Computer Graphics
Homogeneous Notation

Matthias Teschner
What is visible at the sensor?

- Visibility can be resolved by ray casting or by applying transformations.

Ray Casting computes ray-scene intersections to estimate \( q \) from \( p \).

Rasterizers apply transformations to \( p \) in order to estimate \( q \). \( p \) is projected onto the sensor plane.
Outline

– Motivation
– Homogeneous notation
– Transformations
Motivation

- Transformations in modeling and rendering
  - Position, reshape, and animate objects, lights, cameras
  - Project 3D geometry onto the camera plane
- Homogeneous notation
  - 3D vertices (positions) and 3D normals (directions) are represented with 4D vectors
  - Transformations are represented with 4x4 matrices
  - All transformations of positions and directions are consistently realized as a matrix-vector product
Transformations – 2D

Four faces / primitives / polygons, four points / vertices, four normals.

Translation.

Scale.

Identity transform.

Rotation.

Shear.

Transformations change vertex positions and surface normals.
Coordinate Systems and Transformations

Local coordinate system of an object

Local coordinate system of a camera

Model transforms $M_1, M_2, M_3$

View transform $V$

Global coordinate system with one camera and three instances of the same object
Coordinate Systems and Transformations

Global coordinate system with one camera and three objects

Inverse view transform $V^{-1}$ applied to all objects and the camera

View space / Camera space.

Working in view space is motivated by simplified implementations. E.g., rays start at 0 in view space.
Modelview Transform

Transformation from local into view space is realized with the modelview transform. Objects: $V^1M_1, V^1M_2, V^1M_3$ Camera: $V^1V = I$
More Transformations

– To transform from view space positions to positions on the camera plane
  – Projection transform
  – Viewport transform
– See lecture on projections
Transformations - Groups

- Translation, rotation, reflection
  - Preserve shape and size
  - Congruent transformations (Euclidean transformations)
- Translation, rotation, reflection, scale
  - Preserve shape
  - Similarity transformations
Affine Transformations

- Translation, rotation, reflection, scale, shear
  - Angles and lengths are not preserved
  - Preserve collinearity
    - Points on a line are transformed to points on a line
  - Preserve proportions
    - Ratios of distances between points are preserved
  - Preserve parallelism
    - Parallel lines are transformed to parallel lines
Affine Transformations

- 3D position $p$: $p' = T(p) = Ap + t$
- Affine transformations preserve affine combinations $T(\sum_i \alpha_i \cdot p_i) = \sum_i \alpha_i \cdot T(p_i)$ for $\sum_i \alpha_i = 1$
- E.g., a line can be transformed by transforming its control points

$$x = \alpha_1 p_1 + \alpha_2 p_2$$
$$x' = T(x) = \alpha_1 T(p_1) + \alpha_2 T(p_2)$$
Affine Transformations

- 3D position $p$: $p' = Ap + t$
- 3x3 matrix $A$ represents linear transformations
  - Scale, rotation, shear
- 3D vector $t$ represents translation
- Using the homogeneous notation, all affine transformations are represented with one matrix-vector multiplication
Positions and Vectors

- Positions / vertices specify a location in space
- Vectors / normals specify a direction
- Relations
  \[ \text{position} - \text{position} = \text{vector} \]
  \[ \text{position} + \text{vector} = \text{position} \]
  \[ \text{vector} + \text{vector} = \text{vector} \]
  \[ \text{position} + \text{position} \text{ not defined} \]
Positions and Vectors

- Transformations can have different effects on positions and vectors
  - E.g., translation of a point changes its position, but translation of a vector does not change the vector
- Using the homogeneous notation, transformations of vectors and positions are handled in a unified way
Outline

– Motivation
– Homogeneous notation
– Transformations
Homogeneous Coordinates of Positions

- \([x, y, z, w]^T\) with \(w \neq 0\) are the homogeneous coordinates of the 3D position \((\frac{x}{w}, \frac{y}{w}, \frac{z}{w})^T\)

- \([\lambda x, \lambda y, \lambda z, \lambda w]^T\) represents the same position \((\frac{\lambda x}{\lambda w}, \frac{\lambda y}{\lambda w}, \frac{\lambda z}{\lambda w})^T = (\frac{x}{w}, \frac{y}{w}, \frac{z}{w})^T\) for all \(\lambda \neq 0\)

