2: Vanishing points and horizons. Applications of projective transformations.

• Lecture 1: Euclidean, similarity, affine and projective transformations. Homogeneous coordinates and matrices. Coordinate frames. Perspective projection and its matrix representation.

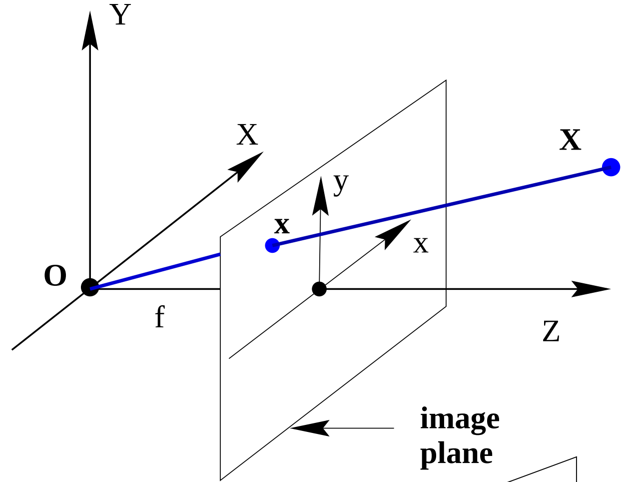
• Lecture 2: **Vanishing points. Horizons. Applications of projective transformations.**

• Lecture 3: Convexity of point-sets, convex hull and algorithms. Conics and quadrics, implicit and parametric forms, computation of intersections.

• Lecture 4: Bezier curves, B-splines. Tensor-product surfaces.

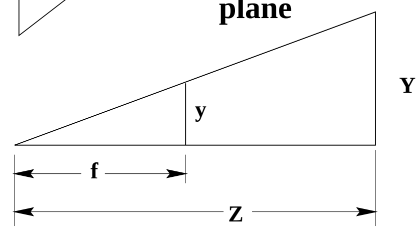
Recall: Perspective (central) projection — 3D to 2D 2.1

The camera model Mathematical idealized camera 3D $\rightarrow$ 2D
- Image coordinates xy
- Camera frame XYZ (origin at optical centre)
- Focal length f, image plane is at $Z = f$.



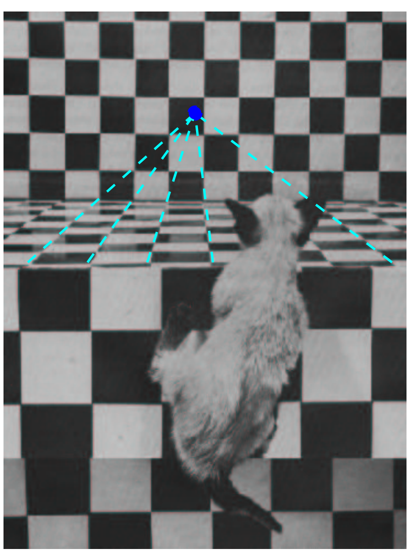
Similar triangles

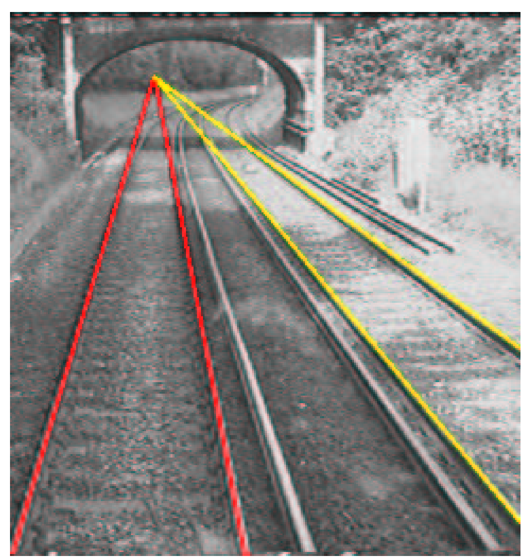
$$\frac{x}{f} = \frac{X}{Z} \qquad \frac{y}{f} = \frac{Y}{Z} \qquad \text{or} \quad \mathbf{x} = f\frac{\mathbf{X}}{Z}$$



