

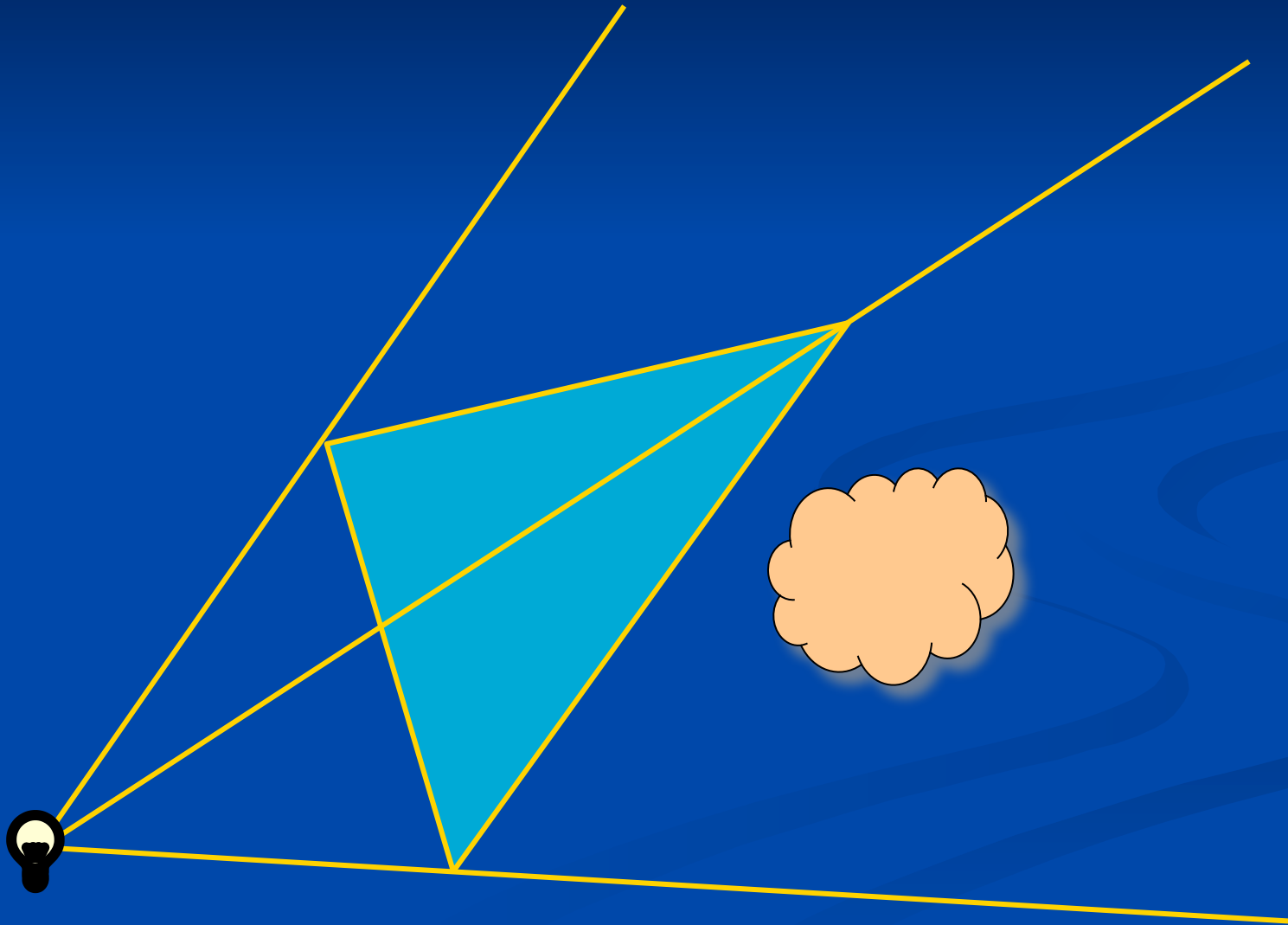
# Computer Graphics

Subodh Kumar

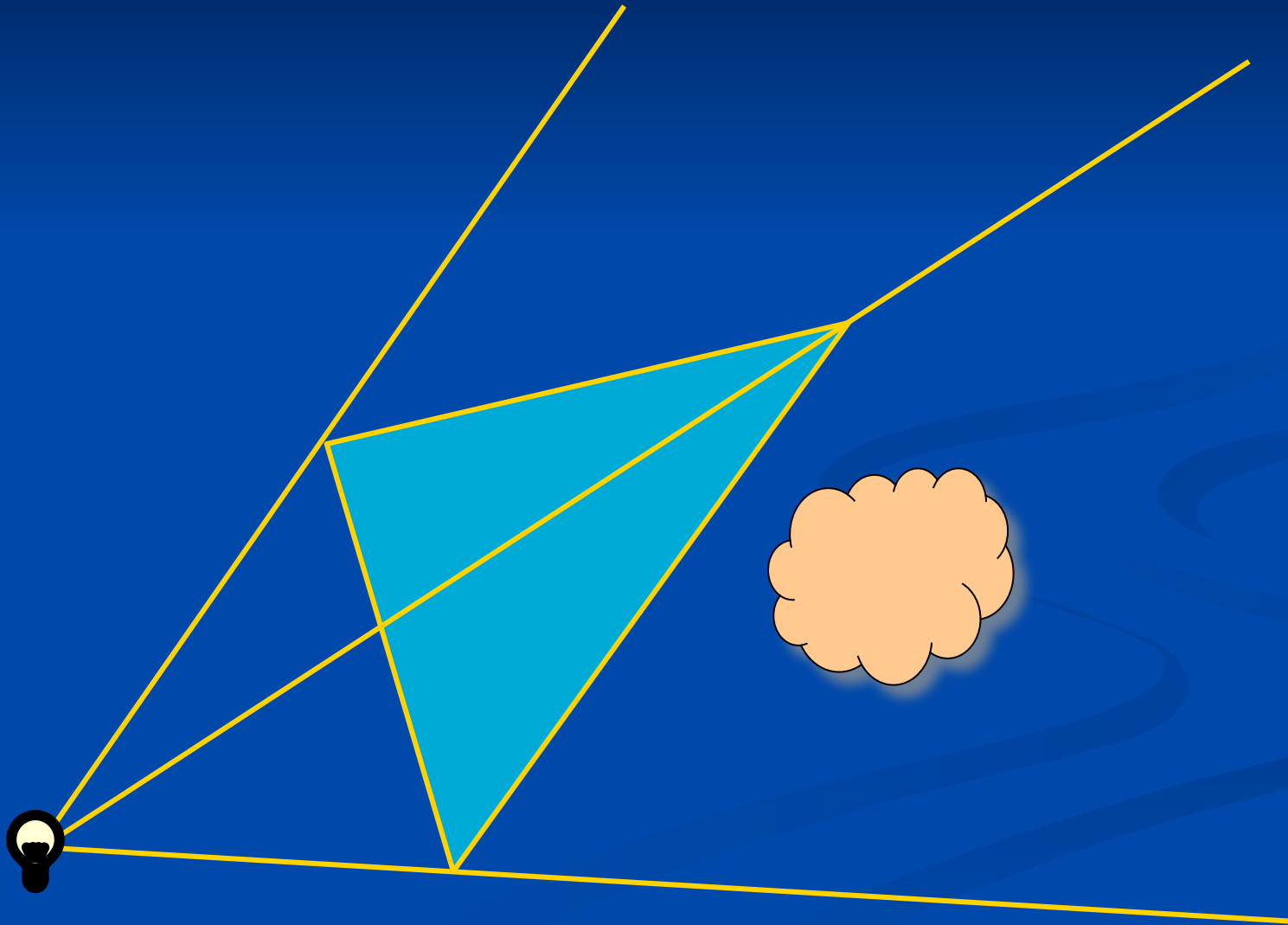
Dept of Computer Sc. & Engg.

