

CG

an almost factual account of the  
history of computer graphics

a long long time ago ...

before buzz ...

before quake ...

before microsloth windows ...

before most of you were born ...

the world existed without computer graphics

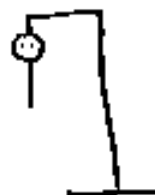




hand-drawn  
graphics

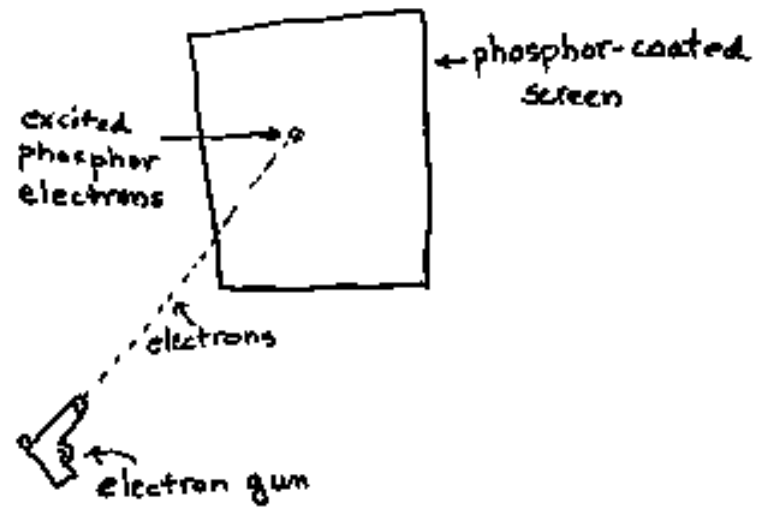


overhead projector game



t \_ \_ s    s v \_ s    b \_ \_

xsz + qvgbv



excited phosphor demo



instant replay

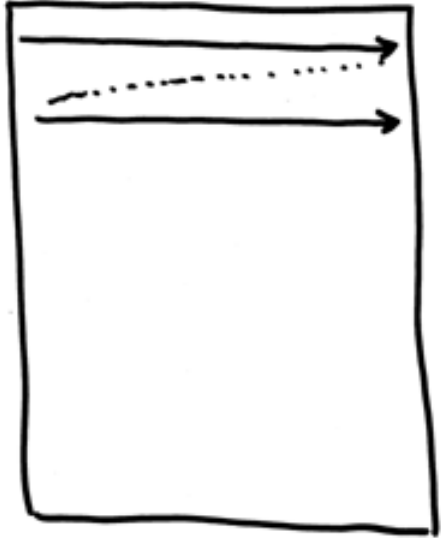


moving electron beam demo



an even less moving  
moving electron beam demo

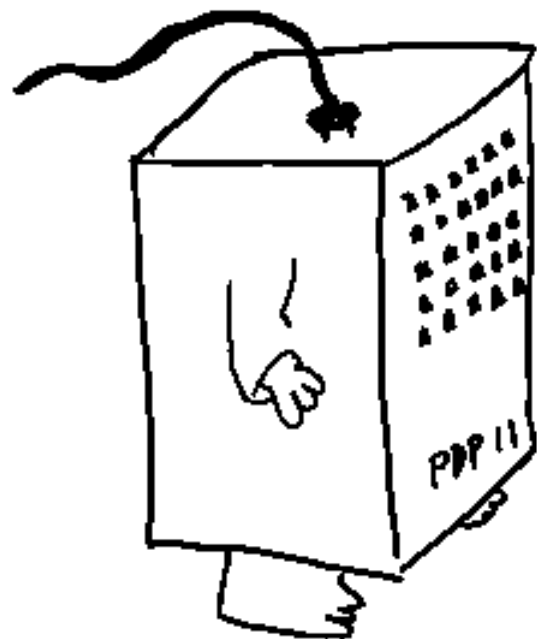


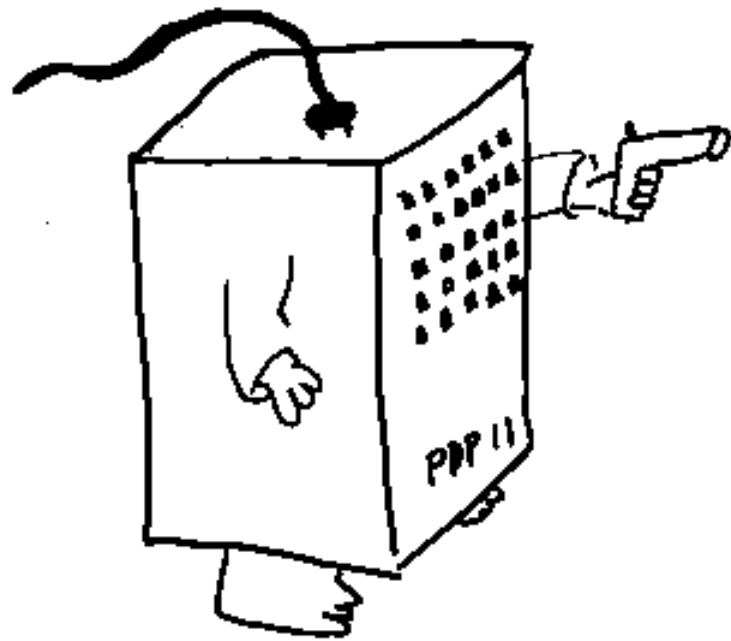


society would reap huge benefits ...

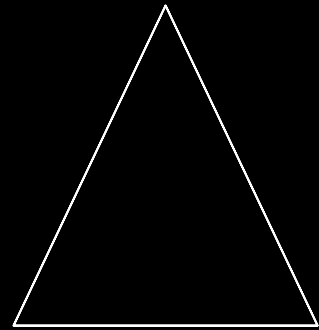
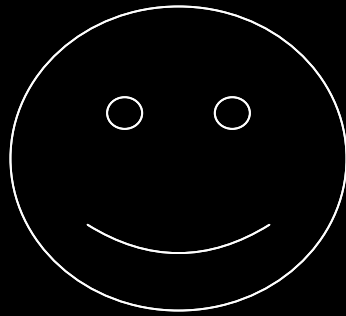
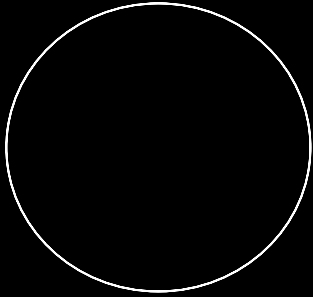


but that is a different story



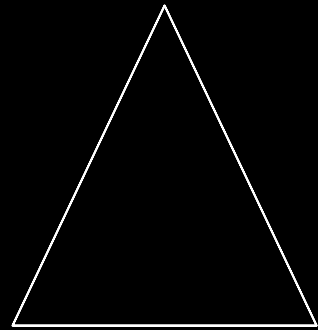
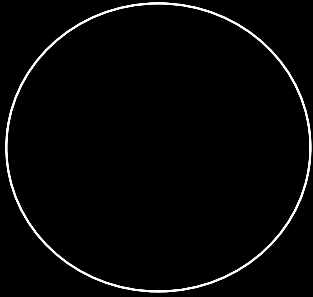


