Image Processing and Computer Graphics

Transformations and Homogeneous Coordinates

Matthias Teschner

Computer Science Department
University of Freiburg

UNI FREIBURG

Motivation

- transformations are used
  - to position, reshape, and animate objects, lights, and the virtual camera
  - to orthographically or perspectivly project three-dimensional geometry onto a plane
- transformations are represented with 4x4 matrices
- transformations are applied to vertices and normals
- vertices (positions) and normals (directions) are represented with 4D vectors

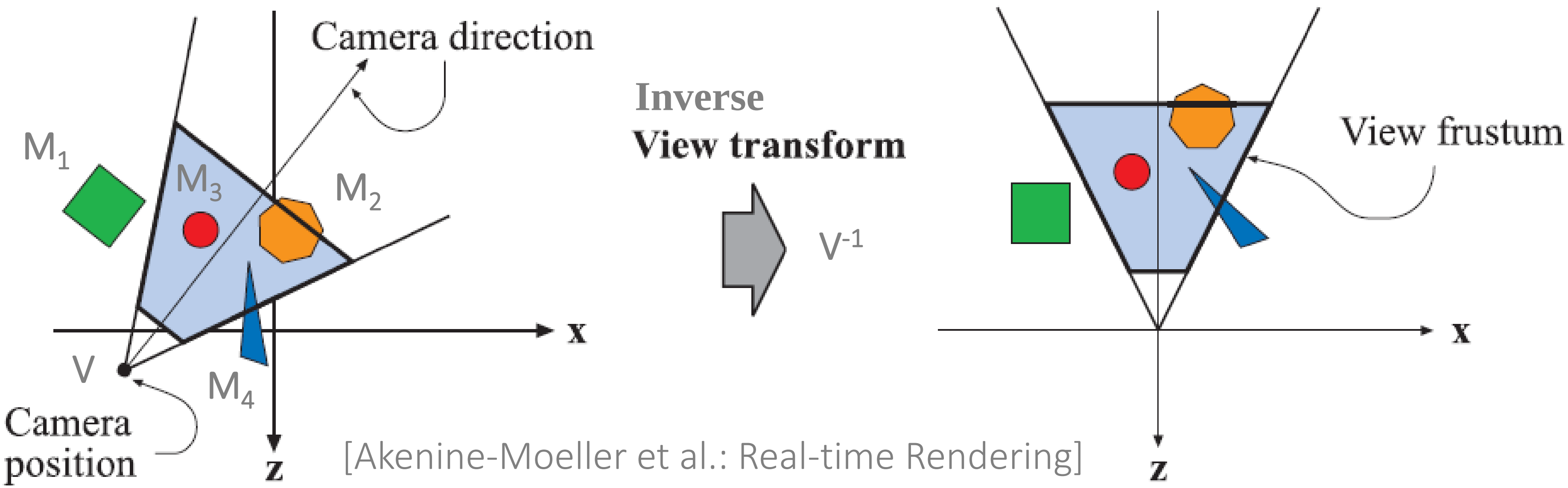
Outline

- transformations in the rendering pipeline
- motivations for the homogeneous notation
- homogeneous notation
- basic transformations in homogeneous notation
- compositing transformations
- summary

Vertex Processing

- modelview transform
- (lighting)
- projection transform
- (clipping)
- viewport transform

Modelview Transform

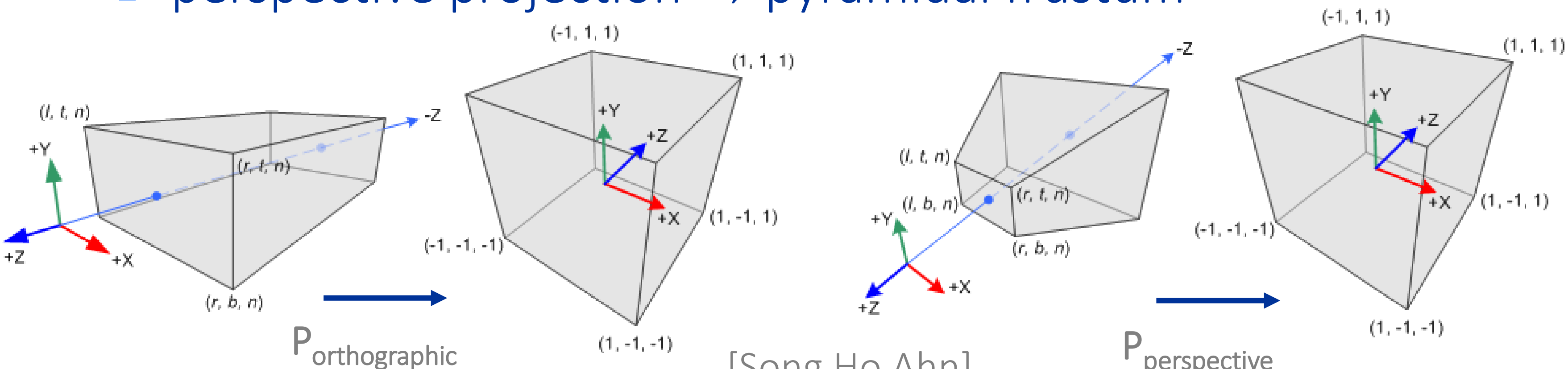


[Akenine-Moeller et al.: Real-time Rendering]

