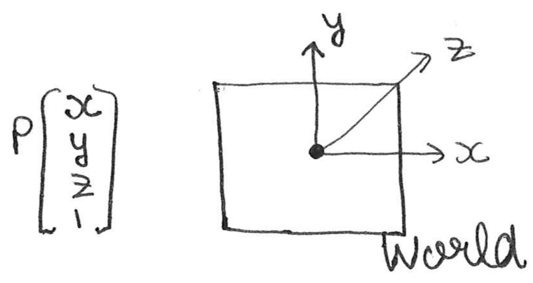
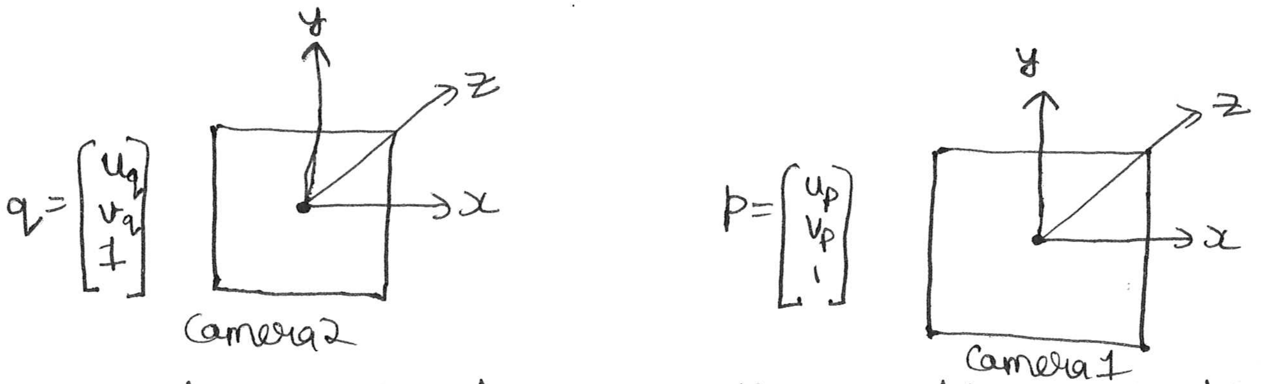
CV-Assignment-1

Naman Kumar
namank

Q3.1





$q=\begin{bmatrix}u_q\\v_q\\1\end{bmatrix}$
y
z
x
Camera2
$p=\begin{bmatrix}u_p\\v_p\\1\end{bmatrix}$
y
z
x
Camera1

We are choosing the frame of the world so that it is aligned with frame of camera2, therefore (Also $z_p=0$)

$\Rightarrow$ $$q = M_1 P \Rightarrow q = K_q[R\ t]P$$

$$= K_q[I\ t]\ P$$ ($\because$ there is no rotation) as they are aligned

$$= K_q\begin{bmatrix}1&0&0&d_x\\0&1&0&d_y\\0&0&1&d_z\\ \end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

Now as z coordinate of $P=0$, we can write it as

$$q = K_q\begin{bmatrix}1&0&d_x\\0&1&d_y\\0&0&d_z\end{bmatrix}\begin{bmatrix}x\\y\\0\\1\end{bmatrix}$$ ($\because$ we can eliminate column 3)

$$\Rightarrow q = \underbrace{K_q \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & dz \end{bmatrix}}_{e} \begin{bmatrix} x \\ y \\ \\ 1 \end{bmatrix}$$

$\hookrightarrow M_1$ is a 3×3 matrix

$\Rightarrow$ For Camera 1

$$p = M_2 P = K_p = [I\ t]\ P \Rightarrow K_p = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Similarly, as $Z = 0$ for P

M_2 reduces to a 3×3 matrix because we can get rid of column 3.

$$p = M_2 P$$
$$q = M_1 P$$

$$\boxed{P = (M_1)^{-1} M_2\, q}$$

Now, as $M_1 = K_q \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & dz \end{bmatrix}$ and $\det(K_q) \neq 0$ and $dz \neq 0$

$\Rightarrow \det(M_1)$ is not zero.

Therefore M_1^{-1} exists which is a 3×3 matrix

Hence $\underline{p \equiv Hq}$ ($M_1^{-1} * M_2$ is a 3×3 matrix which shows H exists)

Note: When axis of 2 cameras are parallel, we have infinite solutions.

