Computational Photography Assignment 5

Single Michael
08-917-445

1 Morphing

The following figures show some static results produced by my morphing function. I also have produced some .avi movies. You can find them either in *outputs/p5/* or in the provided results zip on Ilias. I rendered the movies once using a linear time-stepping function and another time using a cosine-ramp. Furthermore, for my results, I used 42 frames. The duration of the movies is 3 seconds. In order to make your own morphing videos, please make use of the function *makeMorphingVideo.m* and read its description (how to use).

Fig. 1: Source Image (left) and target image (right) used for morphing.

Fig. 2: Selected Features in Source Image (left) and Selected Features in Target Image (right) used for morphing indicated by red crosses.

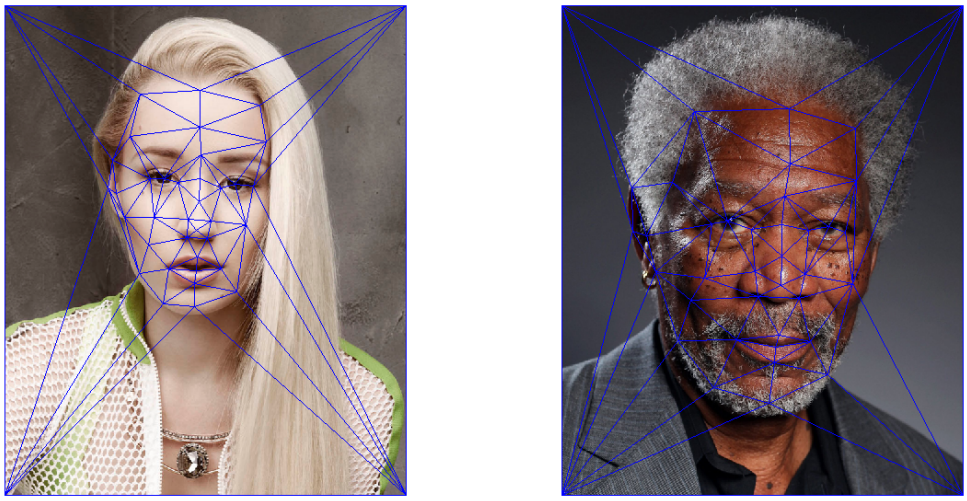

Fig. 3: Delaunay triangulation using selected features in Source image (left) and Delaunay triangulation using selected features in Target image (right) used for morphing.

Fig. 4: An intermediate morphed image (between source and target) using linear timesteps. here t is equal 0.5.

2 Rectification using Homography

Figure 5 is the example shown during the exercise session. Its result, when applying a rectification is shown in figure 6. This example acts as a sanity check whether my implementation seems to work as expected. In addition I have produced rendering for another example, a distorted church shown in figure 7. Figure 8illustrated the user selection for the rectification process (used for computing the homography). Figure 9shows the result of the homographic image rectification applied to this church image.

In order to make your own rectification, please make use of the function *homographicRectification.m* and read its description (how to use).

Fig. 5: Input image of a building which exhibits a notable distortion.

Fig. 6: Rectified input image using Homography. Most of the vertical edges of the building have become parallel.

Fig. 7: Input image of a church which exhibits a notable distortion.

Fig. 8: Image Rectification selection (read lines) from specified user points (in blue).

Fig. 9: Rectified input image using Homography. Most of the vertical edges of the church have become parallel.

3 Panorama Stitching

For this task I have taken two pictures from our balcony at my place. The pictures show the center of the village in which I live. The input images are shown in figure 10. The resulting panorama image is shown in figure 15. The other figures are showing intermediate results used for producing the final panorama.

In order to make your own panorama stitching, please make use of the function *panoramaStitching.m* and read its description (how to use).

Fig. 10: Source Image (left) and target image (right) used for Panorama Stitching. Both images were taken from our balcony at my place.

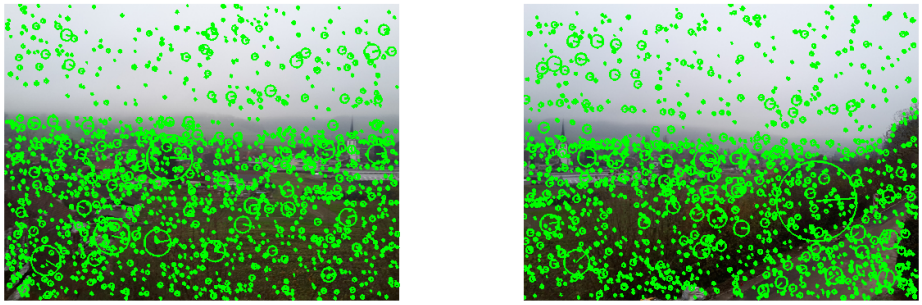

Fig. 11: Illustration of SIFT frames in both given input images using the VLFeat library.

Fig. 12: Showing bleeding masks of both images: On the left the mask for the first image and on the right the mask for the second given image.

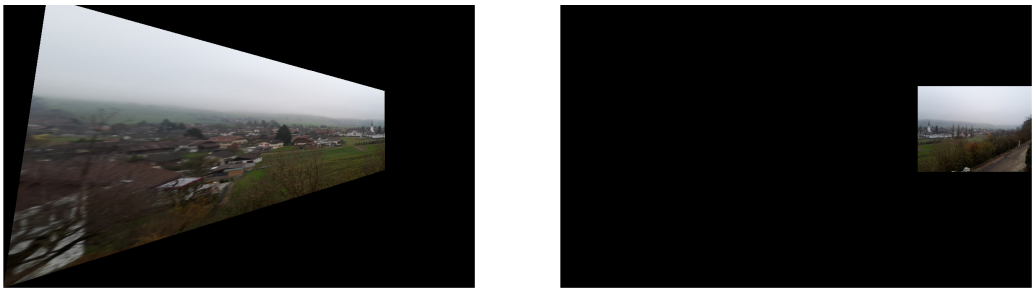

Fig. 13: Showing padded image version of both images (after having applied the homography): On the left the padded image for the first image and on the right the padded image for the second given image.

Fig. 14: Illustration of stitched padded images (using alpha blending - based on the bleeding masks).

Fig. 15: Final result depicting the Panorama Stitched images. Note that this image is just a cropped version (from given two user specified points) of the stitched padded images from before.