I. Multiple View Geometry hw3

student: 임규범, student No.: 2018-22219

A. Problem 1

(i): 월드 평면 π를 $\pi = (\mathbf{n}^T, d)^T$라고 하면 평면 위의 점 $\mathbf{X}$이 존재하는 경우 다음 공식이 성립한다.

$$\mathbf{n}^T \tilde{\mathbf{X}} + d = 0 \tag{1}$$

이를 다시 정리하면 $-\frac{\mathbf{n}^T \tilde{\mathbf{X}}}{d} = 1$이 된다. 따라서 $\mathbf{X}$를 이미지 평면 상에 프로젝션한 점 $\mathbf{x}$는 다음과 같이 나타낼 수 있다.

$$\begin{aligned}
\mathbf{x} = \mathbf{P}\mathbf{X} &= \mathbf{P}\begin{bmatrix}\tilde{\mathbf{X}} \\ 1\end{bmatrix} \\
&= \mathbf{K}\mathbf{R}\tilde{\mathbf{X}} + \mathbf{t}(-\frac{\mathbf{n}^T \tilde{\mathbf{X}}}{d}) \\
&= \mathbf{K}(\mathbf{R} - \frac{\mathbf{t}\mathbf{n}^T}{d})\tilde{\mathbf{X}}
\end{aligned} \tag{2}$$

따라서 $\mathbf{x} = \mathbf{H}\tilde{\mathbf{X}}$에서 $\mathbf{H} = \mathbf{K}(\mathbf{R} - \frac{\mathbf{t}\mathbf{n}^T}{d})$가 된다.

(iv): Homogeneous Coordinate로 나타낸 algebraic surface는 $\mathbf{F}^n(\mathbf{X}) = 0$을 만족한다. 이 때 $\mathbf{F}^n(\cdot)$은 n차 다항식의 형태를 가진다. Algebraic surface의 차수는 surface에 포함되어 있지 않으면서 surface와 교차하는 직선들의 개수와 동일하다. 또한 algebraic surface는 접평면들이 존재하므로 algebraic surface의 차수는 surface를 접하는 접평면의 차수와 동일하다.

$\nabla\mathbf{F}$는 $\mathbf{F}$의 기울기를 의미하므로 접평면과 수직인 특성을 지닌다. 접평면이 이루는 커브 Contour generator는 반드시 카메라의 중심점을 지나므로 다음 공식이 성립한다.

$$\mathbf{C} \cdot \nabla\mathbf{F} = 0 \tag{3}$$

이 때, $\mathbf{F}$의 자유도가 n이고 접평면은 n-1의 자유도를 가진다. Bézout 정리에 의해 apparent contour γ는 n(n-1)의 자유도를 가진다.

Bézout's theorem은 같은 평면에 존재하는 Curve(n dof), Surface(m dof)가 있을 때 둘 사이에는 총 **nm**개의 교점이 생성된다는 정리이다. 즉 **nm**개의 자유도를 가진다는 의미와 동치이다.

reference: link1, link2

(ix): $\mathbf{H}^T\mathbf{l} = (0,0,1)^T$이고 이를 $\mathbf{H} = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}$이면 이를 풀어서 정리하면 다음과 같다.

$$\mathbf{H}^{-T}\mathbf{l} = \mathbf{K}^{-1}\mathbf{R}\mathbf{K}^T\mathbf{l} \tag{4}$$

이 때 Scene 평면의 방향벡터 $\mathbf{n}$을 $\mathbf{n} = \mathbf{K}^T\mathbf{l}$이라고 하면 $\mathbf{l} = \mathbf{K}^{-1}\mathbf{n}$이므로 이를 위 식에 대입하고 정리하면 다음과 같다.

$$\begin{aligned}
\mathbf{H}^{-T}\mathbf{l} &= \mathbf{K}^{-1}\mathbf{R}\mathbf{K}^T\mathbf{l} \\
&= \mathbf{K}^{-1}\mathbf{R}\mathbf{K}^T\mathbf{K}^{-T}\mathbf{n} \\
&= \mathbf{K}^{-T}\mathbf{R}\mathbf{n}
\end{aligned} \tag{5}$$

