In solving the questions in the assignment, I worked together with my classmate [Shimiao Wang 1004779634]. I confirm that I have written the solutions / code / report in my own words.

Part I Q_1 1) # iterations needed $= \frac{\log(1-0.995)}{\log(1-0.7^4)}$.

$= 19.30.$

$= 20$ (since it's integer)

2) fitting an affine transformation would require fewer RANSAC iterations since k would be 3.

$\frac{\log(1-p)}{\log(1-p^{k'})}$ so. $1-p^{k'} > 1-p^k$

$\Rightarrow \frac{\log(1-p)}{\log(1-p^{k'})} < \frac{\log(1-p)}{\log(1-p^k)}$

Q_2 $\vec{p} = \begin{pmatrix} w x \\ w y \\ w \end{pmatrix} = k\vec{P}$

$= \begin{pmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_0 + td_x \\ Y_0 + td_y \\ Z_0 + td_z \end{pmatrix}$

$= \begin{pmatrix} fX_0 + ftd_x + p_x Z_0 + p_x td_z \\ fY_0 + p_y td_y + p_y Z_0 + p_y td_z \\ Z_0 + td_z \end{pmatrix}$

Assume $t \to \infty$ to get vanishing point

$$x = \frac{wx}{w} = \frac{fx_0 + ftdx + p_x Z_0 + p_x t d_z}{Z_0 + t d_z}$$

$$y = \frac{wy}{w} = \frac{fY_0 + ftdy + p_y Z_0 + p_y t d_z}{Z_0 + t d_z}.$$

$$\lim_{t\to\infty} x = \frac{f d_x + p_x d_z}{d_z} = \frac{f d_x}{d_z} + p_x$$

$$\lim_{t\to\infty} y = \frac{f d_y + p_x d_z}{d_z} = \frac{f d_y}{d_z} + p_y.$$

$\therefore$ Vanishing point $= \left(\frac{f d_x}{d_z} + p_x , \frac{f d_y}{d_z} + p_y \right)$.

2) since all lines perpendicular to normal vector. n.

$$n_x dx + n_y dy + n_z dz = 0$$

$$n_x \frac{dx}{dz} + n_y \frac{dy}{dz} + n_z = 0,$$

$$\frac{dy}{dz} = -\frac{n_z}{n_y} - \frac{n_x dx}{n_y dz}.$$

$$V_y = \frac{f dy}{dz} + p_y = f\left(-\frac{n_z}{n_y} - \frac{n_x dx}{n_y dz}\right) + p_y.$$

$$= -\frac{f d_x n_x}{n_y d_z} - f\frac{n_z}{n_y} + p_y$$

$$= -\frac{f d_x n_x}{d_z n_y} - p_x \frac{n_x}{n_y} + p_x \frac{n_x}{n_y} - f\frac{n_z}{n_y} + p_y$$

$$= -\frac{n_x}{n_y}\left(f\frac{dx}{dz} + p_x\right) + p_x \frac{n_x}{n_y} - f\frac{n_z}{n_y} + p_y$$

$$V_y = -\frac{n_x}{n_y} V_x + p_x \frac{n_x}{n_y} - f\frac{n_z}{n_y} + p_y.$$

Therefore the vanishing points lie on the line $Y = \underbrace{-\frac{n_x}{n_y}}_{\text{slope}} X + \underbrace{p_x \frac{n_x}{n_y} - f\frac{n_z}{n_y} + p_y}_{\text{constant}}$

Q3. 1). Let l be $a_1x + b_1y + c_1 = 0$

l' be $a_2x + b_2y + c_2 = 0$.

Let these two line intersect.

$$a_2(a_1x + b_1y + c_1) = a_1(a_2x + b_2y + c_2).$$

$$a_1a_2x + a_2b_1y + a_2c_1 = a_1a_2x + a_1b_2y + a_1c_2$$

$$(a_1b_1 - a_1b_2)y = a_1c_2 - a_2c_1.$$

$$y = \frac{a_1c_2 - a_2c_1}{a_2b_1 - a_1b_2}.$$

$$b_2(a_1x + b_1y + c_1) = b_1(a_2x + b_2y + c_2)$$

$$a_1b_2x + b_2b_1y + b_2c_1 = a_2b_1x + b_1b_2y + b_1c_2$$

$$(a_1b_2 - a_2b_1)x = b_1c_2 - b_2c_1$$

$$x = \frac{b_1c_2 - b_2c_1}{a_1b_2 - a_2b_1}$$

Since $l \times l' = (a_1b_1c_1) \times (a_2b_2c_2)$

$$= \begin{pmatrix} b_1c_2 - b_2c_1 \\ a_2c_1 - a_1c_2 \\ a_1b_2 - a_2b_1 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{b_1c_2 - b_2c_1}{a_1b_2 - a_2b_1} \\ \frac{a_2c_1 - a_1c_2}{a_1b_2 - a_2b_1} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$$ which is the intersection

