Lecture 7

Clipping Concluded;
Introduction to Curve Representations

Wednesday, February 9, 2000

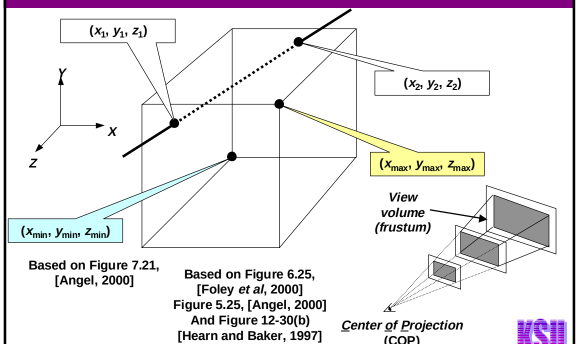
William H. Hsu
Department of Computing and Information Sciences, KSU
http://www.cis.ksu.edu/~bhsu

Readings:
Sections 11.1, 11.2.1-11.2.2, Foley *et al*
(Reference) Appendix A.9, 6.7-6.11, 10.6-10.8, Hearn and Baker 2^e^
(Reference) 10.1-1.3,10.6, Angel 2^e^

Lecture Outline

- **Readings**
  - Sections 11.1, 11.2.1-11.2.2, FVD
  - Optional references: Hearn and Baker 2^e^, Angel 2^e^
- **Quick Review: 3D Viewing**
  - 3D view volume
  - Perspective-to-parallel transformation (Section 6.5, FVD)
- **Clipping (Concluded)**
  - Quick review: clipping using simultaneous equations; parametric clipping
  - Clipping in 3D
  - Clipping polygons, curves and clipping against windows
- **Introduction to Curve Representations**
  - What are cubic curves?
  - How would we use them to represent surfaces?
  - Read about polygon meshes, Bézier curves, B-splines for next time!
- **Next Lecture: Cubic Curves and Surfaces**

Quick Review: 3D View Volumes



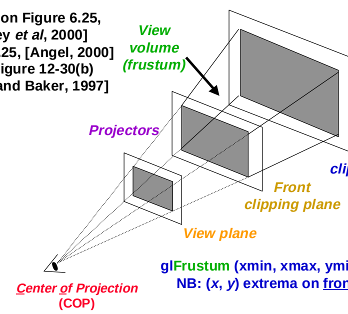
Based on Figure 7.21, [Angel, 2000]

Based on Figure 6.25, [Foley *et al*, 2000] Figure 5.25, [Angel, 2000] And Figure 12-30(b) [Hearn and Baker, 1997]

Quick Review: Specifying an Arbitrary 3D View

Based on Figure 6.25, [Foley *et al*, 2000] Figure 5.25, [Angel, 2000] and Figure 12-30(b) [Hearn and Baker, 1997]

Perspective View Volume



glFrustum (xmin, xmax, ymin, ymax, near, far)
NB: (x, y) extrema on front clipping plane

Quick Review: From Perspective to Parallel

- **Perspective-to-Parallel Viewing Transformation**
  - $N'_{per} = M \cdot S_{per} \cdot SH_{par} \cdot T(-PRP) \cdot R \cdot T(-VRP)$ (Equation 6.49, FVD)
  - $S_{per} \cdot SH_{par} \cdot T(-PRP) \cdot R \cdot T(-VRP) \equiv$ matrix stack
    - At point (VRP) to origin – translate
    - Align (x, y, z) with (u, v, n) – rotate
    - Eye point (COP) to origin – translate
    - Center line of view volume aligned with z – shear
    - Normalize to (+1, -1) canonical view volume – scale
  - But wait…
    - Q: What is M?
    - A: perspective-to-parallel viewing transformation
- **Significance**
  - Q: Why is this important?
  - A: Clipping is much uglier without M
  - Q: What kind of transformation is M?
  - A: Nonuniform scaling (see 6.5, FVD; 12.4, Hearn and Baker)

Clipping

- **Problem Definition**
  - Given: coordinates for primitives (line segments, polygons, circles, ellipses, etc.)
  - Determine: *visible components* of primitives (e.g., line segments)
- **Objectives**
  - Correctness: display primitive components *iff* visible
  - Efficiency
    - Fewer tests
    - Fewer floating point operations *per test*
- **Solution Approaches**
  - Solving simultaneous equations (quick rejection: testing endpoints)
  - Solving parametric equations
- **Clipping in 3D**
  - 2D algorithms extendible to 3D: Cohen-Sutherland, Liang-Barsky
  - *Specification (and implementation) of view volumes needed*
- **Transparent Implementation in Graphics APIs: Last Lecture**

