shoecomp
USER GUIDE

csafe
Center for Statistics and Applications in Forensic Evidence

csafe tools

Last Modified: August 5, 2024

Introduction

This document provides a user guide for ShoeComp, a software tool for the markup and alignment of shoeprint images. ShoeComp is developed by CSAFE, the Center for Statistics and Applications in Forensic Evidence at Iowa State University. It is part of the CSAFE Tools collection, and aims to provide a graphical user interface for annotating, aligning, and comparing shoeprint images. ShoeComp aims to provide the following features:

- mark interest points on footwear impression images
- align footwear impressions using marked interest points, removing differences in rotation, translation, and scale between the images – by difference in scale, we refer to a missing L-scale in the image or if it is unclear how the physical size of the shoeprint translates to a length in pixels.
- compare aligned images to produce similarity scores – we currently provide an example similarity score to showcase how scores for such comparisons might be presented.
- view similarity scores in context of reference distributions – the provided reference distribution is based on an experiment run on data from [**?**], and are not representative.
- save interest point markups and alignments for reports or future comparisons with other metrics

ShoeComp implements a generalized version of the maximum clique alignment algorithms described in [**?**, **?**] to align the shoeprint images. ShoeComp is written in the `Java` programming language as a plugin for `ImageJ/FIJI`, a popular open-source image processing and annotation tool. It also uses ImageJ's `Action Bar` plugin to provide a simple user interface. The source code of ShoeComp is publicly available at:

`https://github.com/CSAFE-ISU/shoecomp`

where you can also download the latest built version from the `Releases` page. Below is a screenshot of the basic ShoeComp user interface:

Figure 1.1: The basic Graphical User Interface (GUI) of ShoeComp. There are six options: `Load Image` to load a new shoeprint image or resume previous markup work, `Save Markup` to save markup work for export or later use, `Run Alignment` to align two shoeprint images, `About` displays information about ShoeComp and CSAFE, `Debug` toggles the classic ImageJ GUI for advanced usage or debugging, and `Exit` closes the application.

ShoeComp is designed in a manner similar to that of `FRSTAT`, which is a tool for the statistical interpretation of friction ridge skin impression evidence. However, we note that ShoeComp at present does not provide any statistical analysis or interpretation capabilities, the scores and reference distributions are illustrative examples for how such things could be done.

User Guide

Let's go through the basic steps to mark and align shoeprint images using ShoeComp. As an example for this guide, we will be using images from the FBI Boots Dataset [**?**], available publicly at

`https://doi.org/10.17605/OSF.IO/EV8MK`.

We will be using the following two images:

- `QK010-QC.JPG` as our questioned image, and
- `QK010-KF.JPG` as our reference image.

You can download the images from `https://doi.org/10.17605/OSF.IO/EV8MK` (available in 'JPEG Images Part 1 of 4'), but if this is your first time trying ShoeComp, we recommend downloading our example markups from `https://github.com/CSAFE-ISU/releases`, the same link where this manual is available.

2.1 Downloading ShoeComp

You can download ShoeComp as a ZIP file `https://github.com/CSAFE-ISU/releases`. Select the ZIP file according to your operating system. For this guide we will be using a Windows computer.

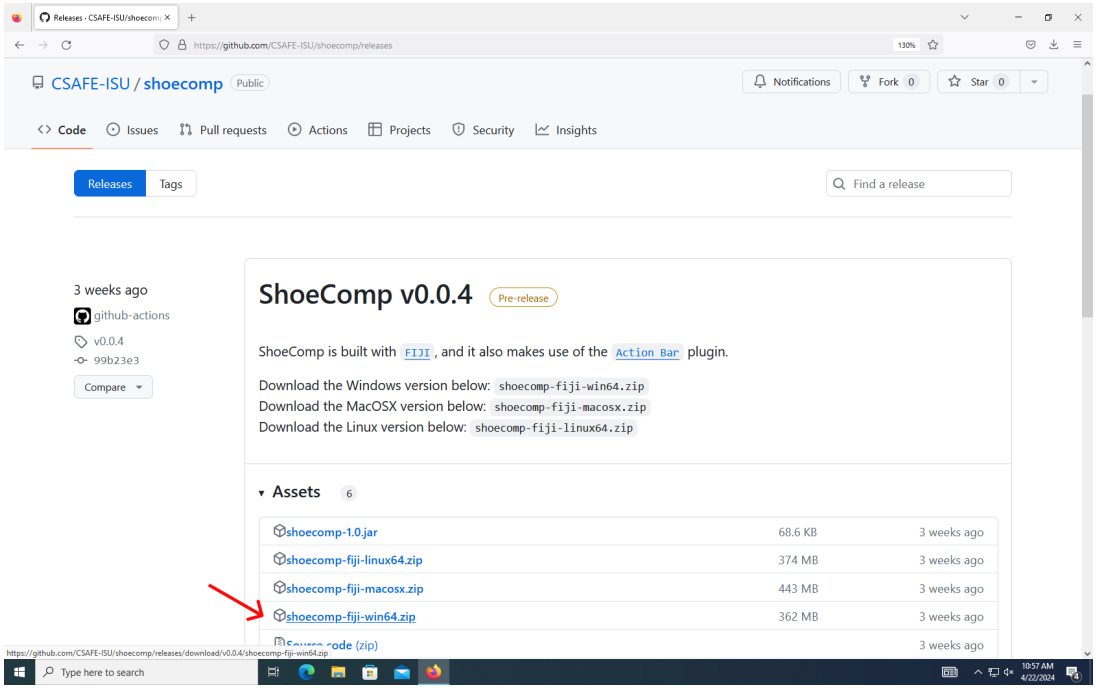

Figure 2.1: Download ShoeComp as a ZIP File from the CSAFE-ISU Github page.