2) Let $p = (x_1, y_1)$ $p' = (x_2, y_2)$

the l pass both points: $\begin{cases} ax_1 + by_1 + c = 0 \\ ax_2 + by_2 + c = 0 \end{cases}$

$ax_1 + by_1 = ax_2 + by_2$.

$a(x_1 - x_2) = by_2 - by_1$

$a = \frac{by_2 - by_1}{x_1 - x_2}$

$\frac{by_2 - by_1}{x_1 - x_2} x_1 + by_1 + c = 0$.

$b = \frac{c(x_1 - x_2)}{x_2 y_1 - x_1 y_2}$.

$a = \frac{c(y_1 - y_2)}{x_1 y_2 - x_2 y_1}$

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} c(y_1 - y_2) \\ c(x_1 - x_2) \\ -c(x_2 y_1 - x_1 y_2) \end{pmatrix} = \begin{pmatrix} y_1 - y_2 \\ x_2 - x_1 \\ x_1 y_2 - x_2 y_1 \end{pmatrix}$$

Since $p \times p' = (x_1, y_1, 1) \times (x_2, y_2, 1)$

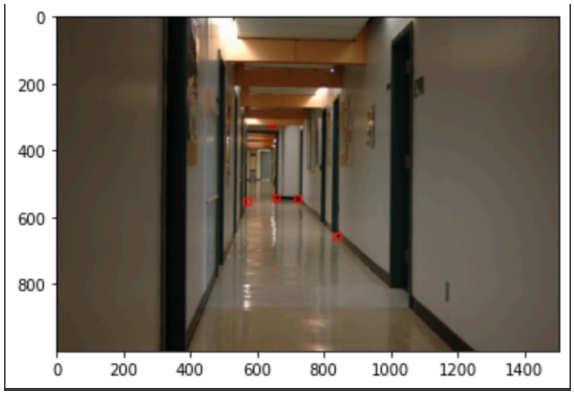
$= (y_1 - y_2, x_2 - x_1, x_1 y_2 - x_2 y_1)$

$= (a, b, c)$.

Therefore $l = p \times p'$

Q4. 1.

hallway 1 right wall

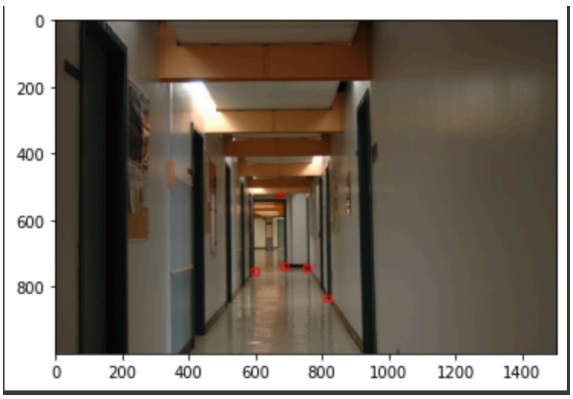


hallway 1 floor.

hallway 2 right wall

hallway 3 right wall

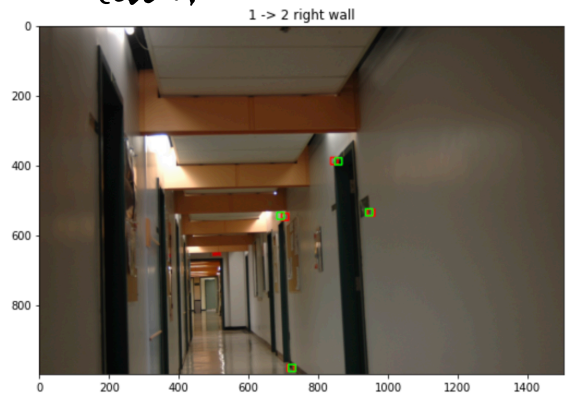
hallway3. floor.



