The Rendering Pipeline – Transformations

Wolfgang Heidrich

Course News

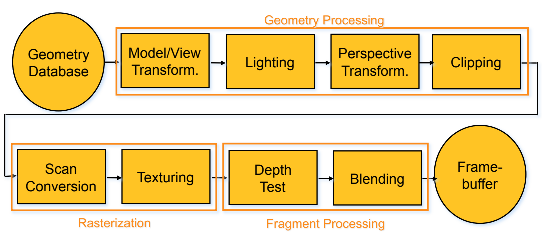
Assignment 0
- Due today!

Assignment 1
- Will be out Wednesday
- Due January 31

Homework 1
- Exercise problems for transformations
- Discussed in labs next week

Reading
- Chapter 5

The Rendering Pipeline



Modeling and Viewing Transformation

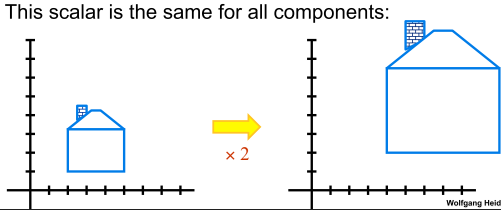
Affine transformations
- Linear transformations + translations
- Can be expressed as a *3x3* matrix + *3* vector

$$\mathbf{x}' = \mathbf{M} \cdot \mathbf{x} + \mathbf{t}$$

Scaling

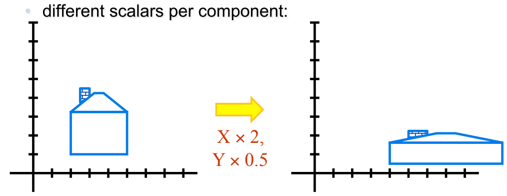
Scaling
- A coordinate means multiplying each of its components by a scalar

Uniform scaling
- This scalar is the same for all components:



Scaling

Non-uniform scaling:
- different scalars per component:



How can we represent this in matrix form?

Scaling (2D)

Scaling operation: $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} ax \\ by \end{pmatrix}$

Or, in matrix form: $\begin{pmatrix} x' \\ y' \end{pmatrix} = \underbrace{\begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}}_{\text{scaling matrix}} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$

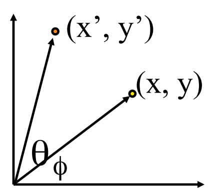
scaling matrix

Scaling (3D)

Scaling operation: $\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} ax \\ by \\ cz \end{pmatrix}$

Or, in matrix form: $\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix}$

2D Rotation From Trig Identities



x = r cos (φ)
y = r sin (φ)
x' = r cos (φ + θ)
y' = r sin (φ + θ)

Trig Identity…
x' = r cos(φ) cos(θ) – r sin(φ) sin(θ)
y' = r sin(φ) cos(θ) + r cos(φ) sin(θ)

Substitute…
x' = x **cos**(θ) - y **sin**(θ)
y' = x **sin**(θ) + y **cos**(θ)

2D Rotation Matrix

Easy to capture in matrix form:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

Even though sin(q) and cos(q) are nonlinear functions of q,
- x' is a linear combination of x and y
- y' is a linear combination of x and y

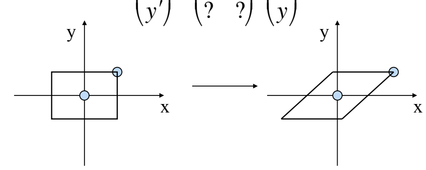
3D Rotation

- About x axis: $\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix}$
- About y axis: $\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix}$
- About z axis: $\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix}$

Shear

Shear along x axis
- push points to right in proportion to height

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} ? & ? \\ ? & ? \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

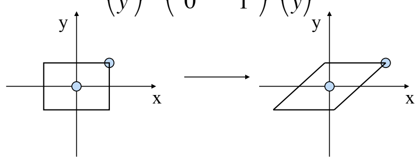


Shear

Shear along x axis

- push points to right in proportion to height

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & sh_x \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

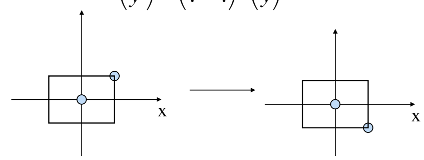


Reflection

Reflect across x axis

- Mirror

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} ? & ? \\ ? & ? \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

