Affine Transformations & Homogeneous Coordinates

Wolfgang Heidrich

Course News

Assignment 1

- Due March 31
- More at end of lecture

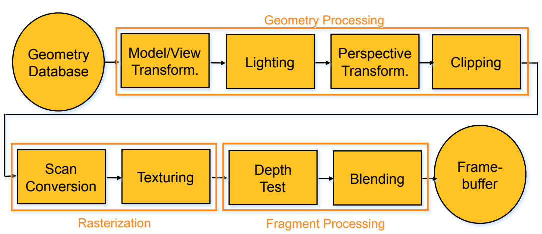
Homework 1

- Exercise problems for transformations
- Discussed in labs next week

Reading

- Chapter 5

The Rendering Pipeline



Recap: Modeling and Viewing Transformation

Affine transformations

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

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

Recap: 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

Recap: 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:

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

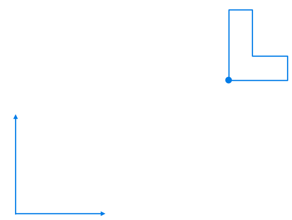
Recap: Compositing of Affine Transformations

Two different interpretations of composite:

- 1) read from inside-out as transformation of object
- 2) read from outside-in as transformation of the coordinate frame by the **inverse** of the stated operation

Recap: Compositing of Affine Transformations

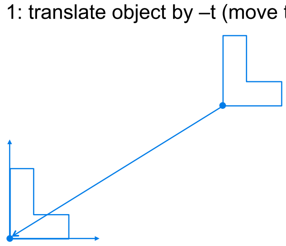
Example scene:



Recap: Compositing of Affine Transformations

First Interpretation:

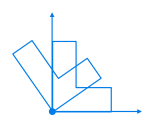
- Step 1: translate object by –t (move to origin)



Recap: Compositing of Affine Transformations

First Interpretation:

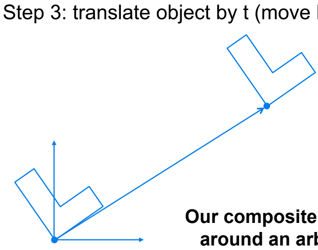
- Step 2: rotate object by Φ



Recap: Compositing of Affine Transformations

First Interpretation:

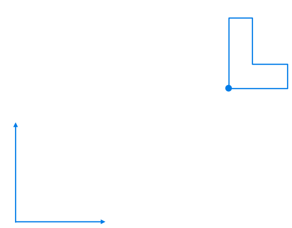
- Step 3: translate object by t (move back)



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

Recap: Compositing of Affine Transformations

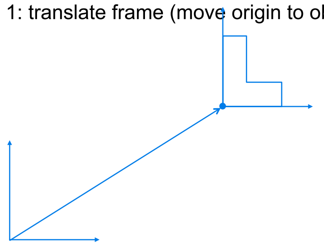
Example scene, second interpretation:



Recap: Compositing of Affine Transformations

Second interpretation:

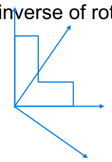
- Step 1: translate frame (move origin to object)



Recap: Compositing of Affine Transformations

Second interpretation:

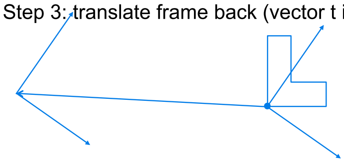
- Step 2: rotate frame by -Φ (inverse of rot. by Φ)



Recap: Compositing of Affine Transformations

Second interpretation:

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



Recap: 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:

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

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

Compositing of Affine Transformations

Yet another example (on whiteboard):

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \cdot \begin{pmatrix} \cos\frac{\pi}{4} & -\sin\frac{\pi}{4} \\ \sin\frac{\pi}{4} & \cos\frac{\pi}{4} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

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

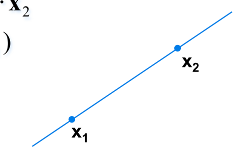
$$\mathbf{x} = \sum_{i=1}^{n} a_i \cdot \mathbf{x}_i, \text{ with } \sum_{i=1}^{n} a_i = 1$$

