CS364/AM792: Assignment #2

Due on March 19, 2020 at 17:00

Unathi Skosana

Problem 1

Solution

(a)

Let $R' = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $\mathbf{r} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$, $\mathbf{r}' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$ and $\mathbf{r}_0 = \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$

Then

$$R'\mathbf{r} = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\cos\theta + y\sin\theta \\ 1 \end{bmatrix}$$

$$R'\mathbf{r}_0 = \begin{bmatrix} x_0\cos\theta - y_0\sin\theta \\ x_0\cos\theta + y_0\sin\theta \\ 1 \end{bmatrix}$$

$$\begin{aligned} \mathbf{x}' = s(R'(\mathbf{r} - \mathbf{r}_0) + \mathbf{r}_0) &= s(R'\mathbf{r} - R'\mathbf{r}_0 + \mathbf{r}_0) \\ &= R's\mathbf{r} - R's\mathbf{r}_0 + s\mathbf{r}_0 \end{aligned}$$

Simplifying further yields

$$\begin{aligned} \mathbf{x}' &= s\begin{bmatrix} x\cos\theta - y\sin\theta - (x_0\cos\theta - y_0\sin\theta) + x_0 \\ x\cos\theta + y\sin\theta - x_0\cos\theta - y_0\sin\theta + y_0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} s\cos\theta & -s\sin\theta & -sx_0\cos\theta + sy_0\sin\theta + sx_0 \\ s\cos\theta & s\sin\theta & -sx_0\cos\theta - sy_0\sin\theta + sy_0 \\ 0 & 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H\mathbf{x} \end{aligned}$$

where

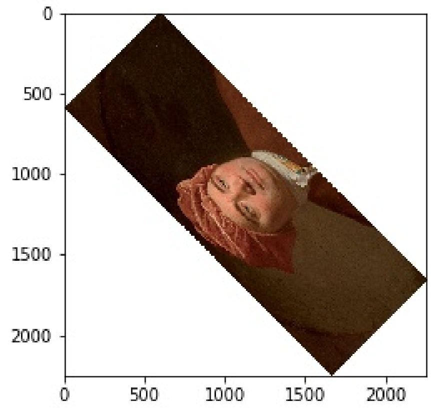
$$H = \begin{bmatrix} s\cos\theta & -s\sin\theta & -sx_0\cos\theta + sy_0\sin\theta + sx_0 \\ s\cos\theta & s\sin\theta & -sx_0\cos\theta - sy_0\sin\theta + sy_0 \\ 0 & 0 & s \end{bmatrix}$$

(b)

The homography H above was applied to a portrait of David Hume below

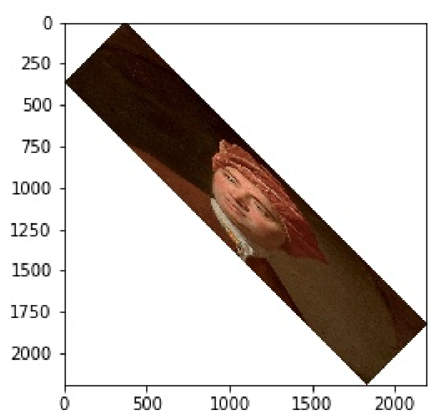
Figure 1: David Hume

for the parameters (s, θ) as $(2.25, -\frac{\pi}{6}), (2.0, \frac{\pi}{3})$ and $(1.25, \frac{\pi}{10})$ respectively. The results are shown below



(a) $(2.0, -\frac{\pi}{6})$

(b) $(2.0, \frac{\pi}{3})$



(c) $(1.25, \frac{\pi}{10})$

Figure 2: Homographically scaled and rotated David Hume

(c)
The reason why the translation part of the homography H is redundant is simply because when applying the homography to image, the shift in the output image's center is compensated thus nullifying the effect of any kind of translation in H.

The reason why for $\theta > 0$ the image appears to rotate clockwise rather counterclockwise is because a planar homography relates the transformation between two planes. Thus the homography rotates the plane rather the points/vectors on it counterclockwise. Such a rotation is equivalent to rotating the points/vectors on that plane clockwise.