여기서 $\mathbf{R} = \begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{n}\end{bmatrix}$이므로 $\mathbf{R}\mathbf{n} = (0,0,1)^T$이 된다. $\mathbf{K}^{-T}$는 scale에만 영향을 미치므로 생략하여 표현하면

$$\therefore \mathbf{H}^T\mathbf{l} = \mathbf{R}\mathbf{n} \tag{6}$$

(xi): 소실선 $\mathbf{l}$과 실제 3차원 공간에서 균일하게 떨어져 있는 직선을 이미지 평면에 프로젝션한 직선 $\mathbf{l}_i,\ i = 1,2,3,\cdots$에 대해 다음과 같은 공식이 성립한다.

$$\begin{aligned} \mathbf{l} &= \alpha \mathbf{l}_1 + \beta \mathbf{l}_2 \\ \mathbf{l}_n &= \mathbf{l}_0 + n\mathbf{l}. \end{aligned} \tag{7}$$

α, β 값을 구하기 위해 우선 $\mathbf{l}_1, \mathbf{l}_2$를 구하면

$$\begin{aligned} \mathbf{l}_1 &= \mathbf{l}_0 + \mathbf{l} \\ \mathbf{l}_2 &= \mathbf{l}_0 + 2\mathbf{l} \end{aligned} \tag{8}$$

$\mathbf{l} = \alpha \mathbf{l}_1 + \beta \mathbf{l}_2$를 위 식에 대입하면

$$\begin{aligned} \mathbf{l}_1 &= \mathbf{l}_0 + \alpha \mathbf{l}_1 + \beta \mathbf{l}_2 \\ \mathbf{l}_2 &= \mathbf{l}_0 + 2(\alpha \mathbf{l}_1 + \beta \mathbf{l}_2) \end{aligned} \tag{9}$$

식 9에서 윗쪽식 양변에 2를 곱한 후 $\mathbf{l}_1$을 외적한 다음 $(\mathbf{l}_1 \times \mathbf{l}_2)$를 양변에 내적하면

$$0 = 2(\mathbf{l}_0 \times \mathbf{l}_1)^T (\mathbf{l}_1 \times \mathbf{l}_2) - 2\beta (\mathbf{l}_2 \times \mathbf{l}_1)^T (\mathbf{l}_2 \times \mathbf{l}_1) \tag{10}$$

이 때, $2(\mathbf{l}_2 \times \mathbf{l}_1)^T (\mathbf{l}_2 \times \mathbf{l}_1)$는 스칼라값이므로 $K_\beta = 2(\mathbf{l}_2 \times \mathbf{l}_1)^T (\mathbf{l}_2 \times \mathbf{l}_1)$로 치환하면 β는 다음과 같이 구해진다.

$$\beta = -2(\mathbf{l}_0 \times \mathbf{l}_1)^T (\mathbf{l}_1 \times \mathbf{2})/K_\beta \tag{11}$$

마찬가지로 식 9에서 아래쪽 공식에 양변에 $\mathbf{l}_2$를 외적하고 $(\mathbf{l}_2 \times \mathbf{l}_1)$을 내적하면

$$0 = (\mathbf{l}_0 \times \mathbf{l}_2)^T (\mathbf{l}_1 \times \mathbf{l}_2) + 2\alpha (\mathbf{l}_1 \times \mathbf{l}_2)^T (\mathbf{l}_1 \times \mathbf{l}_2) \tag{12}$$

이 때, $K_\alpha = 2(\mathbf{l}_1 \times \mathbf{l}_2)^T (\mathbf{l}_1 \times \mathbf{l}_2)$로 치환하고 정리하면 α는 다음과 같이 구할 수 있다.

$$\alpha = -(\mathbf{l}_0 \times \mathbf{l}_2)^T (\mathbf{l}_1 \times \mathbf{l}_2)/K_\alpha \tag{13}$$

α, β값을 통해 소실선 $\mathbf{l}$을 다시 표현해보면 다음과 같다.

$$\therefore \mathbf{l} = ((\mathbf{l}_0 \times \mathbf{l}_2)^T (\mathbf{l}_1 \times \mathbf{l}_2))\mathbf{l}_1 + 2((\mathbf{l}_0 \times \mathbf{l}_1)^T (\mathbf{l}_2 \times \mathbf{l}_1))\mathbf{l}_2 \tag{14}$$

이 때, Scale 값인 -1, K_α, K_β는 모두 소거해준다.