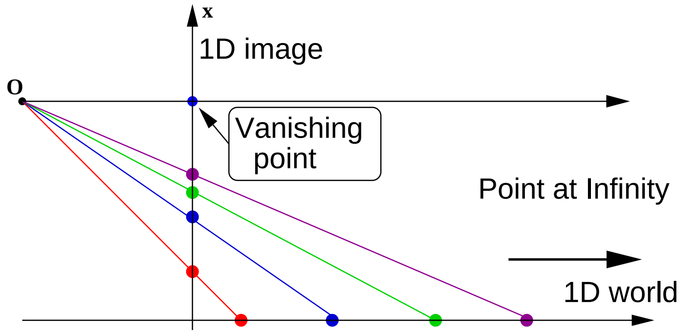
where $\mathbf{x}$ and $\mathbf{X}$ are **3-vectors**, with
$\mathbf{x} = (x, y, f)^\top$, $\mathbf{X} = (X, Y, Z)^\top$.

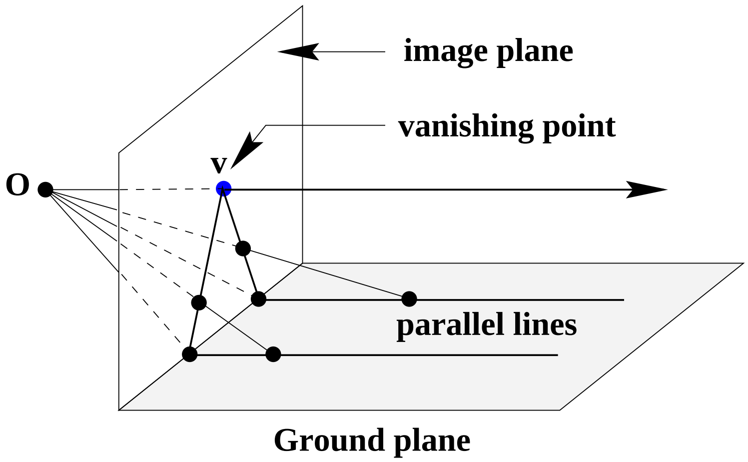
Vanishing Points 2.2

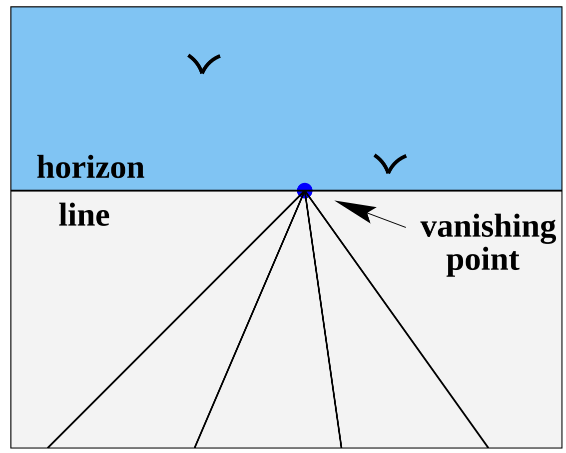




Vanishing Points in 1D and 2D 2.3



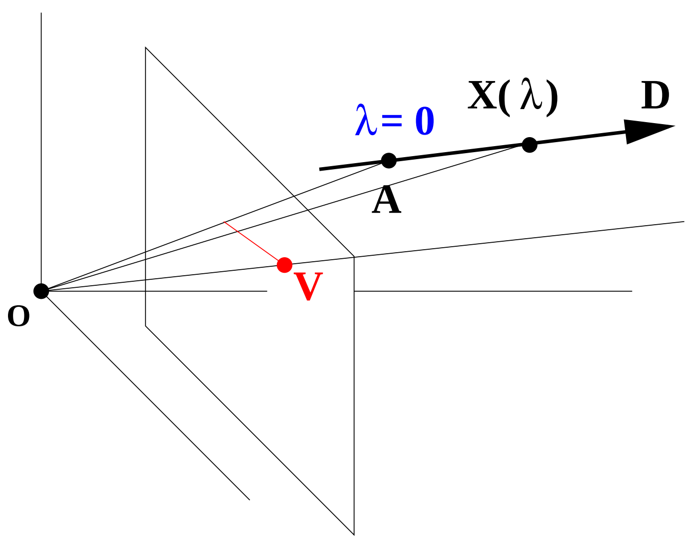




All parallel lines meet at the same vanishing point 2.4

A line of 3D points isrepresented as

$$\mathbf{X}(\lambda) = \mathbf{A} + \lambda \mathbf{D}$$



Using $\mathbf{x} = f\mathbf{X}/Z$ the vanishing point of its image is

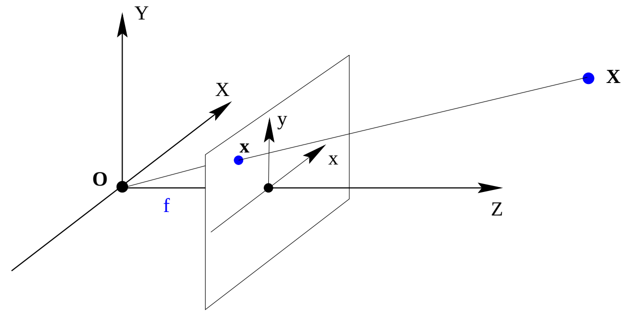
$$\begin{aligned} \mathbf{v} &= \lim_{\lambda \to \pm\infty} \mathbf{x}(\lambda) = f \frac{\mathbf{A} + \lambda \mathbf{D}}{A_Z + \lambda D_Z} \\ &= f \frac{\mathbf{D}}{D_Z} = f \begin{pmatrix} D_X / D_Z \\ D_Y / D_Z \\ 1 \end{pmatrix} \end{aligned}$$

- $\mathbf{v}$ depends only on the direction $\mathbf{D}$, not on $\mathbf{A}$.
- Parallel lines have the same vanishing point.

The Homogeneous 3×4 Projection Matrix 2.5

$$\mathbf{x} = f\frac{\mathbf{X}}{Z}$$

Choose $f = 1$ from now on.



