Multiview geometry EE667
Homework 1

20213569 정병인

Problem 1.

가) Problem1, Problem2 에서 사용한 방법에 대하여 설명하라.

Problem1 은 Two-step rectification 으로, 먼저 projective distortion 을 없애고, 이후 affine distortion 을 없애는 과정을 거치게 된다. Projective distortion 의 경우 이미지에서 평행한 직선 두 쌍이 만나는 점인 vanishing point 를 이은 line at infinity 를 찾는다. 이 때 평행한 직선 두 쌍은 hough transformation 을 이용하여 찾는다. Image 에서의 line at infinity 가 real-world 에서 line at infinity 인 $l = [0,0,1]$ 이 되도록 mapping 하는 projective matrix H_p를 구한다. 임의의 image 에서 line at infinity 가 $l = [l_1, l_2, l_3]$라면, H_p는 다음과 같다.

$$H_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix} \qquad where \quad l_{line\ \mathrm{at}\ infinity} = [l_1, l_2, l_3] \qquad (1)$$

이 H_p를 homogenous coordinate 으로 표현된 camera pixel 들에 곱하여 얻은 새로운 pixel 들의 좌표로 image 를 표현하면, projective distortion 이 제거된 rectified image 를 얻을 수 있다. 이 과정은 ***MATLAB*** 의 image processing toolbox 내부 imwarp 를 이용하였다.

다음 step 으로는 affine distortion 을 없애는 과정이다. 이 과정에서는 2 쌍의 서로 수직한 직선을 찾은 후 dual conic 의 아래와 같은 성질을 이용하여 affine matrix 를 찾을 수 있다.

$$l_1 m_1 + l_2 m_2 = l^T C_\infty^* m = 0 \qquad (2)$$

위 식에서 l과 m은 서로 수직 관계이다. 여기서 affine matrix H_a를 식 (3)과 같다고 가정하면, dual conic 의 성질인 식(4)에 의해 식 (6)을 이끌어 낼 수 있다..

$$H_a = \begin{bmatrix} A & t \\ 0 & 1 \end{bmatrix} \qquad (3)$$

$$C_\infty^{*\prime} = H_a C_\infty^* H_a^T, \quad l^T C_\infty^* m = 0 \qquad (4)$$

$$l' = H_a^{-1} l \qquad (5)$$

$$\begin{aligned} l^T C_\infty^* m &= l'^T H_a H_a^{-1} C_\infty^{*\prime} H_a^{-T} H_a^T m' = l'^T H_a C_\infty^* H_a^T m' \\ &= l'^T H_a \begin{bmatrix} I & 0 \\ 0 & 1 \end{bmatrix} H_a^T m' = l'^T \begin{bmatrix} AA^T & 0 \\ 0 & 0 \end{bmatrix} m' = 0 \end{aligned} \qquad (6)$$

식 (6)에서 AA^T는 다음과 같이 구할 수 있다.

$$\begin{aligned} &[l_1', l_2'] AA^T [m_1', m_2']^T = [l_1', l_2'] \begin{bmatrix} s_1 & s_2 \\ s_2 & 1 \end{bmatrix} [m_1', m_2']^T = 0 \\ &where \quad l' = [l_1', l_2', l_3'], \quad m' = [m_1', m_2', m_3'] \end{aligned} \qquad (7)$$

위 식을 정리하면,

$$[l_1'm_1' \quad l_1'm_2'+l_2'm_1' \quad l_2'm_2']\begin{bmatrix}s_1\\s_2\\1\end{bmatrix}=0 \qquad (8)$$

따라서 위 식을 풀기 위해서는 2 쌍의 orthogonal line 이 필요하다. 이 때 두 쌍의 orthogonal line 들 사이의 관계에 평행이 있다면 결국 한 개의 식과 같으므로, hough transform 으로 찾은 두 쌍의 평행선들이 real-world 에서는 정사각형을 이루고 있다고 가정하고, 정사각형의 대각선을 서로 길게 연결하여 추가적인 orthogonal line 을 만든다.

위 과정을 통해 얻은 AA^T에 singular value decomposition(SVD)를 적용하여 얻은 결과가 $AA^T=UDV^T$라면, $A=U\sqrt{D}V^T$임을 이용하여, affine matrix H_a를 구할 수 있다. Projective distortion 을 없앤 것과 마찬가지로 image processing toolbox 에 있는 imwarp 를 이용하여 affine distortion 이 rectify 된 이미지를 얻을 수 있다.