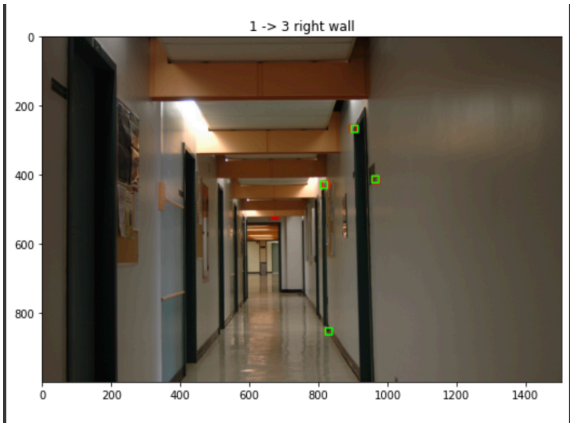
2. see CSC420A4Q4.ipynb.

3.

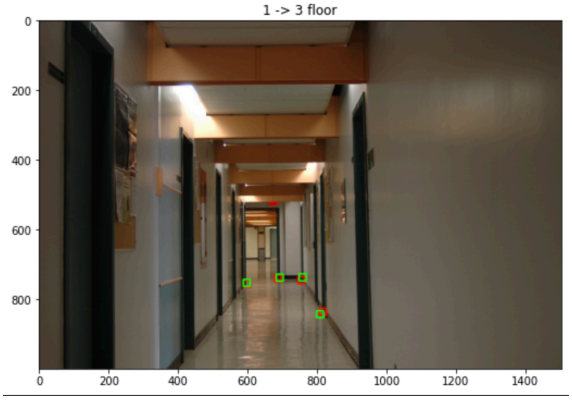
Case A



Case B

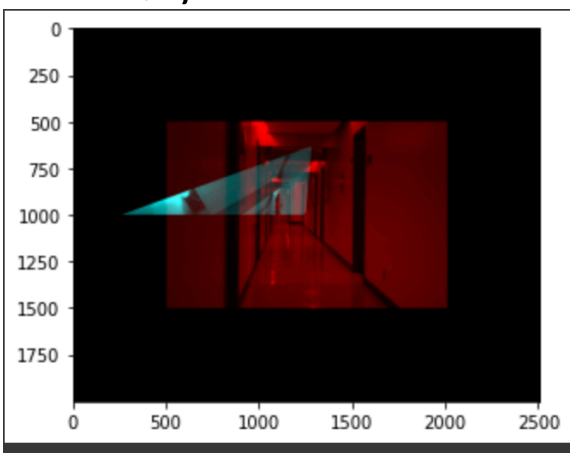


Case C

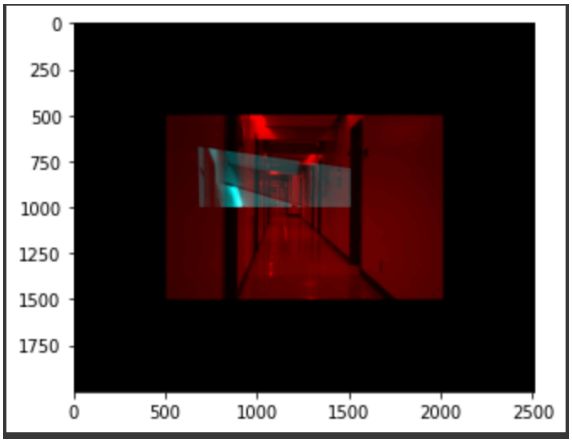


4.

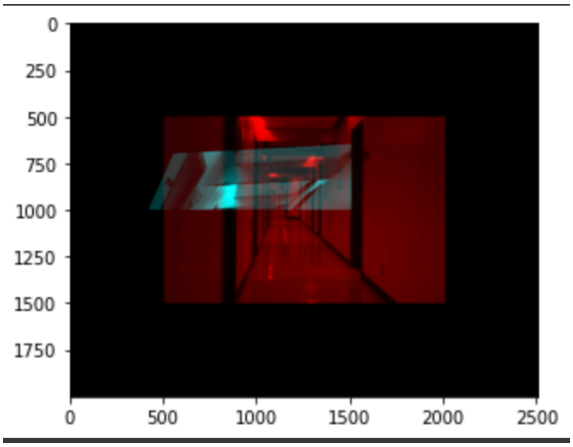
Case A



Case B



Case C.