IIT Delhi

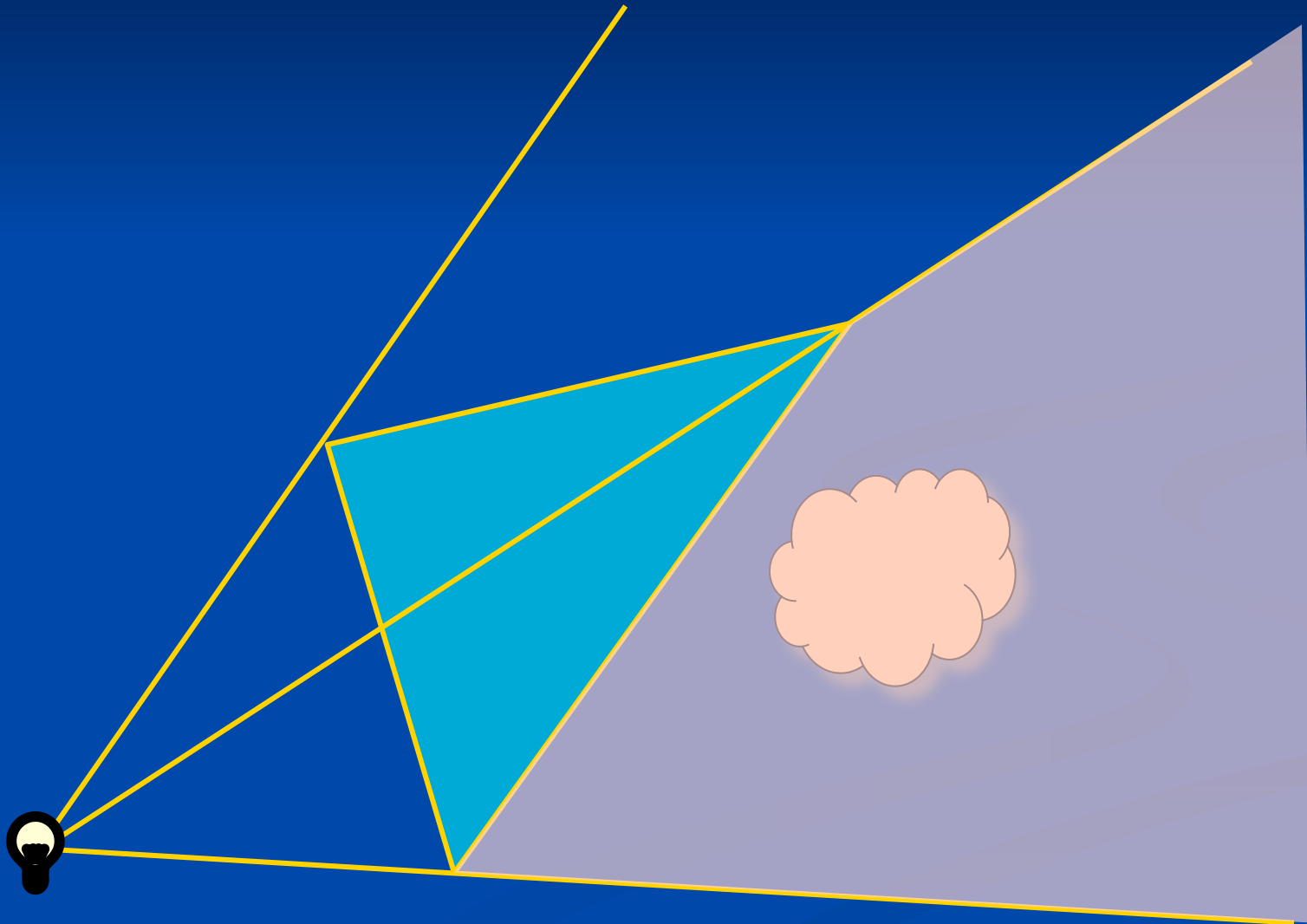
# Shadow Volumes



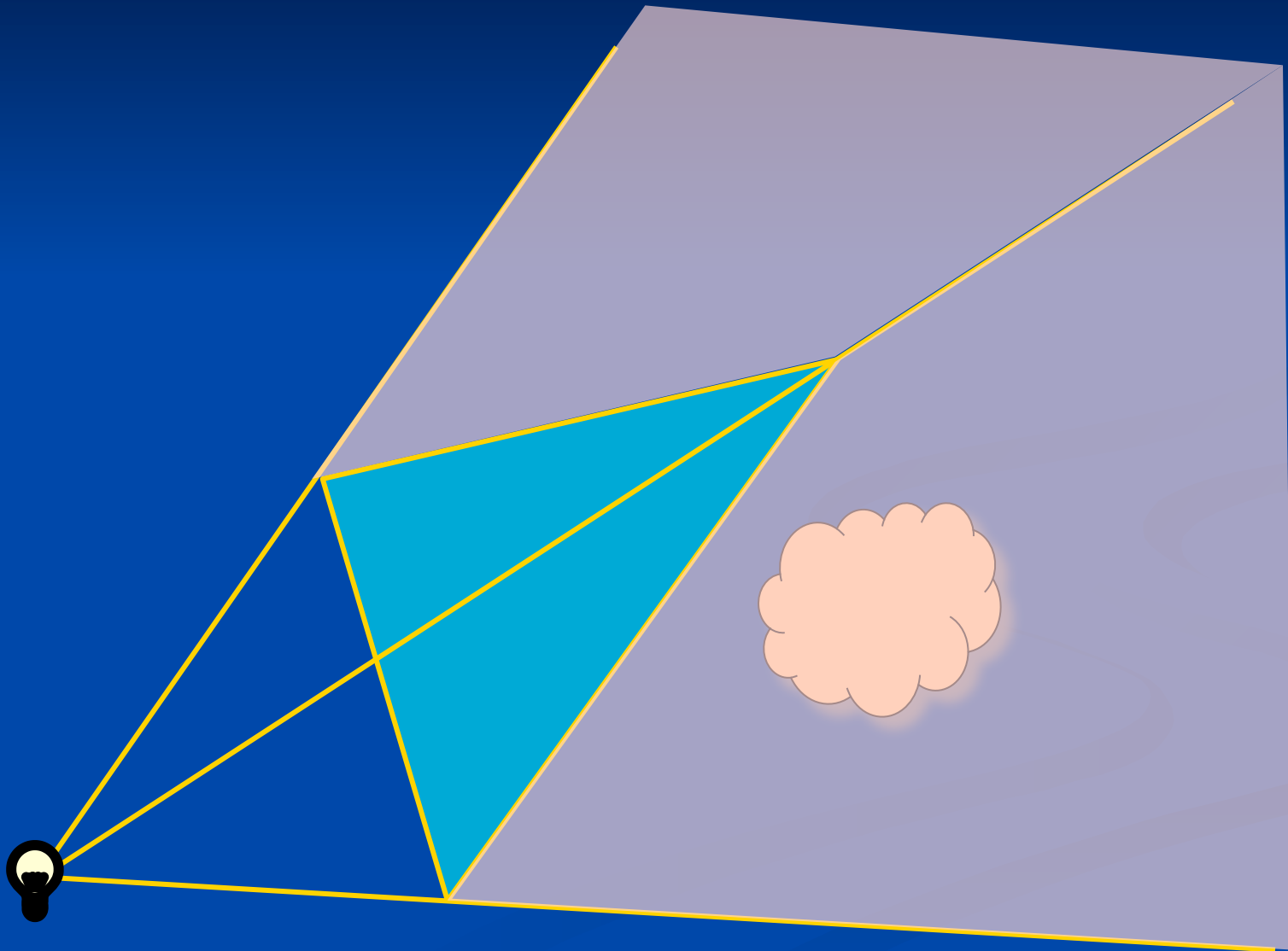
# Shadow Volumes



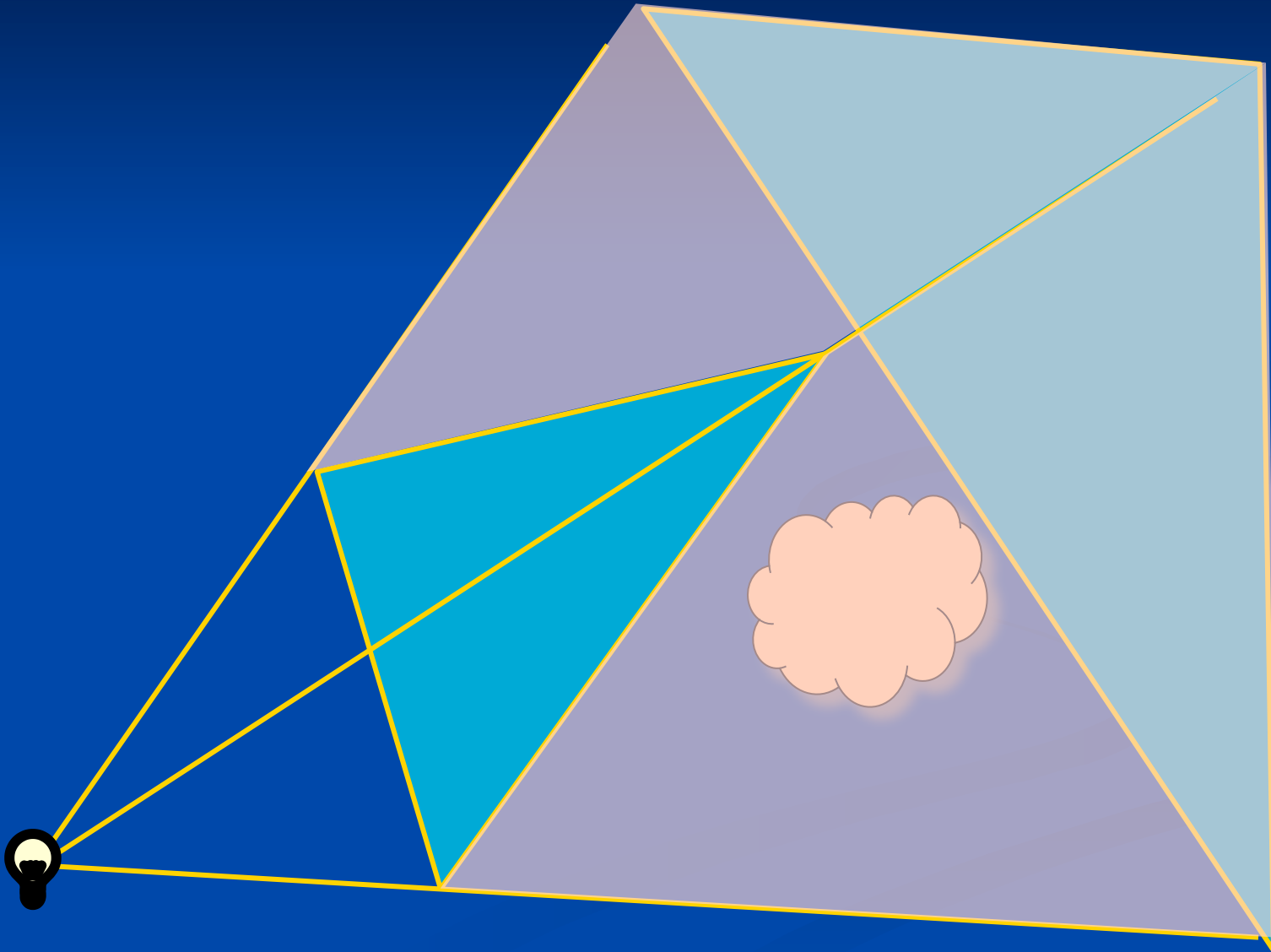
# Shadow Volumes



# Shadow Volumes



# Shadow Volumes



# Stencil Shadow-Volume Algorithm



- Render the scene with shadow color
- For each light source
  - Using the depth from pass 1, construct a mask in the stencil buffer enabling pixels not in shadow
    - Use Shadow volume constructed by computing silhouettes of occluders
      - And capping the volumes
  - Render the scene again, lit this time.
    - Stencil buffer masks out shadowed areas



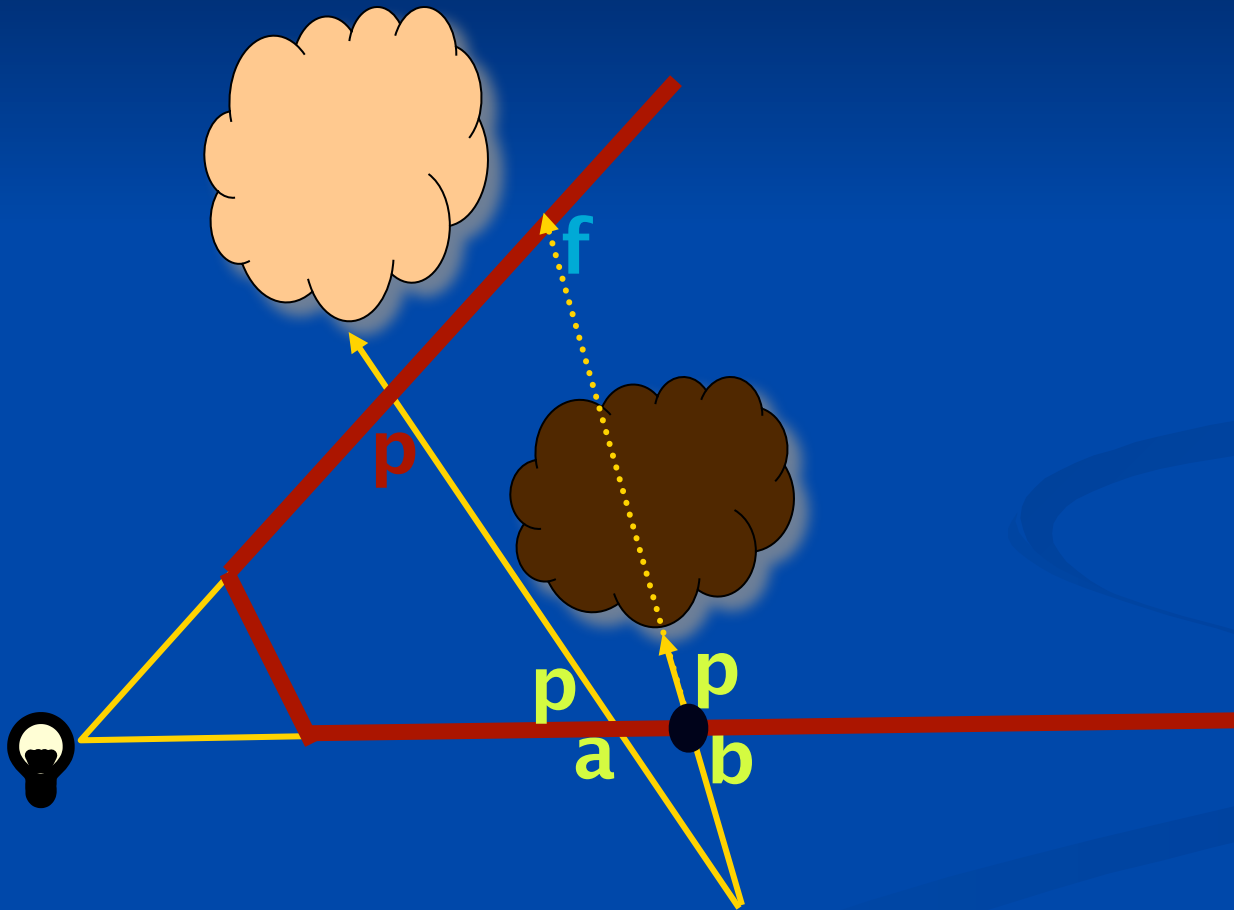
# Stencil Generation

- Disable depth and color writes
- Set the stencil operation to increment on depth pass (counting shadow boundaries in front of the object)
- Draw the shadow volumes front faces
  - Enable back-face culling (ie front drawing)
- Set the stencil operation to decrement on depth pass
- Draw the shadow volume back faces
  - Enable front-face culling