2.2 Installing ShoeComp

ShoeComp does not require any installation on Windows or Linux, you can just unzip the ZIP file and it is ready to use:

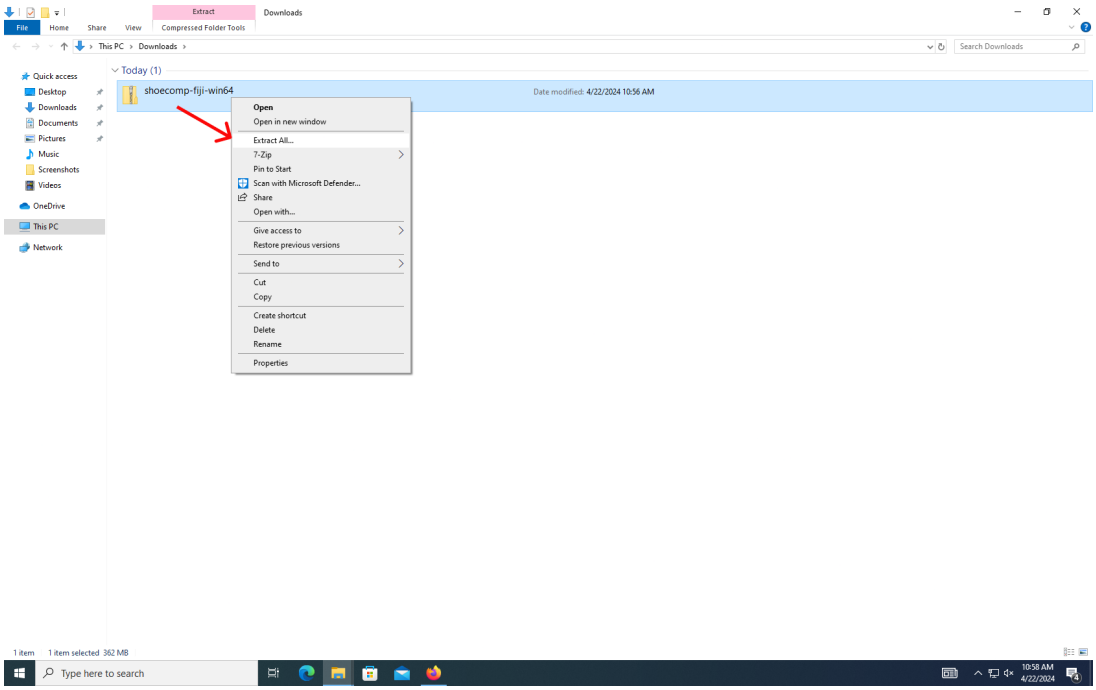

Figure 2.2: Installing ShoeComp. Just unzip the ZIP file and you're ready!

On MacOS, you may not need to unzip, but you will need to access Settings to allow the `FIJI` application to run. After unzipping, you should find the `Fiji.app` folder and find the `ImageJ` application to double-click:

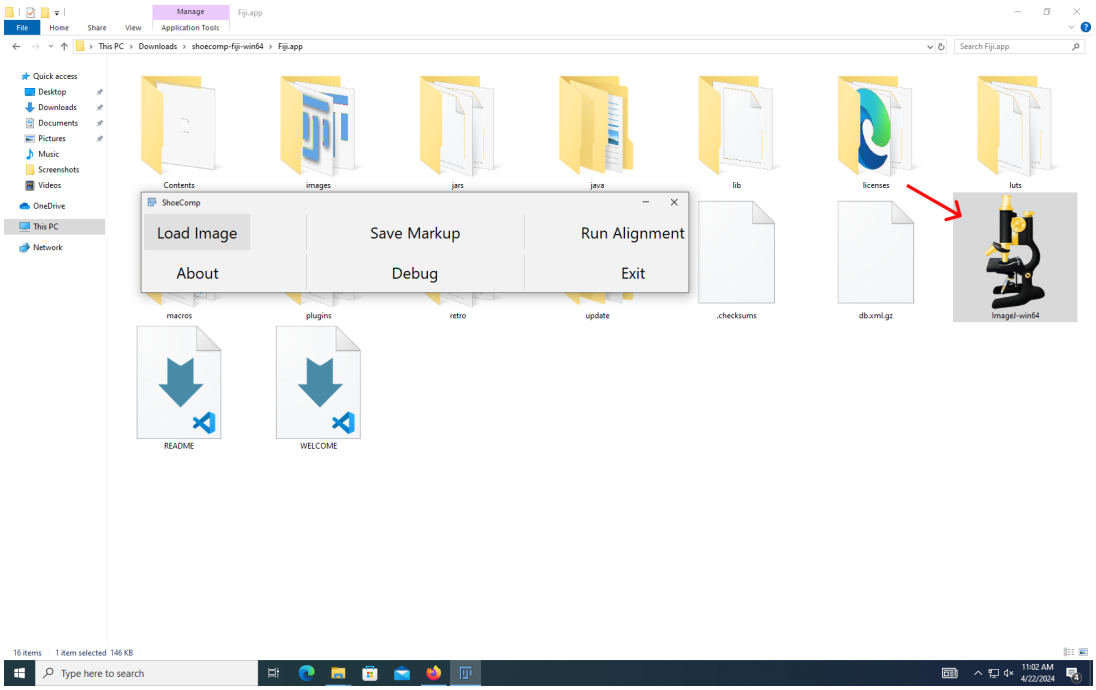

Figure 2.3: ShoeComp is built as a plugin for FIJI/ImageJ. Double-click on the ImageJ application to start ShoeComp.

2.3 Loading an Image

After opening the ShoeComp GUI, we need to load images. Click on the `Load Image` button, select `Load Image` again, and then select the image file to read:

