ECE 472 Robotics and Vision
Prof. K. Dana
Homework 2: Homography Estimation, Image Formation Pipeline

1. Write a python notebook to "image" a simple object described by vertices (e.g., a simple house) in world coordinates from the view indicated by the extrinsic matrix below. . Use command Line2D to draw the 2D lines between the projected vertices. Do not use any **3 d** plotting commands. The projections should be handled by using the camera matrix (image formation pipeline) discussed in class.

The calibration parameters are given as follows:

$$ {}^{c}R_{w} = \begin{bmatrix} 0.707 & 0.707 & 0 \\ -0.707 & 0.707 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad {}^{c}t_{w} = \begin{bmatrix} -5 \\ 0.5 \\ 4 \end{bmatrix} $$

Therefore

$$ {}^{c}T_{w} = \begin{bmatrix} 0.707 & 0.707 & 0 & -5 \\ -0.707 & 0.707 & 0 & 0.5 \\ 0 & 0 & 1 & 4 \end{bmatrix}. $$

Also

$$ K = \begin{bmatrix} -100 & 0 & 200 \\ -0 & -100 & 200 \\ 0 & 0 & 1 \end{bmatrix} $$

2. Where is the camera in the previous question (w.r.t world coordinates)?

$$ {}^{w}t_{c} = -1 * {}^{c}R_{w}^{T} * {}^{c}t_{w} = -1 * \begin{bmatrix} 0.707 & -0.707 & 0 \\ 0.707 & 0.707 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} -5 \\ 0.5 \\ 4 \end{bmatrix} = \begin{bmatrix} 3.8885 \\ 3.1815 \\ -4 \end{bmatrix} $$

3. Show the simple object from another camera view by changing the position and/or orientation of the camera. You may do this in one of the following two ways: (1) modify ${}^{c}T_{w}$ directly, or (2) choose a "target" point to look at in world coordinates and a new position of the camera in world coordinates and follow the method in class.

Coordinate Frame Transformations

4. Coordinate frame B is described as follows: Start with B coincident with a known frame A. Rotate B about $\hat{z}_A$, by -45°. Translate B by the vector $[1, 1, 0]$ (written with respect to A).

(a) Given ${}^{A}P = [-1, 0, 4]$ what is ${}^{B}P$? That is, given a point that is $[-1, 0, 4]$ when written with respect to the A frame, what is that point written with respect to the B frame?

$$
\begin{aligned}
{}^{B}P &= {}^{B}T_{A}\ {}^{A}P \\
{}^{B}T_{A} &= [{}^{B}R_{A}\ {}^{B}t_{A}] \\
{}^{B}T_{A} &= [{}^{A}R_{B}^{T}\ -1 * {}^{A}R_{B}^{T}{}^{A}t_{B}]
\end{aligned}
$$

$$
{}^{A}R_{B} = rot(\hat{z}, -45^{\circ}) = \begin{bmatrix} cos(-45^{\circ}) & -sin(-45^{\circ}) & 0 \\ sin(-45^{\circ}) & cos(-45^{\circ}) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{A}R_{B}^{T} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad {}^{A}R_{B}^{T}{}^{A}t_{B} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \sqrt{2} \\ 0 \end{bmatrix}
$$

$$
{}^{B}T_{A} = [{}^{A}R_{B}^{T}\ -1 * {}^{A}R_{B}^{T}{}^{A}t_{B}] = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & -\sqrt{2} \\ 0 & 0 & 1 & 0 \end{bmatrix}
$$

$$
{}^{B}P = {}^{B}T_{A} * {}^{A}P = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & -\sqrt{2} \\ 0 & 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} -1 \\ 0 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ -\frac{3}{\sqrt{2}} \\ 4 \end{bmatrix}
$$

5. Coordinate frame B is described as follows: Start with B coincident with a known frame A. Rotate B about $\hat{y}_A$, by 90°. Then Rotate B about $\hat{z}_A$ by 90° Translate B by the vector $[1, 1, 0]$ (written with respect to A).

$$
{}^{A}t_{B} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad {}^{A}R_{B} = rot(\hat{y}, 90^{\circ}) * rot(\hat{z}, 90^{\circ}) = \begin{bmatrix} cos(90^{\circ}) & 0 & sin(90^{\circ}) \\ 0 & 1 & 0 \\ -sin(90^{\circ}) & 0 & cos(90^{\circ}) \end{bmatrix} * \begin{bmatrix} cos(90^{\circ}) & -sin(90^{\circ}) & 0 \\ sin(90^{\circ}) & cos(90^{\circ}) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}
$$