- M_1, M_2, M_3, M_4, V are matrices representing transformations
- M_1, M_2, M_3, M_4 are model transforms to place the objects in the scene
- V places and orientates the camera in space
  - V^{-1} transforms the camera to the origin looking along the negative z-axis
- model and view transforms are combined in the modelview transform
- the modelview transform $V^{-1}M_{1..4}$ is applied to the objects

Projection Transform

- **P** transforms the view volume to the canonical view volume
- the view volume depends on the camera properties
  - orthographic projection → cuboid
  - perspective projection → pyramidal frustum

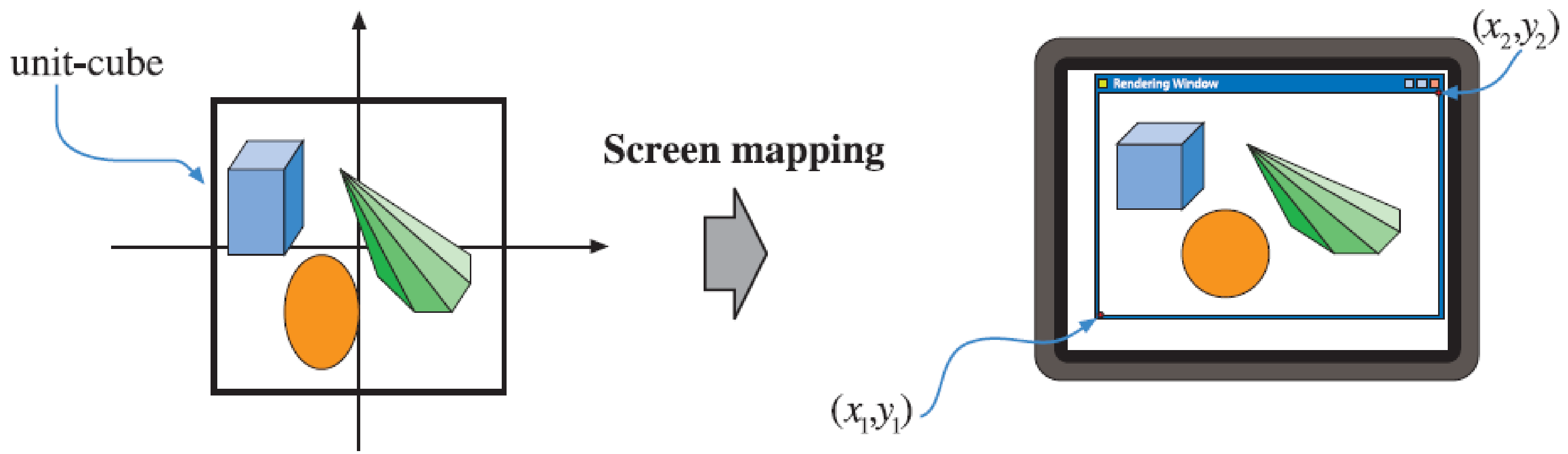


[Song Ho Ahn]

- canonical view volume is a cube from (-1,-1,-1) to (1,1,1)
- view volume is specified by near, far, left, right, bottom, top

Viewport Transform / Screen Mapping

- projected primitive coordinates (x_p, y_p, z_p) are transformed to screen coordinates (x_s, y_s)
- screen coordinates together with depth value are window coordinates (x_s, y_s, z_w)



[Akenine-Moeller et al.: Real-time Rendering]

Vertex Transforms

object space

$\Downarrow$ modelview transform

eye space / camera space

$\Downarrow$ projection transform

normalized
device coordinates

$\Downarrow$ viewport transform

window space

Vertex Transforms

local object space

$\Downarrow$ model transform

world coordinate space

$\Downarrow$ inverse view transform

eye space / camera space

$\Downarrow$ projection transform

normalized
device coordinates

$\Downarrow$ viewport transform

window space

Outline

- transformations in the rendering pipeline
- motivations for the homogeneous notation
- homogeneous notation
- basic transformations in homogeneous notation
- compositing transformations
- summary

Some Transformations

- congruent transformations (Euclidean transformations)
  - preserve shape and size
  - translation, rotation, reflection
- similarity transformations
  - preserve shape
  - translation, rotation, reflection, scale