Problem 2 는 One-step rectification 이다. One step rectification 에서는 dual conic 의 성질을 이용하여, 한 번에 projective distortion 과 affine distortion 을 없앨 수 있는 방법을 사용한다. Dual conic 은 similarity matrix H_s에 대해서 다음 식이 성립한다.

$$C^*_{\infty,image}=H_sC^*_\infty H_s^T \qquad (9)$$

위 식에서 $C^*_{\infty,image}$는 image 에서의 dual conic 을 말한다. 즉, 두 dual conic 이 similarity matrix 에도 변하지 않으므로, image 에서의 dual conic 을 구하면 projective distortion 과 affine distortion 을 제거할 수 있다. 여기서, C^*_∞는 다음과 같다.

$$C^*_\infty=\begin{bmatrix}1&0&0\\0&1&0\\0&0&0\end{bmatrix} \qquad (10)$$

식 (2)와 식 (4)에 의해, 아래와 같은 식이 성립하게 된다.

$$l^TC^*_\infty m=l'^TH_aH_a^{-1}C^{*\prime}_\infty H_a^{-T}H_a^Tm'=l'^TC^{*\prime}_\infty m'=0 \qquad (11)$$

위 식에서, 우리는 $C^{*\prime}_\infty$ 을 아래와 같이 둘 수 있다.

$$C^{*\prime}_\infty=\begin{bmatrix}a&b/2&d/2\\b/2&c&e/2\\d/2&e/2&1\end{bmatrix} \qquad (12)$$

식 (7)에서의 조건과 식 (12)를 식 (11)에 대입하면 아래의 관계를 얻을 수 있다.

$$l'^T C_\infty^{*\prime} m' = [l_1', l_2', l_3'] \begin{bmatrix} a & \frac{b}{2} & \frac{d}{2} \\ \frac{b}{2} & c & \frac{e}{2} \\ \frac{d}{2} & \frac{e}{2} & f \end{bmatrix} [m_1', m_2', m_3']^T$$
$$= \left[l_1' m_1', \frac{l_1' m_2' + l_2' m_1'}{2}, l_2' m_2', \frac{l_1' m_3' + l_3' m_1'}{2}, \frac{l_2' m_3' + l_3' m_2'}{2}, l_3' m_3' \right] [a, b, c, d, e, f]^T = 0 \quad (13)$$

위 식에서 f 는 scale 로 인해 무시될 수 있으므로, 5 개의 orthogonal line 을 구할 수 있다면 dual conic 을 구할 수 있다.

Dual conic 에 singular value decomposition 을 적용하면,

$$C_\infty^{*\prime} = UDU^T = Udiag(\sqrt{\lambda_1}, \sqrt{\lambda_2}, s) diag(1,1,0) diag(\sqrt{\lambda_1}, \sqrt{\lambda_2}, s) U^T$$
$$= Udiag(\sqrt{\lambda_1}, \sqrt{\lambda_2}, s) C_\infty^* diag(\sqrt{\lambda_1}, \sqrt{\lambda_2}, s) U^T = HC_\infty^* H^T \quad (14)$$

를 만족한다. 즉, $H = Udiag(\sqrt{\lambda_1}, \sqrt{\lambda_2}, s)$ 임을 알 수 있다.

위 방법을 이용하여 구한 H로 image 를 rectification 한다면, projective distortion 과 affine distortion 이 제거된 image 를 얻을 수 있다.

위 과정을 진행하면서 distortion 이 제거됨에 따라 원래 이미지의 좌측 상단((1,1) 지점)에 약간의 translation 이 일어나는데, Problem 1 의 projective line 의 경우는 차이가 거의 없어 이를 보정해주지 않았지만 affine distortion 이 제거된 이후의 image 에서는 그 차이가 상당하여 이를 보정해주었다.

나) Problem1, Problem2 의 결과를 영상과 함께 설명하라.

Image rectification 을 적용하기 위한 2 장의 image 를 pix3d dataset[1]에서 얻었다. 2 장의 이미지 모두 bookcase image 로, 평행한 직선이 많이 포함된 물체이기에 rectification 을 적용하기에 적합한 image 이다. 해당 image 두 장은 아래와 같다.

(a) (b)

Figure 1. Sample Images

다음으로, vanishing point 를 찾기 위해, real-world 에서 parallel line 을 찾는 방법으로 Hough transformation 을 이용한 직선 검출 방법을 선택하였다.

