3D World to 2D Image Projection

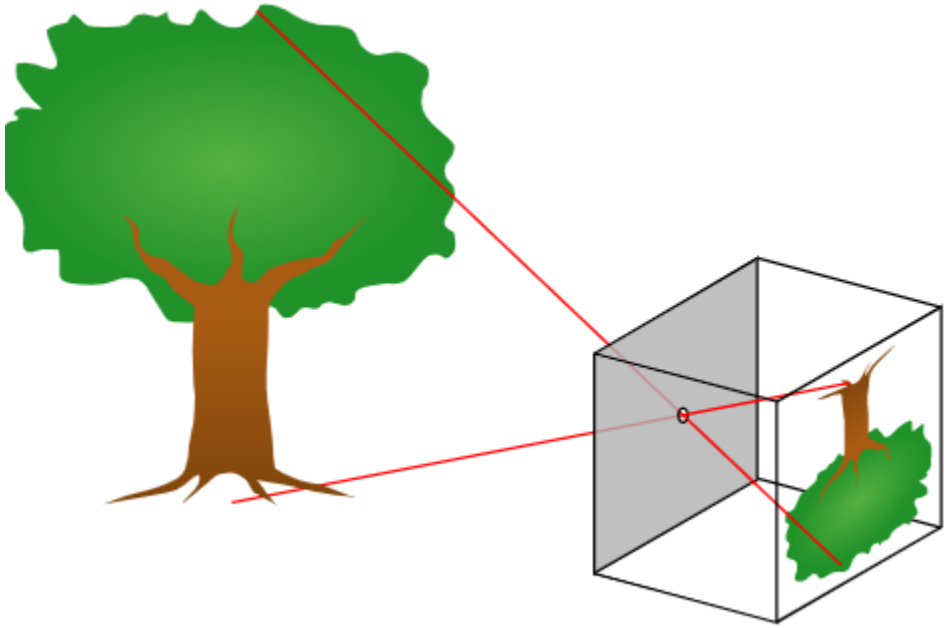
C.-C. Jay Kuo

University of Southern California

Camera Models

- Pinhole camera model
- Perspective camera model
- Orthographic camera model

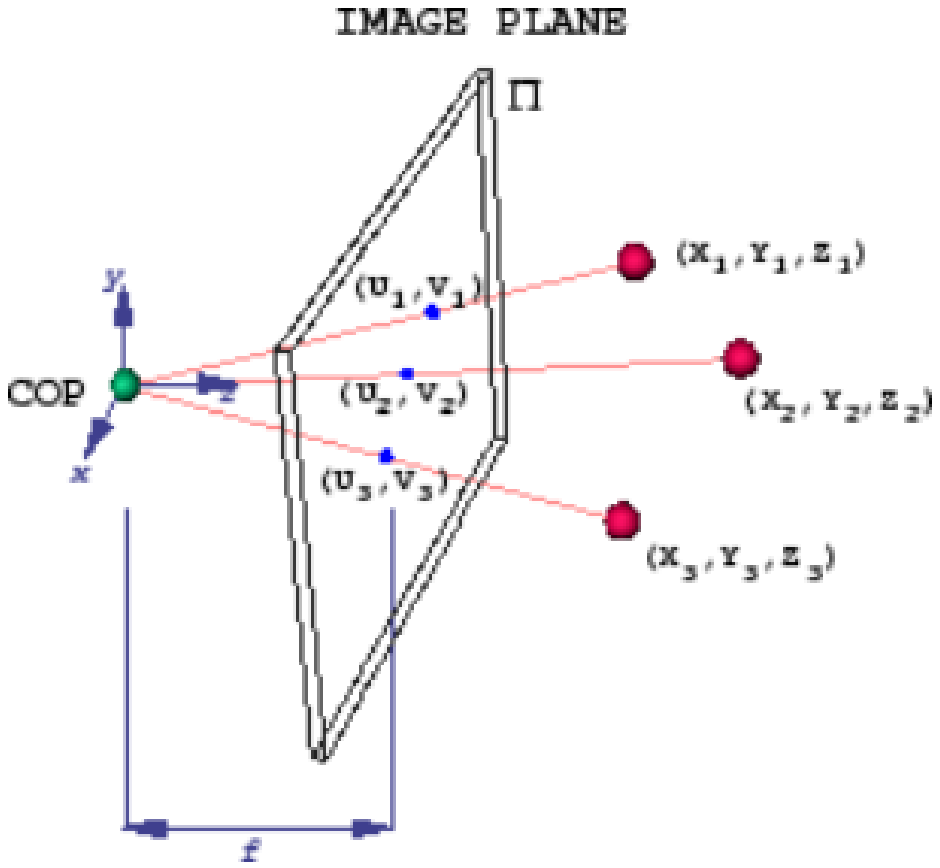
Pinhole Camera Model



Early pinhole camera. Light enters a dark box through a small hole and creates an inverted image on the wall opposite the hole

Perspective Camera Model (1)

