and new slides below.

Refer to slide copies handout.

Finishing derivative of the perspective projection matrix.

CSI 422/502: Lecture 27
2. Given a basis \( \{ \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \} \) for a linear combination \( S \in \mathbf{v} \), every vector in \( \mathbf{v} \) \( \{ \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \} \) will get the same \( \mathbf{v}_i \) no matter how you picked it.

All different bases for the same subspace have the same number of elements.

Basic facts of linear algebra:

If you cannot stop and output anything, then call the new vector \( \mathbf{v}_i \) and set \( i = i + 1 \) and continue.

If you can, pick any other non-zero vector as long as the new vector together with the ones already picked form a linearly independent set. Set \( i = 1 \) and \( \{ \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \} \).

(1) Pick a non-zero vector and call it \( \mathbf{v}_1 \) and set \( i = 1 \).

Greedy algorithm: Given a (non-zero) linear (vector) space \( S \), like all 4-tuples \( (x, y, z, w) \) in \( \mathbf{v} \), a set of vectors is called linearly independent if there are no \( \mathbf{v}_i \) linearly dependent on \( \mathbf{v}_i \) and \( \mathbf{v}_j \) for some \( i, j \) such that \( \mathbf{v}_i = k \mathbf{v}_j \) for some scalar \( k \).

We say there is a linear dependency of \( \mathbf{v} \) on \( \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n \) if

\[ \cdots + \sum_{j=1}^{n} \mathbf{v}_j \mathbf{v}_j = \mathbf{0} \]

If non-zero vector \( \mathbf{v} \) can be written as a linear combination of others from a set of vectors.

University at Albany Computer Science Dept.
We will use this "fundamental theorem" to figure out the projection matrix from a vector \( \mathbf{v} \) in the subspace.

For any arbitrary vector \( \mathbf{v} \) and \( \mathbb{L} \), if you know \( \mathbb{L} \mathbf{v} \), then

\[
\cdots + \mathbb{L} \mathbf{v} = \mathbf{v}
\]

Remember, \( \mathbb{L} \) is a linear transformation for bases unique. (and the coefficients are unique.)

Of basis elements:

\[
\cdots + \mathbb{L} \mathbf{v} + \mathbb{L} \mathbf{n} = \mathbf{v}
\]
and is called a **Bra** (= Bra + Ket) and is called a **Ket** (= Bra + Ket)

equation's first hand side is denoted:

Then the "dot product" the substitution of point coordinates into the plane

\[
|\mathbf{T}\rangle = \mathbf{Bra} \quad \mathbf{a} \left( \begin{array}{ccc}
A & C & B \\
\mathbf{V} & \mathbf{U} & \mathbf{W}
\end{array} \right) = \mathbf{T}
\]

He called a vector used as plane equation coefficients like

\[
\langle \mathbf{p} | \mathbf{Ket}\rangle = \mathbf{p}
\]

He called a vector like $\mathbf{p}$

For those who attended Prof. Tomonaco's lecture:

\[
0 = nD + zC + hB + xV = \left( \begin{array}{ccc}
A & C & B \\
\mathbf{V} & \mathbf{U} & \mathbf{W}
\end{array} \right) \left( \begin{array}{c}
m \\
z \\
h \\
x
\end{array} \right)
\]

Equation of a plane (homogeneous coods):
the (A, B, C) of the plane equation.

color to each OpenGL vertex (generated by \text{vertex}(\text{operations}) is
Each OpenGL normal vector (attached like 3D coordinates and RGBA
emission, etc.

A plane is part of a model for light reflection, scattering, coloring,

more.

OpenGL requires at least 6 be accommodated, allows implementations with
Good for section views.

RedBook.

3. Plane is one of \text{OpenGL optional clipping planes}; see HB 7-12 or
\[ 0 = m + z \quad 0 = m + z \quad 0 = m + h \quad 0 = m + h \quad 0 = m + x \quad 0 = m + x \]

2. Plane is one of the 6 NDC clipping planes:

1. Plane is a \text{projection surface} for perspective or parallel projection.

University at Albany Computer Science Dept.
normal vectors.

The vector cross product and length calculation is used to calculate the normal.