a fast draw





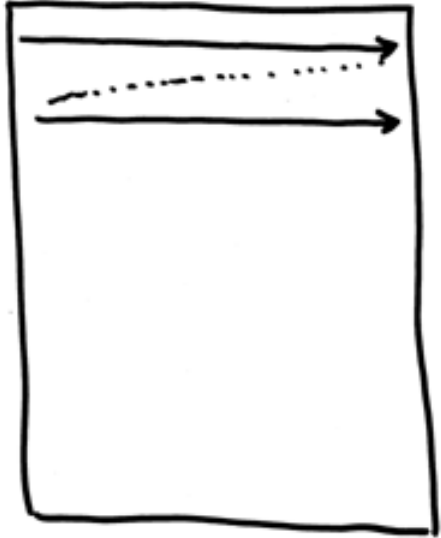
filled triangle

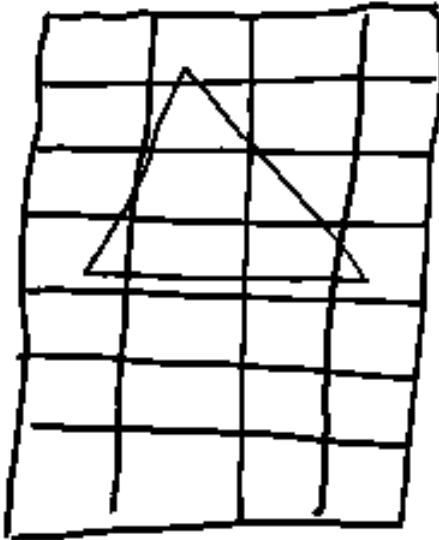


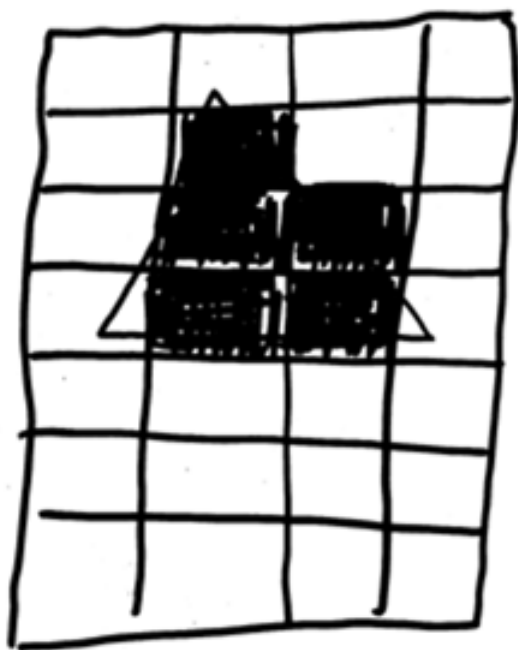


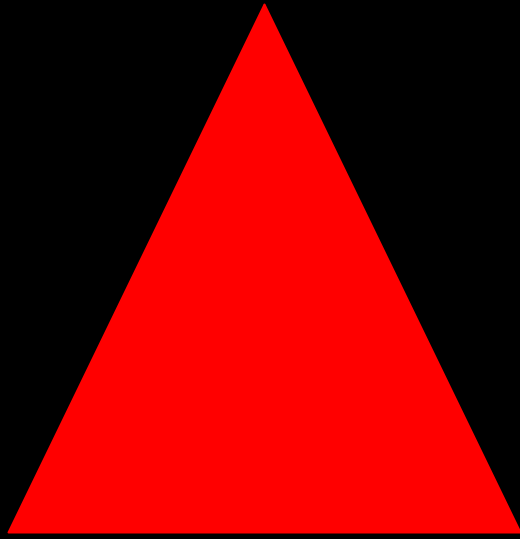
but wait ... there were lessons to be learned



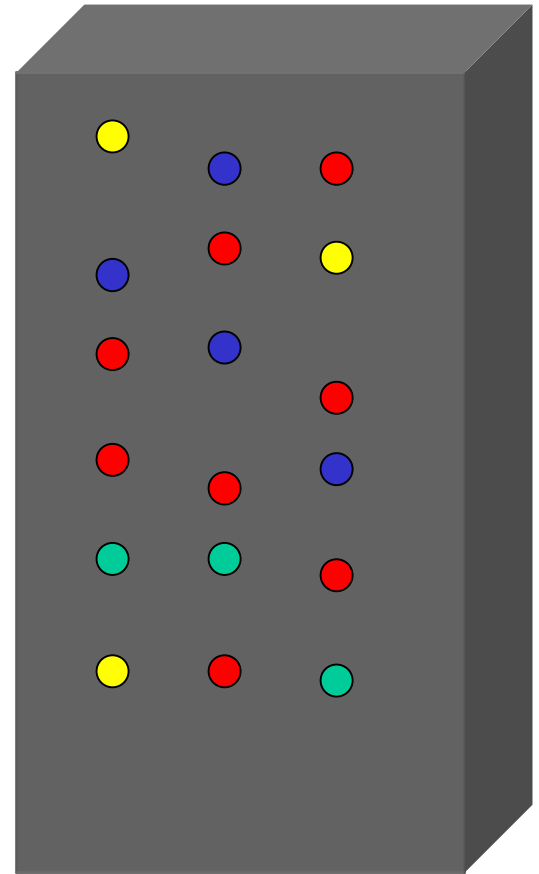
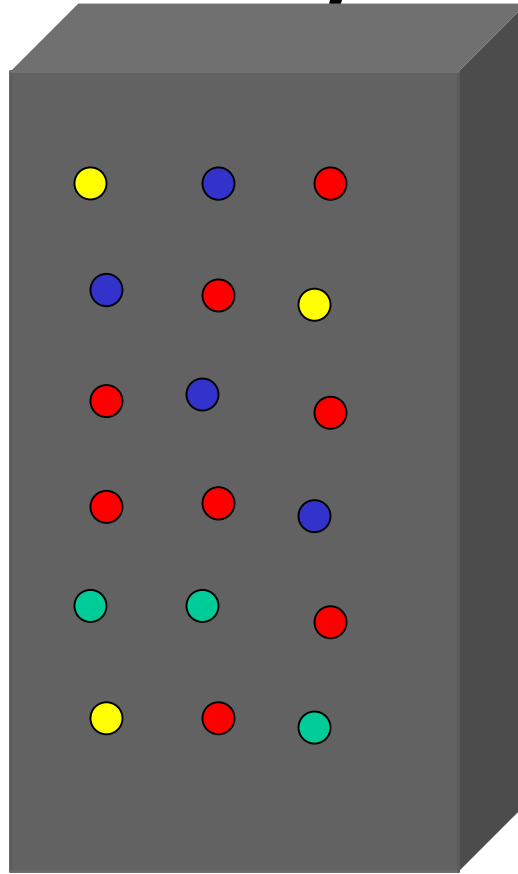
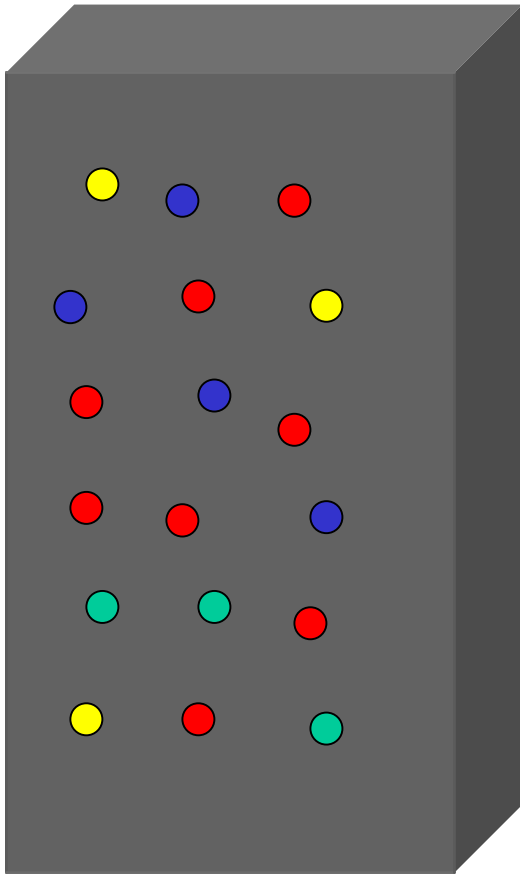




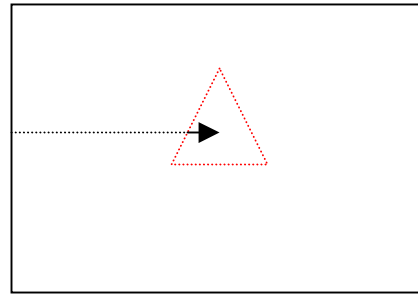
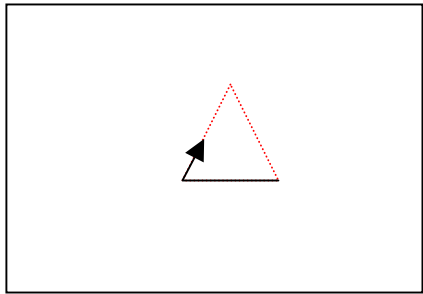