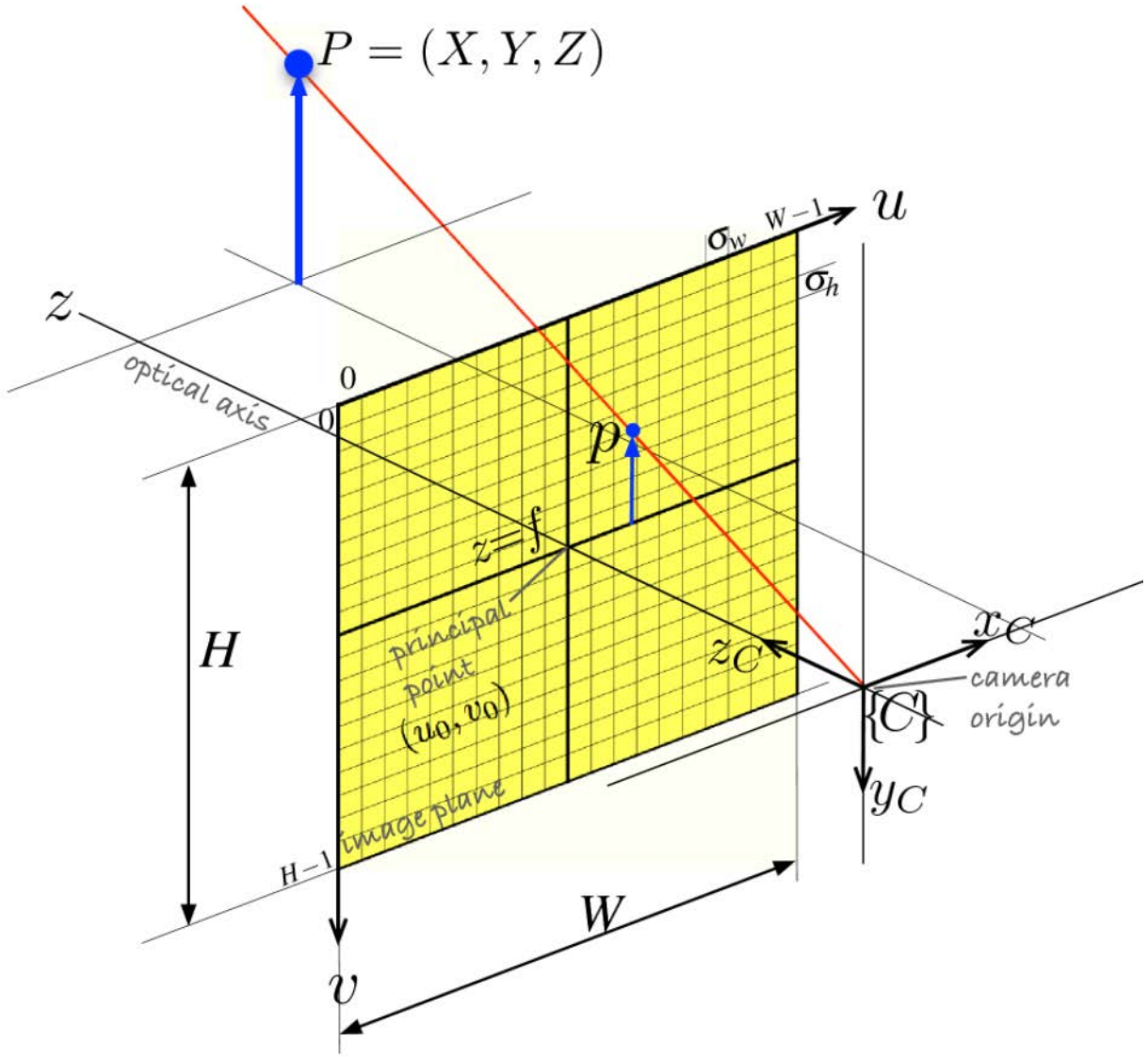
Figure: Perspective Projection



$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} X \\ Y \end{pmatrix} \frac{f}{Z}$$

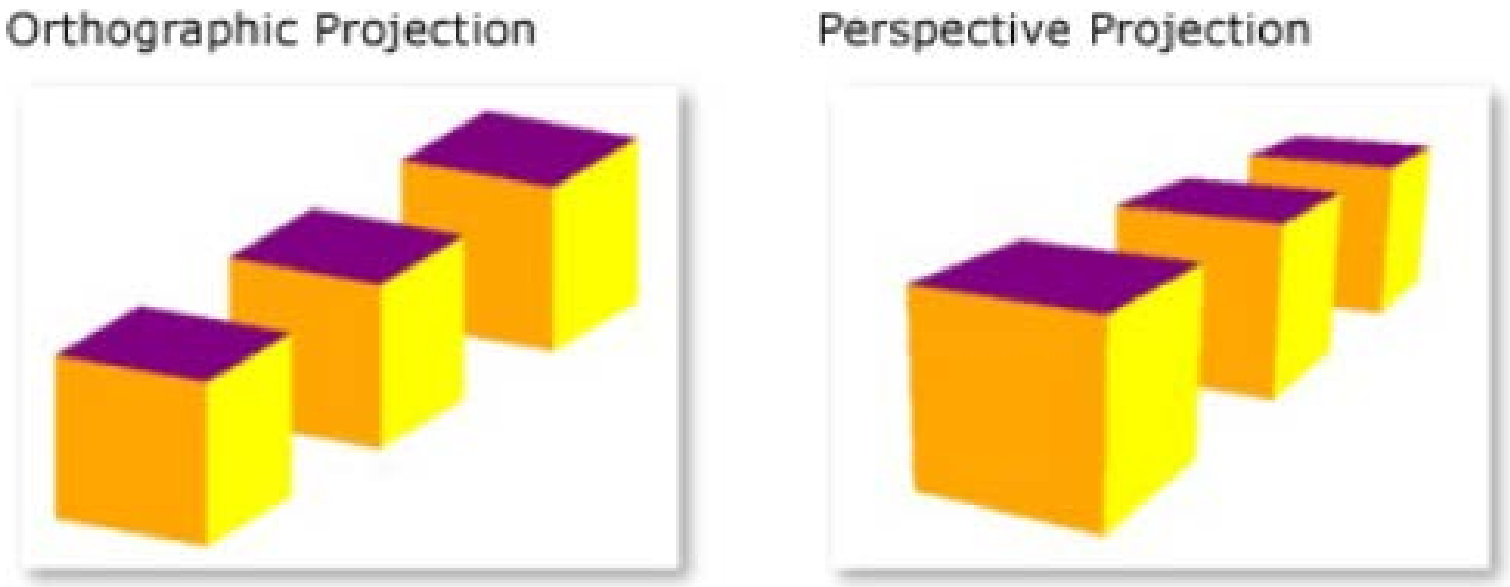
(X,Y,Z) world coordinates
(u,v) image coordinates

Perspective Camera Model (2)



Orthographic versus Perspective Projections

Orthographic projection does not include perspective foreshortening; namely, a viewing box whose sides are parallel, instead of one whose sides meet in a point at the scene's horizon



3D Imaging Geometry

- See Supplemental Material

Perspective Transformation & Imaging Geometry

In this problem, you will use the perspective transformation and imaging geometry to capture a scene in the 3D world coordinates system and project it onto a 2D image plane.