Properties of Affine Transformations

Example:

- Affine combination of 2 points

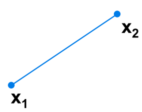
$$\begin{aligned}\mathbf{x} &= a_1 \cdot \mathbf{x}_1 + a_2 \cdot \mathbf{x}_2, \text{ with } a_1 + a_2 = 1 \\ &= (1 - a_2) \cdot \mathbf{x}_1 + a_2 \cdot \mathbf{x}_2 \\ &= \mathbf{x}_1 + a_2 \cdot (\mathbf{x}_2 - \mathbf{x}_1)\end{aligned}$$



Properties of Affine Transformations

Definition:

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



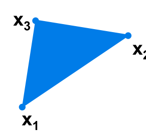
Properties of Affine Transformations

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*



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.*

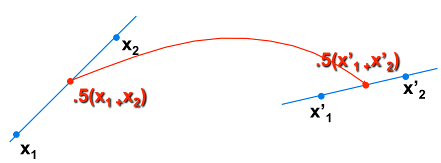
$$T\left(\sum_{i=1}^{n} a_i \cdot \mathbf{x}_i\right) = \sum_{i=1}^{n} a_i \cdot T(\mathbf{x}_i), \text{ for } \sum_{i=1}^{n} a_i = 1$$

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

Properties of Affine Transformations

Preservation of affine combinations:

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



Homogeneous Coordinates

Homogeneous representation of points:

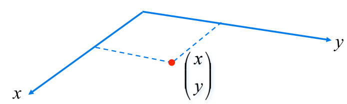
- Add an additional component $w=1$ to all *points*
- All multiples of this vector are considered to represent the same 3D point
- Use square brackets (rather than round ones) to denote homogeneous coordinates (different from text book!)

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \equiv \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \equiv \begin{bmatrix} x \cdot w \\ y \cdot w \\ z \cdot w \\ w \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix}, \forall w \neq 0$$

Wolfgang Heidrich

Geometrically In 2D

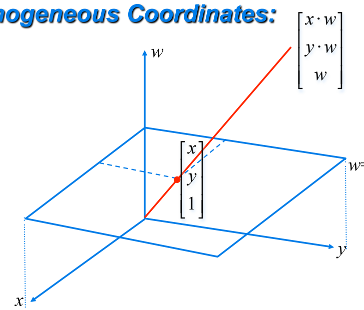
Cartesian Coordinates:



Wolfgang Heidrich

Geometrically In 2D

Homogeneous Coordinates:



Wolfgang Heidrich

Homogeneous Matrices

Affine Transformations

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & 0 \\ m_{2,1} & m_{2,2} & m_{2,3} & 0 \\ m_{3,1} & m_{3,2} & m_{3,3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & 0 \\ m_{2,1} & m_{2,2} & m_{2,3} & 0 \\ m_{3,1} & m_{3,2} & m_{3,3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & t_x \\ 0 & 0 & 0 & t_y \\ 0 & 0 & 0 & t_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Wolfgang Heidrich

Homogeneous Matrices

Combining the two matrices into one:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & 0 \\ m_{2,1} & m_{2,2} & m_{2,3} & 0 \\ m_{3,1} & m_{3,2} & m_{3,3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & t_x \\ 0 & 0 & 0 & t_y \\ 0 & 0 & 0 & t_z \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & t_x \\ m_{2,1} & m_{2,2} & m_{2,3} & t_y \\ m_{3,1} & m_{3,2} & m_{3,3} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Wolfgang Heidrich

Homogeneous Coordinates – Composite Transformations

Example: 2D rotation around arbitrary center

- This:

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

- Corresponds to this in full expansion:

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

Wolfgang Heidrich

Homogeneous Coordinates – Composite Transformations

Example: 2D rotation around arbitrary center

- Euclidean coordinates:

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

- Homogeneous coordinates:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & & a \\ & 1 & b \\ & & 1 \end{bmatrix}}_{translation} \cdot \underbrace{\begin{bmatrix} \cos\phi & -\sin\phi & \\ \sin\phi & \cos\phi & \\ & & 1 \end{bmatrix}}_{rotation} \cdot \underbrace{\begin{bmatrix} 1 & & -a \\ & 1 & -b \\ & & 1 \end{bmatrix}}_{translation} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Wolfgang Heidrich

Homogeneous Transformations

Notes:

- A composite transformation is now just the product of a few matrixes
- Rather than multiply each point sequentially with 3 matrices, first multiply the matrices, then multiply each point with only one (composite) matrix
  - *Much faster for large # of points!*
- The composite matrix describing the affine transformation always has the bottom row
  0,0,1 (2D), or
  0,0,0,1 (3D)

Wolfgang Heidrich

Homogeneous Matrices

Note:

- Multiplication of the matrix with a constant does not change the transformation!

$$\tilde{T}\left(\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}\right) = \begin{bmatrix} m_{1,1}\cdot k & m_{1,2}\cdot k & m_{1,3}\cdot k & t_x\cdot k \\ m_{2,1}\cdot k & m_{2,2}\cdot k & m_{2,3}\cdot k & t_y\cdot k \\ m_{3,1}\cdot k & m_{3,2}\cdot k & m_{3,3}\cdot k & t_z\cdot k \\ 0 & 0 & 0 & k \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x'\cdot k \\ y'\cdot k \\ z'\cdot k \\ k \end{bmatrix}$$

$$\equiv \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = T\left(\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}\right)$$

Wolfgang Heidrich

Homogeneous Vectors

Earlier discussion describes points only

- What about vectors (directions)?
- What is the affine transformation of a vector?
  - *Rotation*
  - *Scaling*
  - *Translation*

Vectors are invariant under translation!

Wolfgang Heidrich

Homogeneous Vectors

Representing vectors in homogeneous coordinates

- Need representation that is only affected by linear transformations, but not by translations
- This is achieved by setting $w=0$

$$T\left(\begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}\right) = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} & t_x \\ m_{2,1} & m_{2,2} & m_{2,3} & t_y \\ m_{3,1} & m_{3,2} & m_{3,3} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 0 \end{bmatrix}$$

Wolfgang Heidrich

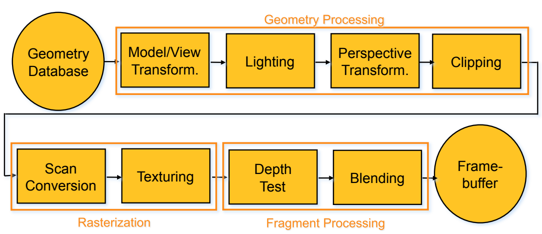
Homogeneous Coordinates

Properties

- Unified representation as 4-vector (in 3D) for
  - *Points*
  - *Vectors / directions*
- Affine transformations become 4x4 matrices
  - *Composing multiple affine transformations involves simply multiplying the matrices*
  - *3D affine transformations have 12 degrees of freedom*
    - Need mapping of 4 points to uniquely define transformation

Wolfgang Heidrich

The Rendering Pipeline



Modeling Transformation

Purpose:

- Map geometry from local *object coordinate system* into a global *world coordinate system*
- *Same as placing objects*

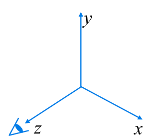
Transformations:

- Arbitrary affine transformations are possible
  - *Even more complex transformations may be desirable, but are not available in hardware*
    - Freeform deformations

Viewing Transformation

Purpose:

- Map geometry from *world coordinate system* into *camera coordinate system*
- Camera coordinate system is *right-handed*, viewing direction is *negative* z-axis
- Same a placing camera



Transformations:

- Usually only *rigid body transformations*
  - *Rotations and translations*
- Objects have same size and shape in camera and world coordinates

Model/View Transformation

Combine modeling and viewing transform.

- Combine both into a single matrix
- Saves computation time if many points are to be transformed
- Possible because the viewing transformation directly follows the modeling transformation without intermediate operations

Coming Up

Next time:

- Transformation hierarchies
- OpenGL commands for transformations/drawing

Next week:

- Perspective transformations