(a) What is $\hat{y}_B$ written with respect to the A frame?

$$
{}^{A}R_{B} = \begin{bmatrix} {}^{A}\hat{x}_{B} & {}^{A}\hat{y}_{B} & {}^{A}\hat{z}_{B} \end{bmatrix}
$$

$$
{}^{A}\hat{y}_{B} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}
$$

(b) What is $\hat{y}_A$ written with respect to the B frame?

$$
{}^{B}R_{A} = {}^{A}R_{B}^{T} = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}
$$

$$
{}^{B}\hat{y}_{A} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
$$

(c) Given ${}^{A}P = [1, 0, 8]$ what is ${}^{B}P$? That is, given a point that is $[1, 0, 8]$ when written with respect to the A frame, what is that point written with respect to the B frame?

$$
\begin{aligned}
&{}^{B}P = {}^{B}T_{A}\ {}^{A}P \\
&{}^{B}T_{A} = [{}^{B}R_{A}\ {}^{B}t_{A}] \\
&{}^{B}t_{A} = -1 * {}^{B}R_{A} * {}^{A}t_{B} = -1 * \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}
\end{aligned}
$$

$$
{}^{B}P = \begin{bmatrix} 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \\ 8 \\ 1 \end{bmatrix} = \begin{bmatrix} -8 \\ 0 \\ -1 \end{bmatrix}
$$

6. Consider Figure 1.

(a) What is the the rotation matrix ${}^{A}R_{B}$?

$$
{}^{A}R_{B} = \begin{bmatrix} {}^{A}\hat{x}_{B} & {}^{A}\hat{y}_{B} & {}^{A}\hat{z}_{B} \end{bmatrix}
$$

$$
{}^{A}R_{B} = \begin{bmatrix} -sin(45^{\circ}) & cos(45^{\circ}) & 0 \\ 0 & 0 & 1 \\ cos(45^{\circ}) & sin(45^{\circ}) & 0 \end{bmatrix}
$$

$$
{}^{A}R_{B} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}
$$

(b) What is the translation vector ${}^{A}t_{B}$?

$$
{}^{A}t_{B} = {}^{A}O_{B} = \begin{bmatrix} 11 \\ 0 \\ 6 \end{bmatrix}
$$

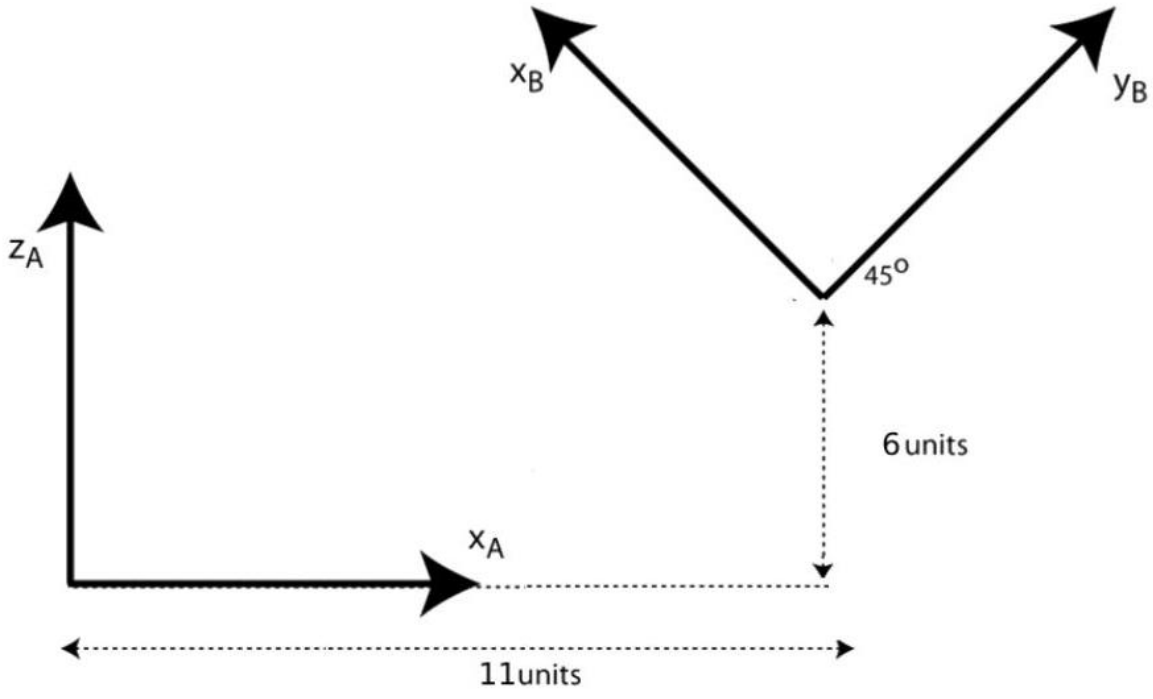

Figure 1: Coordinate frame A and B.

