There were two main problems in my last report: (1) there were still some visible borders in group 57. (2) in some parts of the final montage, the image that I had chosen to be on top was blurrier than some other images at that area. Also, I was supposed to check the non-rigid registration and see if there is a way to handle it for just a narrow area. In the last version, I first found the transformation that best matched the narrow areas around the merging border of the two images. Then I applied that transformation to the whole image.

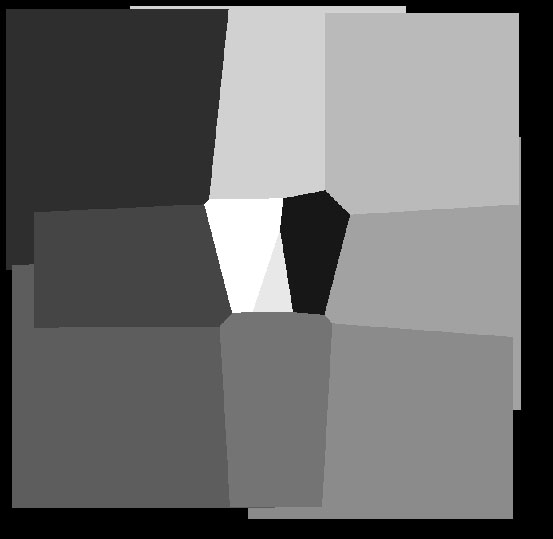
1) The first problem was a terrible mistake in the algorithm. I was trying to cut out some dark regions of the images and it caused visible borders at those areas. They were not because of our merging process, but because of the borders that I used for cutting dark areas. I checked the final images for both group 11 and group 57 and I think the problem of borders is now solved.

2) The problem of blurred regions was because of the criterion that I was using for choosing between images. I used to find gradient of the image and mark regions with low gradient as blurred regions. However, I understood that blurred regions sometimes have even higher gradient values than the other clearer regions. I guess that is because sometimes blurriness puts some disturbances on image that cause intensity difference and as a result, high-value gradients.

Therefore, I tried to find another way. I found a matlab code “fmeasure.m” that contains several criteria for assessing image quality (for each criterion they have mentioned the reference papers too, and it was really helpful, since we can see the references and find out which criterion is the best for us). I tried most of them (actually ones that I thought are related to image blurriness). Finally, I reached the idea that “BREN” algorithm for measuring the focus in the image is related to blurriness of our images.

First, I used this criterion to decide between neighbor areas to find out if we can remove some boundaries as you advised. It worked very well in some cases, but I had a problem with some other cases. When I was comparing some neighboring regions (for example areas 1 and 2 in Figure 1) I found that most parts of image that we are using for area 2 (hereafter I call it image A) are much better than those parts in image that we are using for area 1 (hereafter I call it image B), but there are some small areas in A that are not good. When we compare image A and image B in for example, area 2, the algorithm finds that image B is better than A and uses B for area 2. But the low quality points are disappointing. So, I tried to find out what can we do. First I thought maybe we can do the comparison in smaller blocks. But, it wasn’t a good idea, because it makes more borders, that we are trying to avoid, and also smaller we make the blocks, less reliable will be the result of the BERN algorithm. For example, we can’t decide between two 5\*5 pixels blocks, which one is blurrier. I didn’t consider the time that it requires for that process, but I guess it will also be too time-consuming.

The second idea was that before doing the merging process, I just compare images pairwise and replace the overlapping region in the lower quality image with the overlapping region of the higher quality region. The result is saved in “mergedImNF11\_May29”. However, I found that it is better to reorganize the input images so that successive images be close together (I mean, in group 11, I reorganized images and renamed image 1 to image 12, because it was more similar to image 10 and 11). I can explain why I suggest to do this.

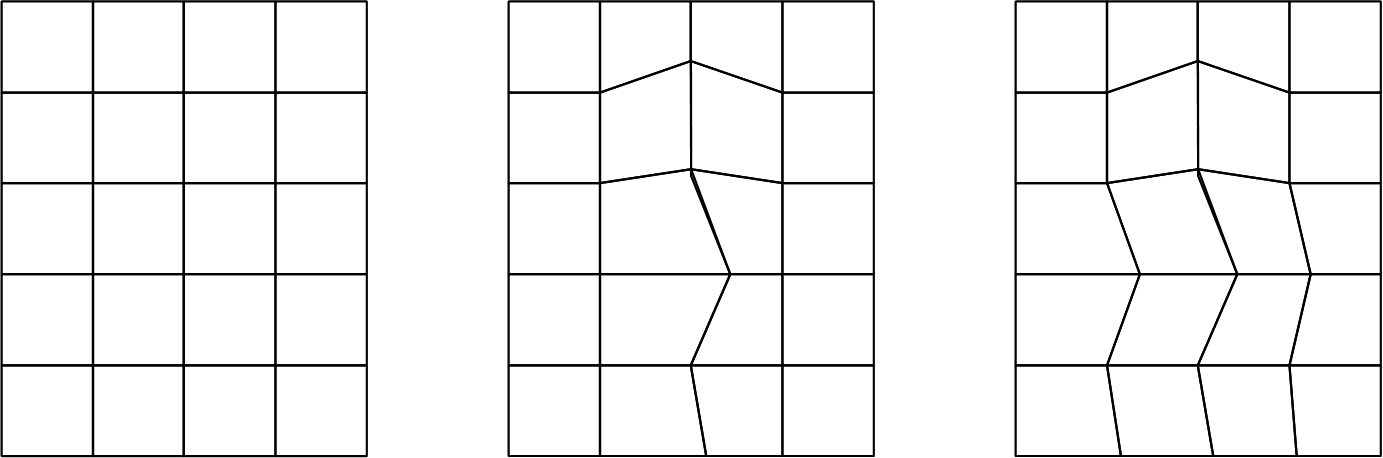


**2**

**1**

Figure

The last problem was about merging only a narrow border of the images. As I understood, your idea is to preserve the boundaries of those narrow areas that we were supposed to register and just register the middle areas (like middle picture in Figure 2 that red boundaries are preserved). I checked the algorithm that MATLAB built-in function uses, and found out that it doesn’t preserve the boundaries. It does something like what I tried to show in right picture in Figure 2 which obviously ruins the boundaries). So, I tried to find another way.



Figure

In my master project I used to use another tool for image registration (I didn’t write the codes by myself, I just downloaded them from internet). I went to those codes and checked them on our images. The result was not still good. Then, I checked some literature and found that none of non-rigid transformation methods can preserve boundaries (Goshtasby, 2005; Hajnal and Hill, 2001).

So I checked affine transformation another time, and as we expected there is no transformation matrix that can transform all parts of the region. I tried affine transformation on two narrow areas of the first image and the second image in group 57.

Figure -(a) the region of figure 1 in group 57 selected for registration. (b) the region of figure 2 in group 57 selected for registration. (c) image 1 after registration. (d) image 2 after registration.

Sorry, I tried to zoom in the image and show the registered area better. But, the quality decreases when I copy the image here. So, please take a look at image “1To2Reg57\_Moving\_May30” and “1To2Reg57\_Static\_May30”.

References

Goshtasby, A.A., 2005. 2-D and 3-D image registration: for medical, remote sensing, and industrial applications. John Wiley & Sons.

Hajnal, J.V., Hill, D.L., 2001. Medical image registration. CRC press.