Describing Shape

By Andries van Dam
**2D Object Definition (1/3)**

*Lines and Polylines*

- **Polylines**: lines drawn between ordered points

![Polyline diagram](image)

- Same first and last point make *closed polyline* or *polygon*
- If it does not intersect itself, called *simple polygon*

**Convex vs. Concave Polygons**

**Convex**: For every pair of points in the polygon, the line between them is fully contained in the polygon.

![Convex polygon](image)

**Concave**: Not convex: some two points in the polygon are joined by a line not fully contained in the polygon.

![Concave polygon](image)
Circles

- Consist of all points equidistant from one predetermined point (the center)
- (radius) $r = c$, where $c$ is a constant
- On a Cartesian grid with center of circle at origin equation is $r^2 = x^2 + y^2$
Circle as polygon

- A circle can be approximated by a polygon with many sides (>15)

(Aligned) Ellipses

A circle scaled along the x or y axis

Example: height, on y-axis, remains 3, while length, on x-axis, changes from 3 to 6
Representing Shape in 2D

- General purpose, simple overhead, reasonable efficiency: Vertex and Edge tables
- Each vertex listed once
- Each edge is ordered pair of indices into vertex table
- Sufficient information to draw shape and perform simple operations.
- Order does not matter, convention is edges listed in counterclockwise order.

<table>
<thead>
<tr>
<th>Vertexes</th>
<th>Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (0,0)</td>
<td>( (1,2) )</td>
</tr>
<tr>
<td>2 (1,0)</td>
<td>( (2,4) )</td>
</tr>
<tr>
<td>3 (0,1)</td>
<td>( (4,5) )</td>
</tr>
<tr>
<td>4 (1,1)</td>
<td>( (5,3) )</td>
</tr>
<tr>
<td>5 (0.5,1.5)</td>
<td>( (3,1) )</td>
</tr>
</tbody>
</table>
Splines (An Alternate Representation)

- How they work: Parametric curves governed by control points
- Mathematically: Several representations to choose from. More complicated than vertex lists. See chapter 23 of the book for more information.

Simple parametric representation:

\[
x(t) = a_3 t^3 + a_2 t^2 + a_1 t + a_0
\]

\[
y(t) = b_3 t^3 + b_2 t^2 + b_1 t + b_0
\]

- Advantage: Smooth with just a few control points
- Disadvantage: Can be hard to control
- Uses:
  - representation of smooth shapes. Either as outlines in 2D or with Patches or Subdivision Surfaces in 3D
  - animation Paths for tweening
  - approximation of truncated Gaussian Filters
2D to 3D Object Definition

*Vertices in motion (“Generative object description”)*

- Line is drawn by tracing path of a point as it moves (one dimensional entity)

- Square drawn by tracing vertices of a line as it moves perpendicularly to itself (two dimensional entity)

- Cube drawn by tracing paths of vertices of a square as it moves perpendicularly to itself (three-dimensional entity)

- Circle drawn by swinging a point at a fixed length around a center point
Building 3D Primitives

- Triangles and tri-meshes
- Parametric polynomials, like the aforementioned splines used to define surface patches.
Triangle Meshes

- Most common representation of shape in three dimensions
- All vertices of triangle are guaranteed to lie in one plane (unlike quadrilaterals or other polygons)
- Uniformity makes it easy to perform mesh operations: subdivision, simplification, etc.
- Many different ways to represent triangular meshes:

http://en.wikipedia.org/wiki/Polygon_mesh
Representing Shape in 3D

- Analogous to 2D: Vertex and Triangle Tables
- Each vertex gets listed once
- Each triangle is ordered triple of indices into the vertex table
- Edges between vertexes inferred from triangles
- Only need the triangular mesh representations to draw the shapes; need not do operations on meshes during CS425
- Counterclockwise ordering of vertices for normals

![Face-Vertex Meshes](http://upload.wikimedia.org/wikipedia/en/thumb/2/2d/Mesh_fv.jpg/500px-Mesh_fv.jpg)