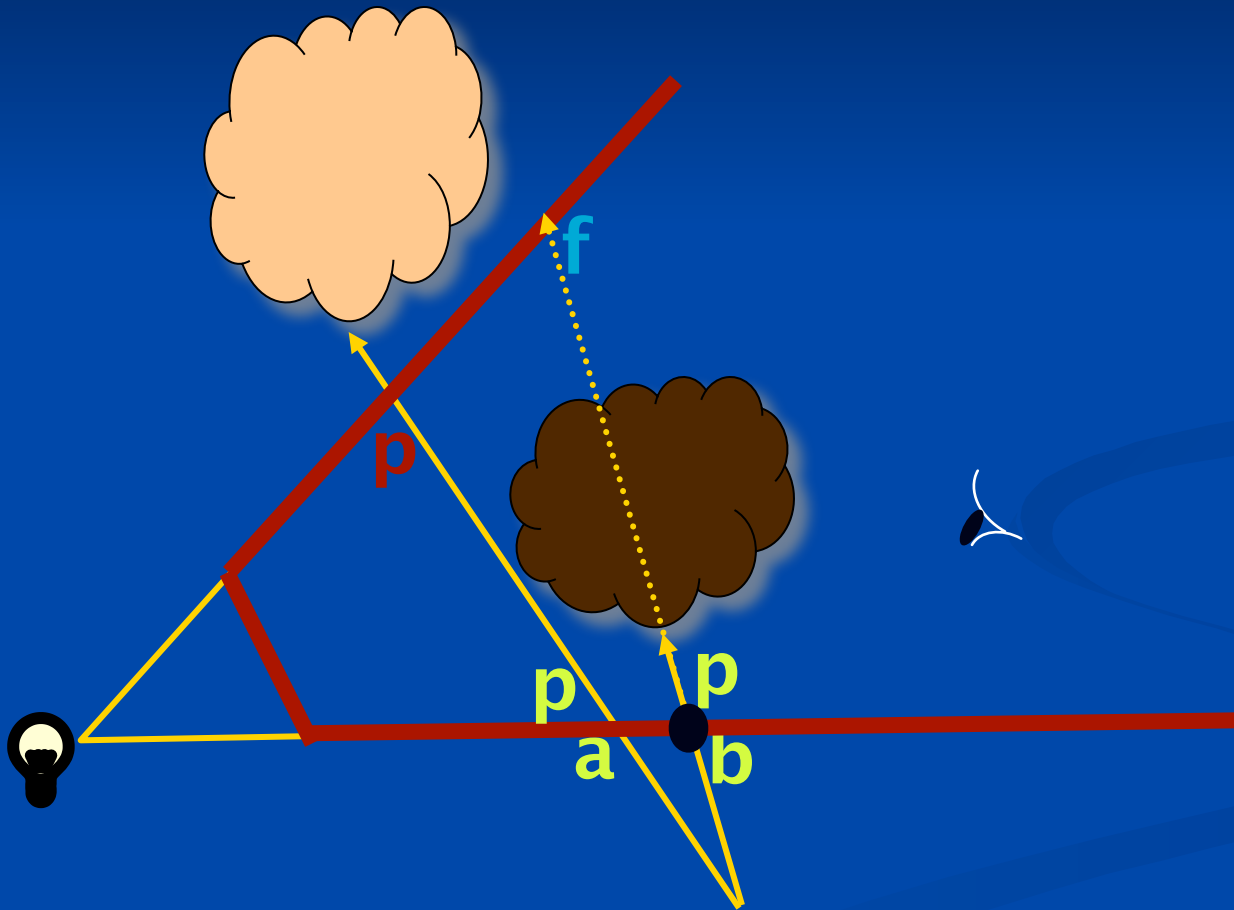




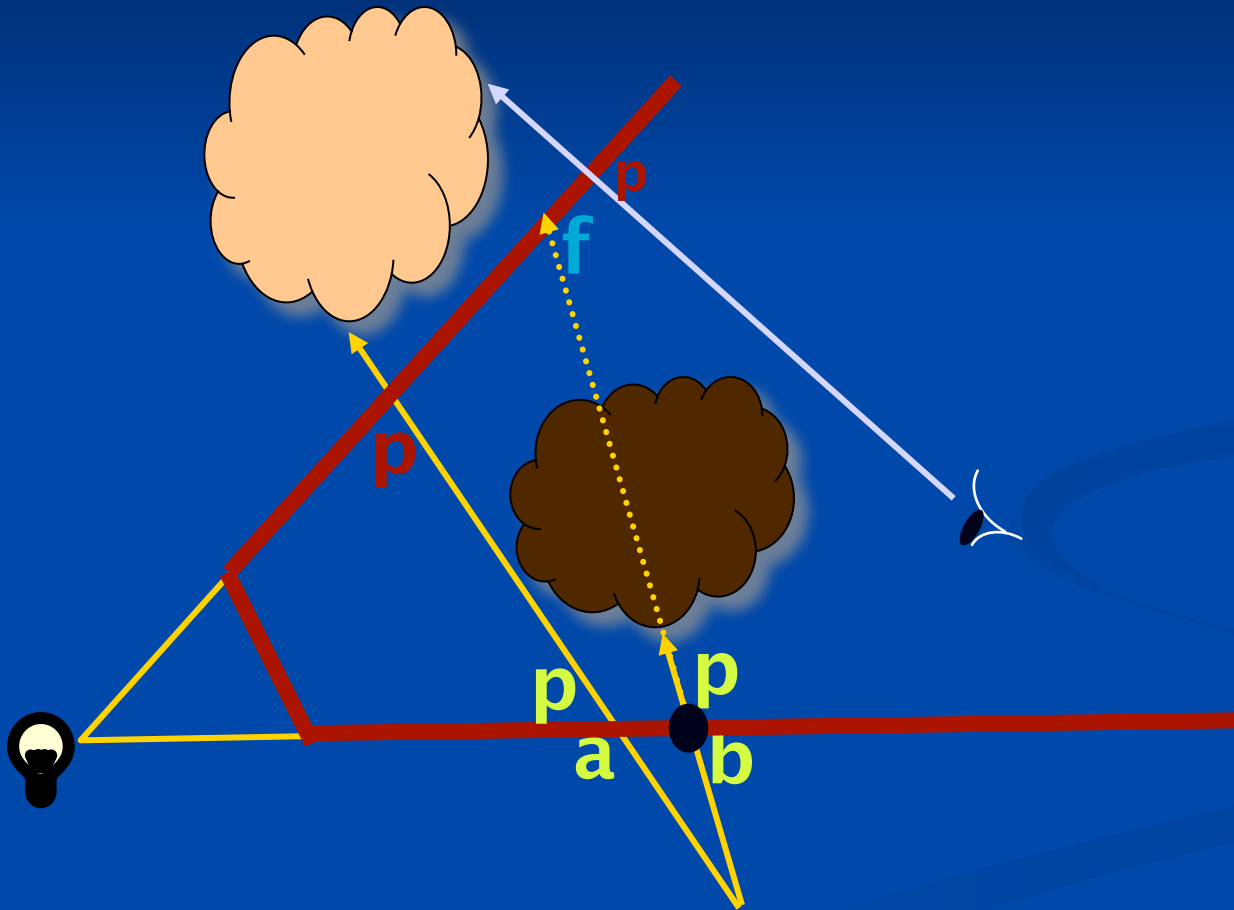
# Stencil Shadow Volumes



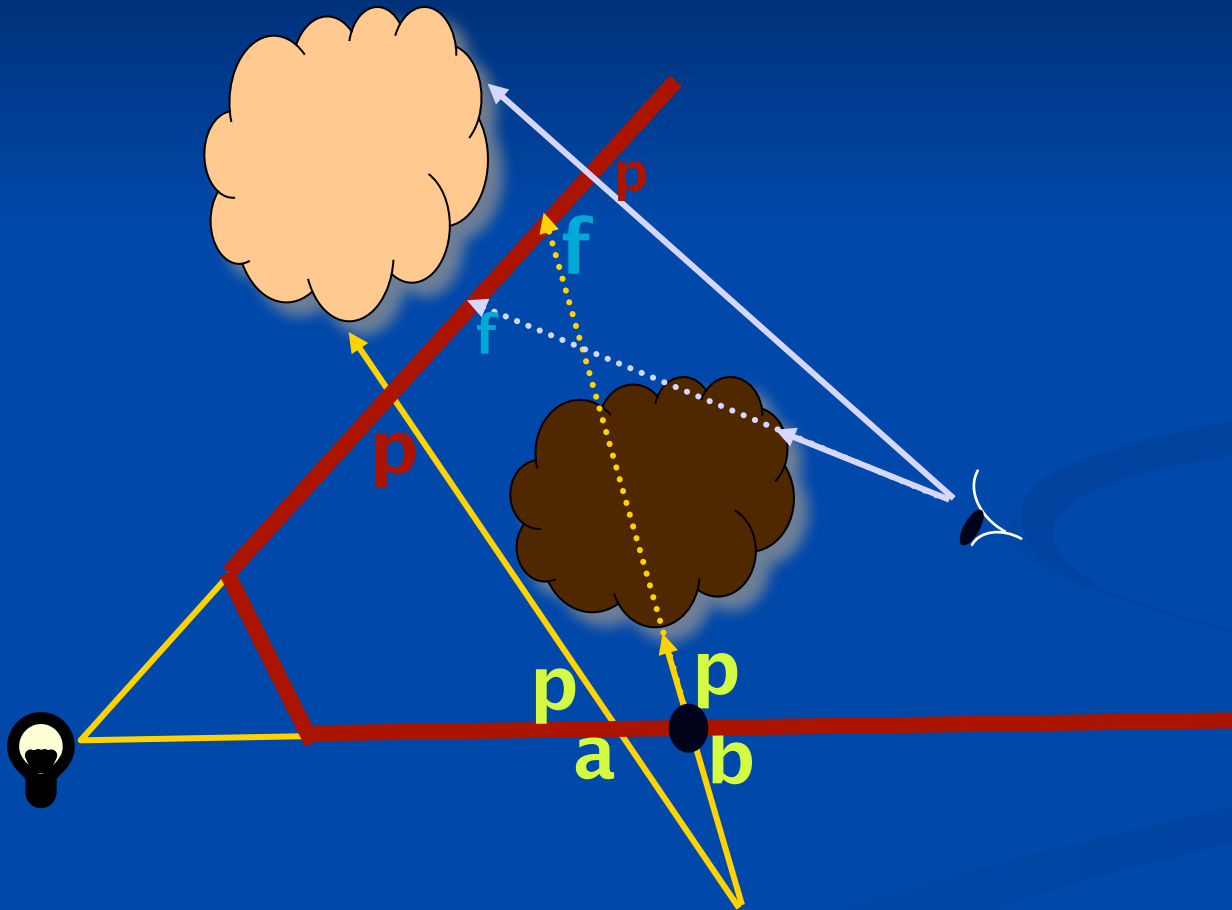
# Stencil Shadow Volumes



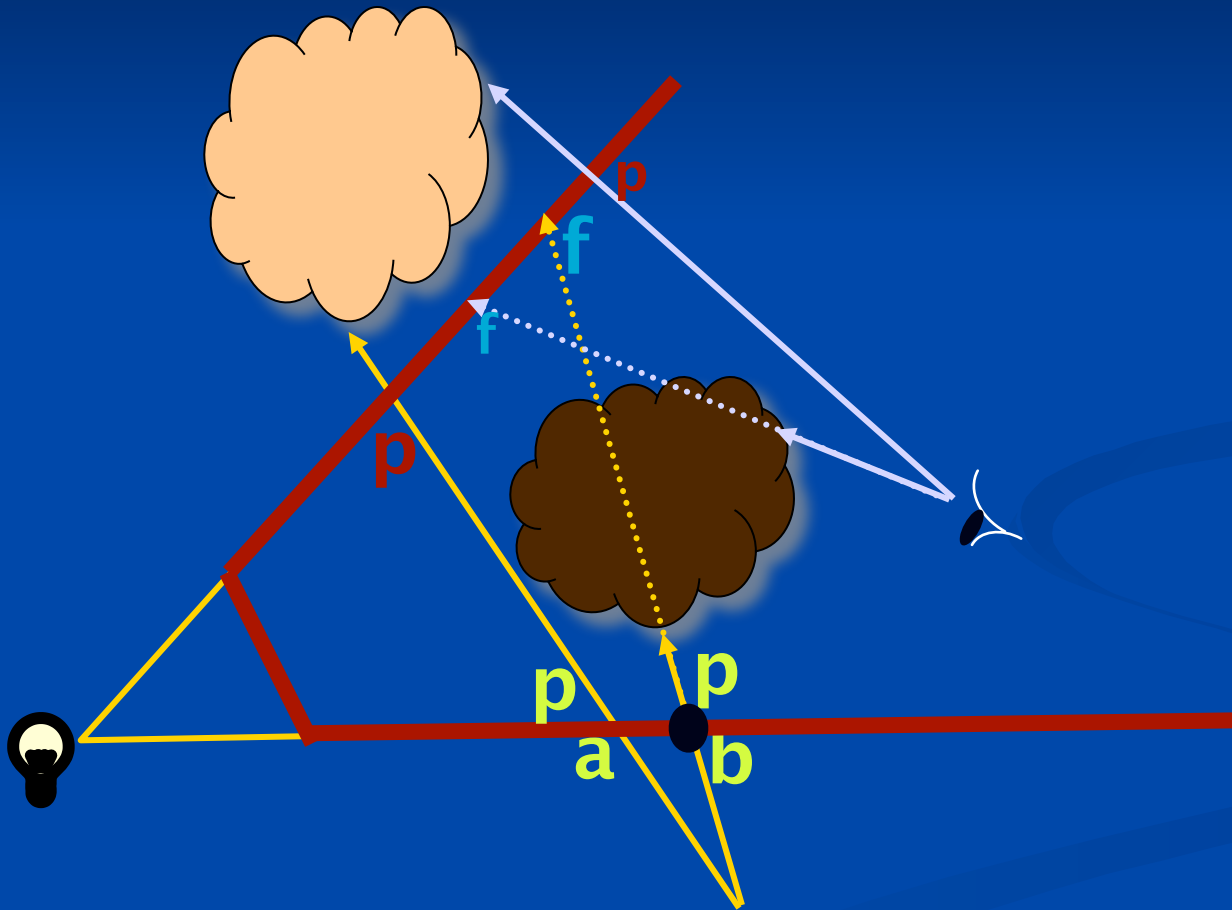
# Stencil Shadow Volumes



# Stencil Shadow Volumes



# Stencil Shadow Volumes



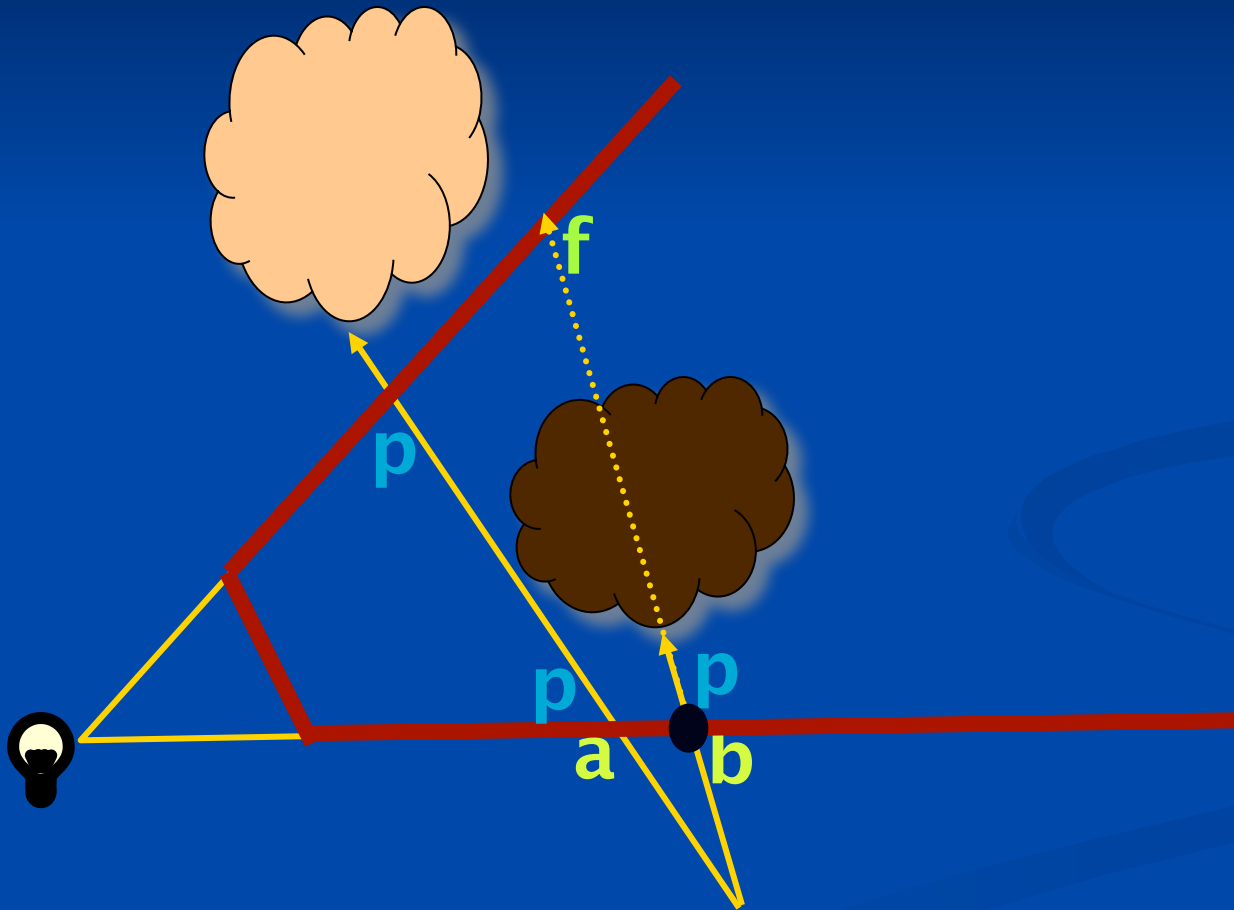
