Introduction to Computer Vision S21 Assignment #1

2018-10457 이준서

1. 2D Fourier Transform of Images

먼저 imread로 img1.jpg와 img2.jpg를 읽어서 img1, img2에 저장하고 img2를 img1의 size에 맞춰 resize 해준다. 다음으로 img를 gray색상으로 바꾸기 위해서 mat2gray를 이용해서 img1_gray, img2_gray에 저장해주었다. 이를 통해 원래 image matrix의 data type인 uint8에서 double type으로 변환되었다(img1_gray. img2_gray). 그 다음으로 fft2와 fftshift를 거쳐서 Fourier transform 해준 것의 absolute value와 angle을 구하여 image의 magnitude와 phase를 찾아주었다. 한편, 서로 magnitude와 phase를 바꾼 image를 생성했더니 다음과 같은 결과가 나왔는데, 특징적인 것은 보이는 사진은 magnitude의 image와 비슷했지만 배경이 각각의 phase에 따라 물결처럼 요동치는 듯한 image가 만들어졌다.

2. Perspective Image Transforms

2D의 image를 우리가 원하는 평면에 projection 하였을 때 어떻게 나올지를 주어진 projective transform matrix를 이용하여 확인하는 것이 문제의 목표였다. 주어진 식인 $[p\ q\ r]^T = \mathbf{M}[x\ y\ 1]^T$을 이용하여 x, y(이 문제에서 x=1,2, … ,375, y=1,2, … ,500)에 대한 p, q를 matrix multiplication을 이용하여 구한 다음, for문을 이용하여 각각의 (x, y) 좌표를 (u, v) 좌표에 대응시켰다. 하지만 이런 forwarding warping은 projected image에 많은 hole이 생기는데, 이를 막기 위해서 동일한 size의 projected image matrix를 만들고 각각의 projected image matrix의 index를 for문으로 돌아가면서 $\mathbf{M}^{-1}[p\ q\ r]^T = [x\ y\ 1]^T$ 을 사용하여 backward warping으로 채워주었더니 hole를 제거할 수 있었지만, projected image의 해상도가 좋지 않았다. 이런 문제는 여러가지 interpolation method들을 이용하여 어느정도 해결할 수 있을 것이다.

3. Camera Calibration

a) 마지막 페이지에 증명 과정 수록하였습니다.

b) (a)의 증명을 통해 $\mathbf{x} = \mathbf{PX}$ 에서 projection matrix P를 구하려고 할 때 이 식을 $\mathbf{Ap} = \mathbf{0}$ ($\mathbf{p}$: vectorized form of $\mathbf{P}$) 의 형태로 바꾸어서 A의 SVD를 구하였을 때 V의 마지막 column vector가 P가 된다는 사실을 알았으므로, 다음과 같은 형태로 A를 [0 0 0 0]과 world_points, image_points를 이용하여 for 문을 통해 concatentate하여 만들어준 다음, Matlab의 svd를 이용하여 V를 구하고 최종적인 p를 추출하여 3X4 matrix P1(SVD method, P2- pseudo inverse method)을 만든 다음, m_{34}로 나누어 normalize 해주었다.

- Method 1 – homogeneous linear system. Solve for m's entries using linear least squares

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 & -v_1 \\ & & & & & & \vdots & & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n & -v_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \\ m_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

c) (b)와 다르게 이번에는 $m_{34} = 1$로 놓고 $\mathbf{x} = \mathbf{PX}$를 $\mathbf{Ap} = \mathbf{b} \rightarrow \mathbf{p} = \mathbf{A}^{+}\mathbf{b}$ ($\mathbf{p}$: vectorized form of $\mathbf{P}$) 형태로 바꾸어 p를 계산하였는데, 여기서 A는 (b)에서의 A에서 마지막 column vector만 빠졌고, $A^{+} = (A^{T}A)^{-1}A^{T}$ 이다.

- Method 2 – nonhomogeneous linear system. Solve for m's entries using linear least squares

Ax=b form

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1X_1 & -v_1Y_1 & -v_1Z_1 \\ & & & & & & \vdots & & & & \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nZ_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_nX_n & -v_nY_n & -v_nZ_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \\ m_{31} \\ m_{32} \\ m_{33} \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ \vdots \\ u_n \\ v_n \end{bmatrix}$$

(b)와 (c)의 projection matrix P의 값을 비교해보면 조금의 오차는 있었지만 상당히 비슷했다.

3. (a)

$A = U\Sigma V^T$, U, V: orthonormal, Σ: diagonal matrix, $\operatorname{rank}(A) = k$, $k \le n$ 이라가정.

$U = \begin{bmatrix} u_1 & u_2 & \cdots & u_m \end{bmatrix}$, $V = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}$

$\Sigma = \begin{bmatrix} \sigma_1 & 0 & & \\ 0 & \sigma_2 & \ddots & \\ \vdots & 0 & & \\ 0 & 0 & & \sigma_n \end{bmatrix}$ ($m \times n$)

cf. $\|v_i\| = 1$, $\sigma_1^2 \ge \sigma_2^2 \ge \cdots \ge \sigma_k^2$, $\sigma_i = 0$ ($\forall i > k$)

$A^TA = V\Sigma^TU^TU\Sigma V^T$

$= V\Sigma^T\Sigma V^T = V\Sigma^2V^T$ ... ①

$A^TAV = V\Sigma^2$

(u_i, v_i : column vector) $\Rightarrow A^TAv_i = \sigma_i^2 v_i$ ... ②

$p = a_1v_1 + a_2v_2 + \cdots + a_nv_n$, $\forall p \in \mathbb{R}^n$, $\|p\| = 1$ 이라하면,

$\Rightarrow a_1^2 + a_2^2 + \cdots + a_n^2 = 1$ ... ③

$\|Ap\|^2 = \|U\Sigma V^Tp\|^2 = (U\Sigma V^Tp)^T(U\Sigma V^Tp) = p^TV\Sigma^T\cancel{U^TU}\Sigma V^Tp$

$= p^TV\Sigma^T\Sigma V^Tp = p^TV\Sigma^2V^Tp = p^TA^TAp$ (∵ ①)

$= p^TA^TA(a_1v_1 + \cdots + a_nv_n) = p^T(a_1\sigma_1^2v_1 + \cdots + a_n\sigma_n^2v_n)$ (∵ ②)

$= (a_1v_1^T + \cdots + a_nv_n^T)(a_1\sigma_1^2v_1 + \cdots + a_n\sigma_k^2v_k)$

$= a_1^2\sigma_1^2 + a_2^2\sigma_2^2 + \cdots + a_k^2\sigma_k^2$

$\sigma_1^2 \ge \sigma_2^2 \ge \cdots \ge \sigma_k^2$ 이므로

$\|Ax\|^2 = a_1^2\sigma_1^2 + \cdots + a_k^2\sigma_k^2 \ge (a_1^2 + \cdots + a_k^2)\sigma_k^2 = \sigma_k^2 = \sigma_k^2 U_k^TV_k$

$= V_k^TA^TAV_k$ (∵ ②)

$= (AV_k)^T(AV_k)$

$= \|AV_k\|^2$

$\therefore$ $\|p\| = 1$, $\forall p \in \mathbb{R}^n$ 인 p 에 대해 $\|Ap\|^2 \ge \|AV_k\|^2$

$\therefore$ V_k 는 least square solution 이다. ∎