(xiv): 3차원 공간 상에서 Quadratic의 방정식은 다음과 같다.

$$Ax^2 + By^2 + Cz^2 + 2Dxy + 2Eyz + 2Fxz + 2Gx + 2Hy + 2Jz + K = 0 \tag{15}$$

Quadratic을 행렬 $\mathbf{Q}$로 표현하면 다음과 같다.

$$\mathbf{Q} = \begin{bmatrix} A & D & F & G \\ D & B & E & H \\ F & E & C & J \\ G & H & J & K \end{bmatrix} \tag{16}$$

Quadratic이 Sphere인 경우에는 원점 (a_1, a_2, a_3)이고 반지름 r에 대하여 다음과 같이 간결하게 나타낼 수 있다.

$$\begin{aligned} \mathbf{Q} &= \begin{bmatrix} 1 & & & -a_1 \\ & 1 & & -a_2 \\ & & 1 & -a_3 \\ -a_1 & -a_2 & -a_3 & r \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I}_3 & -\mathbf{a} \\ -\mathbf{a} & r \end{bmatrix} \end{aligned} \tag{17}$$

Sphere $\mathbf{Q}$ 위의 점 $\mathbf{X}$에 대해 다음과 같은 공식이 성립한다.

$$\mathbf{X}^T \mathbf{Q} \mathbf{X} = 0 \tag{18}$$

Sphere $\mathbf{Q}$의 Dual Quadratic을 $\mathbf{Q}^*$라고 하고 Sphere $\mathbf{Q}$를 이미지 평면에 프로젝션한 Conic $\mathbf{C}$와 Dual Conic $\mathbf{C}^*$가 있다고 하면

$$\mathbf{C}^* = \mathbf{P}\mathbf{Q}^*\mathbf{P}^T \tag{19}$$

가 성립하고 이 때, $\mathbf{C}^* = \mathbf{C}^{-1},\ \mathbf{Q}^* = \mathbf{Q}^{-1}$이라고 하면

$$\mathbf{C}^{-1} = \mathbf{P}\mathbf{Q}^{-1}\mathbf{P}^T \tag{20}$$

가 성립한다. 즉 이미지 평면 상의 Conic $\mathbf{C} = (\mathbf{P}^T)^\dagger\mathbf{Q}\mathbf{P}^\dagger$으로 표현할 수 있다. 카메라 프로젝션 행렬 $\mathbf{P} = \mathbf{K}[\mathbf{I}|0]$이고 $(\mathbf{R} = \mathbf{I}, \mathbf{t} = 0)$ Squared Pixel 조건을 만족하는 경우

$$\mathbf{P} = \begin{bmatrix} f & & c_x & 0 \\ & f & c_y & 0 \\ & & 1 & 0 \end{bmatrix} \tag{21}$$

으로 나타낼 수 있고 $\mathbf{P}$는 역행렬이 존재하지 않으므로 Pseudo Inverse $\mathbf{P}^\dagger = \mathbf{P}^T(\mathbf{P}\mathbf{P}^T)^{-1}$를 계산해보면

$$\mathbf{P}^\dagger = \begin{bmatrix} f^{-1} & 0 & -c_x f^{-1} \\ 0 & f^{-1} & -c_y f^{-1} \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{22}$$

이 된다. 따라서 Conic $\mathbf{C}$는

$$\begin{aligned} \therefore \mathbf{C} &= (\mathbf{P}^T)^\dagger\mathbf{Q}\mathbf{P}^\dagger \\ &= \begin{bmatrix} f^{-2} & 0 & -c_x f^{-2} \\ 0 & f^{-2} & -c_y f^{-2} \\ -c_x f^{-2} & -c_y f^{-2} & 1 + (c_x^2 + c_y^2) f^{-2} \end{bmatrix} \end{aligned} \tag{23}$$

가 된다. Conic $\mathbf{C}$ 위의 점을 $\mathbf{x}$라고 하면 다음의 공식이 성립한다.