Affine Transformations

allowing for or preserving parallel relationships , w/ scale

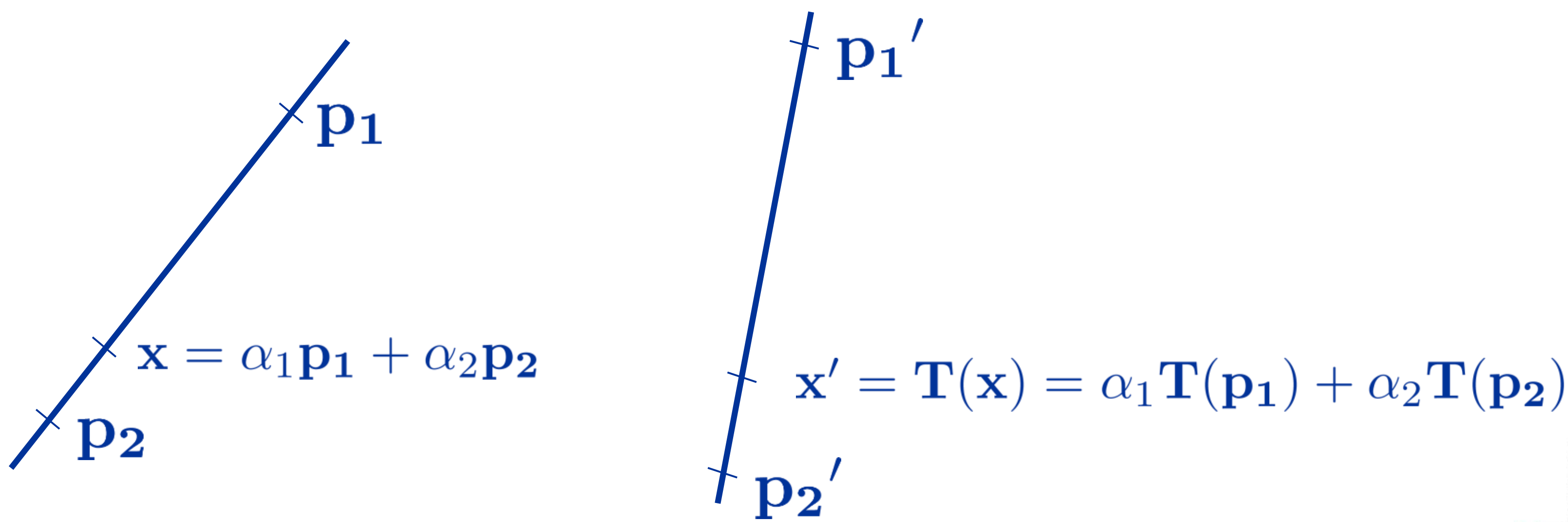
- preserve collinearity
  - points on a line are transformed to points on a line
- preserve proportions
  - ratios of distances between points are preserved
- preserve parallelism
  - parallel lines are transformed to parallel lines
- angles and lengths are not preserved

a strain in the structure of a substance produced by pressure, when its layers are laterally shifted in relation to each other

- translation, rotation, reflection, scale, shear are affine
- orthographic projection is a combination of affine transf.
- perspective projection is not affine

Affine Transformations

- affine transformations of a 3D point **p**: $\mathbf{p}' = \mathbf{T}(\mathbf{p}) = \mathbf{A}\mathbf{p} + \mathbf{t}$
- affine transformations preserve affine combinations
  $\mathbf{T}\left(\sum_i \alpha_i \cdot \mathbf{p}_i\right) = \sum_i \alpha_i \cdot \mathbf{T}(\mathbf{p}_i)$ for $\sum_i \alpha_i = 1$
- e.g., a line can be transformed
  by transforming its control points



Affine Transformations

- affine transformations of a 3D point **p**
  $$\mathbf{p}' = \mathbf{A}\mathbf{p} + \mathbf{t}$$
- the 3x3 matrix **A** represents scale and rotation
- the 3D vector **t** represents translation
- using homogeneous coordinates,
  all affine transformations are represented
  with one matrix-vector multiplication

UNI FREIBURG

Points and Vectors

- the rendering pipeline transforms vertices, normals, colors, texture coordinates
- points (e.g. vertices) specify a location in space
- vectors (e.g. normals) specify a direction
- relations between points and vectors
  - point - point = vector
  - point + vector = point
  - vector + vector = vector
  - point + point = not defined
  - $\vec{\mathbf{p}} = \mathbf{p} - \mathbf{O} \quad \mathbf{p} = \mathbf{O} + \vec{\mathbf{p}}$

Points and Vectors

- transformations can have different effects on points and vectors
- translation
  - translation of a point moves the point to a different position
  - translation of a vector does not change the vector
- using homogeneous coordinates, transformations of vectors and points can be handled in a unified way