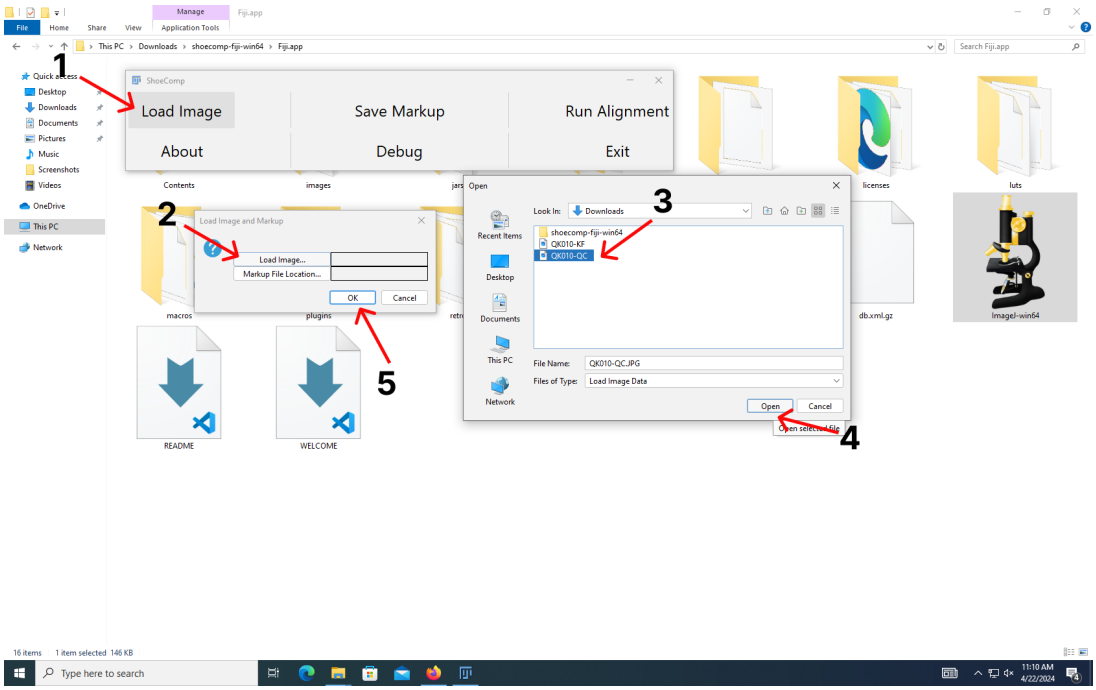

Figure 2.4: Loading an image into ShoeComp: Click on `Load Image`, Click on `Load Image` again, Select the image file, Click on `Open`, Click on `OK`.

After loading the image, you will first need to mark the relevant area of the shoeprint. A window will appear next to the image, which you can close after marking a polygon bounding the relevant area of the shoeprint.

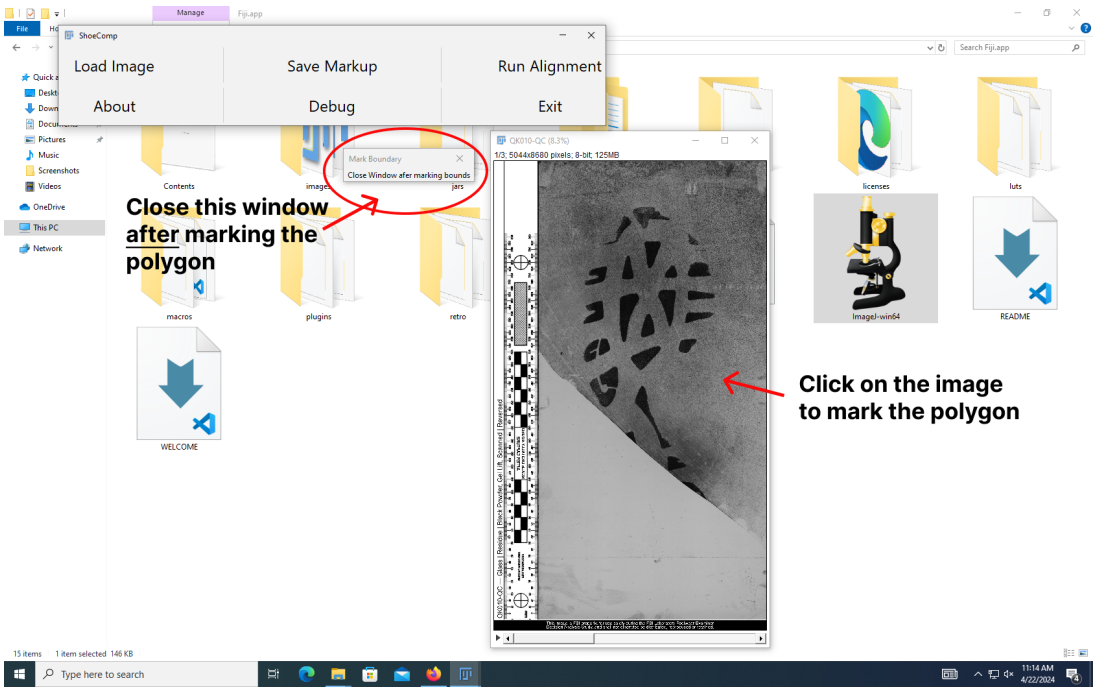

Figure 2.5: Start marking the boundary polygon. Click on the image to mark the boundary. Don't close the window before completing the polygon!

Once you've marked the boundary polygon, you can close that window so you can focus on marking interest points in the relevant area of the shoeprint.

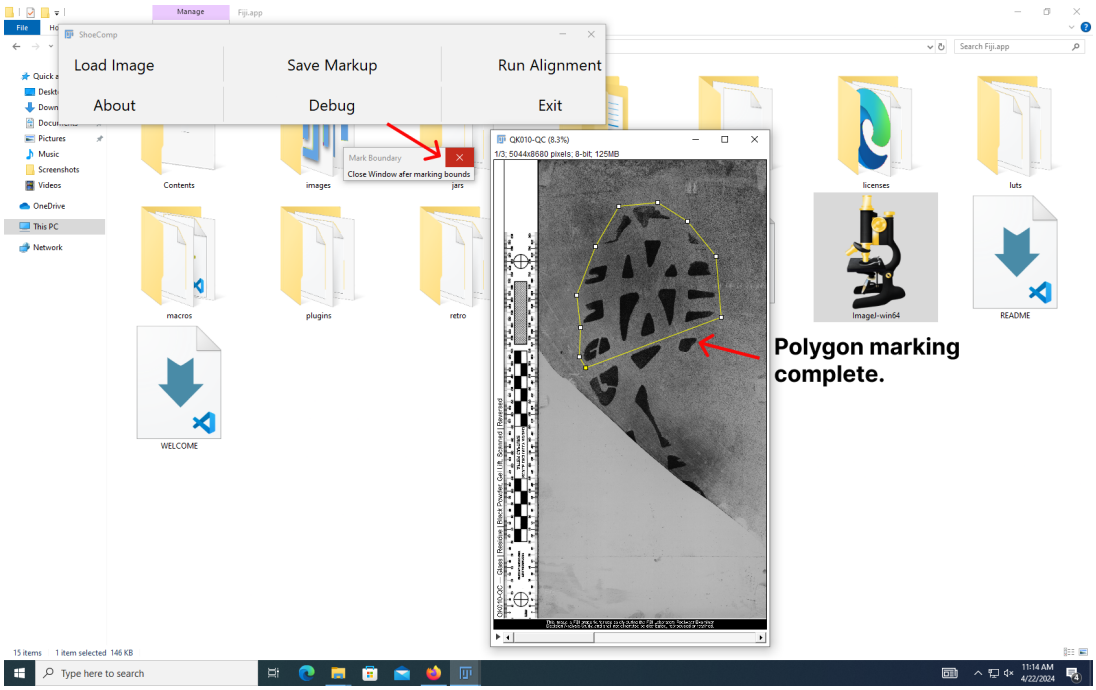

Figure 2.6: The boundary polygon has been marked, so now we can close this window.