Clipping Lines [1]: Simultaneous Equations

- Recall: Cohen-Sutherland Quick Rejection Method
  - Unique 4-bit binary number (*outcode*) for each of 9 regions
  - Check clipping cases (NB: *not necessarily conclusive*!)
    - 2 outcodes per line segment: o_1, o_2
    - Case 1: $o_1 = o_2 = 0000$
    - Case 2: $o_1 = 0000$, $o_2 \neq 0000$ (or vice versa)
    - Case 3: o_1 & $o_2 \neq 0000$
    - Case 4: o_1 & $o_2 = 0000$

1001	1000	1010	y_{max}
0001	0000	0010	y_{min}
0101	0100	0110	
	x_{min}	x_{max}	

- Using Cohen-Sutherland
  - Check all segments with inconclusive outcodes (Case 4)
    - Compute line-rectangle intersection using simultaneous equations
    - $y = y_1 + m(x - x_1)$, $x = x_{min}$ or x_{max}
    - $x = x_1 + (y - y_1) / m$, $y = y_{min}$ or y_{max}
  - In general: parametric tests require more floating-point operations, but better performance in worst-case (or nearly so – see Homework 1)

Clipping Lines [2]: Parametric Equations

- Intuitive Idea
  - Use parametric line formulation: $P(t) = P_0 + (P_1 - P_0)t$
  - Find 4 t values for 4 clip edges
  - Decide which of these form true intersections
  - Calculate (x, y) for those only
  - There can be only… 2
- Cyrus-Beck: Earliest Parametric Line Clipper (Figure 3.44, 3.12.4 FVD)
- Liang-Barsky (Figure 3.45, 3.12.4 FVD)
  - Independently developed
  - Slightly faster for clipping against rectangles (*extra rejection testing*)
  - Main references
    - Section 3.12, FVD
    - vanDam, Lecture 4-3 (09/30/1999)
  - Optional reference: Section 6.7 Hearn and Baker 2[e]
- Next Lecture: Cubic Curves and Surfaces

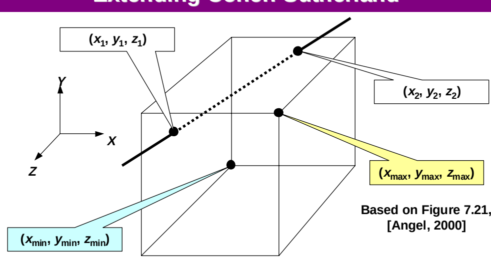
Clipping Polygons

- Section 3.10, FVD; Section 6.8, Hearn and Baker; Section 7.9, Angel
- Intuitive Idea
  - "Can use line-clipping procedures"
  - But it's not so simple!
    - Really want: bounded area after clipping
    - Need: algorithm that will generate *one or more closed areas to be scan-converted*
    - Output specification: sequence of vertices defining clipped polygon boundaries
- Implementation
  - Algorithm (Sutherland-Hodgeman)
    - Process polygon boundary (all segments) against each edge
    - Correctness issue: concave polygons (may have *multiple clipped polygons*)
    - Scan conversion: sweepline algorithm (x, y, or polar coordinates)
  - Can also modify Liang-Barsky (similar idea)

Clipping Curves, Text, and Window Contents

- Curves and Surfaces
  - Section 6.9, Hearn and Baker; Section 7.5.2, Angel
  - Brute-force quick-rejection test (often used)
    - Bounding box
    - Like outcodes, may be inconclusive – need line-curve intersection tests (sigh)
- Text
  - Section 6.10, Hearn and Baker; Section 7.5.1-7.5.2, Angel
  - Can also use bounding box (all text, words, letters) – *aka* all-or-none
  - If have to clip letter, may need geometric or sample (bitmap) font definition
- Windows in GUIs
  - Section 6.11, Hearn and Baker; Section 7.5.1, Angel
  - Sounds easy, right? Not necessarily…
    - Exterior clipping: may want to clip "inverse" of window (e.g., maximized windows in GUIs)
    - Windowing system resolves exterior versus interior clipping

Clipping in 3D [1]: Extending Cohen-Sutherland



Based on Figure 7.21, [Angel, 2000]

Clipping in 3D [2]: Extending Liang-Barsky

- How To Use Liang-Barsky in 3D
  - Now need to clip 6 faces instead of 4 edges
    - Right
    - Left
    - Bottom
    - Top
    - Front
    - Back
  - Performance
    - Cyrus-Beck: for 3-D perspective pyramid
    - Liang-Barsky: can do faster using cuboid
- See Table 6.1, Figure 6.55, FVD

