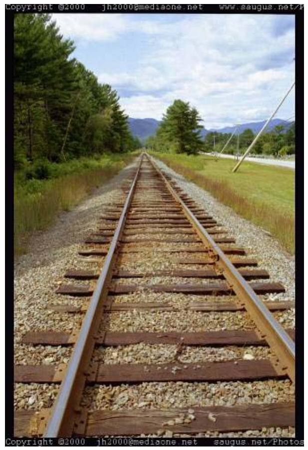
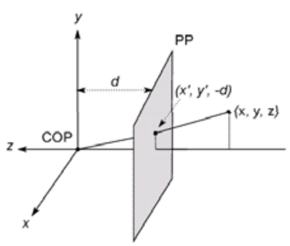
Image Formation



Digital Image Formation

- An image is a 2D array of numbers representing luminance (brightness), color, depth, or other physical quantity
- Luminance / brightness image: $f : \Re^2 \to \Re^+$
- Color image: $f : \Re^2 \to (\Re^+, \Re^+, \Re^+)$
- 2 key issues:
  - **Where** will be image of a scene point appear?
  - **How bright** will the image of a scene point be?

Modeling Perspective Projection



- Projection equations
  - Compute intersection with PP of ray from (x,y,z) to COP
  - Derived using similar triangles

$$(x, y, z) \to (-d\frac{x}{z},\ -d\frac{y}{z},\ -d)$$

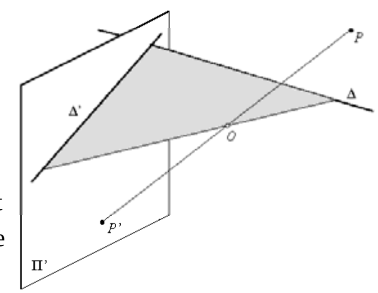
- We get the projection by throwing out the last coordinate:

$$(x, y, z) \to (-d\frac{x}{z},\ -d\frac{y}{z})$$

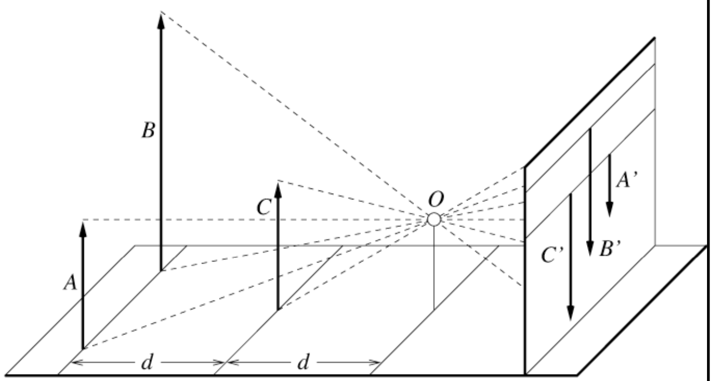
Slide by Steve Seitz

Geometric Properties of Projection

- Points go to points
- Lines go to lines
- Planes go to whole image
- Polygons go to polygons
- Degenerate cases
  - line through COP to point
  - plane through COP to line



