Computer Vision

James L. Crowley

M2R MoSIG option GVR — Fall Semester
3 Oct 2013

Lesson 2

Homogeneous Coordinates and Projective Camera Models

Lesson Outline:

1 Homogeneous Coordinates and Tensor Notation

Homogeneous coordinates allow us to express translation, rotation, scaling, and projection all as matrix operations. The principle is to add an extra dimension to each vector.

For example, points on a plane are expressed as:

$$\vec{\mathrm{P}} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Similarly, points in 3D space become

$$\vec{\mathrm{Q}} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

The line equation, ax+by+c=0 can be expressed as a simple product:

$$\vec{L}^T \vec{P} = \begin{pmatrix} a & b & c \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0 \qquad \text{where} \qquad \vec{\mathrm{L}}^{\mathrm{T}} = \begin{pmatrix} a & b & c \end{pmatrix}$$

Similarly, for a plane equation: *ax+by+cz+d=1:*

$$\vec{S}^T \vec{P} = \begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = 0 \qquad \text{where} \qquad \vec{\mathrm{S}}^{\mathrm{T}} = \begin{pmatrix} a & b & c & d \end{pmatrix}$$

Note that in Homogeneous coordinates, all scalar multiplications are equivalent.

$$a\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = b\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Any vector can be expressed in "canonical" form by normalizing the last coefficient to 1.

$$\begin{pmatrix} ax \\ ay \\ a \end{pmatrix} = \begin{pmatrix} ax/a \\ ay/a \\ a/a \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Our camera model will have the form of a 3 x 4 matrix

$$M_s^i = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix}$$

such that the image point (x_i, y_i) is found from the scene point (x_s, y_s, z_s) by

$$\begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{pmatrix} q_1/q_3 \\ q_2/q_3 \\ 1 \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} = \vec{Q} = M_s^i \vec{P} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix} \begin{pmatrix} x_s \\ y_s \\ z_s \\ 1 \end{pmatrix}$$

We can express this in "canonical form" by dividing out the last coefficient:

$$M_s^i = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & 1 \end{pmatrix}$$

Notice that this gives 11 coefficients. This corresponds to our 11 parameters.

1.1 Tensor Notation:

In tensor notation, the sign " $\vec{\ }$ " is replaced by subscripts and superscripts.
A super-script signifies a column vector.
For example the point $\vec{P}$ is P^i

$$P^i = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix}$$

The line $\vec{L}^T$ is $L_i = (l_1, l_2, l_3)$

A matrix is a line matrix of column matrices (or a column of line matrices)

$$M_i^j = \begin{pmatrix} m_1^1 & m_2^1 & m_3^1 \\ m_1^2 & m_2^2 & m_3^2 \\ m_1^3 & m_2^3 & m_3^3 \end{pmatrix}$$

When homogeneous coordinates are used to represent transforms, these indices can be used to indicate the reference frame.

For example: A transformation from the scene "s" to the image "i"
is a 3 x 4 matrix M

$$M_s^i = \begin{pmatrix} m_1^1 & m_2^1 & m_3^1 & m_4^1 \\ m_1^2 & m_2^2 & m_3^2 & m_4^2 \\ m_1^3 & m_2^3 & m_3^3 & 1 \end{pmatrix}$$

The sub/super scripts indicate the source and destination reference frames.
M_s^i is a transformation from "S" (Scene) to "i" (image).

Einstein summation convention:

The summation symbol is implicit when a superscript adn subscript have the same letter.

$$L_i P^i = l_1 p^1 + l_2 p^2 + l_3 p^3$$

for a matrix and vector, this gives a new vector:

$$p^j = T_i^j p^i$$

This example transforms the point *p* in reference i to a point *p* in reference j.

2 Cross Products

2.1 Two point make a line

A line is a set of points that satisfies a constraint. The constraint is the line equation.

$$\vec{L}^T\vec{P} = ax + by + c = 0$$

The constraint can be determined from any two points on the line $P_1 = (x_1,y_1)$ and $P_2 = (x_2,y_2)$.

