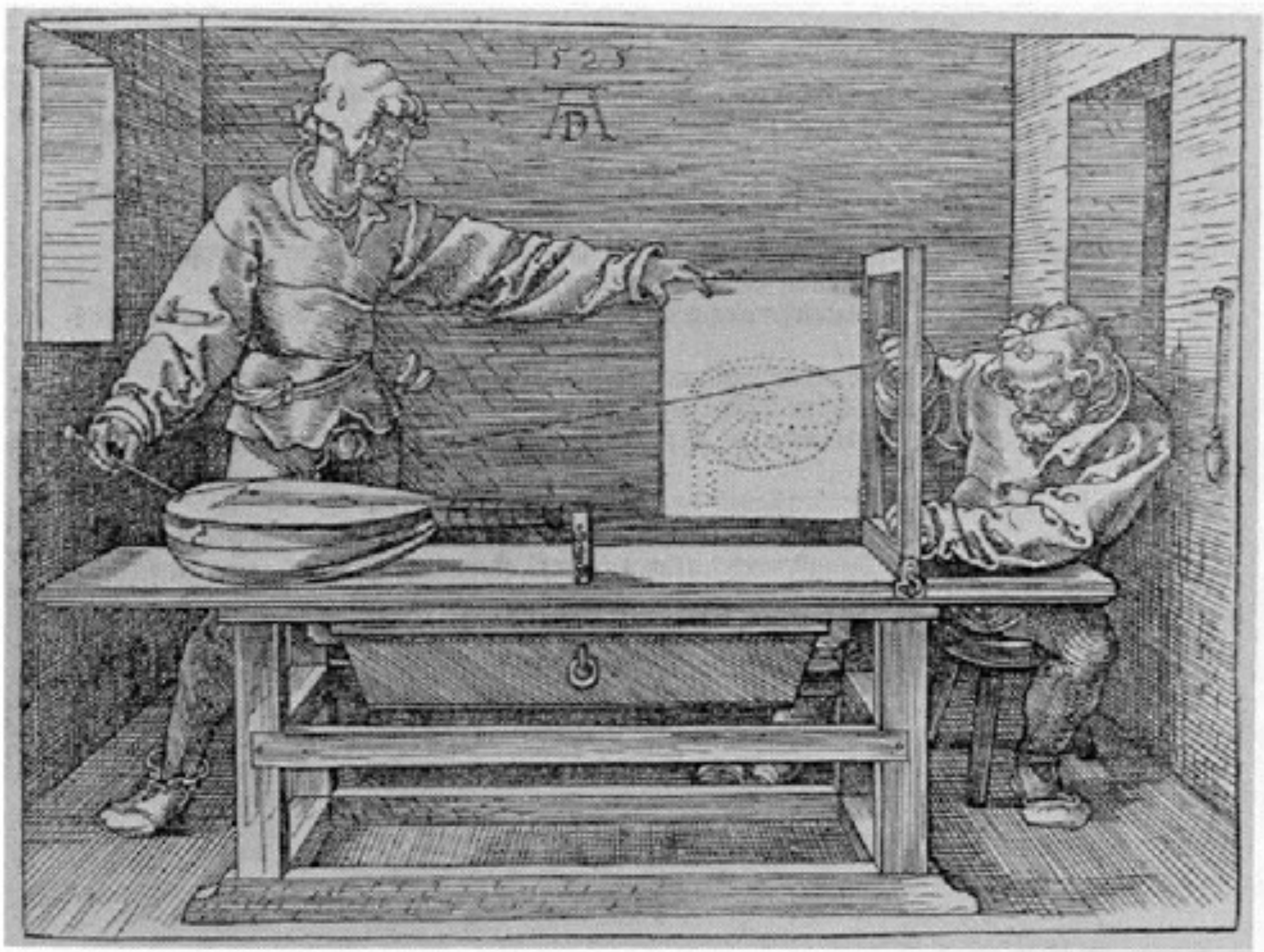
Graphics Pipeline
2D Geometric Transformations

