

Question 7

(1) x 축에 대해 평행한 경우, 두 카메라 center 사이의
거리 d 이라 하면 $T = [-d \ 0 \ 0]^T$ 이고 $R = I_3$ 이다.

$$\text{cf2M } E = [T_x] R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{bmatrix} \text{이다.}$$

y 축에 대해 평행한 경우, 두 카메라 center 사이의
거리 d 이라 하면 $T = [0 \ -d \ 0]^T$ 이고 $R = I_3$ 이다.

$$\text{cf2M } E = [T_y] R = \begin{bmatrix} 0 & 0 & -d \\ 0 & 0 & 0 \\ d & 0 & 0 \end{bmatrix} \text{이다.}$$

(2)

translational image plane all parallel 한 경우
rotation \rightarrow image plane all perpendicular 하면
baseline의 intersection은 infinity이다. cf2M
epipoles 은 infinity이고 epipolar line은
axis of translation에 평행하다.

Question 2

(1) $x_3^T F_{13} x_1 = 0, x_3^T F_{23} x_2 = 0$ ($\because F$ 가 Fundamental matrix) 이고 $e_1 = [1 \ 0 \ 0]^T, e_2 = [0 \ 1 \ 0]^T, e_3 = [0 \ 0 \ 1]^T$ 이라 하자.

$$x_3 = [x \ y \ z]^T$$

$$F_{13} x_1 = \begin{bmatrix} e_1 F_{13} x_1 \\ e_2 F_{13} x_1 \\ e_3 F_{13} x_1 \end{bmatrix}, F_{23} x_2 = \begin{bmatrix} e_1 F_{23} x_2 \\ e_2 F_{23} x_2 \\ e_3 F_{23} x_2 \end{bmatrix}$$

이제

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

$$e_1 F_{13} x_1 x + e_2 F_{13} x_1 y + e_3 F_{13} x_1 z = 0$$

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

$$e_1 F_{23} x_2 x + e_2 F_{23} x_2 y + e_3 F_{23} x_2 z = 0$$

(2)

$$e_1 F_{13} x_1 d + e_2 F_{13} x_1 y + e_3 F_{13} x_1 = 0$$

$$e_1 F_{23} x_2 d + e_2 F_{23} x_2 y + e_3 F_{23} x_2 = 0$$

두 방정식은 직선의 방정식으로 펼쳐놓기

않다면 (x, y) 인 x_3 을 구할 수 있으나

두 직선이 평행하거나 일치하는 경우 x_3 을
구할 수 없다.

Question 3

두 corresponding points x 와 x' 에 대해 $x'^T F x = 0$ 가 성립한다.

이때 $x = x' = (0, 0, 1)^T$ 이다.

$$(0 \ 0 \ 1) \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = F_{33} = 0$$

이고 $\frac{\partial}{\partial} F_{33} = 0$ 이다.

Question 4

- (1) Small aperture의 motion direction이 ambiguous 한 이유는 motion Component가 텁행한 선은 visual input 으로부터 추론할 수 없기 때문이다.
- (2) aperture를 이용해 물체를 보면 motion을 잘 못 인식할 수 있 으므로 다른 부분에서의 image로부터의 motion signal이 필요하다. 전체적으로 motion을 보면 모든 움직임이 파악 가능하므로 aperture problem에는 aperture가 필수적이다.

Question 5

Optical flow equation은 여전히 hold 된다. 이는 실제 움직임이 아니라 pixel의 움직임에 관련 있기 때문이다.

만약 camera components의 velocity를 안다면, 관측된 points의 velocity component를 뺀다면 어떠한 scene point에서 object 가 앞으로 움직이는지 알아낼 수 있다.

Question 6

$$\begin{bmatrix} \sum_{x,y} I_x^2 & \sum_{x,y} I_x I_y \\ \sum_{x,y} I_y I_x & \sum_{x,y} I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_{x,y} I_x (T(x,y) - I(x,y)) \\ \sum_{x,y} I_y (T(x,y) - I(x,y)) \end{bmatrix}$$

lukas-kanade algorithm에 의해 위 식에서
 $\begin{bmatrix} u \\ v \end{bmatrix}$ 을 계산하면 location을 구할 수 있다.

이때 선분의 끝에 집중하여 coordinate of
T(x, y)와 warping된 image에서의
위치인 I(x, y)를 쉽게 계산할 수 있다.
따라서 optical flow의 계산이 어렵지 않을
것이다.

Homework 4

컴퓨터공학부

2017-18538

황선영

3. programming

part 1: Lucas-Kanade Method

먼저 image2를 warping한다. 좌표를 RectBivariateSpline 함수를 이용해 interpolate하고 각 point에서의 spline을 evaluate 하는 방식이다. 그리고 gradient vector를 계산하고 아래의 식에 따라 jacobian matrix를 계산한다.

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$$

그리고 hessian matrix를 아래의 식에 따라 계산한다.

$$\mathbf{H} = \sum_{\mathbf{x}} [\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}]^T [\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}]$$

그 후 최종적으로 delta p를 계산한다. 이때의 식은

$$\Delta \mathbf{p} = \mathbf{H}^{-1} \sum_{\mathbf{x}} [\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}]^T [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))]$$

이다. 그 후 delta p를 반환한다.

part 2: Affine Motion Subtraction

먼저 part 1의 lucas kanade affine 함수를 이용하여 p를 계산한다. 그리고 image2를 앞서 계산한 p를 이용하여 warping한다. 좌표를 RectBivariateSpline 함수를 이용해 interpolate하고 각 point에서의 spline을 evaluate 하는 방식이다. 그리고 hysteresis threshold를 적용하여 반환한다. 이때 high threshold는 $0.2 * 256$ 이고 low threshold는 $0.15 * 256$ 이다.