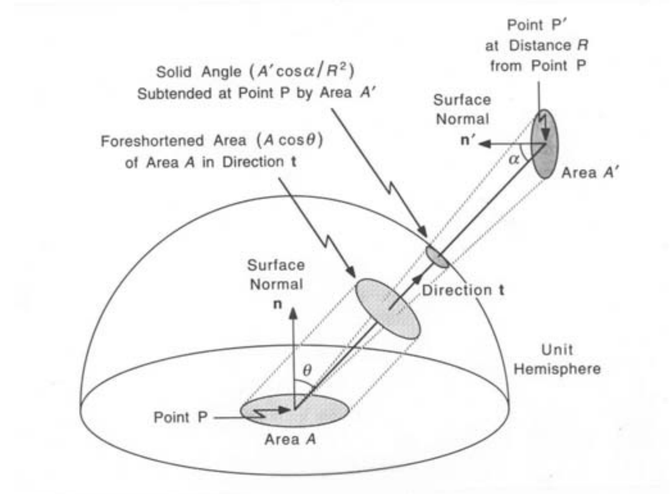
Distant Objects are Smaller



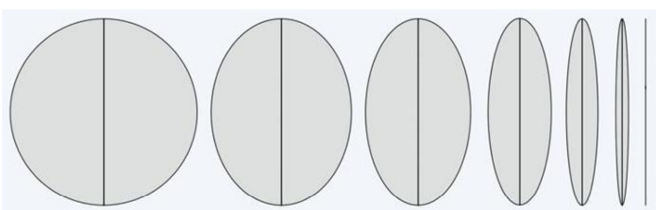
Magnification = f/z

22

Tilted Objects are Foreshortened



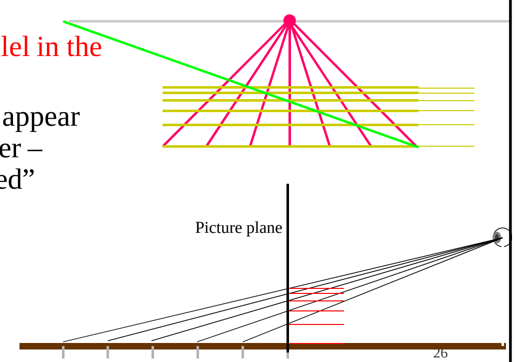
Tilted Objects are Foreshortened



25

Parallel Lines Perpendicular to the Optical Axis

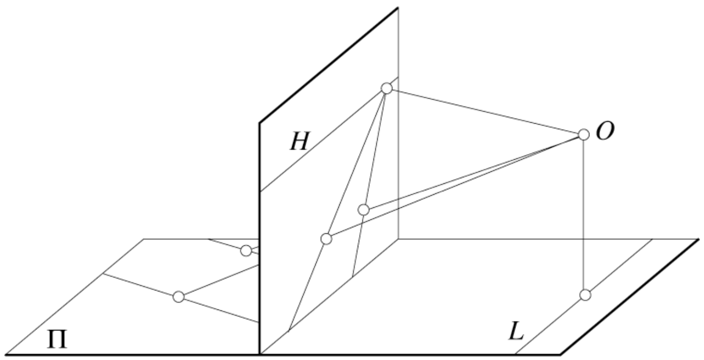
- Will be parallel in the image
- Distant lines appear closer together – "foreshortened"

