Image Formation and Analysis
(Formation et Analyse d'Images)

James L. Crowley

ENSIMAG 3 - MMIS Option MIRV First Semester 2010/2011

Lesson 3 11 Oct 2010

Calibrating a Camera Model

Lesson Outline:

Calibrating a Camera Model

The Complete Camera Model

$$P^i = \mathbf{C}_r^i \; \mathbf{P}_c^r \; \mathbf{T}_s^c \, P^s = \mathbf{M}_s^i \; P^s$$

$$\begin{bmatrix} w\,i \\ w\,j \\ w \end{bmatrix} = \mathbf{M}_s^i \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}$$

and thus

$$i = \frac{w\,i}{w} = \frac{M_s^1 \cdot P^s}{M_s^3 \cdot P^s} \qquad j = \frac{w\,j}{w} = \frac{M_s^2 \cdot P^s}{M_s^3 \cdot P^s}$$

or

$$i = \frac{w\,i}{w} = \frac{M_{11} X_s + M_{12} Y_s + M_{13} Z_s + M_{14}}{M_{31} X_s + M_{32} Y_s + M_{33} Z_s + M_{34}}$$

$$j = \frac{w\,j}{w} = \frac{M_{21} X_s + M_{22} Y_s + M_{23} Z_s + M_{24}}{M_{31} X_s + M_{32} Y_s + M_{33} Z_s + M_{34}}$$

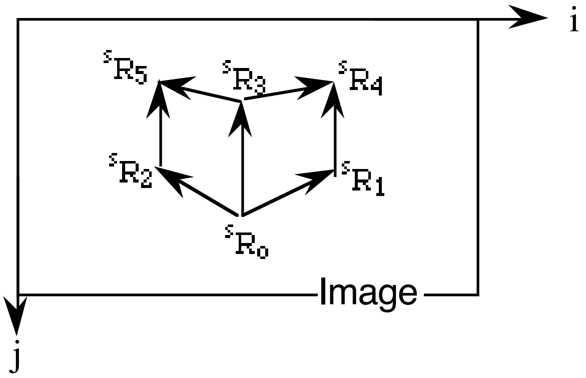
Calibrating the Camera

How can we obtain $\mathbf{M}_s^i$? By a process of calibration.

Observe a set of at least 6 non-coplanar points whose position in the world is known.

R^s_k for k=0,1,2,3,4,5 (s are the scene coordinate axes s=1,2,3)

For example, we can use the corners of a cube. Define the lower front corner as the origin, and the edges as unit distances.



The matrice $\mathbf{M}_s^i$ is composed of 3x4=12 coefficients. However because, $\mathbf{M}_s^i$ is in homogeneous coordinates, the coordinate m_{34} can be set to 1.

Thus there are 12-1 = 11.
We can determine these coefficients by observing known points in the scene. (R^s_k).

Each point provides two coefficients. Thus, for 11 coefficients we need at least $5\frac{1}{2}$ points. With 6 points the system is over-constrained.

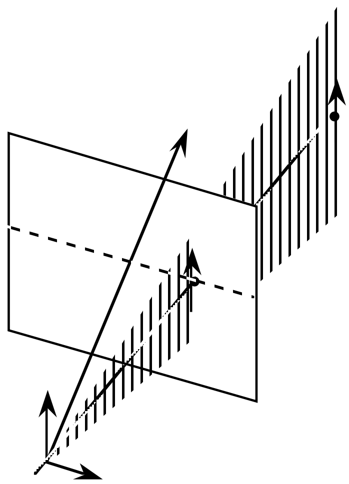
For each known calibration point R^s_k given its observed image position P^s_k, we can write:

$$i_k = \frac{w_k\, i_k}{w_k} = \frac{M_s^1 \cdot R_k^s}{M_s^3 \cdot R_k^s} \qquad j_k = \frac{w_k\, j_k}{w_k} = \frac{M_s^2 \cdot R_k^s}{M_s^3 \cdot R_k^s}$$

This gives 2 equations for each point.

$$(M_s^1 \cdot R^s_k) - i_k\,(M_s^3 \cdot R^s_k) = 0 \qquad (M_s^2 \cdot R^s_k) - j_k\,(M_s^3 \cdot R^s_k) = 0$$

Each pair of equations corresponds to the planes that pass though the image row and the image column of the observed image point P^s_k



The equation $(M_s^1 \cdot R^s{}_k) - i_k\,(M_s^3 \cdot R^s{}_k) = 0$ is the vertical plane that includes the projective center through the pixel $i=i_k$.

The equation $(M_s^2 \cdot R^s{}_k) - j_k\,(M_s^3 \cdot R^s{}_k) = 0$ is the horizontal plane that includes the projective center and the row $j=j_k$.

