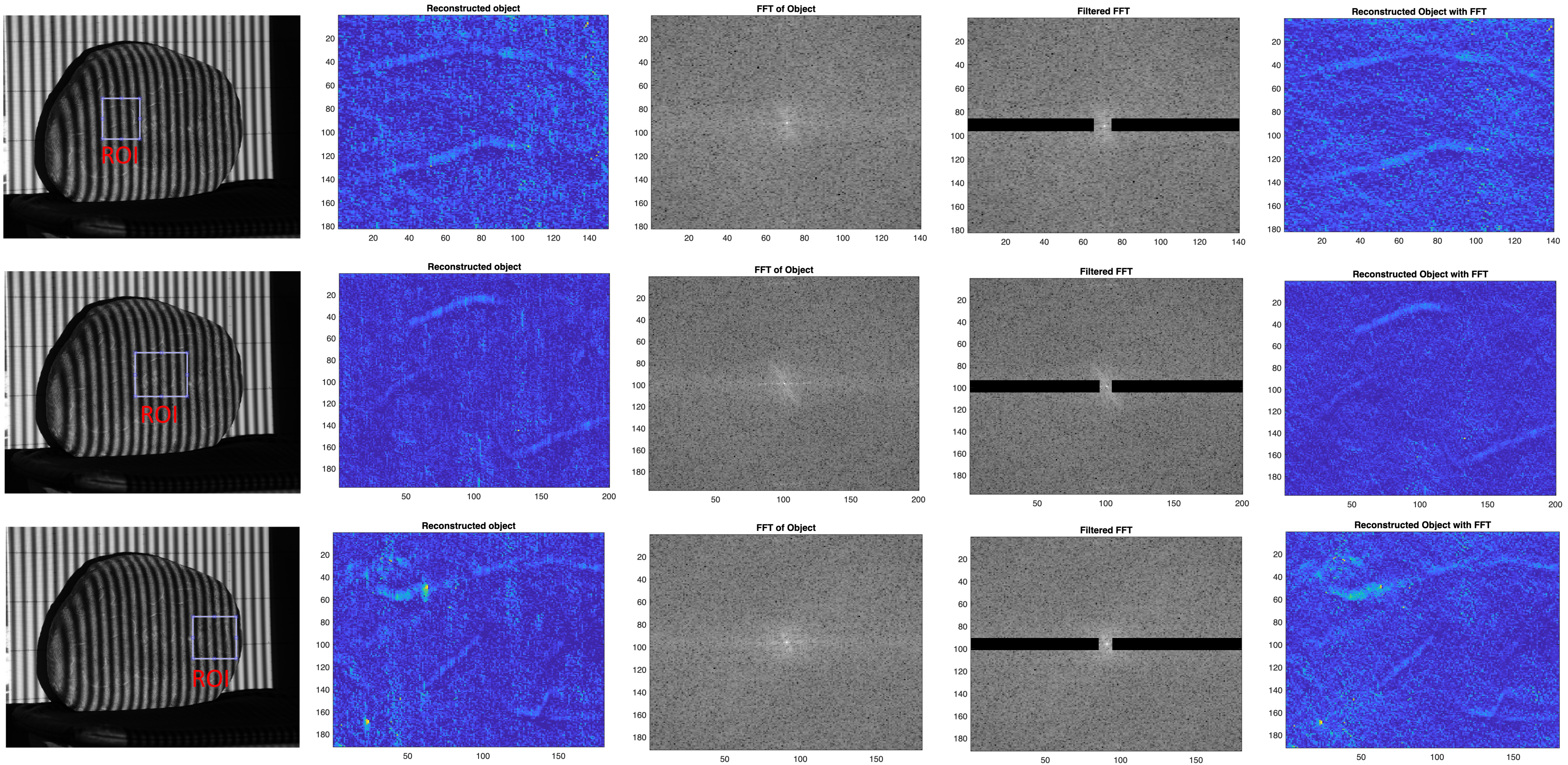


Here are other different ROIs (top row of the stone)



