Camera motion assignment (synthetic data)

In this assignment, you will estimate the motion of a camera (i.e., its rotation and translation) given a set of known 3-D points and their corresponding image points. As we already know, the camera model maps 3-D points onto 2-D image points, i.e., for each 3-D point $\mathbf{w}_i = (u_i, v_i, w_i)^\mathsf{T}$, we can use the pinhole camera model (as a function) to output an image point $\mathbf{x}_i = (x_i, y_i)^\mathsf{T}$ as follows:

$$\mathbf{x}_i = \mathrm{pinhole}\left[\mathbf{w}_i, \Lambda, \Omega, \boldsymbol{\tau}\right]. \tag{1}$$

Here, we will use the rotated and translated camera from the previous assignment. We will then create a set of synthetic measurements consisting of 3-D points $\mathbf{w}_i = (u_i, v_i, w_i)^\mathsf{T}$ and their corresponding image points $\mathbf{x}_i = (x_i, y_i)^\mathsf{T}$. To generate the synthetic measurements, we will simply augment the 3-D vertex data from the cube shape by adding midpoints between pair of vertices, and then use all points as input to the full perspective-camera model to produce image points. Figure 1 shows an example of the sample points that you can create.

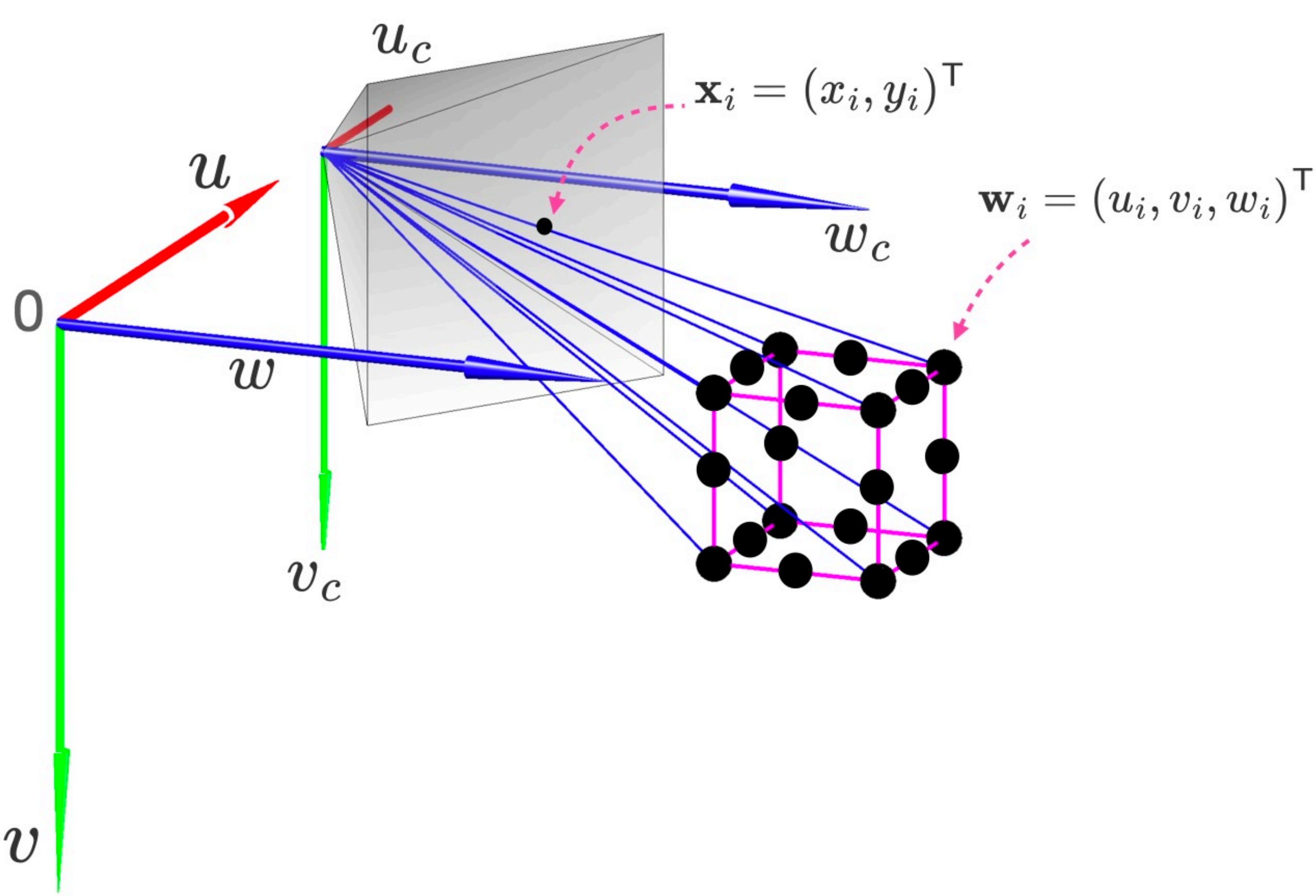

Figure 1: Perspective projection of a set of vertices representing a cube. Extra midpoints for each original pair of vertices have been added. Camera has its own coordinate frame $\{u_c, v_c, w_c\}$ which has been rotated and translated with respect to the global coordinate frame.