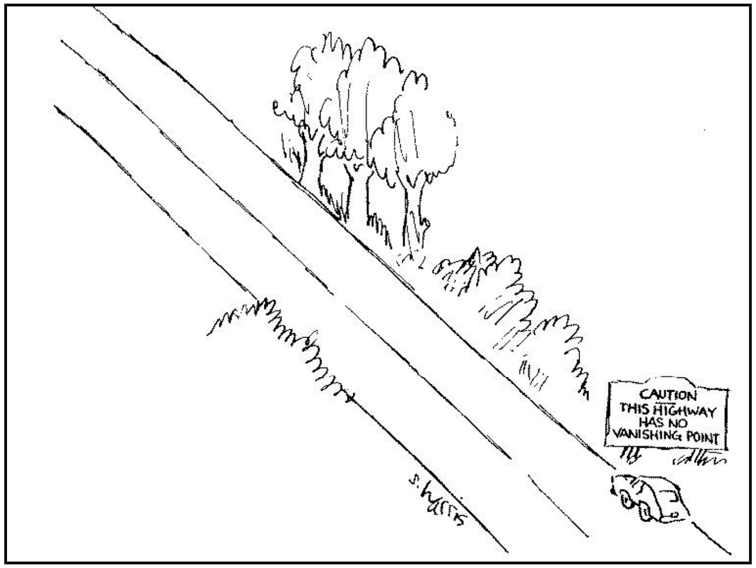


26

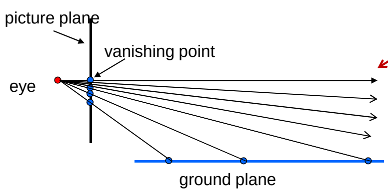
In General, Parallel Lines Meet

Moving the image plane merely scales the image





Vanishing Points

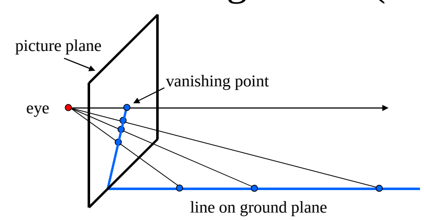


Line parallel to scene line and passing through optical center

- Vanishing point
  - projection of a point "at infinity"
  - Point in image beyond which projection of straight line cannot extend

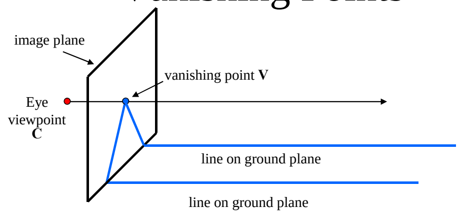
29

Vanishing Points (2D)



30

Vanishing Points



- Properties

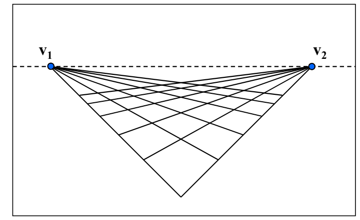
  Any two parallel lines have the same vanishing point **v**

  The ray from **C** through **v** is parallel to the lines

  An image may have more than one vanishing point

31

Vanishing lines



- Multiple Vanishing Points

  Any set of parallel lines on a plane define a vanishing point

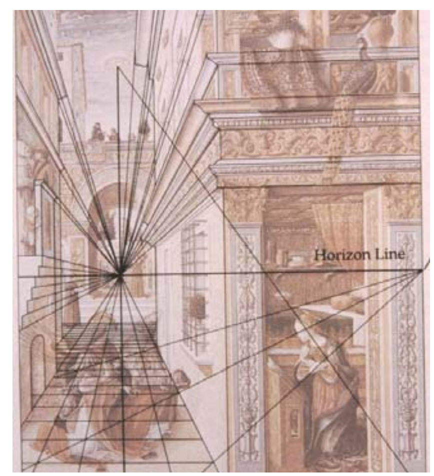
  The union of all of these vanishing points is the *horizon line*

  - also called *vanishing line*

  Note that different planes define different vanishing lines

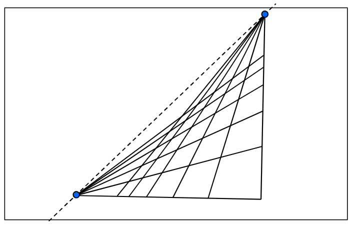
32

Carlo Crivelli (1486) *The Annunciation, with St. Emidius*



Perspective analysis of Crivelli's *Annunciation*

Vanishing Lines

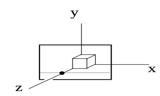
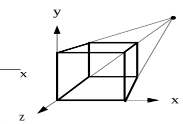


- Multiple Vanishing Points
  - Any set of parallel lines on a plane define a vanishing point
  - The union of all of these vanishing points is the *horizon line*
    - also called *vanishing line*
  - Note that different planes define different vanishing lines

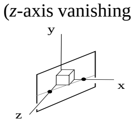
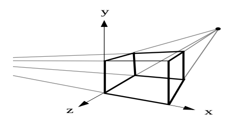
35

