Guide to Using the Scripts in the Paper

The steps and comments in the scripts follow closely the steps and terminology used in the paper. Please make sure to read the paper before reading this guide. We frequently make references to the figures and sections in the paper.

The software in this repository has been tested by using Python 3.11.5 on a Mac Book Pro. The following packages are needed (you can use pip for installation):

pip3 install pandas
pip3 install scipy
pip3 install matplotlib
pip3 install Pillow (for the PIL import)

The software is split into three scripts found in the repository folder *"paper_software"*

1. *paper_signal_from_ArchSpiral.py*
2. *paper_fft_from_signal.py*
3. *paper_signal_from_flattened_image.py*

Run the scrips in a console window, with a command line prompt as:

python3 *paper_signal_from_ArchSpiral.py*

For undistorted Archimedes spirals, the scripts *paper_signal_from_ArchSpiral.py* and *paper_fft_from_signal.py* are used.

For distorted spirals or situations where editing in an external image editing software is required, all three scripts are required.

The repository folder *"paper_software"* also contains three spiral images S2, S3, and S6 which have been used as examples in the paper, and all the generated artifact files (such as log files, csv files, etc.) that are produced when the scripts are run.

Interactive matplotlib plots and user clicks

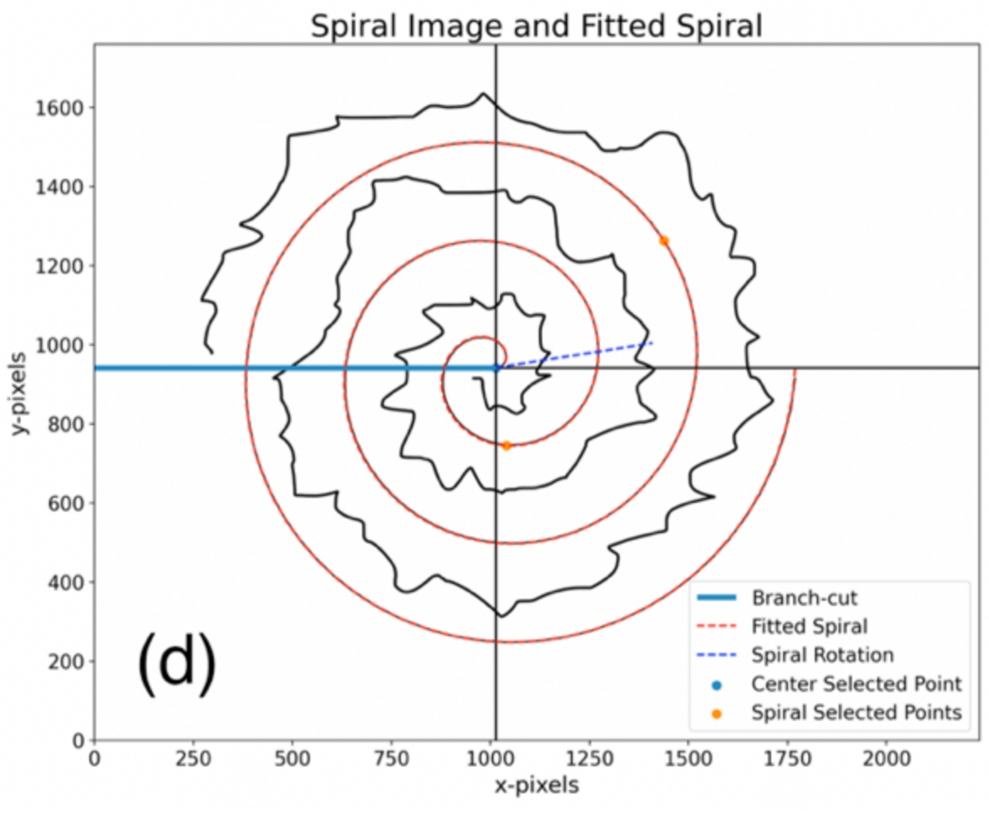
The scripts use matplotlib plots in interactive mode so that the user can click on the plots to record pixel (x, y) values. Depending on the step in the script, either a single click (to record one pixel, such as the center of the spiral) or two clicks (to record two pixels, such as a rectangular region in a plot) are required. The scripts **only use the last (one or two) clicked pixel values** and ignore all other clicks that the user might have accidentally made earlier. This ensures that if a user accidentally double-clicks, or makes a wrong click selection, the script does not need to be restarted; the user just makes sure that the last one or two clicks are valid.

