Johns Hopkins
Engineering for Professionals

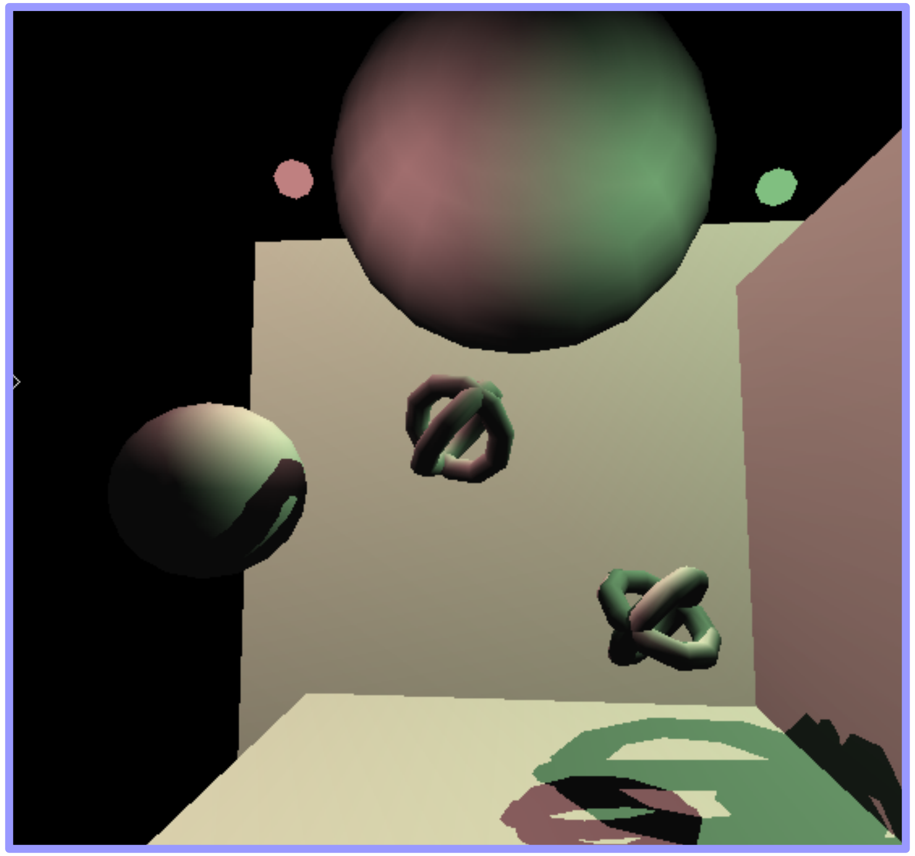
605.767 Applied Computer Graphics

Brian Russin

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Module 8D
Shadow Volumes

Shadow Volumes



Good overall reference: http://www.gamedev.net/reference/articles/article1873.asp

Shadow Volumes

- Crow (1977) developed an algorithm to generate shadows by creating **shadow volumes** or **volumetric shadows**
  - Object precision method
  - Volume that each object blocks from a light source
    - Invisible region of space swept out by the shadow of an object
  - One of the most common and practical methods of generating shadows for polygonal objects
- Shadow volumes are found by evaluating the **contour edge** or **silhouette edge** of the object
  - Contour edge separates those polygons that can see the light source from those that cannot
- Shadow volume is itself polygonal
  - Volumes are semi-infinite pyramids
  - Shadow volume polygons are invisible - not rendered
    - They are used to determine shadows on 'real' polygons
  - Normals point to the outside of the shadow volume