Vanishing Points

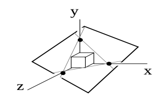
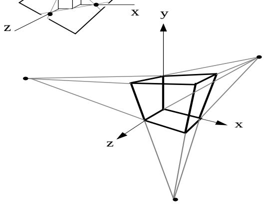
- For right-angled objects whose face normals are perpendicular to the *x, y, z* coordinate axes, number of vanishing points = number of principal coordinate axes intersected by projection plane

One Point Perspective
(*z*-axis vanishing point)

Two Point Perspective
(*z*- and *x*-axis vanishing points)

Three Point Perspective
(*z*-, *x*-, and *y*-axis vanishing points)

36

Vanishing Points

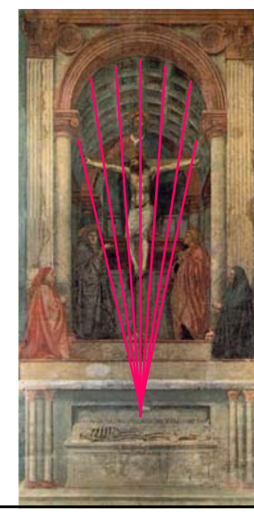
- each set of parallel lines (= direction) meets at a different point
  - The *vanishing point* for this direction
- Sets of parallel lines on the same plane lead to *collinear* vanishing points
  - The line is called the *horizon* for that plane

- Good ways to spot faked images
  - scale and perspective don't work
  - vanishing points behave badly
  - supermarket tabloids are a great source

37

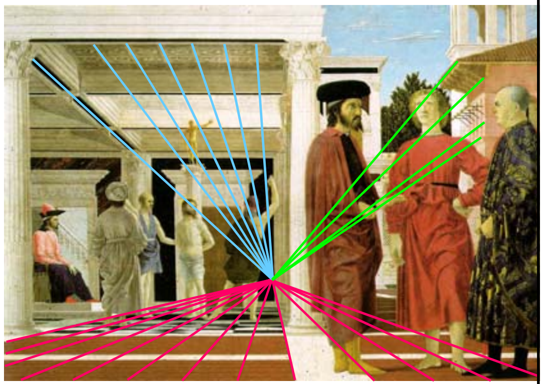
Masaccio's "Trinity" (c. 1425-8)

- The oldest existing example of linear perspective in Western art
- Use of "snapped" rope lines in plaster
- Vanishing point **below** orthogonals implies looking **up** at vaulted ceiling



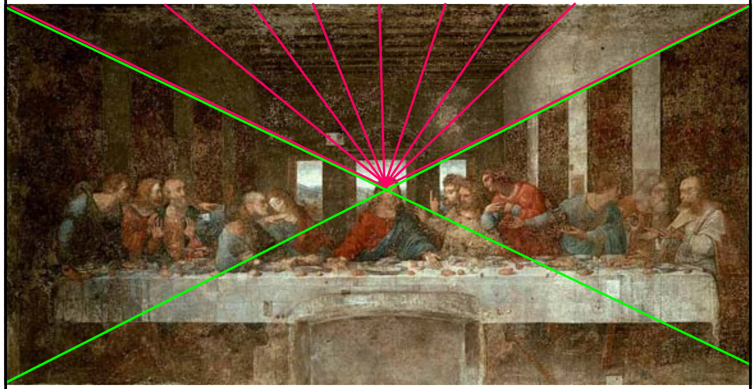
Piero della Francesca, “Flagellation of Christ” (c. 1455)

- Carefully planned
- Strong sense of space
- Low eye level



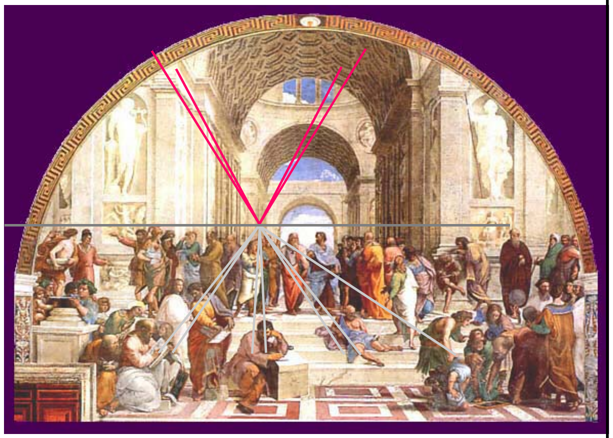
Leonardo da Vinci, “Last Supper” (c. 1497)