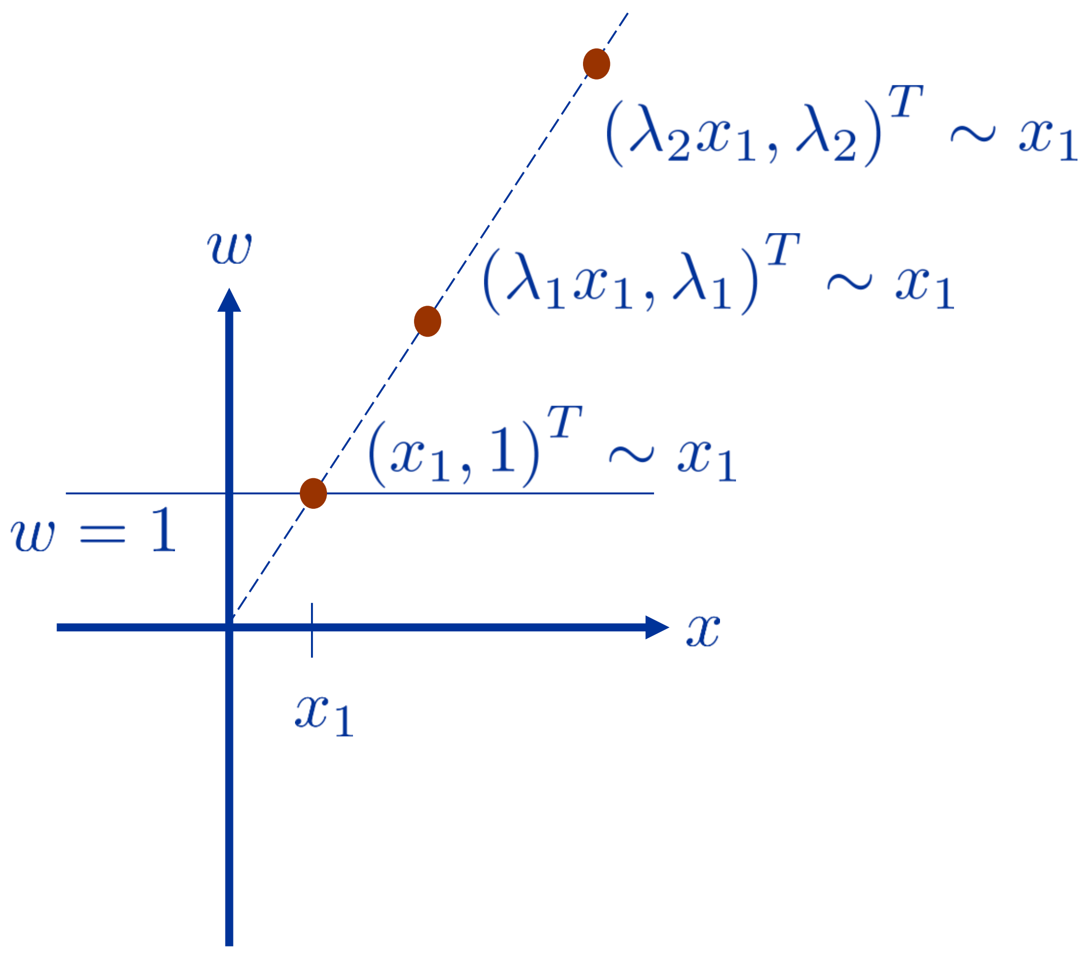
Outline

- transformations in the rendering pipeline
- motivations for the homogeneous notation
- homogeneous notation
- basic transformations in homogeneous notation
- compositing transformations
- summary

Homogeneous Coordinates of Points

- $(x, y, z, w)^T$ with $w \neq 0$ are the homogeneous coordinates of the 3D point $(\frac{x}{w}, \frac{y}{w}, \frac{z}{w})^T$
- $(\lambda x, \lambda y, \lambda z, \lambda w)^T$ represents the same point $(\frac{\lambda x}{\lambda w}, \frac{\lambda y}{\lambda w}, \frac{\lambda z}{\lambda w})^T = (\frac{x}{w}, \frac{y}{w}, \frac{z}{w})^T$ for all λ with $\lambda \neq 0$
- examples
  - (2, 3, 4, 1) ~ (2, 3, 4)
  - (2, 4, 6, 1) ~ (2, 4, 6)
  - (4, 8, 12, 2) ~ (2, 4, 6)

