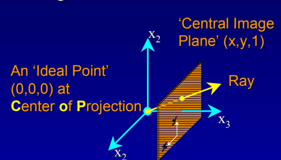
CS 395/495-26: Spring 2002

IBMR: Week 2 B
2-D Projective Geometry

Jack Tumblin
jet@cs.northwestern.edu

Useful 3D Graphics Ideas

- Every xy point describes a 3D ray in (x_1,x_2,x_3)
- 'Phantom' dimension x_3 is the 'z-buffer' value
- Homogeneous coords allows translation matrix:



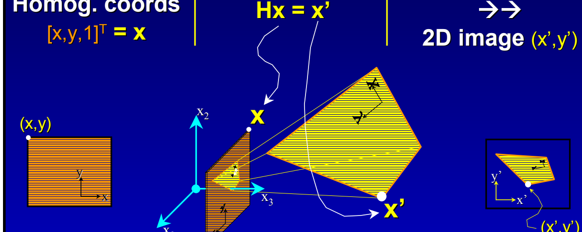
$$\begin{bmatrix}1&0&3\\0&1&5\\0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix}=\begin{bmatrix}x+3\\y+5\\1\end{bmatrix}$$

- (**BUT** a 3x3 transform in $\mathbf{R}^3$ Euclidean Space (x,y,z) can't translate—only rotate,scale, skew!)

$$\begin{bmatrix}1&0&3\\0&1&5\\0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\z\end{bmatrix}=\begin{bmatrix}x+z\\y+z\\z\end{bmatrix}$$

Projective Transform H

2D image (x,y) →→ Homog. coords $[x,y,1]^T$ = **x**	Apply the 3x3 matrix **H** **Hx = x'**	Homog. coords **x' =** $[x',y',1]^T$ →→ 2D image (x',y')



Projective Transform: Hx = x’

Finding H from point pairs (correspondences)

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{21} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} \qquad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Input (or output) image is on central plane

$$x' = \frac{x'_1}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}$$

$$y' = \frac{x'_2}{x'_3} = \frac{h_{21}x + h_{22}y + h_{21}}{h_{31}x + h_{32}y + h_{33}}$$

Write R^2 expressions;

$$x'(h_{31}x + h_{32}y + h_{33}) = (h_{11}x + h_{12}y + h_{13})$$
$$y'(h_{31}x + h_{32}y + h_{33}) = (h_{11}x + h_{12}y + h_{13})$$

Rearrange, solve as a matrix problem…

Projective Transform: Hx = x’

Finding H from point pairs (correspondences)

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{21} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} \qquad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Input (or output) image is on central plane

$$x' = \frac{x'_1}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + \ldots}$$

$$y' = \frac{x'_2}{x'_3} = \frac{h_{21}x + \ldots}{h_{31}x + \ldots}$$

But don’t do this—you’ll get better results using l_∞ or conics methods…

expressions;

$$x'(h_{31}x + h_{32}y + h_{33}) = (h_{11}x + h_{12}y + h_{13})$$
$$y'(h_{31}x + h_{32}y + h_{33}) = (h_{11}x + h_{12}y + h_{13})$$

Rearrange, solve as a matrix problem…

Projective Transform: H

- Can also transform Lines:
  - Recall point **x** is on line **l** iff $\mathbf{x}^T\mathbf{l} = 0 = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$
  - Lines transform ‘covariantly’
    (points: ‘contravariantly’: x’ = Hx)

$$\mathbf{l}' = \mathbf{H}^{-T}\mathbf{l}$$

- And transform Conics:
  - Recall point **x** is on a conic curve defined by **C** iff $\mathbf{x}^T\mathbf{C}\mathbf{x} = 0$

$$\mathbf{C}' = \mathbf{H}^{-T}\,\mathbf{C}\,\mathbf{H}^{-1}$$

Projective Transform: H

- Comp. Graphics 'View interpolation' notion
  - Fixed, rigid 2D viewing point, viewing plane
  - Rigid 2D world plane positioned in 3D (x_1 x_2 x_3)
  - **only 6 DOF**: world plane rotate & position.
- Comp. Vision 'Projective Transform' notion
  - Fixed, rigid 2D viewing point, viewing plane
  - 'Stretchy' 2D world plane: allow affine changes
  - result: (up to) **8 DOF**

