**Objectives**

- Introduce the elements of geometry
  - Scalars
  - Vectors
  - Points
- Develop mathematical operations among them in a coordinate-free manner
- Define basic primitives
  - Line segments
  - Polygons

**Basic Elements**

- Geometry is the study of the relationships among objects in an n-dimensional space
  - In computer graphics, we are interested in objects that exist in three dimensions
- Want a minimum set of primitives from which we can build more sophisticated objects
- We will need three basic elements
  - Scalars
  - Vectors
  - Points

**Coordinate-Free Geometry**

- When we learned simple geometry, most of us started with a Cartesian approach
  - Points were at locations in space \( p = (x, y, z) \)
  - We derived results by algebraic manipulations involving these coordinates
- This approach was nonphysical
  - Physically, points exist regardless of the location of an arbitrary coordinate system
  - Most geometric results are independent of the coordinate system
- Example Euclidean geometry: two triangles are identical if two corresponding sides and the angle between them are identical

**Scalars**

- Need three basic elements in geometry
  - Scalars, Vectors, Points
- Scalars can be defined as members of sets which can be combined by two operations (addition and multiplication) obeying some fundamental axioms (associativity, commutativity, inverses)
- Examples include the real and complex number systems under the ordinary rules with which we are familiar
- Scalars alone have no geometric properties

**Vectors**

- Physical definition: a vector is a quantity with two attributes
  - Direction
  - Magnitude
- Examples include
  - Force
  - Velocity
  - Directed line segments
  - Most important example for graphics
  - Can map to other types

**Geometric Objects**

- Define basic geometric objects
  - Line segments
  - Polygons

**Mathematical Operations Among Them**

- Develop mathematical operations among them in a coordinate-free manner

**Examples**

- Example Euclidean geometry: two triangles are identical if two corresponding sides and the angle between them are identical
Vector Operations

- Every vector has an inverse
  - Same magnitude but points in opposite direction
- Every vector can be multiplied by a scalar
- There is a zero vector
  - Zero magnitude, undefined orientation
- The sum of any two vectors is a vector
  - Use head-to-tail axiom

Vectors, \( V (x, y, z) \)

- Length and direction. Absolute position not important
- Usually written as \( \vec{a} \) or in bold. Magnitude written as \( |\vec{V}| \)
- Use to store offsets, displacements, locations
  - But strictly speaking, positions are not vectors and cannot be added: a location implicitly involves an origin, while an offset does not.

Vectors OP

- A unit vector: a vector can be normalised such that it retains its direction, but is scaled to have unit length:
  \[ \frac{\vec{V}}{|\vec{V}|} \]

Dot Product

\[ u \cdot v = x_u \cdot x_v + y_u \cdot y_v + z_u \cdot z_v \]

\[ u \cdot v = |u| |v| \cos \theta \]

\[ \therefore \cos \theta = \frac{u \cdot v}{|u| |v|} \]

- This is purely a scalar number not a vector.
- What happens when the vectors are unit
- What does it mean if dot product == 0 or == 1?
Dot product: some applications in CG

- Find angle between two vectors (e.g. cosine of angle between light source and surface for shading)
- Finding projection of one vector on another (e.g. coordinates of point in arbitrary coordinate system)
- Advantage: can be computed easily in cartesian components

\[ \mathbf{a} \cdot \mathbf{b} = (x_a, y_a) \cdot (x_b, y_b) = ? \]

\[ \mathbf{a} \cdot \mathbf{b} = (x_a, y_a) \cdot (x_b, y_b) = x_a x_b + y_a y_b \]

Projections (of \( \mathbf{b} \) on \( \mathbf{a} \))

\[ \| \mathbf{b} \| \rightarrow \mathbf{a} = ? \]

\[ \| \mathbf{b} \| \cos \phi = \frac{\mathbf{a} \cdot \mathbf{b}}{\| \mathbf{a} \|} \]

\[ \mathbf{b} \rightarrow \mathbf{a} = \frac{\mathbf{a} \cdot \mathbf{b}}{\| \mathbf{a} \|^2} \mathbf{a} \]