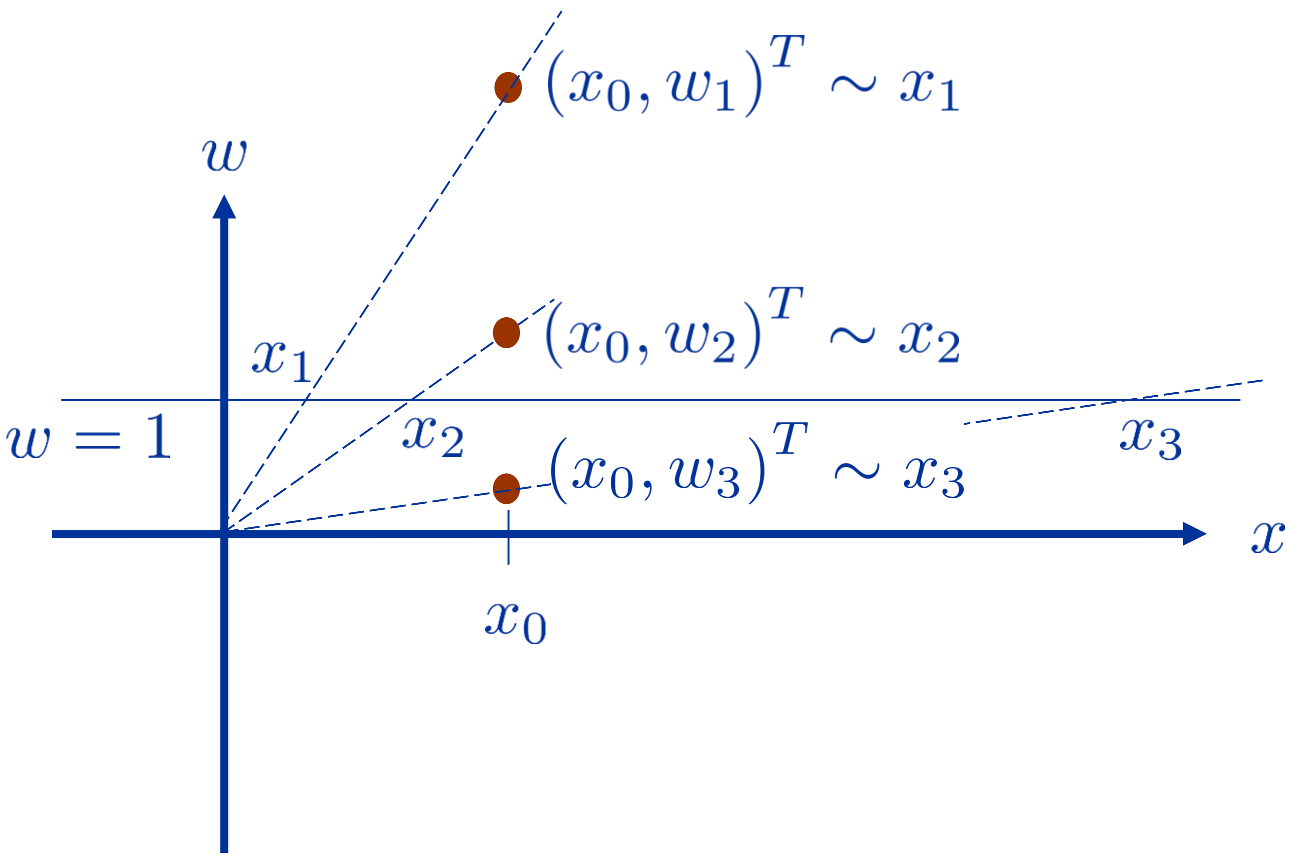
1D Illustration



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 characterized by $(x, y, z)^T$
- for *w* → 0, the point $(\frac{x}{w}, \frac{y}{w}, \frac{z}{w})^T$ moves to infinity in the direction $(x, y, z)^T$
- $(x, y, z, 0)^T$ is a point at infinity in the direction of $(x, y, z)^T$
- $(x, y, z, 0)^T$ is a vector in the direction of $(x, y, z)^T$

1D Illustration



Points and Vectors

- if points are represented in the homogeneous (normalized) form, point - vector relations can be represented
- vector + vector = vector $$\begin{pmatrix} u_x \\ u_y \\ u_z \\ 0 \end{pmatrix} + \begin{pmatrix} v_x \\ v_y \\ v_z \\ 0 \end{pmatrix} = \begin{pmatrix} u_x + v_x \\ u_y + v_y \\ u_z + v_z \\ 0 \end{pmatrix}$$
- point + vector = point $$\begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} + \begin{pmatrix} v_x \\ v_y \\ v_z \\ 0 \end{pmatrix} = \begin{pmatrix} p_x + v_x \\ p_y + v_y \\ p_z + v_z \\ 1 \end{pmatrix}$$
- point - point = vector $$\begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} - \begin{pmatrix} r_x \\ r_y \\ r_z \\ 1 \end{pmatrix} = \begin{pmatrix} p_x - r_x \\ p_y - r_y \\ p_z - r_z \\ 0 \end{pmatrix}$$

Homogeneous Representation 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

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

- m_{ij} represent rotation, scale
- t_i represent translation
- p_i represent projection
- w is analog to the fourth component for points / vectors