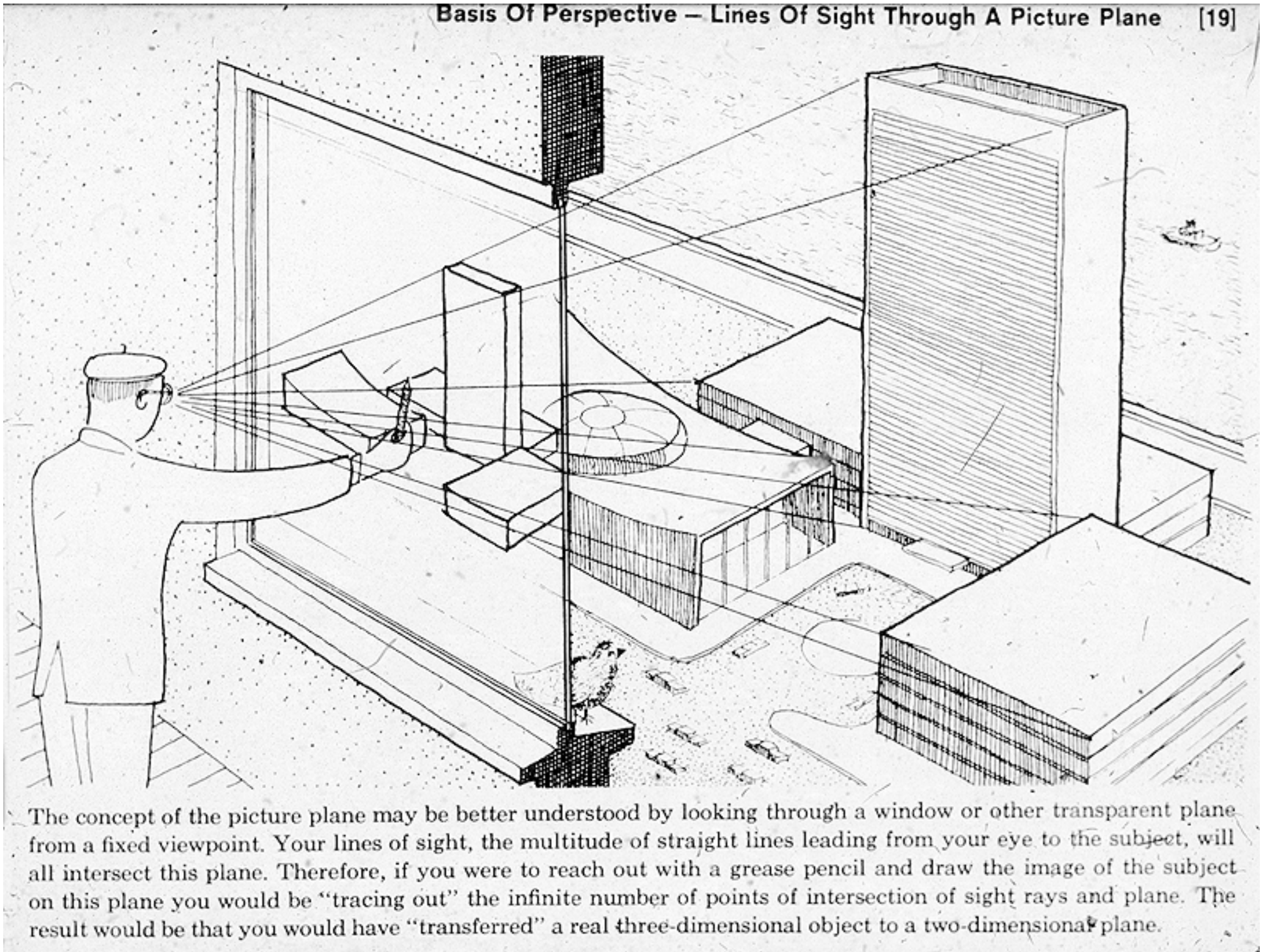
CS 4620 Lecture 8

Plane projection in drawing



Albrecht Dürer

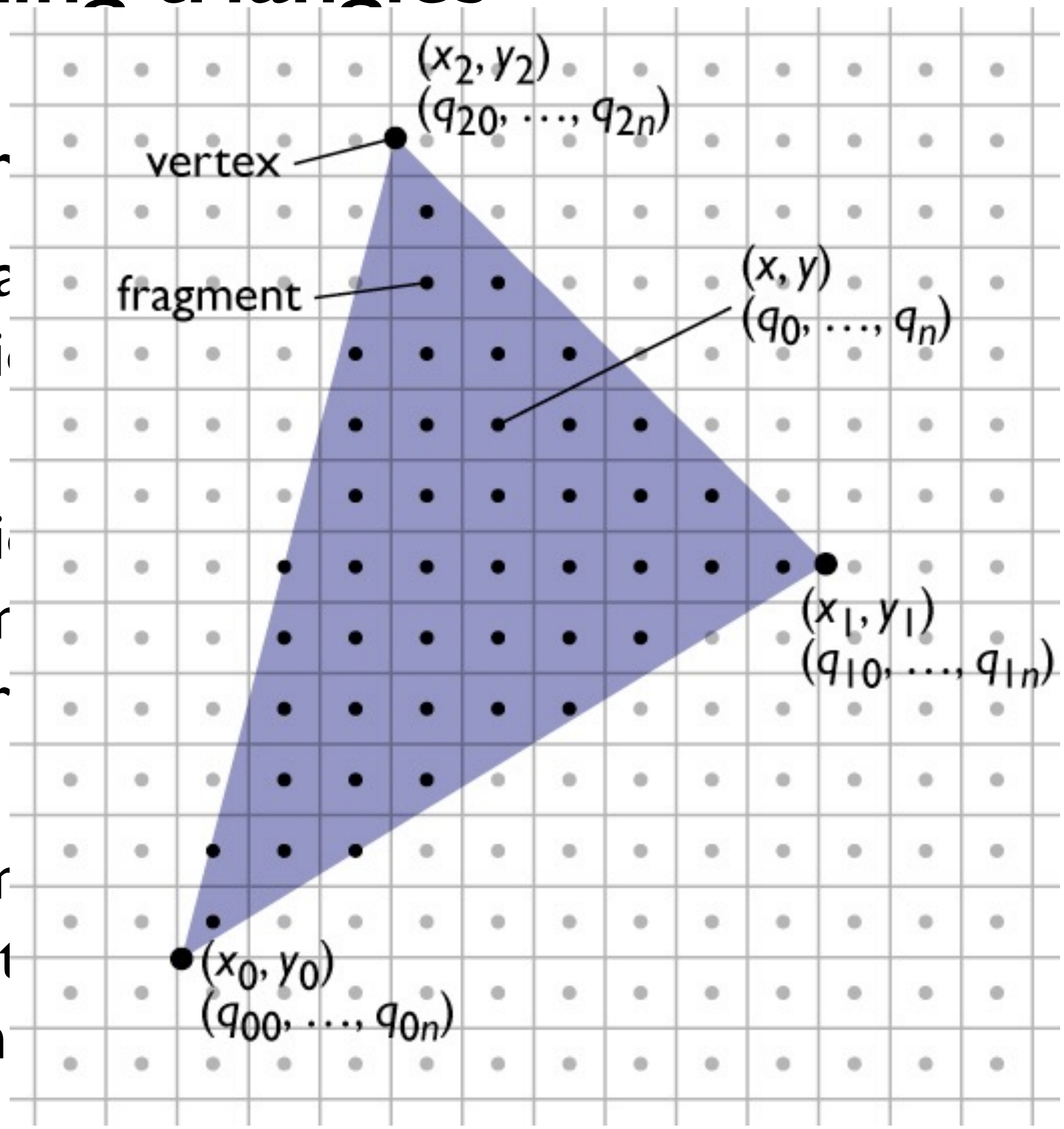
Plane projection in drawing



source unknown

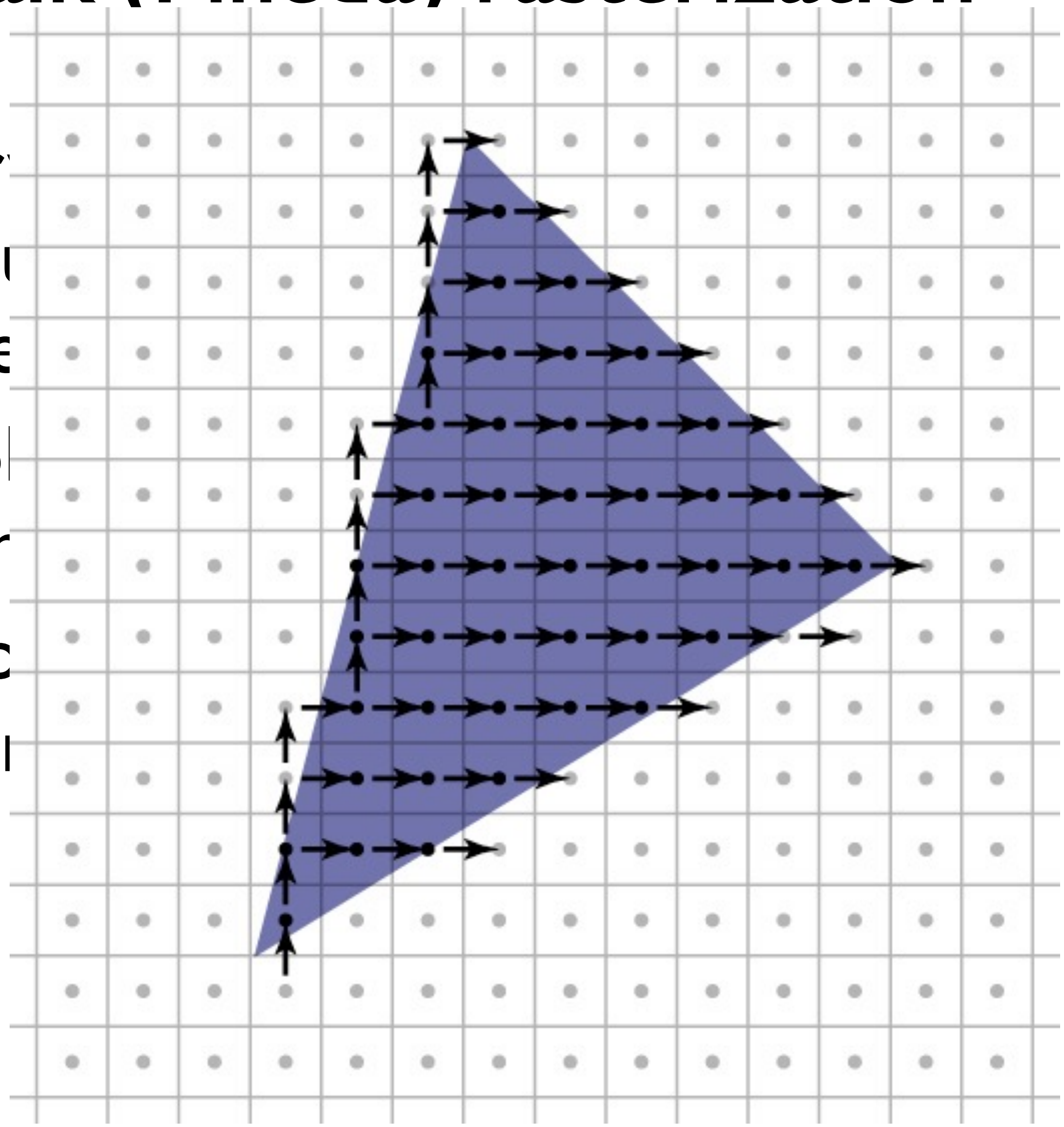
Rasterizing triangles