In tensor notation

given $P^i = \begin{pmatrix} wi \\ wj \\ w \end{pmatrix}$ we write : $P^i = \mathbf{M}_s^i\ R^s$

with k scene points, R_k^S and their image correspondences P_k^i we can write

$$P_k^i = \mathbf{M}_s^i\ R_k^s$$

with $i \cdot w = P_k^1/P_k^3$ et $j \cdot w = P_k^2/P_k^3$ for each image point k, there are two independent equations

$$\begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix} = \begin{pmatrix} wi \\ wj \\ w \end{pmatrix} \quad \text{donc} \quad \begin{pmatrix} i \\ j \\ 1 \end{pmatrix} = \begin{pmatrix} p^1/p^3 \\ p^2/p^3 \\ 1 \end{pmatrix}$$

and with $P_k^3 = \mathbf{M}_s^3\, R_k^3$

$$i = p^1/p^3 = \mathbf{M}_s^1\ R_k^s \ /\ \mathbf{M}_s^3\ R_k^s \ \Rightarrow i\,\mathbf{M}_s^3\ R_k^s - \mathbf{M}_s^1\ R_k^s = 0$$
$$j = p^2/p^3 = \mathbf{M}_s^2\ R_k^s \ /\ \mathbf{M}_s^3\ R_k^s \ \Rightarrow j\,\mathbf{M}_s^3\ R_k^s - \mathbf{M}_s^2\ R_k^s = 0$$

We can write this as:

$$\begin{pmatrix} R^1 & R^2 & R^3 & 1 & 0 & 0 & 0 & 0 & -iR^1 & -iR^2 & -iR^3 & -i \\ 0 & 0 & 0 & 0 & R^1 & R^2 & R^3 & 1 & -jR^1 & -jR^2 & -jR^3 & -j \end{pmatrix} \begin{pmatrix} \mathbf{M}_1^1 \\ \mathbf{M}_2^1 \\ \mathbf{M}_3^1 \\ \mathbf{M}_4^1 \\ \mathbf{M}_1^2 \\ \mathbf{M}_2^2 \\ \mathbf{M}_3^2 \\ \mathbf{M}_4^2 \\ \mathbf{M}_1^3 \\ \mathbf{M}_2^3 \\ \mathbf{M}_3^3 \\ \mathbf{M}_4^3 \end{pmatrix} = 0$$

For N non-coplanair points we can write 2N equations.

$$\mathbf{A}\ \mathbf{M}_s^i = 0.$$

We then use least squares to minimize the criteria:

$$\mathbf{C} = \| \mathbf{A}\ \mathbf{M}_s^i \|$$

For example, give a cube with observed corners

$P^L{}_o = (101, 221)$ $P^L{}_1 = (144, 181)$ $P^L{}_2 = (22, 196)$
$P^L{}_3 = (105, 88)$ $P^L{}_4 = (145, 59)$ $P^L{}_5 = (23, 67)$

Least squares will give:

$$\mathbf{M}_s^i = \begin{pmatrix} 55.886873 & -79.292084 & 1.276703 & 101.917630 \\ -22.289319 & -17.878203 & -134.345576 & 221.300658 \\ 0.100734 & 0.038274 & -0.008458 & 1.000000 \end{pmatrix}$$

Alternate Derivation using the Cross product

In classic matrix notation:

$$\vec{P} \times M_s^i \vec{R} = 0$$

The term $\vec{R}$ can be factored to set $\vec{P}\ \vec{R} \times M_s^i = 0$

This gives $$\begin{pmatrix} 0 & -wR^s & jwR^s \\ -wR^s & 0 & -iwR^s \\ wR^s & -wR^s & 0 \end{pmatrix} \begin{pmatrix} \mathbf{M}_s^1 \\ \mathbf{M}_s^2 \\ \mathbf{M}_s^3 \end{pmatrix} = 0$$

Where $\vec{R}$ and $\mathbf{M}_s^i$ are vectors. Thus:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & wX & wY & wZ & w1 & -jwX & -jwY & -jwZ & -jw1 \\ -wX & -wY & -wZ & -w & 0 & 0 & 0 & 0 & -iwX & -iwY & -iwZ & -iw1 \\ wX & wY & wZ & w & -wX & -wY & -wZ & -w & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{M}_1^1 \\ \mathbf{M}_2^1 \\ \mathbf{M}_3^1 \\ \mathbf{M}_4^1 \\ \mathbf{M}_1^2 \\ \mathbf{M}_2^2 \\ \mathbf{M}_3^2 \\ \mathbf{M}_4^2 \\ \mathbf{M}_1^3 \\ \mathbf{M}_2^3 \\ \mathbf{M}_3^3 \\ \mathbf{M}_4^3 \end{pmatrix} = 0$$

Any two of the equations are independent.