Homogeneous image coordinates $(x_1, x_2, x_3)^\top$ correctly represent $\mathbf{x} = \mathbf{X}/Z$ if

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \doteq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = [\mathtt{I} \mid \mathbf{0}] \begin{pmatrix} \mathbf{X} \\ 1 \end{pmatrix}$$

because then

$$x = \frac{x_1}{x_3} = \frac{X}{Z} \qquad y = \frac{x_2}{x_3} = \frac{Y}{Z}$$

Then perspective projection is a linear map, represented by a 3×4 **projection matrix**, from 3D to 2D.

Vanishing points using homogeneous notation 2.6

A line of points in 3D through the point $\mathbf{A}$ with direction $\mathbf{D}$ is

$$\mathbf{X}(\mu) \doteq \mathbf{A} + \mu\mathbf{D}$$

Writing this in homogeneous notation

$$\begin{pmatrix} X_1(\mu) \\ X_2(\mu) \\ X_3(\mu) \\ X_4(\mu) \end{pmatrix} \doteq \begin{pmatrix} \mathbf{A} \\ 1 \end{pmatrix} + \mu \begin{pmatrix} \mathbf{D} \\ 0 \end{pmatrix} \doteq \frac{1}{\mu} \begin{pmatrix} \mathbf{A} \\ 1 \end{pmatrix} + \begin{pmatrix} \mathbf{D} \\ 0 \end{pmatrix}$$

In the limit $\mu \to \infty$ the point on the line is $\begin{pmatrix} \mathbf{D} \\ 0 \end{pmatrix}$