Problem 2

Solution
For this question I used the movie poster below :

Figure 3: Interstellar movie poster

I manually selected the following four corners on the building where the corners of the poster should map to.

$$
\begin{aligned}
(318, 111) &\mapsto (w-1, 0) \\
(104, 239) &\mapsto (0, 0) \\
(31, 622) &\mapsto (0, h-1) \\
(315, 567) &\mapsto (w-1, h-1)
\end{aligned}
$$

where $w = 1028$ and $h = 1500$ are the width and height of the poster respectively.

from which, following the necessary steps produces the homography H

$$
H = \begin{bmatrix} 5.7000\text{e-}4 & -2.1000\text{e-}4 & 3.9900\text{e-}1 \\ -5.6000\text{e-}4 & 5.8000\text{e-}4 & 9.1694\text{e-}1 \\ 0 & 0 & 3.8400\text{e-}3 \end{bmatrix}
$$

Naively applying the homography above to the poster and try overlay on the building we encounter the issues aforementioned, i.e. pixels containing no image data and axes shift. The first issue can easily be addressed by initializing the transformed image with completely black (or white) pixels then we can replace these pixels with the pixels of the building when overlaying the two images.

The second issue is can solved by specifying that the transformed image should have same dimensions as griest.jpg and remove the compensation for the transformed image's origin shift as we want the origin of griest.jpg and transformed image to align.

Problem 2 continued on next page...

The issues were above were modified by changing the already provided apply_homography function ever so slightly in the following manner.

```
def apply_homography(A,H,output_shape=None):
    ...
    ...
    h, w = (None, None) if output_shape is None else output_shape
    B = np.zeros((mm,nn,c) if output_shape is None else (h,w,c)) + 255
    for x in range(nn if w is None else w):
        for y in range(mm if h is None else h):
        p = np.array([x + minx if w is None else x, \
            y + miny if h is None else y, 1]).reshape((3,1))
        ...
        ...
    ...
```

Equipped with this modification, we can now apply the homography to the poster and overlay onto the building

(a) Perspectively warped poster

(b) Overlaid result

Aliasing effects

We have already encountered aliasing effects in first problem set. If one attempts to upscale (or downscale) an image by a considerably large scale s, either using nearest-neighbour or bilinear interpolation. we can get aliasing effects such as jagged edges. We saw this when we zoomed in on the scale images. This happens due to the fact whenever do image transformations such as scale transformations or rotations there is a loss of image detail. Non-adaptive interpolation algorithms such as nearest-neighbour and bilinear interpolation treat all pixels on the same footing and do not try to minimize aliasing artifacts, and can only become more accurate one includes more adjacent pixels at expense of computational time.

Aliasing effects due to downscaling are highlighted in the following experiment. Consider the poster of the blade runner movie below, with dimensions 2000 by 3085. This is considering larger than required size of the output image i.e 528 by 704.

Figure 5: Blade runner

For the movie poster above will be downscaled for it to fit on the output image and see the effects of this downscaling below

Figure 6: Aliased blade runner

The aliasing artifacts I can notice in particular example are blurring effects and jagged edges. I conjecture that the we would also see aliasing artifacts if the dimensions of the input image were considerably smaller than those of the output image.

Problem 3

Solution

Inspecting the image, looking at the three bricks in the second row on the side where the gentlemen is standing. Assuming that each brick is 300 pixels wide and that the art is on a square tile, we get that the height and width of the transformed image should both be 900 pixels.

Under the assumptions above, we can construct the following set of correspondences.

$$
\begin{aligned}
(54.1045, 716.468) &\mapsto (0, h-1) \\
(775.672, 809.152) &\mapsto (w-1, h-1) \\
(817.91, 547.516) &\mapsto (w-1, 0) \\
(352.118, 509.971) &\mapsto (0, 0)
\end{aligned}
$$

where $w = 900$ and $h = 900$. Computing the homography matrix gives

$$
H = \begin{bmatrix} -5.3000\text{e-}4 & -7.6000\text{e-}4 & 5.7353\text{e-}1 \\ 1.4000\text{e-}4 & -1.7000\text{e-}3 & 8.1918\text{e-}1 \\ 0 & 0 & 1.0000\text{e-}4 \end{bmatrix}
$$

Applying the homography above with the modified function produces the result below.