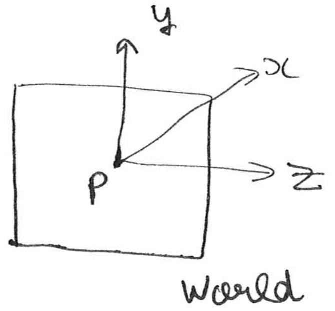
Q3.2

P is a point in the world.

$$P = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

p is a point on C_1

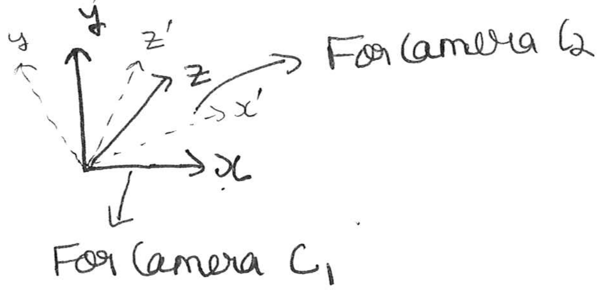
p' is a point on C_2



$\Rightarrow$

$p = K_1 [I \ \ 0] P$

$p' = K_2 [R \ \ 0] P$



1. $P = K_1 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

As $t = 0$, therefore we can get rid of 4th column

$P = K_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Rightarrow P = \underbrace{K_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

M_1 is a 3x3 matrix

2. $p' = K_2 \begin{bmatrix} R & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \end{bmatrix} P \Rightarrow p' = K_2 \begin{bmatrix} R \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$ $(\because t = 0)$

$\Rightarrow$ (we can get rid of 3rd column)

$p' = \underbrace{K_2 R}_{} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$

$\downarrow$

M_2 is a 3x3 matrix

$$P = M_1 \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$P' = M_2 \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\frac{P'}{P} = (M_1)^{-1} M_2 \Rightarrow P' = (M_1)^{-1} M_2 P$$

$$M_1 = K_1 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow \det(M_1) \neq 0$$

$$\Rightarrow \text{inv}(M_1) \text{ exists} \qquad \boxed{H = (M_1)^{-1} M_2}$$

$P' = HP$ where H is a 3×3 matrix

Q3.3 Planar homography is not completely sufficient to map any arbitrary sceneimage to another viewpoint because:

1. It considers only the scene which is common from both the cameras and it ignores the remaining part which is not common.
2. It requires that images should be far away to avoid parallax effects.
3. Occlusion can ~~the~~ create a problem in case of planar homography.
4. It restricts the points to a plane and hence reduces Degrees of freedom by 1.

Q3.4

$$p_1^i \equiv H p_2^i$$

$$p_1' = (x_1', y_1', 1)^T \Rightarrow p_1' = \begin{bmatrix} x_1' \\ y_1' \\ 1 \end{bmatrix}$$

$$p_2' = (x_2', y_2', 1)^T \Rightarrow p_2' = \begin{bmatrix} x_2' \\ y_2' \\ 1 \end{bmatrix}$$

$$\Rightarrow x_1' = \frac{H_1^T p_2'}{H_3^T p_2'}$$

$$y_1' = \frac{H_2^T p_2'}{H_3^T p_2'} \qquad \Rightarrow \qquad A = \begin{bmatrix} (p_2')^T & [0,0,0] & -p_1'(1,1) * (p_2')^T ; \\ [0,0,0] & (p_2')^T & -p_1'(2,1) * (p_2')^T \end{bmatrix}$$

$$A * h = 0 \Rightarrow \begin{bmatrix} (p_2')^T & [0,0,0] & -p_1'(1,1) * (p_2')^T ; \\ [0,0,0] & (p_2')^T & -p_1'(2,1) * (p_2')^T ; \\ & \vdots & \\ & \vdots & \end{bmatrix} * \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ \vdots \\ h_9 \end{bmatrix} = 0$$

(a) There are 9 elements in h

(b) We need atleast 4 points (no. of pts ≥ 4) to solve the system.