and it only took

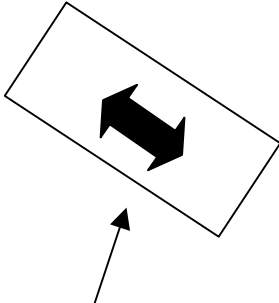
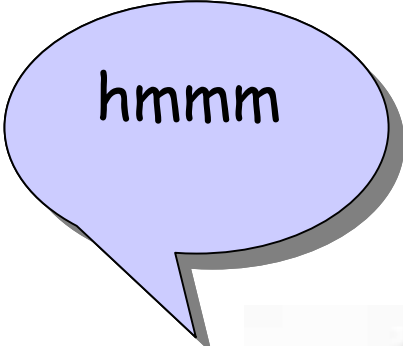
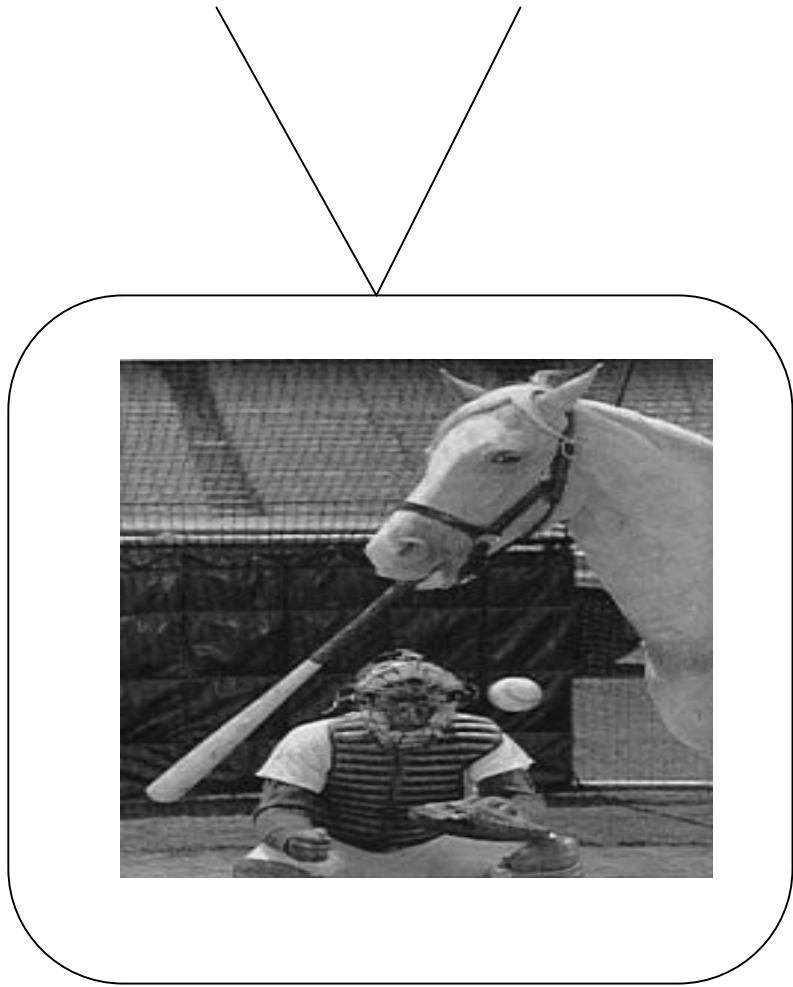


# vector vs. raster graphics



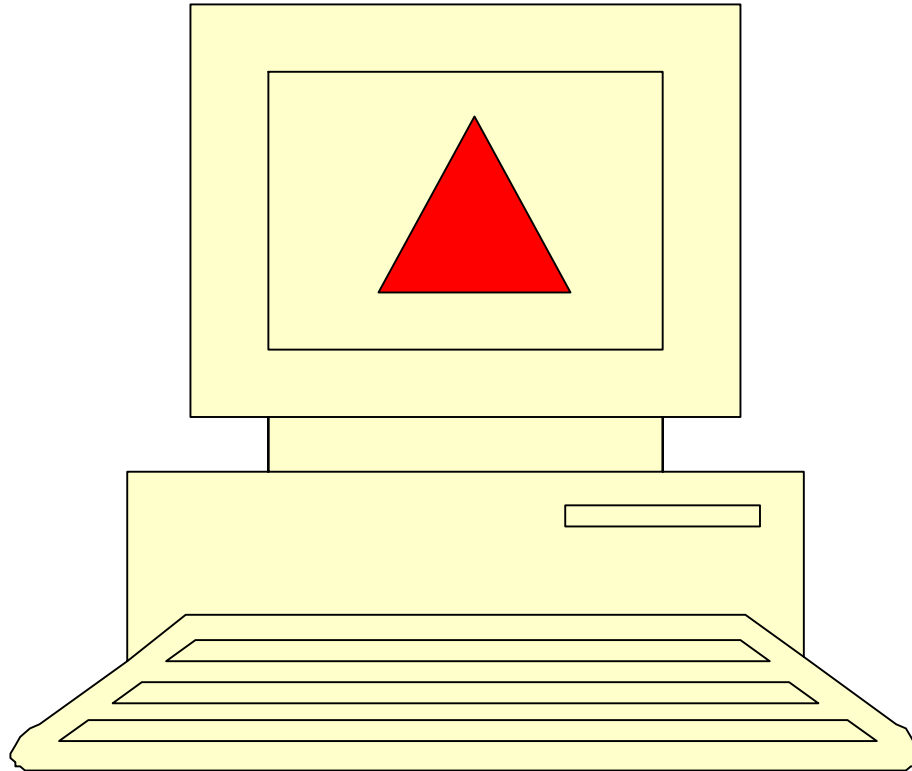
meanwhile

back in cambridge

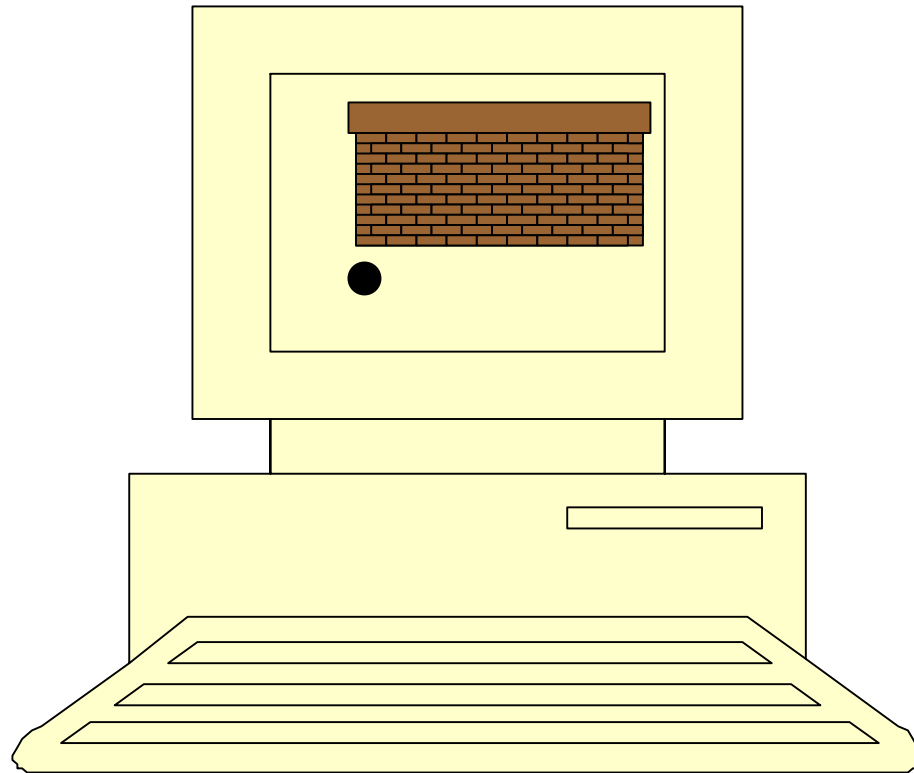


remote  
control

and a few years later...



