Graphics Pipeline
2D Geometric Transformations

CS 4620 Lecture 8
Plane projection in drawing
Plane projection in drawing

The concept of the picture plane may be better understood by looking through a window or other transparent plane from a fixed viewpoint. Your lines of sight, the multitude of straight lines leading from your eye to the subject, will all intersect this plane. Therefore, if you were to reach out with a grease pencil and draw the image of the subject on this plane you would be “tracing out” the infinite number of points of intersection of sight rays and plane. The result would be that you would have “transferred” a real three-dimensional object to a two-dimensional plane.
Rasterizing triangles

• **Summary**
  1. Evaluate linear functions on pixel grid.
  2. Functions defined by parameter values at vertices.
Pixel-walk (Pineda) rasterization

- Conserve by visiting a superset of the pixels you want.
- Interpolate linear functions.
- Use those functions to determine when to emit a fragment.
The z buffer

- another example of a memory-intensive brute force approach that works and has become the standard

```plaintext

<table>
<thead>
<tr>
<th>0 0 0 0 0 0 0 0</th>
<th>0 0 0 0 0 0 0 0</th>
<th>0 0 0 0 0 0 0 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>5 5 5 5 5 5 5 5</td>
<td>5 5 5 5 5 5 5 5</td>
<td>5 5 5 5 5 5 5 5</td>
</tr>
<tr>
<td>5 5 5 5 5 5 5 5</td>
<td>5 5 5 5 5 5 5 5</td>
<td>5 5 5 5 5 5 5 5</td>
</tr>
<tr>
<td>5 5 5 5 5 5 5 5</td>
<td>5 5 5 5 5 5 5 5</td>
<td>5 5 5 5 5 5 5 5</td>
</tr>
<tr>
<td>5 5 5 5 5 5 5 5</td>
<td>5 5 5 5 5 5 5 5</td>
<td>5 5 5 5 5 5 5 5</td>
</tr>
<tr>
<td>5 5 5 5 5 5 5 5</td>
<td>5 5 5 5 5 5 5 5</td>
<td>5 5 5 5 5 5 5 5</td>
</tr>
<tr>
<td>5 5 5 5 5 5 5 5</td>
<td>5 5 5 5 5 5 5 5</td>
<td>5 5 5 5 5 5 5 5</td>
</tr>
</tbody>
</table>

+----------------+----------------+----------------|
| 3 4 3 | 5 6 6 | 7 7 7 |
| 5 5 5 | 6 6 6 | 7 7 7 |
| 5 5 5 | 6 6 6 | 7 7 7 |
| 5 5 5 | 6 6 6 | 7 7 7 |
| 5 5 5 | 6 6 6 | 7 7 7 |
| 5 5 5 | 6 6 6 | 7 7 7 |

=----------------+----------------+----------------|
| 5 5 5 5 5 5 5 5 | 5 5 5 5 5 5 5 5 | 5 5 5 5 5 5 5 5 |
| 5 5 5 5 5 5 5 5 | 5 5 5 5 5 5 5 5 | 5 5 5 5 5 5 5 5 |
| 5 5 5 5 5 5 5 5 | 5 5 5 5 5 5 5 5 | 5 5 5 5 5 5 5 5 |
| 5 5 5 5 5 5 5 5 | 5 5 5 5 5 5 5 5 | 5 5 5 5 5 5 5 5 |
| 5 5 5 5 5 5 5 5 | 5 5 5 5 5 5 5 5 | 5 5 5 5 5 5 5 5 |
| 5 5 5 5 5 5 5 5 | 5 5 5 5 5 5 5 5 | 5 5 5 5 5 5 5 5 |
| 5 5 5 5 5 5 5 5 | 5 5 5 5 5 5 5 5 | 5 5 5 5 5 5 5 5 |

```
Pipeline

3D transformations; shading

conversion of primitives to pixels

blending, compositing, shading

user sees this

APPLICATION

COMMAND STREAM

VERTEX PROCESSING

TRANSFORMED GEOMETRY

RASTERIZATION

FRAGMENTS

FRAGMENT PROCESSING

FRAMEBUFFER IMAGE

DISPLAY
three.js - dynamic procedural terrain using 3d.simplex noise
birds by mirada from ro.me - textures by qubodup and davis123 - music by Kevin MacLeod
Some demos

- http://mrdoob.github.io/three.js/examples/
- http://carvisualizer.plus360degrees.com/threejs/
- http://madebyevan.com/webgl-water/
- http://akirodic.com/p/jellyfish/
Pipeline of transformations

- Standard sequence of transforms
A little quick math background

- Notation for sets, functions, mappings
- Linear transformations
- Matrices
  - Matrix-vector multiplication
  - Matrix-matrix multiplication
- Geometry of curves in 2D
  - Implicit representation
  - Explicit representation
Implicit representations

• Equation to tell whether we are on the curve
  \[ \{ v | f(v) = 0 \} \]

• Example: line (orthogonal to \( u \), distance \( k \) from \( 0 \))
  \[ \{ v | v \cdot u + k = 0 \} \quad (u \text{ is a unit vector}) \]

• Example: circle (center \( p \), radius \( r \))
  \[ \{ v | (v - p) \cdot (v - p) - r^2 = 0 \} \]

• Always define boundary of region
  – (if \( f \) is continuous)
Explicit representations

- Also called parametric
- Equation to map domain into plane
  \[ \{ f(t) \mid t \in D \} \]
- Example: line (containing \( p \), parallel to \( u \))
  \[ \{ p + tu \mid t \in \mathbb{R} \} \]
- Example: circle (center \( b \), radius \( r \))
  \[ \{ p + r[ \cos t \: \sin t ]^T \mid t \in [0, 2\pi) \} \]
- Like tracing out the path of a particle over time
- Variable \( t \) is the “parameter”
Transforming geometry

\[ S \rightarrow \{T(v) \mid v \in S\} \]