See Redbook's Building an Icosahedron Example.

vector and vertex coordinates for each.

The mesh, for a 3D world of solid objects, lights, cameras, etc.

Typically has mesh data structures and hierarchical world models using.

Higher level software:

quadilateral and (plane) polygon primitives.

OpenGL supports drawing of surfaces with triangle, (plane)
and which are outside?

In 2D, given a polygon (possibly with crossing sides), which regions are inside

inside of the surface. (Blender’s rendering HV shows this.)

But backface culling should not be done if we sometimes want to see the

side view. This way is called backface culling.

So, all rendering work can be avoided if the outside side faces cannot from the

any ID boundary edges), none of the inside will be visible from the outside.

2. If the polygon is part of a “closed surface” (actually a 2D surface without

I. The two different sides might have different colors.

Reasons

1. Faces the viewing direction?

2. Faces the outside of a (mesh represented) surface which the polygon is part

Important question: “Issue” in 3D? Which side of a (planar) polygon

University at Albany Computer Science Dept.
determine which side is the OUTSIDE.

Mesh models will typically use the order of the vertices in a polygon to

traverse the triangle in a positive or counter-clockwise order.

The resulting vector will point TOWARD a viewer who sees the sequence A, B, C.

\[
\frac{|\mathbf{Y}|}{N} = \mathbf{Y}
\]

dividing (3 times) to get \(N\) and \(\mathbf{Y}\) then calculate \(\mathbf{X} = (\mathbf{A} - \mathbf{C}) \times (\mathbf{A} - \mathbf{B})\) and normalize \(\mathbf{Y}\) by calculating

\[
[a \hat{z} - q_{z} \hat{a} - q_{y} \hat{y} \hat{a}, a \hat{y} \hat{a} - q_{y} \hat{a} - q_{x} \hat{x} \hat{a}, a \hat{x} \hat{a} - q_{x} \hat{a} - q_{y} \hat{y} \hat{a}] = \mathbf{A} - \mathbf{C}
\]

and

\[
[a \hat{z} - q_{z} \hat{a} - q_{y} \hat{y} \hat{a}, a \hat{y} \hat{a} - q_{y} \hat{a} - q_{x} \hat{x} \hat{a}, a \hat{x} \hat{a} - q_{x} \hat{a} - q_{y} \hat{y} \hat{a}] = \mathbf{A} - \mathbf{B}
\]

calculate all in one line.

Way I: Given coordinates of 3 points \(A, B, C\) not

\[
[A, B, C]
\]

How to calculate the normal for a planar surface:

University at Albany, Computer Science Dept.
The equation of the plane determined by 3 non-coplanar points \( A, \ B, \ C \) is:

\[
0 = \varrho + \varphi + \psi + \chi
\]

This is one scalar equation.

A plane equation in terms of 3 points in the plane.

Way 2: (Really cool way to use projective geometry to remember the formula for
Flat shading: the one normal belonging to the last polygon vertex is used.

OpenGL shading:

Calculation of normals to be attached to OpenGL vertices is necessary for the yz, xz and xy planes. The book says: (α, β, γ) are proportional to the 3 areas of the projections of P on the face, Van Dam, Reimer, Hughes' "Computer Graphics Principles and Practice." Quite all in the same plane? Way (3): What if the mesh's polygon P has 4 or more vertices, and they are not
2. Light source features: shape (of emitting surface and/or angular
surface texture: patterns and/or bumpy roughness.
ambient light, diffuse light, and specular light.
color (different colors can be specified for different model aspects; e.g.,
I. Material properties: Opacity, transparency, shiny or dullness,
Factors variously used in various models:
non-photorealistic images that are attractive and/or informative.)
Models simplify physics, physiology, and psychology, (They can generate
viewer, calculate what RGB value to color the corresponding pixel,
Surface rendering: Given the light intensity, plus the normal, color and
each point on a surface.
Illumination or lighting model: for calculating the light intensity at
HB's vocabulary:
Blender Homework for Lighting and shading.
READ HB 10-1, 10-2, 10-3, 10-20 pages 637-647. Start 10-10 for now.
3. Viewing Features: Distance from surface, flatness, ...

