EECS 442 F14
HW3
WU Tongshuang 40782356

1.

a) Points on a line in 3-space through the point A and with direction $D = (d^T, 0)^T$:

With rotation, $d = Rd$

Points can be written as $x(\lambda) = PX(\lambda) = PA + \lambda PD = a + \lambda KRd$

Where a is the image of A. Then the vanishing point v of the line is obtained as the limit:

$$v = \lim_{\lambda \to \infty}(\lambda) = \lim_{\lambda \to \infty}(a + \lambda KRd) = KRd$$

Where a is the image of A. Then the vanishing point v of the line is obtained

b) Since rotation matrix

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y \\ 0 & 1 & 0 \\ \sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\det(R) = \det(R_x) \cdot \det(R_y) \cdot \det(R_z)$$

$$\det(R) = 1 \cdot 1 \cdot 1 = 1$$

Thus, the R^{-1} exist.

Since for camera matrix $K = \begin{bmatrix} fk_u & fk_u cot\theta & u_0 \\ 0 & \frac{fk_v}{sin\theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$

$det(K) = \frac{fk_u fk_v}{sin\theta} = \frac{f^2 k_u k_v}{sin\theta} \neq 0, K^{-1}$ exists.

$$d = R^{-1}K^{-1}v$$

c) Since there's no rotation, R = $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and K, the calibration matrix remain the same

$d_1 = K^{-1}v_1, d_2 = K^{-1}v_2, d_3 = K^{-1}v_3$

Since $d_1^T d_2 = 0$, $d_1^T d_3 = 0$, $d_2^T d_3 = 0$,

For $i, j \in \{1, 2, 3\}$ and $i \neq j$

$$(K^{-1}v_i)^T (K^{-1}v_j) = 0$$

2.

a)

1)

To find the focal length and the camera center, we need to derive the intrinsic matrix for the camera, K. Algorithm computing K is shown as the following:

- Since the camera has zero-skew and square matrix, IAC takes the form:

$$\omega = \begin{bmatrix} w_1 & 0 & w_2 \\ 0 & w_1 & w_3 \\ w_2 & w_3 & w_4 \end{bmatrix}$$

- Represent ω as a homogeneous 6-vector $w = (w_1, w_2, w_3, w_4)^T$
- Get three vanishing points that correspond to three orthogonal lines: v_1, v_2 and v_3
- Three more constraints:

$$v_1^T w v_2 = 0$$
$$v_1^T w v_3 = 0$$
$$v_2^T w v_3 = 0$$

- The constraints from the three pairs of vanishing points are stacked together to form an equation $Aw = 0$, where A is a 3×4 matrix, with each column equal to $(u_1v_1 + u_2v_2, u_1 + v_1, u_2 + v_2, 1)$
- Compute w, the null space of A, using SVD.
- $\omega = (KK^T)^{-1}$, so K can be obtained by Cholesky factorization of ω, followed by inversion.

The computed K

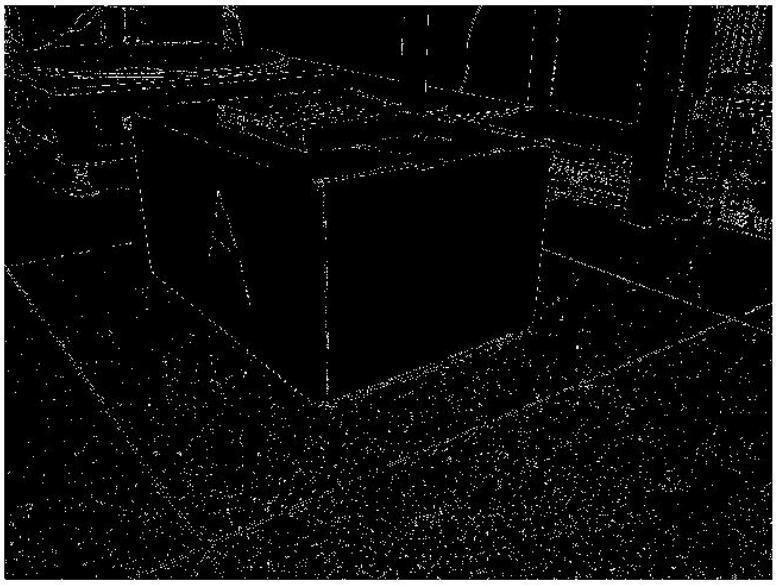
$$K = \begin{bmatrix} 2496.4 & 0 & 939.8 \\ 0 & 2494.6 & 956.1 \\ 0 & 0 & 1 \end{bmatrix}$$

Therefore the focal length is approximately 2496.4, and the camera center is (939, 956).