The bits and pieces of H

H has 8 independent variables (DOF)

- Computer Graphics method (3x3 matrix):
  - 2D Translation(t_x, t_y)
  - 3D Scale (s_x,s_y,s_z)
  - 3D Rotation (θ_x,θ_y,θ_z)
- Computer Vision method(2D projective):
  - Isometry--3DOF(2D translate t_x,t_y; 2Drotate θ_z;)
  - Similarity--4DOF (add uniform scale s;)
  - Affine --6DOF (add orientable scale s_θ/s, $s_{\theta\perp}$/s)
  - Projective--8DOF (changes x_3; 3D-rotation-like)

The bits and pieces of H

H has 8 independent variables (DOF)

- Computer Graphics method (3x3 matrix):
  - 2D Translation(t_x, t_y)
  - 3D Scale (s_x,s_y,s_z)
  - 3D Rotation (θ_x,θ_y,θ_z)
- Computer Vision method(2D projective):
  - Isometry--3DOF(2D translate t_x,t_y; 2Drotate θ_z;)
  - Similarity--4DOF (add uniform scale s;)
  - Affine --6DOF (add orientable scale s_θ/s, $s_{\theta\perp}$/s)
  - Projective--8DOF (changes x_3; 3D-rotation-like)

Affects only x1,x2

The bits and pieces of H

H has 8 independent variables (DOF)

- Decomposable into 3 useful parts: (pg 22)

$$H = H_S\, H_A\, H_P = \left[\begin{array}{cc|c} sR & \cdot & t \\ \cdot & \cdot & \cdot \\ \hline 0^T & \cdot & 1 \end{array}\right] \left[\begin{array}{cc|c} K & \cdot & 0 \\ \cdot & \cdot & \cdot \\ \hline 0^T & \cdot & 1 \end{array}\right] \left[\begin{array}{cc|c} I & \cdot & 0 \\ \cdot & \cdot & \cdot \\ \hline v^T & \cdot & v \end{array}\right]$$

- Similarity H_S:
  - 2D translate, rotate, uniform scale only (4DOF)
- Affine H_A:
  - non-uniform scale (2DOF)
- Similarity H_P:
  - Projective coupling for x_3 (2DOF)

The bits and pieces of H

H has 8 independent variables (DOF)

- Decomposable into 3 useful parts: (pg 22)

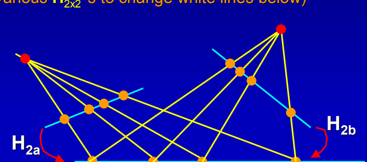
$$H = H_S\, H_A\, H_P = \left[\begin{array}{cc|c} s\,\varepsilon\cos\theta & -s\sin\theta & t_x \\ s\,\varepsilon\sin\theta & s\cos\theta & t_y \\ \hline 0 & 0 & 1 \end{array}\right] \left[\begin{array}{cc|c} k1 & k3 & 0 \\ k2 & k3 & 0 \\ \hline 0 & 0 & 1 \end{array}\right] \left[\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \hline v_1 & v_2 & v_S \end{array}\right]$$

- Similarity H_S:
  - 2D translate, rotate, uniform scale only (4DOF)
- Affine H_A:
  - non-uniform scale (2DOF)
- Similarity H_P:
  - Projective coupling for x_3 (2DOF)

1-D Projective Geometry

(?Why? we use it later)

- A 'side' view of 2D projective geometry
- Convert R^1 scalar **b** to a 2-vector $\begin{bmatrix} b \\ 1 \end{bmatrix}$ in P^1
- As with P^2, we can transform points:
  - (use various H_{2x2}'s to change white lines below)



1-D Projective Geometry

What is invariant here?

all the positions, lengths and ratios change with each transform…

Answer: 'Cross Ratio': cross(a,b,c,d)

- let a= $\begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$, b= $\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$ …
- THEN:
- cross = $\dfrac{|ab||cd|}{|ac||bd|}$

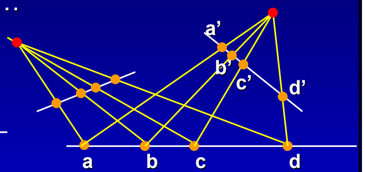


Image Rectifying: Undo parts of H

x' = H x where $H = H_S H_A H_P$

GOAL: Put world plane x' into view plane

- Affine Rect.; (find only H_P (2DOF));
- Similarity Rect.; (find only $H_A H_P$ (6DOF));
- Full Rect.; (find all of H (8DOF));

METHODS: (mix & match?)