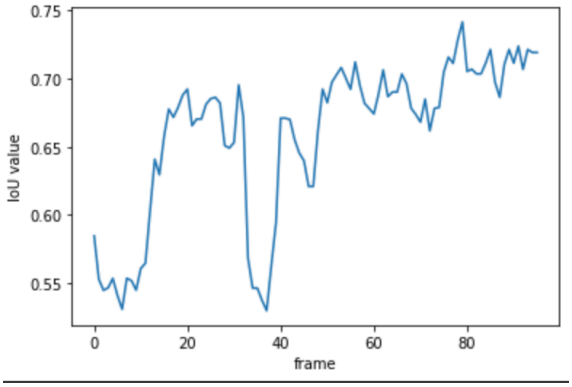


Conclusion: If the light go through the same point on lense. from reflection on an object, then they have the same 3D relative position no matter how the orientation of camera's angle move. The plane of one orientation can be projected on to anther without changing features..

switch case. also in CSC420A4Q4.ipynb.

```
# switch case
def run_case(case_num='A'):
  hallway1 = cv2.imread('/content/drive/MyDrive/Colab_Notebooks/hallway1.jpg')
  hallway2 = cv2.imread('/content/drive/MyDrive/Colab_Notebooks/hallway2.jpg')
  hallway3 = cv2.imread('/content/drive/MyDrive/Colab_Notebooks/hallway3.jpg')
  hallway1 = cv2.cvtColor(hallway1, cv2.COLOR_RGB2BGR)
  hallway2 = cv2.cvtColor(hallway2, cv2.COLOR_RGB2BGR)
  hallway3 = cv2.cvtColor(hallway3, cv2.COLOR_RGB2BGR)
  hallway1_right_wall = [[849, 228], [1001, 79], [1099, 230], [840, 661]]
  hallway2_right_wall = [[705, 549], [846, 389], [952, 537], [721, 982]]
  hallway3_right_wall = [[822, 428], [901, 266], [967, 416], [829, 853]]
  hallway1_floor = [[571, 554], [657, 544], [722, 546], [840, 658]]
  hallway3_floor = [[599, 755], [687, 740], [755, 745], [817, 836]]
  if case_num == 'A':
    H = homography(hallway1_right_wall, hallway2_right_wall)
    mapped_source_pts = map_func(hallway1_right_wall, H)
    hallway2_cpy = hallway2.copy()
    for coordinates in hallway2_right_wall:
      x, y = coordinates
      cv2.rectangle(hallway2_cpy, (x - 10, y - 10), (x + 10, y + 10), (255, 0, 0
```

Q5 see CSC420Q5. .ipynb.



number of frames vs IoU

- red is tracked box. green is detected box

```
greater_than_50 = 0
greater_than_60 = 0
for element in IoU_list:
  if element > 0.5:
    greater_than_50 += 1
  if element > 0.6:
    greater_than_60 += 1
print("The percentage of frames with iou > 50% is {}".format(greater_than_50 / len(IoU_list)))
print("The percentage of frames with iou > 60% is {}".format(greater_than_60 / len(IoU_list)))
```

```
The percentage of frames with iou > 50% is 1.0
The percentage of frames with iou > 60% is 0.9270833333333334
```

Same frame with low IoU (< 60%) since 10% has nothing.

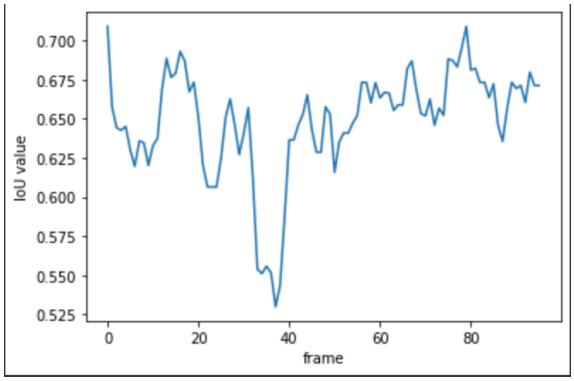
```
print(len(store_frames))
for frame in store_frames:
  plt.imshow(frame)
```

```
19
```

didn't find frame with iou under 10%, so I choose 60%

- The tracked box (red) is more often correct. It's capturing most part of the person's face. It might be possible that the majority of skin and hair is unchanged in this case. Which makes mean shift detect easier.

2)



number of frames vs IOU

```
greater_than_50 = 0
greater_than_60 = 0
for element in IoU_list:
  if element > 0.5:
    greater_than_50 += 1
  if element > 0.6:
    greater_than_60 += 1
print("The percentage of frames with iou > 50% is {\".format(greater_than_50 / len(IoU_list)))
print("The percentage of frames with iou > 60% is {}".format(greater_than_60 / len(IoU_list)))
```

```
The percentage of frames with iou > 50% is 1.0
The percentage of frames with iou > 60% is 0.8020833333333334
```