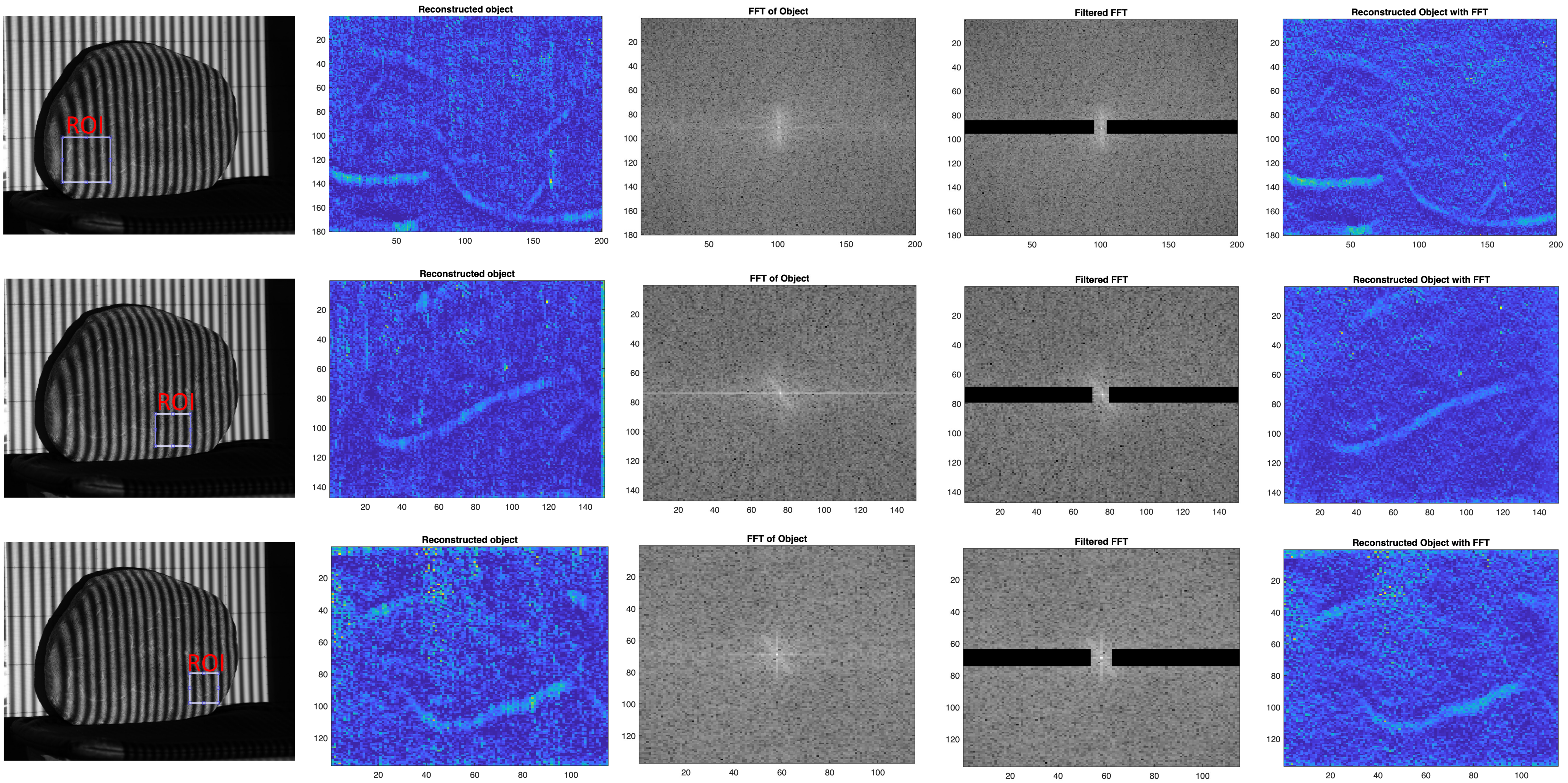
Smoother reconstructed object is apparent if we apply FFT. It is harder to 3D the top row of the stone. Perhaps one explanation is the non-flat surface/shape of the rock itself in this portion. There exists two distortions - from the shape of the stone and from the carvings.

Here are other different ROIs (middle row of the stone)