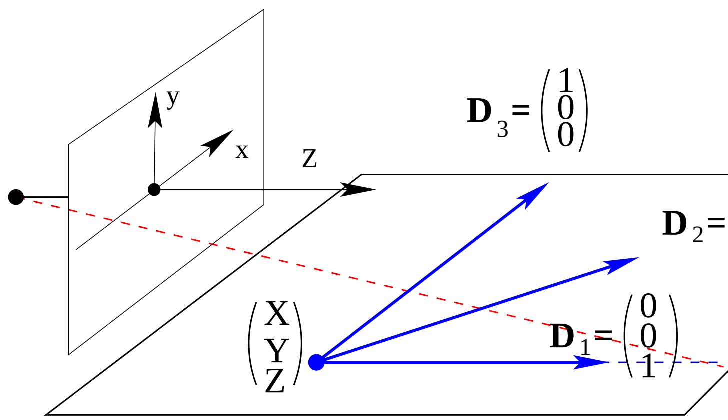
- So, homogeneous vectors with $X_4 = 0$ represent points "at infinity".
- **Points at infinity are equivalent to directions**

Vanishing points using homogeneous notation: Example

The vanishing point of a line with direction **D** is the image of the point at ∞ ...

$$\mathbf{v} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} \mathbf{D} \\ 0 \end{pmatrix} = \begin{pmatrix} D_X \\ D_Y \\ D_Z \end{pmatrix}$$

Exercise: Compute the vanishing points of lines on an XZ plane:
(1) parallel to the Z axis; (2) at $45°$ to the Z axix; (3) parallel to the X axis.



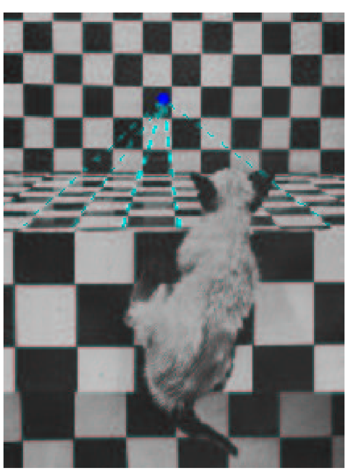
$$1: \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$2 \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

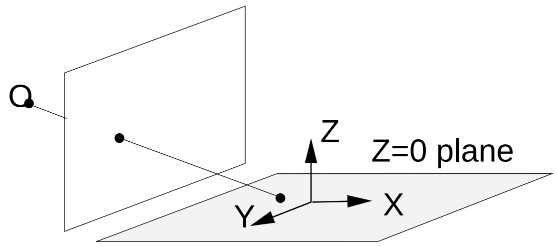
$$3: \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

The advantages of homogeneous notation ...

Jump you daft cat.



There are two advantages of using homogeneous notation to represent perspective projection:
1. Non-linear projections equations are turned into linear equations. The tools of linear algebra can then be used.
2. Vanishing points are treated naturally, and awkward limiting procedures are then avoided.



$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathtt{P} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix}$$

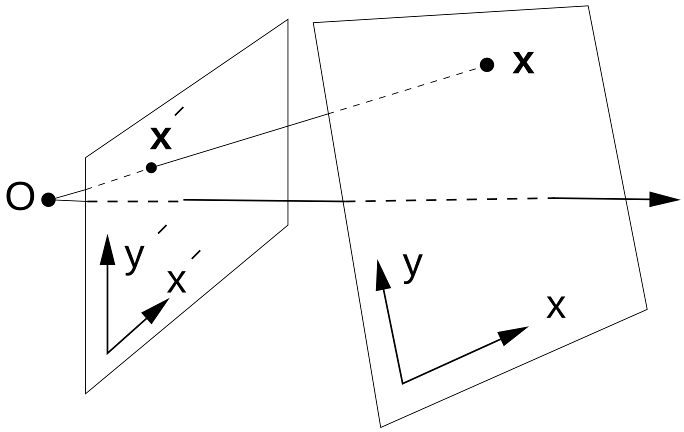
Choose the world coordinate system such that the world plane has zero Z co-ordinate. Then the 3×4 matrix $\mathtt{P}$ reduces to a 3×3 plane to plane **projective transformation**.