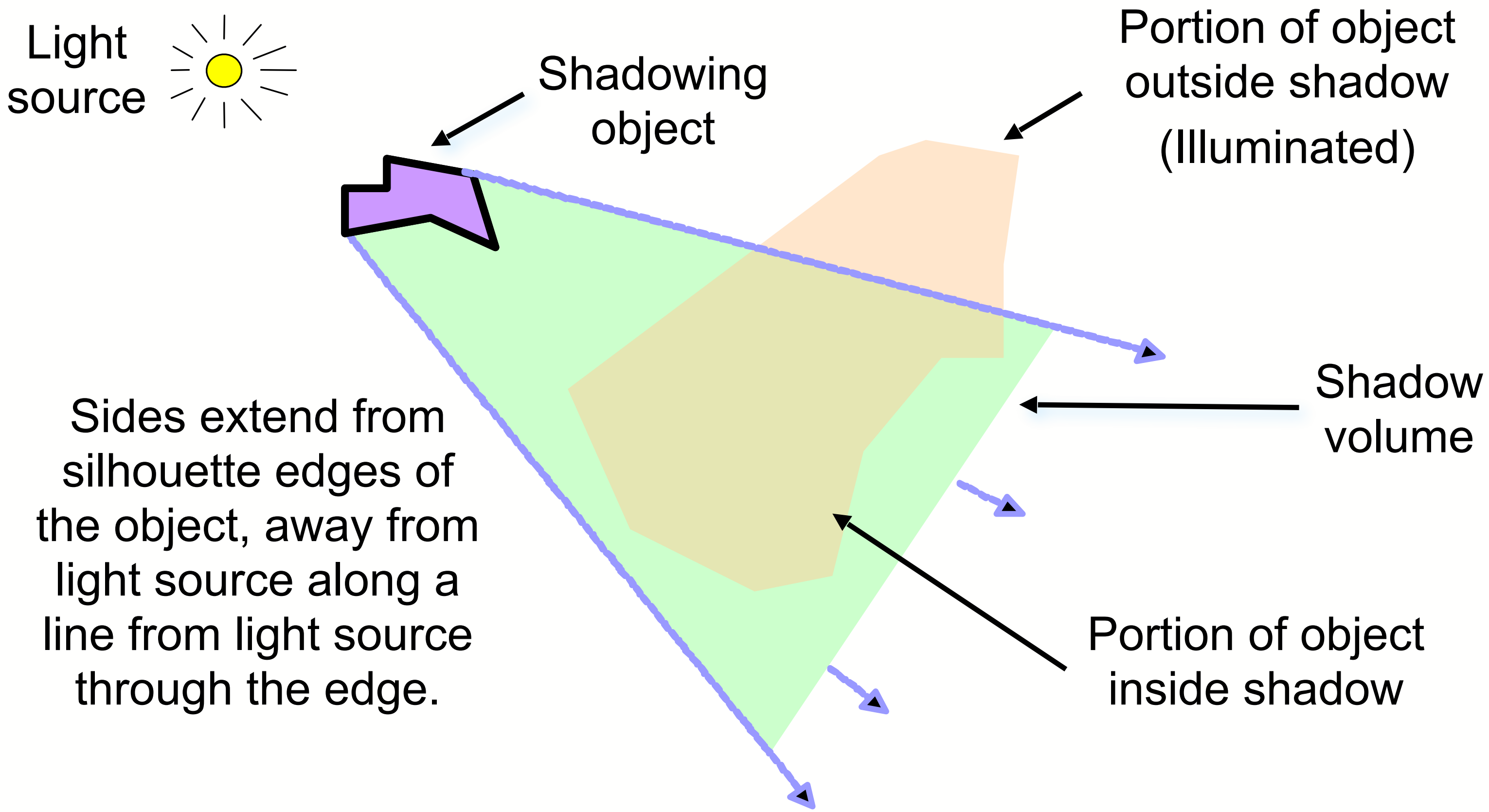
Shadow Volumes (cont.)

- Can be integrated with a number of hidden surface algorithms
  - Polygons defining the shadow volume are processed along with object polygons
  - Distinction is made between front-facing and back-facing polygons
  - A point on an object is in shadow if it is behind a front-facing shadow polygon and in front of a back facing shadow polygon
    - Contained within a shadow volume
    - A front facing shadow polygon puts anything behind it in shadow
    - Back-facing shadow polygon cancels the effects of a front facing

Shadow Volumes (cont.)