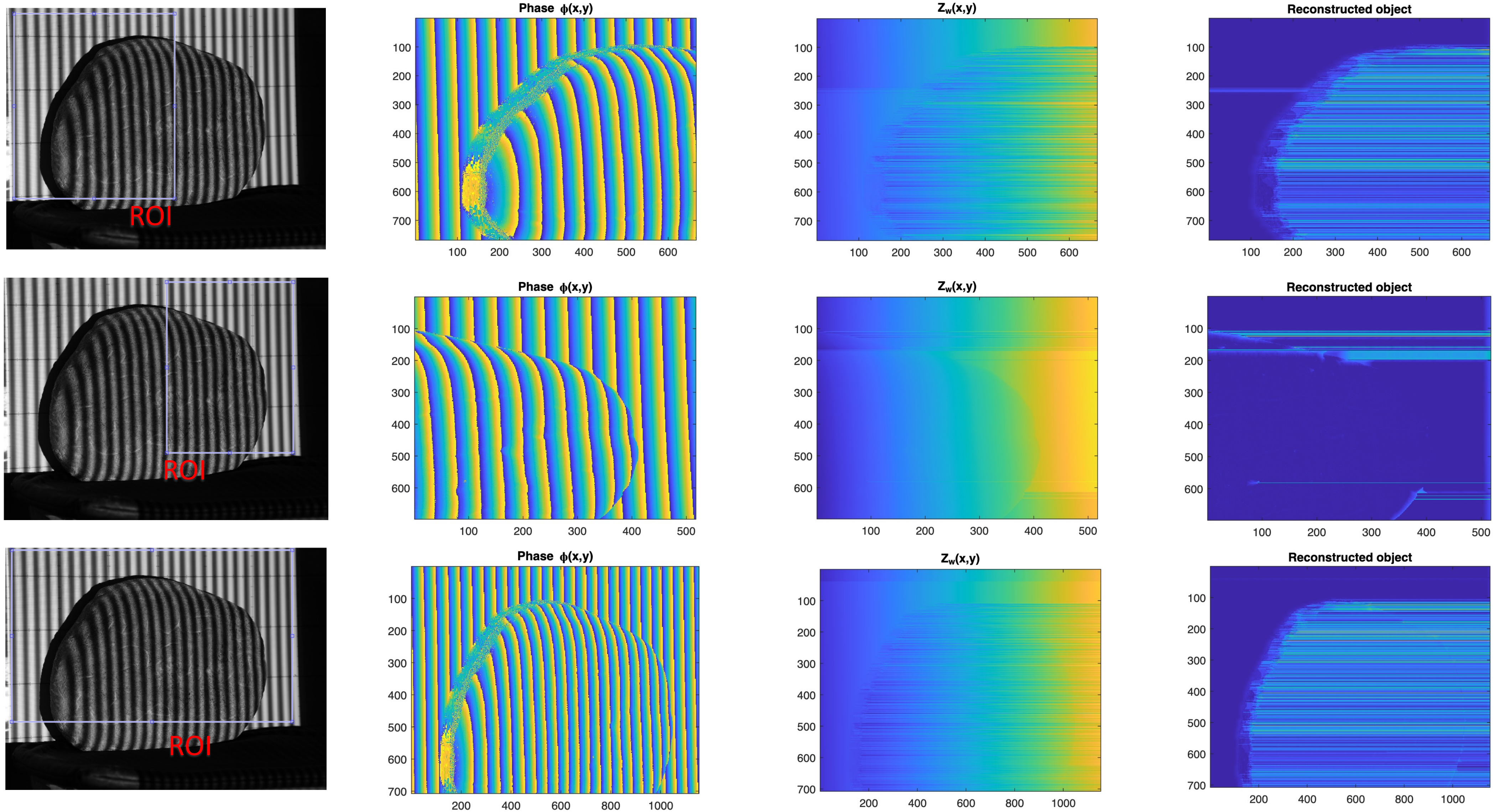
Similarly, smoother reconstructed object is apparent if we apply FFT. The carvings in the middle row of the rock is easier to 3D than the top row. The shape/surface of the rock is relatively flat here (no curvature on the shape itself). In this case, PSP can focus more on reconstructing the carvings.

Here are other different ROIs (bottom row of the stone)



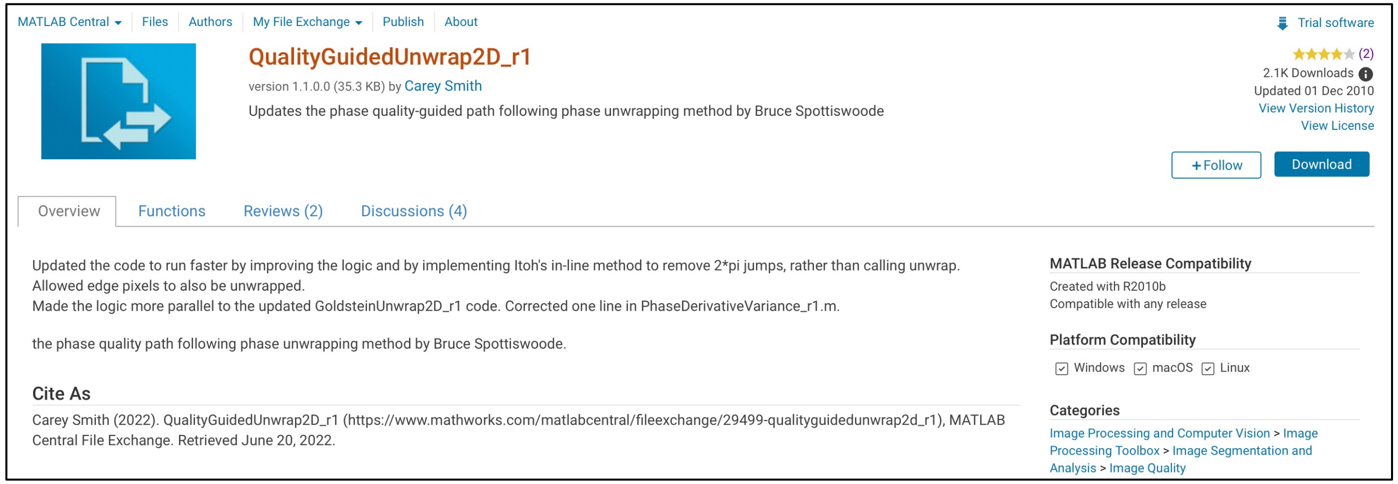
Same observation as in the middle row – smoother reconstructed object is apparent if we apply FFT, better reconstruction of the carvings.