Hough transformation 은 이미지 내에서 직선 성분을 찾아주는 것으로, 몇몇 threshold 를 조정함으로써 위와 같이 얻을 수 있었다. Hough transformation 을 행하기 위해 불러온 image 를 gray scale 로 변경을 하고, canny edge detection 을 이용하여 image 의 canny 를 먼저 구하였다. 이렇게 구해진 이미지는 아래와 같다.

Figure 2. Canny Edge

이후 MATLAB 에서 지원하는 Image Processing Toolbox[2]를 이용하여 hough transformation 을 이용한 line 검출을 할 수 있다. houghpeaks 함수의 경우 threshold 를 hough transform matrix 에서 가장 큰 값의 0.3 배에서 올림을 한 것을 기준으로 5 개의 maximum point 를 찾고, 이후 houghlines 에서는 FillGap 을 5 로 두어, parallel 한 선분이 같은 곳에 위치하지 않도록 방지하였다.

MinLength 의 경우에는, 두 이미지의 크기가 서로 달라, 첫번째 image 에는 100 을, 두번째 image 의 경우 200 으로 설정하였다. 그 결과는 아래와 같다.

Figure 3. Find parallel lines using Hough transformation

Figure 3 에 표현된 직선은, 시작점의 좌표, 끝나는 점의 좌표, θ와 ρ로 표현이 된다. ρ와 θ는 각각 원점에서 직선까지의 거리, x 축(horizontal axis)와 ρ vector 사이의 각도를 의미한다. 이러한 θ와 ρ를 이용하여 아래와 같이 선들을 plot 할 수 있다. 이 과정에 필요한 식은 아래와 같이 표현할 수 있으며, 각 값을 이용해 선들의 vector 를 구할 수 있다. plot 과정에서 사용에 적합하다고 판단한 2 쌍의 parallel lines 를 각 이미지마다 찾을 수 있었다.

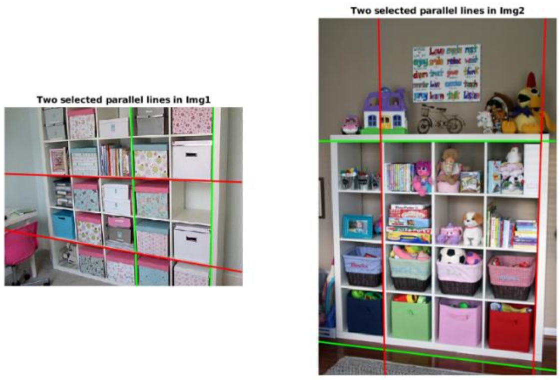

Figure 4. Plot parallel lines on images

서로 평행한 선은 같은 색(적색, 녹색)으로 표현하였다. projective distortion 을 없앤 결과는 아래와 같다.

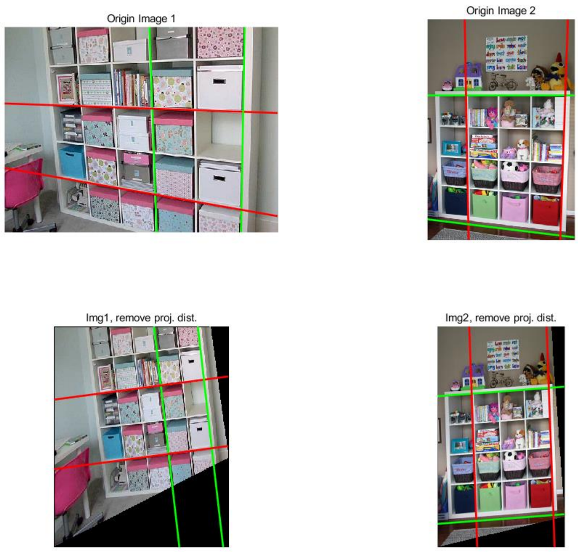

Figure 5. Remove projective distortion

다음 단계는 affine distortion 을 없애는 것이다. 이를 위해선 추가적인 orthogonal line 이 필요하여, Figure 5 의 적색/녹색 선이 이루는 평행사변형이 real-world 에서는 정사각형이라 가정하고, 해당 정사각형의 대각선을 푸른색 선으로 표현하였다.