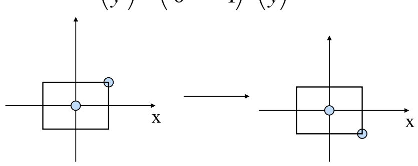


Reflection

Reflect across x axis

- Mirror

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

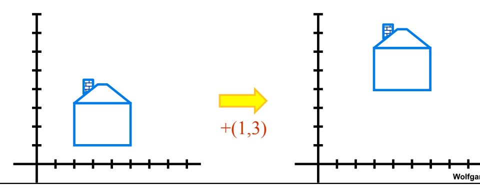


Affine Transformations

Translation:

- Add a constant (2D or 3D) vector:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$



Compositing of Affine Transformations

In general:

- Transformation of geometry into coordinate system where operation becomes simpler
- Perform operation
- Transform geometry back to original coordinate system

Compositing of Affine Transformations

Example: 2D rotation around arbitrary center

- Consider this transformation

$$\mathbf{x}' = \underbrace{\mathbf{Id} \cdot (\overbrace{R(\phi) \cdot \underbrace{(\mathbf{Id} \cdot \mathbf{x} - \mathbf{t})}_{\text{translate by } -\mathbf{t}}}^{\text{rotate by } \phi}) + \mathbf{t}}_{\text{translate by } \mathbf{t}}$$

- i.e:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & \\ & 1 \end{pmatrix} \cdot \left(\begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \cdot \left(\begin{pmatrix} 1 & \\ & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} a \\ b \end{pmatrix} \right) \right) + \begin{pmatrix} a \\ b \end{pmatrix}$$

Compositing of Affine Transformations

Composite transformation:

- Note that this is again an affine transformation

$$\mathbf{x}' = \mathbf{Id} \cdot (R(\phi) \cdot (\mathbf{Id} \cdot \mathbf{x} - \mathbf{t})) + \mathbf{t}$$
$$= \mathbf{Id} \cdot (R(\phi) \cdot \mathbf{x} - R(\phi) \cdot \mathbf{t}) + \mathbf{t}$$
$$= R(\phi) \cdot \mathbf{x} + (R(\phi) \cdot (-\mathbf{t}) + \mathbf{t})$$
$$= R(\phi) \cdot \mathbf{x} + \mathbf{t}'$$

This holds in general!

- All composites of affine transformations are themselves affine transformations!

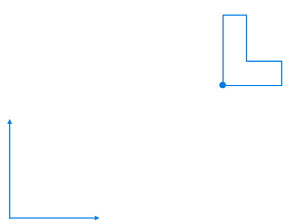
Compositing of Affine Transformations

Two different interpretations of composite:

- 1) read from inside-out as transformation of object
  - *1a) Translate object by –t*
  - *1b) Rotate object by Φ*
  - *1c) Translate object by t*
- 2) read from outside-in as transformation of the coordinate frame
  - *2c) Translate frame by t*
  - *2b) Rotate frame by –Φ (i.e. rotate object by Φ)*
  - *2a) Translate frame by –t*

Compositing of Affine Transformations

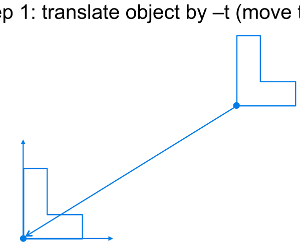
Example scene:



Compositing of Affine Transformations

First Interpretation:

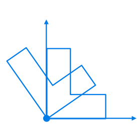
- Step 1: translate object by –t (move to origin)



Compositing of Affine Transformations

First Interpretation:

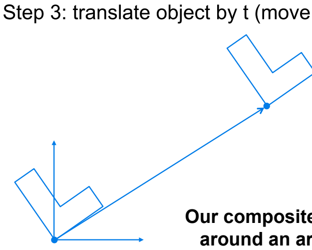
- Step 2: rotate object by Φ



Compositing of Affine Transformations

First Interpretation:

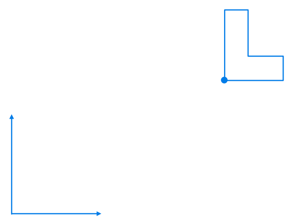
- Step 3: translate object by t (move back)



Our composite example is a rotation around an arbitrary 2D point with position t!

Compositing of Affine Transformations

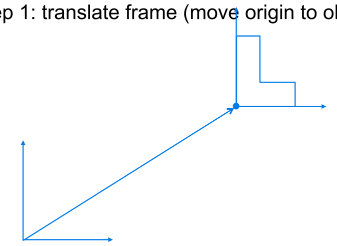
Example scene, second interpretation:



Compositing of Affine Transformations

Second interpretation:

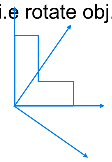
- Step 1: translate frame (move origin to object)



Compositing of Affine Transformations

Second interpretation:

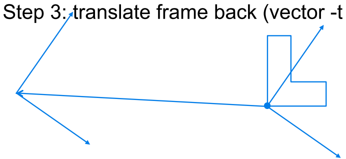
- Step 2: rotate frame by -Φ (i.e rotate obj. by Φ)



Compositing of Affine Transformations

Second interpretation:

- Step 3: translate frame back (vector -t in new frame!)