# and more



meanwhile

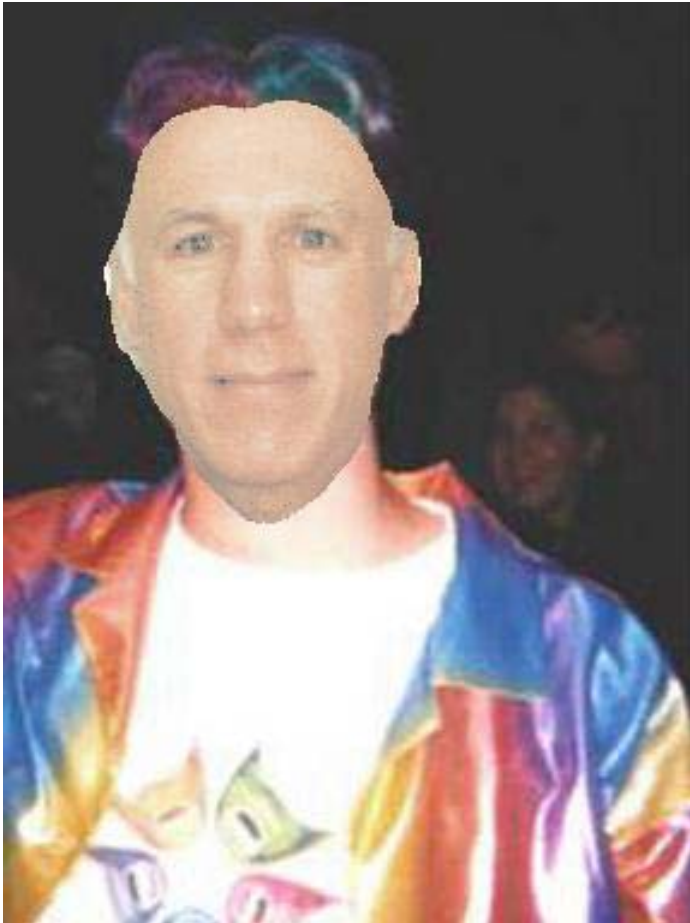
back in hollywood

# digital compositing

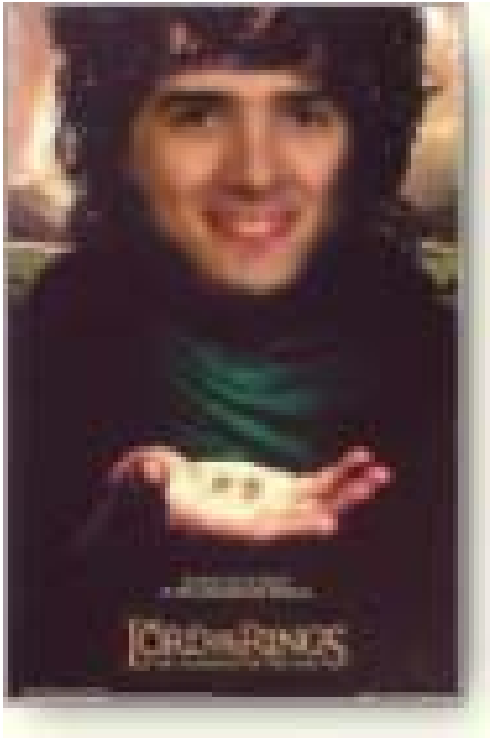
blend parts from different  
digital images to create a new  
image