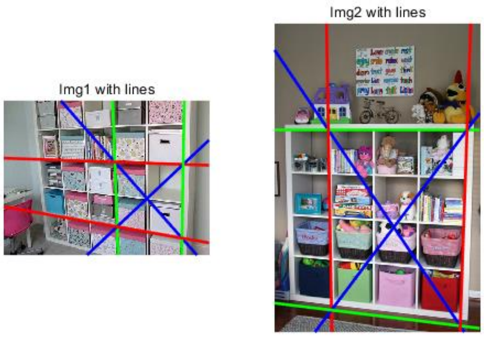

Figure 6. Add one more orthogonal line

위 그림에서 식 (7)과 식 (8)을 통해 affine matrix H_a를 구할 수 있고, imwarp 를 이용하여 transformation 을 수행하면 아래 그림을 얻을 수 있다.

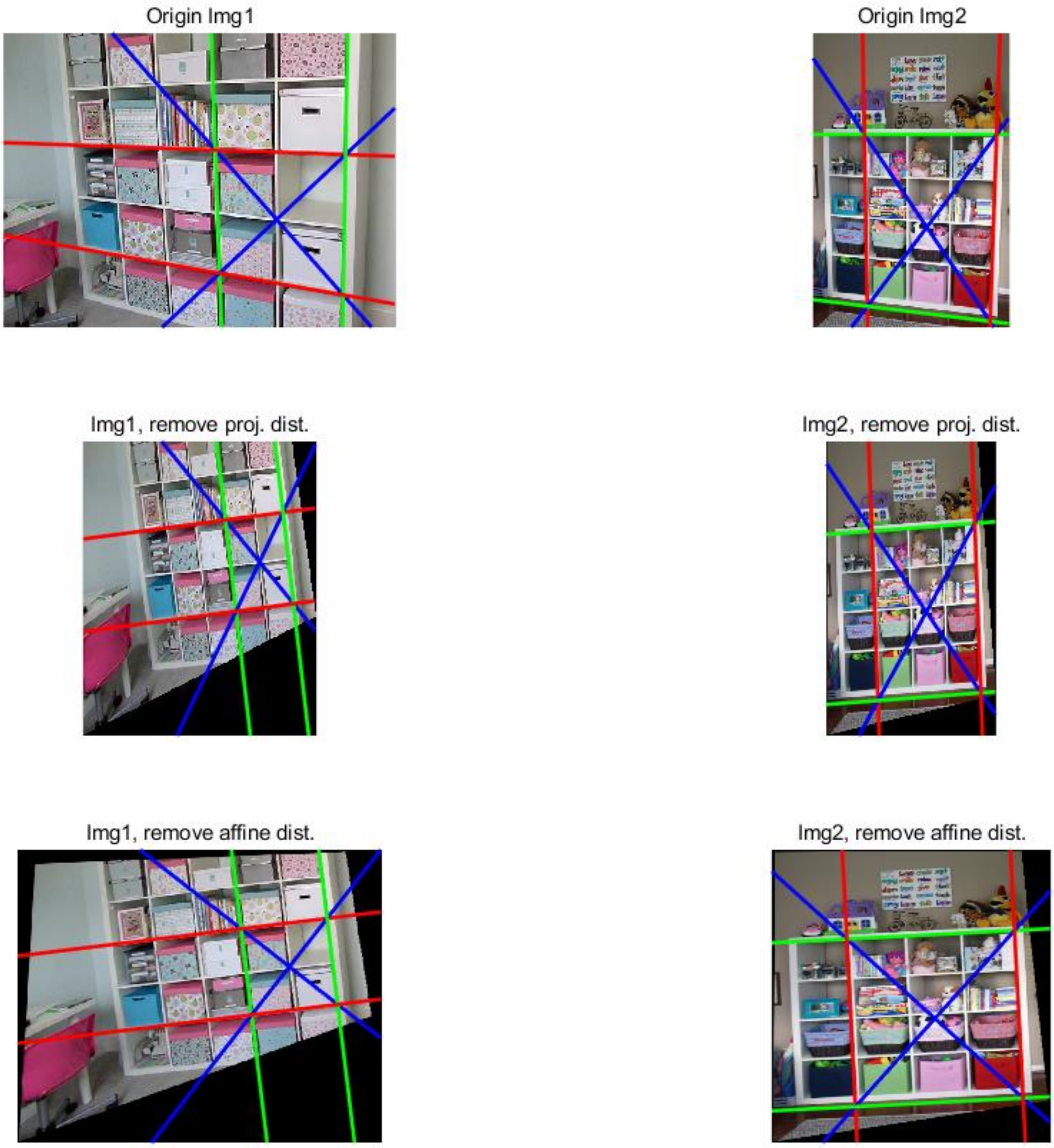

Figure 7. Origin images and Rectified images For Two-step rectification

One-step rectification 은 Two-step rectification 에서 얻은 2 쌍의 parallel lines 와 한 쌍의 orthogonal line 을 이용하여 총 5 종류의 orthogonal line 을 얻을 수 있었다. 이를 이용하여 식 (13), 식 (14)를 풀고 얻은 결과로 rectification 을 수행하면 다음과 같다.