What if Eye is in shadow?



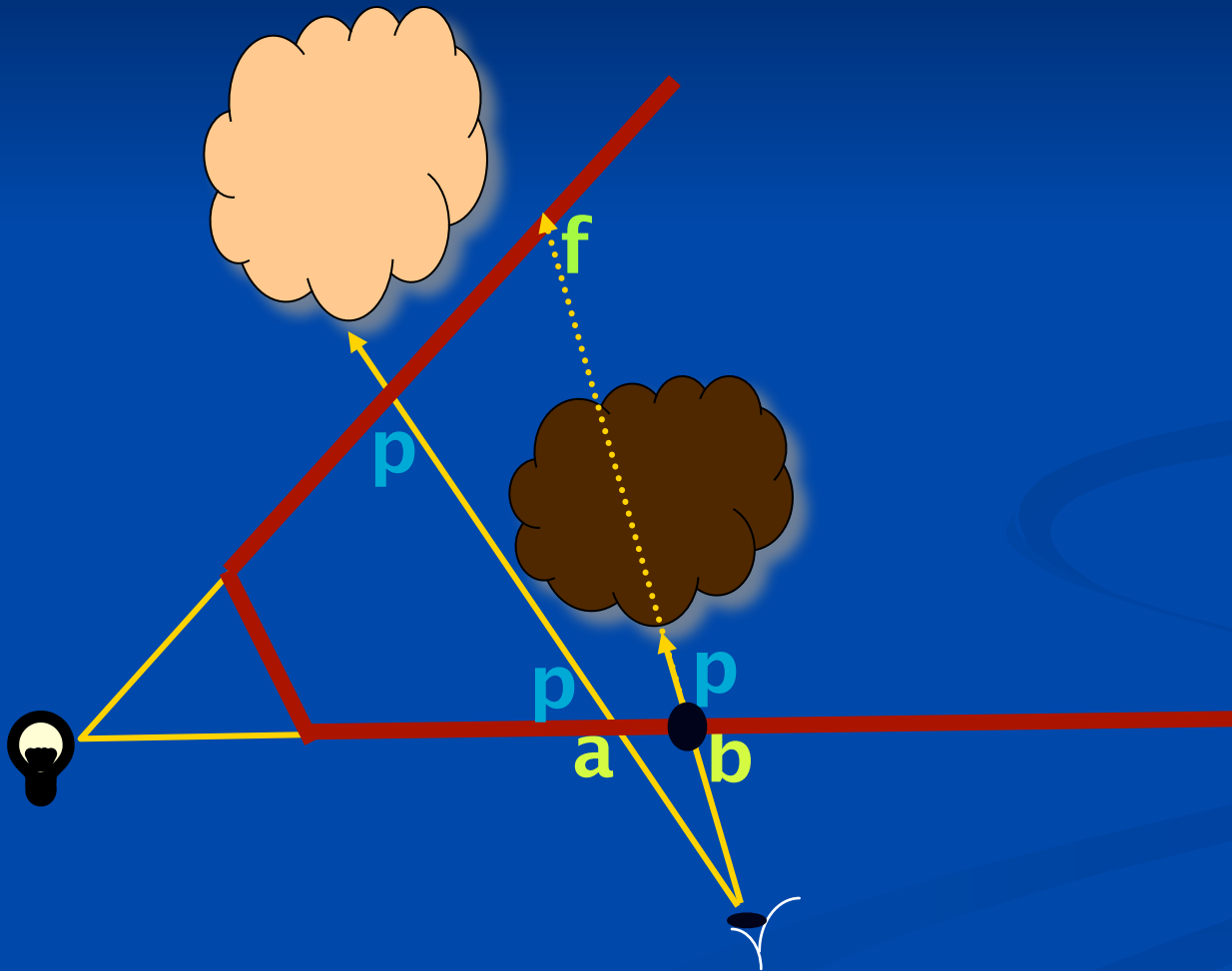
# Reverse Stencil

- Disable depth and color writes
- Set the stencil operation to increment on depth fail (counting shadow boundaries behind the object)
- Render the shadow volumes back faces
- Set the stencil operation to decrement on depth fail.
- Render the shadow volumes front faces

