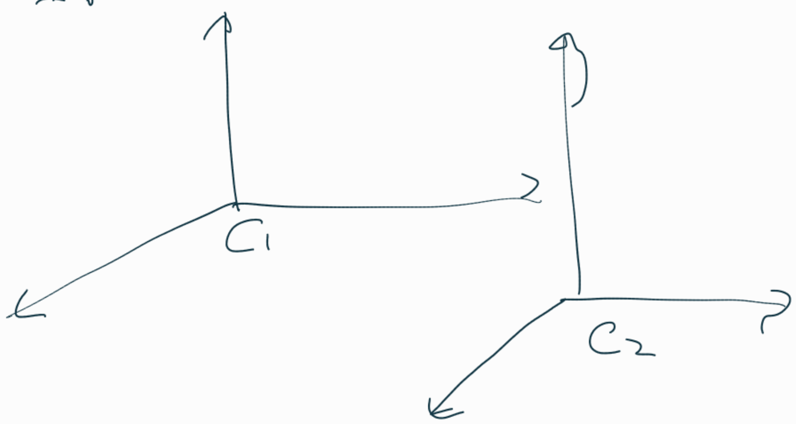
Question 1.

(1-1)



$R=I_3,\ T=[d,0,0]^T$ / $R=I_3,\ T=[0,d,0]^T$

$$X'=RX+T$$

$$T\times X' = T\times RX = [T]RX$$

$$X'\cdot(T\times X')=0 \rightarrow X'^T[T]RX=0$$

$\therefore$ for $E \rightarrow E=[T]R$

when x axis shift

$$E=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -d \\ 0 & d & 0 \end{bmatrix}$$

when y axis shift

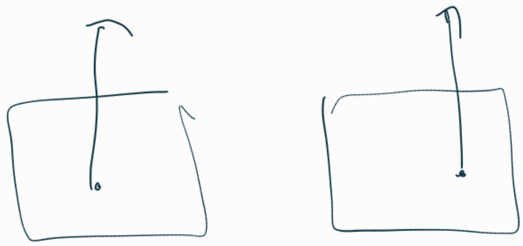
$$E=\begin{bmatrix} 0 & 0 & d \\ 0 & 0 & 0 \\ -d & 0 & 0 \end{bmatrix}$$

(1-2)

when assuming only the case of x-shift

(without losing generality)

we can see that Epipolar



since Epipolar contains baseline

$y = z = 0$

$\rightarrow ax + by + cz = 0$ } true for all $(x, 0, 0)$

$\rightarrow$ plane becomes

$\underline{by + cz = 0}$ $\rightarrow$ to find epipolar line

when $z = f$ $\rightarrow$ $y = -\frac{c}{b} f$

$\rightarrow$ $\therefore (x, -\frac{c}{b} f, f) \rightarrow$ epipolar line parallel to $(x, 0, 0)$

Question 2,

2.1

$x_3^T F_{13} x_1 = 0$

$x_3^T F_{23} x_2 = 0$

$\rightarrow$ $\underline{x_3 = k (F_{13} x_1) \times (F_{23} x_2)}$

$\therefore x_3 = k [F_{13} x_1] F_{23} x_2$

$= k F_{13} [x_1] F_{13}^T F_{23} x_2$

2.2 It's when

$F_{13} x_1$ and $F_{23} x_2$ is parallel

this is when to epipolar plane is same

when O_1, O_2, O_3 is collinear → epipolar plane

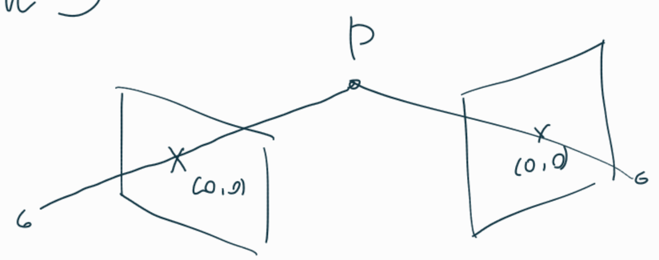
between 1,3 = 2,3

and when epipolar plane contains O_1, O_2, O_3 points

→ epipolar plane identical

∴ in both case unavailable

Question 3



$$\rightarrow \begin{pmatrix} 0 \\ 0 \\ f_1 \end{pmatrix}^T F \begin{pmatrix} 0 \\ 0 \\ f_2 \end{pmatrix} = 0$$