Once you generate the measurements, you should have the following set of pairs $M = \{\{\mathbf{w}_1, \mathbf{x}_1\}, \{\mathbf{w}_2, \mathbf{x}_2\}, \ldots, \{\mathbf{w}_N, \mathbf{x}_N\}\}$ where N is the total number of point pairs (i.e., 3-D point and corresponding image point in 2-D), i.e.:

$$
\begin{aligned}
(u_1, v_1, w_1)^\mathsf{T} &\leftrightarrow (x_1, y_1)^\mathsf{T} \\
(u_2, v_2, w_2)^\mathsf{T} &\leftrightarrow (x_2, y_2)^\mathsf{T} \\
&\vdots \\
(u_N, v_N, w_N)^\mathsf{T} &\leftrightarrow (x_N, y_N)^\mathsf{T}
\end{aligned}
\tag{2}
$$

Of course, these are not actual measurements taken from real 3-D points and image points. However, we will use them as if they were real so we can learn how to calculate the camera motion. Another advantage of using synthetic measurements is that we can compare the results of calculations with the ground truth (i.e., the rotation and translation of the synthetic camera).

To calculate the camera motion, we will use the solution described in Section 14.4 (Prince). Specifically, we will use the values of the measurements and the camera parameters to create the measurement matrix in Equation 14.30. For instance, for a 3-D point $\mathbf{w}_1 = (20, 10, 30)^\mathsf{T}$ and its (corresponding) image point in *normalized* coordinates $\mathbf{x}_1' = (1, 5)^\mathsf{T}$, the first two rows of the measurement matrix in Equation 14.30 are:

$$
A \quad = \begin{bmatrix}
u_1 & v_1 & w_1 & 1 & 0 & 0 & 0 & 0 & -u_1 x_1' & -v_1 x_1' & -w_1 x_1' & -x_1' \\
0 & 0 & 0 & 0 & u_1 & v_1 & w_1 & 1 & -u_1 y_1' & -v_1 y_1' & -w_1 y_1' & -y_1' \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots
\end{bmatrix}
\tag{3}
$$

With the complete measurement matrix at hand, we calculate its singular value decomposition and obtain the solution vector. Finally, we will perform the necessary remaining steps to enforce the necessary constraints such as *the orthonormality of the rotation matrix*, and *the removal of the unknown scale factor that affects the estimated translation vector*.

In addition to the linear solution of Equation 14.30, the method described in Section 14.4 requires a final refinement step that involves non-linear optimization. This step does not need to be done in this assignment. Instead, you can simply accept the linear solution as a good-enough solution for the camera motion.

At the end of estimation process, display the rotation matrix and translation vector. Then, use the estimated rotation and translation to "re-project" the 3-D points of the cube to see how accurate your solution is. Figure 2 illustrates the reproduction error, which is the error between the image produced by the actual camera and the image produced by the estimated camera.

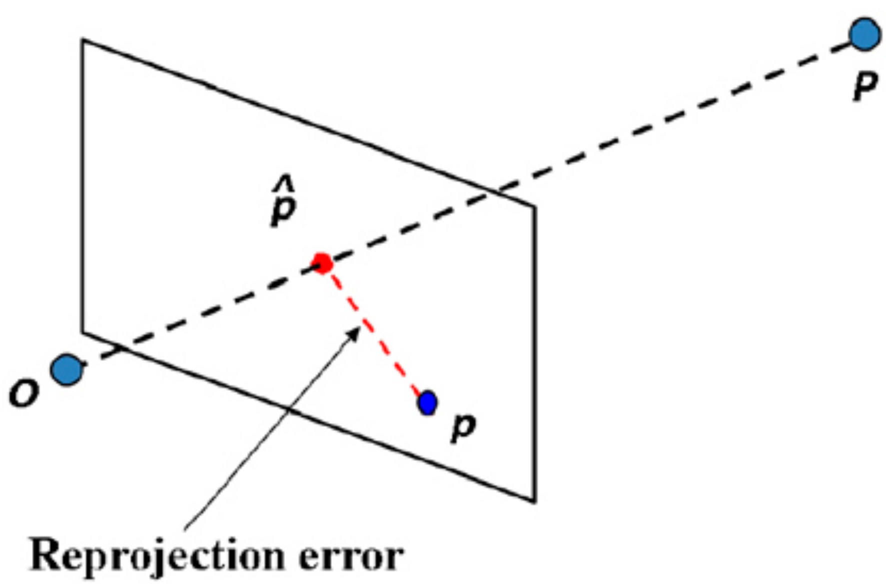

Figure 2: Re-projection error. Here, $\mathbf{p}$ is the actual image point and $\hat{\mathbf{p}}$ is the image point under the estimated camera parameters.

If you want, you can also produce a graphical output that displays the coordinate frame of the camera with respect to the coordinate frame of the cube.