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

• This is the most general transformation between the world plane and image plane under imaging by a perspective camera.

• A projective transformation is also called a "homography" and a "collineation".

Computing a projective transformation 2.10



$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} \doteq \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

$$\doteq \mathtt{H} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

where H is a 3×3 non-singular homogeneous matrix with EIGHT degrees of freedom.

• Each point correspondence gives two constraints

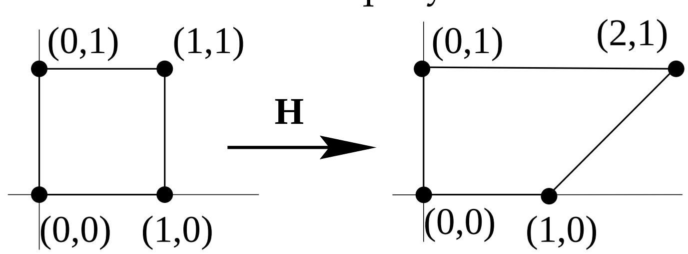
$$x' = \frac{x'_1}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \qquad y' = \frac{x'_2}{x'_3} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

and multiplying out give two equations *linear* in the elements of H

$$\begin{aligned} x'\,(h_{31}x + h_{32}y + h_{33}) &= h_{11}x + h_{12}y + h_{13} \\ y'\,(h_{31}x + h_{32}y + h_{33}) &= h_{21}x + h_{22}y + h_{23} \end{aligned}$$

Simple Example 2.11

• Suppose the correspondences $(x, y) \leftrightarrow (x', y')$ are known for **four** points (no three collinear), then H is determined uniquely.



First correspondence $(0, 0) \to (0, 0)$

$$\lambda_1 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} h_{13} \\ h_{23} \\ h_{33} \end{pmatrix}$$

Whence $h_{13} = h_{23} = 0$.

Second correspondence $(1, 0) \to (1, 0)$

$$\lambda_2 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & 0 \\ h_{21} & h_{22} & 0 \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} \\ h_{21} \\ h_{31} + h_{33} \end{pmatrix}$$

Whence $h_{21} = 0$ and $h_{11} = h_{31} + h_{33}$

Simple example Continued 2.12

Third correspondence $(0,1) \to (0,1)$ gives $h_{12} = 0$ and $h_{22} = h_{32} + h_{33}$.
Fourth correspondence $(1,1) \to (2,1)$

$$\lambda_4 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = \begin{bmatrix} h_{31} + h_{33} & 0 & 0 \\ 0 & h_{32} + h_{33} & 0 \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} h_{31} + h_{33} \\ h_{32} + h_{33} \\ h_{31} + h_{32} + h_{33} \end{pmatrix}$$

Take ratios $\Rightarrow$ 2 equations in 3 unknowns $\Rightarrow$ solve for ratio of matrix elements only.

$$\mathtt{H} = \lambda \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 2 \end{bmatrix} \doteq \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 2 \end{bmatrix}$$

Computational Algorithm 2.13

The equations,

$$\begin{aligned} x' \left(h_{31} x + h_{32} y + h_{33} \right) &= h_{11} x + h_{12} y + h_{13} \\ y' \left(h_{31} x + h_{32} y + h_{33} \right) &= h_{21} x + h_{22} y + h_{23} \end{aligned}$$

can be rearranged as

$$\begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x'x & -x'y & -x' \\ 0 & 0 & 0 & x & y & 1 & -y'x & -y'y & -y' \end{bmatrix} \mathbf{h} = \mathbf{0}$$

where $\mathbf{h} = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33})^\top$ is the matrix $\mathtt{H}$ written as a 9-vector.

continued ... 2.14

For 4 points,

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x_1 & -y'_1y_1 & -y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 & -x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y'_2x_2 & -y'_2y_2 & -y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x'_3x_3 & -x'_3y_3 & -x'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y'_3x_3 & -y'_3y_3 & -y'_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x'_4x_4 & -x'_4y_4 & -x'_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -y'_4x_4 & -y'_4y_4 & -y'_4 \end{bmatrix} \mathbf{h} = \mathbf{0}$$

which has the form $\mathtt{A}\mathbf{h} = \mathbf{0}$, with $\mathtt{A}$ a 8×9 matrix. The solution $\mathbf{h}$ is the (one dimensional) null space of $\mathtt{A}$.