- Parametric representation:
  \[ \{f(t) \mid t \in D\} \rightarrow \{T(f(t)) \mid t \in D\} \]

- Implicit representation:
  \[ \{v \mid f(v) = 0\} \rightarrow \{T(v) \mid f(v) = 0\} \]
  \[ = \{v \mid f(T^{-1}(v)) = 0\} \]
Translation

- Simplest transformation: $T(v) = v + u$
- Inverse: $T^{-1}(v) = v - u$
Linear transformations

- One way to define a transformation is by matrix multiplication:
  \[ T(v) = Mv \]

- Such transformations are linear, which is to say:
  \[ T(au + v) = aT(u) + T(v) \]
  (and in fact all linear transformations can be written this way)
Geometry of 2D linear trans.

• 2x2 matrices have simple geometric interpretations
  – uniform scale
  – non-uniform scale
  – rotation
  – shear
  – reflection

• Reading off the matrix
Linear transformation gallery

- Uniform scale

\[
\begin{bmatrix}
  s & 0 \\
  0 & s \\
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
\end{bmatrix} =
\begin{bmatrix}
  sx \\
  sy \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
  1.5 & 0 \\
  0 & 1.5 \\
\end{bmatrix}
\]
Linear transformation gallery

- Nonuniform scale

\[
\begin{bmatrix}
    s_x & 0 \\
    0 & s_y
\end{bmatrix}
\begin{bmatrix}
    x \\
    y
\end{bmatrix}
=
\begin{bmatrix}
    s_x x \\
    s_y y
\end{bmatrix}
\begin{bmatrix}
    1.5 & 0 \\
    0 & 0.8
\end{bmatrix}
\]
Linear transformation gallery

- Reflection
  - can consider it a special case of nonuniform scale

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

\[\text{R} \quad \text{R} \]

Linear transformation gallery

- Rotation

\[
\begin{bmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix}
= 
\begin{bmatrix}
x \cos \theta - y \sin \theta \\
x \sin \theta + y \cos \theta
\end{bmatrix}
\]

\[
\begin{bmatrix}
0.866 & -0.5 \\
0.5 & 0.866
\end{bmatrix}
\]
Linear transformation gallery

- Shear

\[
\begin{bmatrix}
1 & a \\
0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
\end{bmatrix}
= 
\begin{bmatrix}
x + ay \\
y \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 0.5 \\
0 & 1 \\
\end{bmatrix}
\]
Composing transformations

- Want to move an object, then move it some more
  - \( p \rightarrow T(p) \rightarrow S(T(p)) = (S \circ T)(p) \)
- We need to represent \( S \circ T \) ("S compose T")
  - and would like to use the same representation as for \( S \) and \( T \)
- Translation easy
  - \( T(p) = p + u_T; S(p) = p + u_S \)
  - \( (S \circ T)(p) = p + (u_T + u_S) \)
- Translation by \( u_T \) then by \( u_S \) is translation by \( u_T + u_S \)
  - commutative!
Composing transformations

• Linear transformations also straightforward

\[ T(p) = M_T p; S(p) = M_S p \]

\[ (S \circ T)(p) = M_S M_T p \]

• Transforming first by \( M_T \) then by \( M_S \) is the same as transforming by \( M_S M_T \)
  – only sometimes commutative
    • e.g. rotations & uniform scales
    • e.g. non-uniform scales w/o rotation
  – Note \( M_S M_T \), or \( S \circ T \), is \( T \) first, then \( S \)
Combining linear with translation

• Need to use both in single framework
• Can represent arbitrary seq. as \( T(p) = M p + u \)
  \(- T(p) = M_T p + u_T \)
  \(- S(p) = M_S p + u_S \)
  \(- (S \circ T)(p) = M_S(M_T p + u_T) + u_S \)
    \[ = (M_S M_T) p + (M_S u_T + u_S) \]
  \(- \text{e.g. } S(T(0)) = S(u_T) \)
• Transforming by \( M_T \) and \( u_T \), then by \( M_S \) and \( u_S \), is the same as transforming by \( M_S M_T \) and \( u_S + M_S u_T \)
  \(- \text{This will work but is a little awkward} \)
Homogeneous coordinates

- A trick for representing the foregoing more elegantly
- Extra component $w$ for vectors, extra row/column for matrices
  - for affine, can always keep $w = 1$
- Represent linear transformations with dummy extra row and column

\[
\begin{bmatrix}
  a & b & 0 \\
  c & d & 0 \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix} =
\begin{bmatrix}
  ax + by \\
  cx + dy \\
  1
\end{bmatrix}
\]
Homogeneous coordinates

- Represent translation using the extra column

\[
\begin{bmatrix}
1 & 0 & t \\
0 & 1 & s \\
0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
1 \\
\end{bmatrix}
= 
\begin{bmatrix}
x + t \\
y + s \\
1 \\
\end{bmatrix}
\]
Homogeneous coordinates

- Composition just works, by 3x3 matrix multiplication

\[
\begin{bmatrix}
M_S & \mathbf{u}_S \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
M_T & \mathbf{u}_T \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
\mathbf{p} \\
1
\end{bmatrix}
\]

\[
= \begin{bmatrix}
(M_SM_T)\mathbf{p} + (M_S\mathbf{u}_T + \mathbf{u}_S) \\
1
\end{bmatrix}
\]

- This is exactly the same as carrying around \(M\) and \(\mathbf{u}\)
  - but cleaner
  - and generalizes in useful ways as we’ll see later