CV Assignment

1) We know that for $\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \in \mathbb{P}^2$ then $\begin{pmatrix} x_1/x_3 \\ x_2/x_3 \end{pmatrix}$ is the corresponding pt. in $\mathbb{R}^2$

$\therefore \begin{pmatrix} 4 \\ 8 \\ 12 \end{pmatrix} \longrightarrow \begin{pmatrix} 4/12 \\ 8/12 \end{pmatrix} = \begin{pmatrix} 1/3 \\ 2/3 \end{pmatrix} \in \mathbb{R}^2$

2) If $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$ then $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \in \mathbb{P}^2$.

$\therefore$ The possible representations are $k\begin{pmatrix} 3 \\ 8 \\ 1 \end{pmatrix}$ where $k \in \mathbb{R}$

3) $L = (5,2,1)$, $L' = (5,2,9)$.

Now $a = a'$, $b = b' \Rightarrow L$ and L' are parallel.

Intersection of L and L' $= L \times L'$

$$= \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ 5 & 2 & 1 \\ 5 & 2 & 9 \end{vmatrix} = \begin{pmatrix} 16 \\ -40 \\ 0 \end{pmatrix} = 8\begin{pmatrix} 2 \\ -5 \\ 0 \end{pmatrix}$$

$\therefore$ The intersection of $(5,2,1)$ and $(5,2,9)$ in $\mathbb{P}^2$ is the point $\begin{pmatrix} 2 \\ -5 \\ 0 \end{pmatrix}$.

4) $H = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$, $C = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 1 \end{pmatrix}$, $\underline{x}' = H\underline{x}$

The conic equation is $\underline{x}^T C \underline{x} = 0 \Rightarrow \underline{x}'^T H^{-T} C H^{-1} \underline{x}' = 0$

$\therefore$ The conic representation is $H^{-T} C H^{-1}$.

$$H^{-1} = \begin{pmatrix} 1/2 & 1/2 & -1/2 \\ 1/2 & -1/2 & 1/2 \\ -1/2 & 1/2 & 1/2 \end{pmatrix}.$$

$$\therefore H^{-T} C H^{-1} = \begin{pmatrix} 1/2 & 1/2 & -1/2 \\ 1/2 & -1/2 & 1/2 \\ -1/2 & 1/2 & 1/2 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1/2 & 1/2 & -1/2 \\ 1/2 & -1/2 & 1/2 \\ -1/2 & 1/2 & 1/2 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1/2 & 1/2 & -1/2 \\ 1/2 & -1/2 & 1/2 \\ -1/2 & 1/2 & 1/2 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$\therefore$ the projective transformation I_1 under H is $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

5) $H = \begin{pmatrix} 1 & 3 & 5 \\ 4 & 3 & 7 \\ 5 & -4 & 2 \end{pmatrix}$, $(5,6) \in \mathbb{R}^2$ in I_1.

$(5,6) \in \mathbb{R}^2 \longmapsto (5,6,1) \in \mathbb{P}^2$

$H : \mathbb{P}^2 \longrightarrow \mathbb{P}^2$, $x \longmapsto Hx$

$$\begin{pmatrix} 1 & 3 & 5 \\ 4 & 3 & 7 \\ 5 & -4 & 2 \end{pmatrix} \begin{pmatrix} 5 \\ 6 \\ 1 \end{pmatrix} = \begin{pmatrix} 28 \\ 45 \\ 50 \end{pmatrix} \in \mathbb{P}^2$$

The corresponding pt. in $\mathbb{R}^2$ is $\begin{pmatrix} 28/50 \\ 45/50 \end{pmatrix} = \begin{pmatrix} 14/25 \\ 9/10 \end{pmatrix} \in \mathbb{R}^2$ in I_2.

6) $L_\infty = (0,0,1)$. $H = \begin{pmatrix} 2 & 3 & -4 \\ -4 & 3 & 2 \\ 3 & 1 & 1 \end{pmatrix}$

Note that, if x lies on L_∞ then Hx lies on $H^{-T} L_\infty$.