Mark interest points by clicking on the image. We usually use corners of geometric shapes, but you can use any points that can be marked consistently (like centers of circles). If you want to mark the points with a bit more accuracy, you can use `Ctrl+Scroll` or the + key to zoom the image at the location of the mouse. If you've marked a point wrongly or have too many points marked, use `Alt+Click` to remove a marked point.

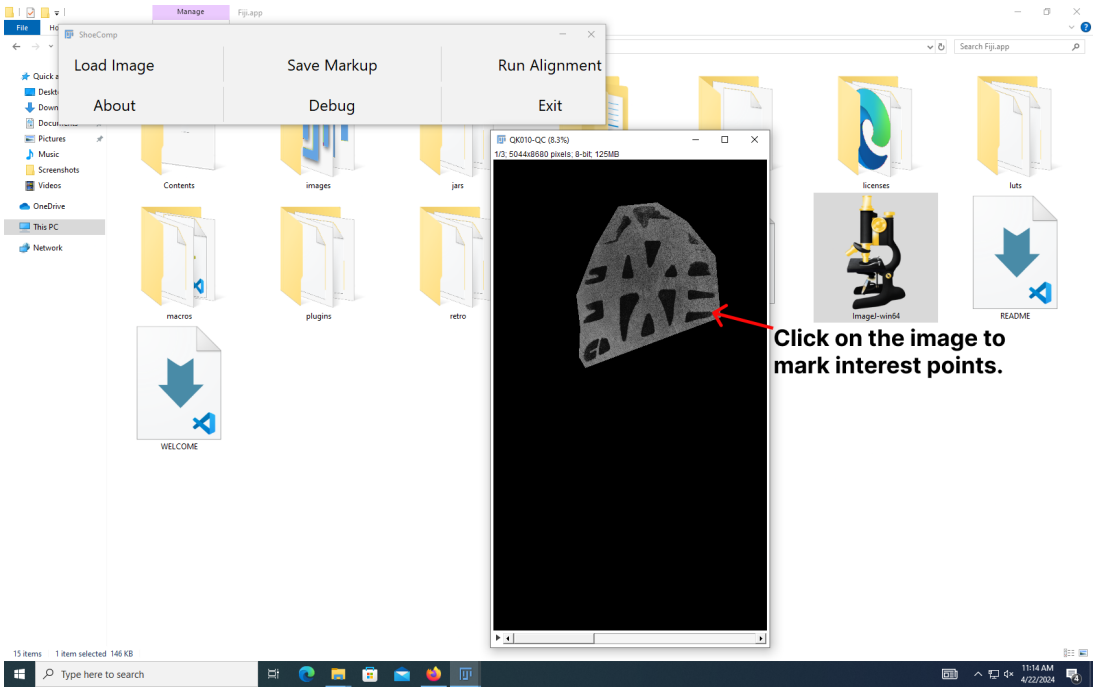

Figure 2.7: Mark interest points by clicking. Use `Ctrl+Scroll` or the + key to zoom the image at the location of the mouse, and `Alt+Click` to remove a marked point. If you just scroll by mistake, you will see the polygon mask and original image, so scroll back to the markup.

Around 20-30 marked points on the image should be enough for trying out an alignment Theoreti-

cally, we should only need 3 points for solving rotation, translation, and scale, but there might be noise in the image, non-linear distortions, and marking error, so 20-30 is a rule of thumb for alignment.

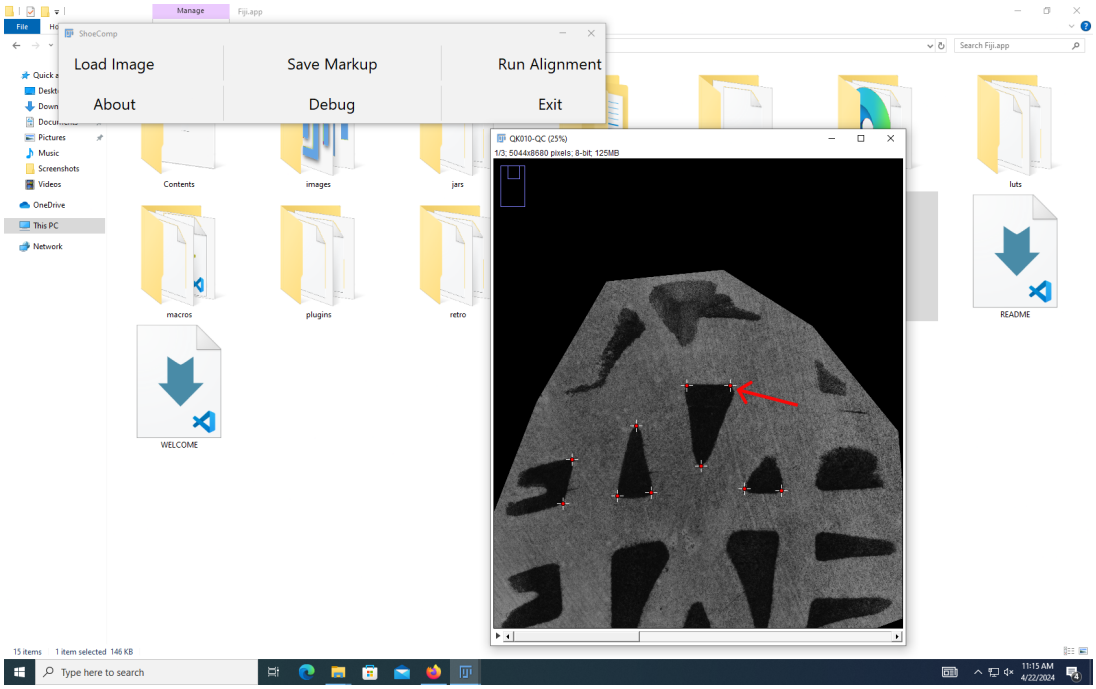

Figure 2.8: A few interest points have been marked on the image.