A cube in the 3D world coordinates system is shown in Figure 3, where its center is located at the origin and each side is parallel to one of the XY-, XZ- and YZ-planes. One viewing camera has its lens centered at (5,5,5), viewing angle along the direction of (-1,-1,-1), and focal length *f*. Please show the image observed at the image plane in this problem.

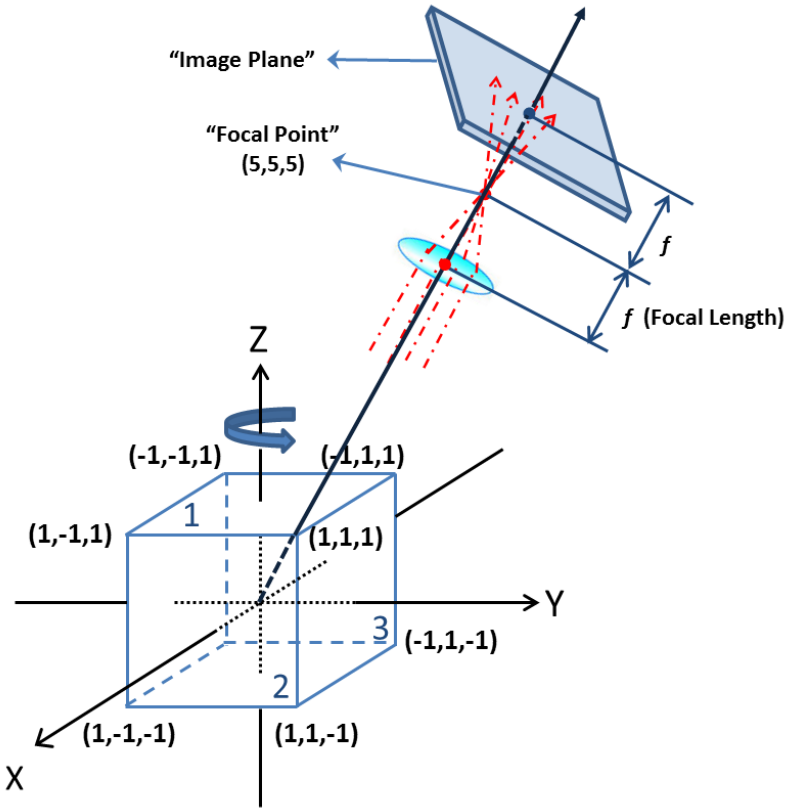


Figure 3: 3D Cube in the World Geometry

To conduct this task, you need to understand and implement two coordinates transform matrices as given below. Generally, the image plane coordinates, $[x, y]^T$, are related to the 3D world coordinates $[X, Y, Z]^T$ via

$$w\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = K[R|t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

where K and $[R|t]$ are called the **intrinsic** camera matrix and the **extrinsic** camera matrix, respectively, and w is a scaling factor. Matrices K and $[R|t]$ are given below.

1. **Extrinsic Camera Matrix: from World Coordinates to Camera Coordinates**
To map the location of a point in the world coordinates to that of the image plane, we need to change it to the camera coordinates system (see Figure 4) as an intermediate step, and $[R|t]$ in Eq. (1) is used to accomplish this task.