demo

# more famous composites



# and more



# another technique: warping



# morphing

transform one image  
into another by  
compositing and warping  
across time

# morphing



steve diverdi

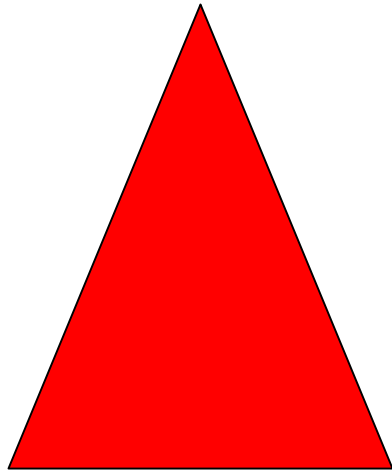
# computer graphics

1. image processing
2. rendering
3. modeling
4. animation

meanwhile

back in utah

# the lowly triangle



# triangle mesh





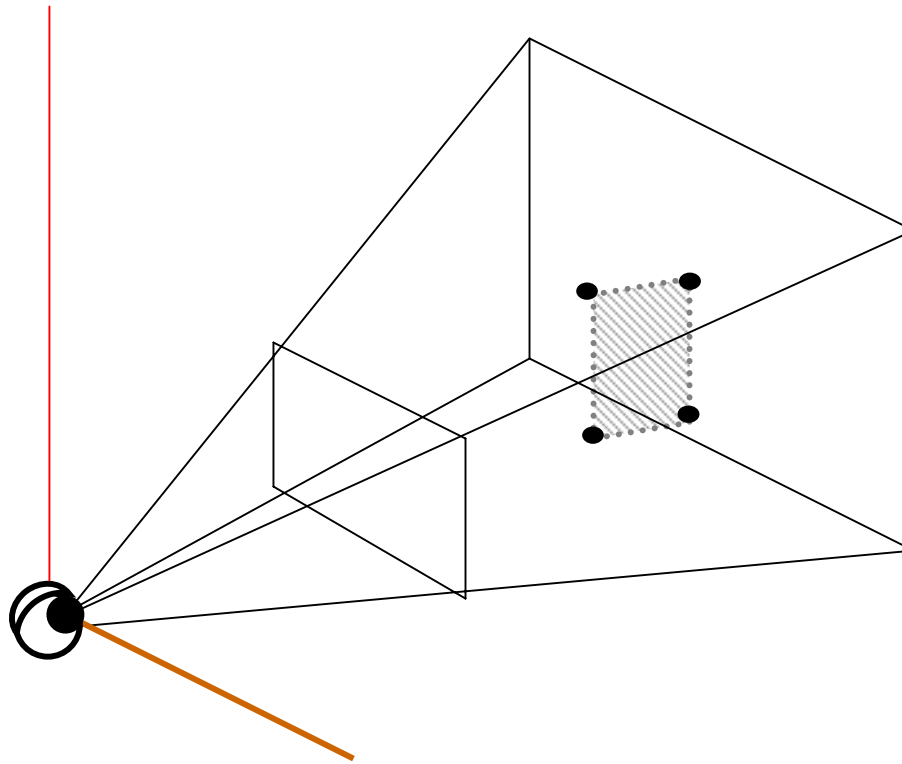
# triangle mesh



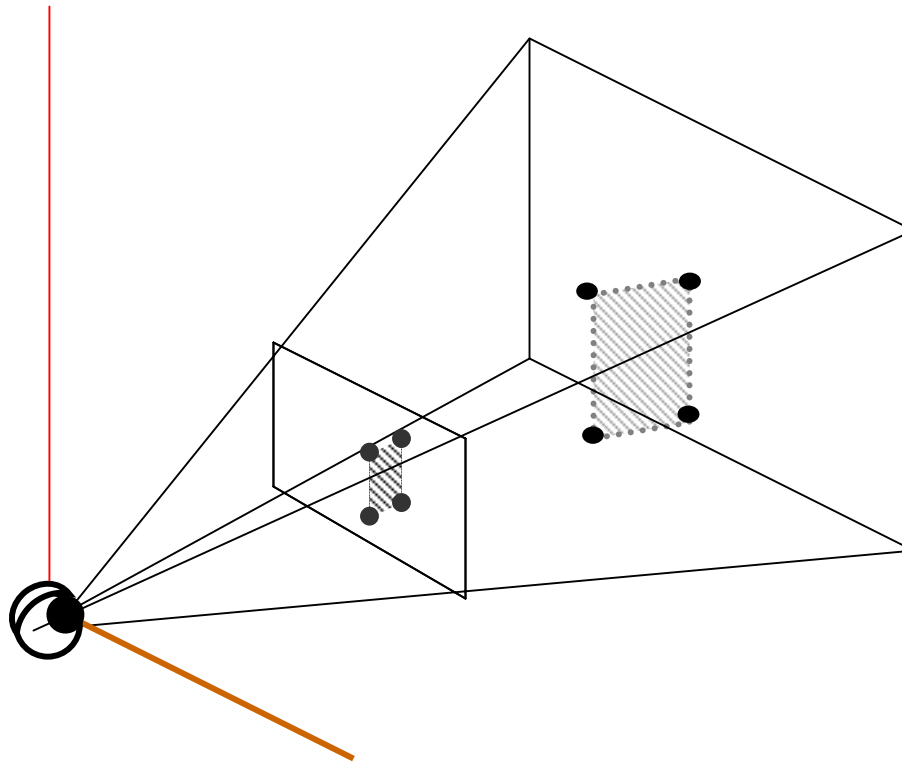
stanford michaelangelo project

# the graphics pipeline

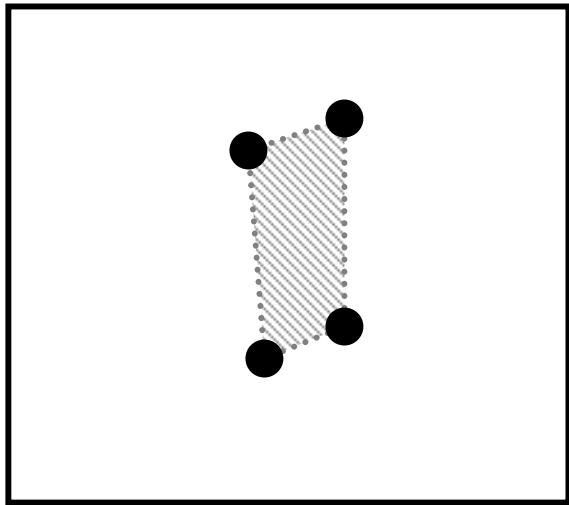
# pipeline: build scene



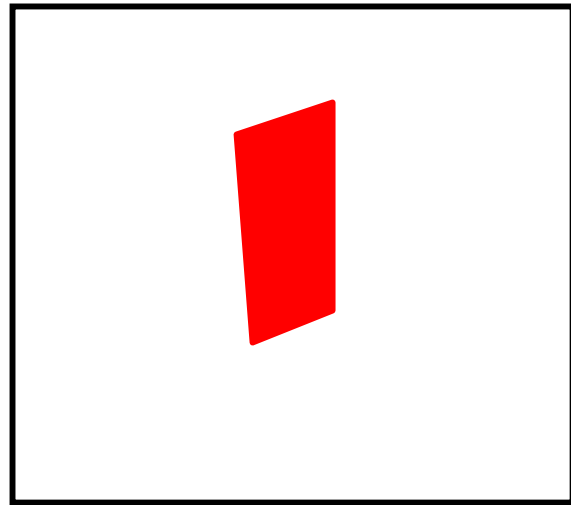
# pipeline: project



# pipeline: scan convert

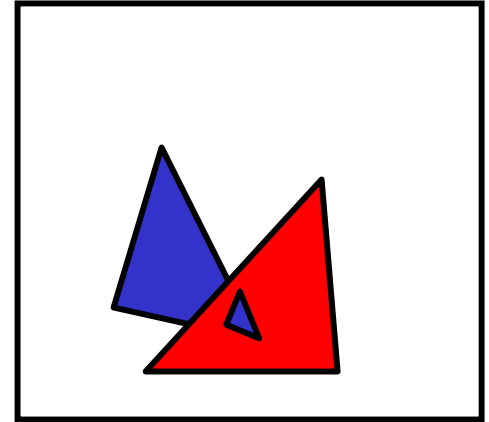
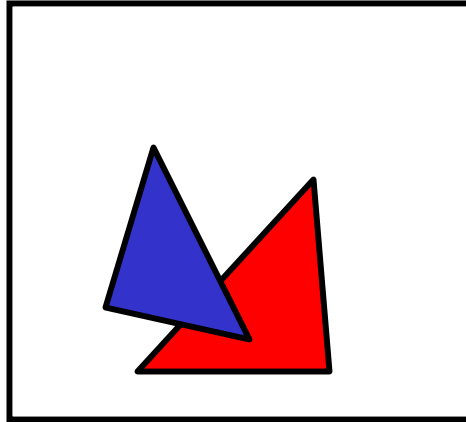
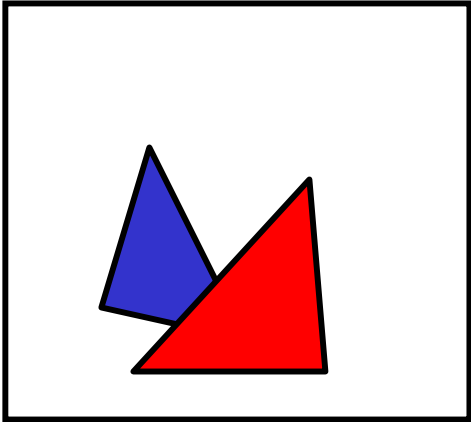


vertices in view plane



frame buffer

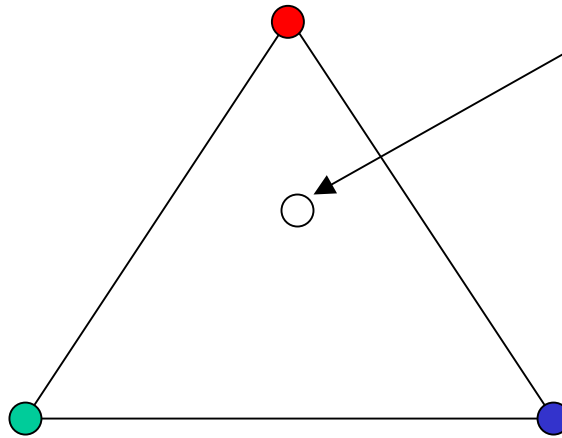
# hidden surface removal



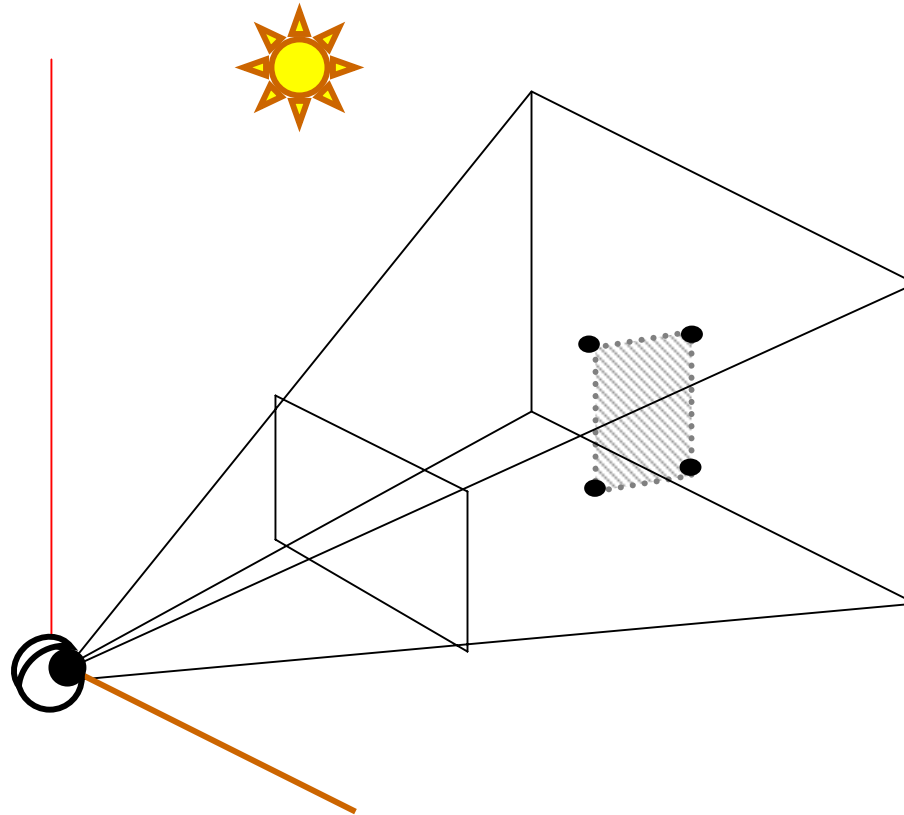
which is right?

# shading

what is color here?