If using many points, one can use least squares. Solution best found then using SVD of $\mathtt{A}$ — ie $\mathtt{USV}^\top \leftarrow \mathtt{A}$ Then $\mathbf{h}$ is the column of $\mathtt{V}$ corresponding to smallest singular value. (The smallest singular value would be zero of all the data were exact ...)

Some Matlab 2.15

```
npoints = 4  (or  5 later -- 5th point is noisy)
x  = [0,1,0,1, 1.01];   y  = [0,0,1,1, 0.99];
xd = [0,1,0,2, 2.01];  yd  = [0,0,1,1, 1.01];
A = zeros(2*npoints,9);
for i=1:npoints,
 A(2*i-1,:)= [x(i),y(i),1,0,0,0,  -x(i)*xd(i),-xd(i)*y(i),-xd(i)];
 A(2*i,  :)= [0,0,0,x(i),y(i),1,  -x(i)*yd(i),-yd(i)*y(i),-yd(i)];
end;
if npoints==4
  h = null(A);
else
  [U,S,V] = svd(A);
  h=V(:,9);
end;
H=[h(1),h(2),h(3);h(4),h(5),h(6);h(7),h(8),h(9);];
```

With the 4 exact points ...

$$\mathtt{H} = \begin{bmatrix} 0.6325 & -0.0000 & 0.0000 \\ 0.0000 & 1.0000 & -0.0000 \\ 0.0000 & -0.3162 & 0.6325 \end{bmatrix}$$

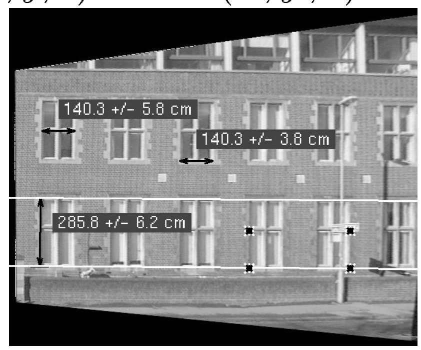
Adding the fifth noisy point ...

$$\mathtt{H} = \begin{bmatrix} 0.6295 & -0.0000 & -0.0000 \\ -0.0001 & 0.9999 & 0.0001 \\ -0.0050 & -0.3155 & 0.6344 \end{bmatrix}$$

Example 1: Removing Perspective Distortion 2.16

Objective: Back project to world plane

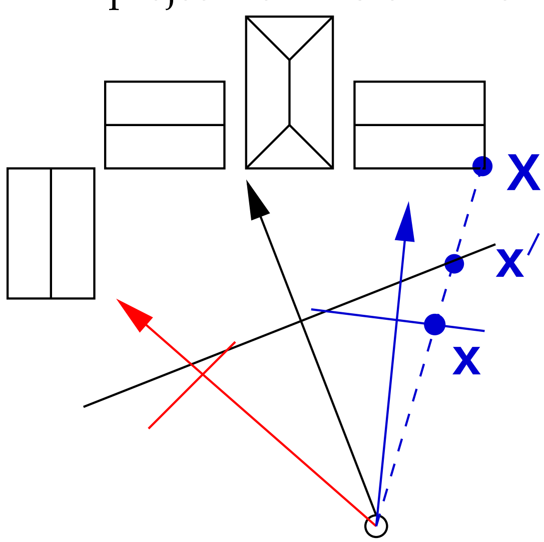
1. Find Euclidean coordinates of four points on the flat object plane $(x_i, y_i)^\top$.
2. Measure the corresponding image coordinates of these four points $(x'_i, y'_i)^\top$.
3. Compute H from the four $(x_i, y_i)^\top \leftrightarrow (x'_i, y'_i)^\top$.
4. Euclidean coords of any image point are $(x, y, 1)^\top = \mathtt{H}^{-1}(x', y', 1)^\top$.