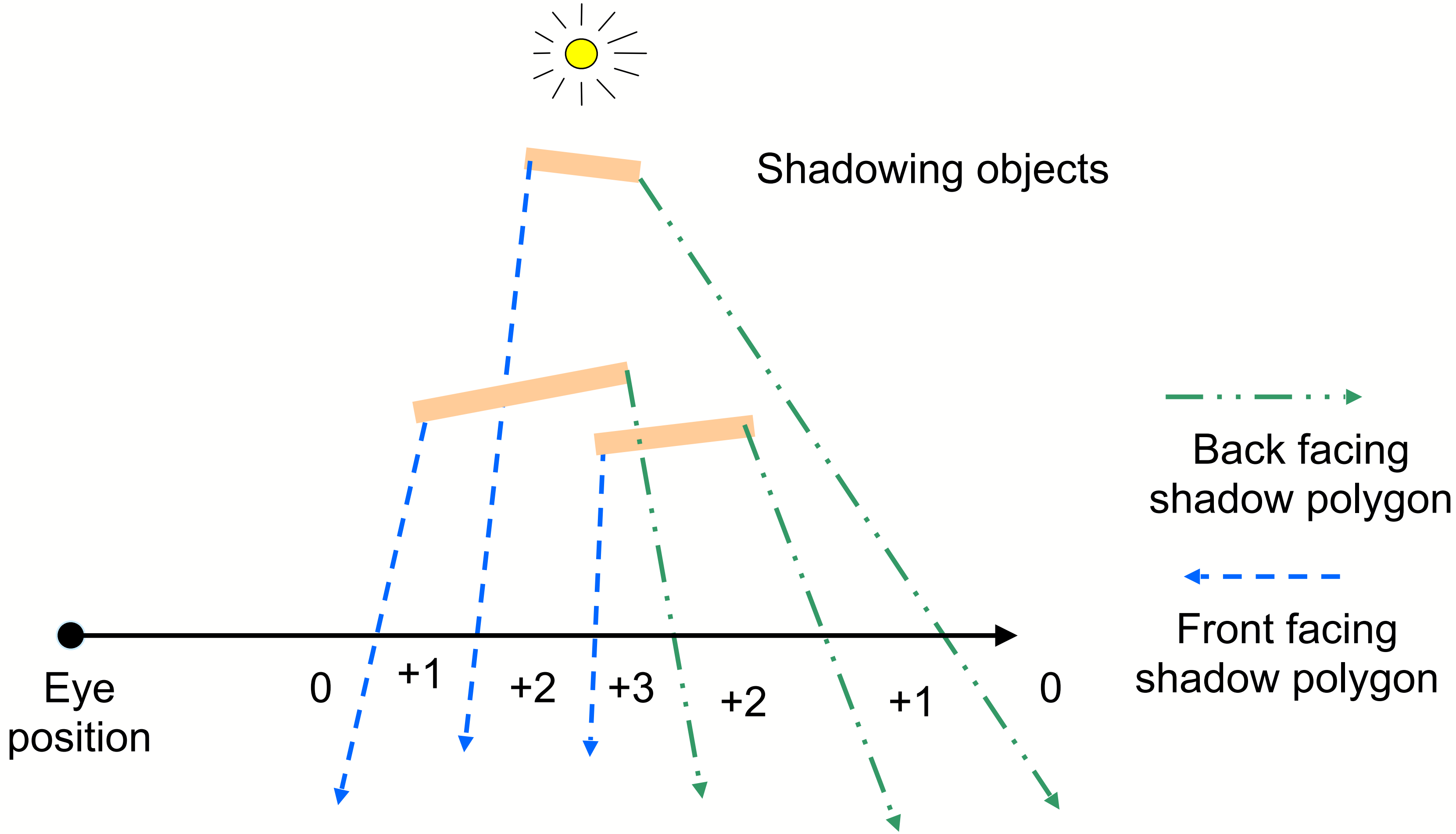
- Most easily integrated with a depth sort hidden surface algorithm
  - Maintain a counter for each pixel
    - Increment it when a front facing shadow polygon is encountered
    - Decrement it when a rear facing shadow polygon is encountered
    - When we encounter a real polygon the value will tell us if we are in shadow
- Can be integrated with a stencil buffer
  - Heidmann (1991) first proposed use of stencil buffer to assist with shadow volumes
  - Extra memory cost
  - “Optimized Stencil Shadow Volumes”, Mark Kilgard and Cass Everitt of NVidia, presented at GDC 2003
    - https://www.nvidia.com/docs/IO/8230/GDC2003_ShadowVolumes.pdf