1. Full: 4-point correspondence
2. 'Vanishing Point', Infinity line methods
3. Conics and circular points

Image Rectifying: Undo parts of H

x' = H x where $H = H_S H_A H_P$

GOAL: Put world plane x' into view plane

- Affine Rect.; (find only H_P (2DOF));
- Similarity Rect.; (find only $H_A H_P$ (6DOF));
- Full Rect.; (find all of H (8DOF));

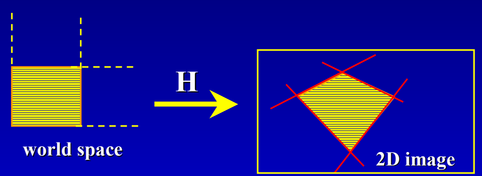
METHODS: (mix & match?)

1. Full: 4-point correspond…
2. 'Vanishing Point', Infinity…
3. Conics and circular points

Other DOF? make assumptions, or ignore

Vanishing-Point Methods: 1

– In 2D image, find two pairs of lines that are parallel in 'world' space.

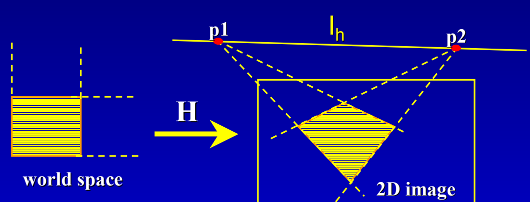


Vanishing-Point Methods: 1

– Find intersection (vanishing points) p1, p2

(compute line intersections with 3D cross-products (see last lecture)

– Horizon line l_h connects p1, p2.

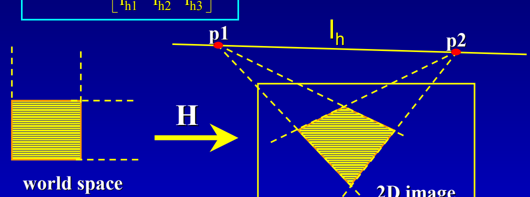


Vanishing-Point Methods: 1

– $\mathbf{H}^{-T}$ transforms world infinity line $\mathbf{l}_\infty \equiv (0,0,1)^T$ to l_h

– Answer (see book) copy the l_h coordinates!

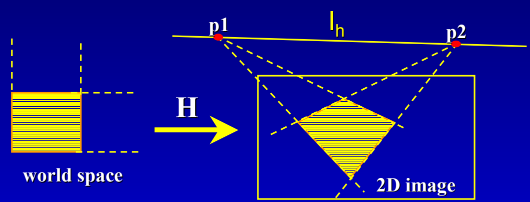
$$\mathbf{H}_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_{h1} & l_{h2} & l_{h3} \end{bmatrix}$$



Vanishing-Point Methods: 1

Limitations:

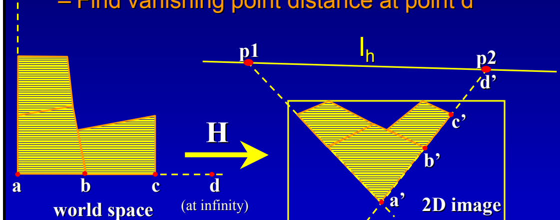
- Accurate point, line-finding can be tricky
- Can have high error sensitivity: ‘twitchy’
- Only rectifies H_p– what if H_A, H_S needed?



Vanishing-Point Methods: 2

No parallel lines?