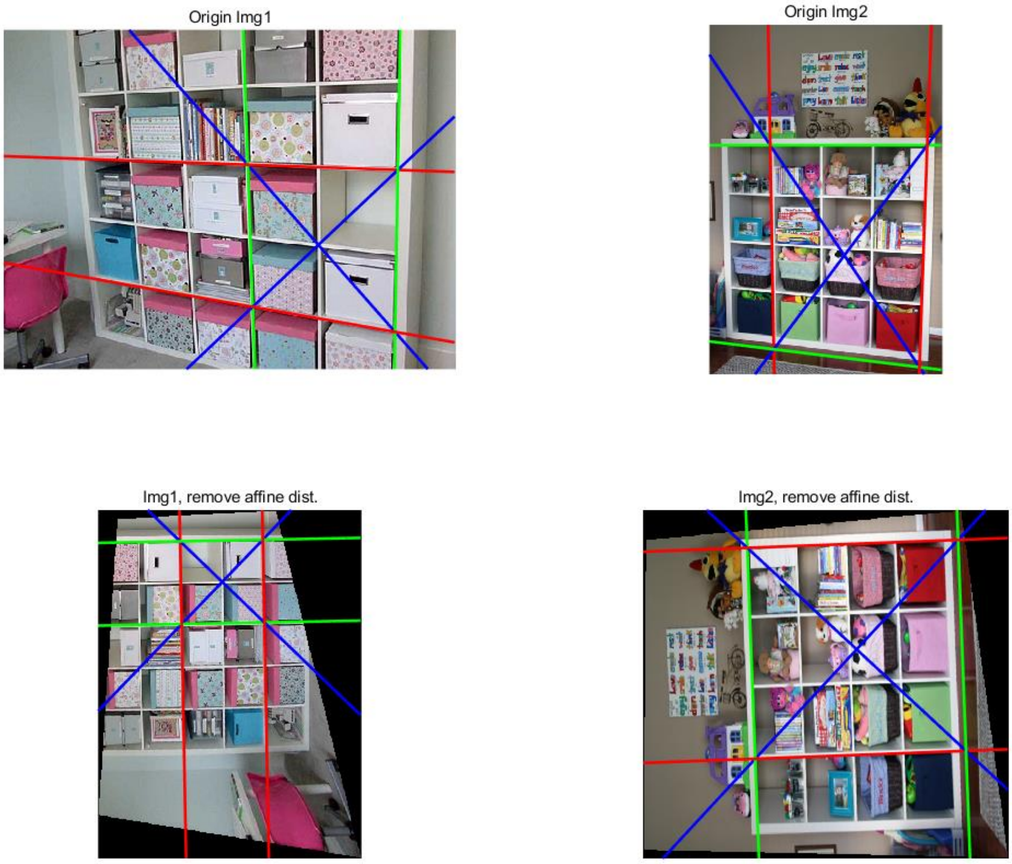

Figure 8. Origin images and Rectified images For One-step rectification

다) Problem 2 의 One-step Rectification 이 불안정한 이유에 대하여 논하라.

One-step rectification 을 수행하는 데에 있어 가장 중요한 것은 image 상의 dual conic C_∞^*를 구하는 것이다. 5 개의 선택된 orthogonal line 들로부터 구하게 되는데, 이 때 마지막 eigenvalue 는 0 이 아니라 0 에 가까운 아주 작은 수가 나오게 된다. 이러한 값에 의해 우리는 real-world 에서의 dual conic 이 diag(1,1,0)이 아닌, diag(1,1,s)의 값을 가지게 된다.(s 는 0 에 가까운 아주 작은 수) 이로 인해 불안정한 rectification 이 진행이 되고, 이로 인해 Figure 8 과 Figure 9 의 rectified image 가 서로 다른 각도로 보이는 결과를 유발했다고 생각할 수 있다.

라) Reference

[1] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2974–2983, 2018