Shadow Volume


Light source
Shadowing object
Portion of object outside shadow (Illuminated)
Shadow volume
Sides extend from silhouette edges of the object, away from light source along a line from light source through the edge.
Portion of object inside shadow

A shadow volume is the half-space defined by a light source and a shadowing object. Any object within the shadow volume is within shadow from the light source.

Counting Approach



Depth Pass Stencil Volumes

- For shadow volumes with intersecting polygons
    - Use a stencil enter/leave counting approach
- Two-pass approach to populate the stencil buffer
    - Draw shadow volume twice using face culling
        - 1st pass: render front faces and increment when depth test passes
            - glStencilOp(GL_KEEP, GL_KEEP, GL_INCR);
        - 2nd pass: render back faces and decrement when depth test passes
            - glStencilOp(GL_KEEP, GL_KEEP, GL_DECR);
        - Disable drawing to framebuffer and depth buffer
- When drawing the scene, pixels in shadow will have non-zero stencil values
- Haines and Moller discuss multi-pass method
    - Clear stencil buffer
    - Render scene with just ambient and emission
    - Perform the 2 passes to populate stencil buffer
    - Render scene with diffuse and specular components

Depth Fail Stencil Volumes

- Depth pass shadow volume techniques fail if camera enters a shadow volume
- Carmack as well as Bilodeau and Songy came up with an alternate approach
  - Often call Carmack’s Reverse algorithm
- Two-pass approach to populate the stencil buffer
  - Draw shadow volume twice using face culling
    - 1st pass: render back faces and increment when depth test fails
      - glStencilOp(GL_KEEP, GL_INCR, GL_KEEP);
    - 2nd pass: render front faces and decrement when depth test passes
      - glStencilOp(GL_KEEP, GL_DECR, GL_KEEP);
    - Disable drawing to framebuffer and depth buffer
- Shadow volumes must be capped to form a closed volume
  - Haines and Moller discuss methods to cap shadow volumes
  - One method include hardware support for depth clamping
  - Software method by Kilgard is described in text

Generating Shadow Volumes

- Naïve shadow volume methods draw many shadow volume polygons
  - One quadrilateral for every triangle
- More efficient to only draw shadow volumes using the silhouette edges of an object
  - Silhouette edges separate front facing from back facing polygons
    - One “adjoining” polygon faces the light and the other faces away
  - Haines and Moller describe using geometry shaders to do this
    - Quadrilaterals created as output from the geometry shader
- Predefine simple polygons for each shadowing object
  - Does not work well on complex objects

Generating Shadow Volumes (cont.)

- Software structures and methods also available to form silhouette edges
  - Efficiency improved if neighboring polygon information is stored within the object
    - Edge-based structures help as each edge points to polygons sharing
  - http://www.gamedev.net/page/resources/_/technical/graphics-programming-and-theory/the-theory-of-stencil-shadow-volumes-r1873
- Also see “Real-Time Shadow Casting Using Shadow Volumes”
  - Jason Bestimt and Bryan Freitag
  - https://www.gamedeveloper.com/business/real-time-shadow-casting-using-shadow-volumes
- Shadow volumes have for the most part been abandoned in recent applications: in favor of Shadow Mapping