- Summar
  1 evalua
  functi
  grid
  2 functi
  param
  at ver
  3 using
  param
  to det
  fragm



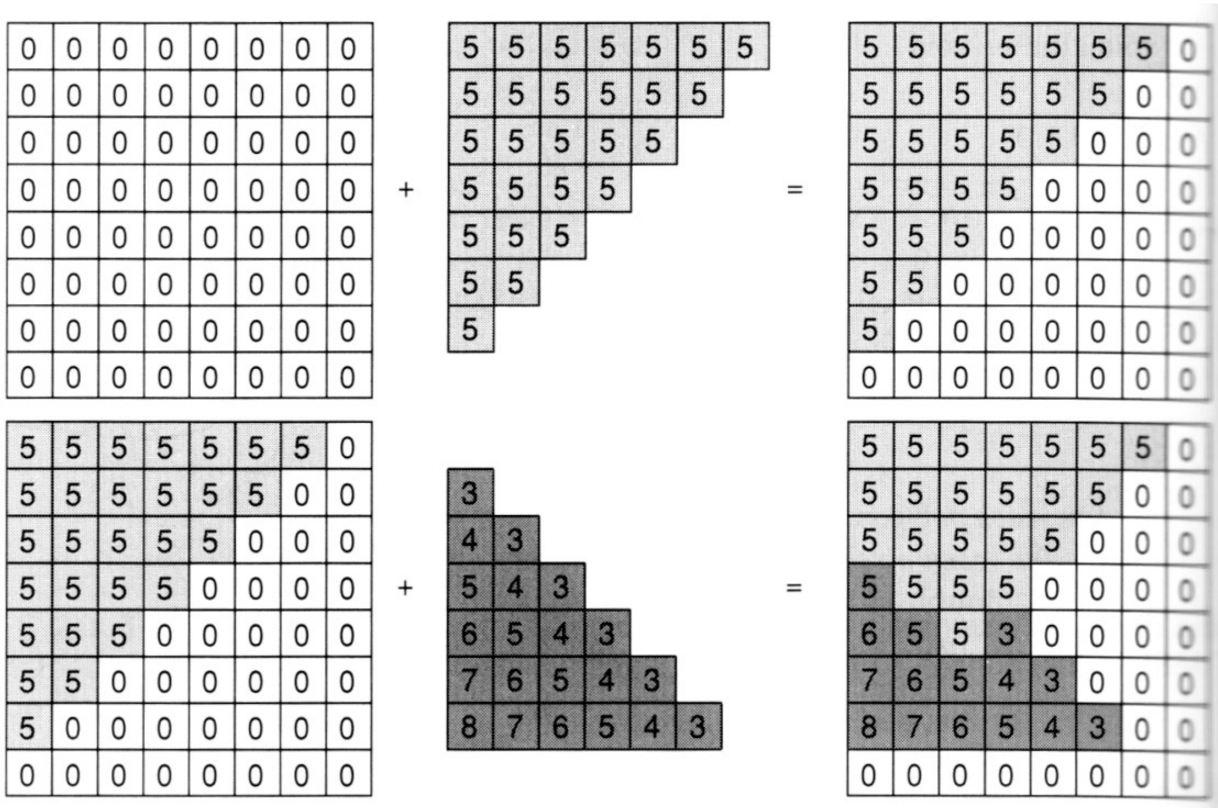
Pixel-walk (Pineda) rasterization

- Conser
  visit a s
  the pixe
- Interpo
  functior
- Use tho
  to deter
  to emit