Examples

- \([2, 3, 4, 1]^T \sim (2, 3, 4)^T\)
- \([2, 4, 6, 1]^T \sim (2, 4, 6)^T\)
- \([4, 8, 12, 2]^T \sim (2, 4, 6)^T\)
- \([0.2, 0.4, 0.6, 0.1]^T \sim (2, 4, 6)^T\)
Homogeneous Coordinates of Positions

– From Cartesian to homogeneous coordinates
\[(x, y, z)^T \rightarrow [x, y, z, 1]^T\] The most obvious way, but an infinite number of options.
\[(x, y, z)^T \rightarrow [\lambda x, \lambda y, \lambda z, \lambda]^T \quad \lambda \neq 0\]

– From homogeneous to Cartesian coordinates
\[[x, y, z, w]^T \rightarrow (\frac{x}{w}, \frac{y}{w}, \frac{z}{w})^T\]
1D Illustration

- Homogeneous points $[\lambda x, \lambda]^T$ represent the same position $x$ in Cartesian space.
- Homogeneous points $[\lambda x, \lambda]^T$ lie on a line in the 2D space $[x, w]$. 

\[
\begin{align*}
[\lambda x, \lambda]^T &\sim x_1 \\
[\lambda_2 x_1, \lambda_2]^T &\sim x_1 \\
[\lambda_1 x_1, \lambda_1]^T &\sim x_1 \\
[x_1, 1]^T &\sim x_1 \\
w = 1
\end{align*}
\]
Homogeneous Coordinates of Vectors

- For varying $w$, a point $[x, y, z, w]^T$ is scaled and the points $(\frac{x}{w}, \frac{y}{w}, \frac{z}{w})^T$ represent a line in 3D space.
- The direction of this line is $(x, y, z)^T$.
- For $w \to 0$, the position $(\frac{x}{w}, \frac{y}{w}, \frac{z}{w})^T$ moves to infinity in the direction $(x, y, z)^T$.
- $[x, y, z, 0]^T$ is a position at infinity in the direction of $(x, y, z)^T$.
- $[x, y, z, 0]^T$ is a vector in the direction of $(x, y, z)^T$. 
1D Illustration

\[ [x_0, w_1]^T \sim x_1 \]

\[ [x_0, w_2]^T \sim x_2 \]

\[ [x_0, w_3]^T \sim x_3 \]
Positions at Infinity

- Can be processed by graphics APIs, e.g. OpenGL
- Used, e.g. in shadow volumes

Rendering of a triangle with vertices

\[
\begin{bmatrix}
0 \\
0.5 \\
0 \\
1
\end{bmatrix}
\begin{bmatrix}
-0.5 \\
0 \\
0 \\
1
\end{bmatrix}
\begin{bmatrix}
0.5 \\
0 \\
0 \\
1
\end{bmatrix}
\]

Rendering of a triangle with vertices

\[
\begin{bmatrix}
0 \\
0.5 \\
0 \\
0
\end{bmatrix}
\begin{bmatrix}
-0.5 \\
0 \\
0 \\
1
\end{bmatrix}
\begin{bmatrix}
0.5 \\
0 \\
0 \\
1
\end{bmatrix}
\]
Positions and Vectors

- If positions are in normalized form, position-vector relations can be represented

\[
\begin{align*}
\text{vector} + \text{vector} &= \text{vector} \\
\begin{bmatrix} u_x \\ u_y \\ u_z \\ 0 \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_z \\ 0 \end{bmatrix} &= \begin{bmatrix} u_x + v_x \\ u_y + v_y \\ u_z + v_z \\ 0 \end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\text{position} + \text{vector} &= \text{position} \\
\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} + \begin{bmatrix} v_x \\ v_y \\ v_z \\ 0 \end{bmatrix} &= \begin{bmatrix} p_x + v_x \\ p_y + v_y \\ p_z + v_z \\ 1 \end{bmatrix}
\end{align*}
\]

\[
\begin{align*}
\text{position} - \text{position} &= \text{vector} \\
\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} - \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} &= \begin{bmatrix} p_x - r_x \\ p_y - r_y \\ p_z - r_z \\ 0 \end{bmatrix}
\end{align*}
\]
Homogeneous Notation of Linear Transformations

\[
\begin{pmatrix}
  m_{00} & m_{01} & m_{02} \\
  m_{10} & m_{11} & m_{12} \\
  m_{20} & m_{21} & m_{22}
\end{pmatrix}
\begin{pmatrix}
  p_x \\
  p_y \\
  p_z
\end{pmatrix}
\sim
\begin{pmatrix}
  m_{00} & m_{01} & m_{02} & 0 \\
  m_{10} & m_{11} & m_{12} & 0 \\
  m_{20} & m_{21} & m_{22} & 0 \\
  0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
  p_x \\
  p_y \\
  p_z \\
  1
\end{pmatrix}
\]