- find 2D image line with a known distance ratio
- Use Cross-Ratio (in $\mathbf{P}^1$ along that line) to
- Find vanishing point distance at point d’



Conic Methods

- Better-behaved, easier to use(?)
- Determines H_A H_P (4DOF) (all but 2D trans, rot, scale)

Go back and review conics first (pg.8)

- ‘Conics’ == intersection of cone & plane:
- Many possible shapes:
  circles, ellipses, parabola, hyperbola,
  degenerates (lines & points

Conic Methods

- Equation of any/all conics solve a 2D quadratic:
  $ax^2 + bxy + cy^2 + dx + ey + f = 0$
- Write in homogeneous coordinates:

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$$

$$\mathbf{x}^T\mathbf{C}\mathbf{x} = 0$$

- C is symmetric, 5DOF (because x_3 scaling)
- Find any **C** from 5 homogeneous points

a 'Point Conic'

(solve for null space—see book pg 9)

Conic Methods

- Matrix C makes conics from points:
  $\mathbf{x}^T\mathbf{C}\mathbf{x} = 0$ **C** is a **'point conic'**
- Given a point **x** on a conic curve, the homog. tangent line **l** is given by **l = C x**
- Matrix C* makes conics from lines:
  $\mathbf{l}^T\mathbf{C}^*\mathbf{l} = 0$ **C* is a 'Dual Conic'**
  defined by tangent lines **l** instead of points.

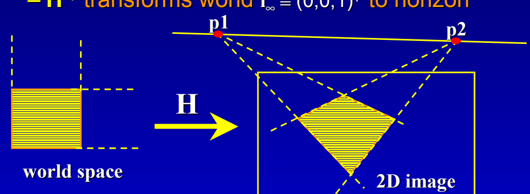
Conic Methods

- If **C** is non-singular (rank 3), then $\mathbf{C}^* = \mathbf{C}^{-1}$
- If **C** (or **C***) has…
  Rank 3: it is an ellipse, circle, parab., hyperb.
  Rank 2: it is a pair of lines (forms an 'x')
- Projective transform of a conic **C** is conic **C'**:
  $\mathbf{C}' = \mathbf{H}^{-T}\,\mathbf{C}\,\mathbf{H}^{-1}$

END

Vanishing-Point Methods: 1

- In 2D image, find two pairs of lines that are parallel in ‘world’ space.
- Find intersection (vanishing points) p1, p2
- Horizon line connects p1’, p2’
- $\mathbf{H}^{-T}$ transforms world $\mathbf{l}_\infty \equiv (0,0,1)^T$ to horizon



The bits and pieces of H

H has 8 independent variables (DOF)

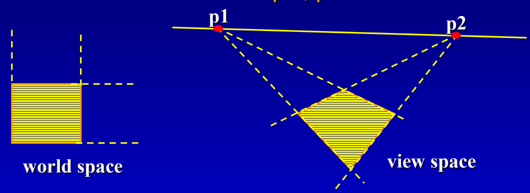
- Decomposable into 3 useful parts: (pg 22)

$$H_S \, H_A \, H_P = \begin{bmatrix} s\,\varepsilon\cos\theta & -s\sin\theta & t_x \\ s\,\varepsilon\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} k1 & k2 & 0 \\ k2 & k3 & \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ v1 & v2 & v \end{bmatrix}$$

- Similarity H_S:
  - 2D translate, rotate, uniform scale only (4DOF)
- Affine H_A:
  - non-uniform scale (2DOF)
- Similarity H_P:
  - Projective coupling for x_3 (2DOF)

Vanishing-point Methods

- In 2D image, find two pairs of parallel lines.
- Find intersection (vanishing points) p1, p2
- H transforms them to 'ideal' points' p1',p2'
  p1' = H p1 = $(x1_1, x1_2, 0)^T$, p2' = Hp2 = $(x2_1, x2_2, 0)^T$
- Horizon line connects p1', p2'



Projective Transform H

Finding H from point pairs (correspondences)

- We know that **Hx' = x**, and
- we know at least 4 point pairs x' and x that satisfy it:
- **ATTEMPT 1:** 'plug & chug' make a matrix of **x** and x' values…

H x' = x

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{21} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\left[\mathbf{H}\right] \left[\mathbf{x'_1} \mid \mathbf{x'_2} \mid \mathbf{x'_3} \mid \mathbf{x'_4}\right] = \left[\mathbf{x_1} \mid \mathbf{x_2} \mid \mathbf{x_3} \mid \mathbf{x_4}\right]$$

Projective Transform: Hx = x'

Finding H from point pairs (correspondences)