2.4 Saving/Reloading Markup

After you've marked the relevant area as polygon and some interest points, you might want to save your markup to export to other tools and resume at a later point. Click on `Save Markup`, and select file locations for saving the markup. The markup is saved as a text file in the JSON format.

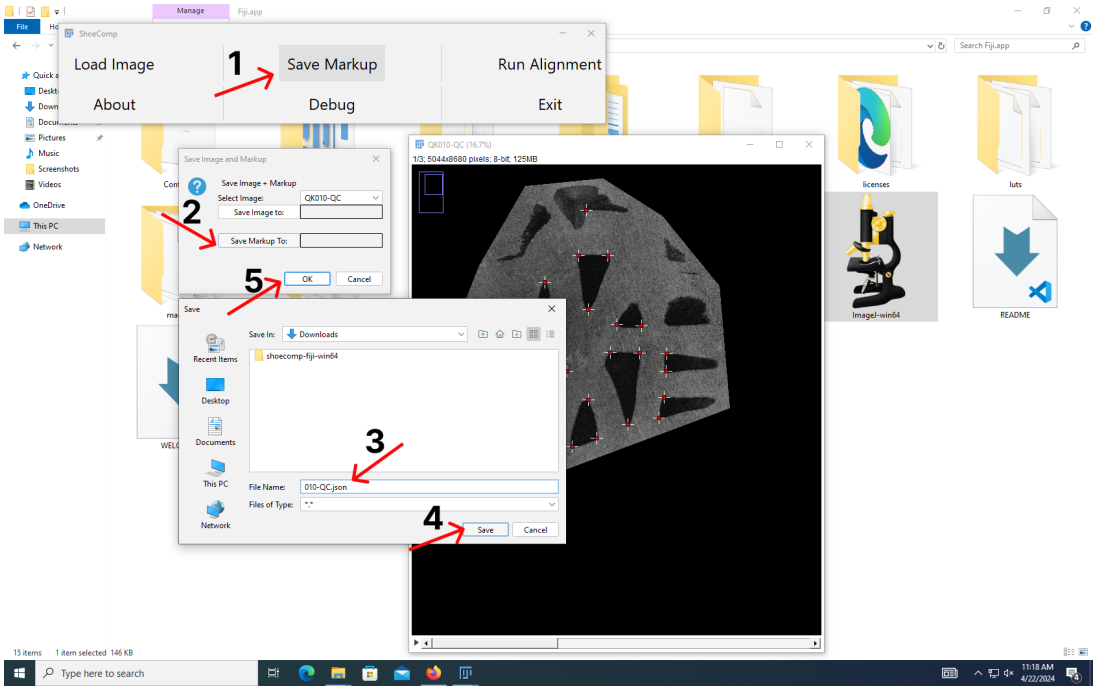

Figure 2.9: Saving your markup as JSON. Click on `Save Markup`, then `Save Markup To:`, then write a filename to save, click `Save`, and then click `OK`. You can also save the cropped image if necessary.

Once the markup has been saved, you can close the application or just the window. When reloading the image, you can use the saved JSON markup later to resume your work:

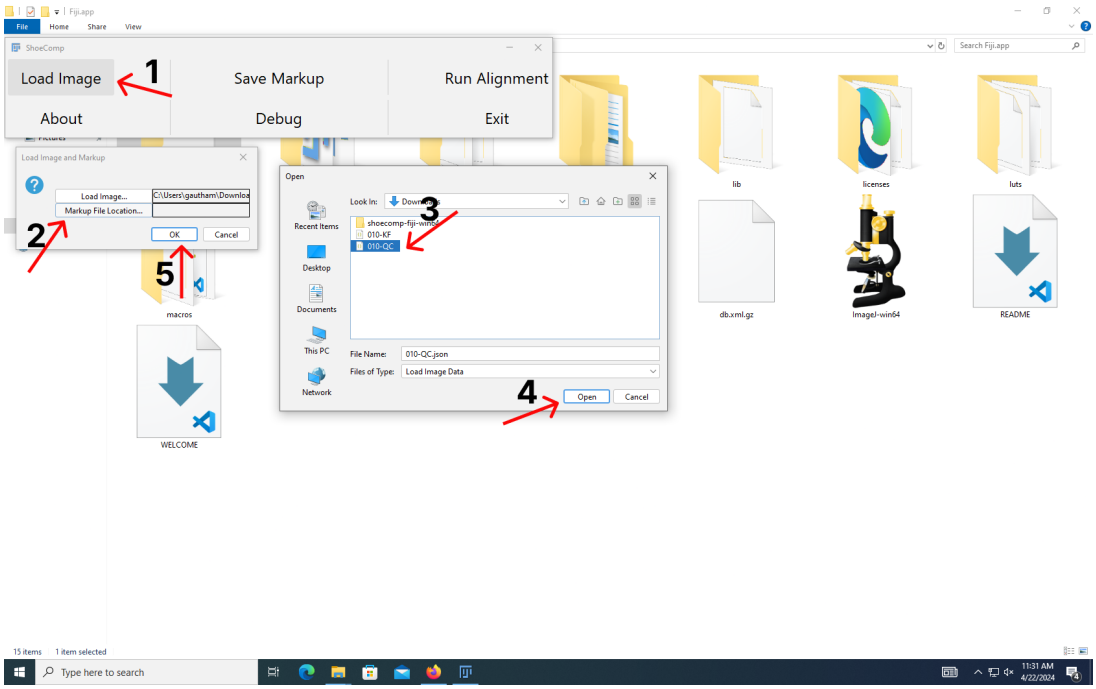

Figure 2.10: When re-loading the image similar to **??**, you can now click on `Markup File Location` and load the JSON file to resume with your previously loaded markup.

2.5 Aligning Marked Images

Now, let's suppose we have marked up the two images `QK010-QC.JPG` and `QK010-KF.JPG` as below. To obtain an alignment, we hope that the marked points are reasonably accurate, and that a decent number of points (at least 3 points) are common between both images.

