Lab04

- Submission: Submission must be made on Canvas
- Format: Your submission will consist of two files:
    - a Python (or MATLAB) file (or tar/zip/archive of code), and
    - a PDF file. Please upload these as two **separate files** – do not group both code and report into the same tar/zip/archive.

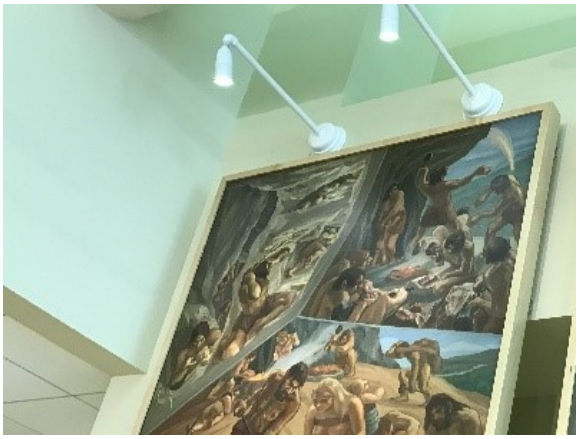
1 Assignment

A "flatbed scanner" is used to take a picture of a flat object, by moving a sensor linearly over the object. However, the size of the object is limited to the size of what will fit in the scanner. In this assignment you will develop software to take a picture of an arbitrarily large flat object, by moving a camera in a freeform motion across the object, and then stitching the images together.

This capability is similar to the "panorama" applications found on smart phones. However, panorama apps require the user to rotate or translate in a straight line. In contrast, your app will allow any rotational or translational motion.

The following are images of the mural in the geology museum, taken by moving a handheld camera in a freeform motion. On the course website, there are 12 such images (mural01.jpg, mural02.jpg, mural03.jpg, mural04.jpg, mural05.jpg, mural06.jpg, mural07.jpg,

mural08.jpg, mural09.jpg, mural10.jpg, mural11.jpg, mural12.jpg). The size of the left-most panel is specified in mural01_dim.docx.




Using the methods covered in class (there are two versions posted for those who are not using MATLAB), develop a program that stitches all 12 images together to make a composite mosaic image of the mural. You should change the first image to an "orthophoto", so that the rest of the images are registered to an orthophoto frame. You can do this manually, by picking points on the first image with the mouse. The orthophoto should have a scale of 1 pixel = 0.5 cm.

Then, match (manually as we did in class) from each successive image to the preceding image, and estimate the homography from the current to the previous image. Then compute the homography from

the current image to the mosaic, and warp the current image to the mosaic. Finally, fuse the warped current image with the mosaic image, using MATLAB code used in class or you can use the function "fuse_color_images" given in the lab files, ".py".

Choose an output image size such that the complete mural fits in the image. Note - due to accumulation of errors, the output orthophoto may not have perfectly straight edges. My results are shown below (this image is subsampled so that the scale is not 1 pixel = 0.5 cm):

2 **Turn in**

1. Your program listing, with comments.
2. A PDF file containing a description of your method of solution and the architecture of your program, your input images, and your output composite images for both examples (the mural example and the one you create).