Curve Representations: Overview

- Representations
  - Polygon mesh
  - Parametric cubic equations (splines, etc.)
- Parametric Cubic Curves (Table 11.2, FVD)
  - Hermite curves
  - Bezier curves
  - Cubic splines
    - Uniform B-splines
    - Uniformly shaped β-spline
    - Nonuniform B-splines (rational: NURBS; nonrational)
    - Catmull-Rom
    - Kochanek-Bartels
- References
  - Section 11.2, FVD
  - 10.6-10.8, Hearn and Baker 2[e], 10.1-1.3,10.6, Angel 2[e]

Cubic Curves: Interpolation Methods

- Cubic Curves
  - Defined using control points
- Intuitive Idea
  - Interpolation: fitting curve
    - Touches specified control points (e.g., endpoints and midpoint)
    - Smooth (twice differentiable)
    - Fit to other control points
  - See properties in Table 11.2
    - Convex hull defined by control points?
    - Interpolation method
    - Ease of subdivision
- Next Time
  - Defining cubic curves
  - Computing, render interpolations
  - Rendering bicubic surfaces

Course Project: Suggested Topics

- Photorealistic Rendering
  - Scene illumination – ray tracing, radiosity, fast shading, etc.
  - Artificial objects – transparent surfaces, faces, natural scenes
- Realistic Animation
  - Animating specific entities – human faces, hair, bodies, etc.
  - Advanced topics – physically based modeling, particle systems, etc.
- Image Processing
  - 2D image transformation – compression, correction, etc.
  - 2D image analysis – edge detection, pattern recognition, KDD, etc.
  - Mapping – texture, bump, etc.
- Visualization
  - Statistical data visualization, information visualization
  - Scientific visualization – fluid dynamics, geology, groundwater, etc.
- Mathematical Modeling
  - Geometric models: curves, surfaces, 3D solid models, etc.
  - Fractal image synthesis, fractal image compression, other

Course Project: More Guidelines

- Project Proposal (Due 02/14/2000)
  - 1-3 page description of project topic, plan
  - See: implementation practicum links (Brown, Cornell, UNC, others) on 736 page
- Guidelines
  - Specify focus of your project
    - Implementation (e.g., "write a simple animator for articulated figures")
    - Experiments with existing algorithms (e.g., "compare texture mappers *A*, *B*")
    - Experiments with visualizing data and processes (e.g., "write a visualization front end for integrity checking in DBMS *D*")
    - Extending or combining existing algorithms (e.g., "add multiple light-source technique to ray tracer *R*")
    - Applying CG techniques to specific problem (e.g., "rendering back end for solid CAD/CAM machining model")
  - State how will you demonstrate your results
    - How will you measure or show success?
    - *Make sure your goals are sufficiently narrow and realistic*
- Three Parts: Proposal (20%), Implementation (50%), Report (30%)

Terminology

- Review: Normalizing Transformations
  - $N'_{per} = M \cdot S_{per} \cdot SH_{par} \cdot T(-PRP) \cdot R \cdot T(-VRP)$ (Equation 6.49, FVD)
  - *M*: nonuniform scaling transformation (perspective-to-parallel)
- Line Clipping: Determining Parts of Line Segment Primitives to Display
  - Quick rejection testing for simultaneous equations (Cohen-Sutherland)
    - Division of plane into 9 regions with (4-bit) outcodes
    - Testing endpoints of line segment
  - Parametric clipping: line / rectangle intersection using parametric equation
    - Cyrus-Beck: general convex 3D polyhedron
    - Liang-Barsky: more efficient, specialized variant (upright 2D, 3D clip regions)
- Clipping in 3D
  - Cuboid: truncated viewing pyramid used to clip after N_{par}
  - Frustum: truncated viewing pyramid
- Cubic Curves
  - Definition: representation of curve by polynomial (usually smooth) of order 3
  - Interpolation: fitting curves given specified (control) points

Summary Points

- Quick Review: 3D Viewing
  - 3D view volume
  - Perspective-to-parallel transformation (Section 6.5, FVD)
- Clipping (Concluded)
  - Quick review: problem definition, objectives, general approaches
    - Simultaneous equations and quick rejection test (Cohen-Sutherland)
    - Parametric clipping (Cyrus-Beck / Liang-Barsky)
  - Clipping in 3D (extending Cohen-Sutherland, Liang-Barsky)
  - Clipping polygons, curves
  - Clipping against windows (exterior clipping)
- Introduction to Curve Representations
  - Cubic curves: Bézier curves, cubic B-splines
  - Cubic surfaces: next
  - Read about polygon meshes, Bézier curves, B-splines for next time!
- Next Lecture: Cubic Curves and Surfaces