The image can be warped onto the world plane using H. How?

Moving the image plane 2.17

An image is the intersection of a plane with the cone of rays between points in 3-space and the optical centre. Any two such "images" (with the same optical centre) are related by a planar projective transformation.



As the camera is rotated the points of intersection of the rays with the image plane are related by a planar projective transformation. Image points $\mathbf{x}$ and $\mathbf{x}'$ correspond to the same scene point $\mathbf{X}$.

Relationship in terms of rotations

For corresponding points $\mathbf{x}_1$ and $\mathbf{x}_2$ in two views 1 and 2,

$$\mathbf{x}_1 \doteq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathtt{R}_1 & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = \mathtt{R}_1 \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \qquad \mathbf{x}_2 \doteq \mathtt{R}_2 \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

Hence

$$\mathbf{x}_2 = \mathtt{R}_2 \mathtt{R}_1^{-1} \mathbf{x}_1$$

The cameras could have different focal lengths — so one can do all of this while rotationing *and* zooming. Then

$$\mathbf{x}_2 = \mathtt{K}_2 \mathtt{R}_2 \mathtt{R}_1^{-1} \mathtt{K}_1^{-1} \mathbf{x}_1$$

where in the simplest case

$$\mathtt{K}_i = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

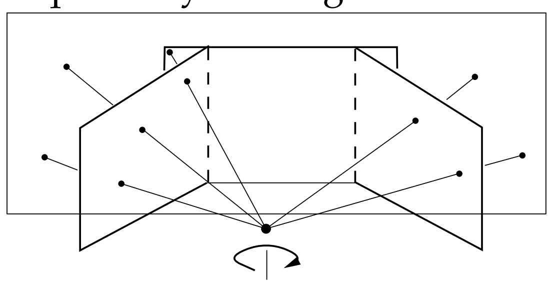
Example 2: Synthetic Rotations

Original image Warped: floor tile square Warped: door square

The synthetic images are produced by projectively warping the original image so that four corners of an imaged rectangle map to the corners of a rectangle. Both warpings correspond to a synthetic rotation of the camera about the (fixed) camera centre.