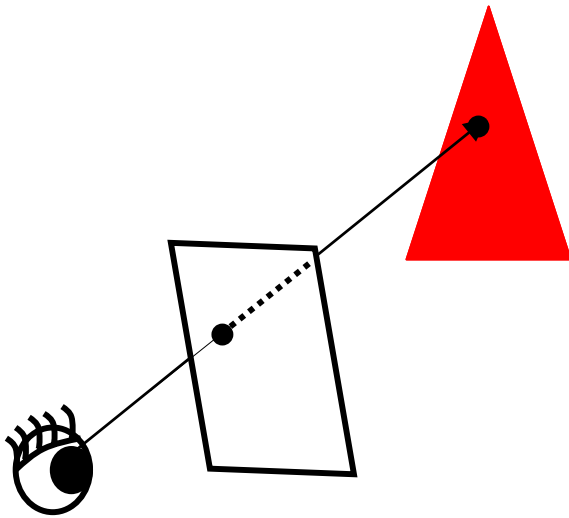
# local illumination



# the holy grail

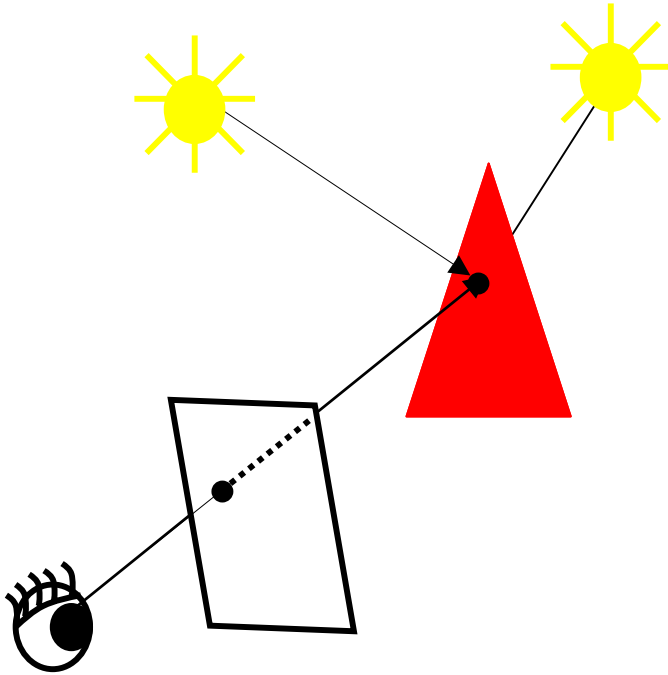
photo-realism

# ray tracing



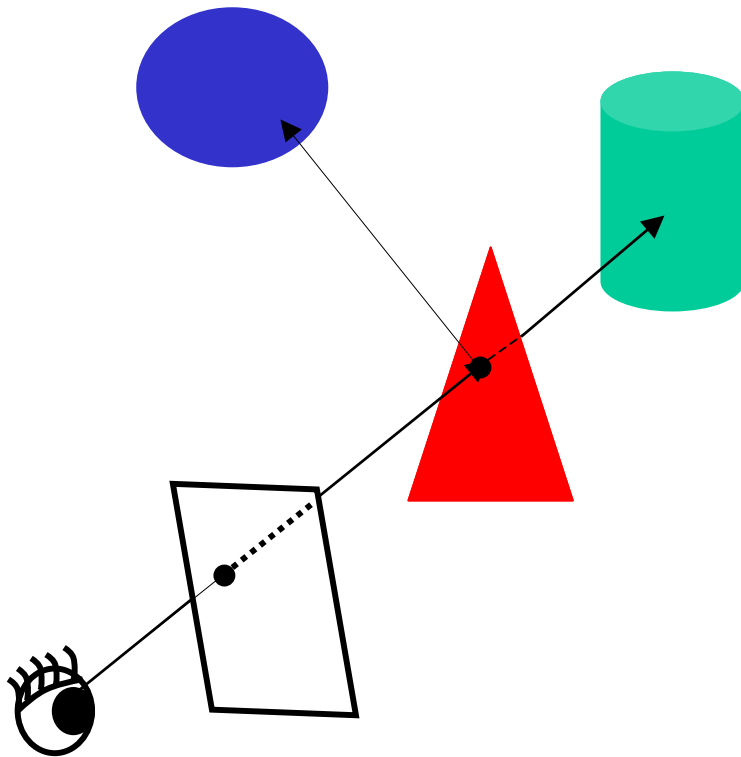
- cast ray through pixel into scene
- find intersection point (if any) that is closest to eye
- compute luminance at intersection

# luminance



- direct illumination from light sources
  - reflection off surface
  - transmission through surface
  - subject to occlusions

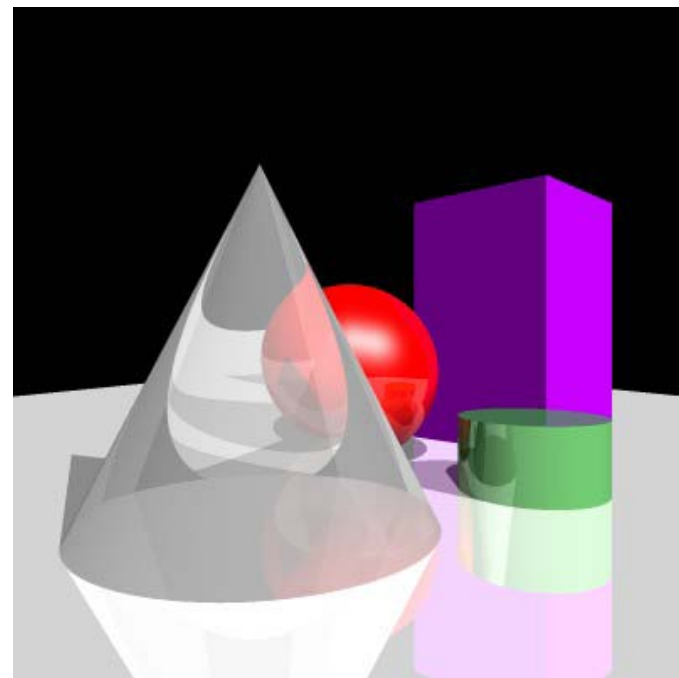
# luminance



- direct illumination
- inter-object specular reflection

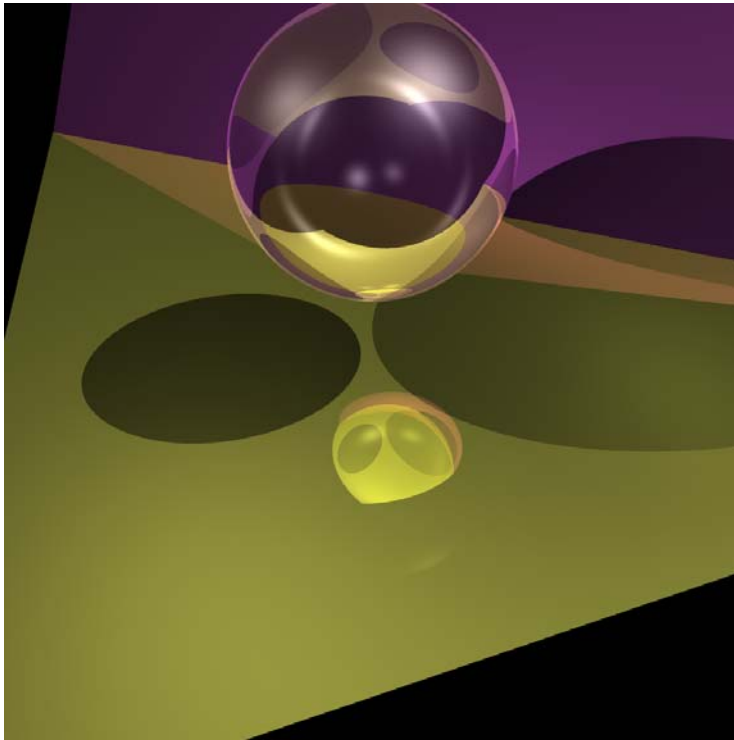
# ray tracing

```
<scene>  
  <cone material="glass">  
  <sphere color="red">  
  <box color="purple">  
  <floor material="marble">  
</scene>
```