Homogeneous Coordinates - Summary

- $(x, y, z, w)^T$ with $w \neq 0$ are the homogeneous coordinates of the 3D point $(\frac{x}{w}, \frac{y}{w}, \frac{z}{w})^T$
- $(x, y, z, 0)^T$ 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 $(x, y, z)^T$
- $\begin{pmatrix} m_{00} & m_{01} & m_{02} & t_0 \\ m_{10} & m_{11} & m_{12} & t_1 \\ m_{20} & m_{21} & m_{22} & t_2 \\ p_0 & p_1 & p_2 & w \end{pmatrix}$ is a transformation, representing rotation, scale, translation, projection

Outline

- transformations in the rendering pipeline
- motivations for the homogeneous notation
- homogeneous notation
- basic transformations in homogeneous notation
- compositing transformations
- summary

Translation

- of a point

$$\mathbf{T}(\mathbf{t})\mathbf{p} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = \begin{pmatrix} p_x + t_x \\ p_y + t_y \\ p_z + t_z \\ 1 \end{pmatrix}$$

- of a vector

$$\mathbf{T}(\mathbf{t})\mathbf{v} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ v_z \\ 0 \end{pmatrix} = \begin{pmatrix} v_x \\ v_y \\ v_z \\ 0 \end{pmatrix}$$

- inverse ($\mathbf{T}^{-1}$ "undoes" the transform $\mathbf{T}$)

$$\mathbf{T}^{-1}(\mathbf{t}) = \mathbf{T}(-\mathbf{t})$$

Rotation

- positive (anticlockwise) rotation with angle ϕ around the *x*-, *y*-, *z*-axis

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

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

$$\mathbf{R_z}(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 & 0 \\ \sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Inverse Rotation

- $$\mathbf{R_x}(-\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos -\phi & -\sin -\phi & 0 \\ 0 & \sin -\phi & \cos -\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi & \sin \phi & 0 \\ 0 & -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \mathbf{R_x}^T(\phi)$$
- $\mathbf{R_x}^{-1} = \mathbf{R_x}^T \qquad \mathbf{R_y}^{-1} = \mathbf{R_y}^T \qquad \mathbf{R_z}^{-1} = \mathbf{R_z}^T$
- the inverse of a rotation matrix corresponds to its transpose

UNI FREIBURG

Mirroring / Reflection

- mirroring with respect to x =0, y =0, z =0 plane
- changes the sign of the x -, y -, z - component

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

- the inverse of a reflection corresponds to its transpose

$$\mathbf{P_x}^{-1} = \mathbf{P_x}^T \qquad \mathbf{P_y}^{-1} = \mathbf{P_y}^T \qquad \mathbf{P_z}^{-1} = \mathbf{P_z}^T$$

Orthogonal Matrices

- rotation and reflection matrices are orthogonal
  $\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I}$
  $\mathbf{R}^{-1} = \mathbf{R}^T$
- $\mathbf{R}_1, \mathbf{R}_2$ are orthogonal $\Rightarrow$ $\mathbf{R}_1\mathbf{R}_2$ is orthogonal
- rotation: $\det \mathbf{R} = 1$ reflection: $\det \mathbf{R} = -1$
- length of a vector does not change $\|\mathbf{R}\mathbf{v}\| = \|\mathbf{v}\|$
- angles are preserved $\langle \mathbf{R}\mathbf{u}, \mathbf{R}\mathbf{v} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle$

Scale

- scaling *x* -, *y* - , *z* - components of a point or vector

$$\mathbf{S}(s_x, s_y, s_z)\mathbf{p} = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = \begin{pmatrix} s_x p_x \\ s_y p_y \\ s_z p_z \\ 1 \end{pmatrix}$$

- inverse $\mathbf{S}^{-1}(s_x, s_y, s_z) = \mathbf{S}(\frac{1}{s_x}, \frac{1}{s_y}, \frac{1}{s_z})$
- uniform scaling: $s_x = s_y = s_z = s$

$$\mathbf{S}(s, s, s) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{s} \end{pmatrix}$$

UNI FREIBURG

Shear

- one component is offset
  with respect to another component
- six basic shear modes in 3D
- e.g., shear of *x* with respect to *z*

