## **CS4620/5620:** Transformations

Professor: Kavita Bala




## Announcements

- Check course web page
  - Piazza information posted for 4620/5620, 4621/5621
- 24x7 online book for 2nd version of Shirley posted
- 4621/5621 starting from next week (we have still not changed registration and are waiting to hear)
- First homework will be out next week




## Homogeneous coordinates

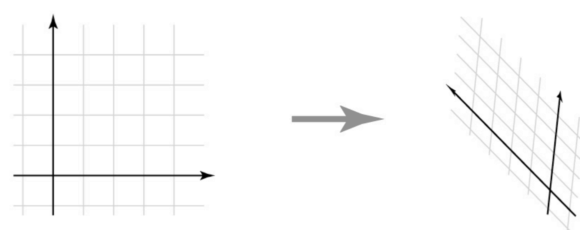
- A trick for representing the foregoing more elegantly
- Extra component *w* for vectors, extra row/column for matrices
  - for affine, can always keep $w = 1$
- Represent linear transformations with dummy extra row and column

$$\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \\ 1 \end{bmatrix}$$




## Affine transformations

- The set of transformations we are interested in is known as the "affine" transformations
  - straight lines preserved; parallel lines preserved
  - ratios of lengths along lines preserved (midpoints preserved)






## Transforming points and vectors

- Homogeneous coords. let us exclude translation
  - just put 0 rather than 1 in the last place

$$\begin{bmatrix} M & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} = \begin{bmatrix} M\mathbf{p} + \mathbf{t} \\ 1 \end{bmatrix} \quad \begin{bmatrix} M & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix} = \begin{bmatrix} M\mathbf{v} \\ 0 \end{bmatrix}$$

  - and note that subtracting two points cancels the extra coordinate, resulting in a vector!

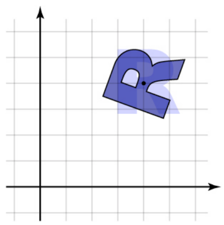



## More math background

- Coordinate systems
  - Expressing vectors with respect to bases
  - Linear transformations as changes of basis

## Composing to change axes

- Want to rotate about a particular point
  - could work out formulas directly…
- Know how to rotate about the origin
  - so translate that point to the origin

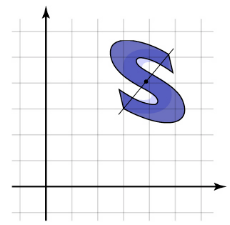


$$M = T^{-1}RT$$



## Composing to change axes

- Want to scale along a particular axis and point
- Know how to scale along the y axis at the origin
  - so translate to the origin and rotate to align axes



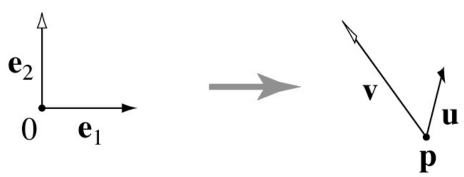
$$M = T^{-1}R^{-1}SRT$$



## Affine change of coordinates

- Six degrees of freedom

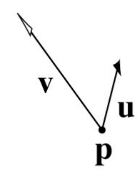
$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{p} \\ 0 & 0 & 1 \end{bmatrix}$$






## Affine change of coordinates

- Coordinate frame: point plus basis
- Interpretation: transformation changes representation of point from one basis to another
- "Frame to canonical" matrix has frame in columns
  - takes points represented in frame
  - represents them in canonical basis
  - e.g. [0 0], [1 0], [0 1]
- Seems backward but bears thinking about




$$\begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{p} \\ 0 & 0 & 1 \end{bmatrix}$$



## Affine change of coordinates

- When we move an object to the origin to apply a transformation, we are really changing coordinates
  - the transformation is easy to express in object's frame
  - so define it there and transform it

$$T_e = FT_FF^{-1}$$

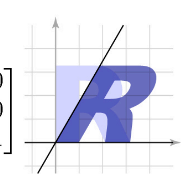
  - $T_e$ is the transformation expressed wrt. $\{e_1, e_2\}$
  - $T_F$ is the transformation expressed in natural frame
  - $F$ is the frame-to-canonical matrix [$u$ $v$ $p$]
- This is a *similarity transformation*



## Affine change of coordinates

- A new way to "read off" the matrix
  - e.g. shear from earlier
  - can look at picture, see effect on basis vectors, write down matrix

$$\begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



- Also an easy way to construct transforms
  - e. g. scale by 2 across direction (1,2)
    - (homework)

## Coordinate frame summary

- Frame = point plus basis
- Frame matrix (frame-to-canonical) is

$$F = \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{p} \\ 0 & 0 & 1 \end{bmatrix}$$