Classically, we subsitute P_1 and P_2 in the line equation to obtain two equations

$$ax_1 + by_1 + c = 0$$
$$ax_2 + by_2 + c = 0$$

We then solve the two equations for a, b and c. This should give :

$$a = (y_1 - y_2) \quad b = (x_2 - x_1)$$
$$c = -(a\,x_1 - b\,y_1) = -x_1(y_1 - y_2) - y_1(x_2 - x_1) = x_1y_2 - x_2y_1$$

if we normalize such that $\| (a, b) \| = 1$ then

$$d = ax + by + c$$

where d is a signed perpendicular distance from the line, positive to the left and negative to the right.

In fact the constraint imposed by the line can be discovered from the cross product of the two points:

$$L^T = \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \vec{P}_1 \times \vec{P}_2 = \begin{pmatrix} 0 & -1 & y_1 \\ 1 & 0 & -x_1 \\ -y_1 & x_1 & 0 \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} y_1 - y_2 \\ x_2 - x_1 \\ x_1y_2 - x_2y_1 \end{pmatrix}$$

This is equivalent to a setting the determinant to zero with free variables in one of the rows.

$$\begin{vmatrix} x & y & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix} = x(y_1 - y_2) + y(x_2 - x_1) + x_1y_2 - x_2y_1 = 0$$

Thus $a = (y_1 - y_2) \quad b = (x_2 - x_1)$
$c = \; = x_1y_2 - x_2y_1$

This is a simple example of a general principle that can be useful in higher dimensions.

2.2 Two lines make a point

There is a perfect duality between points and lines. So naturally, two lines make a point.

Given non-parallel lines L and M.

$$L:\ a_1x + b_1y + c_1 = 0$$

and

$$M:\ a_2x + b_2y + c_2 = 0.$$

In classic notation $L^T = (a_1, b_1, c_1)$ et $M^T = (a_2, b_2, c_2)$

The unique point is $x = \dfrac{b_1c_2 - b_2c_1}{a_1b_2 - a_2b_1}$ $\quad y = \dfrac{c_1a_2 - c_2a_1}{a_1b_2 - a_2b_1}$

The cross product is

$$\vec{P} = \vec{L} \times \vec{M} = \vec{P} = \vec{L} \times \vec{M} = \begin{pmatrix} 0 & -c_1 & b_1 \\ c_1 & 0 & -a_1 \\ -b_1 & a_1 & 0 \end{pmatrix}\begin{pmatrix} a_2 \\ b_2 \\ c_2 \end{pmatrix} = \begin{pmatrix} -c_1b_2 + b_1c_2 \\ c_1a_2 - a_1c_2 \\ -a_2b_1 + a_1b_2 \end{pmatrix} = \begin{pmatrix} b_1c_2 - c_1b_2 \\ c_1a_2 - a_1c_2 \\ a_1b_2 - a_2b_1 \end{pmatrix}$$

as before the cross product is equivalent to setting a determinant to zero. In this case:

$$\begin{vmatrix} a & b & c \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{vmatrix} = a(b_1c_2 - c_1b_2) + b(c_1a_2 - a_1c_2) + c(a_1b_2 - b_1a_2) = 0$$

To obtain the form $ax + by + c = 0$ we divide by $(a_1b_2 - a_2b_1)$ to get

$$a\frac{(b_1c_2 - c_1b_2)}{(a_1b_2 - b_1a_2)} + b\frac{(c_1a_2 - a_1c_2)}{(a_1b_2 - b_1a_2)} + c = 0$$

The unique point is $x = \dfrac{b_1c_2 - b_2c_1}{a_1b_2 - a_2b_1}$ $\quad y = \dfrac{c_1a_2 - c_2a_1}{c_1a_2 - c_2a_1}$

3 Coordinate Transforms in 2D

Homogeneous coordinates allow us to unify projective transformations using only matrix multiplication. This includes both Affine and Projective transformations