(c) To find h, we have to find the eigen vector corresponding to the minimum eigen value of $(A^T * A)$. This is Rayleigh Quotient Theorem. After getting a 9X1 vector for h, reshape it to get a 3X3 H matrix. (Linear Least Squares)

$$Ah = 0$$

$$\min \frac{\|Ah\|^2}{\|h\|^2} \quad \rightarrow h^T A^T A h = \|Ah\|^2$$

Q4.

i) Here, we have to find H value using ComputeH function. For that I wrote A matrix using
A = [A;p2(:,i)' [0,0,0] -p1(1,i)*p2(:,i)'; [0,0,0] p2(:,i)' -p1(2,i)*p2(:,i)'];
Then A*h = 0. To find H, I found the eigen vector corresponding to the minimum eigen value of (A'*A) and then reshape it to 3*3 Matrix to get H.

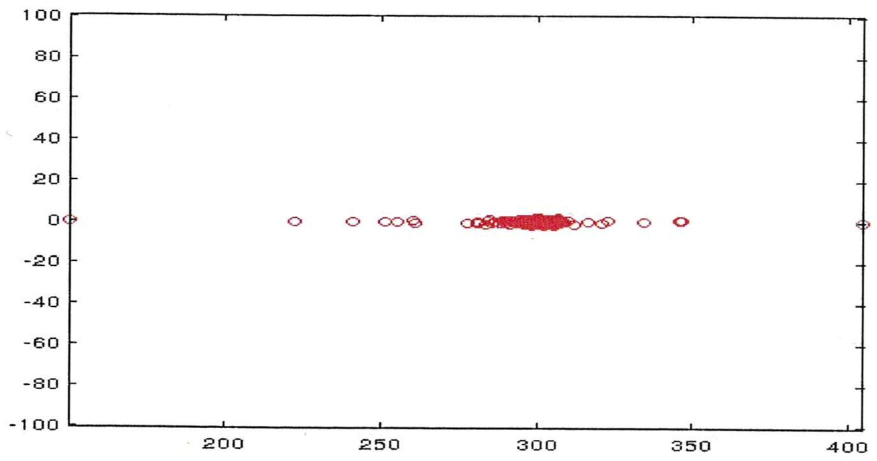
ii) Here, before applying ComputeH, we have to normalize input set of points that is make centroid as [0,0] and average distance to origin as sqrt(2).
For that, I used a tranformation matrix :
Hnorm2 = [S2 0 -S2*centx2; 0 S2 -S2*centy2; 0 0 1]; where S is the scaling factor.
I used 2 transformation matrices for 2 different set of points.
After normalization, run computeH using normalized set of points to get Hnorm and then go back to H because we will be working on images which are not normalized.

Q5. i) Here we will be finding Hnorm and H using P and Pcorrupt (noise is added). Then I found two corresponding set of points (1000 points) for ptest:
a) Set of points by multipying with H.
b) Set of points by multiplying with Hnorm.

ii) After that, I plotted ptestH and ptestHnorm.

iii) The I found out the covariance and standard deviation. The standard deviation ratio (unnormalized/normalized) is very large. After running normCompare 1000times, the average value of standard deviation ratio came out to be around 80.

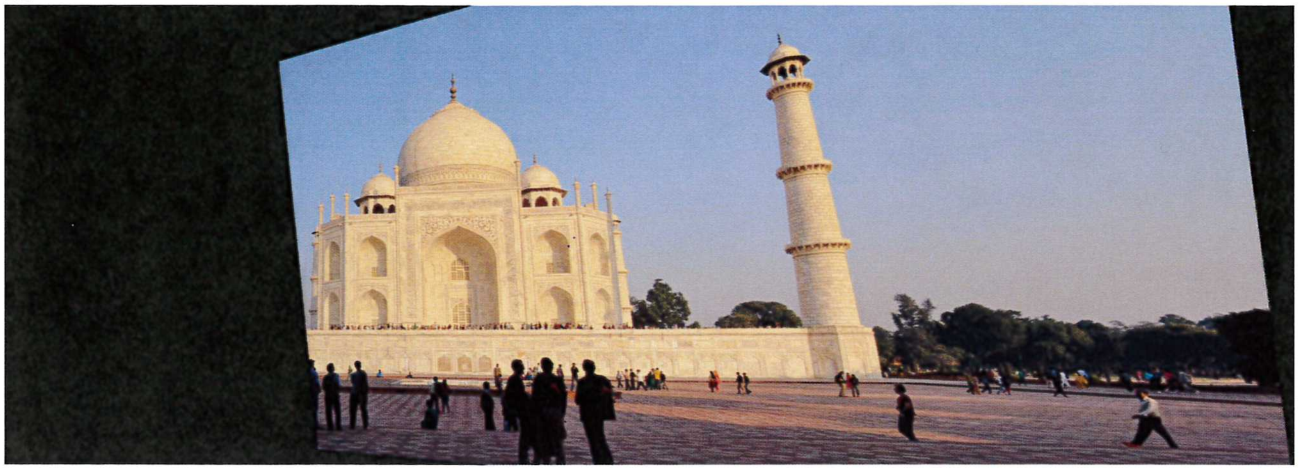
All the ptest points corresponding to Hnorm are very close to each other and look almost like a dot on the plot which shows that ptest is homogaphed to almost the same point everytime and noise does not have much effect on it whereas in case where we have H which is not normalized, all the corresponding ptest points are scattered which shows that ptest gets homographed to a different point every time depending on the noise and is largely affected by the noise. So, clearly it is better to normalize the points before computing H, so **computeH_norm (Normalized) algorithm is better** because it is not affected by noise.