Homography between two planes.

The projection of a plane to another plane is a degenerate case of the the project transform. In this case, the transform is bijective and reduces to a 3 x 3 invertible

This matrix can be used to rectify an image to a perpendicular view.

$$\boxed{Q^B = \mathbf{H}_A^B \; P^A}$$

In classic notation

$$\begin{pmatrix} w\,x_B \\ w\,y_B \\ w \end{pmatrix} = \mathbf{H}_A^B \begin{pmatrix} x_A \\ y_A \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \begin{pmatrix} x_A \\ y_A \\ 1 \end{pmatrix}$$

$$x_B = \frac{w\,x_B}{w} = \frac{m_{11}\,x_A + m_{12}\,y_A + m_{13}}{m_{31}\,x_A + m_{32}\,y_A + m_{33}}$$

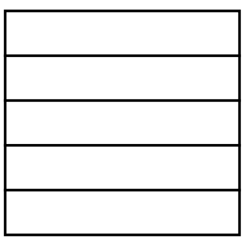
$$y_B = \frac{w\,y_B}{w} = \frac{m_{21}\,x_A + m_{22}\,y_A + m_{23}}{m_{31}\,x_A + m_{32}\,y_A + m_{33}}$$

In tensor notation:

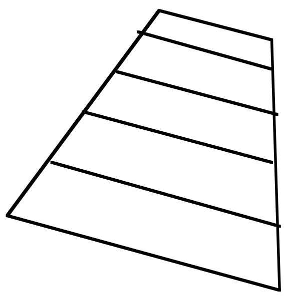
$$\boxed{Q^B = \mathbf{H}_A^B \; P^A}$$

$$\begin{pmatrix} q^1 \\ q^2 \\ q^3 \end{pmatrix} = \mathbf{H}_A^B \begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix} = \begin{pmatrix} h_1^1 & h_2^1 & h_3^1 \\ h_1^2 & h_2^2 & h_3^2 \\ h_1^3 & h_2^3 & h_3^3 \end{pmatrix} \begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix}$$

$$x_B = \frac{q^1}{q^3} \qquad y_B = \frac{q^2}{q^3}$$



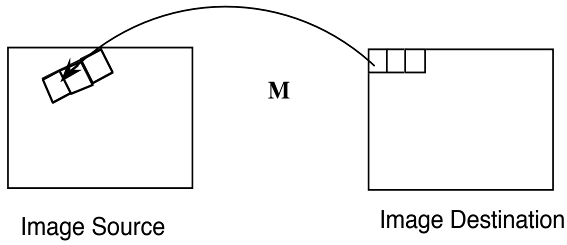
Image



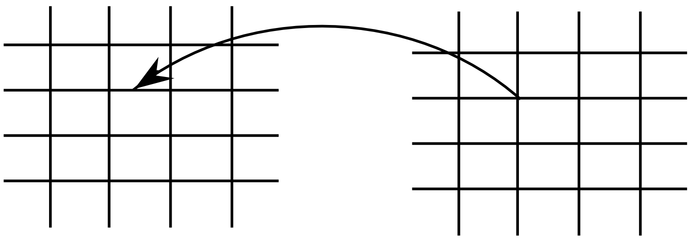
Homographic projection

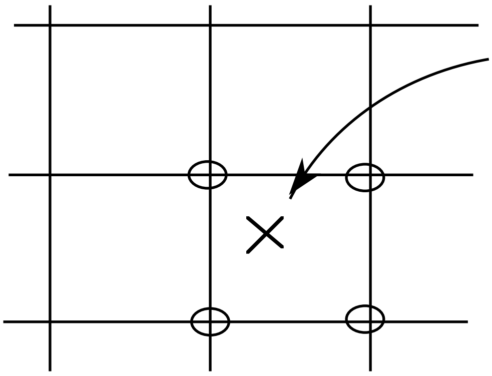
Image Transformations

For each pixel in the destination image, (x_d, y_d) compute its position in the source image(x_s, y_s)



Determine the appropriate pixel value (intensity or color) for the source image and put this pixel value in the destination.



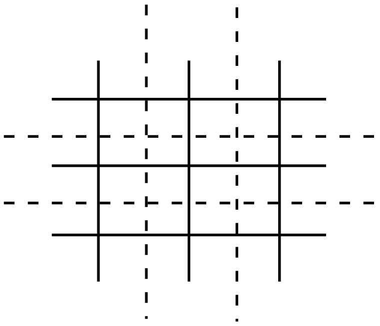


The problem is that the calculated pixel is a real number.
To obtain a destination pixel value we need to interpolate. This can be done by

zeroth order:	Nearest neighbor
First order:	Linear or bilinear interpolation
second order	Cubic spline.