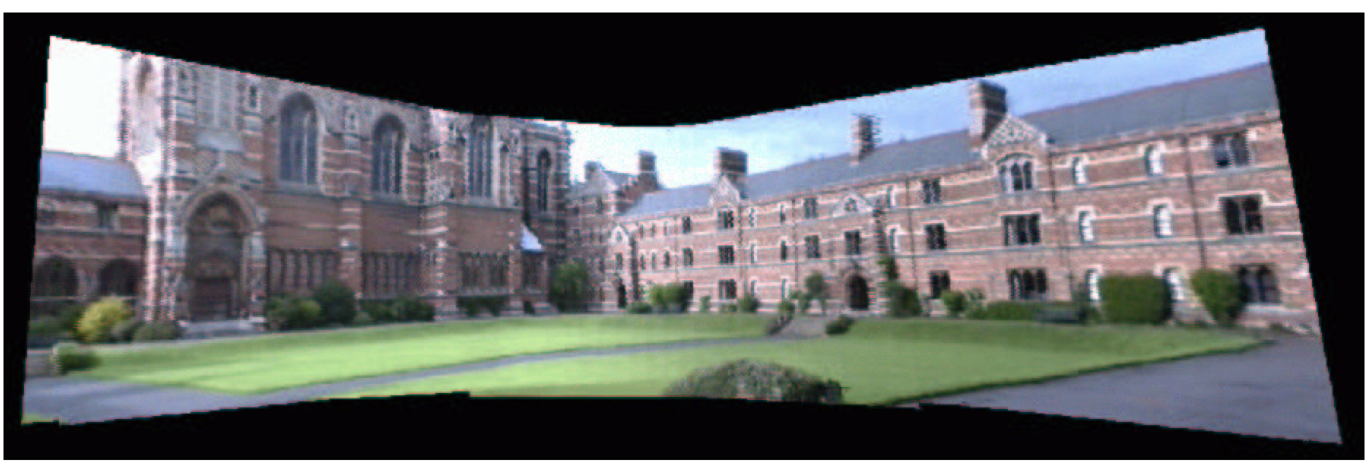
Example 3: Mosaics

Eight images (out of 30) acquired by rotating a camcorder about its optical centre.



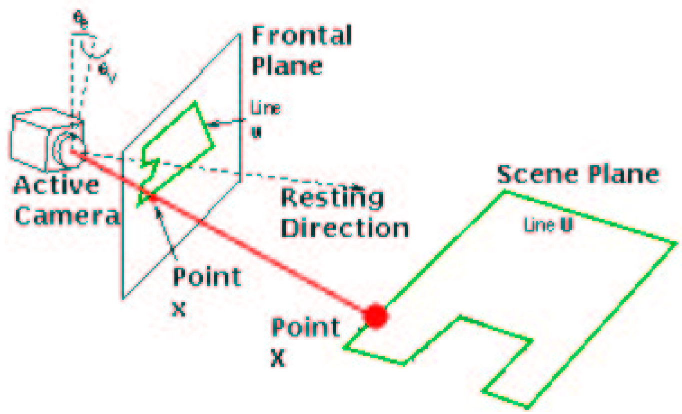
Register all the images to one reference image by projective transformations.

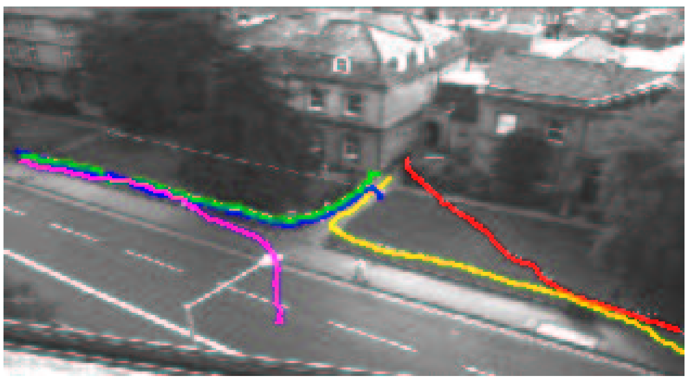
Keble Panoramic Mosaic

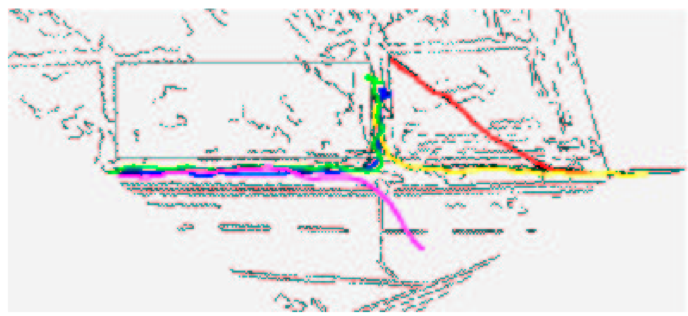


Example 4: Projective pointing devices: not just for artists! 2.22

Here the rotation joint of a pan-tilt camera become the projection centre, and tracking people in the ground plane produces a track on a notional frontal plane — like Durer's marks. These frontal plane tracks are then converted in Cartesian tracks viewed from above.







Summary 2.23

We have looked at four classes of transformation (in 2D):

Euclidean: 3 DOF	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$	Similarity: 4 DOF	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$
Affine: 6 DOF	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$	Projective: 8 DOF	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$

and their 3D counterparts.