- Use of perspective to direct viewer’s eye
- Strong perspective lines to corners of image

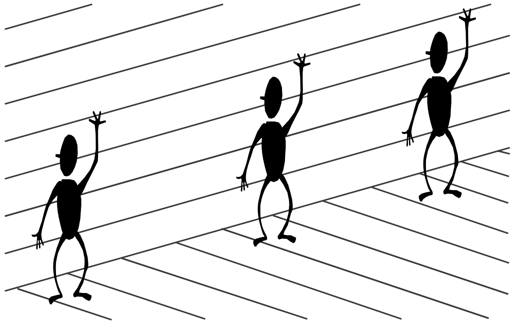


Raphael, “School of Athens” (1510-11)

- Single-point perspective
- Central
- Strong, coherent space

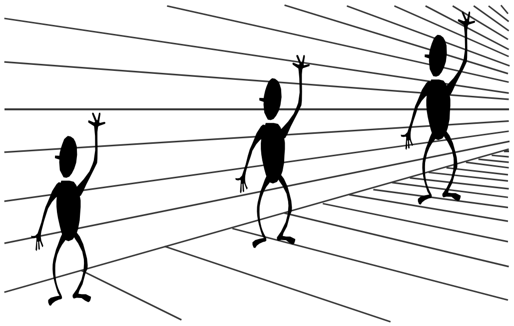


Perspective Cues from Parallel Lines in the Scene



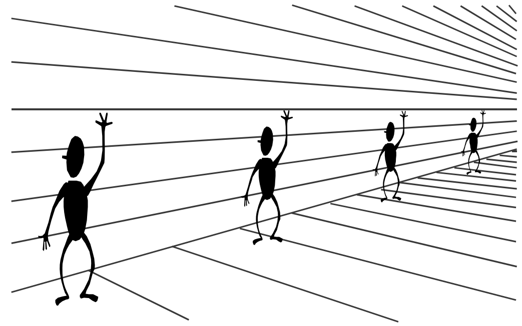
42

Perspective Cues



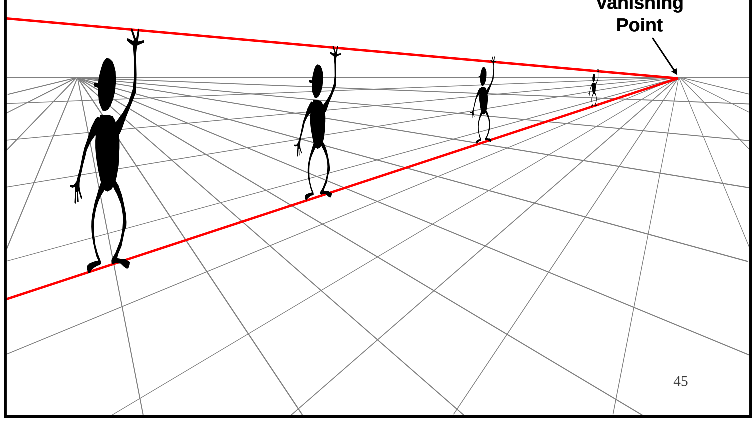
43

Perspective Cues



44

Comparing Heights



45

Painters have used Heuristics to aid in Robust Perception of Perspective

Example: Leonardo's Moderate Distance Rule

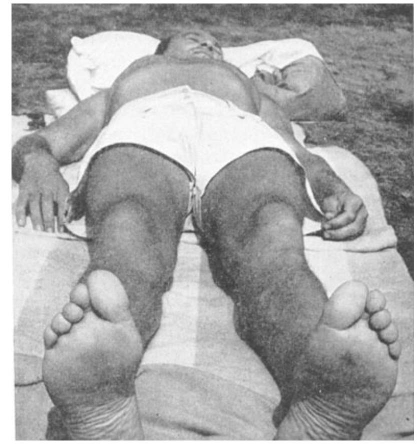
To minimize noticeable distortion, use shallow perspective:

> *"Make your view at least 20 times as far off as the greatest width or height of the objects represented, and this will satisfy any spectator placed anywhere opposite to the picture."*

-- Leonardo

Example: Extreme Viewpoints

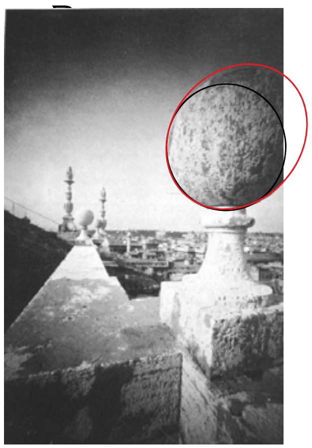
Mantegna, *Lamentation over the dead Christ*, 1480



Ogden's photo recreation of *The dead Christ*.

Example 2: Marginal View

Objects that are close to the viewer and at edge of field of view, are elongated by perspective projection



Pinhole camera photo of a marginal sphere

Raphael, *School of Athens, 1511*

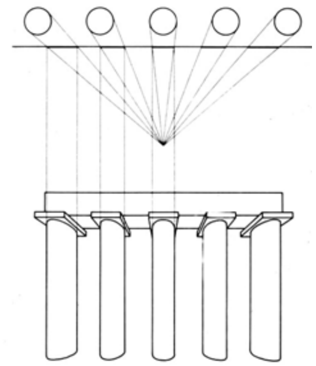
Zoroaster

Ptolomy

Spheres should be elongated to be perspectively correct, but they are not

Detail of Raphael's *School of Athens*

Leonardo's Solution to the Problem



"Make your view at least 20 times as far off as the greatest width or height of the objects represented, and this will satisfy any spectator placed

Pirenne's pinhole camera photo of marginal columns

Camera Transformations using Homogeneous Coordinates

- Computer vision and computer graphics usually represent points in Homogeneous coordinates instead of Cartesian coordinates
- Homogeneous coordinates are useful for representing perspective projection, camera projection, points at infinity, etc.
- Cartesian coordinates (x, y) represented as Homogeneous coordinates (wx, wy, w) for any scale factor $w \neq 0$
- Given 3D homogeneous coordinates (x, y, w), the 2D Cartesian coords are (x/w, y/w). I.e., a point projects to w=1 plane

Homogeneous Coordinates

Converting *to* homogeneous coordinates:

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image coordinates

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

homogeneous scene coordinates

Converting *from* homogeneous coordinates:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

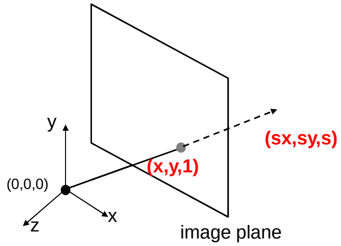
$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

Slide by Steve Seitz

The Projective Plane

- Geometric intuition

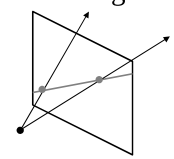
A ***point*** in the image is a ***ray*** in projective space from origin



- Each ***point*** **(x,y)** on the plane is represented by a ***ray*** **(sx,sy,s)**
  - all points on the ray are equivalent: **(x, y, 1) ≅ (sx, sy, s)**

Projective Lines

- What does a line in the image correspond to in projective space?