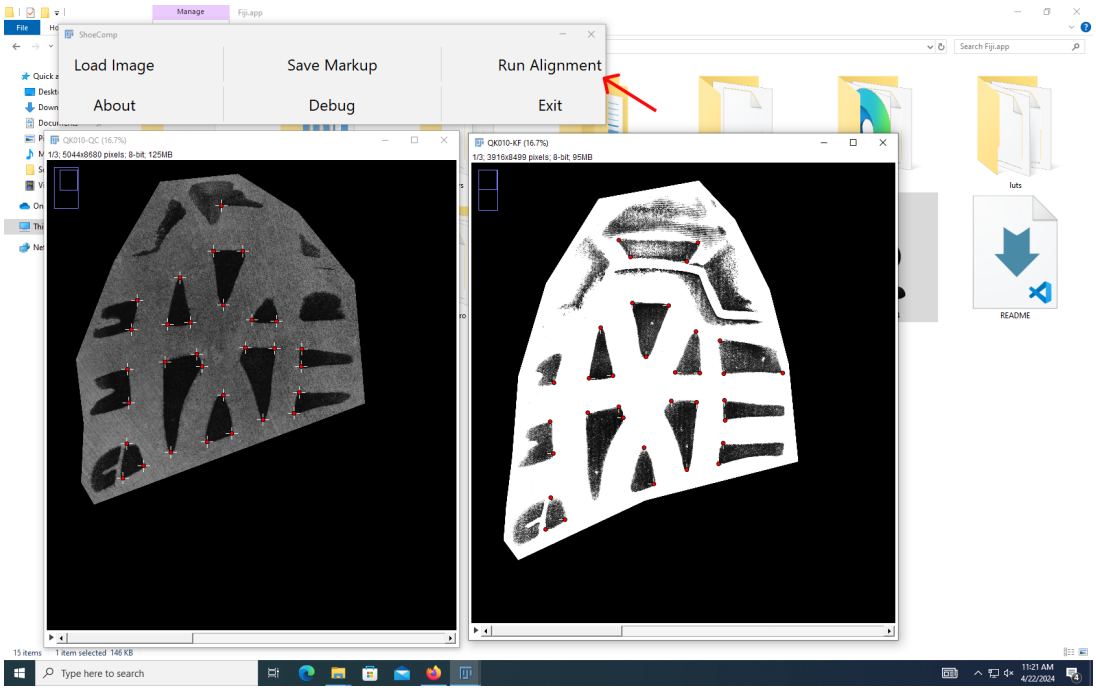

Figure 2.11: We have marked around 20-30 points in both images, and now we are ready to try the alignment algorithm.

We would like to align these shoeprint images, obtain a visualization of how well these images can be aligned, and also obtain some numerical scores that tell us how good the alignment is. We can do that by clicking on the `Run Alignment` button:

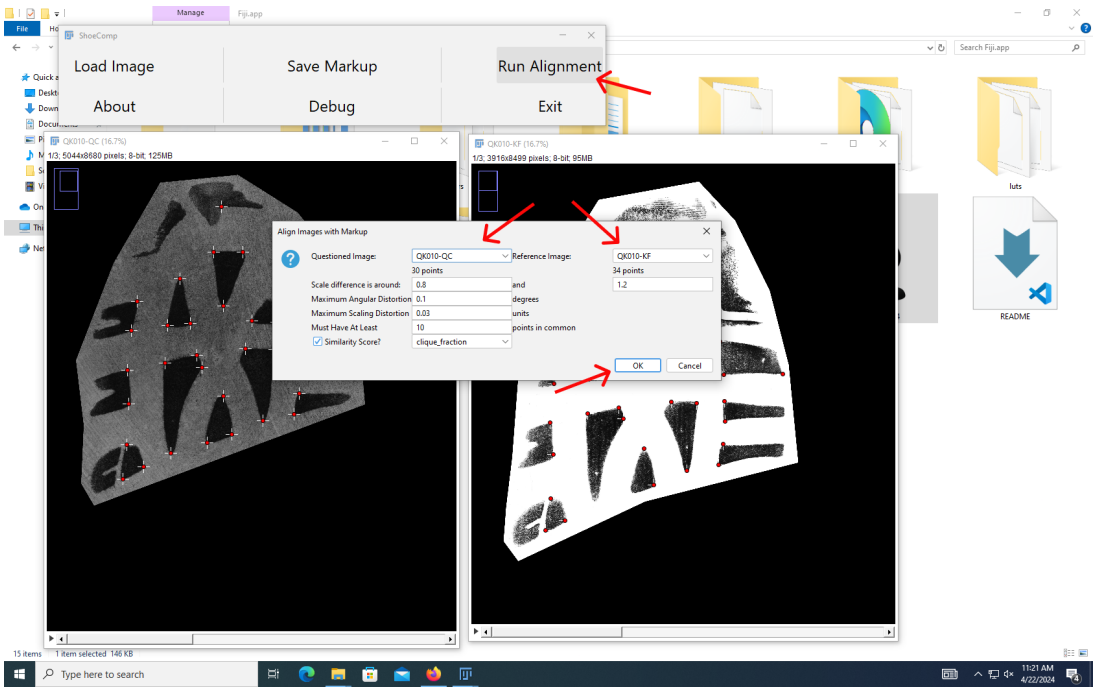

Figure 2.12: Align shoeprint images. Click on `Run Alignment`, select `QK010-QC` as the questioned image, `QK010-KF` as the reference image, fill an approximate range for scale difference between these images (it should be around 1 in this case). For now, let's use the defaults for the maximum allowed distortion, but generally smaller values indicate more rigid transformations. We would like the alignment to align at least 10 points in common from the points we marked. Finally, we select one of the similarity scores to view.

The alignment might take some time to run, depending on the number of points marked and how large the allowed ranges for scaling and distortion are.