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{21} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Input (or output) image is on central plane

$$\frac{x_1h_{21} \;\; x_2h_{22} \;\; x_3h_{21}}{x_1h_{31} \;\; x_2h_{32} \;\; x_3h_{33}}$$

x_1h_{11} x_2h_{12} x_3h_{13} y'
x_1h_{31} x_2h_{32} x_3 h_{33}

x_1h_{11} x_2h_{12} x_3h_{13}
x_1h_{21} x_2h_{22} x_3h_{21}
x'=
x_1h_{31} x_2h_{32} x_3 h_{33}

$$\left[\mathbf{H}\right] \left[\mathbf{x'_1} \mid \mathbf{x'_2} \mid \mathbf{x'_3} \mid \mathbf{x'_4}\right] = \left[\mathbf{x_1} \mid \mathbf{x_2} \mid \mathbf{x_3} \mid \mathbf{x_4}\right]$$

y'

Projective Transform H

Finding H from point pairs (correspondences)

- We know that **Hx' = x**, and
- we know at least 4 point pairs x' and x that satisfy it:
- **ATTEMPT 1:** 'plug & chug' make a matrix of **x** and x' values…

H x' = x

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{21} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

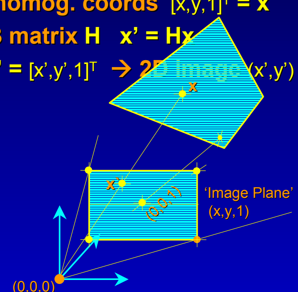
UNKNOWN!

$$\mathbf{H} \begin{bmatrix} \mathbf{x'_1} & \mathbf{x'_2} & \mathbf{x'_3} & \mathbf{x'_4} \end{bmatrix} = \begin{bmatrix} \mathbf{x_1} & \mathbf{x_2} & \mathbf{x_3} & \mathbf{x_4} \end{bmatrix}$$

2D Projective Transform

View Interpolation ('Projective Transform')

- **2D image** (x,y) → **homog. coords** $[x,y,1]^T$ **= x**
- **Transform by 3x3 matrix H x' = Hx**
- **homog. coords x' =** $[x',y',1]^T$ → **2D image** (x',y')



H x' = x

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{21} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Projective Transform H

H matrix: Plane-to-Plane mapping

H x' = x

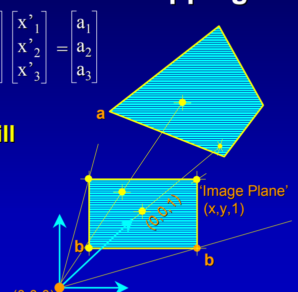
$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{21} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

$x' = x_1'/x_3'$

$y' = x_2'/x_3'$

Find H matrix that will move screen points from (x',y') **to** (x,y)

(CAREFUL! we don't know x_1, x_2, x_3 or x_1', x_2', x_3' !)



Projective Transform H

H matrix: Plane-to-Plane mapping

H x' = x

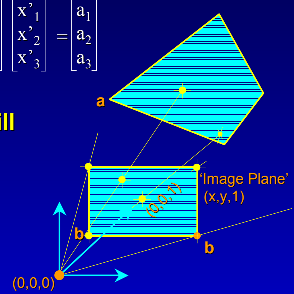
$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{21} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

$x' = x_1'/x_3'$

$y' = x_2'/x_3'$

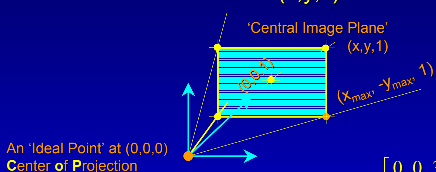
Find H matrix that will move screen points from (x',y') **to** (x,y)

(CAREFUL! we don't know x_1, x_2, x_3 or x_1', x_2', x_3' !)



Useful 3D Graphics Ideas

- Every xy point describes a 3D ray in (x_1,x_2,x_3)
- 'Phantom' dimension x_3 is the 'z-buffer' value
- Homogeneous coords allows translation matrix: **BUT** a 3x3 transform in (x,y,z) can't do translation



$$\begin{bmatrix} 0 & 0 & 3 \\ 0 & 0 & 5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x+3 \\ y+5 \\ 1 \end{bmatrix}$$

Projective Transform