4. Geometric Relationships: Angles among (1) surface normal vector, (2) ...

(3) direction to a light source and (3) direction to the viewer.
Directional Light: Lambert surfaces show dull highlights.

Ambient Light: no directional light, Lambert surfaces look flat.

Diffuse reflected light depends ONLY on the intensity of the incident light.

So the intensity (average power over area, brightness of one spot) of the composite of irregular, sphere-like particles.

Dull, matte surfaces such as chalk or sand reflect light this way. Sand is all directions because whatever light is incident on the surface is reflected equally in all directions. Surface appears equally bright from all Lambert or diffuse reflection.
Angle between the surface normal and the light direction. So, the image intensity is independent of viewing direction, but is proportional to ambient lighting of the diffuse reflector.

The reflection has the same intensity independent of viewing angle. (Just like the angle between $\mathbf{I}$ and the surface normal $\mathbf{N}$ of the angle between $\mathbf{I}$ and the surface normal $\mathbf{N}$ is independent of the cosine of the particular direction $\mathbf{I}$. The intensity at the surface is proportional to the cosine of the angle of incidence from a directional light source. The light comes from a directional reflection of a directional light source. The image has the same intensity no matter what angle is between the normal orientation. The surface orientation is perpendicular to the surface normal.)

All surfaces are illuminated the same way independently of the surface orientation. (The surface orientation is perpendicular to the surface normal.)
between efficiency and realism. It is a compromise between intensity from \( I \), \( N \) and \( \Lambda \) together with parameters. The \( \Phi \) model is one of several (shading models) used to calculate the "angle of reflectance" (away from the normal in the same plane as the normal). The "angle of incidence" (away from the normal in the opposite direction). Law for perfect mirrors: given by Snell's law for point to the light, \( \Phi \) is the direction from the surface. Reflection of a directional light source: Vector is the direction from the viewer. It is most intense in the direction \( R \).
The light intensity of light reflected toward the viewer.

Light intensity is light power per unit area.

An intensive or "local" quantity pertains to a point in an object or an average measured in Calories per Day.

Electric power is usually measured in Kilowatts or Watts. Food (people) power is measured in Calories.

Electric Energy is sometimes measured in Kilowatt-hours. Food energy is measured in Calories.

Light Power = \text{Energy}/\text{Time} is the rate of light energy transfer.

Particular Period of Time

Light energy emitted, reflected or absorbed by a given area during a

Energy, Power, ...

Volume, Weight, Mass, Cross National Product, Area, Length, Electric Charge, etc.

An extensive or "bulk" quantity pertains to an extended object. Examples:

Some Physics...
Winding number at a "test point" of the polygon.

In 2D, the answer is defined using the
A convex polygon has no "cavities" or "harbors" or "indentations".

The polygon, for every pair of points $P_1$, $P_2$ inside or on the boundary and convex endpoints, simple: edges never cross or touch except for adjacent edges at their endpoints; all vertices in one plane.

Only use polygons (like non-degenerate triangles) that are planar. The practical, OpenGL solution:
LINES for OpenGL to approximate them.

Also, GLU support for defining curved surfaces and lines (like circles) by quadratic
formulas (degree 2 polynomials) and then generating lots of polygons or straight
has support for non-convex (i.e., concave) polygons.

GLU (OpenGL standard utility library ON TOP OF, but not INSIDE OpenGL)

See Redbook Chapter II:

tessellate each non-convex face before asking OpenGL to render it.

If an application generates models with non-convex polygons as faces, it must
(3) Use a floating point line-plane intersection calculation for the hard cases.

Entirely on the outside side of any one clipping plane.

(2) Use non-floating point logical calculations to rule out segments that lie
endpoints, based on the 9 or 27 regions the 2 or 3 pairs of clipping planes determine.

Cohen-Sutherland algorithm (1) Calculate two region codes, one for each
line segment, all in NDC.

Input: The 6 clipping planes and the coordinates of the 2 endpoint vertices of a
Clipping Algorithms for line segments in 2D or 3D:
Liang-Barsky algorithm:

University at Albany Computer Science Dept.

21