Euclidean Transformations:
- Isometries (Translation, Rotation)
- Scale Change

Affine Tranformations
- Isometry
- Sheer

Projective Transformation
- Homographies
- Camera Models

3.1 Translation and Rotation in Homogeneous Coordinates.

$x_2 = x_1 + t_X$,
$y_2 = y_1 + t_Y$

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

In tensorial notation:

$$Q^b = T_a^b P^a \qquad \text{for a,b = 1, 2, 3.}$$

Thus T_a^b is a translation from the "a" reference frame to the "b" reference frame.

$$\begin{pmatrix} q^1 \\ q^2 \\ q^3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix}$$

where $x_1=q^1/q^3$ and $x_2=q^2/q^3$

Similarly, rotation by an angle of θ is classicallywritten as:

$x_2 = \text{Cos}(\theta)\, x_1 - \text{Sin}(\theta)\, y_1$,
$y_2 = \text{Sin}(\theta)\, x_1 + \text{Cos}(\theta) y_1$

Or, in matrix form.

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} Cos(\theta) & -Sin(\theta) & 0 \\ Sin(\theta) & Cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

In tensor form with homogeneous coordinates:

$$Q^c = T_b^c P^b = \begin{pmatrix} q^1 \\ q^2 \\ q^3 \end{pmatrix} = \begin{pmatrix} Cos(\theta) & -Sin(\theta) & 0 \\ Sin(\theta) & Cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix}$$

Combining translation and rotation gives

$$x_2 = \text{Cos}(\theta)\ x_1 - \text{Sin}(\theta)\ y_1 + t_x$$
$$x_2 = \text{Sin}(\theta)\ x_1 = \text{Cos}(\theta)\ y_1 + t_y$$

which can be expressed as either:

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} Cos(\theta) & -Sin(\theta) & t_x \\ Sin(\theta) & Cos(\theta) & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

or as

$$\begin{pmatrix} q^1 \\ q^2 \\ q^3 \end{pmatrix} = \begin{pmatrix} Cos(\theta) & -Sin(\theta) & t_x \\ Sin(\theta) & Cos(\theta) & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix}$$

Note, in such transforms, weexpress the position of the origin of the source coordinates O^1 in the destination coordinates.

3.2 Similitude

If we write :

$$\begin{pmatrix} x_2/s \\ y_2/s \\ 1 \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \\ s \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

this gives a scale change of 1/s : $x_2 = x_1/s$

alternatively we can write:

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} sx_1 \\ sy_1 \\ 1 \end{pmatrix} = \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

so that $x_2 = sx_1$.

The full similitude (translation, rotation and scale change) is:

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} sCos(\theta) & -sSin(\theta) & t_x \\ sSin(\theta) & sCos(\theta) & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

3.3 Affine transformations:

If we write

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} s_x x_1 \\ s_y y_1 \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

We obtain a change in scale of axes.

The full isometric transformation is:

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} s_x Cos(\theta) & -s_y Sin(\theta) & t_x \\ s_x Sin(\theta) & s_y Cos(\theta) & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

The complete affine tranformation

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \text{ or } \begin{pmatrix} q^1 \\ q^2 \\ q^3 \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix} \text{ or } Q^b = A_a^b P^a$$

The affine transform includes similitude and isometry as a special cases, but also includes sheer.

3.4 Projection between two planes (Homography)

The projective transformation from one plane to another is called a homography.
A homography is bijective (reversible).

as a matrix :

$$\begin{pmatrix} x_b \\ y_b \\ 1 \end{pmatrix} = \begin{pmatrix} wx_b \\ wy_b \\ w \end{pmatrix} = \begin{pmatrix} h_1^1 & h_2^1 & h_3^1 \\ h_1^2 & h_2^1 & h_3^1 \\ h_1^3 & h_2^3 & 1 \end{pmatrix} \begin{pmatrix} x_a \\ y_a \\ 1 \end{pmatrix}$$