– If the transform of \( \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \) results in \( \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} \), then

the transform of \( \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} \) results in \( \begin{pmatrix} r_x \\ r_y \\ r_z \\ 1 \end{pmatrix} \sim \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} \).
Affine Transformations and Projections

- General form

\[
\begin{bmatrix}
  m_{00} & m_{01} & m_{02} & t_0 \\
  m_{10} & m_{11} & m_{12} & t_1 \\
  m_{20} & m_{21} & m_{22} & t_2 \\
  p_0 & p_1 & p_2 & w
\end{bmatrix}
\]

- $m_{ij}$ represent rotation, scale, shear
- $t_i$ represent translation
- $p_i$ are used for projections (see lecture on projections)
- $w$ is the homogeneous component
Homogeneous Coordinates - Summary

- \([x, y, z, w]^T\) with \(w \neq 0\) are the homogeneous coordinates of the 3D position \(\left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}\right)^T\)
- \([x, y, z, 0]^T\) is a point at infinity in the direction of \((x, y, z)^T\)
- \([x, y, z, 0]^T\) is a vector in the direction of \((x, y, z)^T\)
- \[
\begin{bmatrix}
m_{00} & m_{01} & m_{02} & t_0 \\
m_{10} & m_{11} & m_{12} & t_1 \\
m_{20} & m_{21} & m_{22} & t_2 \\
p_0 & p_1 & p_2 & w
\end{bmatrix}
\] is a transformation that represents rotation, scale, shear, translation, projection
Outline

– Motivation
– Homogeneous notation
– Transformations
Transformations

Four faces / primitives / polygons, four points / vertices, four normals.

Translation

Scale

Identity transform.

Rotation

Shear
Translation

- Of a position

\[ T(t)p = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} p_x + t_x \\ p_y + t_y \\ p_z + t_z \\ 1 \end{bmatrix} \]

- Of a vector

\[ T(t)v = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ 0 \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ 0 \end{bmatrix} \]

- Inverse transform

\[ T^{-1}(t) = T(-t) \]
Rotation

- Positive (anticlockwise) rotation with angle $\phi$ around the $x$, $y$, $z$-axis

Rotation matrices for rotations around arbitrary axes are built by combining simple rotations and translations.

$$R_x(\phi) = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos \phi & -\sin \phi & 0 \\
0 & \sin \phi & \cos \phi & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}$$

$$R_y(\phi) = \begin{bmatrix}
\cos \phi & 0 & \sin \phi & 0 \\
0 & 1 & 0 & 0 \\
-\sin \phi & 0 & \cos \phi & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}$$

$$R_z(\phi) = \begin{bmatrix}
\cos \phi & -\sin \phi & 0 & 0 \\
\sin \phi & \cos \phi & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}$$
Rotation – Inverse Transform

– The inverse of a rotation matrix is its transpose

\[
R_x(-\phi) = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos(-\phi) & -\sin(-\phi) & 0 \\
0 & \sin(-\phi) & \cos(-\phi) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & \cos(\phi) & \sin(\phi) & 0 \\
0 & -\sin(\phi) & \cos(\phi) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} = R_x^T(\phi)
\]

\[
R_x^{-1} = R_x^T \\
R_y^{-1} = R_y^T \\
R_z^{-1} = R_z^T
\]
Mirroring / Reflection

- Mirroring with respect to $x = 0, y = 0, z = 0$ plane
- Changes the sign of the $x$, $y$, $z$-component

$$P_x = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad P_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad P_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- The inverse of a reflection is its transpose

$$P_x^{-1} = P_x^T \quad P_y^{-1} = P_y^T \quad P_z^{-1} = P_z^T$$
Orthogonal Matrices

- Rotation and reflection matrices are orthogonal
  \[ RR^T = R^T R = I \quad R^{-1} = R^T \]
- \( R_1, R_2 \) are orthogonal \( \Rightarrow R_1 R_2 \) is orthogonal
- Rotation: \( \det R = 1 \), Reflection: \( \det R = -1 \)
- Length of a vector is preserved \( \|Rv\| = \|v\| \)
- Angles are preserved \( \langle Ru, Rv \rangle = \langle u, v \rangle \)
Scale