Cross Product

- The result is not a scalar but a vector which is normal to the plane of the other 2
- Can be computed using the determinant of:

\[ \mathbf{u} \times \mathbf{v} = i(y_v z_u - z_v y_u) - j(x_v z_u - z_v x_u) + k(x_v y_u - y_v x_u) \]

- Size is \( \| \mathbf{u} \| \| \mathbf{v} \| \sin \theta \)
- Cross product of vector with itself is null

Cross (vector) product

- Cross product orthogonal to two initial vectors
- Direction determined by right-hand rule
- Useful in constructing coordinate systems (later)

\[ \mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a} \]

\[ \| \mathbf{a} \times \mathbf{b} \| = \| \mathbf{a} \| \| \mathbf{b} \| \sin \phi \]

\[ xx y = +z \]
\[ yx x = -z \]
\[ yz z = +x \]
\[ zx y = -x \]
\[ zx x = +y \]
\[ xx z = -y \]

Cross product: Properties

- \( \mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a} \)
- \( \mathbf{a} \times \mathbf{a} = 0 \)
- \( \mathbf{a} \times (\mathbf{b} + \mathbf{c}) = \mathbf{a} \times \mathbf{b} + \mathbf{a} \times \mathbf{c} \)
- \( \mathbf{a} \times (k \mathbf{b}) = k (\mathbf{a} \times \mathbf{b}) \)
Cross product: Cartesian formula?

\[
a \times b = \begin{vmatrix}
    x & y & z \\
    x_a & y_a & z_a \\
    x_b & y_b & z_b
\end{vmatrix} = (y_a z_b - y_b z_a) \\
\begin{vmatrix}
    y_a & z_a \\
    y_b & z_b
\end{vmatrix} = (x_a z_b - x_b z_a) \\
\begin{vmatrix}
    x_a & y_a \\
    x_b & y_b
\end{vmatrix} = (y_a x_b - y_b x_a)
\]

\[
a \times b = A^t b = \begin{pmatrix}
    0 & -z_a & y_a \\
    z_a & 0 & -x_a \\
    -y_a & x_a & 0
\end{pmatrix}
\]

dual matrix of vector a

Linear Vector Spaces

- Mathematical system for manipulating vectors
- Operations
  - Scalar-vector multiplication: \( u \cdot v \)
  - Vector-vector addition: \( W = u + v \)
- Expressions such as \( v = u + 2w + 3r \)
  Make sense in a vector space

Vectors Lack Position

- These vectors are identical
  - Same length and magnitude
- Vectors spaces insufficient for geometry
  - Need points

Points

- Location in space
- Operations allowed between points and vectors
  - Point-point subtraction yields a vector
  - Equivalent to point-vector addition

Affine Spaces

- Point + a vector space
- Operations
  - Vector-vector addition
  - Scalar-vector multiplication
  - Point-vector addition
  - Scalar-scalar operations
- For any point define
  - \( 1 \cdot P = P \)
  - \( 0 \cdot P = \emptyset \) (zero vector)
Lines

• Consider all points of the form
  \[ P(\alpha) = P_0 + \alpha d \]
  • Set of all points that pass through \( P_0 \) in the direction of the vector \( d \)

Parametric Form

• This form is known as the parametric form of the line
  • More robust and general than other forms
  • Extends to curves and surfaces
  Two-dimensional forms
  • Explicit: \( y = mx + b \)
  • Implicit: \( ax + by + c = 0 \)
  • Parametric:
    \[
    x(\alpha) = x_0 + (1-\alpha)x_1 \\
    y(\alpha) = y_0 + (1-\alpha)y_1
    \]

