Homography from Polygon in R^3 to Image plane

Li Zhang　　Steve Seitz

1 A 2D Coordinate System for a Plane in 3D Space

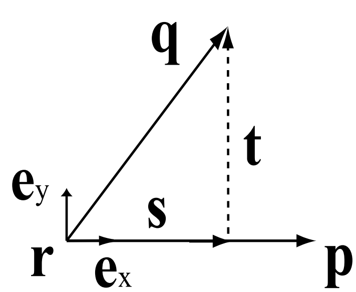

Figure 1: plane coordinate system from 3 points in R^3

Let $\mathbf{p}, \mathbf{q}, \mathbf{r}$ be 3 points in R^3. If they are not colinear, they define a unique plane. We want to set up a 2D coordinate system in the plane such that each point $\mathbf{a}$ in the plane has a two dimensional coordinate (u, v).

Let $\mathbf{r}$ be the origin of the coordinate system and

$$\mathbf{e}_x = \frac{\mathbf{p} - \mathbf{r}}{\|\mathbf{p} - \mathbf{r}\|}$$

be the base vector for x axis in the plane. We then decompose vector $\mathbf{q} - \mathbf{r}$ into two components, one is parallel to $\mathbf{e}_x$ and one is orthogonal to $\mathbf{e}_x$. The parallel component is

$$\mathbf{s} = < \mathbf{q} - \mathbf{r}, \mathbf{e}_x > \mathbf{e}_x$$

and the orthogonal component is

$$\mathbf{t} = (\mathbf{q} - \mathbf{r}) - \mathbf{s}$$

where $< \cdot, \cdot >$ is the dot product of two vectors. The base vector for y axis in the plane is

$$\mathbf{e}_y = \frac{\mathbf{t}}{\|\mathbf{t}\|}$$

For any point $\mathbf{a}$ in the plane, its two dimensional coordinate in the plane with respect to $\mathbf{e}_x$ and $\mathbf{e}_y$ is

$$(< \mathbf{a} - \mathbf{r}, \mathbf{e}_x >, < \mathbf{a} - \mathbf{r}, \mathbf{e}_y >)$$

2 The Homography from a Polygon to its Image

Let $\{\mathbf{p}_1 = (X_1, Y_1, Z_1), \mathbf{p}_2 = (X_2, Y_2, Z_2), \ldots\}$ be the vertices of a polygon in 3D space. Assumming $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ are not colinear, e.g. define a unique plane, they will introduce a 2D coordinate system in the plane. Using the method described in Section 1, each $\mathbf{p}_i$ can have a 2D coordinate (u_i, v_i) in the plane.

Let

$$u_{\min} = \min\{u_i\}, u_{\max} = \max\{u_i\}, v_{\min} = \min\{v_i\}, v_{\max} = \max\{v_i\}$$

and we normalize (u_i, v_i) by

$$\hat{u}_i = \frac{u_i - u_{\min}}{u_{\max} - u_{\min}}, \hat{v}_i = \frac{v_i - v_{\min}}{v_{\max} - v_{\min}}$$

Now each $(\hat{u}_i, \hat{v}_i)$ is between $[0, 1]$ and can be used as texture coordinates.

Suppose the image coordinate of $\mathbf{p}_i$ is (x_i, y_i), we estimate the homography $\mathbf{H}$ which maps each $(\hat{u}_i, \hat{v}_i)$ to (x_i, y_i).

(Note: this is the H in the skeleton code in SVMPolygon structure. $(\hat{u}_i, \hat{v}_i)$ is used as texture coordinate for point $\mathbf{p}_i$.)

The estimation algorithm is covered in Steve's lecture. That is, $\mathbf{h}$ is the eigenvector of the 9 by 9 semi-positive-definite matrix, whose eigenvalue is the smallest. The code for eigen-decomposition is provided in jacob.h/cpp. The main.cpp file associated with them show how to use the function `jacob(...)`. Those files are downloaded seperately from our skeleton code.

2.1 More Accurate Estimation

As we do for solving vanishing point, we recommend you normalize (x_i, y_i) first before estimating the homograph $\mathbf{h}$. That is, we first compute

$$x_{\min} = \min\{x_i\}, x_{\max} = \max\{x_i\}, y_{\min} = \min\{y_i\}, y_{\max} = \max\{y_i\}$$

and normalize (x_i, y_i) as

$$\hat{x}_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \hat{y}_i = \frac{y_i - v_{\min}}{y_{\max} - y_{\min}}$$

The normalization can be written as

$$\begin{pmatrix} \hat{x}_i \\ \hat{y}_i \\ 1 \end{pmatrix} = \mathbf{S} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}$$

where $\mathbf{S} = \begin{bmatrix} \frac{1}{x_{\max}-x_{\min}} & 0 & \frac{-x_{\min}}{x_{\max}-x_{\min}} \\ 0 & \frac{1}{y_{\max}-y_{\min}} & \frac{-y_{\min}}{y_{\max}-y_{\min}} \\ 0 & 0 & 1 \end{bmatrix}$.

You want to first estimate a homograph $\mathbf{H}_n$ from $(\hat{u}_i, \hat{v}_i)$ to $(\hat{x}_i, \hat{y}_i)$ and then compute $\mathbf{H} = \mathbf{S}^{-1}\mathbf{H}_n$. Some theoretical analysis proves that $\mathbf{H}_n$ can be more accurately estimated, which is beyond the scope of the class.