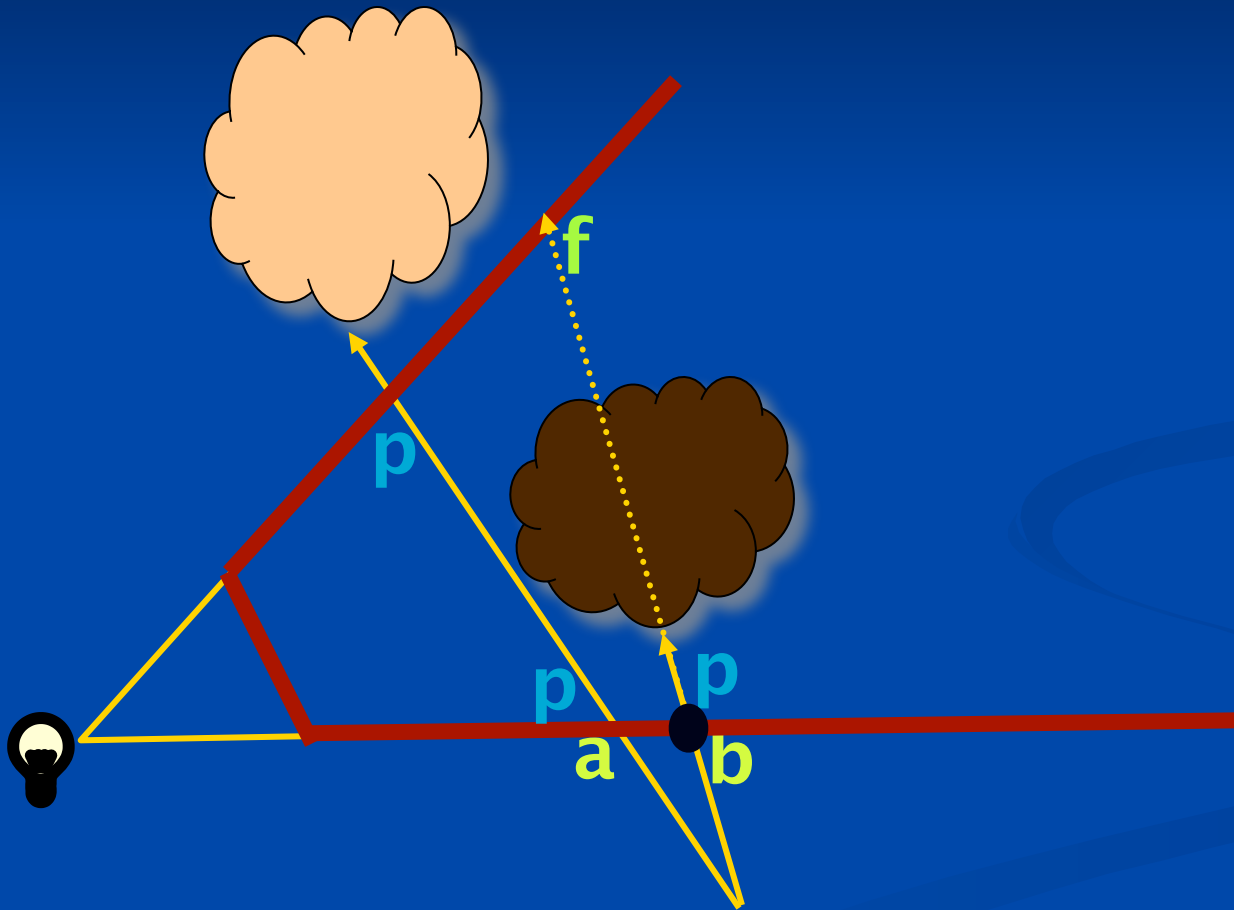
# Stencil Shadow Volumes



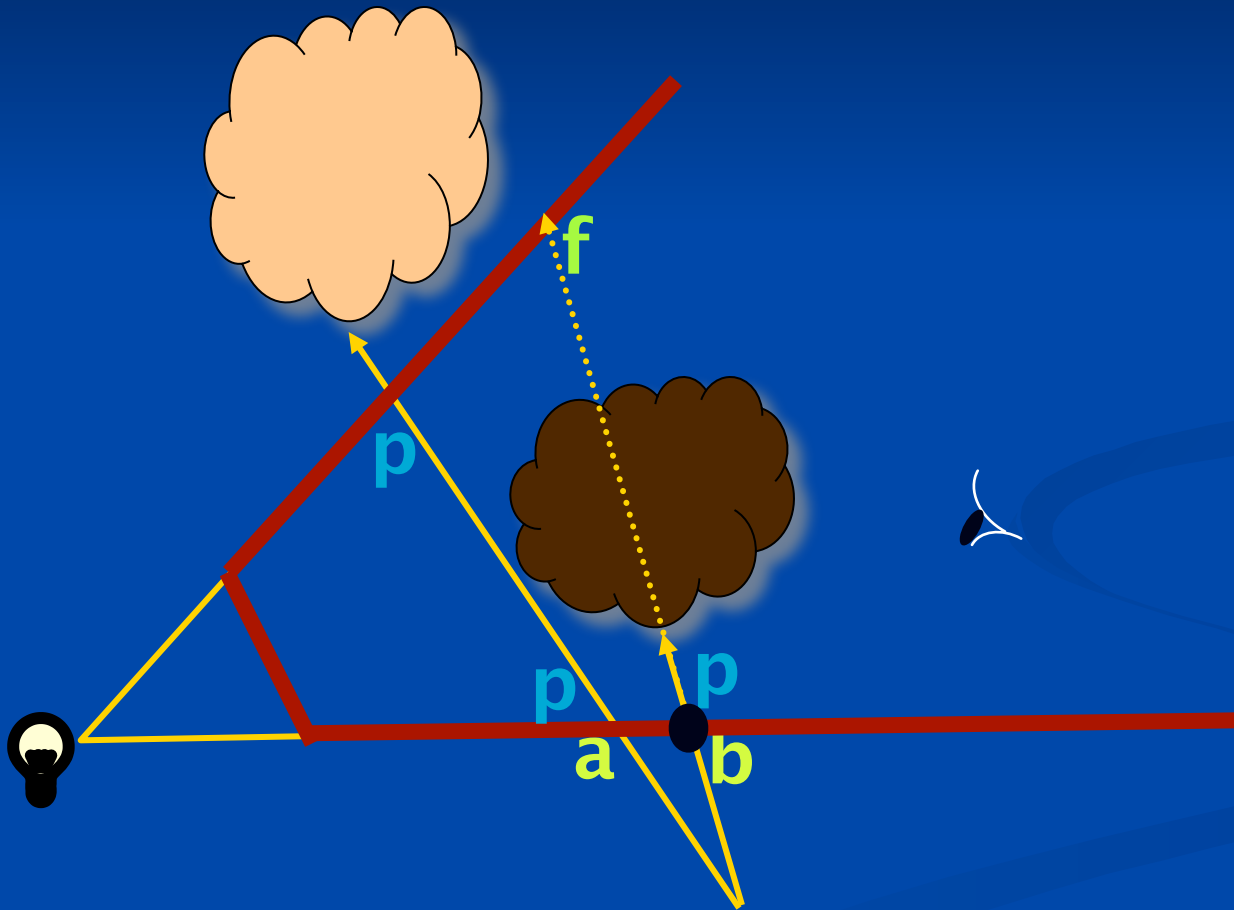
# Stencil Shadow Volumes



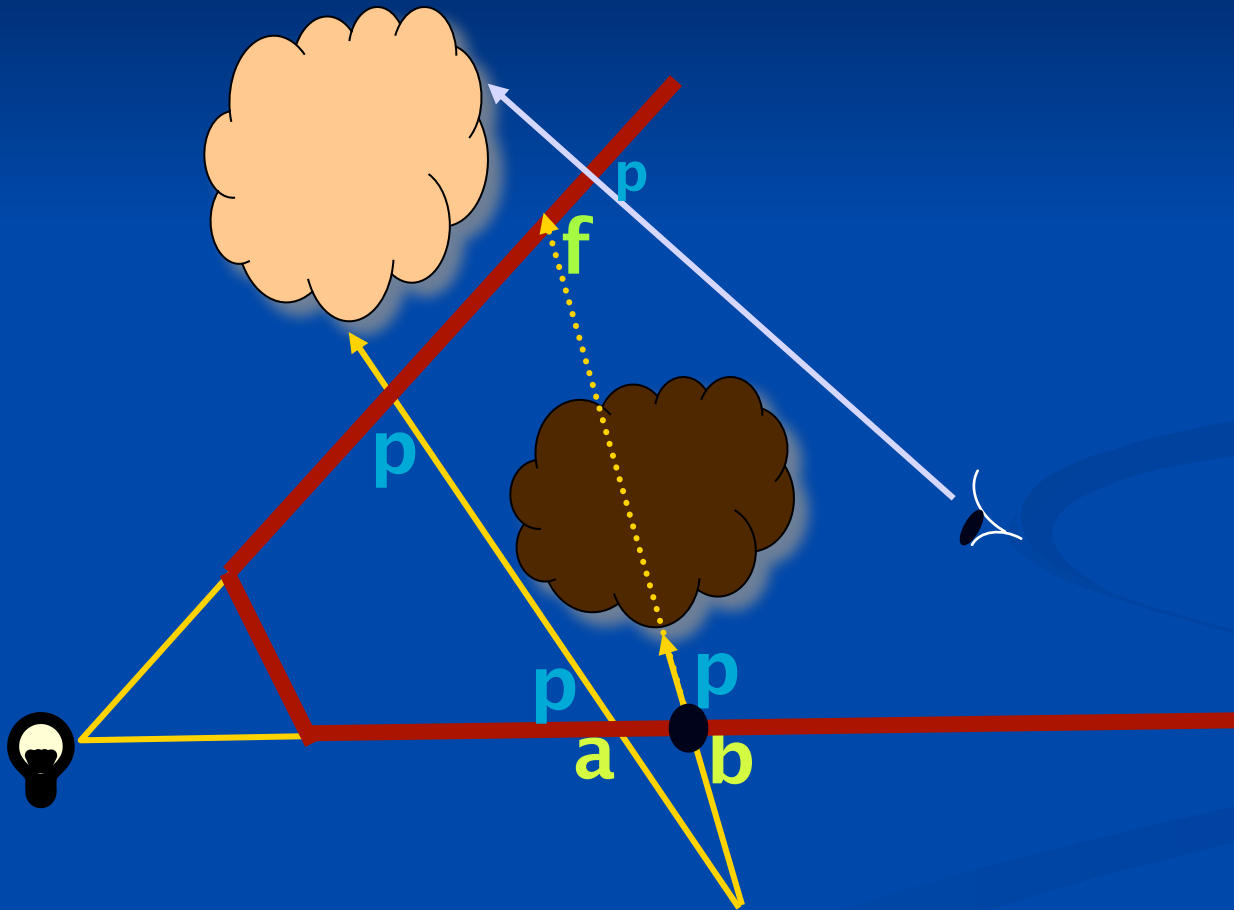
# Stencil Shadow Volumes



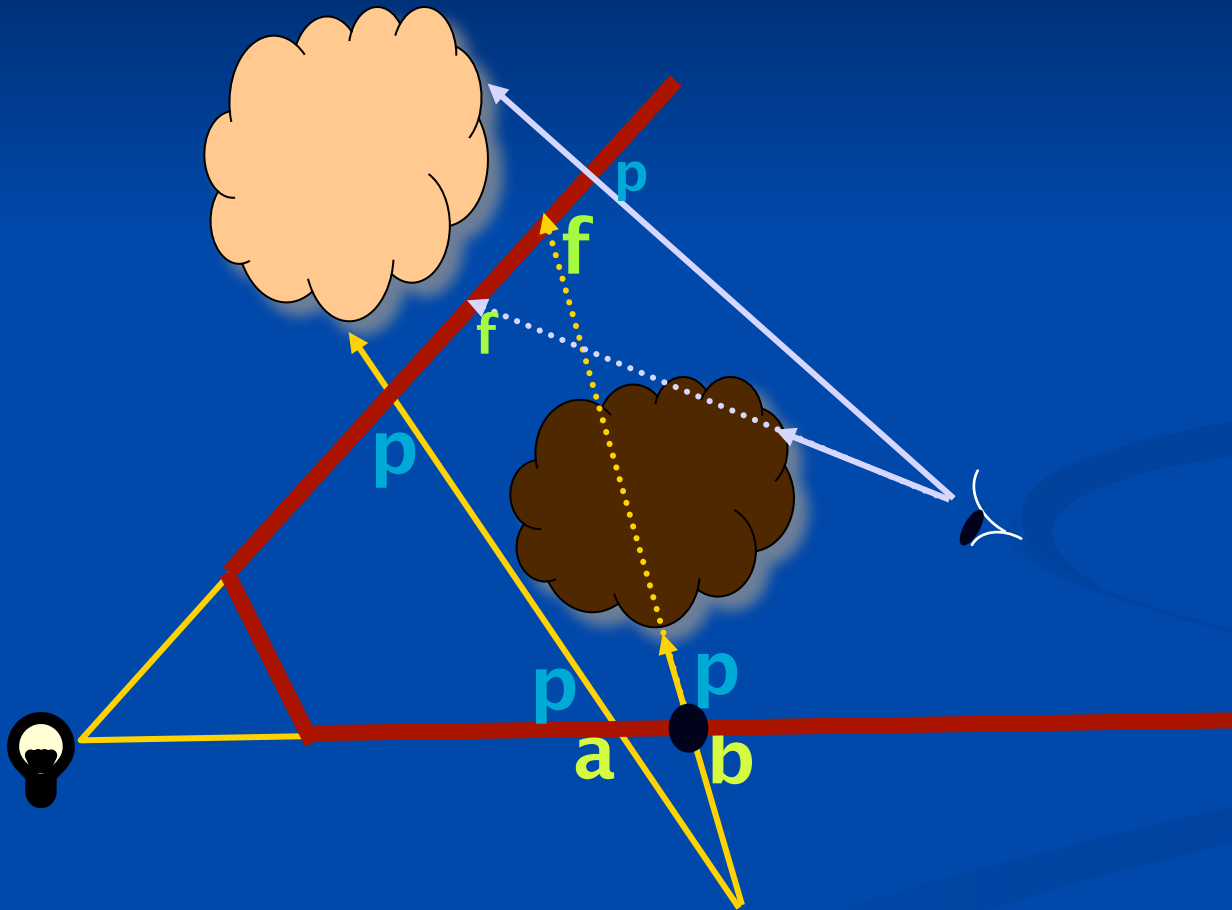
