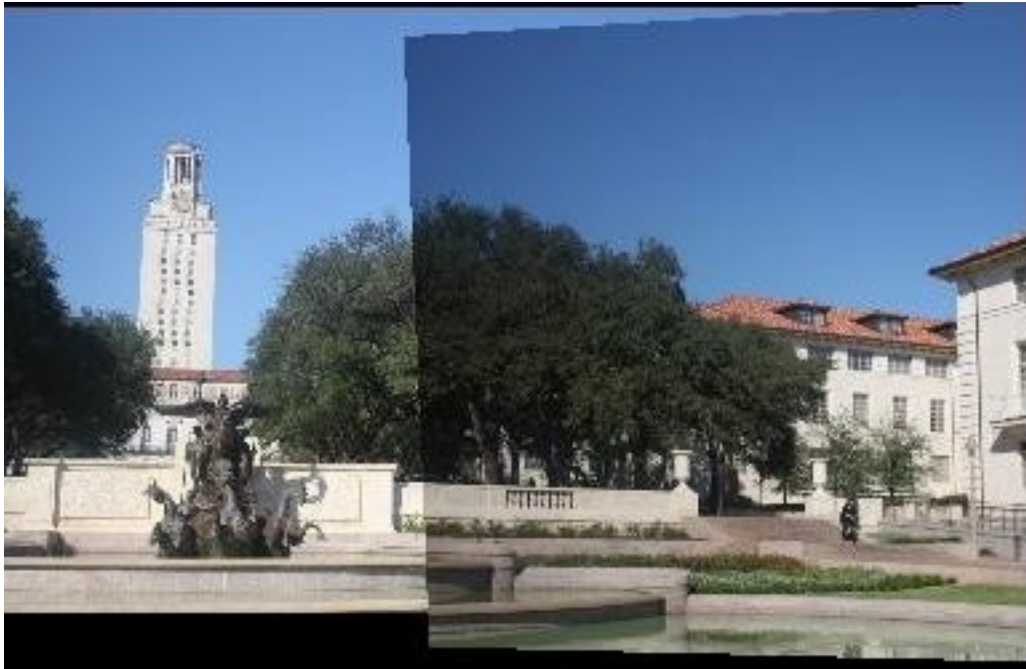


Ashley Bourgoyne  
aab2438

Pset3



This is from the provided images. The results came out relatively well. Any inconsistencies were from manual errors in choosing the interest points.



For this one, I overlaid Botticelli's *Birth of Venus* onto a billboard. The white left over is

Ashley Bourgoyne  
aab2438

### Pset3

simply from my error in choosing the corners well.

Billboard image: [http://www.freakingnews.com/images/app\\_images/billboard.jpg](http://www.freakingnews.com/images/app_images/billboard.jpg)

Birth of Venus: <http://eroticmadscience.com/wp-content/uploads/2010/02/botticelli.birth-of-venus.jpg>



This is my own image. Turned out fairly well, given that it was taken on a phone. Everything is pretty well aligned, showing a good choice in corresponding features.

I first began my creation of this mosaic by letting the user manually enter corresponding points using the `ginput` function in Matlab. The accuracy greatly depended on this step, so these correspondences were carefully chosen and any inaccuracies occurred due to this step.

Next, I computed the homography by building a matrix of knowns, and multiplying it by a matrix containing the points. Using this homography I was able to warp one image onto the image plane of the second.

I began the warping process by determining where the corners of the image to be warped would fall after warping. I used these corners to calculate the size of the warped image. Then I created an array of all the  $x$  and  $y$  coordinates that would be in the transformed image. Using these arrays, I created a meshgrid. Finally, I applied the homography inversely on the  $X$  and  $Y$  matrices returned from the meshgrid to get the points that I would need to extract from the original image (a.k.a. inverse warping). Using these points, I interpolated between the images and was able to create a new, transformed image, with no holes.

The last part of the process was to combine the two images into one panorama. The key part to this step was to replace the NaN pixels in the transformed image with the pixels that were in the same location in the second image. After this was done, the two images (one the second image, and one the transformed image now containing the overlapping pixels from the second image) were placed in the output image.