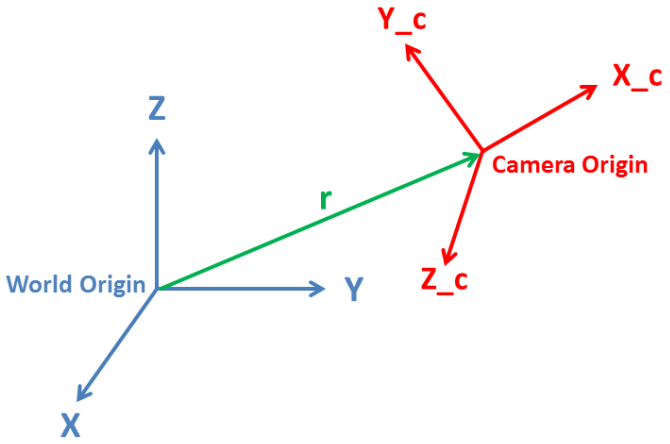


Figure 4: Relationship between the world coordinates and the camera coordinates.

It can be rewritten as

$$[R|t] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} = \begin{bmatrix} X_c^X & X_c^Y & X_c^Z & -\mathbf{r}\cdot\mathbf{X_c} \\ Y_c^X & Y_c^Y & Y_c^Z & -\mathbf{r}\cdot\mathbf{Y_c} \\ Z_c^X & Z_c^Y & Z_c^Z & -\mathbf{r}\cdot\mathbf{Z_c} \end{bmatrix},$$

where $\mathbf{r}$ is the vector from the origin of the world coordinates to the origin of the camera coordinates, and $\mathbf{X}_c=(X_c^X, X_c^Y, X_c^Z)^T$, $\mathbf{Y}_c=(Y_c^X, Y_c^Y, Y_c^Z)^T$, $\mathbf{Z}_c=(Z_c^X, Z_c^Y, Z_c^Z)^T$ are the unit vectors of the camera coordinates system with respect to the world coordinates system, respectively. Based on the configuration in Figure 3, we have

$$\mathbf{Z_c} = (-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}})^T,$$

since the image plane faces the origin of the world coordinates and $\mathbf{r}$=(5,5,5). Without loss of generality, we choose the following initial values for $\mathbf{X}_c$ and $\mathbf{Y}_c$ in matrix $[R|t]$:

$$\mathbf{X_c} = (-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0)^T, \qquad \mathbf{Y_c} = (\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, -\frac{2}{\sqrt{6}})^T.$$

2. **Intrinsic Camera Matrix: from Camera Coordinates to Image Plane Coordinates**
Next, we map the location of a point in the camera coordinates to that of the image plane. In Figure 5, X_c, Y_c and Z_c are three axes of the camera coordinates, and **P** and **P'** are points in the camera coordinates and its projective point onto the image plane, respectively. Matrix K in Eq. (1) can be derived using the perspective image transformation as

$$\omega \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} f \cdot X_c + c_x \cdot Z_c \\ f \cdot Y_c + c_y \cdot Z_c \\ Z_c \end{bmatrix}$$

where (c_x, c_y) is the image coordinates of the intersecting point between the optical axis and the image plane.
After the simplification, we have

$$x = \frac{f \cdot X_c + c_x \cdot Z_c}{Z_c} = f \cdot \frac{X_c}{Z_c} + c_x, \qquad y = \frac{f \cdot Y_c + c_y \cdot Z_c}{Z_c} = f \cdot \frac{Y_c}{Z_c} + c_y.$$

Note that $c_x = \frac{ImageWidth}{2}$ and $c_y = \frac{ImageHeight}{2}$ based on the image coordinates convention. Also, by shifting the origin of the image plane to (c_x, c_y), one can verify that

$$x : X_c = f : Z_c \leftrightarrow x = f \cdot \frac{X_c}{Z_c}, \qquad y : Y_c = f : Z_c \leftrightarrow y = f \cdot \frac{Y_c}{Z_c}$$

as shown in Figure 5.