Rays and Line Segments

• If \( \alpha \geq 0 \), then \( P(\alpha) \) is the ray leaving \( P_0 \) in the direction \( d \)
  If we use two points to define \( v \), then
  \[ P(\alpha) = Q + \alpha (R - Q) = Q + \alpha v \]
  For \( 0 \leq \alpha \leq 1 \) we get all the points on the line segment joining \( R \) and \( Q \)

Convexity

• An object is convex if for any two points in the object all points on the line segment between these points are also in the object

Affine Sums

• Consider the “sum”\n  \[ P = \alpha_1 P_1 + \alpha_2 P_2 + \ldots + \alpha_n P_n \]
  Can show by induction that this sum makes sense iff
  \( \alpha_1 + \alpha_2 + \ldots + \alpha_n = 1 \)
  in which case we have the affine sum of the points \( P_1, P_2, \ldots, P_n \)
  • If, in addition, \( \alpha_i \geq 0 \), we have the convex hull of \( P_1, P_2, \ldots, P_n \)
Convex Hull

- Smallest convex object containing \( P_1, P_2, \ldots, P_n \)
- Formed by “shrink wrapping” points

Curves and Surfaces

- Curves are one parameter entities of the form \( P(\alpha) \) where the function is nonlinear
- Surfaces are formed from two-parameter functions \( P(\alpha, \beta) \)
  - Linear functions give planes and polygons

Planes

- A plane can be defined by a point and two vectors or by three points

Barycentric Coordinates

Triangle is convex so any point inside can be represented as an affine sum
\[
P(\alpha_1, \alpha_2, \alpha_3) = \alpha_1 P_1 + \alpha_2 P_2 + \alpha_3 P_3
\]
where
\[
\alpha_1 + \alpha_2 + \alpha_3 = 1
\]
\[
\alpha_i > 0
\]
The representation is called the \textit{barycentric coordinate} representation of \( P \)

Normals

- Every plane has a vector \( n \) normal (perpendicular, orthogonal) to it
- From point-two vector form \( P(\alpha, \beta) = R + \alpha u + \beta v \), we know we can use the cross product to find \( n = u \times v \) and the equivalent form
\[
(P(\alpha)-P) \cdot n = 0
\]
**Objectives**

- Introduce concepts such as dimension and basis
- Introduce coordinate systems for representing vectors spaces and frames for representing affine spaces
- Discuss change of frames and bases
- Introduce homogeneous coordinates

---

**Vectors and Matrices**

- Matrix is an array of numbers with dimensions $M$ (rows) by $N$ (columns)
  - $3$ by $6$ matrix
  - Element $2,3$ is $(3)$

- Vector can be considered a $1 \times M$ matrix
  
  $$v = (x \ y \ z)$$

---

**Types of Matrix**

- **Identity matrices** - $I$
  
  $$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- **Symmetric**
  
  $$\begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

- **Diagonal**
  
  $$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

---

**Operation on Matrices**

- **Addition**
  
  - Done elementwise
  
  $$\begin{pmatrix} c & \alpha \\ \beta & \gamma \end{pmatrix} + \begin{pmatrix} \nu & \zeta \\ \lambda & \delta \end{pmatrix} = \begin{pmatrix} c+\nu & \alpha+\zeta \\ \beta+\lambda & \gamma+\delta \end{pmatrix}$$

- **Transpose**
  
  - “Flip” ($M$ by $N$ becomes $N$ by $M$)
  
  $$\begin{pmatrix} 1 & 4 & 9 \\ 5 & 2 & 8 \\ 6 & 7 & 3 \end{pmatrix}^T = \begin{pmatrix} 1 & 5 & 6 \\ 4 & 2 & 7 \\ 9 & 8 & 3 \end{pmatrix}$$

---

**Operations on Matrices**

- **Multiplication**
  
  - Only possible to multiply if dimensions $x_1$ by $y_1$ and $x_2$ by $y_2$, if $y_1 = x_2$
  
