

Homework #4
Question #7

Assume 2 cameras have the same K . & no relative rotation between the cameras
 $R=I$.

2 situations depending on what direction to translate.

(i) only translation along x axis
 $T = (T_x, 0, 0)$

$$E = [T_x] R = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -T_x \\ 0 & T_x & 0 \end{pmatrix}$$

7-2. WLOG

Suppose only a translation along x axis

$$P = (x, y, f)$$

$$P' = (x', y', f)$$

$$P^T E P = 0$$

$$\Leftrightarrow (x' y' f) \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -T_x \\ 0 & T_x & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ f \end{pmatrix} = 0$$

$$\Leftrightarrow y = y' \text{ for any } x. \quad \text{즉, } y \text{는 } x \text{에 대해 일정한 직선.}$$

$$\Leftrightarrow \text{epipolar line } P - P' = \begin{pmatrix} x - x' \\ 0 \\ 0 \end{pmatrix} \text{은 } T \text{ translation 방향과 평행한 직선.}$$

$$\Leftrightarrow P - P' \parallel T$$

Question #2

$$x_3^T F_{13} x_1 = 0 \quad (1) \Leftrightarrow x_3 \perp F_{13} x_1$$

$$x_3^T F_{23} x_2 = 0 \quad (2) \Leftrightarrow x_3 \perp F_{23} x_2$$

$$F_{13} x_1 = \begin{pmatrix} e_1^T F_{13} x_1 \\ e_2^T F_{13} x_1 \\ e_3^T F_{13} x_1 \end{pmatrix}$$

$$F_{23} x_2 = \begin{pmatrix} e_1^T F_{23} x_2 \\ e_2^T F_{23} x_2 \\ e_3^T F_{23} x_2 \end{pmatrix}$$

to be more specific,
 (3차원 분수로 정의)

$$\text{Let } x_3 = (x_3, y_3, 1)$$

(1) 과 (2)의 조건에 따라 다음이 성립한다

$$\begin{pmatrix} e_1^T F_{13} x_1 & e_2^T F_{13} x_1 & e_3^T F_{13} x_1 \\ e_1^T F_{23} x_2 & e_2^T F_{23} x_2 & e_3^T F_{23} x_2 \end{pmatrix} \begin{pmatrix} x_3 \\ y_3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

7-2. $(x_3, y_3, 1)$ 은 x_3 과 x_2 의 임계 $(x_3, y_3, 1)$ 이 결정되지 않은 것.

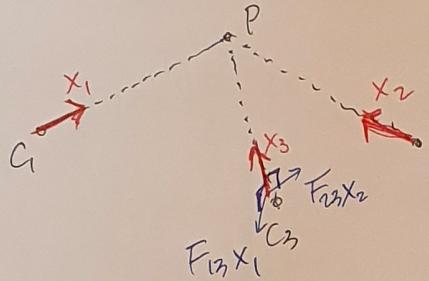
\Leftrightarrow 광학적 이거나 영상 없는 서로 평행한 평면의 방정식.

\Leftrightarrow 평면의 방정식의 벡터 $F_{13} x_1$ 과 $F_{23} x_2$ 가 평행, $F_{13} x_1 \parallel F_{23} x_2$.

2-2 epipolar geometry에서

$$x_3 \perp F_{13}x_1$$

$$x_3 \perp F_{23}x_2$$



$F_{13}x_1$ 과 $F_{23}x_2$ 가 직교이므로 각도 이어짐

x_3 는 $(F_{13}x_1) \times (F_{23}x_2)$ 로 결정될 수 있다.

하지만 두 벡터가 평행하거나 일치할 경우 외적이 ~~0~~로 결정되지 않게 된다. x_3 또한 관계로 구할 수 없게 된다.

Question #3

For two corresponding points P and P' , $P'^T E P = 0$ 가 성립된다.

증명: 성장에 따라 $P = P' = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

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

$\therefore F_{33} = 0$

Question #4

4-1 While motion component in the gradient direction can be inferred based on input, the component parallel to the line (perpendicular to the gradient) cannot be inferred.

⇒ aperture problem (ambiguity)

4-2 aperture 이 없으면 사물의 일정 부분이 어느 방향으로 움직이는지 알 수 없다 (모든 재학각의 움직임)
따라서 크기로 줄여 aperture 부분의 motion 방향을 알아내어야 전체 이미지의 각 부분의 motion 방향을 알 수 있다는 점에서 중요하다. (하지만依然 aperture problem은 gradient 방향 만큼 강지만 가능성을 multiple constraints를 가지고 결정짓는다.)

Question #5

이미지의 높은 pixel에 대해 optical flow는 성립하지 않는다. optical flow equation이 2가지 assumptions을 따르아 한다.

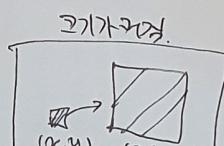
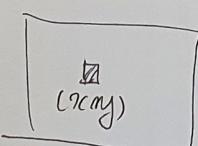
(i) color constancy

카메라에 물체가 더 가까이 갈수록

물체에 대비되는 배경을 희석을 차지하는

또는 빛의 brightness와 더 대기 matching이 힘들다.

즉, (x, y, t) 와 $(x+u\Delta t, y+v\Delta t, t+\Delta t)$ 의 brightness는 확대된 색상 중



(ii) small motion (color)

5 (color) (d) small motion

동안 물체가 매우 균형하게 움직였을 때 물체의 위치는 속도 (물체가 움직이는 속도)는 거의
반비례하게 움직이므로 (증가하는) true constancy 가정을 할 수 있게 된다. $O \rightarrow O \rightarrow O$

물체가 어느 scene point에 다가가는지 확인하기 위해 사용하는 aperture approximation
각각의 지점에 gradient로 구해 역방향으로 연장했을 때의 경로를 추적한다.
예로 들어 물체가 다가온다고 가정하면. 시간이 지남에 따라 물체의 보정의 optical flow는
여행경로의 경로의 경로가 되어 scene point가 된다.