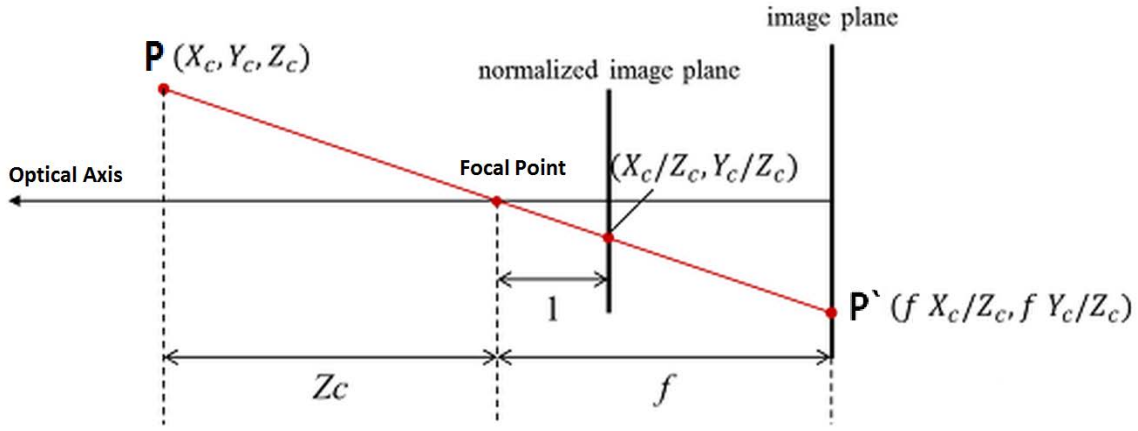


Figure 5: Relationship between the camera and the image plane coordinates.

i. Pre-processing
Figure 6 shows five 200x200 images which will be placed on five faces of the cube as shown in 6(f). You can ignore the bottom face in this problem. With the configuration in Figure 3, the camera can observe images on face no.1, no. 2 and no. 3.

Write a program to generate the location/intensity information of the five images by assuming the unit length of the world coordinates is 0.01. Because the resolution of the image is 200x200 and the side length of the cube is equal to 2, we have $\frac{2}{200} = 0.01$. Explain the basic idea of your implementation. Please specify the input and the output of your program clearly.

ii. Capturing 3D scene

In this part, you need to display the cube image on the image plane captured by the camera. By following the aforementioned steps, you can determine K and $[R|t]$.

Please first implement the forward mapping algorithm (from the world coordinates to the camera coordinates) with the following three fixed parameters:

- Focal length $f = \sqrt{3}$,
- Image width w = 200 pixels,
- Image height h = 200 pixels.

There is one more parameter called the pixel density (namely, the number of pixels per unit length) for you to determine. Note that a poor choice of pixel density will result in a projected image either too small or too big. Please specify one good value of pixel density and provide the reason for your choice. Discuss the resulting projected image. If there are undesirable artifacts, explain the source of the distortions. Is there any way to improve the result? If so, describe your own algorithm (you do not need to actually implement it).

Then, with the set of parameters obtained above, please implement the reverse mapping (from the camera coordinates to the world coordinates). What are the challenges in the reverse mapping?

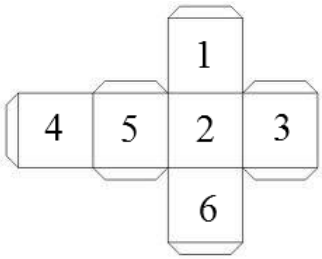
(a) baby

(b) baby cat

(c) baby dog

(d) baby panda

(e) baby bear



(f) Unfolded cube image

Figure 6: Mapping five images to five faces of the cube.

Homographic Transformation

- See Supplemental Material

Homographic Transformation and Image Overlay

One can use the homographic transformation and the image overlap techniques to synthesize interesting images. One example is shown in Figure. 1, where the left two images are seed images and the right one is the desired output. The field image is the host image and the *Tartans* image is the embedded text image. Note that the black region outside the yellow contour of the embedded image is removed.