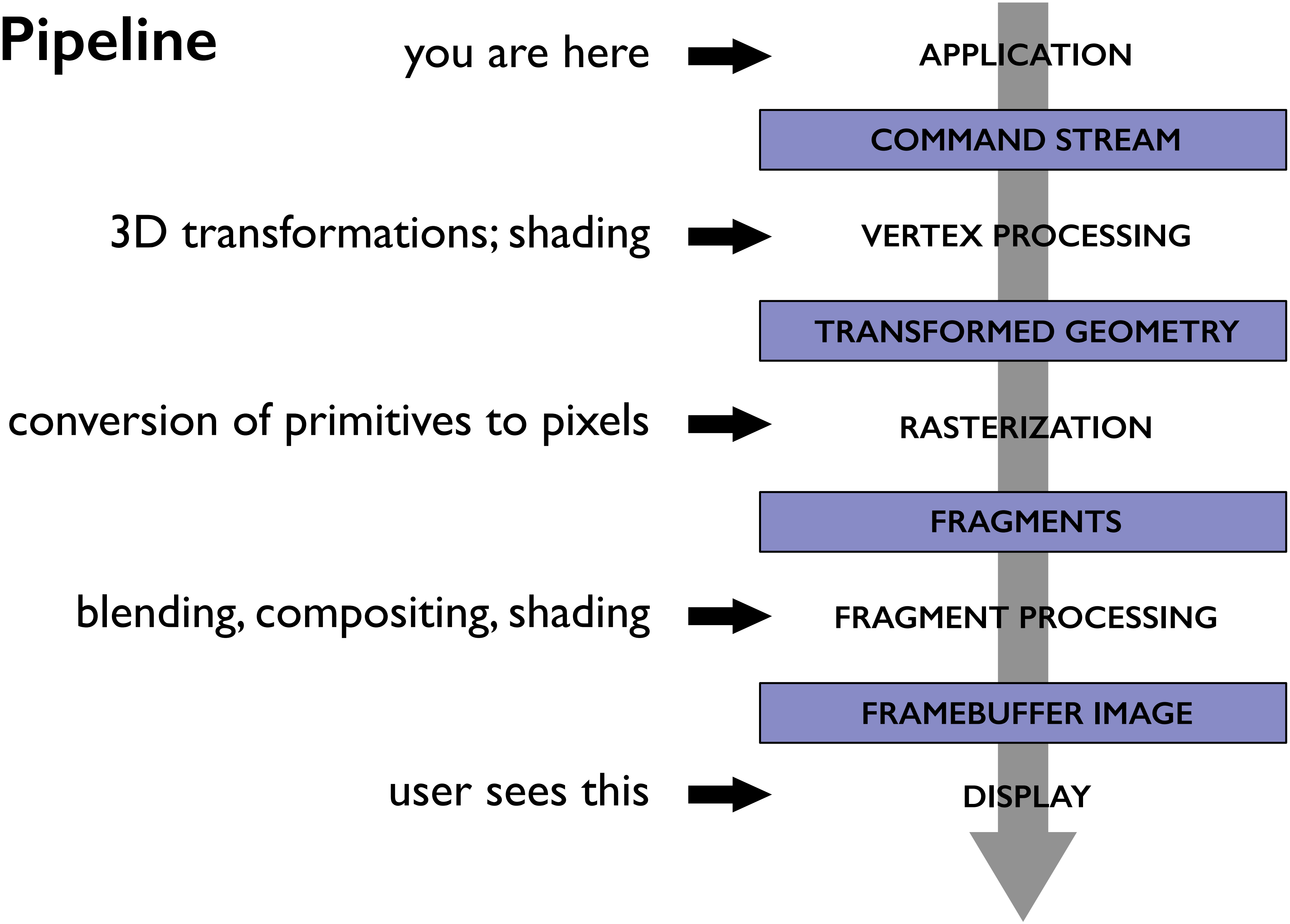
The z buffer

– anothe rce
approa d



[Foley et al.]

Pipeline


you are here
APPLICATION
COMMAND STREAM
3D transformations; shading
VERTEX PROCESSING
TRANSFORMED GEOMETRY
conversion of primitives to pixels
RASTERIZATION
FRAGMENTS
blending, compositing, shading
FRAGMENT PROCESSING
FRAMEBUFFER IMAGE
user sees this
DISPLAY