  - Resulting matrix is $x_1$ by $y_2$ (and only)

  - e.g. Matrix $A$ is $2 \times 3$ and Matrix $B$ is $3 \times 4$
  
  - Resulting matrix is $2 \times 4$

  - Just because $A \times B$ is possible doesn't mean $B \times A$ is possible!
A is n by k, B is k by m

\[ C = A \times B \text{ defined by} \]

\[ c_{ij} = \sum_{k} a_{ik} b_{kj} \]

\[ \text{BxA not necessarily equal to AxB} \]

**Matrix Multiplication Order**

**Example Multiplications**

\[ \begin{pmatrix} 2 & 3 & 1 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 1 & -1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} ____ & ____ \\ ____ & ____ \end{pmatrix} \]

\[ \begin{pmatrix} 2 & -2 & 3 \\ -3 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} ____ & ____ & ____ \\ ____ & ____ & ____ \end{pmatrix} \]

**Computation: A x B = C**

\[ A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \text{ [2 x 2]} \]

\[ B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} \text{ [2 x 3]} \]

\[ C = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} & a_{11}b_{13} + a_{12}b_{23} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} & a_{21}b_{13} + a_{22}b_{23} \end{bmatrix} \text{ [2 x 3]} \]

\[ \text{A and B can be multiplied} \]

**Computation: A x B = C**

\[ A = \begin{bmatrix} 2 & 3 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \text{ and } B = \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 2 \end{bmatrix} \text{ [2 x 3]} \]

\[ C = \begin{bmatrix} 2^1+3^1=5 & 2^1+3^0=2 & 2^1+3^2=8 \\ 1^1+1^1=2 & 1^1+1^0=1 & 1^1+1^2=3 \end{bmatrix} \text{ [2 x 3]} \]

**What is a matrix**

- Array of numbers (m x n = m rows, n columns)

\[ \begin{bmatrix} 1 & 3 \\ 5 & 2 \\ 0 & 4 \end{bmatrix} \]

- Addition, multiplication by a scalar simple: element by element
**Matrix-matrix multiplication**

- Number of columns in first must = rows in second
- Element (i,j) in product is dot product of row i of first matrix and column j of second matrix

\[
\begin{pmatrix}
1 & 3 \\
5 & 2 \\
0 & 4
\end{pmatrix}
\begin{pmatrix}
3 & 6 & 9 & 4 \\
2 & 7 & 8 & 3
\end{pmatrix}
\]

**Matrix-matrix multiplication**

- Number of columns in first must = rows in second
- Element (i,j) in product is dot product of row i of first matrix and column j of second matrix

\[
\begin{pmatrix}
1 & 3 \\
5 & 2 \\
0 & 4
\end{pmatrix}
\begin{pmatrix}
3 & 6 & 9 & 4 \\
2 & 7 & 8 & 3
\end{pmatrix} =
\begin{pmatrix}
9 & 27 & 33 & 13 \\
17 & 44 & 61 & 24
\end{pmatrix}
\]

**Matrix-vector Multiplication**

- Key for transforming points (next lecture)
- Treat vector as a column matrix (m×1)

- E.g. 2D reflection about y-axis (from textbook)

\[
\begin{pmatrix}
-1 & 0 \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
x \\
y
\end{pmatrix} =
\begin{pmatrix}
-x \\
y
\end{pmatrix}
\]
Inverse

If \( A \times B = I \) and \( B \times A = I \) then
\( A = B^{-1} \) and \( B = A^{-1} \)

Matrix Inversion

\[
B^{-1}B = BB^{-1} = I
\]

Like a reciprocal in scalar math
Like the number one in scalar math

Transpose of a Matrix (or vector?)

\[
\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}^T = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}
\]

\[(AB)^T = B^TA^T\]

Identity Matrix and Inverses

\[
I_{3 \times 3} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}
\]

\[
AA^{-1} = A^{-1}A = I
\]

\[(AB)^{-1} = B^{-1}A^{-1}\]