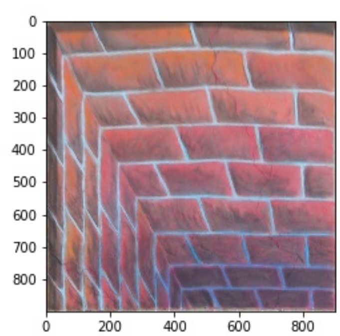


Figure 7: Warped bricks

Problem 4

Solution
(a)

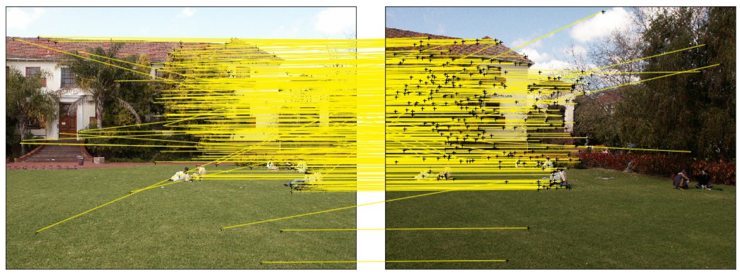


Figure 8: SIFT matches between the two images

(b) The RANSAC procedure was implemented as described in the lecture slides. In my implementation, a match mapped from the first image to the second image is 'close enough' to its SIFT match if the distance between them is less than 0.25.

Problem 4 continued on next page…

On the plot below, the largest consensus set was of size 34

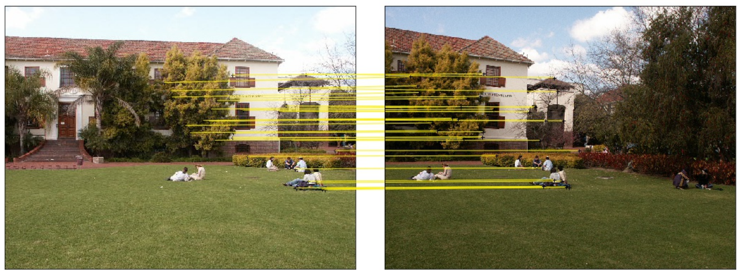


Figure 9: Largest consensus set of SIFT matches between the two images

and the homography determined from this set of inliers was :

$$H = \begin{bmatrix} 3.9300\text{e-}3 & -1.1000\text{e-}4 & -9.7799\text{e-}1 \\ 4.7000\text{e-}4 & 3.5700\text{e-}3 & -2.0857\text{e-}1 \\ 0 & 0 & 2.8700\text{e-}3 \end{bmatrix}$$

(c)

Before we apply the homography to first image we need to figure out the dimensions of the transformed image. We can do this by using the homography matrix, applying the homography to the point $(0, 0, 1)$ gives us where origin of the original would be on the image axis of the second image, call this point p_1. The transformed image should negative since we are stitching to the right and the roof in the second image is slightly tilting upwards.

Mapping the point $(w-1, h-1, 1)$ with H where w and h are the width and height of original image respectively, and call this point p_2. Taking the difference between the homogenized points p_1 and p_2, call this homogenized vector o. This vector gives the values of how much we need to add to the dimensions of the first image to accommodate the second image on the transformed image.

In this scheme, the dimensions of the transformed image becomes

```
nh = h + o[0]
nw = w + o[1]
```

Finally we compensate for the axis shift in the transformed image with determined homogenized vector p_1. This required another simple modification to apply_homography.

```
def apply_homography(A,H,output_shape=None, offsets=(0,0)):
    ...
    ...
    fx, fy = offsets
    for x in range(nn if w is None else w):
        for y in range(mm if h is None else h):
            # compensate for the shift in B's origin
            p = np.array([x + minx if w is None else x + fx, \
                y + miny if h is None else y + fy, 1]).reshape((3,1))
            ...
            ...
    ...
    ...
```

Using the function above with appropriate parameters to apply the homography and get the transformed image t_im1 then we can stitch to it, im2 in place like so

```
fx, fy = abs(int(p1[0])), abs(int(p1[1]))
t_im1[fy-1:fy+im2_h, fx-1:fx+im2_w] = im2
```

where im2_h and im2_w are the height and width of the second image. Below is the stitched output

Figure 10: Homography stitch