2) No. In order to use the constraint $v_1^T w v_2 = 0$, the vanishing points v_1 and v_2 must correspond to orthogonal lines.

3) Minimum N is 3. Since solving ω requires 5 constraints, and the camera has "zero skew" and "square pixels", which provides two constraints, meaning that three more independent constraints will do. Therefore we can use 3 vanishing points that corresponding to 3 lines that orthogonal to each other, and construct the constraints as $v_1^T w v_2 = 0$, $v_1^T w v_3 = 0$, $v_2^T w v_3 = 0$. This gives us five constraints, which is enough to compute w and k.

4) Since sets of parallel lines on the same plane lead to collinear vanishing points, it's possible to get sufficient vanishing points, and calibrate them using methods like Least Square Error. Or, purely implementation based, since the image introduces noise due to less distinguishable color distributions, we can extract its edges (e.g. using Sobel method), and interactively get vanishing points with edges clearly marked.



b)

Determine vanishing lines of ground plane and plane-B l_g and l_b respectively, determining vanishing points for two sets of lines parallel to the plane, and then to construct the line through the two vanishing points.

A normal way to do this under low-noise condition:

- Compute n_g and n_b respectively using $n = K^T l$ and normalize them
- Determine if ground plane is orthogonal to plane-B by testing if $n_g \cdot n_b = 0$

Due to the way obtaining vanishing points is approximate and prone to noise, the proper way here:

- Compute $\omega = (KK^T)^{-1}$
- Compute $\theta = arccos(\frac{l_b^T \omega l_g}{\sqrt{l_g^T \omega l_g}\sqrt{l_b^T \omega l_b}})$, compare how close it is to $\frac{\pi}{2}$

Numerical result: $cos\theta = 0.0164, \theta = 1.5544$ which is close to $\frac{\pi}{2}$

c)

Vanishing points and their associated unit vectors do not change their coordinates for a rigid translation of the viewing system. Due to the relative component of rotation, for a vanishing point in two images,

$$v' = Rv$$

Using three non-collinear vanishing points, we can get a 3×3 matrix V whose columns are unit vectors v_1, v_2 and v_3 associated with three vanishing points.

$$V' = RV$$

$\therefore$

$$R = V'V^{-1}$$

∴ The rotation matrix is approximately

$$R = \begin{bmatrix} 1.0004 & -0.0120 & -0.0124 \\ -0.1092 & 1.0322 & -0.0102 \\ 0 & 0 & 1 \end{bmatrix}$$

d)

Let x be the point for world point X before camera transformation, and x' be the one for the post-transformation. So:

$$x = K[I \mid 0]X$$
$$x' = K[R \mid 0]X = KRK^{-1}K[I \mid 0]X = KRK^{-1}x$$

∴ $x' = Hx$ where $H = KRK^{-1}$

$$H = \begin{bmatrix} 2448 & 0 & 1253 \\ 0 & 2438 & 986 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1.0004 & -0.0120 & -0.0124 \\ -0.1092 & 1.0322 & -0.0102 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2448 & 0 & 1253 \\ 0 & 2438 & 986 \\ 0 & 0 & 1 \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} 1.0004 & 1.1340 & -18.9759 \\ -0.1088 & 7.0831 & 79.6519 \\ 0 & 0 & 1 \end{bmatrix}$$

Other details please refer to the code.

3.

a) No. Since the camera matrix is computed using inverse, when using just one image, all points' z is the same, meaning the vertex matrix is not of full rank, and it's not invertible. Unless we count pseudo inverse, or we cannot compute it.

b) err = 2.5707

c) Camera matrix =

$$\begin{bmatrix} 1.3491 & 0.0581 & 0.0526 & 157.5794 \\ -0.0555 & 1.3494 & 0.1140 & 62.6824 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Details please see the code.