Going back, we said before that it is better to focus on smaller regions instead of the whole rock. Below are the examples of 3D reconstructed objects with larger ROIs. The Ticao stone is a tricky object to 3D and so, it is not surprising that PSP fails to reconstruct even the shape of the rock. Also, we can see that while it has the potential to reconstruct the shape of the object (first row), it cannot reconstruct the carvings on the stone.



Using quality-guided phase unwrapping technique

This code is publicly available here: https://www.mathworks.com/matlabcentral/fileexchange/29499-qualityguidedunwrap2d_r1 See screenshot on the right. There are three sets of code here, two are functions to be used in unwrapping – **GuidedFloodFill2** and **PhaseDerivativeVariance**. The **QualityGuidedUnwrap2D** holds the main code for unwrapping which calls the two functions.



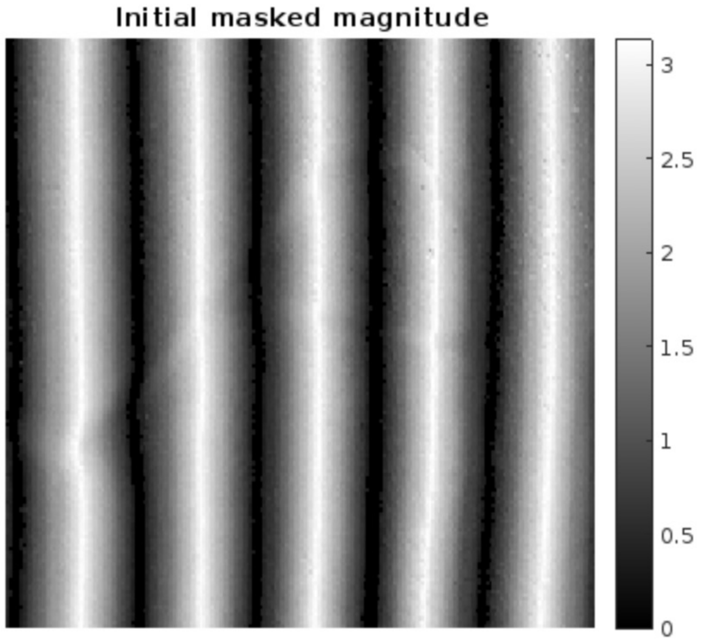
Here are the **things to note** (using Matlab online) when using this quality-guided phase unwrapping technique *(mistakes that I experienced so you don't have to):*

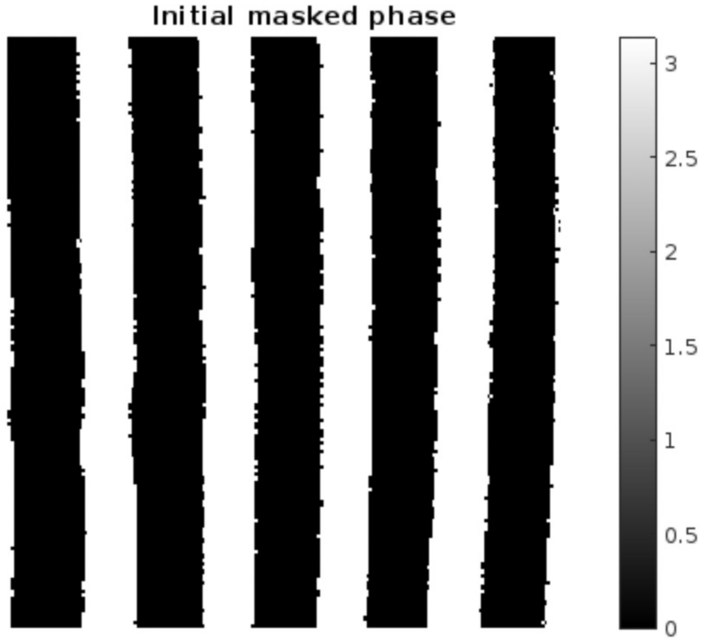
1. In Matlab, make sure that the two functions (**GuidedFloodFill2** and **PhaseDerivativeVariance**) are placed at the end of your script code, before it is being called.
2. The **QualityGuidedUnwrap2D** calls different functions **(PhaseDerivativeVariance_r1** and **GuidedFloodFill_r1)**. Find this portion of the code then change it to its appropriate name (**PhaseDerivativeVariance** and **GuidedFloodFill2**). You will not be able to run the code if you won't do this. You will get errors.
3. We insert this quality-guided phase unwrapping technique into our code starting after line 25.
4. The **QualityGuidedUnwrap2D** accepts IM.mat file. We don't need to create another IM.mat file. We can directly substitute of **PHI** in *line 24* of our original code as IM. Thus, we make **PHI = IM** removing the if-end statement in the first lines.