- Scaling \( x, y, z \)-components of a position or vector

\[
S(s_x, s_y, s_z)p = \begin{bmatrix}
    s_x & 0 & 0 & 0 \\
    0 & s_y & 0 & 0 \\
    0 & 0 & s_z & 0 \\
    0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
p_x \\
p_y \\
p_z \\
1
\end{bmatrix} = \begin{bmatrix}
    s_xp_x \\
    s_y p_y \\
    s_z p_z \\
    1
\end{bmatrix}
\]

- Inverse \( S^{-1}(s_x, s_y, s_z) = S\left(\frac{1}{s_x}, \frac{1}{s_y}, \frac{1}{s_z}\right)\)

- Uniform scaling: \( s_x = s_y = s_z = s \)

\[
S(s) = \begin{bmatrix}
s & 0 & 0 & 0 \\
0 & s & 0 & 0 \\
0 & 0 & s & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

or, e.g. \( S(s) = \begin{bmatrix}
    1 & 0 & 0 & 0 \\
    0 & 1 & 0 & 0 \\
    0 & 0 & 1 & 0 \\
    0 & 0 & 0 & \frac{1}{s}
\end{bmatrix} \)
Shear

- Offset of one component with respect to another component
- Six shear modes in 3D
- E.g., shear of $x$ with respect to $z$

\[
H_{xz}(s)p = \begin{bmatrix}
1 & 0 & s & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
p_x \\
p_y \\
p_z \\
1
\end{bmatrix}
= \begin{bmatrix}
p_x + sp_z \\
p_y \\
p_z \\
1
\end{bmatrix}
\]

- Inverse $H_{xz}^{-1}(s) = H_{xz}(-s)$
Compositing Transformations

Composition is achieved by matrix multiplication

- A translation $T$ applied to $p$, followed by a rotation $R$
  \[ R(Tp) = (RT)p \]
- A rotation $R$ applied to $p$, followed by a translation $T$
  \[ T(Rp) = (TR)p \]
- Note that generally $TR \neq RT$
- The order of composed transformations matters
Examples

- Rotation around a line through $t$ parallel to the $x$, $y$, $z$-axis

$T(t)R_{xyz}(\phi)T(-t)$

- Scale with respect to an arbitrary axis

$R_{xyz}(\phi)S(s_x, s_y, s_z)R_{xyz}(-\phi)$

- E.g., $b_1, b_2, b_3$ represent an orthonormal basis, then scaling along these vectors is realized with

$$
\begin{pmatrix}
  b_1 & b_2 & b_3 & 0 \\
  0 & 0 & 0 & 1 
\end{pmatrix}
S(s_x, s_y, s_z)
\begin{pmatrix}
  b_1 & b_2 & b_3 & 0 \\
  0 & 0 & 0 & 1 
\end{pmatrix}^T
$$
We want to rotate the object points $p_i$ around point $t$.

Translation by $-t$.  

$$T(-t)p_i$$

Rotation by $\phi$.  

$$R(\phi)T(-t)p_i$$

Translation by $t$.  

$$T(t)R(\phi)T(-t)p_i$$
Rigid-Body Transform

- In Cartesian coordinates: \( \mathbf{p}' = \mathbf{Rp} + \mathbf{t} \) with \( \mathbf{R} \) being a rotation and \( \mathbf{t} \) being a translation
- In homogeneous notation: 
  \[
  \begin{bmatrix}
  \mathbf{p}' \\
  1
  \end{bmatrix} = 
  \begin{bmatrix}
  \mathbf{R} & \mathbf{t} \\
  \mathbf{0}^\top & 1
  \end{bmatrix} 
  \begin{bmatrix}
  \mathbf{p} \\
  1
  \end{bmatrix}
  \]
- The inverse transform in Cartesian coordinates

  \[
  \mathbf{p} = \mathbf{R}^{-1}(\mathbf{p}' - \mathbf{t}) = \mathbf{R}^{-1}\mathbf{p}' - \mathbf{R}^{-1}\mathbf{t} = \mathbf{R}^\top\mathbf{p}' - \mathbf{R}^\top\mathbf{t}
  \]
- The inverse in homogeneous notation