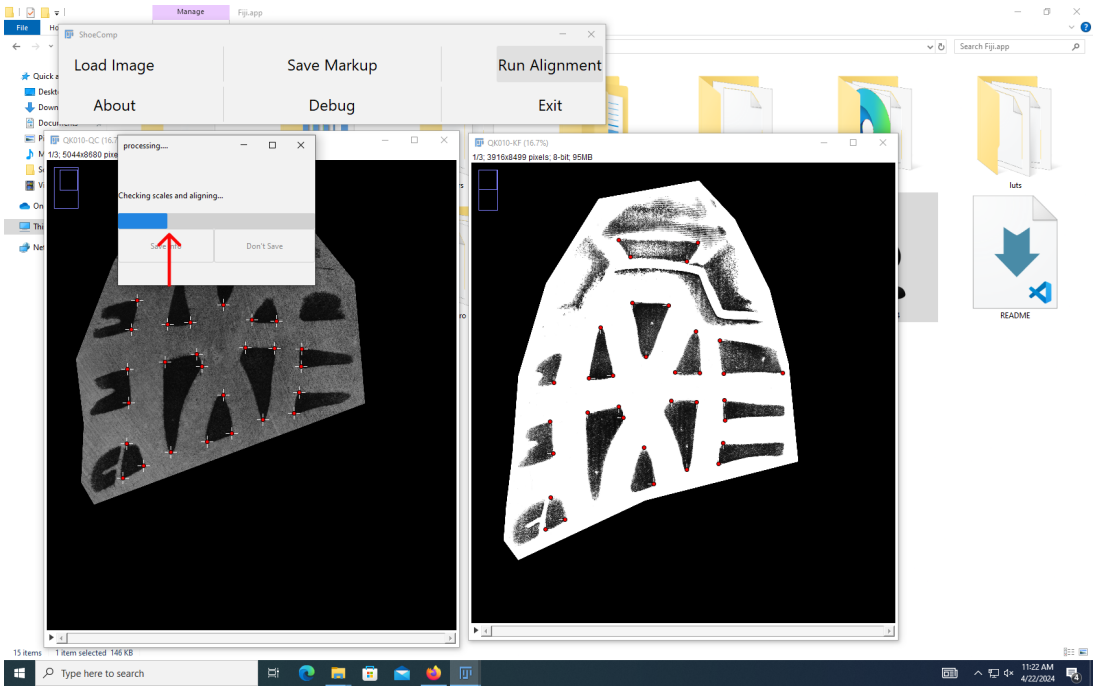

Figure 2.13: The alignment process is running, trying to resolve differences in rotation, translation, and scale.

After the alignment process has completed, we see two new windows: one showing the overlay of the images, and the other showing a similarity score measurement related to the alignment.

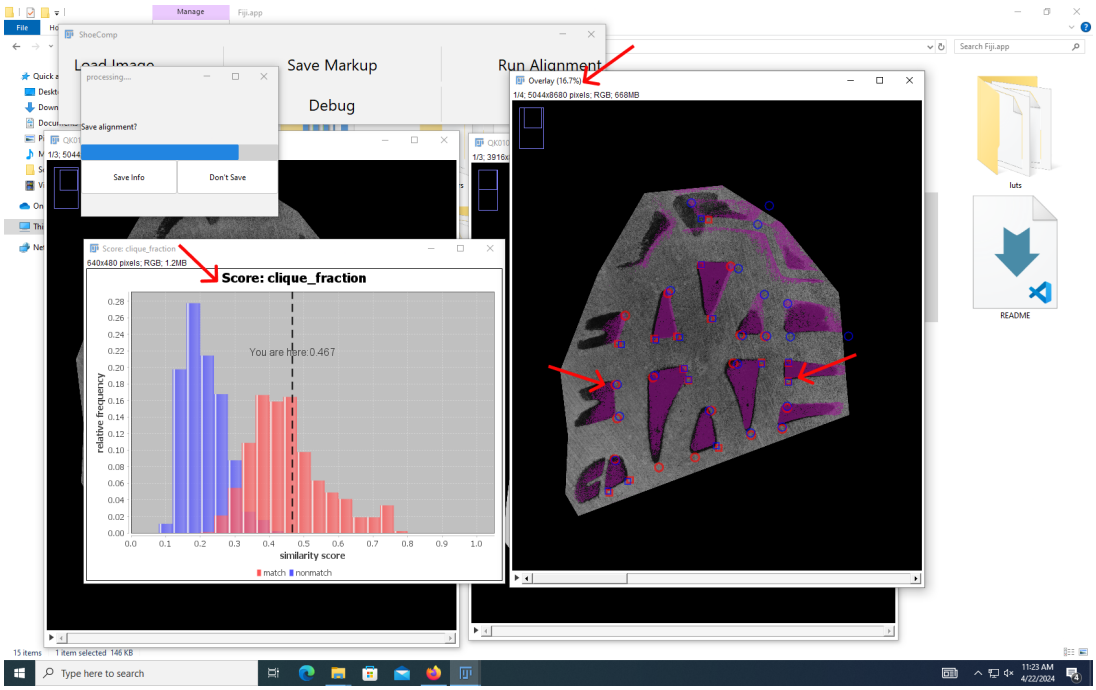

Figure 2.14: Alignment complete. You can look at the purple overlay and the points to see how well the reference image has been aligned to the questioned image. Square points indicate those points were selected in the alignment, and circular points indicate those points which were not. Generally, we would expect the square points to be almost on top of one another, but that depends on the alignment. The similarity score provides a numerical measurement with respect to an example background distribution.

If you'd like to retry the alignment, you can click `Don't Save` and try again. However, this alignment seems okay, so let's click on `Save Info`:

