Warm up

original image



horizontally 2x stretched image



vertically 2x stretched image



when the empty spaces are filled with the nearest values, some parts were unnatural.

so I averaged the two values in the direction of the stretch.

I just put the values in the stretched position and filled empty spaces with the averages.

parameters

number of ranges(orientation used in sift)

nr = 10

maximum gaussian filter size

xs=13

30 dots per one iteration of difference of gaussian

ndots=30

for line detection used in rectification

sigma=1

threshold=1.5

rhoRes=1

thetaRes=math.pi/60

nLines=4

observations

H for stitching can be computed more precisely by using as many points as possible.

possibly the black region which are generated after stitching can be covered with nearest pixels to look more natural.

since pixels describes something in discrete manner, sometimes stitching cannot be perfectly made.

when computing h,

I used the last row of V^T which is the result, of SVD.

to compute h, matrix is calculated and the form is Ah=0

for normalization, firstly the points that are going to be transformed are firstly normalized.

normalization is done by considering the mean value and the standard deviation so that it is normalized. same thing is done for the target points.

so by multiplying the normalization matrix of input points to computed H matrix and reverse normalization matrix, the final matrix is obtained.

when mosaicing, I calculated the sift closeness and if a pair is not too close, then I didn’t use it.

each iteration of laplacian gave ndots results which are not necessarily different from previous dots.

I used as many pairs as possible.

the criterion of dropping a pair was the closeness value larger than 0.3.

I used difference of gaussian detector and sift descriptor.

I used hough transform to find lines and used the intersection points for rectification.

I used ransac to choose valid pairs.

if this is done manually, it would be good to find more than or equal to four points that are far away from each other.

dynamic programming to reuse previously computed values when doing convolution.

all convolutions were done in separate manner.

printed something when the program is doing something, such as “appl gauss 7”

when stitching, I used laplacian method with the mask line being the line connecting two points where the tangent changes.

to make the result better, I also duplicated the pixels from around that line to the end.

I got the intersection of lines by giving votes on the trail where lines exist.

So the places where two votes are given are the starting points.

then I gave a rectangular form by giving the same height value to both the left points and the right points.

there are two rectification results.

one image with parallel edges and the other with the letters facing front.

the latter one is done by switching the height value(downward coordinate).