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

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

In tensor notation

$$Q^B = H_A^B \; P^A$$

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

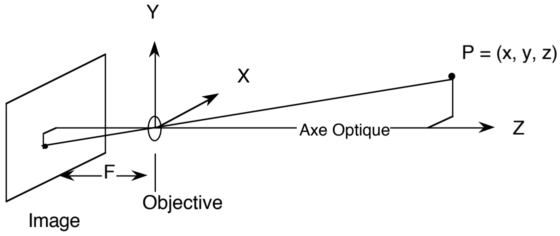
$$x_B = \frac{q^1}{q^3} \qquad y_B = \frac{q^2}{q^3}$$

4 The Camera Model

A "camera" is a closed box with an aperature (a "camera obscura"). Photons are reflected from the world, and pass through the aperture to form an image on the retina. Thus the camera coordinate system is defined with the aperture at the origin.

The Z (or depth) axis runs perpendicular from the retina through the aperture.
The X and Y axes define coordinates on the plane of the aperture.

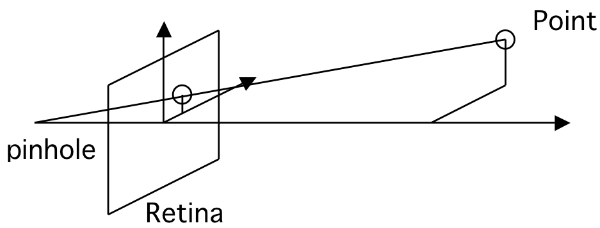
4.1 The Pinhole Camera



Points in the scene are projected to an "up-side down" image on the retina.

This is the "Pin-hole model" for the camera.

The scientific community of computer vision often uses the "Central Projection Model". In the Central Projection Model, the retina is placed in Front of the projective point.



We will model the camera as a projective transformation from scene coordinates, S, to image coordinates, i.

$$\vec{Q}^{i} = \mathbf{M}_{s}^{i} \vec{P}^{s}$$

This transformation is expressed as a 3x4 matrix:

$$M_s^i = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{pmatrix}$$

composed from 3 transformations between 4 reference frames.

4.2 Extrinsic and Intrinsic camera parameters

The camera model can be expressed as a function of 11 parameters.
These are often separated into 6 "extrinsic" parameters and 5 "intrinsic" parameters:

Thus the "extrinsic" parameters of the camera describe the camera position and orientation in the scene. These are the six parameters:

Extrinsic Parameters = $(x, y, z, \theta, \varphi, \gamma)$

The intrinsic camera parameters express the projection to the retina, and the mapping to the image. These are :

F : The "focal" length
C_x, C_y : the image center (expressed in pixels).
D_x, D_y : The size of pixels (expressed in pixels/mm).

4.3 Coordinate Systems

This transformation can be decomposed into 3 basic transformations between 4 reference frames. The reference frames are:

Coordinate Systems:

Scene Coordinates:

Point Scène: $P^s = (x_s, y_s, z_s, 1)^T$

Camera Coordinates:

external world: $P^c = (x_c, y_c, z_c, 1)^T$

Retina: $Q^r = (x_r, y_r, 1)^T$

Image Coordaintes

Image: $Q^i = (i, j, 1)^T$

The transformations are represented by Homogeneous projective transformations.

$$\vec{Q}^i = C_r^i \vec{Q}^r \qquad \vec{Q}^r = P_c^r \vec{P}^c \qquad \vec{P}^c = T_s^c \vec{P}^s$$

These express
1) A translation/rotation from scene to camera coordinates: T_s^c
2) A projection from scene points in camera coordinates to the retina: P_c^r
3) Sampling scan and A/D conversion of the retina to give an image: C_r^i

When expressed in homogeneous coordinates, these transformations are composed as matrix multiplications.

