

# Project Proposal

18-645 How to Write Fast Code

Team Members: Ruolin Zan, Taihai He

We want to work on an image processing project, featuring alpha blending, which uses Python with NumPy and SciPy to blend two images together through matrix transformation and computation. The main program is supposed to read two input images as multi-dimensional arrays and output a result image by calculating how each pixel in the image should be modified based on a blending factor. Usually, input images are split into two different sets of relatively small triangular areas of interest, which are connected with corresponding coordinates. and the program uses Affine transformation, a matrix computation technic, to calculate how each area needs to be blended, and eventually, combine all the triangles together to form a final image. There is no upper limit for the resolutions of input images, and the number and sizes of areas of interest are determined by users solely. In general, the program needs to handle every pixel in all input images and the output image for all three channels and perform matrix transformation for them. An uncountable amount of calculation is unavoidable, due to the fact rectangle matrixes used for calculations are always larger than their corresponding triangle areas. By optimizing the whole process, we expect to deliver a faster program, defined by the improvement of benchmarking results. The project is designed to run on a x86 CPU.

Since we only have two members in a team, the workload in total should be a bit lighter.

First, we need to translate the original 300-line Python code to its C equivalent in order to benchmark. Then we plan to implement a modified C version of the program with same algorithm to compare against the translated C program.

Second, this project could be separated into two parts, dividing the input images and reform the output image. When dividing, triangle masks is necessary to provide specific pigments of the input image. In this way, the order of masks matters and one member of our team will be responsible for it. After that, we will need to put all useful pixels back to the original canvas. There are many different orders to form a picture using pixels, so the other team member will take charge of this part. The difference between the two parts is that triangle masks are quite important in part one since only the part of input images in the triangular area will be kept. However, every pixel we get from part 1 will be put back.

We are both available during the following time slots:

Every Friday, from 4:30 to 5:30 pm

Every Monday, from 3:00 to 4:00 pm

Every Wednesday, from 3:00 to 4:00 pm

If more options are needed, we are also fine with every Friday, from 10:30 to 11:30 am.