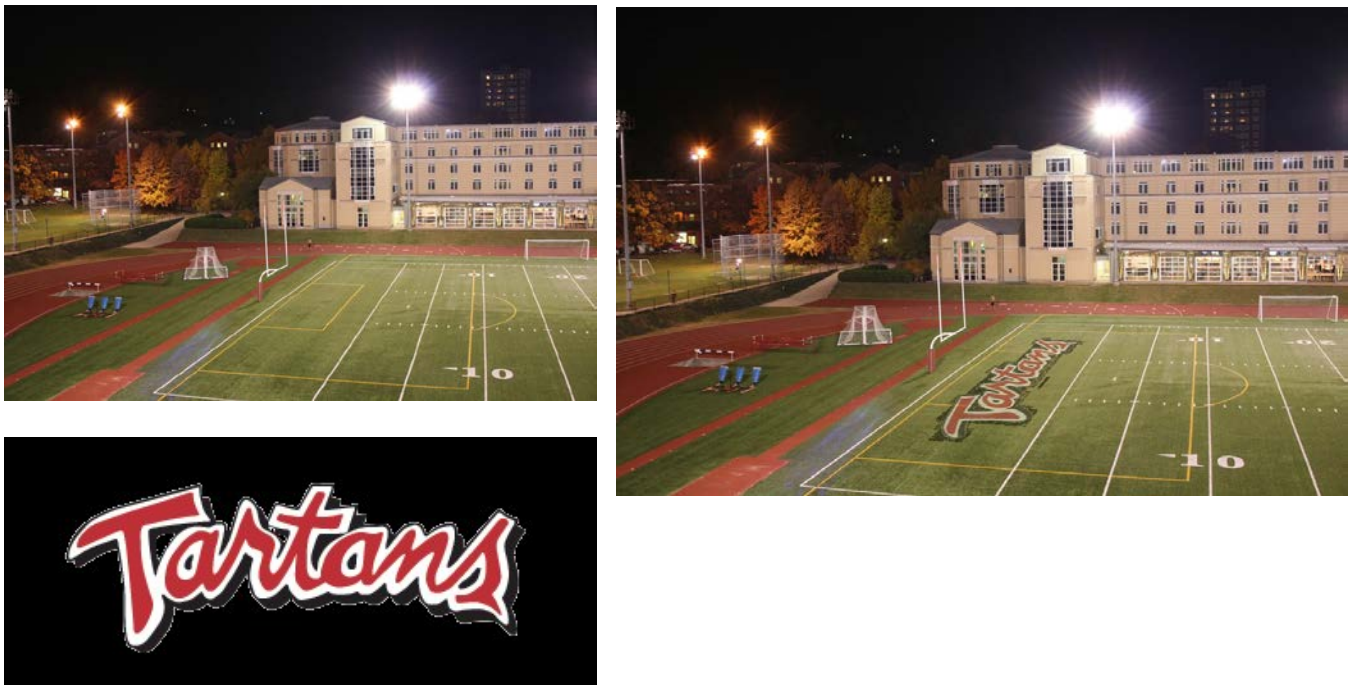


Figure 1: An example of homographic transformation and image overlay

The homographic transformation procedure is stated below. Images of points in a plane, from two different camera viewpoints, under perspective projection (pin hole camera models) are related by a homography:

$$P_2 = HP_1$$

where H is a 3x3 homographic transformation matrix, P_1 and P_2 denote the corresponding image points in homogeneous coordinates before and after the transform, respectively. Specifically, we have

$$\begin{bmatrix} x_2' \\ y_2' \\ w_2' \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \text{ and } \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \frac{x_2'}{w_2'} \\ \frac{y_2'}{w_2'} \end{bmatrix}$$

To estimate matrix H, you can proceed with the following steps:

- Fix $H_{33} = 1$ so that there are only 8 parameters to be determined.
- Select four point pairs in two images to build eight linear equations.
- Solve the equations to get the 8 parameters of matrix H.

- After you determine matrix H, you can project all points from one image to another by following the backward mapping procedure and applying the interpolation technique.

Implement above homographic transformation steps to generate the desired result as shown in Figure 1.

Now, let us keep the same host image and the same overlay region, apply the algorithm to the embedded text image *Trojans* in Figure 2 to generate a new image overlay. Show the result and make discussion on the performance.

Figure 2: The Trojans text image.