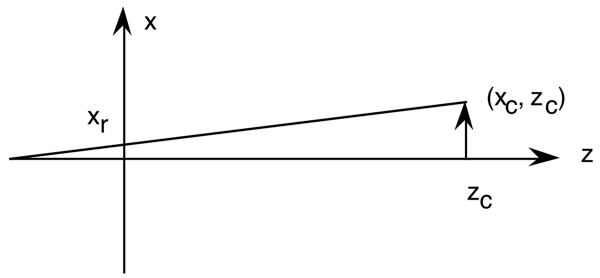
$$\vec{Q} = M_s^i \vec{P} = C_r^i P_c^r T_s^c \vec{P}$$

We will use "tensor notation" to keep track of our reference frames:

4.4 Projective Transforms: from the scene to the retina

Projection through an aperture is a projective transformation

Consider the central projection model for a 1D camera:



In camera coordinates:

$P^c = (x_c, y_c, z_c, 1)^T$ is a scene point in camera (aperture centered) coordinates
$Q^r = (x_r, y_r, 1)^T$ is a point on the retina.

By similar triangles:

$$\frac{x_r}{F} = \frac{x_c}{z_c} \Leftrightarrow x_r = x_c \frac{F}{z_c} \Leftrightarrow x_r \frac{z_c}{F} = x_c$$

$$\frac{y_r}{F} = \frac{y_c}{z_c} \Leftrightarrow y_r = y_c \frac{F}{z_c} \Leftrightarrow y_r \frac{z_c}{F} = y_c$$

Assume: $w = \frac{z_c}{F}$

then: $w\, x_r = x_c$
$w\, y_r = y_c$
$w = \frac{z_c}{F}$

As a matrix:

$$\begin{pmatrix} wx_r \\ wy_r \\ w \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{F} & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} wx_r \\ wy_r \\ w \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{F} & 0 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix}$$

The transformation from Scene points in camera coordinates to retina points is:

$$Q^r = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{F} & 1 \end{pmatrix} \begin{pmatrix} p^1 \\ p^2 \\ p^3 \\ 1 \end{pmatrix} = P_c^r \vec{P}^c$$

and $\begin{pmatrix} x_r \\ y_r \\ 1 \end{pmatrix} = \begin{pmatrix} q_1/q_3 \\ q_2/q_3 \\ 1 \end{pmatrix}$

thus:

$$P_c^r = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{F} & 1 \end{pmatrix}$$

Note that P_c^r is not invertible.

Remark: If we place the origin in the retina:

$$\frac{x_r}{F} = \frac{x_c}{(F + z_c)} \qquad => \qquad x_r = \frac{x_c\, F}{(F + z_c)}$$

Which gives:

$$x_r = \frac{x_c\, F}{(F+z_c)} \qquad\qquad y_r = \frac{y_c\, F}{(F+z_c)}$$

$$z_r = 0$$

and thus : $P_c^r = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{F} & 1 \end{pmatrix}$

4.5 From Scene to Camera

The following matrix represents a translation Δx, Δy, Δz and a rotation R.

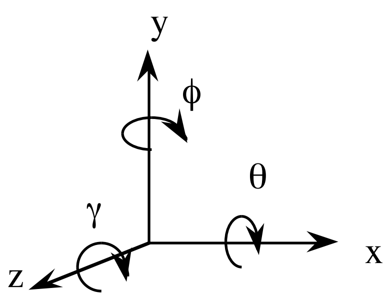
$$T_s^c = \begin{pmatrix} & & & \Delta x \\ & R & & \Delta y \\ & & & \Delta z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The transformation is composed by expressing the position of the source reference frame in the destination reference frame.

The rotation part is a 3x3 matrix that can be decomposed into 3 smaller rotations.

$\mathbf{R} = \mathbf{R}_z(\gamma)\mathbf{R}_y(\varphi)\mathbf{R}_x(\theta)$

En 3D



$\mathbf{R}_x(\theta)$ is a rotation around the x axis.