Vector multiplication in Matrix form

- **Dot product?**
  \( a \cdot b = a^Tb \)
  \[
  \begin{pmatrix} x_a & y_a & z_a \end{pmatrix} \begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix} = (x_a x_b + y_a y_b + z_a z_b)
  \]

- **Cross product?**
  \( a \times b = A'b = \begin{pmatrix} 0 & -z_a & y_a \\ z_a & 0 & -x_a \\ -y_a & x_a & 0 \end{pmatrix} \begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix} \)
  Dual matrix of vector \( a \)

Linear Independence

- A set of vectors \( v_1, v_2, ..., v_n \) is **linearly independent** if
  \( \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n = 0 \) if \( \alpha_1 = \alpha_2 = \ldots = 0 \)
- If a set of vectors is linearly independent, we cannot represent one in terms of the others
- If a set of vectors is linearly dependent, at least one can be written in terms of the others
• AB ≠ BA
• ABC ≠ CBA
• So order matters!

Dimension

• In a vector space, the maximum number of linearly independent vectors is fixed and is called the dimension of the space
• In an n-dimensional space, any set of n linearly independent vectors form a basis for the space
• Given a basis \( v_1, v_2, \ldots, v_n \), any vector \( v \) can be written as
  \[
  v = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n
  \]
  where the \( \alpha_i \) are unique

Example

• \( v = 2v_1 + 3v_2 - 4v_3 \)
• \( a = [2 \ 3 \ -4]^T \)
• Note that this representation is with respect to a particular basis
• For example, in OpenGL we start by representing vectors using the object basis but later the system needs a representation in terms of the camera or eye basis

Coordinate Systems

• Consider a basis \( v_1, v_2, \ldots, v_n \)
• A vector is written \( v = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n \)
• The list of scalars \( \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \) is the representation of \( v \) with respect to the given basis
• We can write the representation as a row or column array of scalars
  \[
  a = [\alpha_1 \ \alpha_2 \ \ldots \ \alpha_n]^T
  \]

• Both are because vectors have no fixed location

Representation

• Until now we have been able to work with geometric entities without using any frame of reference, such as a coordinate system
• Need a frame of reference to relate points and objects to our physical world.
  • For example, where is a point? Can’t answer without a reference system
  • World coordinates
  • Camera coordinates

Coordinate Systems

• Which is correct?

- Both are because vectors have no fixed location
Frames

- A coordinate system is insufficient to represent points
- If we work in an affine space we can add a single point, the origin, to the basis vectors to form a frame

Confusing Points and Vectors

Consider the point and the vector

\[ P = \beta_0 v_1 + \beta_1 v_2 + \ldots + \beta_n v_n \]
\[ v = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n \]

They appear to have the similar representations

\[ p = [\beta_1 \beta_2 \beta_3] \quad v = [\alpha_1 \alpha_2 \alpha_3] \]

which confuses the point with the vector.

A vector has no position

Vector can be placed anywhere

point: fixed

Homogeneous Coordinates

The homogeneous coordinates form a three-dimensional point \([x \ y \ z]\) is given as

\[ p = [x' y' z' 1] \]

We return to a three dimensional point (for \(w=0\)) by

\[ x \leftarrow x'/w \]
\[ y \leftarrow y'/w \]
\[ z \leftarrow z'/w \]

If \(w=0\), the representation is that of a vector

Note that homogeneous coordinates replaces points in three dimensions by lines through the origin in four dimensions

For \(w=1\), the representation of a point is \([x \ y \ z \ 1]\)
Homogeneous Coordinates and Computer Graphics

- Homogeneous coordinates are key to all computer graphics systems
- All standard transformations (rotation, translation, scaling) can be implemented with matrix multiplications using 4 x 4 matrices
- Hardware pipeline works with 4 dimensional representations
- For orthographic viewing, we can maintain \( w=0 \) for vectors and \( w=1 \) for points
- For perspective we need a perspective division