$\therefore L_\infty$ is mapped to $H^{-T} L_\infty$.

$$H^{-1} = \begin{pmatrix} 1/84 & -1/12 & 3/14 \\ 5/42 & 1/6 & 1/7 \\ -13/84 & 1/12 & 3/14 \end{pmatrix}$$

$$H^{-T} L_\infty = \begin{pmatrix} 1/84 & 5/42 & -13/84 \\ -1/12 & 1/6 & 1/12 \\ 3/14 & 1/7 & 3/14 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -13/84 \\ 1/12 \\ 3/14 \end{pmatrix}.$$

$\therefore$ the resulting line is $(-13/18 \quad 1/12 \quad 3/14)$

Question 7

Solution 7.a)

```
import cv2
import numpy as np
import matplotlib.pyplot as plt


img1 = cv2.imread("image1.jpg")
img2 = cv2.imread("image2.jpg")


plt.figure(figsize=(10, 10))
plt.imshow(cv2.cvtColor(img1, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7faee9df4130>
```


```
pts1 = np.array([(850,1250),(1700,2790),(3750,1500),(2750,150)])


img1_copy = img1.copy()
for pt in pts1:
  cv2.circle(img1_copy, pt,radius =0,color=(0,0,255),thickness=64)
plt.figure(figsize=(10, 10))
plt.imshow(cv2.cvtColor(img1_copy, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7faee99447c0>
```


```
plt.figure(figsize=(10, 10))
plt.imshow(cv2.cvtColor(img2, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7faee80aa670>
```

```
pts2 = np.array([(1400, 600), (1400, 2180),(3700, 2100), (3500, 500)])


img2_copy = img2.copy()
for pt in pts2:
  cv2.circle(img2_copy, pt,radius =0,color=(0,0,255),thickness=64)
plt.figure(figsize=(10, 10))
plt.imshow(cv2.cvtColor(img2_copy, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7faee807b910>
```

Solution 7.b)

$X_i' = (x_i', y_i', w_i')^T$, Where i =1,2,3,4

We have A =

$$\begin{bmatrix} 0 & 0 & 0 & -w_1' * X_1^T & y_1' * X_1^T \\ w_1' * X_1^T & 0 & 0 & 0 & x_1' * X_1^T \\ 0 & 0 & 0 & -w_2' * X_2^T & y_2' * X_2^T \\ w_2' * X_2^T & 0 & 0 & 0 & x_2' * X_2^T \\ 0 & 0 & 0 & -w_3' * X_3^T & y_3' * X_3^T \\ w_3' * X_3^T & 0 & 0 & 0 & x_3' * X_3^T \\ 0 & 0 & 0 & -w_4' * X_4^T & y_4' * X_4^T \\ w_4' * X_4^T & 0 & 0 & 0 & x_4' * X_4^T \end{bmatrix}$$

```
A = []
for i in range(len(pts1)):
  A.append([0, 0, 0, pts1[i][0], pts1[i][1], 1, -pts2[i][1]*pts1[i][0], -pts2[i][1]*pts1[i][1], -pts2[i][1]])
  A.append([pts1[i][0], pts1[i][1], 1, 0, 0, 0, -pts2[i][0]*pts1[i][0], -pts2[i][0]*pts1[i][1], -pts2[i][0]])
A=np.array(A)
U,D,V = np.linalg.svd(A)
print("The rank of the matrix is:",np.linalg.matrix_rank(A))
```

```
The rank of the matrix is: 8
```

```
V.shape
```

```
(9, 9)
```

```
H_cal = V[-1,:].reshape(3,3)
H_cal
```

```
array([[-5.10572212e-04,  2.74170651e-04, -8.54353586e-01],
       [-2.73962711e-04, -5.53673045e-04,  5.19691045e-01],
       [ 1.43314724e-08, -1.33664334e-08, -6.70921319e-04]])
```

```
H_cal = H_cal/V[-1,-1]
H_cal
```

```
array([[ 7.61001622e-01, -4.08647994e-01,  1.27340354e+03],
       [ 4.08338062e-01,  8.25242885e-01, -7.74593132e+02],
       [-2.13608839e-05,  1.99225050e-05,  1.00000000e+00]])
```

Solution 7.c)

```
H, status = cv2.findHomography(pts1,pts2)
```

```
H
```

```
array([[ 7.61001622e-01, -4.08647994e-01,  1.27340354e+03],
       [ 4.08338062e-01,  8.25242885e-01, -7.74593132e+02],
       [-2.13608839e-05,  1.99225050e-05,  1.00000000e+00]])
```

Solution 7.d)

```
img1_warp = cv2.warpPerspective(img1, H_cal, (img2.shape[1], img2.shape[0]))
plt.figure(figsize=(10, 10))
plt.imshow(cv2.cvtColor(img1_warp, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7faee8026af0>
```

As we can see in image1, the book was tilted and the table on which it has been kept was almost straight but now after applying homography, the book is straight but the table has becme tilted. There was also a dark brown floor below the table which can be seen in image1 as well as image2, but since our image1 after homography didn't know what to do with the remaining portion, it has been coloured black.