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix}$$

$$R_y(\varphi) = \begin{pmatrix} \cos(\varphi) & 0 & \sin(\varphi) \\ 0 & 1 & 0 \\ -\sin(\varphi) & 0 & \cos(\varphi) \end{pmatrix}$$

$$R_z(\gamma) = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Scale Change:

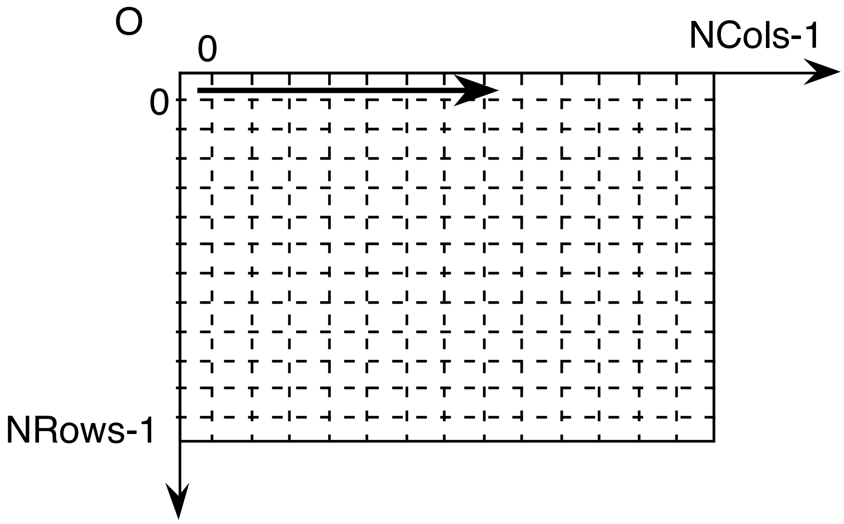
We can change the scale of each axis with a scale transformation

$$S_i^j = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

4.6 From the Retina to Digitized Image

The "intrinsic parameters of the camera are F and C_x, C_y, D_x, D_y

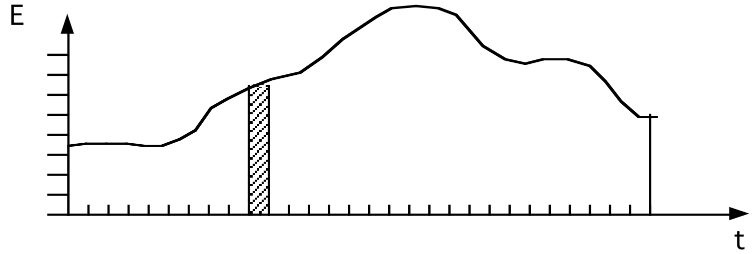
The image frame is composed of pixels (picture elements)



Note that pixels are not necessarily square.

Typical image sizes VGA : 640 x 480

Sampling and A/D Conversion.



The mapping from retina to image can be expressed with 4 parameters:

C_x, C_y : the image center (expressed in pixels).
D_x, D_y : The size of pixels expressed in pixels/mm.

$i = x_r D_i$ (mm · pixel/mm) + C_i (pixel)

$j = y_r D_j$ (mm · pixel/mm) + C_j (pixel)

Transformation from retina to image :

$$Q^i = \mathbf{C}_r^i \; Q^r$$

$$\begin{pmatrix} i \\ j \\ 1 \end{pmatrix} = \begin{pmatrix} D_i & 0 & C_i \\ 0 & D_j & C_j \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_r \\ y_r \\ 1 \end{pmatrix}$$

4.7 The Complete Camera Model

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

$$Q^i = M_s^i P^s$$

$$\begin{pmatrix} wi \\ wj \\ w \end{pmatrix} = \begin{pmatrix} m_1^1 & m_2^1 & m_3^1 & m_4^1 \\ m_1^2 & m_2^2 & m_3^2 & m_4^2 \\ m_1^3 & m_2^3 & m_3^3 & m_4^3 \end{pmatrix} \begin{pmatrix} x_s \\ y_s \\ z_s \\ 1 \end{pmatrix}$$

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

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

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