Problem Set #3

I.

1. For the provided pair of images, view-1 (uttower2.jpg) and view-2 (uttower1.jpg), two different mosaics are created. One mosaic is created by stitching view-2 onto view-1, while the other is created by stitching view-1 onto view-2. For the first mosaic, Figure 1 and 2 show verification of the clicked and transformed corresponding points between two views, and Figure 3 shows the stitched result. For the other mosaic, Figure 4 and 5 show verification of the clicked and transformed corresponding points between two views, and Figure 6 shows the stitched result. While running this program, both views are presented together, and the user needs to pick one pair of corresponding points at a time, which means the user click one point at view-1 and then click the corresponding point at view-2.

![View-1 with Clicked Corresponding Points](uttower2.jpg) ![View-2 with Transformed Corresponding Points](uttower1.jpg)

Figure 1: Verification of clicked and transformed corresponding points.

![View-1 with Transformed Corresponding Points](uttower2.jpg) ![View-2 with Clicked Corresponding Points](uttower1.jpg)

Figure 2: Verification of clicked and transformed corresponding points.
The Mosaic by Stitching View−2 onto View−1

Figure 3: The mosaic by stitching view-2 onto view-1.

View−1 with Clicked Corresponding Points
View−2 with Transformed Corresponding Points

Figure 4: Verification of clicked and transformed corresponding points.

View−1 with Transformed Corresponding Points
View−2 with Clicked Corresponding Points

Figure 5: Verification of clicked and transformed corresponding points.
2. Figure 7 shows a mosaic by stitching three views together, where view-1 (scene1.jpg) and view-2 (scene2.jpg) are stitched onto view-3 (scene3.jpg).

3. In this problem of warping one image into a frame region in the second image, I have extended the capability of my stitching program to warp a polygon image in the first view into a polygon region in the second view. It means that the target frame region can not only be a rectangle, but also any four-sided polygons. Therefore, any four-sided polygon image in one view can be warped into any four-sided polygon frame in the other view. Figure 8 to 15 show four examples of warping one four-sided polygon image into another four-sided polygon frame, including the original image/frame and the result. While running this program,
the four-sided polygon frame is first selected in the first view, and then the four-sided polygon image is selected in the second view. To select the polygon, click the left mouse button to pick the vertex, drag the mouse to form a line, and again click the left mouse button to pick the next vertex. After picking all vertices, double-click the left mouse button to complete selection.

An extra advantage of this system is that it can also handle polygons with more than four sides, though not perfectly. Sometimes the homography matrix will give coordinates way beyond the appropriate range, and sometimes the warped result will be heavily distorted. Since homography deals with projective transformation, if a polygon with more than four sides wants to be warped, it may be more than projective transformation, resulting in incorrect warped polygons. Figure 16 and 17 show the last example where a eight-sided polygon image is warped into a eight-sided polygon frame.
Figure 8: Example 1: original frame in view-1 and original image in view-2.

The Mosaic by Warping a Polygon Image in View–2 into a Polygon Frame in View–1

Figure 9: Example 1: the result.
Figure 10: Example 2: original frame in view-1 and original image in view-2.

The Mosaic by Warping a Polygon Image in View–2 into a Polygon Frame in View–1

Figure 11: Example 2: the result.
Figure 12: Example 3: original frame in view-1 and original image in view-2.

The Mosaic by Warping a Polygon Image in View–2 into a Polygon Frame in View–1

Figure 13: Example 3: the result.
Figure 14: Example 4: original frame in view-1 and original image in view-2.

The Mosaic by Warping a Polygon Image in View–2 into a Polygon Frame in View–1

Figure 15: Example 4: the result.
Figure 16: Example 5: original frame in view-1 and original image in view-2.

The Mosaic by Warping a Polygon Image in View–2 into a Polygon Frame in View–1

Figure 17: Example 5: the result.
II. Extra Credit

1. Figure 18 to 23 show two different stitched mosaics with automatic corresponding point detection and matching using SIFT. The first mosaic, Figure 18 to 20, is created by stitching view-2 onto view-1, while the other mosaic, Figure 21 to 23, is created by stitching view-1 onto view-2. Comparing the same, manually-created mosaics, we can see that SIFT does a almost perfect job of automatically detecting and matching corresponding points, and the output mosaics look even better than the manually-created ones. Note that when using SIFT, some parameters need to be adjusted to give reasonable results, e.g. peak thresholds and edge thresholds, which both control the number of detected features. In addition, after matching SIFT features, all matched pairs are sorted from low to high based on the distance between the two features in each matched pair. The four matched pairs with lowest distances are chosen as the final four pairs of corresponding points.

2. Figure 24 to 29 demonstrate the effectiveness of using RANSAC for image stitching. From Figure 24 to 26, we can see that the output mosaic is incorrect if there is some outlier corresponding point and no RANSAC is used. On the other hand, RANSAC can get rid of that outlier and produce the desired mosaic, as shown in Figure 27 to 29.

3. Figure 30 and 31 show an example of rectifying an interested planar surface into a fronto-parallel view. We can see that there are unrecognizable characters in the original planar surface, and after the rectification, they become quite clear and readable.
Figure 18: Verification of clicked and transformed corresponding points.

Figure 19: Verification of clicked and transformed corresponding points.

Figure 20: The mosaic by stitching view-2 onto view-1.
Figure 21: Verification of clicked and transformed corresponding points.

Figure 22: Verification of clicked and transformed corresponding points.

Figure 23: The mosaic by stitching view-1 onto view-2.
Figure 24: Verification of clicked and transformed corresponding points.

Figure 25: Verification of clicked and transformed corresponding points.

Figure 26: The mosaic by stitching view-2 onto view-1.
Figure 27: Verification of clicked and transformed corresponding points.

Figure 28: Verification of clicked and transformed corresponding points.

Figure 29: The mosaic by stitching view-1 onto view-2.
Figure 30: Example: output rectangle in view-1 and interested planar surface in view-2.

The Output Fronto-Parallel View

Figure 31: Example: output fronto-parallel view.