- A ***line*** in the image is a ***plane of rays*** through origin
  - all rays **p** = (*x*,*y*,*z*) satisfying: *ax* + *by* + *cz* = 0

$$\text{in vector notation:} \quad 0 = \begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

l **p**

- A line is also represented as a homogeneous 3-vector **l**

2D Mappings

- 2D **translation** - 2 DOFs

$$\begin{cases} x = u + a \\ y = v + b \end{cases} \qquad [x, y, 1]^T = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- 2D **rotation** (counterclockwise about the origin) - 1 DOF

$$\begin{cases} x = u\cos\theta + v\sin\theta \\ y = -u\sin\theta + v\cos\theta \end{cases} \qquad [x, y, 1]^T = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- 2D **rigid** (**Euclidean**) transformation: translation and rotation – 3 DOFs

2D Mappings (cont.)

- 2D **scale** - 2 DOFs

$$\begin{cases} x = \alpha u \\ y = \beta v \end{cases} \qquad [x, y, 1]^T = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- Composite translation, rotation, scale (**similarity** transformation) - 5 DOFs

$$[x, y, 1]^T = \begin{bmatrix} \alpha\cos\theta & \beta\cos\theta & \alpha(a\cos\theta - b\sin\theta) \\ -\alpha\sin\theta & \beta\cos\theta & \beta(a\sin\theta + b\cos\theta) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

2D Mappings (cont.)

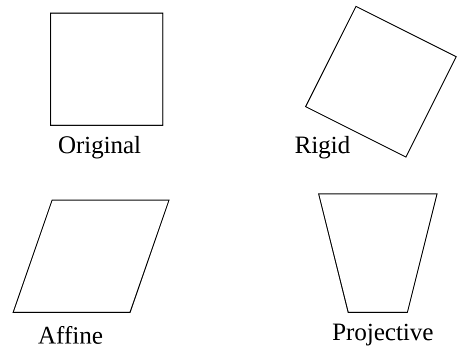
- **Affine** (linear) - 6 DOFs

$$\begin{cases} x = a_{11}u + a_{12}v + a_{13} \\ y = a_{21}u + a_{22}v + a_{23} \end{cases} \qquad [x, y, 1]^T = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- **Projective** (allows skewing) - 8 DOFs (a_{33} is a scale factor)

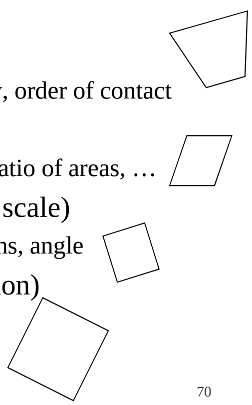
$$\begin{cases} x = \dfrac{a_1u + a_2v + a_3}{a_4u + a_5v + 1} \\ y = \dfrac{b_1u + b_2v + b_3}{a_4u + a_5v + 1} \end{cases} \qquad [x, y, 1]^T = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

Examples of 2D Transformations



Properties of Transformations

- **Projective**
  - Preserves collinearity, concurrency, order of contact
- **Affine** (linear transformations)
  - Preserves above plus parallelism, ratio of areas, …
- **Similarity** (rotation, translation, scale)
  - Preserves above plus ratio of lengths, angle
- **Euclidean** (rotation and translation)
  - Preserves above plus length, area



Using Homogeneous Coordinates

- Translation by (a, b, c)

$$\begin{cases} x' = x - a \\ y' = y - b \\ z' = z - c \end{cases}$$

or

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$P' = T P$$

- Scale Change by (S_x, S_y, S_z)

$$\begin{cases} x' = x S_x \\ y' = y S_y \\ z' = z S_z \end{cases}$$

or

$P' = S P$ where

$$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Rotation about coordinate axes (counterclockwise looking towards origin) by [redacted] θ