Change of Coordinate Systems

- Consider two representations of a the same vector with respect to two different bases. The representations are

\[
a = [\alpha_1 \alpha_2 \alpha_3]
\]
\[
b = [\beta_1 \beta_2 \beta_3]
\]

where

\[
v = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 = [\alpha_1 \alpha_2 \alpha_3] [v_1 v_2 v_3]^T
\]
\[
= \beta_1 u_1 + \beta_2 u_2 + \beta_3 u_3 = [\beta_1 \beta_2 \beta_3] [u_1 u_2 u_3]^T
\]

Matrix Form

The coefficients define a 3 x 3 matrix

\[
M = \begin{bmatrix}
\gamma_{11} & \gamma_{12} & \gamma_{13} \\
\gamma_{21} & \gamma_{22} & \gamma_{23} \\
\gamma_{31} & \gamma_{32} & \gamma_{33}
\end{bmatrix}
\]

and the bases can be related by

\[
a = M^T b
\]

see text for numerical examples

Representing One Frame in Terms of the Other

Extending what we did with change of bases defining a 4 x 4 matrix

\[
M = \begin{bmatrix}
\gamma_{11} & \gamma_{12} & \gamma_{13} & 0 \\
\gamma_{21} & \gamma_{22} & \gamma_{23} & 0 \\
\gamma_{31} & \gamma_{32} & \gamma_{33} & 0 \\
\gamma_{41} & \gamma_{42} & \gamma_{43} & 1
\end{bmatrix}
\]

- We can apply a similar process in homogeneous coordinates to the representations of both points and vectors
- Consider two frames:
  \((P_0, v_1, v_2, v_3)\)
  \((Q_0, u_1, u_2, u_3)\)
- Any point or vector can be represented in either frame
- We can represent \(Q_0, u_1, u_2, u_3\) in terms of \(P_0, v_1, v_2, v_3\)

Representing second basis in terms of first

Each of the basis vectors, \(u_1, u_2, u_3\), are vectors that can be represented in terms of the first basis

\[
\begin{align*}
u_1 &= \gamma_{11}v_1 + \gamma_{12}v_2 + \gamma_{13}v_3 \\
u_2 &= \gamma_{21}v_1 + \gamma_{22}v_2 + \gamma_{23}v_3 \\
u_3 &= \gamma_{31}v_1 + \gamma_{32}v_2 + \gamma_{33}v_3
\end{align*}
\]
Working with Representations

Within the two frames any point or vector has a representation of the same form

\[ a = [\alpha_1, \alpha_2, \alpha_3, \alpha_4] \] in the first frame
\[ b = [\beta_1, \beta_2, \beta_3, \beta_4] \] in the second frame

where \( \alpha_4 = \beta_4 = 1 \) for points and \( \alpha_4 = \beta_4 = 0 \) for vectors and

\[ a = M^T b \]

The matrix \( M \) is 4 x 4 and specifies an affine transformation in homogeneous coordinates.

Affine Transformations

- Every linear transformation is equivalent to a change in frames.
- Every affine transformation preserves lines.
- However, an affine transformation has only 12 degrees of freedom because 4 of the elements in the matrix are fixed and are a subset of all possible 4 x 4 linear transformations.

The World and Camera Frames

- When we work with representations, we work with n-tuples or arrays of scalars.
- Changes in frame are then defined by 4 x 4 matrices.
- In OpenGL, the base frame that we start with is the world frame.
- Eventually we represent entities in the camera frame by changing the world representation using the model-view matrix.
- Initially these frames are the same \( (M=I) \).

Moving the Camera

If objects are on both sides of \( z=0 \), we must move camera frame.

\[ M = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & -d \\
0 & 0 & 0 & 1
\end{bmatrix} \]