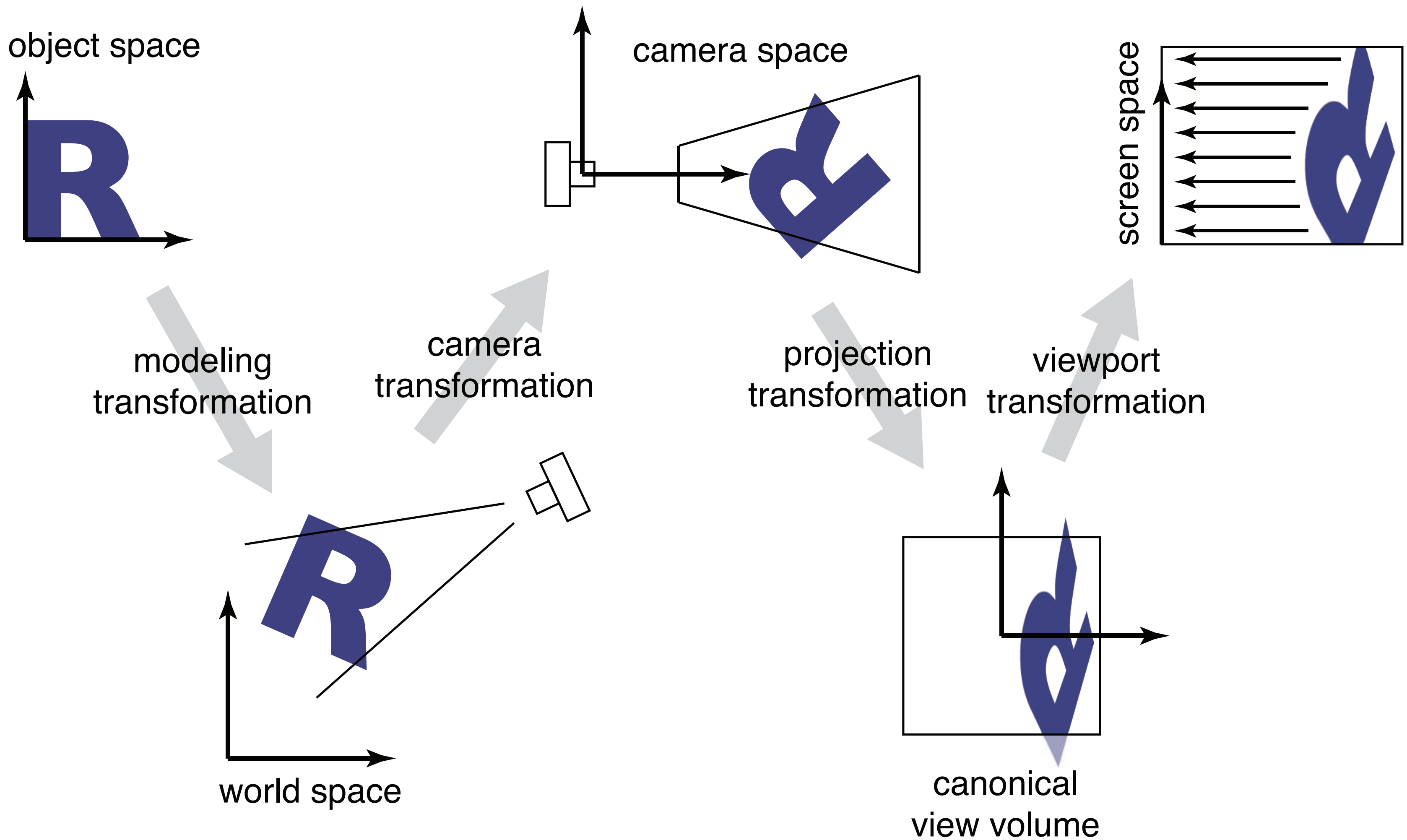

three.js - dynamic procedural terrain using 3d simplex noise
birds by mirada from ro.me - textures by qubodup and davis123 - music by Kevin MacLeod

Some demos

- http://mrdoob.github.io/three.js/examples/
- http://carvisualizer.plus360degrees.com/threejs/
- http://madebyevan.com/webgl-water/
- http://akirodic.com/p/jellyfish/

Pipeline of transformations

- Standard sequence of transforms



A little quick math background

- Notation for sets, functions, mappings
- Linear transformations
- Matrices
  - Matrix-vector multiplication
  - Matrix-matrix multiplication
- Geometry of curves in 2D
  - Implicit representation
  - Explicit representation

Implicit representations

- Equation to tell whether we are on the curve
  $$\{\mathbf{v} \mid f(\mathbf{v}) = 0\}$$
- Example: line (orthogonal to **u**, distance *k* from **0**)
  $$\{\mathbf{v} \mid \mathbf{v} \cdot \mathbf{u} + k = 0\}$$ (**u** is a unit vector)
- Example: circle (center **p**, radius *r*)
  $$\{\mathbf{v} \mid (\mathbf{v} - \mathbf{p}) \cdot (\mathbf{v} - \mathbf{p}) - r^2 = 0\}$$
- Always define boundary of region
  - (if *f* is continuous)

Explicit representations

- Also called parametric
- Equation to map domain into plane

$$\{f(t)\,|\,t \in D\}$$

- Example: line (containing **p**, parallel to **u**)

$$\{\mathbf{p} + t\mathbf{u}\,|\,t \in \mathbb{R}\}$$

- Example: circle (center **b**, radius *r*)

$$\{\mathbf{p} + r[\cos t\ \ \sin t]^T\,|\,t \in [0, 2\pi)\}$$

- Like tracing out the path of a particle over time
- Variable *t* is the "parameter"

Transforming geometry

$$S \to \{T(\mathbf{v}) \,|\, \mathbf{v} \in S\}$$

- Parametric representation:

$$\{f(t) \,|\, t \in D\} \to \{T(f(t)) \,|\, t \in D\}$$

- Implicit representation:

$$\{\mathbf{v} \,|\, f(\mathbf{v}) = 0\} \to \{T(\mathbf{v}) \,|\, f(\mathbf{v}) = 0\}$$
$$= \{\mathbf{v} \,|\, f(T^{-1}(\mathbf{v})) = 0\}$$

Translation

- Simplest transformation: $T(\mathbf{v}) = \mathbf{v} + \mathbf{u}$
- Inverse: $T^{-1}(\mathbf{v}) = \mathbf{v} - \mathbf{u}$

Linear transformations

- One way to define a transformation is by matrix multiplication:

$$T(\mathbf{v}) = M\mathbf{v}$$

- Such transformations are *linear*, which is to say:

$$T(a\mathbf{u} + \mathbf{v}) = aT(\mathbf{u}) + T(\mathbf{v})$$

(and in fact all linear transformations can be written this way)

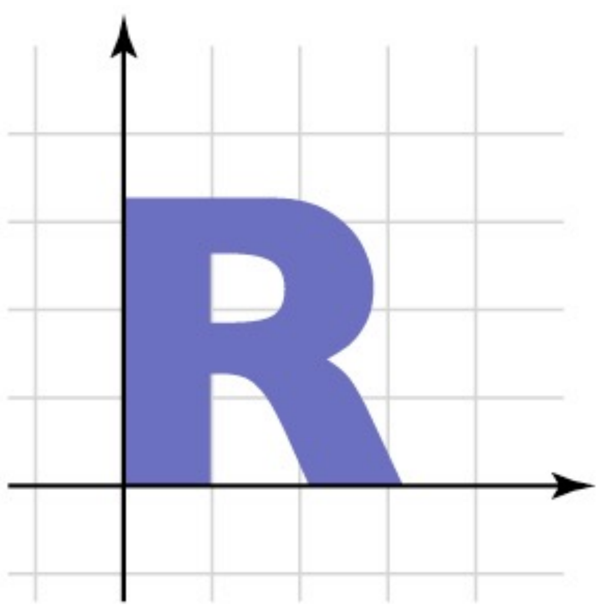
Geometry of 2D linear trans.