Compositing of Affine Transformations

NOTES:

- All transformations are **always with respect to the current coordinate frame**
- The results of both interpretations are **identical**
  - *Note that the object has the same relative position and orientation with respect to the coordinate frame!*

Compositing of Affine Transformations

Another Example: 3D rotation around arbitrary axis

- Rotate axis to z-axis
- Rotate by ϕ around z-axis
- Rotate z-axis back to original axis
- Composite transformation:

$$R(v,\phi) = R_z^{-1}(\alpha)\cdot R_y^{-1}(\beta)\cdot R_z(\phi)\cdot R_y(\beta)\cdot R_z(\alpha)$$

$$= (R_y(\beta)\cdot R_z(\alpha))^{-1}\cdot R_z(\phi)\cdot (R_y(\beta)\cdot R_z(\alpha))$$

Properties of Affine Transformations

Definition:

- A *linear combination* of points or vectors is given as

$$\mathbf{x} = \sum_{i=1}^{n} a_i \cdot \mathbf{x}_i, \text{ for } a_i \in \Re$$

- An affine combination of points or vectors is given as

$$\mathbf{x} = \sum_{i=1}^{n} a_i \cdot \mathbf{x}_i, \text{ with } \sum_{i=1}^{n} a_i = 1$$

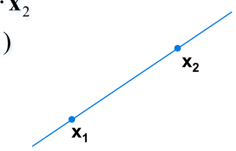
Wolfgang Heidrich

Properties of Affine Transformations

Example:

- Affine combination of 2 points

$$\begin{aligned} \mathbf{x} &= a_1 \cdot \mathbf{x}_1 + a_2 \cdot \mathbf{x}_2, \text{ with } a_1 + a_2 = 1 \\ &= (1 - a_2) \cdot \mathbf{x}_1 + a_2 \cdot \mathbf{x}_2 \\ &= \mathbf{x}_1 + a_2 \cdot (\mathbf{x}_2 - \mathbf{x}_1) \end{aligned}$$

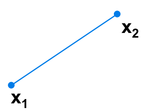


Wolfgang Heidrich

Properties of Affine Transformations

Definition:

- A convex combination is an affine combination where all the weights a_i are positive
- Note: this implies $0 \le a_i \le 1$, $i = 1 \ldots n$



Wolfgang Heidrich

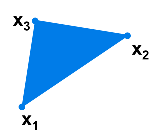
Properties of Affine Transformations

Example:

- Convex combination of 3 points

$$\mathbf{x} = \alpha \cdot \mathbf{x}_1 + \beta \cdot \mathbf{x}_2 + \gamma \cdot \mathbf{x}_3$$
$$\text{with } \alpha + \beta + \gamma = 1, \; 0 \le \alpha, \beta, \gamma \le 1$$

- α, β, and γ are called *Barycentric coordinates*



Wolfgang Heidrich

Properties of Affine Transformations

Theorem:

- The following statements are synonymous
  - *A transformation T(x) is affine, i.e.:*

$$\mathbf{x}' = T(\mathbf{x}) := \mathbf{M} \cdot \mathbf{x} + \mathbf{t},$$
$$\text{for some matrix } \mathbf{M} \text{ and vector } \mathbf{t}$$

  - *T(x) preserves affine combinations, i.e.*

$$T\left(\sum_{i=1}^{n} a_i \cdot \mathbf{x}_i\right) = \sum_{i=1}^{n} a_i \cdot T(\mathbf{x}_i), \text{ for } \sum_{i=1}^{n} a_i = 1$$

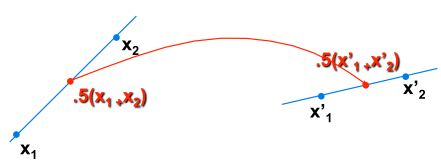
  - *T(x) maps parallel lines to parallel lines*

Wolfgang Heidrich

Properties of Affine Transformations

Preservation of affine combinations:

- Can compute transformation of every point on line or triangle by simply transforming the *control points*



Wolfgang Heidrich

Coming Up

This week:

- Affine Transformations with Homogeneous Coordinates
- Transformation Hierarchies

Next week

- Perspective Transformations