Question #6

$$\begin{pmatrix} \sum I_x^2 \\ \sum I_y I_x \\ \sum I_x I_y \\ \sum I_y^2 \end{pmatrix} (u) = \begin{pmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{pmatrix}$$

하지만 이런 물체의 보정이 aperture에 속하는지에 따라 (물체 / 꼭짓점 / 면 내부)
(u, v)를 결정하기 어렵다. \Leftrightarrow A의 역행렬을 구하기 어렵다.

$$\Leftrightarrow \det A = 0 \text{ 일 경우}$$

문제 상황을 간단히 표현하기 위해 다음과 같은 image 상황을 가정하자

$$I_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$I_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$I_1 \rightarrow I_2$

우리는 쉽게 우측 상방의 방향으로 물체가 움직인 것을 알 수 있다. $(u, v) = (1, -1)$
쉽게 알 수 있는 이유는 우리가 선분상의 지점만 있는 것이 아니라 꼭짓점이 있어 꼭짓점 이동 방향을 통해
선을 찾는 데 도움이다. ($\det A \neq 0$ 온 결정하는 순간 가능하다)

I_1 에 대해서 I_x, I_y 를 구해보자. 평면상 tuple (ptr)의 3개로 표현하면 (I_x, I_y)
초 두 가지 형태의 성분으로 나눌 수 있다

(1) 꼭짓점이 아닌 선분 주위의 픽셀 $(\pm 1, \mp 1)$
(2) 꼭짓점 주위의 픽셀 $(0, \mp 1)$ or $(\mp 1, 0)$

(1)의 경우 $\begin{pmatrix} I_x^2 & I_y I_x \\ I_x I_y & I_y^2 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$

(2)의 경우 $\begin{pmatrix} I_x^2 & I_y I_x \\ I_x I_y & I_y^2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$

이를 통해

(1) 만일 경우 \det 는 0이므로 (u, v) 를 결정하기 힘들지만 (2)가 가능해
optical flow estimation이 필요하다. \rightarrow 소집

Part 1: Lucas–Kanade Method

The Lucas–Kanade Algorithm

Iterate:

- (1) Warp I with $\mathbf{W}(\mathbf{x}; \mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$
- (2) Compute the error image $T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$
- (3) Warp the gradient ∇I with $\mathbf{W}(\mathbf{x}; \mathbf{p})$
- (4) Evaluate the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $(\mathbf{x}; \mathbf{p})$
- (5) Compute the steepest descent images $\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
- (6) Compute the Hessian matrix using Equation (11)
- (7) Compute $\sum_{\mathbf{x}} [\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}]^T [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))]$
- (8) Compute $\Delta \mathbf{p}$ using Equation (10)
- (9) Update the parameters $\mathbf{p} \leftarrow \mathbf{p} + \Delta \mathbf{p}$

until $\|\Delta \mathbf{p}\| \leq \epsilon$

([Lucas-Kanade 20 Years On: A Unifying Framework: Part 1](#) 참고)

위와 같은 알고리즘으로 구현하였다. (1) 먼저 RectBivariateSpline을 이용하여 img2에 대해 image warping을 시켜 주었다. affine transformation을 위해서 아래와 같이 warping matrix를 구성하였다.

$$\mathbf{W}(\mathbf{x}; \mathbf{p}) = \begin{pmatrix} (1 + p_1) \cdot x & + & p_3 \cdot y & + & p_5 \\ p_2 \cdot x & + & (1 + p_4) \cdot y & + & p_6 \end{pmatrix} = \begin{pmatrix} 1 + p_1 & p_3 & p_5 \\ p_2 & 1 + p_4 & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

(2) error image를 warping된 이미지와 target 이미지와의 차이를 통해 구하고
 (3) Gx와 Gy를 flatten한 값을 가지고 새로운 행렬을 만들어 (5) steepest descent를
 만들어간다. 이 때 필요한 행렬인 (4) 야코비안 행렬은 다음과 같이 구성한다.

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \begin{pmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{pmatrix}$$

(6) gradient에 야코비안 행렬을 곱한 행렬을 이용해 Hessian을 구성한다.

(8) \mathbf{p} 를 업데이트하기 위해 \mathbf{p} 의 변화량을 계산하고 (9) 업데이트를 한다.

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

Part 2: Affine Motion Subtraction

"All 9 steps of the algorithm must be repeated in every iteration because the estimates of the parameters \mathbf{p} vary from iteration to iteration".

위의 $\Delta \mathbf{p}$ 의 L2 norm을 이용한 수렴 조건을 통해서 정확한 값의 \mathbf{p} 벡터를 구할 수 있을 때까지 반복을 해주었으며, 이때 수렴을 하기 위한 hyperparameter로 epsilon=1, converging_rate=100을 설정하고 실행을 시켜주었다.

다시 말해서 $\Delta \mathbf{p}$ 가 더 이상 변화하지 않을 때 (수렴할 때)까지 lucas_kanade_affine을 호출하였다. 이때까지 구한 \mathbf{p} 를 가지고 img2를 새롭게 warping 시키고 warping한 이미지와

img1을 비교하여 변화한 부분을 절대값을 취하고 apply_hysteresis_threshold 함수를 호출하여 움직이는 부분을 아래와 같이 표시한다.