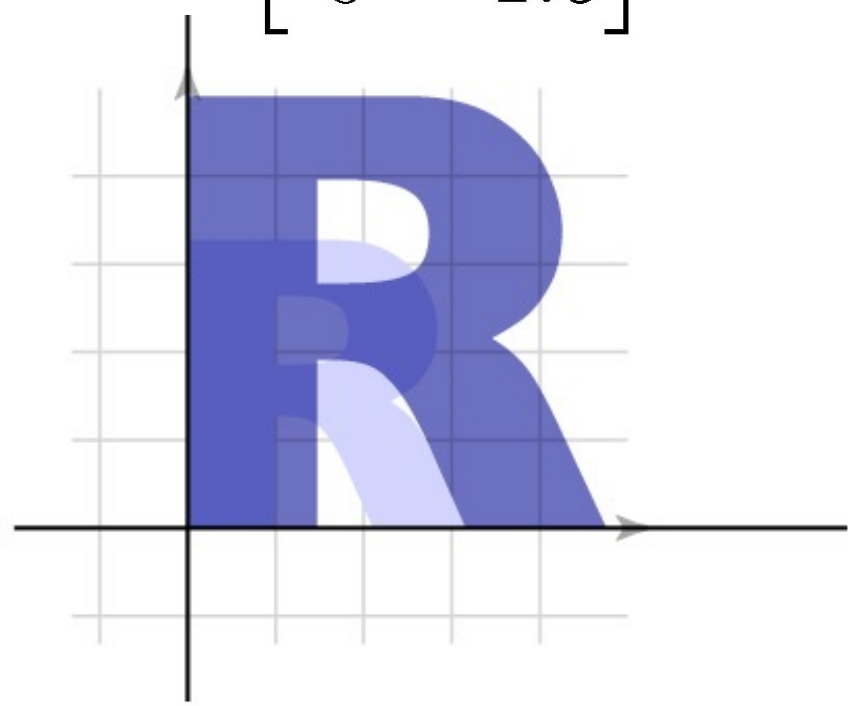
- 2x2 matrices have simple geometric interpretations
  - uniform scale
  - non-uniform scale
  - rotation
  - shear
  - reflection
- Reading off the matrix

Linear transformation gallery

- Uniform scale $$\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} sx \\ sy \end{bmatrix}$$

$$\begin{bmatrix} 1.5 & 0 \\ 0 & 1.5 \end{bmatrix}$$

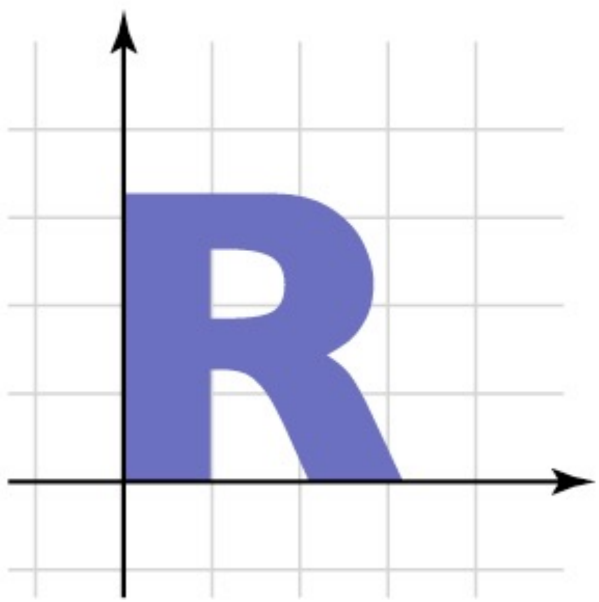


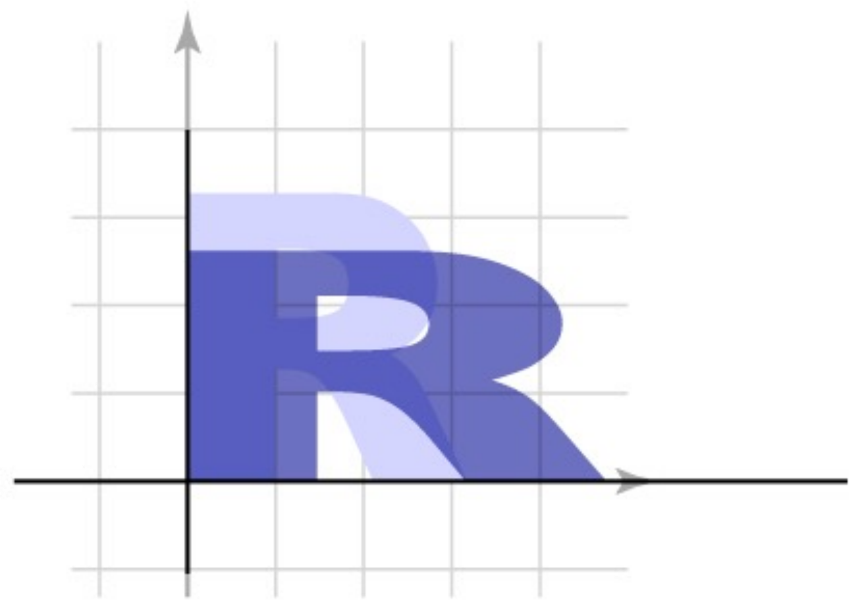


Linear transformation gallery

- Nonuniform scale $\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x x \\ s_y y \end{bmatrix}$