$$\rightarrow F_{33} f_1 f_2 = 0 \qquad \therefore F_{33} = 0$$

Question 4

(1) Aperture does not fully improvides information of motion. Since Apeture gives only part of informateons and thus we perceive motion ambiguously

(2) Aperture is necessary since we obtain the lack of information related to Apeture thats why we need Apeture. block of motion component

Question 5

Optical flow Eqn still holds through Brightness consistency, spatial coherence, and temporal persistance by temporal persistance $u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$

$$\frac{dx}{dt}\frac{\partial I}{\partial x} + \frac{dy}{dt}\frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = \frac{dI}{dt} = 0$$ (Since I is consistent.)

$I_x u + I_y v + I_t = 0$ $\therefore$ still holds true

when camera moves forward $\rightarrow u = v = 0$

$\therefore I_t = 0$ is the point

because objects moves toward homothetic center but center remains not moving

Question 6

Answer will be difficult case.

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -I_x I_t \\ -I_x I_y \end{bmatrix}$$

At the part inside the Apeture

It won't be easy to detect the tangatial movement if no corner exists.

for example for simple case let's think of linear line and this line will be composed with n-different pixels with same gradient

$\sum I_x^2 = n I_x^2$ $\quad \sum I_x I_y = n I_x I_y$

$\sum I_y^2 = n I_y^2$

→ then det of matrix $= 0$ which leaves the solution undetermined.

which only gives solution for normal axis and leaves no clue for tangential action

∴ it is difficult.

Writeup 2020-11120 Gene Chung

#1. Lucas-Kanade Method

In the implementation, the first thing I've done was to delete the template where no response of the warping is valid. First, I defined the warping matrix as from the textbook, and followed the step designed in textbook. In algorithm I added two for sentences and made all important calculation inside to minimize the operating time. And used functions from numpy to make code efficient such as np.where. The main progress follows step below.

- 1. Warp $\boldsymbol{I}$ with $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})$ to $I(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))$
- 2. Compute error image $T(\boldsymbol{x}) - I(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))$
- 3. Warp gradient of $\boldsymbol{I}$ to compute ∇I
- 4. Evaluate Jacobian $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$
- 5. Compute Hessian $\boldsymbol{H} = \sum_{\boldsymbol{x}} [\nabla I \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}]^T [\nabla I \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}]$
- 6. Compute $\Delta \boldsymbol{p}$ $\Delta \boldsymbol{p} = \boldsymbol{H}^{-1} \sum_{\boldsymbol{x}} [\nabla I \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}]^T [T(\boldsymbol{x}) - I(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))]$
- 7. Update parameters $\boldsymbol{p} \leftarrow \boldsymbol{p} + \Delta \boldsymbol{p}$ 

First determine the template where valid response value exist, and then by the affine matrix, warp gradient and image into new plane then, calculate the jacobian matrix then, calculate the hessian matrix. And successfully obtain the gradient values. However, when using inverse matrix of hessian some errors of singular matrix occurs by the initial value of gradient defined at 1 so I used pinv rather inv to make the code successfully work for the case of singular hessian matrix occurs.

#2. Affine motion Subtraction

First, Normalized the image and its gradient by dividing each with 255, and maximum value to make it between 0 to 1 for stability during the computation. Then I defined the affine matrix and obtained dp to add in p. The same algroithm of #1 was used in getting values for #2 but, in this case, the interpolation and warping was tested backward. By subtracting the image into warping image, we can obtain the moving image, and then denormalized the values to get the answer successfully.