

Shadow Volumes

⇒ Agenda:

- Assignment #2
 - Discuss
 - Hand in!
- Shadow volumes
 - Z-pass
 - Z-fail
- Lab time:
 - Work on assignment #3

Shadow volume intro

- ⇒ Proposed by Frank Crow in 1977
 - Add new geometry to the scene that describes the volume occluded from the light source
 - Objects within the volume are in shadow, objects not within the volume are not
- ⇒ In 1991, Tim Heidmann showed how the stencil buffer can be used to apply these volumes to a scene

Shadow volume intro (cont.)

⇒ Basic algorithm:

1. Render scene using only ambient light
2. For each light in the scene:
 - a. Using the depth information from the initial pass, construct a stencil with “holes” where there the light is not occluded.
 - b. Render scene again with normal lighting. Use the stencil mask to only draw where the light is not occluded.

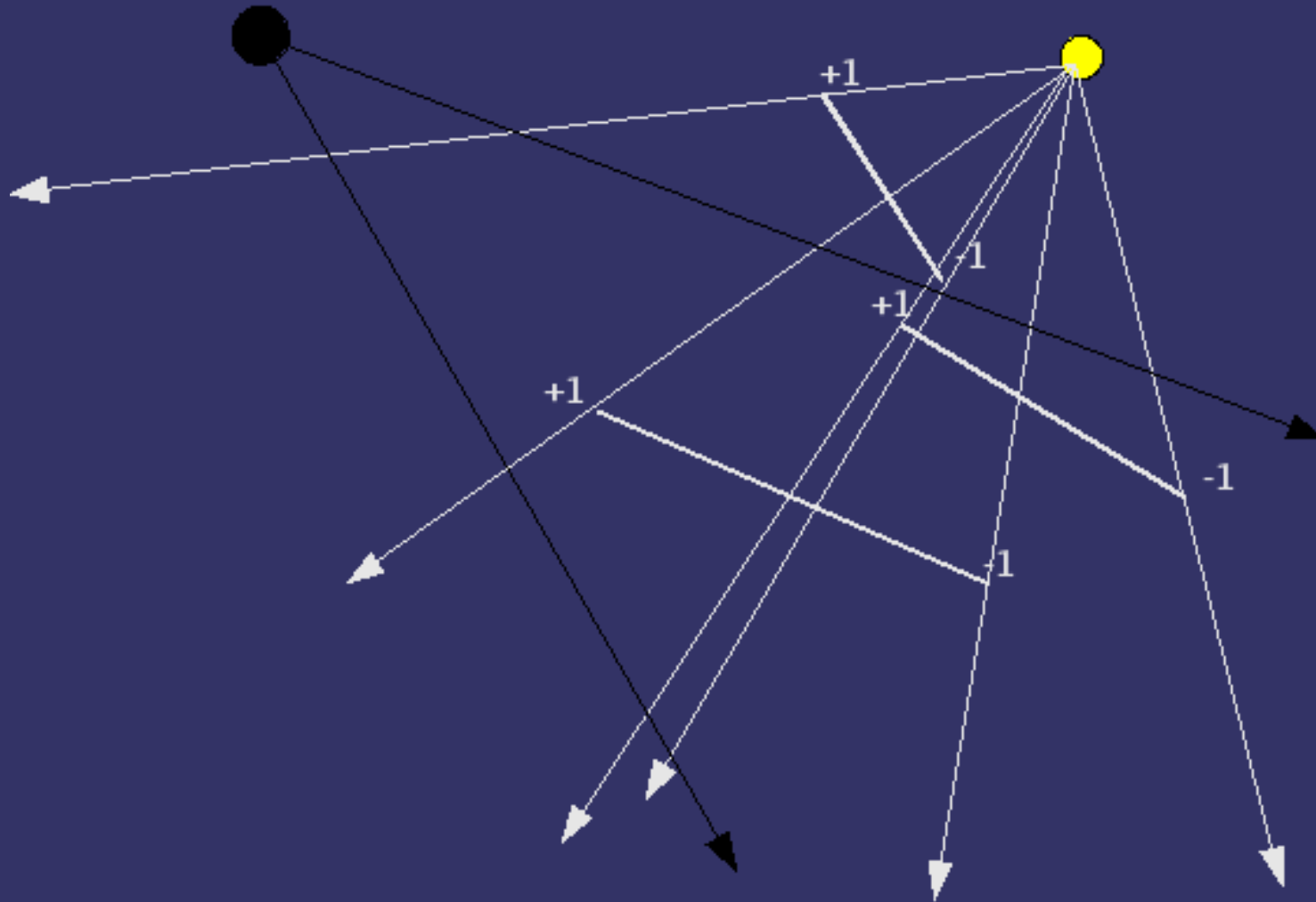
Shadow volume problems

- ⇒ *Very* fill-rate intensive
- ⇒ Calculating shadow volumes can be complex and time consuming
- ⇒ Difficult to extend to soft-shadows

Z-pass overview

1. Disable depth and color writes
2. Enable back-face culling
3. Use `GL_INCR_WRAP` on depth pass
4. Draw shadow volumes
5. Enable front-face culling
6. Use `GL_DECR_WRAP` on depth pass
7. Draw shadow volumes

Z-pass (cont.)



Z-pass' big problem

➔ What if the camera is inside a shadow volume?

Solutions

- ⇒ Clear stencil to +1 for each volume the camera is inside
 - Expensive to compute!
- ⇒ Add a “clip” at the near plane for each volume the camera is inside
 - Expensive to compute!
- ⇒ Use Z-fail

Z-fail

1. Disable depth and color writes
2. Enable front-face culling
3. Use `GL_INCR_WRAP` on depth fail
4. Draw shadow volumes
5. Enable back-face culling
6. Use `GL_DECR_WRAP` on depth fail
7. Draw shadow volumes

Z-fail's big problem

➔ US Patent #6,384,822

Creating shadow volume geometry

- ➔ Two passes over object geometry are required:
 - Each edge that is shared by a front-facing polygon and a back-facing polygon, it is on the silhouette.
 - Project each edge on the silhouette away from the light to “infinity”. Create a new quad using these two edges. Add this quad to the shadow volume.
 - Add each back-facing polygon to the volume.
 - Project each back-facing polygon away from the light to infinity and add it *again*.

Shadow volume creation problems

- ⇒ New volume must be created each time the object or the light move
- ⇒ Time consuming and must be performed on the CPU
 - Re-upload data to the GPU *each frame!*
- ⇒ Bad interactions with vertex shaders
- ⇒ We'll see how to resolve these issues next week!

Questions?

Legal Statement

- ➔ This work represents the view of the authors and does not necessarily represent the view of IBM or the Art Institute of Portland.
- ➔ OpenGL is a trademark of Silicon Graphics, Inc. in the United States, other countries, or both.
- ➔ Khronos and OpenGL ES are trademarks of the Khronos Group.
- ➔ Other company, product, and service names may be trademarks or service marks of others.