After the user has made the desired pixel selection, **the matplotlib window should be closed**. The script then opens the next window and waits for the user's clicks. The scripts use the console window to display messages to prompt the user to perform the next action, such as "click on the center of the spiral" etc.

The scripts follow numbered steps which have the same numbering used in the "Results" section of the paper. There is a one-to-one correspondence between the paper and the scripts. The scripts are commented and follow the same terminology used in the paper.

Advice on selecting pixels in a plot

Here are a few hints on selecting the pixels in a plot. You can use the magnifying glass symbol on the matplotlib interactive window to select a smaller region to make the selection more precise. This is especially important when selecting the center of the spiral. Also, expand the window to full size to increase the precision of your selections. When selecting the two points on the spiral, expand the window and make sure to **select the two pixels in zones 1 and 2**, like the two selected pixels (orange dots) in the example below taken from Fig. 2 (d). The scripts expect the two points to be in zones 1 and 2.



Now we show how the logic in the section “Results” in the paper is implemented in the scripts by modifying the “Results” section of the paper for this user guide.

Results (modified)

Perfect Archimedes Spirals

Obtaining the Discrete Signal from the Spiral Image

This logic is implemented in the script *paper_signal_from_ArchSpiral.py*.

Step 1-1: Read the Spiral Image File
(STEP 1 in the script) Pixels from the image file which contains the spiral as well as the patient’s hand-drawn spiral are read. A file that contains a log of important events is also defined.

Step 1-2: Determine the Equation of the Spiral
(STEP 2 in the script) First, the user clicks on the interactive plot to determine the origin of the spiral, and then the user clicks twice to select two points on the spiral. Section “Spiral equation from two points” is followed to determine the parameters θ_r and b. Our program assumes that the first user-selected point lies in zone-one while the other belongs to zone-two. These parameters give us the equation of the fitted spiral (4). This fitted spiral is used next to flatten the Spiral Image.

Step 1-3: Flatten the Spiral Image
(STEP 3 in the script) The section “Zones and Branch Cut in the Flattened Space” is followed, and the equation for the spiral is used to map the Spiral Image to Flattened Space. Fig. 2b shows the family of stitched spiral segments, as well as the family of stitched patient’s hand-drawn segments for an example spiral. It takes approximately 40 seconds on a 2000 Mac Book Pro laptop to complete this step in Python v3.11.5. The user is prompted to select the endpoint of the longest stitched hand-drawn spiral.

Step 1-4: Crop to get a Single Stitched Patient’s Hand-Drawn Spiral
(STEP 4 in the script) The user clicks twice on the interactive plot to select a cropping rectangle to extract the part that only contains the pixels of a single stitched patient’s hand drawing, Fig. 2c.

Step 1-5: Derive and Save the Discrete Signal
(STEP 5 in the script) The patient’s hand-drawn spiral image from Steps 1-4 is multiple pixels wide with pixels having different greyscale (or color) values; it is not yet in the form of a discrete signal on which mathematical manipulations could be performed, such as determining its DFT (Discrete Fourier Transform). A further complication is introduced by a scanned image having light grey (light color) pixels spread throughout the background pixels, which should ideally be white. The approach we take to extract a single-valued signal is to select each x-pixel (in the Python program) and search in the vertical direction to select the y-pixel having the

highest greyscale (or color) value; we assign this y-pixel to the value of the signal corresponding to that x-pixel. **This technique works even better if the contrast of the original image is enhanced before processing.** Now, we have a one-to-one correspondence between the x-values and the y-values, which constitutes a signal upon which mathematical manipulations can be performed. At this point, we have a signal with a large number of samples (pixels). For the spiral of Fig 2, we end up with a number of x-pixels = 3228. We also need to correct the signal based on (10). This signal can potentially have some noise spikes in a very small region near the origin; in our program, the user is prompted to crop out this noise region. Finally, we resample the signal to N = 1000 samples to reduce the number of parameters. The program keeps track of how many pixels are being cropped in various steps throughout the procedure in a variable named cropping_ratio. The final cropped discrete signal with N = 1000 samples is then converted to a (Python) Panda's data frame and saved as a csv (comma-separated value) file for the next step. The variable **cropping_ratio is saved in the log file**.