  \[
  \begin{bmatrix}
  \mathbf{p} \\
  1
  \end{bmatrix} = 
  \begin{bmatrix}
  \mathbf{R} & \mathbf{t} \\
  \mathbf{0}^\top & 1
  \end{bmatrix}^{-1} 
  \begin{bmatrix}
  \mathbf{p}' \\
  1
  \end{bmatrix} = 
  \begin{bmatrix}
  \mathbf{R}^\top & -\mathbf{R}^\top\mathbf{t} \\
  \mathbf{0}^\top & 1
  \end{bmatrix} 
  \begin{bmatrix}
  \mathbf{p}' \\
  1
  \end{bmatrix}
  \]
Planes and Normals

- Planes can be represented by a surface normal $\mathbf{n}$ and a point $\mathbf{r}$. All points $\mathbf{p}$ with $\mathbf{n} \cdot (\mathbf{p} - \mathbf{r}) = 0$ form a plane.

$$nxpx + nyp_y + nzp_z + (-n_xrx - n_yry - n_zrz) = 0$$
$$nxpx + nyp_y + nzp_z + d = 0$$
$$(nx ny nz d)(px py pz 1)^T = 0$$
$$(nx ny nz d)A^{-1}A(px py pz 1)^T = 0$$

- The transformed points $A[px py pz 1]^T$ are on the plane represented by $(nx ny nz d)A^{-1} = ((A^{-1})^T(nx ny nz d)^T)^T$

- If a surface is transformed by $A$, its homogeneous notation (including the normal) is transformed by $(A^{-1})^T$
Basis Transform - Translation

- Two coordinate systems

\[ C_1 = (O_1, \{e_1, e_2, e_3\}) \]
\[ C_2 = (O_2, \{e_1, e_2, e_3\}) \]
\[ O_2 = T(t)O_1 \]
Basis Transform - Translation

- The coordinates of \( p_1 \) with respect to \( C_2 \) are given by \( p_2 = p_1 - t \) \( p_2 = T(-t)p_1 \)
- The coordinates of a point in the transformed basis correspond to the coordinates of the point in the untransformed basis transformed by the inverse basis transform
  - Translating the origin by \( t \) corresponds to translating the object by \(-t\)
  - Rotating the basis vectors by an angle corresponds to rotating the object by the same negative angle
Basis Transform - Rotation

- Two coordinate systems

\[ C_1 = (O, \{e_1, e_2, e_3\}) \]
\[ C_2 = (O, \{b_1, b_2, b_3\}) \]
Basis Transform - Rotation

- Coordinates of $p_1$ with respect to $C_2$ are given by

$$p_2 = \begin{pmatrix} b_1^T \\ b_2^T \\ b_3^T \end{pmatrix} p_1 \sim \begin{bmatrix} b_{1,x} & b_{1,y} & b_{1,z} & 0 \\ b_{2,x} & b_{2,y} & b_{2,z} & 0 \\ b_{3,x} & b_{3,y} & b_{3,z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}$$

- $b_1, b_2, b_3$ are the basis vectors of $C_2$ with respect to $C_1$
- $b_1, b_2, b_3$ are orthonormal, represent a rotation
- Rotating the basis vectors by an angle corresponds to rotating the object by the same negative angle
Basis Transform - Application

- The view transform can be seen as a basis transform
- Objects are in a global system $C_1 = (O_1, \{e_1, e_2, e_3\})$
- The camera is at $O_2$ and oriented with $\{b_1, b_2, b_3\}$
- After the view transform, all objects are represented in the eye or camera coordinate system $C_2 = (O_2, \{b_1, b_2, b_3\})$
- Placing and orienting the camera is a transformation $v$
- The basis transform is realized by applying $v^{-1}$ to all objects
View Transform

\[ C_2 = (O_2, \{b_1, b_2, b_3\}) \]

\[ C_1 = (O_1, \{e_1, e_2, e_3\}) \]

\[ C_1 \rightarrow V \rightarrow C_2 \]

Inverse view transform \( V^1 \) applied to all objects and the camera.

Global space.

View space / Camera space.
Summary

– Usage of the homogeneous notation is motivated by a unified processing of affine transformations, perspective projections, points, and vectors
– All transformations of points and vectors are represented by a matrix-vector multiplication
– “Undoing" a transformation is represented by its inverse
– Compositing of transformations is accomplished by matrix multiplication