# Stencil Shadow Volumes



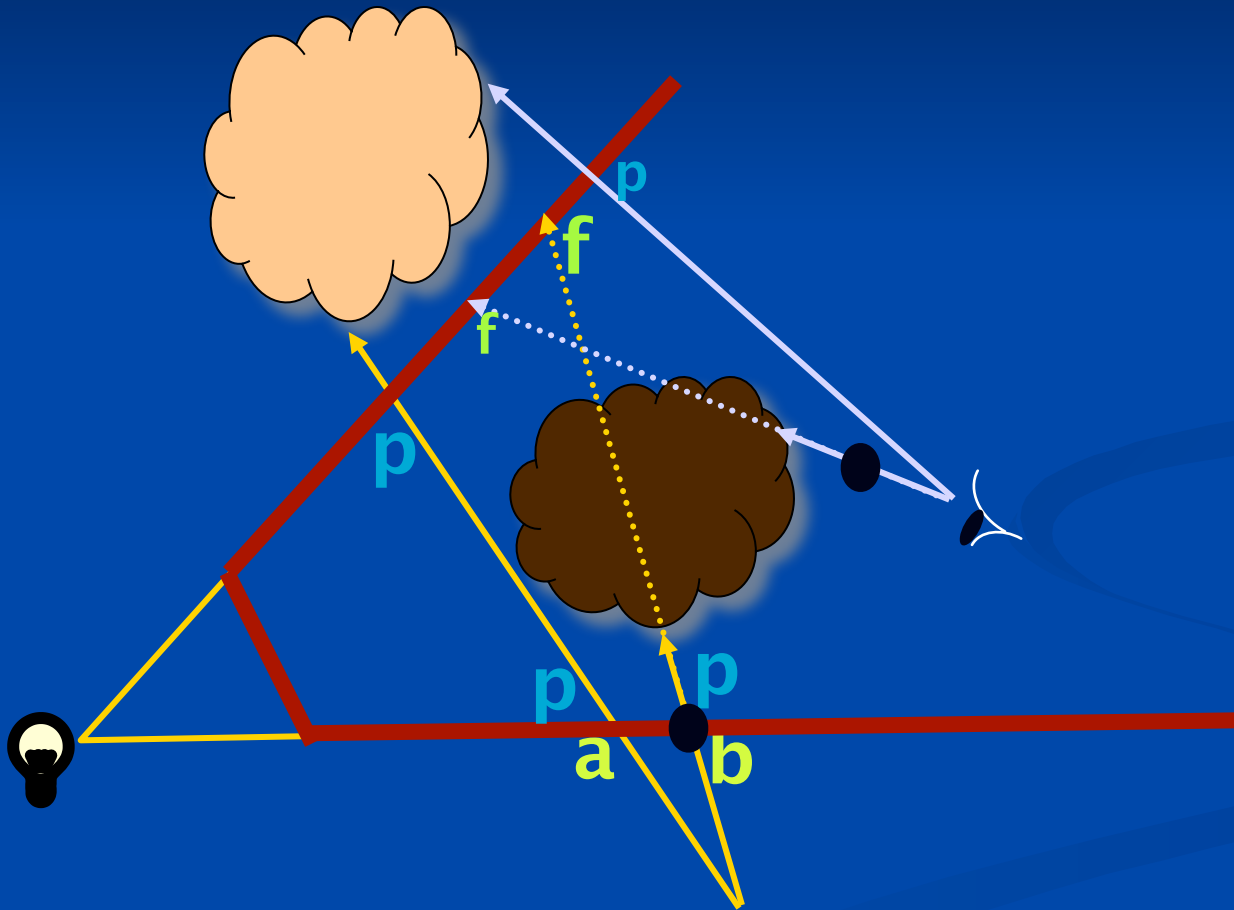
# Stencil Shadow Volumes



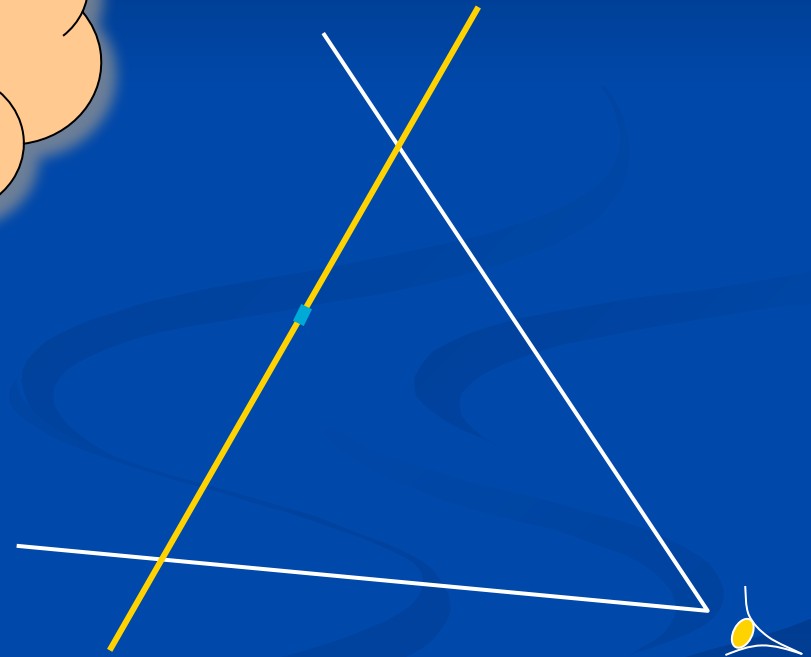
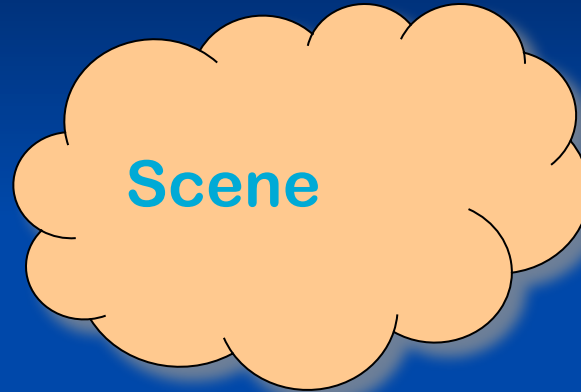
# Stencil Shadow Volumes