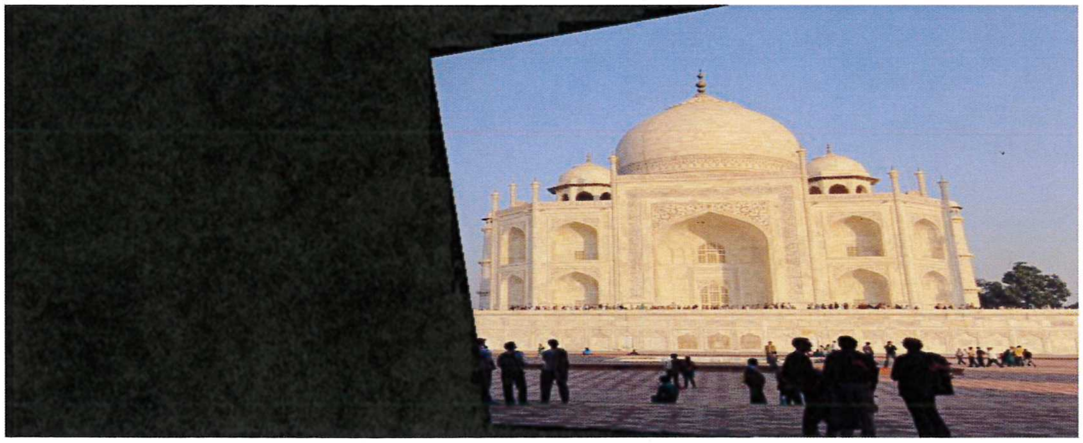
iv) After running it for 1000 times, the average value of standard deviation ratio came out to be around 80. Also, Standard deviation for the normalized case remains almost constant whereas standard deviation for the un-normalized case keeps on varying by a huge amount. This shows that normalized case is not affected by noise whereas the un-normalized case is greatly affected by noise which is not a good thing. Therefore normalized case algorithm is better.

Q6 i) `H2to1 = T1\(Hnorm*T2);`

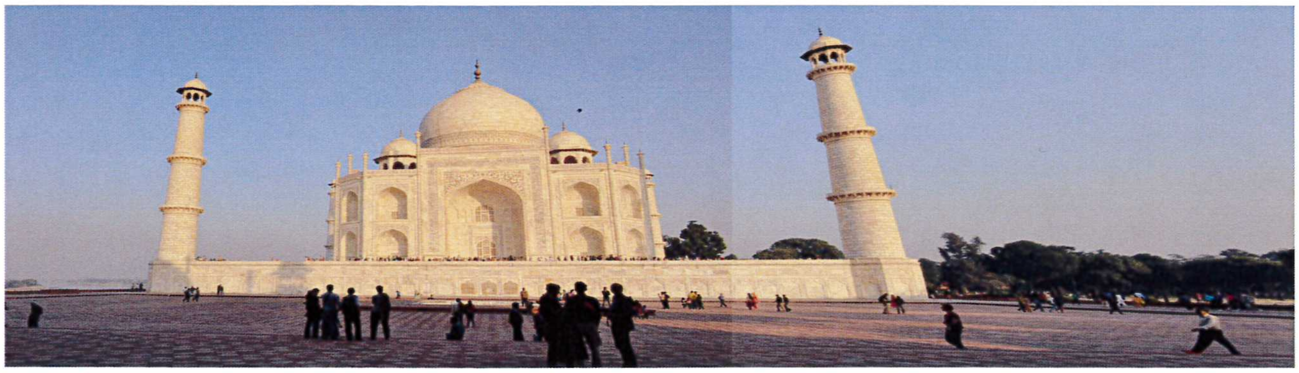
```
Now warp Image2 using warpH.m
Output =
1.
```