$$\begin{bmatrix} 1.5 & 0 \\ 0 & 0.8 \end{bmatrix}$$

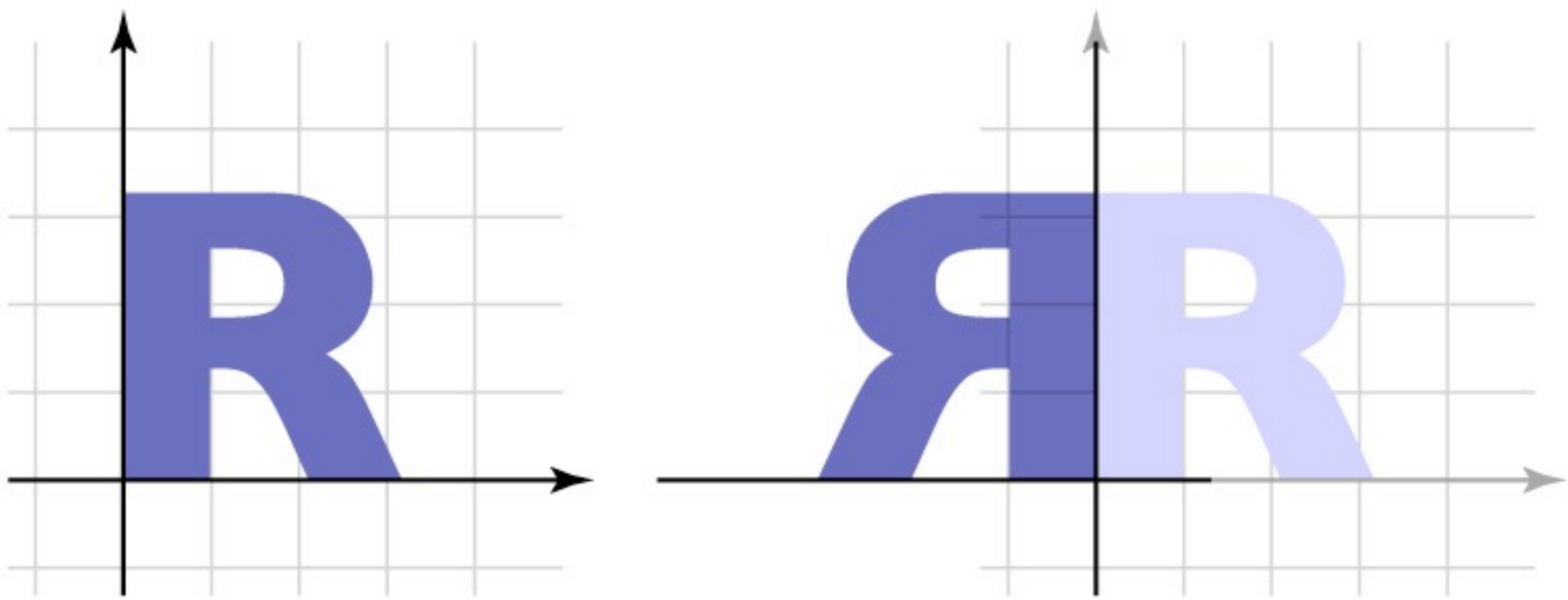




Linear transformation gallery

- Reflection
  - can consider it a special case of nonuniform scale

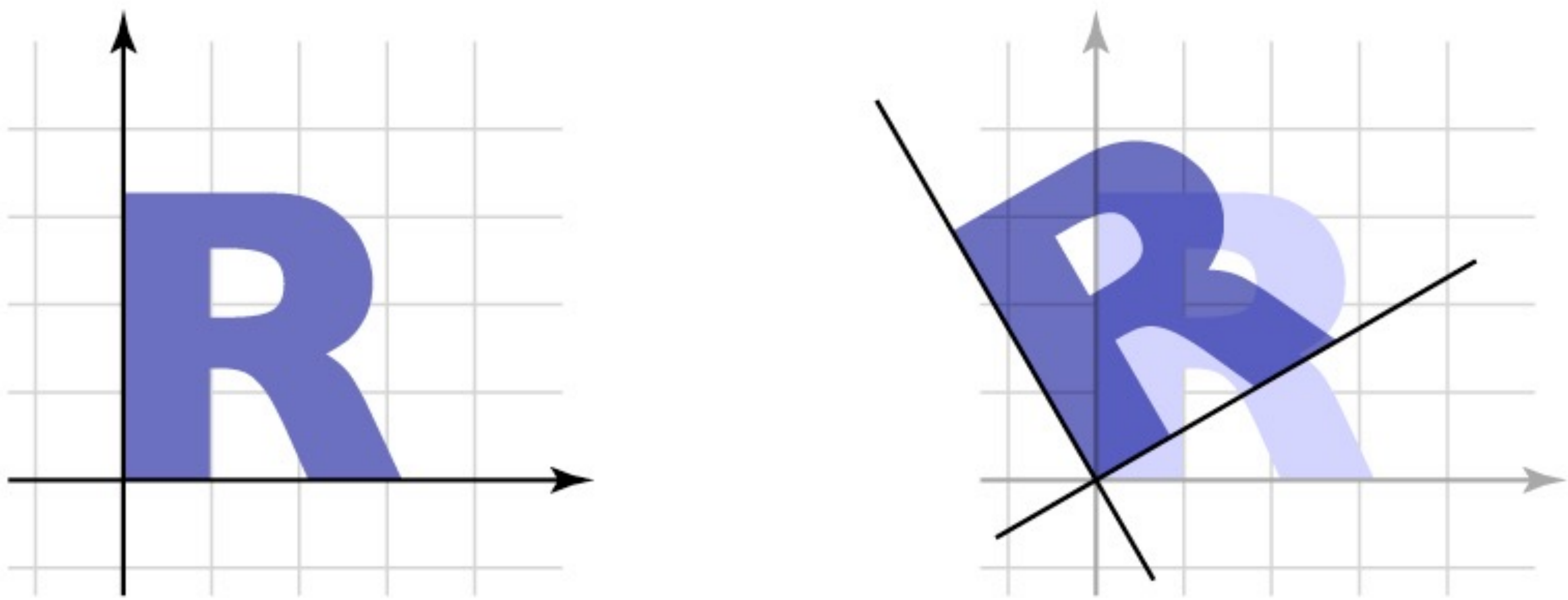
$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$