Zero Order Interpolation.

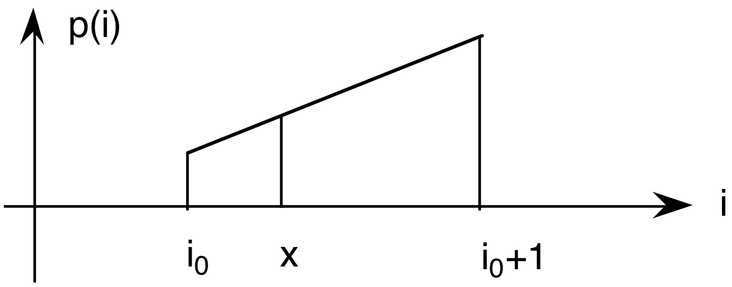
Use the pixel value of the position i_2, j_2 that is closest to the (x_B, y_B)
This is essentially rounding (x_B, y_B) to the nearest integer value



The dashed lines represent decision lines

Linear Interpolation

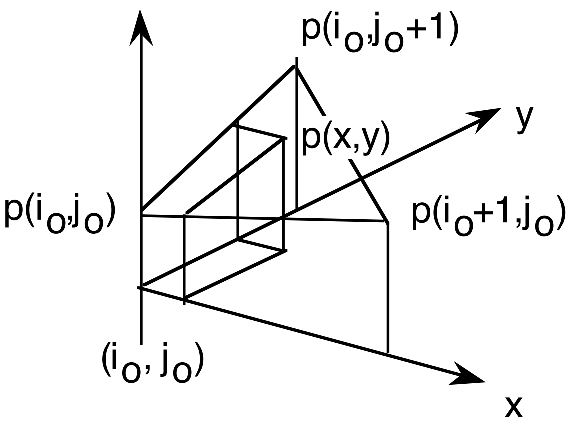
For a 1 D signal, interpolation, between pixel i_o and its neighbor is $i_o \leq x \leq i_o+1$



Calculate the slope : $m_x \equiv \frac{\Delta P}{\Delta x} = = p(i+1) - p(i)$

Then: $p(x) = (x-i)\, m_x + p(i)$

In 2 D, linear interpolation is only valid in the triangle defined by the three points p(i,j), p(i+1, j), p(i,j+1).



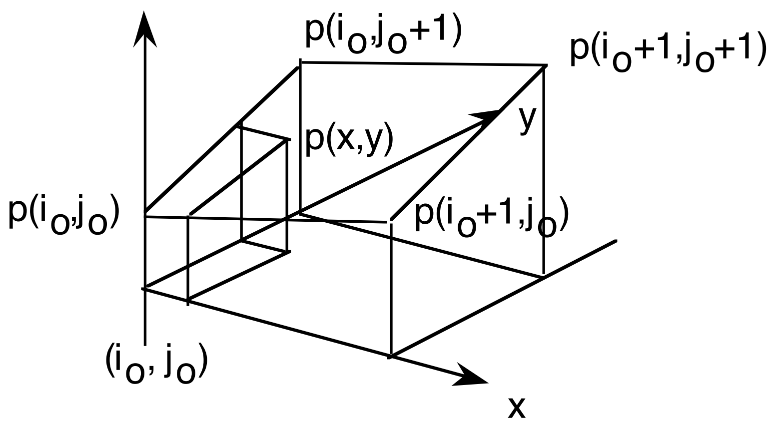
$$m_x \equiv \frac{\Delta P}{\Delta x} = p(i+1, j) - p(i, j)$$

$$m_y \equiv \frac{\Delta P}{\Delta y} = p(i, j+1) - p(i, j)$$

$$p(x, y) = m_x \cdot (x-i) + m_y \cdot (y-j) + p(i, j)$$

If we are closer to p(i+1,j+1) the value is not accurate. It is better to use bilinear interploation

Bilinear Interpolation:



The mathematical form is a hyperbolic parabaloid.

$$p(x, y) = a\,x + b\,y + c\,x\,y + d.$$

This is equivalent to the interpolation in y between a pair of points computed as interpolations in x at y and y+1.

Derivation :

$$a \equiv m_x = \frac{\Delta P}{\Delta x} = p(i+1, j) - p(i, j)$$

$$b \equiv m_y = \frac{\Delta P}{\Delta y} = p(i, j+1) - p(i, j)$$

$$c \equiv m_{xy} = p(i+1, j) + p(i, j+1) - p(i, j) - p(i+1, j+1)$$

$$d = p(i, j)$$

$$p(x, y) = a \cdot (x-i) + b \cdot (y-j) + c \cdot (x-i) \cdot (y-j) + p(i, j)$$