7. Write a python program to estimate a homography between a frontal view and side view image of a planar surface (e.g. side of a building, sign, photo). The program input should also be corresponding point pairs (Matlab's ginput is useful here). The program should estimate the homography using the DLT method. Report the homography parameters and warp the side view image to match a frontal view of the window. An example image is provided in hpworld.png (you may use a different image). You may use an external warping function only if it takes the homography as input (e.g. skimage.transform.warp).

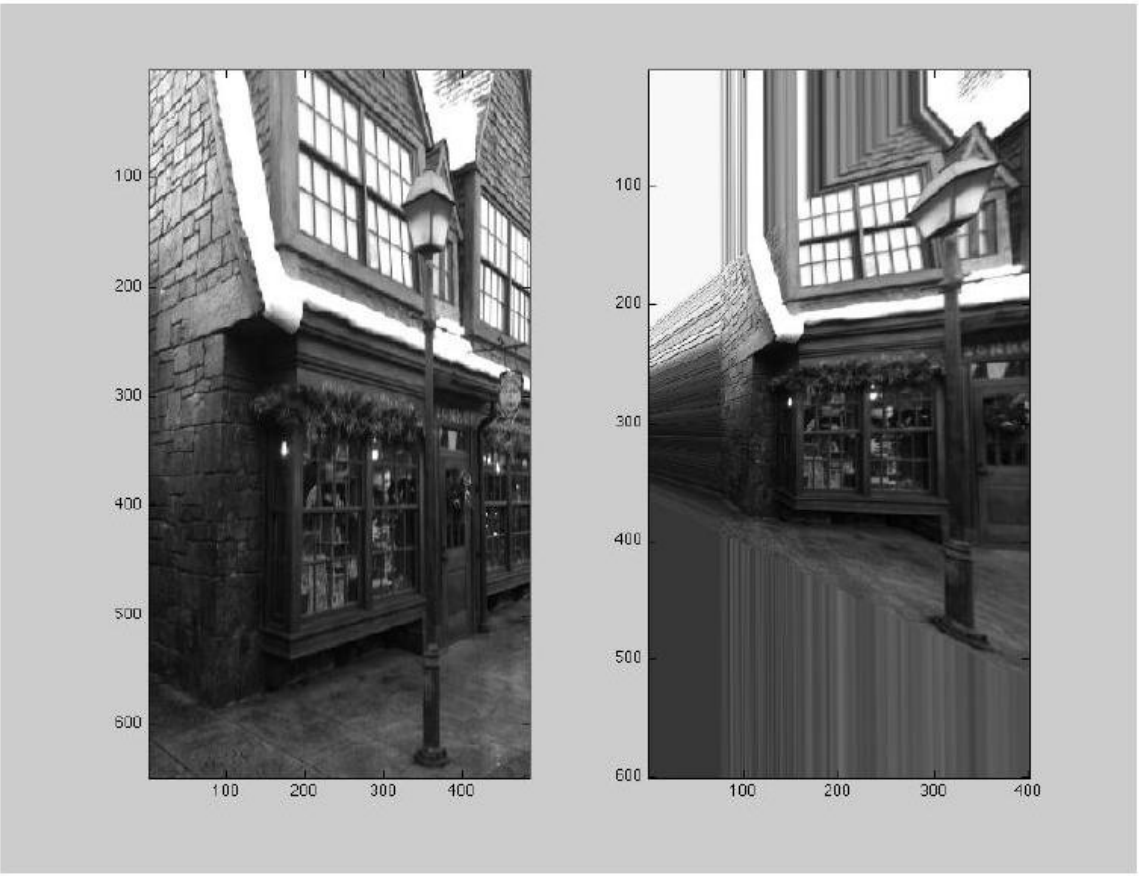

Figure 2: Remap the window region (left) to a frontal view (right) by estimating the appropriate homography and warping the image.