Ex. About z-axis:

$$\begin{cases} x' = x\cos\theta - y\sin\theta \\ y' = x\sin\theta + y\cos\theta \\ z' = z \end{cases}$$

$$\Rightarrow R_z = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Also,

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Any transformation involving translation, scale or rotation can be written as

$$P' = MP$$

where M constructed by composing transformation matrices

Ex.

$$\begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \alpha\cos\theta & \beta\sin\theta & \alpha(a\cos\theta - b\sin\theta) \\ -\alpha\sin\theta & \beta\cos\theta & \beta(a\sin\theta + b\cos\theta) \\ 0 & 0 & 1 \end{bmatrix}$$

- Translations are commutative, rotations are not
- General transformation matrix of form:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & T_x \\ a_{21} & a_{22} & a_{23} & T_y \\ a_{31} & a_{32} & a_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Affine Transformation

3D Mappings

- Cartesian coordinates $(x, y, z) \rightarrow (x, y, z, w)$ in homogeneous coordinates
- 4 x 4 matrix for **affine transformations**:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where r_{ij} specify aggregate rotation and scale change, and t_i specify translation

Perspective Projection

- Projection is a matrix multiply using homogeneous coordinates:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ -z/d \end{bmatrix} \Rightarrow (-d\frac{x}{z},\ -d\frac{y}{z})$$

divide by third coordinate

This is known as **perspective projection**

- The matrix is the **camera perspective projection matrix**
- Can also formulate as a 4x4

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ -z/d \end{bmatrix} \Rightarrow (-d\frac{x}{z},\ -d\frac{y}{z})$$

divide by fourth coordinate

Slide by Steve Seitz

Perspective Projection

- How does multiplying the projection matrix by a constant change the transformation?

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ -z/d \end{bmatrix} \Rightarrow (-d\frac{x}{z}, \ -d\frac{y}{z})$$

$$\begin{bmatrix} -d & 0 & 0 & 0 \\ 0 & -d & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} -dx \\ -dy \\ z \end{bmatrix} \Rightarrow (-d\frac{x}{z}, \ -d\frac{y}{z})$$

Homographies

- Perspective projection of a plane

Lots of names for general plane-to-plane transformations:

- **homography**, texture-map, colineation, planar projective map

Modeled as a 2D warp using homogeneous coordinates

$$\underset{\mathbf{p'}}{\begin{bmatrix} sx' \\ sy' \\ s \end{bmatrix}} = \underset{\mathbf{H}}{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}} \underset{\mathbf{p}}{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}$$

To apply a homography **H**

- Compute $\mathbf{p}' = \mathbf{H}\mathbf{p}$ (regular matrix multiply)
- Convert $\mathbf{p}'$ from homogeneous to image coordinates
  - divide by *s* (third) coordinate

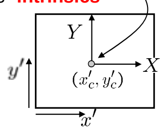
Camera Parameters

A camera is described by several parameters

- Translation **T** of the optical center from the origin of world coords
- Rotation **R** of the image plane
- focal length **f**, principle point **(x'_c, y'_c)** , pixel size **(s_x, s_y)**
- blue parameters are called "**extrinsics**," red are "**intrinsics**"

Projection equation

$$\mathbf{x} = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi X}$$



- The projection matrix models the cumulative effect of ***all*** parameters
- Useful to decompose into a series of operations

identity matrix

$$\mathbf{\Pi} = \underset{\substack{\text{intrinsics} \\ \mathbf{K}}}{\begin{bmatrix} -fs_x & 0 & x'_c \\ 0 & -fs_y & y'_c \\ 0 & 0 & 1 \end{bmatrix}} \underset{\text{projection}}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}} \underset{\text{rotation}}{\begin{bmatrix} \mathbf{R}_{3x3} & \mathbf{0}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}} \underset{\text{translation}}{\begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{T}_{3x1} \\ \mathbf{0}_{1x3} & 1 \end{bmatrix}}$$

Note: Can also add other parameters to model lens distortion