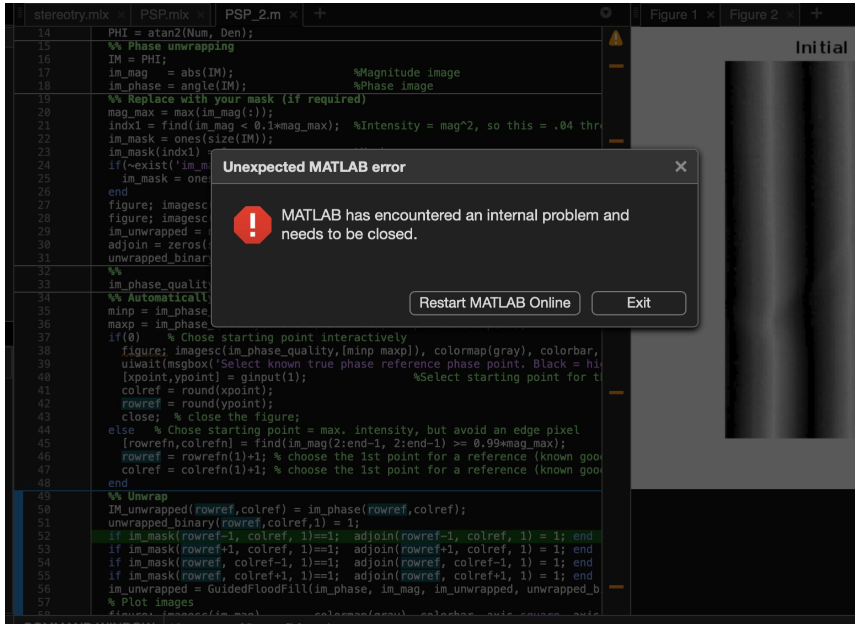
Using quality-guided phase unwrapping technique

Here are the preliminary result of the quality-guided phase unwrapping technique. I used the same ROI in our first example. The code was running smoothly until the PhaseDerivativeVariance function was called. Below are the results of the initial masked magnitude and phase. We can already see from the Baybayin symbol on the initial masked magnitude.

However, during the unwrapping portion, it took a very long time to run, around 1 hour, until I got this message on Matlab online (see 3rd pic). *MATLAB has encountered an internal problem and needs to be closed.*







I tried running this *many, many* times but to no luck, I keep getting the same message after a very long time of running. Unfortunately, the unwrapping failed. I am relatively confident, however, that this will work in the offline version of Matlab.

CRITERIA	QUALIFICATIONS	SCORE
Technical correctness	☑ Met all objectives. ☑ Results are complete. Results are verifiably correct. ☑ Understood the lesson.	30
Quality of presentation.	☑ All text and images are of good quality. ☑ A picture or diagram of the setup is shown (if the activity has one). ☑ Code has sufficient comments and guides. ☑ All plots are properly labeled and are visually understandable. ☑ The report is clear.	30
Reflection	☑ Explained the validity of results. ☑ Discussed what went right or wrong in the activity. ☑ Justified the self score. ☑ Acknowledged sources (e.g. persons consulted, references, etc.)	30
Ownership	❑ Applied the technique on other data (I have no other data since I don't have a projector at home) ☑ Discovered limitations in the technique (Selecting small ROIs is better than the whole rock itself) ☑ Introduced improvements to the technique (applying FFT for better reconstruction of the object) ❑ Deducting 3 points because although I attempted to apply the quality-guide phase unwrapping technique, Matlab online was not able to run it so I was not able to unwrap it.	7

97