CS 763/764: Computer Vision

Task 02b

Question 1. What linear algebra technique is used in 1(a)? Explain.

Solution.
For finding the homography matrix from 4 pairs of points, linear algebra techniques were used. We define 4 points from image-1 as X_1, X_2, X_3, X_4 and corresponding points from image-2 as X'_1, X'_2, X'_3, X'_4. Where all X_i and X'_i are in form $[x_i, y_i, 1]^T$ and $[x'_i, y'_i, 1]^T$. We also define a $\tilde{X_i}'$ where $\tilde{X_i}'$ is $[u'_i, v'_i, w'_i]^T$ and related with X'_i as $x'_i = \frac{u'_i}{w'_i}$ and $y'_i = \frac{v'_i}{w'_i}$ Then Homography transform can be given by equation:

$$\tilde{X_i}' = HX$$

Expanding above equation for X_1 and $\tilde{X}_1$ we have:

$$\begin{bmatrix} u'_1 \\ v'_1 \\ w'_1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_6 & h_7 & h_8 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix}$$

After Expanding the above matrix and rewriting the equation for all eight points we will have:

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1 \cdot x'_1 & -x'_1 \cdot y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x'_1 \cdot y_1 & -y_1 \cdot y'_1 & -y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2 \cdot x'_2 & -x'_2 \cdot y_2 & -x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x'_2 \cdot y_2 & -y_2 \cdot y'_2 & -y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3 \cdot x'_3 & -x'_3 \cdot y_3 & -x'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x'_3 \cdot y_3 & -y_3 \cdot y'_3 & -y'_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4 \cdot x'_4 & -x'_4 \cdot y_4 & -x'_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x'_4 \cdot y_4 & -y_4 \cdot y'_4 & -y'_4 \end{bmatrix} \cdot \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = P \cdot H = 0$$

H has 8 degrees of freedom and we apply an additional constraint that $|H| = 1$ in order to ensure a non-trivial solution. Then H is found using Singular Value Decomposition(SVD) of P, and taking the solution corresponding to the least eigen value as H. We use this matrix and all the points from image-1 to transform image-1 into image-2 from $\tilde{X_i}' = HX$. This way the linear algebra techniques like SVD and matrix multiplication are used in 1(a).

Question 2. What attempts did you make to get a meaningful frame-rate? Explain.
Solution.
Initially when we were loading the video into the image, the image being very large, overflowed across the device's screen. So to cure this, we had to resize the frames before sending them out to be displayed. We tried using the cv2.resize() function inside the while loop for playing the video which caused the while loop to slow down and thus leading to a very poor frame rate due to the computational load of resizing the image. To correct this, we resized the original image of the wall containing photo frame to the frame dimensions of the video, and then used the conventional method to map the video on to the photo frame using cv2.getPerspectiveTransform() and cv2.warpPerspective(). This resulted in a very smooth video with a good frame rate. For saving the video we are using the same frame rate as in original video.

Question 3. How did γ achieve the task? Explain.
Solution.
The algorithm we used is explained as below. Three functions namely **alpha()**, **beta()** and **gamma(img_bg, img_insert)** are defined. The **gamma()** function is called and 3.jpg and arch.jpg are passed as arguments. Now this function maps the arch.jpg on to the image 3.jpg. The function **gamma()**, inside its body calls two functions, **alpha()** and **beta()** which use the pair of images (2.jpg, 1.jpg), and (3.jpg, 2.jpg) to compute the corresponding transformation matrices. **alpha()** function returns a matrix M_1 to convert perspective of an image from 2.jpg to 1.jpg, similarly **beta()** returns a matrix M_2 to convert perspective of an image from 3.jpg to 2.jpg. These matrices M_1 and M_2 are multiplied with each other, and essentially used in the function cv2.warpPerspective() in **gamma()** to get the perspective of the image arch.jpg in 1.jpg directly. This is in accordance with the perspective of the image the instructor has in the thought experiment.

Question 4. Explain your understanding of rmse computation over the domain of choice.
Solution.
The RMSE (Relative Mean Squared Error) computation as instructed in the assignment was to compute the relative mean squared difference of the images averaged over the pixels. That is essentially to compute the average over pixels of both image and taking square of the difference between the two averages. We believe that is not a correct evaluation method as averaging over the pixels of two images might be same even if the images are widely different (consider a toy example: a = [1,3,2], b = [4,1,1] and MSE = 0). We remove this by first taking the square of differences between each and every pixels of two images and then taking the mean over the squared difference to maintain the generality (if images are of different sizes). RMSE found between output of image_perspect.py and sequential.py using method described in assignment is 538.37 and 47.69 using the modified method.

Question 5. Is (d) same as (c)? Explain.
Solution.
No, (d) is not same as (c). The reason is that the mapping of the 16 reference points we took in both the images, do not correspond to the points we mapped in the transformed image due to perspective projection. For example, it is not necessary that midpoint of the picture frame in the distorted image will directly correspond to the midpoint of PQ segment of the arch.jpg, hence there is slight misalignment in the output. We believe if the corresponding pairs of images is taken perfectly (cross ratio between the points is maintained) then taking 16 points will give better results than just 4 points. RMSE found between (c) and (d) using the method described in assignment is 10.62 and using the modified method is 52.968. Code for this portion can be found in convincingDirectory.

Question 6. Why did you choose the images that you did for (e)?
Solution.
The 3 images that we used in this task were taken from different perspective so as to observe the fitted picture, and how it responds to the transformations depending upon the perspective of the frame in which it is put in. Also the image we used to put inside the frame to make it seem as if we have a window in the room that opens up to the snow mountains, i.e. Narnia movie but cabinet replaced by a bulletin board. As for the video, it was a completely random choice. Code for experimentation with self images can be found in convincingDirectory.

Question 7. What codec did you use in (e)?
Solution.
We used .avi codec for (e) because we tried .mp4 already and wanted to test on some other codec.

Question 8. (Optional) What is missing in Fig. 1d and how is that relevant to “students and test-takers”?
Solution.
On the top of the arch, there is an important message that is not visible well in the image. ”Gyanam Parmam Dhyeyam” that closely translates to knowledge is the supreme goal. We believe this philosophy is what distinguishes students from test-takers.