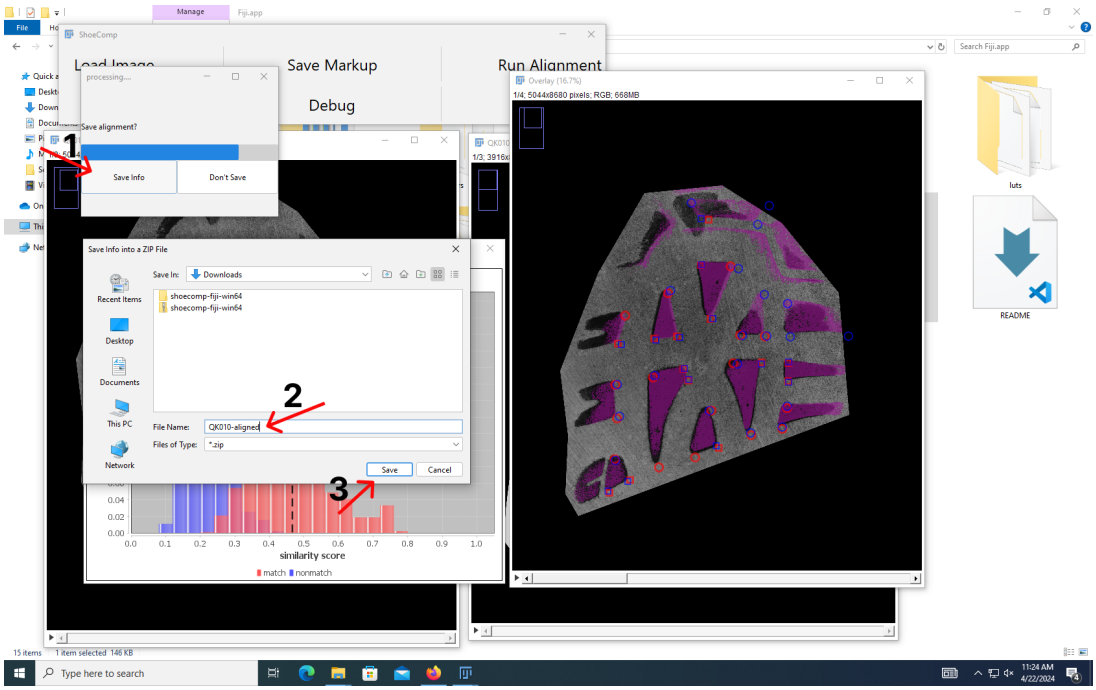

Figure 2.15: Saving the alignment into a ZIP file. Click on `Save Info`, and write the filename of ZIP file into which the alignment will be saved, and then click `Save`.

Once the ZIP file has been saved, you can open it to view the components of the alignment process. The ZIP file at present contains:

- three JSON files: one for each image markup, and one for the points calculated during the alignment,
- three TIFF files corresponding to the questioned image: the original image, the polygon mask indicating the relevant region, and the points marked.
- three TIFF files corresponding to the reference image: the original image, the polygon mask indicating the relevant region, and the points marked.
- three TIFF files corresponding to the reference image, *transformed so that they align to the questioned image*: the original image, the polygon mask indicating the relevant region, and the points marked.

The TIFF images can be overlaid in software like Photoshop to obtain something like the purple overlay for use in a report. In the future, the entire ZIP file can be submitted for further processing and calculation of more similarity scores.

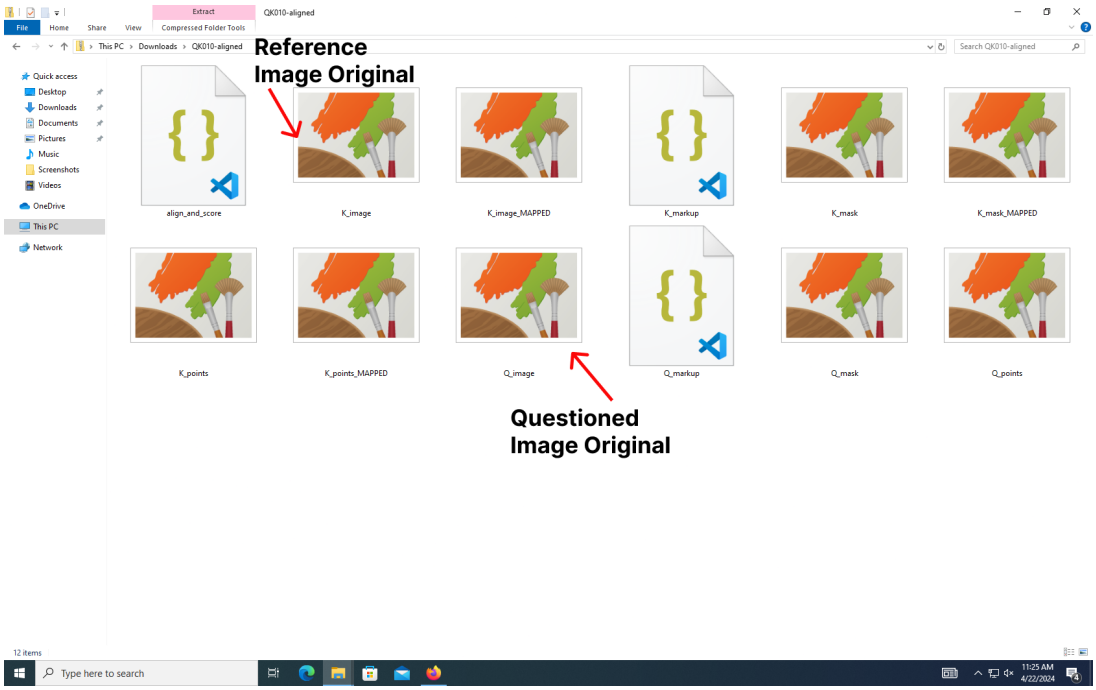

Figure 2.16: The components of the saved ZIP file, indicating a completed alignment.

Summary and Future Work

This document provides a user guide to ShoeComp, a shoeprint markup and alignment tool developed by CSAFE at Iowa State University. The Java-based tool provides a simple user interface to mark interest points on shoeprints, and aligns them using a maximum-clique based method that handles rotation, translation, scale, and a bit of distortion as well. The user can view an overlay of the alignment, and save the aligned images for use in reports, or submit them for further similarity calculations.

3.1 Updating ShoeComp

By default, ShoeComp does not need to connect to the internet to run any updates, and we would prefer to avoid automatic/background internet connections. However, it may sometimes run a local check at startup to confirm all the libraries are available. This check at startup is harmless – you can close the complaining windows if any appear. We hope figure out the relevant code causing this automatic check, and disable it soon.

To update ShoeComp, you can just delete the relevant ZIP file and folders, and download a new one from the latest release on Github: `https://github.com/CSAFE-ISU/shoecomp`

Future improvements for ShoeComp include: adding more similarity scores and visualizations, checking for multiple possible optimal alignments, enabling polynomial/thin-plate-spline alignment transformations, creating a smoother user interface, and of course, fixing errors in the code. Let us know what you'd like to see: `https://forensicstats.org`

3.2 Performance Considerations

The alignment algorithm might take a bit of time if you have more than 30 points marked in each image. The speed of the algorithm is connected to:

- Number of points marked in each image: around 30 is good, but more points means a lot slower. At present, there doesn't seem to be a way around this, but we might find better approximation algorithms.
- The allowable scale range: the default range is 0.8 to 1.2. We recommend picking scale range of $\frac{1}{N}$ to N, but larger ranges will slow down the algorithm a bit.
- Maximum allowed distortion: the more distortion you allow, the slower the algorithm will be. At present, we recommend around 0.5 to 5 degrees of angular distortion at most, and around 0.1 to 1 units of scaling distortion at most.
- Lower bound of common points: the minimum number of points the alignment should have to be considered valid. Keeping a high lower bound means the algorithm will run faster, but it might also not find any alignments that are so good.