Linear transformation gallery

- Rotation $$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{bmatrix}$$

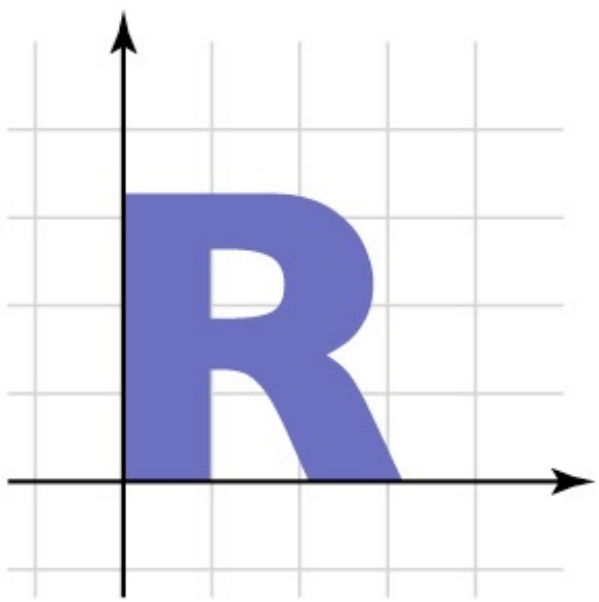
$$\begin{bmatrix} 0.866 & -0.5 \\ 0.5 & 0.866 \end{bmatrix}$$

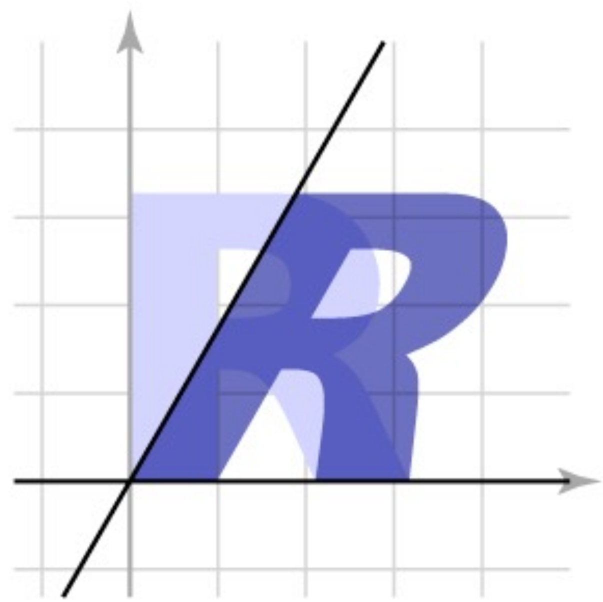


Linear transformation gallery

- Shear $\begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + ay \\ y \end{bmatrix}$

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





Composing transformations

- Want to move an object, then move it some more
  - $\mathbf{p} \to T(\mathbf{p}) \to S(T(\mathbf{p})) = (S \circ T)(\mathbf{p})$
- We need to represent *S* o *T* ("S compose T")
  - and would like to use the same representation as for *S* and *T*
- Translation easy
  - $T(\mathbf{p}) = \mathbf{p} + \mathbf{u}_T; S(\mathbf{p}) = \mathbf{p} + \mathbf{u}_S$

$$(S \circ T)(\mathbf{p}) = \mathbf{p} + (\mathbf{u}_T + \mathbf{u}_S)$$

- Translation by $\mathbf{u}_T$ then by $\mathbf{u}_S$ is translation by $\mathbf{u}_T + \mathbf{u}_S$
  - commutative!

Composing transformations

- Linear transformations also straightforward
  - $T(\mathbf{p}) = M_T\mathbf{p}; S(\mathbf{p}) = M_S\mathbf{p}$

$$(S \circ T)(\mathbf{p}) = M_S M_T \mathbf{p}$$

- Transforming first by M_T then by M_S is the same as transforming by $M_S M_T$
  - only sometimes commutative
    - e.g. rotations & uniform scales
    - e.g. non-uniform scales w/o rotation
  - Note $M_S M_T$, or $S \circ T$, is T first, then S

Combining linear with translation

- Need to use both in single framework
- Can represent arbitrary seq. as $T(\mathbf{p}) = M\mathbf{p} + \mathbf{u}$
  - $T(\mathbf{p}) = M_T\mathbf{p} + \mathbf{u}_T$
  - $S(\mathbf{p}) = M_S\mathbf{p} + \mathbf{u}_S$
  - $(S \circ T)(\mathbf{p}) = M_S(M_T\mathbf{p} + \mathbf{u}_T) + \mathbf{u}_S$
    $= (M_S M_T)\mathbf{p} + (M_S\mathbf{u}_T + \mathbf{u}_S)$
  - e. g. $S(T(0)) = S(\mathbf{u}_T)$
- Transforming by M_T and $\mathbf{u}_T$, then by M_S and $\mathbf{u}_S$, is the same as transforming by $M_S M_T$ and $\mathbf{u}_S + M_S\mathbf{u}_T$
  - This will work but is a little awkward

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

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

Homogeneous coordinates

- Represent translation using the extra column

$$\begin{bmatrix} 1 & 0 & t \\ 0 & 1 & s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t \\ y + s \\ 1 \end{bmatrix}$$

Homogeneous coordinates

- Composition just works, by 3x3 matrix multiplication

$$\begin{bmatrix} M_S & \mathbf{u}_S \\ 0 & 1 \end{bmatrix} \begin{bmatrix} M_T & \mathbf{u}_T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}$$
$$= \begin{bmatrix} (M_S M_T)\mathbf{p} + (M_S \mathbf{u}_T + \mathbf{u}_S) \\ 1 \end{bmatrix}$$

- This is exactly the same as carrying around *M* and **u**
  - but cleaner
  - and generalizes in useful ways as we'll see later