Question 8

```
img3 = cv2.imread("image3.jpg")
plt.figure(figsize=(10, 10))
plt.imshow(cv2.cvtColor(img3, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7faee8011cd0>
```


```
pts3 = [(190,140),(200,2820),(3800,2800),(3750,70)]
img3_copy = img3.copy()
for pt in pts3:
  cv2.circle(img3_copy, pt,radius =0,color=(0,0,255),thickness=64)
plt.figure(figsize=(10, 10))
plt.imshow(cv2.cvtColor(img3_copy, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7faee7f6de50>
```


```
A = []
for i in range(len(pts1)):
  A.append([0, 0, 0, pts1[i][0], pts1[i][1], 1, -pts3[i][1]*pts3[i][0], -pts3[i][1]*pts1[i][1], -pts3[i][1]])
  A.append([pts1[i][0], pts1[i][1], 1, 0, 0, 0, -pts3[i][0]*pts3[i][0], -pts3[i][0]*pts1[i][1], -pts3[i][0]])
A=np.array(A)
U,D,V = np.linalg.svd(A)


H_cal = V[-1,:].reshape(3,3)
H_cal = H_cal/V[-1,-1]
img1_warp = cv2.warpPerspective(img1, H_cal, (img3.shape[1], img3.shape[0]))
plt.figure(figsize=(10, 10))
plt.imshow(cv2.cvtColor(img1_warp, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7faee7f4fe20>
```

Solution 9

1st use case of Homograpgy: Virtual Billboards

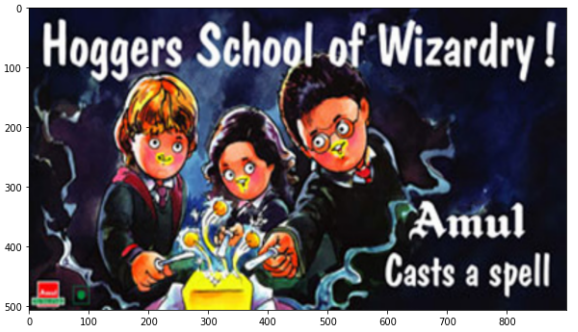
```
b1 = cv2.imread("board1.jpg")
b2 = cv2.imread("board2.jpg")
b2.shape
```

```
(506, 900, 3)
```

```
b1 = cv2.resize(b1,(900,506))
```

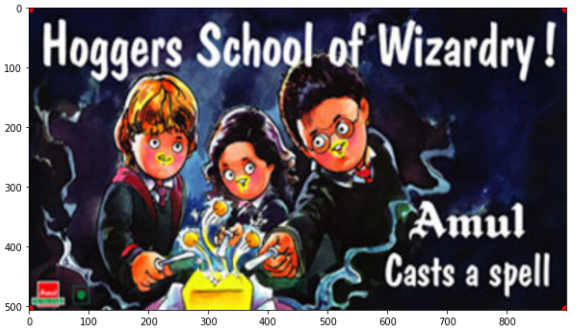
```
plt.figure(figsize=(10, 10))
plt.imshow(cv2.cvtColor(b1, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7faee7eae490>
```




```
pts1_b = [(1,1),(1,505),(899,505),(899,1)]
b1_copy = b1.copy()
for pt in pts1_b:
  cv2.circle(b1_copy, pt,radius =0,color=(0,0,255),thickness=14)
plt.figure(figsize=(10, 10))
plt.imshow(cv2.cvtColor(b1_copy, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7faee7e87d60>
```




```
plt.figure(figsize=(10, 10))
plt.imshow(cv2.cvtColor(b2, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7faee7de06d0>
```

```
pts2_b = [(150,0),(140,220),(410,260),(420,120)]
```

```
b2_copy = b2.copy()
for pt in pts2_b:
  cv2.circle(b2_copy, pt,radius =0,color=(0,255,0),thickness=14)
plt.figure(figsize=(10, 10))
plt.imshow(cv2.cvtColor(b2_copy, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7faee7dbbfa0>
```