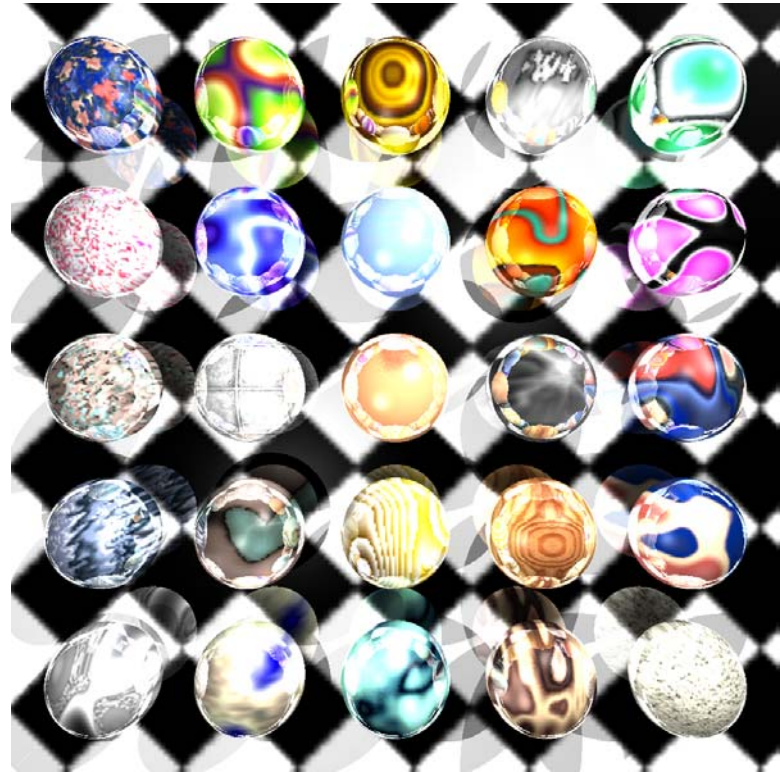


peter henry

# ray tracing

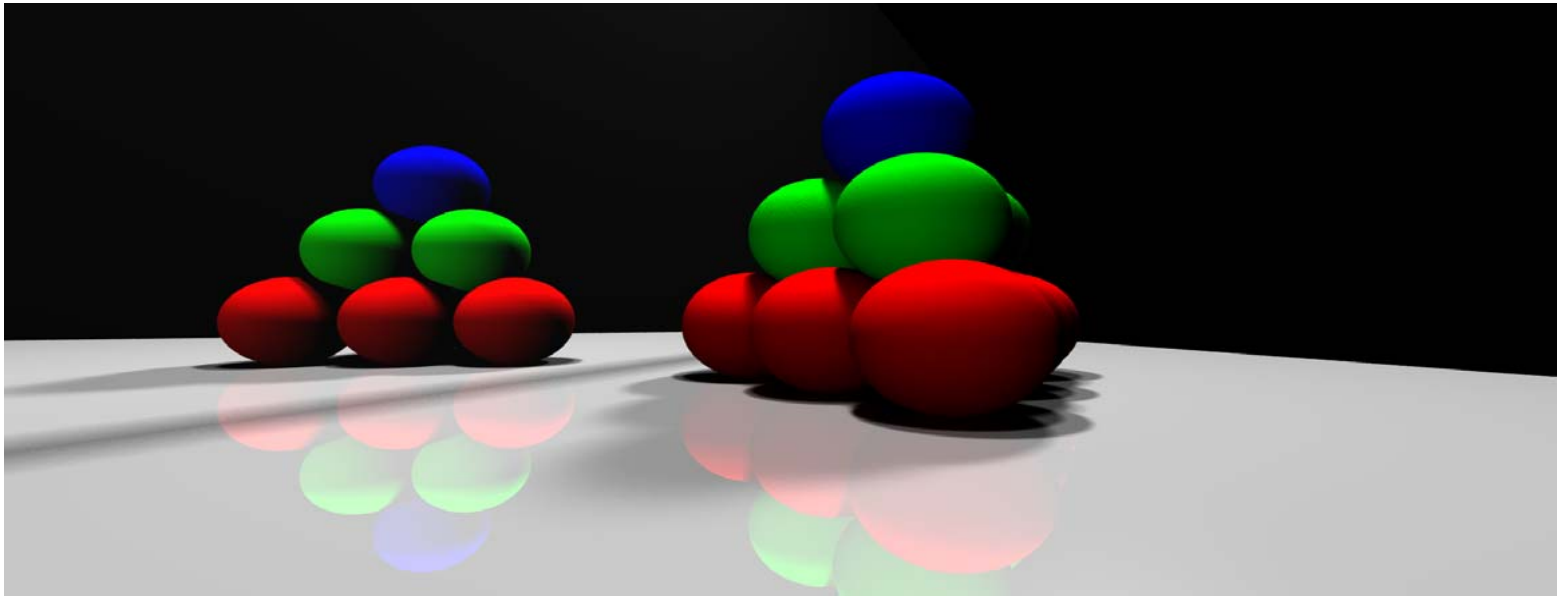


andrew mcdonnell



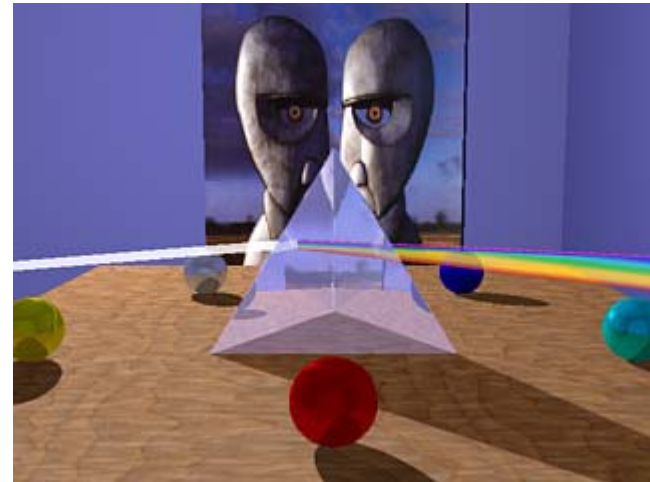
drew levin

# ray tracing

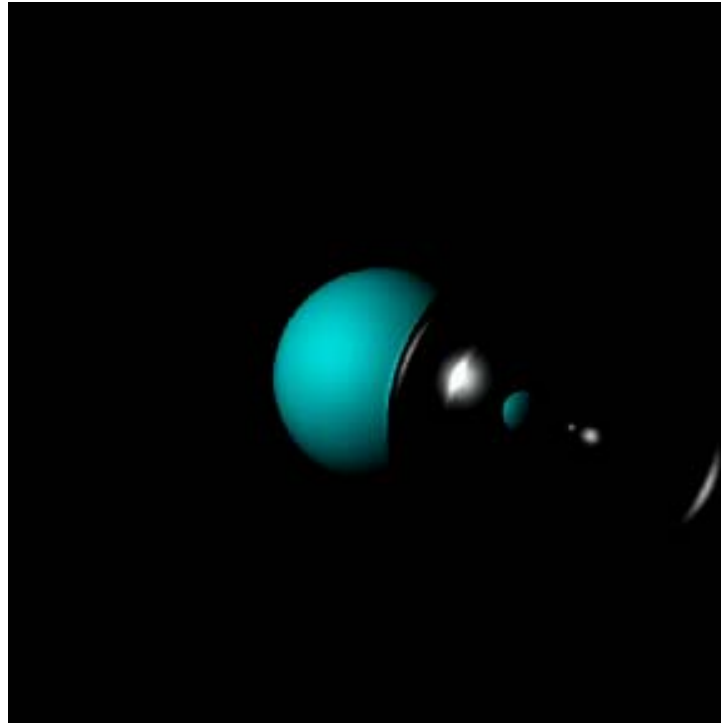


jason wither

# cmu ray tracing

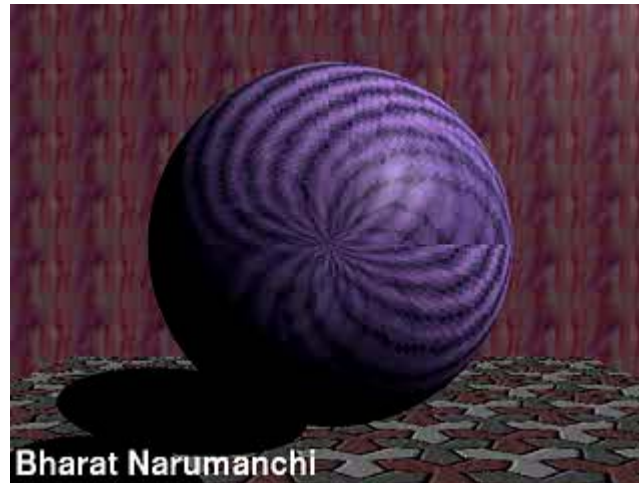


# ray tracing in action



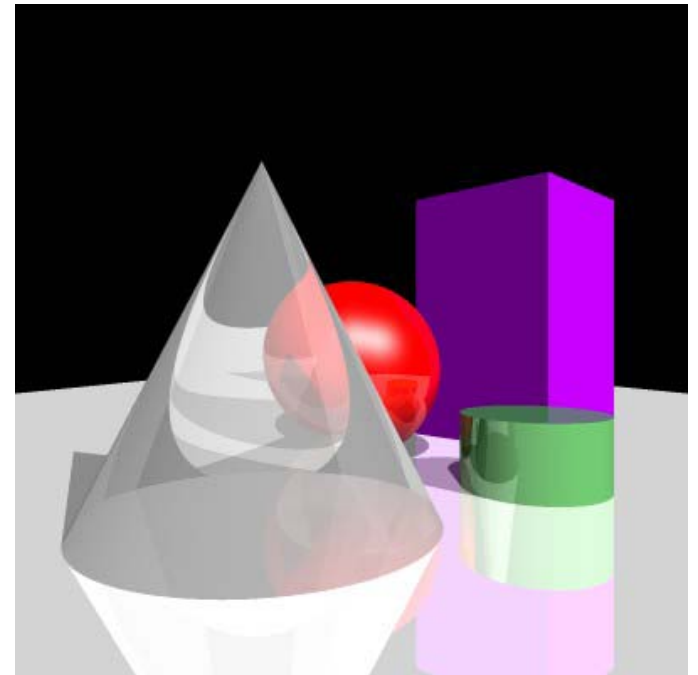
steve diverdi

# cmu ray tracing in action



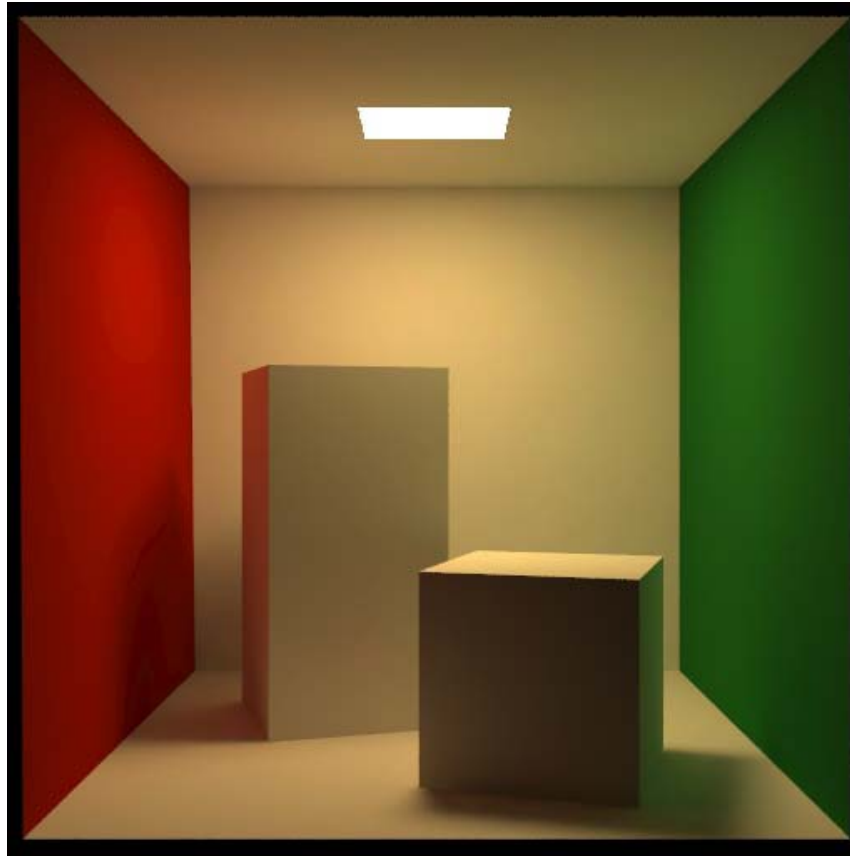
# semi-local illumination

what is wrong with  
this picture?



peter henry

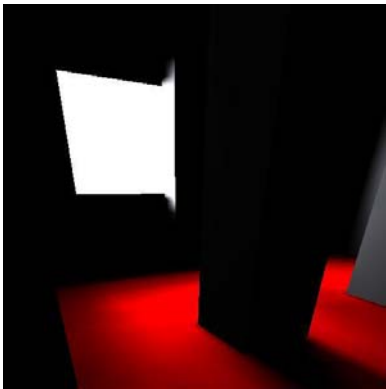
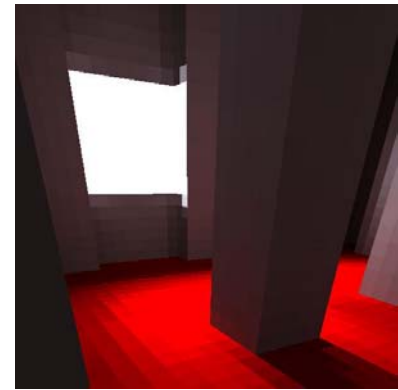
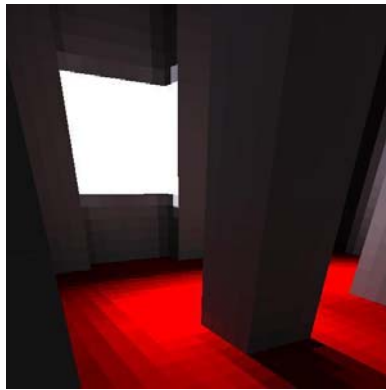
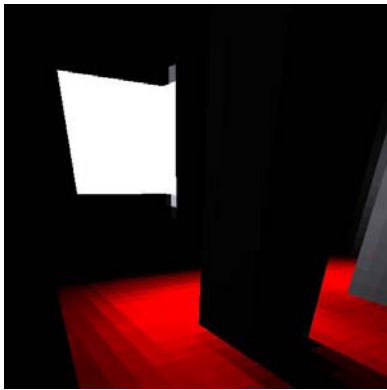
# diffuse reflections



# the holy grail

photo-realism

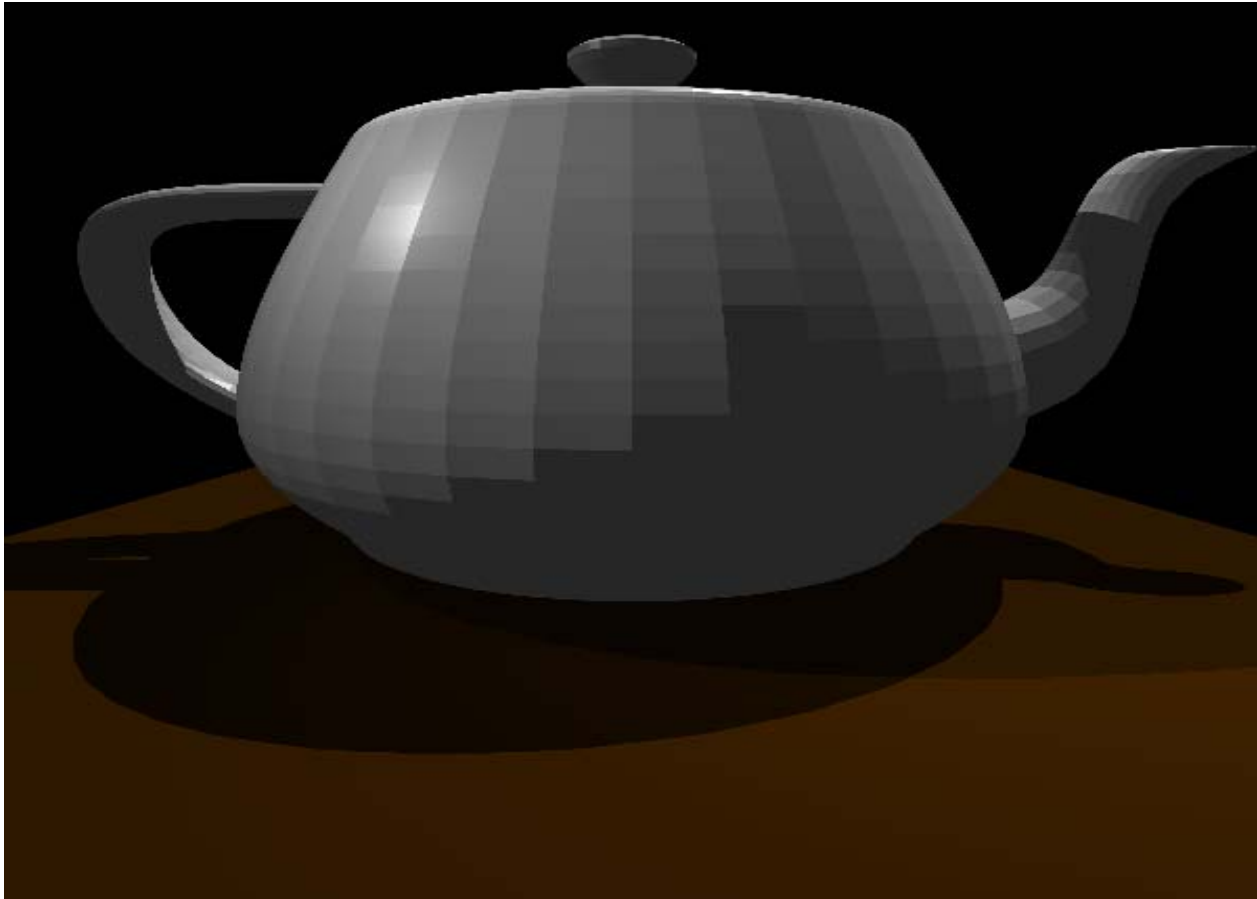
# radiosity



# computer graphics

1. image processing
2. rendering
3. modeling
4. animation

# polygon mesh



# surface modeling



# computer graphics

1. image processing
2. rendering
3. modeling
4. animation

# a few short animations

- skylark
- just dessert
- gas planet

# computer graphics

1. image processing

2. rendering

3. modeling

3. modeling

4. animation

CS155: Computer  
Graphics

CS181a: Computer  
Animation

# CS155 Prerequisites

- Linear algebra
- C++ programming
- Algorithms & data structures

# warning

this class is a lot of work!! drop it now  
if

- you have a heavy load this semester
- you haven't had the prerequisites
- you aren't a capable programmer

P.S. this class is also a lot of fun!

# basic course requirement

you cannot blow off any assignment

to pass the class you must submit a solution for each project that successfully implements at least 50% of the assigned features.

# CS155 Computer Graphics

## Projects

- Image processor (15%)
- Ray tracer (20%)
- Mini-pipeline system (15%)
- OpenGL application (20%)
- Final project (25%)

Lab & Class Participation (5%)

# course web page

[www.cs.hmc.edu/courses/2003/spring/cs155](http://www.cs.hmc.edu/courses/2003/spring/cs155)