- Move points to and from frame by multiplying with *F*

$$p_e = F p_F \quad p_F = F^{-1} p_e$$

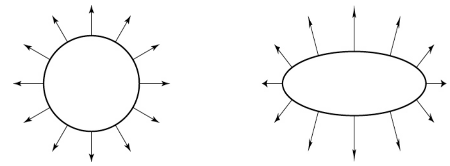
- Move transformations using similarity transforms

$$T_e = F T_F F^{-1} \quad T_F = F^{-1} T_e F$$




## Transforming normal vectors

- Transforming surface normals
  - differences of points (and therefore tangents) transform OK
  - normals do not --> use inverse transpose matrix



have: $\mathbf{t} \cdot \mathbf{n} = \mathbf{t}^T \mathbf{n} = 0$
want: $M\mathbf{t} \cdot X\mathbf{n} = \mathbf{t}^T M^T X \mathbf{n} = 0$
so set $X = (M^T)^{-1}$
then: $M\mathbf{t} \cdot X\mathbf{n} = \mathbf{t}^T M^T (M^T)^{-1} \mathbf{n} = \mathbf{t}^T \mathbf{n} = 0$




## Perspective




## History of projection

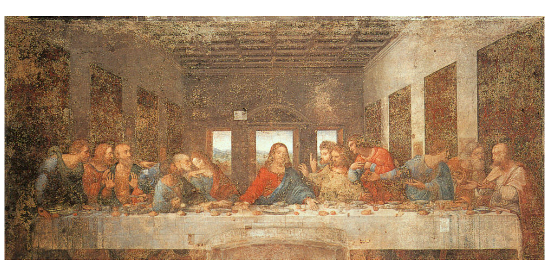
- Ancient times: Greeks wrote about laws of perspective
- Renaissance: perspective is adopted by artists

Duccio c. 1308




## History of projection

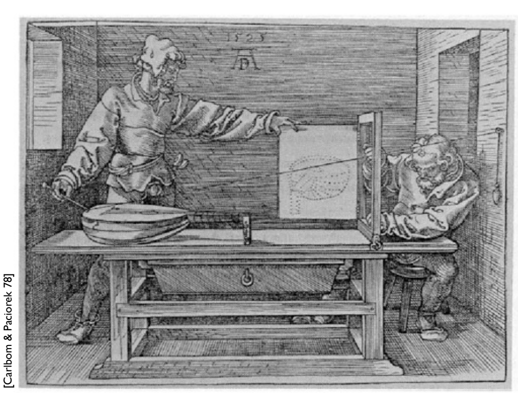
- Later Renaissance: perspective formalized precisely



da Vinci c. 1498

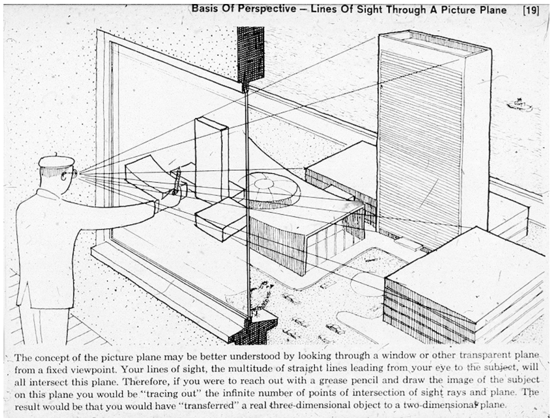



## Plane projection in drawing



[Carlbom & Paciorek 78]

# Plane projection in drawing



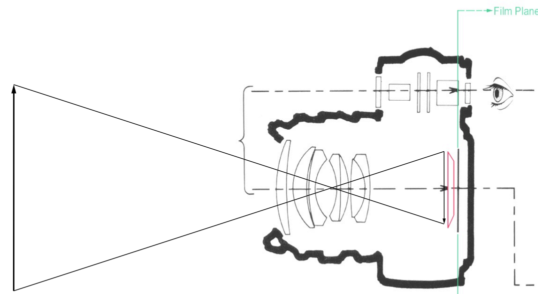

[CS 417 Spring 2002]





# Plane projection in photography

- This is another model for what we are doing
  - applies more directly in realistic rendering




[Source unknown]





# Plane projection in photography

[Richard Zakia]





# Ray generation vs. projection

- Viewing in ray tracing
  - start with image point
  - compute 3D point that projects to that point using ray
  - do this using geometry
- Viewing by projection
  - start with 3D point
  - compute image point that it projects to
  - do this using transforms
- Inverse processes