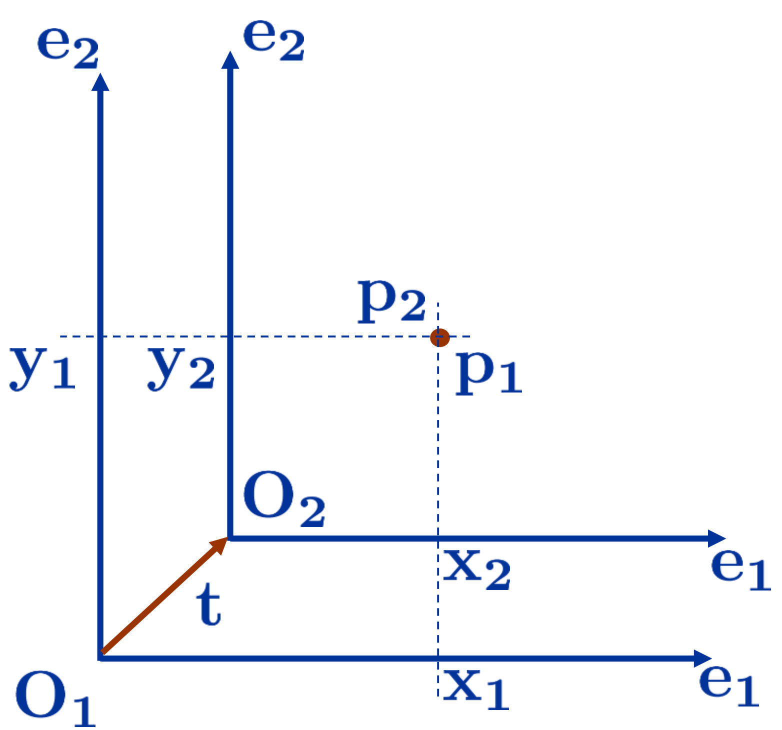
$$\mathbf{H_{xz}}(s)\mathbf{p} = \begin{pmatrix} 1 & 0 & s & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \\ 1 \end{pmatrix} = \begin{pmatrix} p_x + sp_z \\ p_y \\ p_z \\ 1 \end{pmatrix}$$

- inverse
  $\mathbf{H_{xz}}^{-1}(s) = \mathbf{H_{xz}}(-s)$

Basis Transform - Translation

- two coordinate systems

$$\mathbf{C_1} = (\mathbf{O_1}, \{\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}\}) \qquad \mathbf{C_2} = (\mathbf{O_2}, \{\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}\})$$



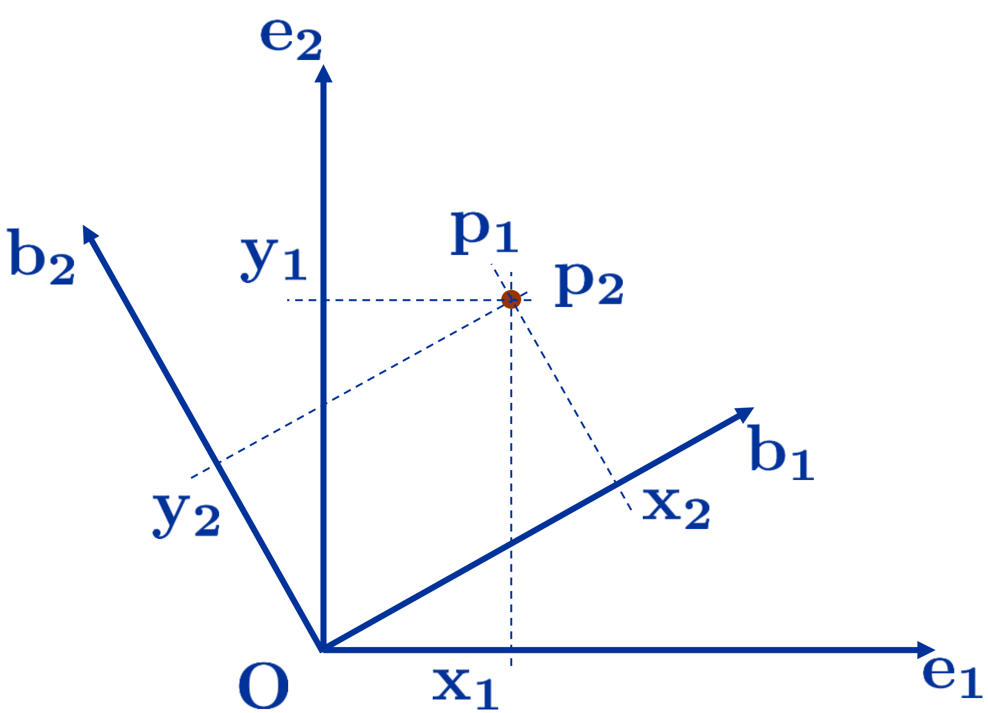
Basis Transform - Translation

- the coordinates of $\mathbf{p_1}$ with respect to $\mathbf{C_2}$ are given by $\mathbf{p_2} = \mathbf{p_1} - \mathbf{t}$ $\quad \mathbf{p_2} = \mathbf{T}(-\mathbf{t})\mathbf{p_1}$
- the coordinates of a point in the transformed basis correspond to the coordinates of point in the untransformed basis transformed by the inverse basis transform
  - translating the origin by **t** corresponds to translating the object by -**t**
  - also: rotating the basis vectors by an angle corresponds to rotating the object by the same negative angle