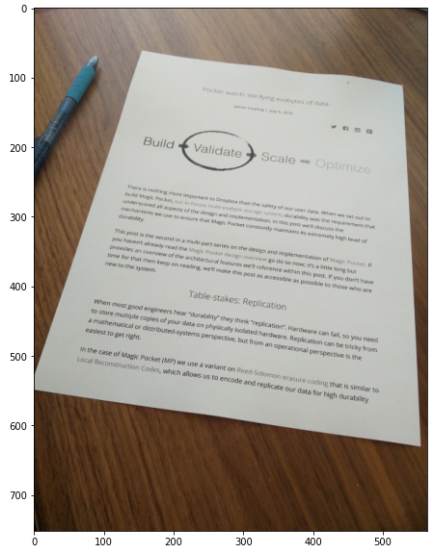
```
A = []
for i in range(len(pts1_b)):
  A.append([0, 0, 0, pts1_b[i][0], pts1_b[i][1], 1, -pts2_b[i][1]*pts1_b[i][0], -pts2_b[i][1]*pts1_b[i][1], -pts2_b[i][1]])
  A.append([pts1_b[i][0], pts1_b[i][1], 1, 0, 0, 0, -pts2_b[i][0]*pts1_b[i][0], -pts2_b[i][0]*pts1_b[i][1], -pts2_b[i][0]])
A=np.array(A)
U,D,V = np.linalg.svd(A)
H_cal = V[-1,:].reshape(3,3)
H_cal = H_cal/V[-1,-1]
b1_warp = cv2.warpPerspective(b1, H_cal, (b2.shape[1], b2.shape[0]))
pts2_b = np.array(pts2_b)

cv2.fillConvexPoly(b2, pts2_b.astype(int), 0, 16)

# Add warped source image to destination image.
b_final = b2 + b1_warp

# Display image.
plt.figure(figsize =(10,10))
plt.imshow(cv2.cvtColor( b_final,cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7faee7d29880>
```

2nd use case of Homography: Aligning Documents

```
tilt = cv2.imread("tilt.png")
plt.figure(figsize=(10, 10))
plt.imshow(cv2.cvtColor(tilt, cv2.COLOR_BGR2RGB))
```

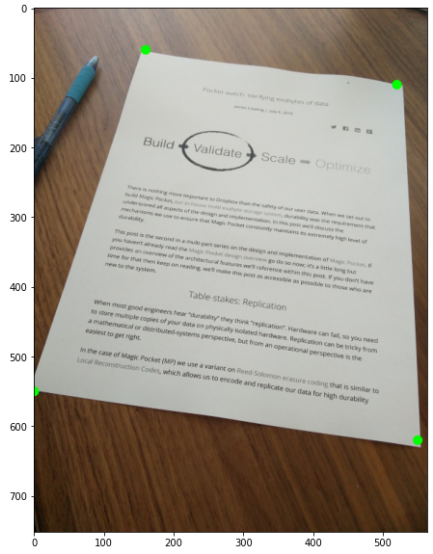
```
<matplotlib.image.AxesImage at 0x7faee7cfb850>
```




```
pts1 = [(160,60),(0,550),(550,620),(520,110)]
pts2= [(0,0),(0,750),(600,750),(600,0)]
tilt_copy = tilt.copy()
for pt in pts1:
```

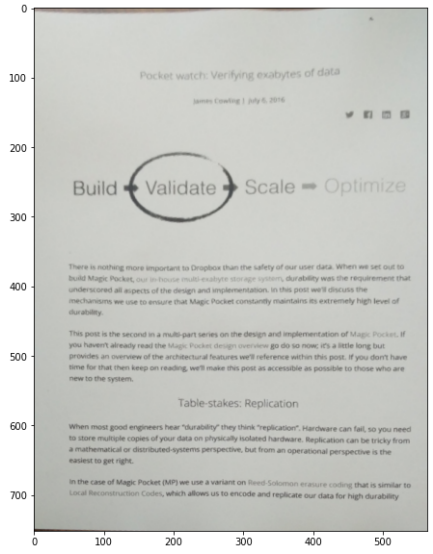
```
  cv2.circle(tilt_copy, pt,radius =0,color=(0,255,0),thickness=14)
plt.figure(figsize=(10, 10))
plt.imshow(cv2.cvtColor(tilt_copy, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7faee80ef880>
```




```
A = []
for i in range(len(pts1)):
  A.append([0, 0, 0, pts1[i][0], pts1[i][1], 1, -pts2[i][1]*pts1[i][0], -pts2[i][1]*pts1[i][1], -pts2[i][1]])
  A.append([pts1[i][0], pts1[i][1], 1, 0, 0, 0, -pts2[i][0]*pts1[i][0], -pts2[i][0]*pts1[i][1], -pts2[i][0]])
A=np.array(A)
U,D,V = np.linalg.svd(A)
H_cal = V[-1,:].reshape(3,3)
H_cal = H_cal/V[-1,-1]
b1_warp = cv2.warpPerspective(b1, H_cal, (b2.shape[1], b2.shape[0]))


img1_warp = cv2.warpPerspective(tilt, H_cal, (tilt.shape[1], tilt.shape[0]))
plt.figure(figsize=(10, 10))
plt.imshow(cv2.cvtColor(img1_warp, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7faee80e24f0>
```



✓ 0s completed at 9:05 PM