Obtaining the DFT (FFT) from the Signal

This logic is implemented in the script *paper_signal_from_ArchSpiral.py.*

Step 2-1: Read the Discrete Signal from the csv File
(STEP1 in the script) The cropped discrete signal with $N = 1000$ samples generated in Step 1 above is read from the csv file and normalized. A file that contains the log of important events is created. The user also initializes the variable named *cropping_ratio* which was calculated in the steps above and saved in the log file. If time-dependent information has been recorded for the hand-drawn spirals then the user also initializes the variables, *time_to_draw*, and *estimated_tremor_freq.* If timing information has been recorded then we set the variable *time_to_draw_measured_flag* to *True*, otherwise to *False*.

Step 2-2: Determine the FFT of the Discrete Signal
(STEP2 in the script) To determine the DFT, the popular and efficient algorithm, FFT, is employed. The FFT of a real-valued function is a complex number having a magnitude and a phase. For N sample points in the discrete signal, the number of coefficients, N_{coeff}, in the magnitude and phase FFT equals $N_{coeff} = \frac{N}{2} + 1$. A low pass filter is applied with a cutoff of $0.4 \times N_{coeff}$ to eliminate high-frequency noise outside the band relevant to tremors. A Savitzky-Golay filter (window size:51, poly order:3) is applied for the plots to better visualize the discrete spectrum; it is only applied for plotting and does not alter the data for subsequent calculations.

Step 2-3: Determine the Inverse FFT of the Truncated FFT
(STEP3 in the script) The FFT is truncated to a relatively low number of coefficients, N_{trunc}. Then, we take the inverse FFT of the *truncated* FFT to get the *approximate discrete signal*. The RMS error between the two signals is calculated.

Step 2-4: Save the coefficients of the truncated FFT to Pandas data frames
(STEP4 in the script) The coefficients of the truncated FFT are save to a Pandas data frame.

Distorted Archimedes Spiral

For distorted spirals, or situations where handling in an external image editing software is required, all three scripts are required in this order:

1. *paper_signal_from_ArchSpiral.py*
2. *paper_signal_from_flattened_image.py*
3. *paper_fft_from_signal.py*

We start by using *paper_signal_from_ArchSpiral.py.* All the steps in the above two sections remain unchanged, we simply insert a new step between steps 1-3 and 1-4.

Results of Steps 1-1 through 1-3 of the above Sections
[STEP 1 – STEP 3 in the script] The steps are identical to the steps outlined in the section above. We begin with the script *paper_signal_from_ArchSpiral.py* and input a distorted spiral image. After selecting the center and two points on this distorted spiral, we obtain an equation for the Archimedes spiral to fit the distorted spiral. The fitted Archimedes spiral does not completely follow the distorted spiral. Flattening it by using the equation of the fitted Archimedes spiral gives us a family of flattened spiral images (an example is shown in Fig. 4c). Since the fitted spiral does not exactly follow the distorted spiral, all the stitched spirals have a low amplitude, slowly varying wave superimposed on them. For example, the distorted spiral does not map into a family of horizontal straight lines.

Step between 1-3 and 1-4 Manually Clean the Flattened Image.
[STEP 3 in the script] The superimposed waves make it difficult to select a rectangle to cleanly crop out a single stitched patient's spiral since the rectangle would possibly include pixels from other parts of the image. At this point, we copy (via. a screenshot) the image with all the stitched spirals and use Photoshop to edit the image and erase the unwanted pixels in the cropped region. **It is important to keep the left edge of the crop flesh with x=0** so that our program can calculate the $N_{original-pixels}$ accurately. An example is shown in Fig. 4d. We save the cropped file image and force quit the script, *paper_signal_from_ArchSpiral.py.*

Now we switch over to the script, *paper_signal_from_flattened_image.py*. This script is mainly the last part (STEPS 3 and 4) of the script, *paper_signal_from_ArchSpiral.py.*

[STEP 1 of the script] The user inputs the cropped image file obtained from the image editing program.

[STEP 4 and STEP 5 in the script] These steps are identical to STEPS 4 and 5 of the script, *paper_signal_from_ArchSpiral.p* described in the section above.

Now we use the script *paper_fft_from_signal.py* and follow all the steps in the section "Obtaining the DFT (FFT) from the Signal".