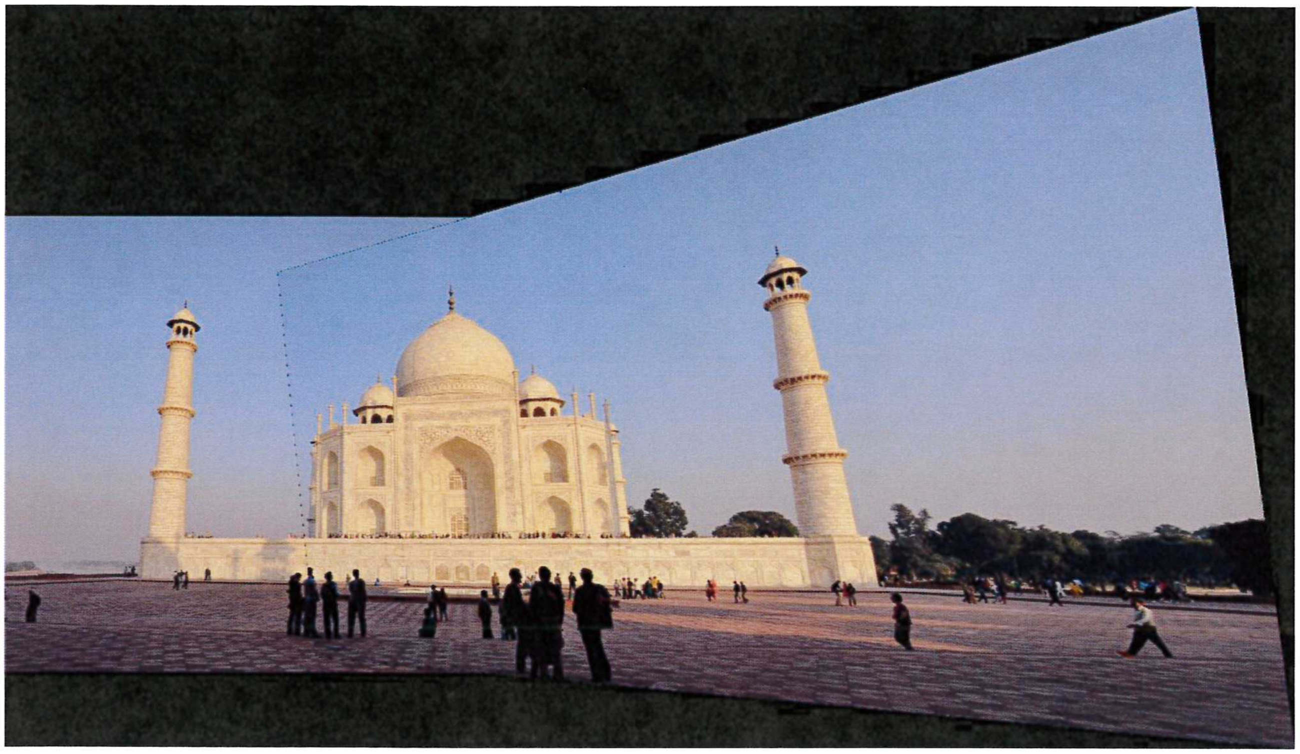
2. Common Part



Then form the Panorma image:



ii) First, find out the H matrix using corresponding points of 2 images.
Then, form a Transformation matrix which takes into consideration Scaling and translation. Let it be S. It is done by first finding the homographed vertices of the 4 vertices of the original image 2 and then finding the scaling value using those vertices and then do the translation to fit the warped image 2 properly.
Now use S to get warped image 1 and use S*H2to1 to get warped image 2.
After that, add warped image 1 to warped image 2 to get the final mosaic image.



Q7. I used cpselect to find the correspondent points between two images. Then saved those points as ecpts.mat andthen found panorma from 2 input images ec1.jpg and ec2.jpg. I saved the result as ec_pan.jpg and the script as ec.m

Final Panorma Image