Basis Transform - Rotation

- two coordinate systems

$$\mathbf{C_1} = (\mathbf{O}, \{\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}\}) \qquad \mathbf{C_2} = (\mathbf{O}, \{\mathbf{b_1}, \mathbf{b_2}, \mathbf{b_3}\})$$



Basis Transform - Rotation

- the coordinates of $\mathbf{p_1}$ with respect to $\mathbf{C_2}$ are given by

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

- $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$ are the basis vectors of $\mathbf{C}_2$ with respect to $\mathbf{C}_1$
- $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$ are orthonormal, therefore the basis transform is 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 placed with respect to a (global) coordinate system $\mathbf{C_1} = (\mathbf{O_1}, \{\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}\})$
- the camera is also positioned at $\mathbf{O_2}$ and oriented at $\{\mathbf{b_1}, \mathbf{b_2}, \mathbf{b_3}\}$ given by viewing direction and up-vector
- after the view transform, all objects are represented in the eye or camera coordinate system $\mathbf{C_2} = (\mathbf{O_2}, \{\mathbf{b_1}, \mathbf{b_2}, \mathbf{b_3}\})$
- placing and orienting the camera corresponds to the application of the inverse transform to the objects
- rotating the camera by **R** and translating it by **T**, corresponds to translating the objects by $\mathbf{T}^{-1}$ and rotating them by $\mathbf{R}^{-1}$

Planes and Normals

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

$$n_x p_x + n_y p_y + n_z p_z + (-n_x r_x - n_y r_y - n_z r_z) = 0$$

$$n_x p_x + n_y p_y + n_z p_z + d = 0$$

$$(n_x \; n_y \; n_z \; d)(p_x \; p_y \; p_z \; 1)^T = 0$$

$$(n_x \; n_y \; n_z \; d)\mathbf{A}^{-\mathbf{1}}\mathbf{A}(p_x \; p_y \; p_z \; 1)^T = 0$$

- the transformed points $\mathbf{A}(p_x \; p_y \; p_z \; 1)^T$ are on the plane represented by $(n_x \; n_y \; n_z \; d)\mathbf{A}^{-\mathbf{1}} = ((\mathbf{A}^{-\mathbf{1}})^T(n_x \; n_y \; n_z \; d)^T)^T$
- if a surface is transformed by **A**, its homogeneous notation (including the surface normal) is transformed by $(\mathbf{A}^{-1})^T$

Outline

- transformations in the rendering pipeline
- motivations for the homogeneous notation
- homogeneous notation
- basic transformations in homogeneous notation
- compositing transformations
- summary

Compositing Transformations

- composition is achieved by matrix multiplication
  - a translation **T** applied to **p**, followed by a rotation **R**
    $$\mathbf{R}(\mathbf{T}\mathbf{p}) = (\mathbf{R}\mathbf{T})\mathbf{p}$$
  - a rotation **R** applied to **p**, followed by a translation **T**
    $$\mathbf{T}(\mathbf{R}\mathbf{p}) = (\mathbf{T}\mathbf{R})\mathbf{p}$$
  - note that generally $\mathbf{TR} \neq \mathbf{RT}$
  - the order of composed transformations matters

Examples

- rotation around a line through **t** parallel to the *x*-, *y*-, *z*- axis

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

- scale with respect to an arbitrary axis

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

- e.g., $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$ represent an orthonormal basis, then scaling along these vectors can be done by

  $$\begin{pmatrix} \mathbf{b_1} & \mathbf{b_2} & \mathbf{b_3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{S}(s_x, s_y, s_z) \begin{pmatrix} \mathbf{b_1} & \mathbf{b_2} & \mathbf{b_3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}^T$$

Rigid-Body Transform

- $\begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \mathbf{p} = \mathbf{T}(\mathbf{t})\mathbf{R}\mathbf{p}$

  with **R** being a rotation and **t** being a translation is a combined transformation
- inverse

  $(\mathbf{T}(\mathbf{t})\mathbf{R})^{-1} = \mathbf{R}^{-1}\mathbf{T}(\mathbf{t})^{-1} = \mathbf{R}^T\mathbf{T}(-\mathbf{t})$
- in Euclidean coordinates $\mathbf{p}' = \mathbf{R}\mathbf{p} + \mathbf{t}$
- the inverse transform $\mathbf{p} = \mathbf{R}^{-1}(\mathbf{p}' - \mathbf{t}) = \mathbf{R}^{-1}\mathbf{p}' - \mathbf{R}^{-1}\mathbf{t}$
- therefore $\begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \\ 0 & 1 \end{pmatrix}$

Outline

- transformations in the rendering pipeline
- motivations for the homogeneous notation
- homogeneous notation
- basic transformations in homogeneous notation
- compositing transformations
- summary

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

UNI FREIBURG