$$\mathbf{x}^T\mathbf{C}\mathbf{x} = 0 \tag{24}$$

이 때 $\mathbf{C} = \begin{bmatrix} a & 0 & b/2 \\ 0 & a & c/2 \\ b/2 & c/2 & d \end{bmatrix}$로 치환하고 위 공식을 풀어서 정리하면

$$\begin{aligned} 0 &= \mathbf{x}^T\mathbf{C}\mathbf{x} \\ &= \begin{bmatrix} a & b & 1 \end{bmatrix} \begin{bmatrix} a & 0 & b/2 \\ 0 & a & c/2 \\ b/2 & c/2 & d \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} ax + b/2 & ay + c/2 & bx/2 + cy/2 + d \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ &= ax^2 + bx/2 + ay^2 + cy/2 + bx/2 + cy/2 + d = 0 \\ &= ax^2 + ay^2 + bx + cy + d = 0 \\ &= f^{-2}x^2 + f^{-2}y^2 - 2c_x f^{-2}x - 2c_y f^{-2}y + 1 + (c_x^2 + c_y^2)^2 = 0 \\ &= \left(\frac{x - c_x}{f}\right)^2 + \left(\frac{y - c_y}{f}\right)^2 + 1 = 0 \end{aligned} \tag{25}$$

따라서 Sphere의 이미지 프로젝션된 Conic $\mathbf{C}$는 Sphere의 위치에 관계없이 이미지 센터(c_x, c_y)와 반지름의 크기(f)만 알면 내부 파라미터 $\mathbf{K}$를 구할 수 있다. 그리고 Sphere의 위치에 관계없이 Conic이 결정되므로 Ellipse의 비율 또한 Sphere의 위치와 독립적이다.

B. Problem 2

소실선을 활용한 Metric Rectification을 수행하기 위해서 스마트폰을 사용하여 이미지를 촬영하였다. 소실점 $\mathbf{v}_1, \mathbf{v}_2$를 찾기 위해 실제 3차원 공간에서 수평선 점 4개($p_i,\ i=1,2,3,4$)를 미리 알고있다고 가정하였다. 사용한 이미지는 아래와 같다.

rectification_using_vanishing_points.m 파일은 다음과 같다.

```
image =  imread('test.jpg');

% four points
p1 = [1032,50,1];
p2 = [2678,876,1];
p3 = [928,2119,1];
p4 = [2791,2003,1];

% four lines
L1 = cross(p1,p2);
L2 = cross(p3,p4);
L3 = cross(p1,p3);
L4 = cross(p2,p4);

%v1 = vanishing point of lines #1 and #2
v1 = transpose(null([L1; L2]));
v1=[v1(1)/v1(3), v1(2)/v1(3),1];

%v2 = vanishing point of lines #3 and #4
v2 = transpose(null([L3; L4]));
v2=[v2(1)/v2(3), v2(2)/v2(3),1];

% vanishing line of v1 and v2
vanishing_line = null([v1; v2]);
vanishing_line = vanishing_line / vanishing_line(3);

% find projective transform
H = [1 0 0 ;
    0 1 0 ;
    transpose(vanishing_line)];

Hv1 = (H*v1')';
Hv1 = Hv1 / sqrt(Hv1(1)*Hv1(1) + Hv1(2)*Hv1(2));

Hv2 = (H*v2')';
Hv2 = Hv2 / sqrt(Hv2(1)*Hv2(1) + Hv2(2)*Hv2(2));

% find directions corresponding to vanishing points.
dir = [Hv1(1), -Hv1(1), Hv2(1), -Hv2(1) ;
       Hv1(2), -Hv1(2), Hv2(2), -Hv2(2)];

thetas = [atan2(dir(1,1),dir(2,1)),
          atan2(dir(1,2),dir(2,2)),
          atan2(dir(1,3),dir(2,3)),
          atan2(dir(1,4),dir(2,4))];

% find direction closest to vertical axis.
% and find the positive angle the rest for the horizontal axis.
[none,vidx] = max([thetas(1), thetas(2)]);
[none,hidx] = min([thetas(3), thetas(4)]);
hidx = hidx + 2;

Hmetric = [dir(1,vidx), dir(1,hidx), 0 ;
           dir(2,vidx), dir(2,hidx), 0 ;
           0 0 1];

% remove reflection
if det(Hmetric) < 0
    Hmetric(:,1) = -Hmetric(:,1);
```

```
end

load cameraCalibrator.mat;
K = cameraParams.IntrinsicMatrix';

% apply homography and crop.
metric_img = applyHomography(image, inv(K*Hmetric)*H);
metric_img_crop = imcrop(metric_img, [0 1500 2000 2000]);
imshow(metric_img_crop)

% save image.
imwrite(metric_img_crop, 'test_warped.jpg');
```

위 스크립트를 사용하여 Metric Rectification된 이미지는 다음과 같다.