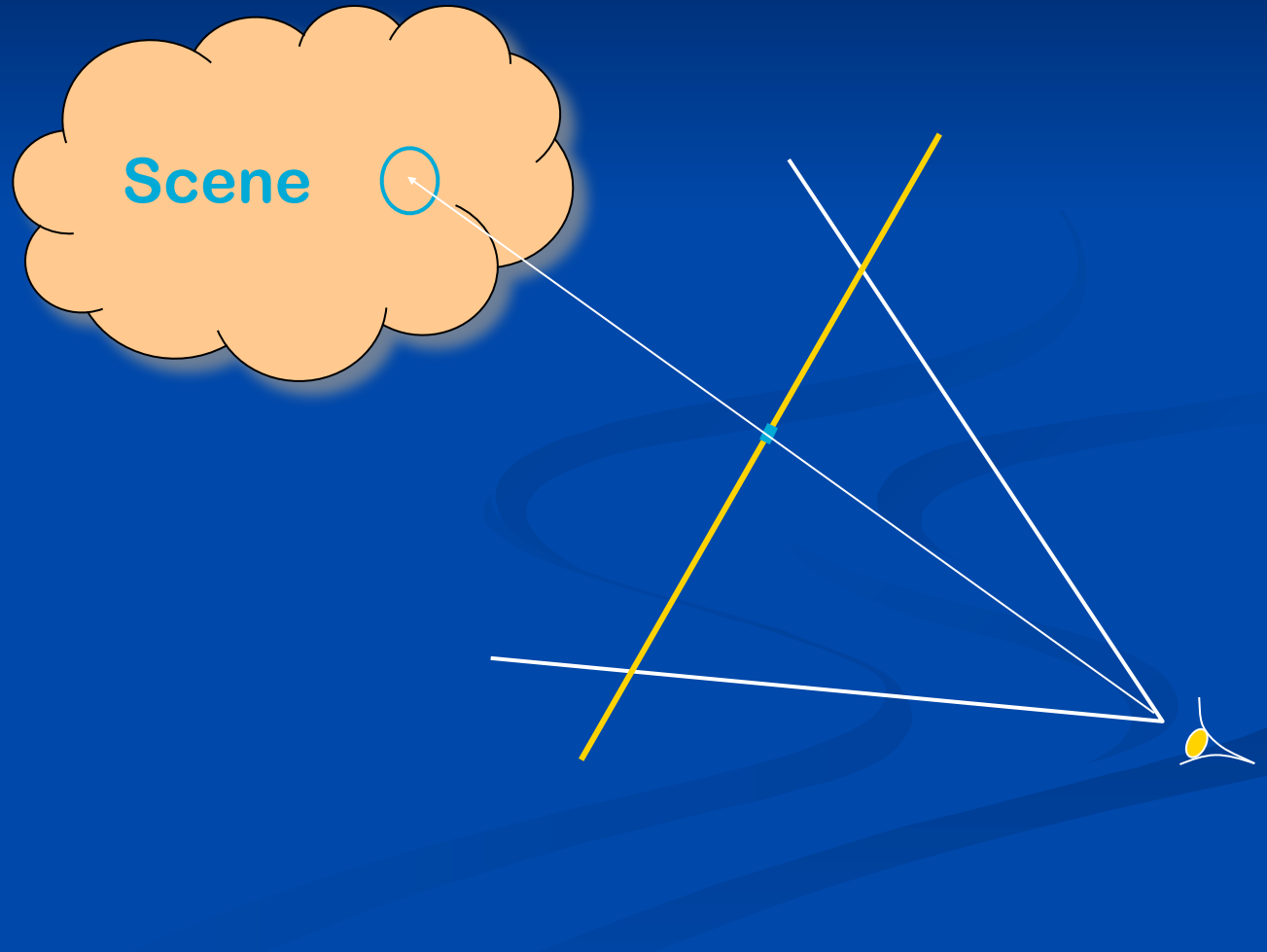
# Stencil Shadow Volumes



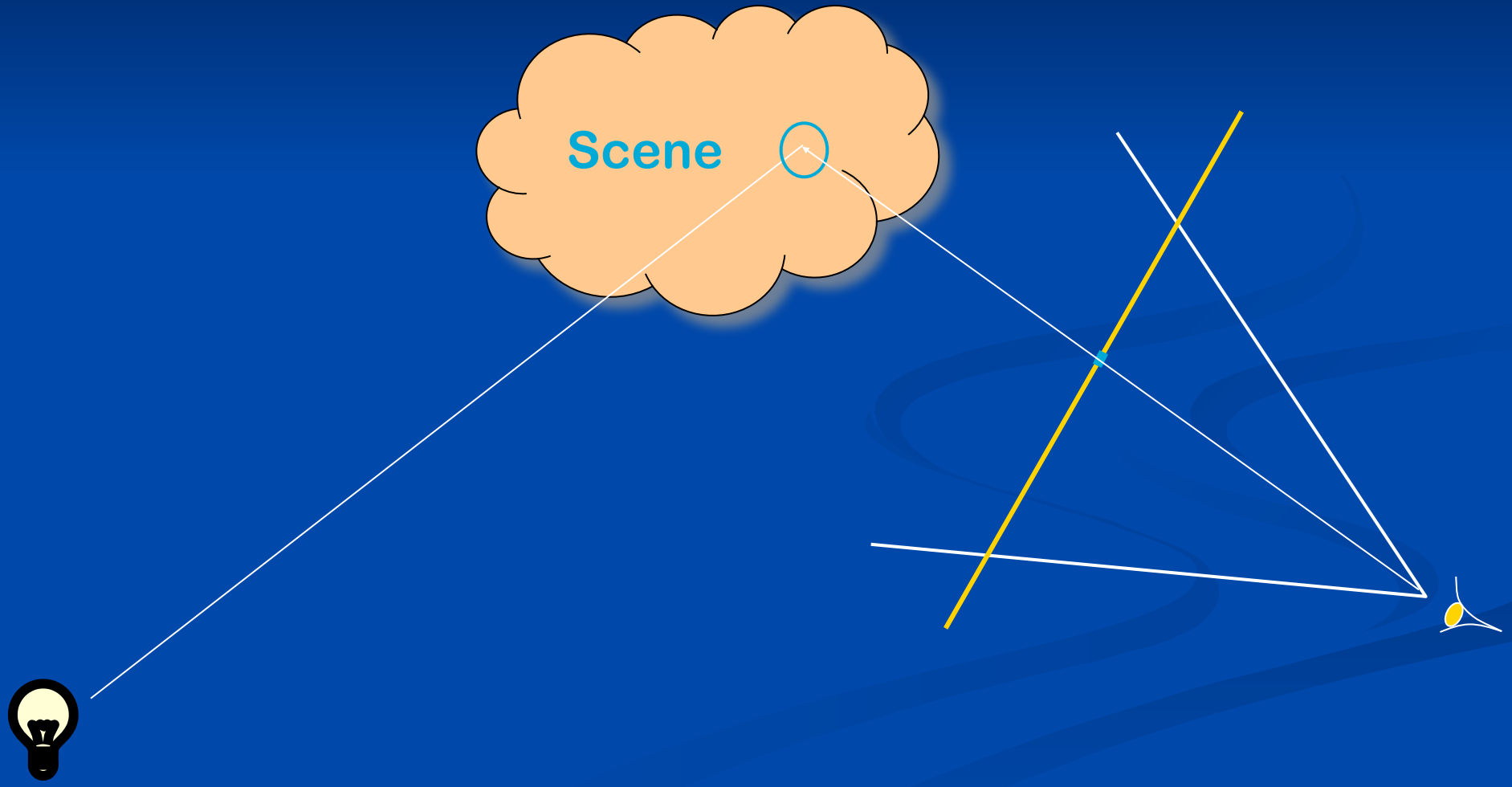
# Shadow Map



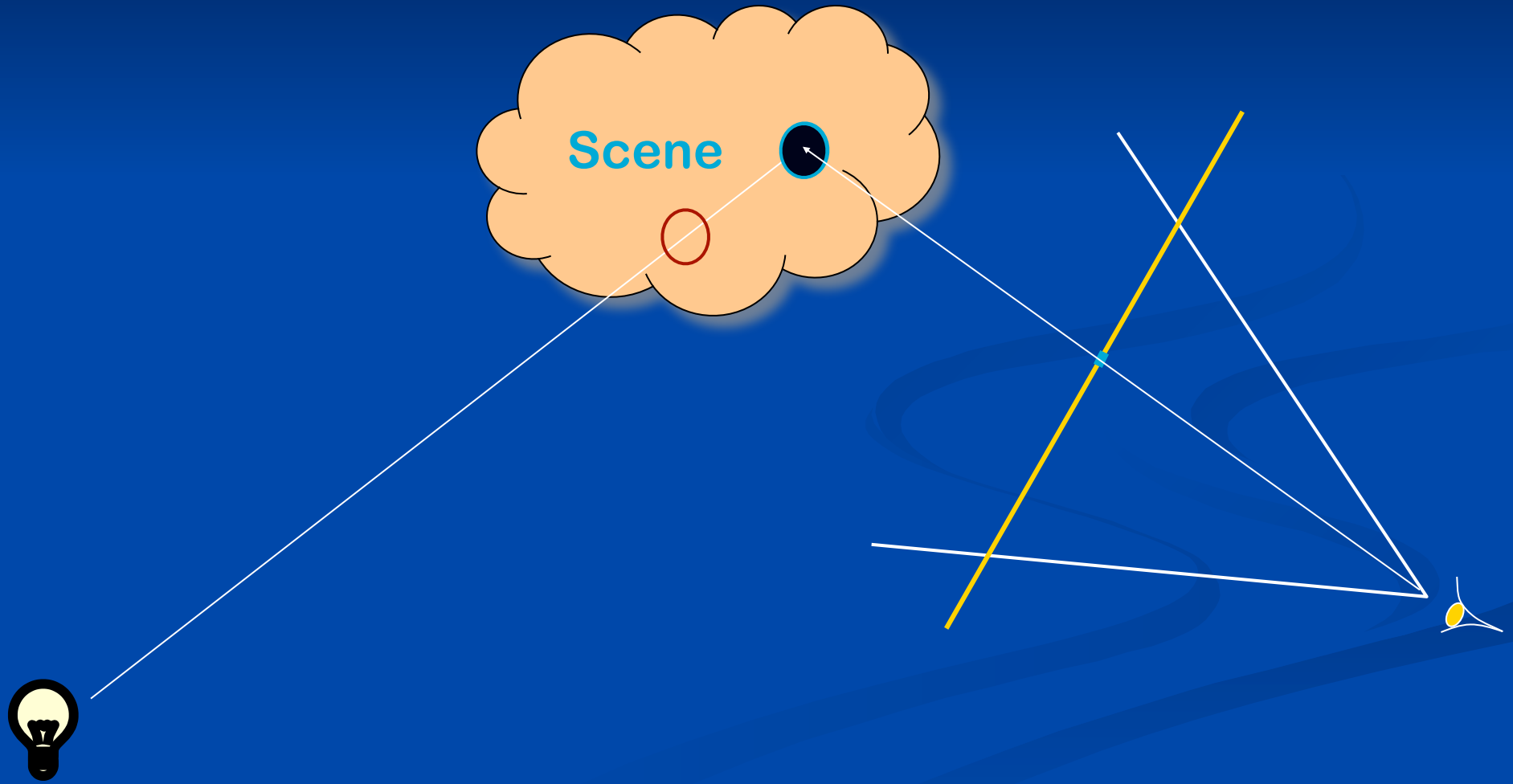
# Shadow Map



# Shadow Map

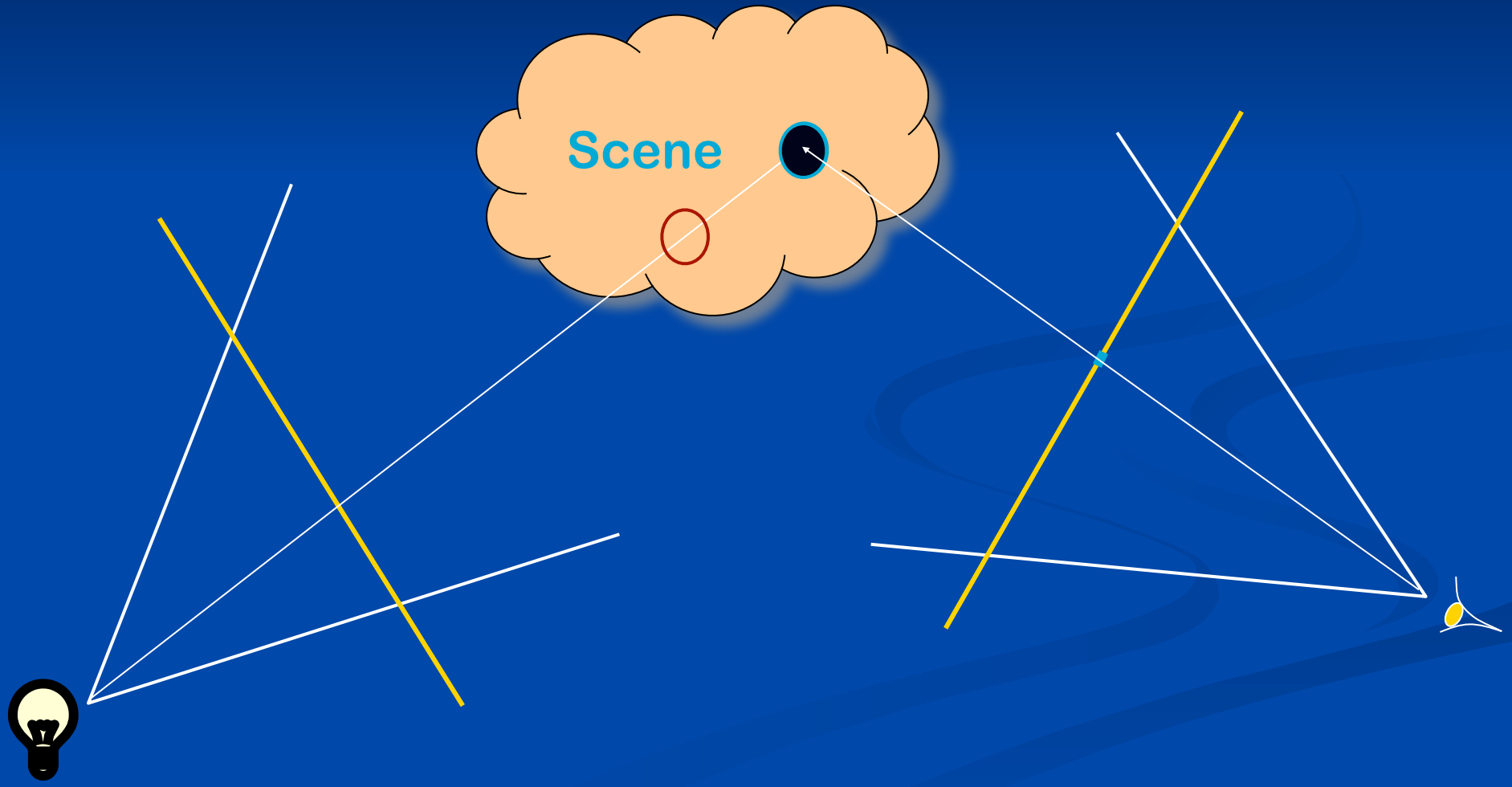


# Shadow Map



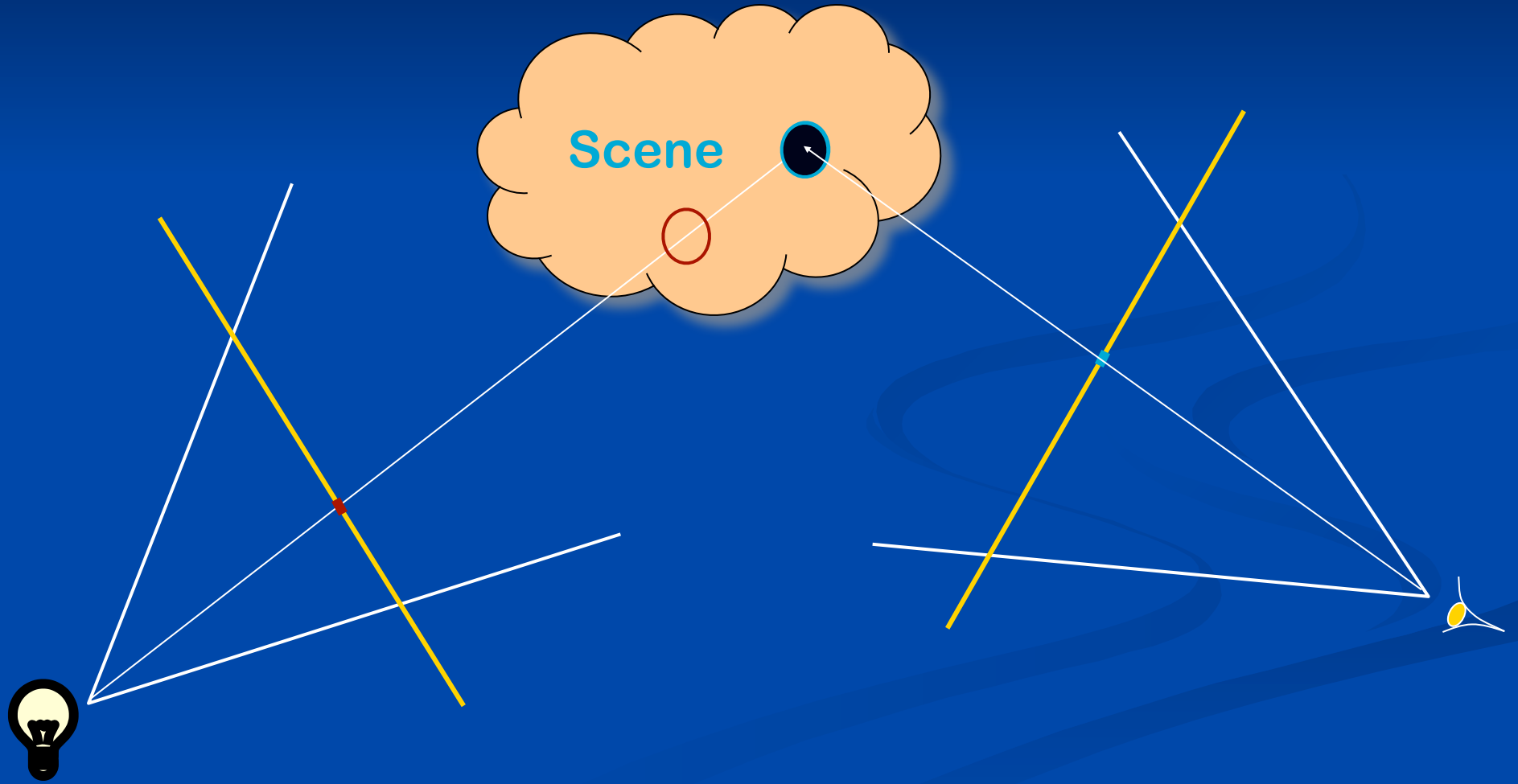


# Shadow Map

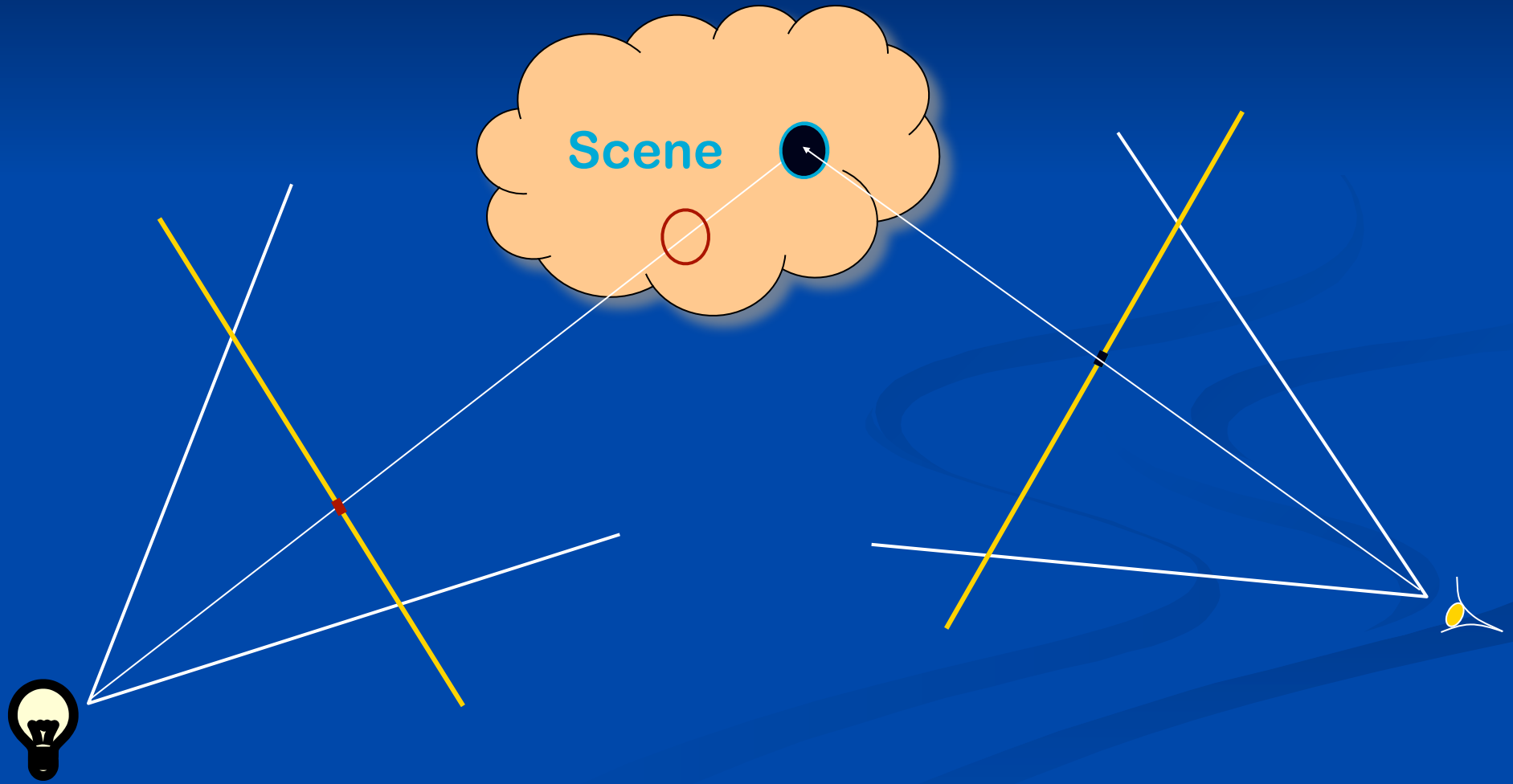




# Shadow Map



# Shadow Map

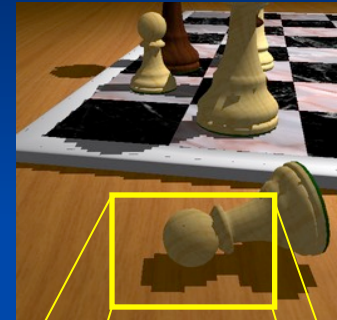




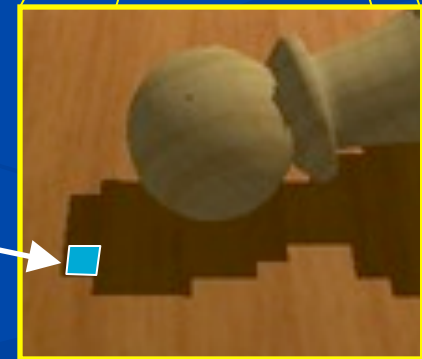
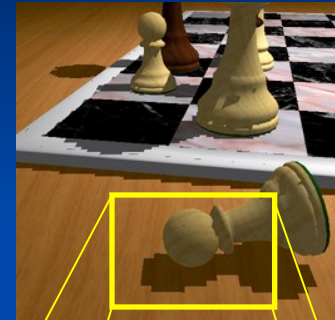
# Shadow Map

- Find  $x, y$  in the light space
- Unproject camera, then Project light
- But  $x$  and  $y$  may not be integers
  - Find nearest integers?
- Read Depth buffer
  - Compare projected  $z$  with stored  $z$ .

# Shadow Map Aliasing



# Shadow Map Aliasing



single shadow map pixel