CS4160

COMPUTER GRAPHICS

class 1

1

today's class

course mechanics & administrivia

what is "computer graphics"?

history

course overview

<break>

rendering architectures: ray tracing and pipeline rendering

graphics-intensive-software: AutoDesk's Maya

2

course mechanics & administrivia

3

cs4160 : course info

when:	thursday 6:10-8:00
where:	516 hamilton
instructor:	michael reed
office hours	thursday after class, or by appointment
email:	he li - hl2918@columbia.edu
TAs:	justin chang - sz2558@columbia.edu

4

online presence

canvas :

class notes

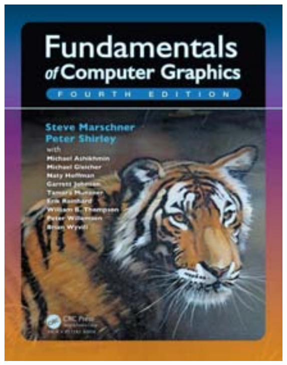
supplementary reading & resources

homework assignments & submits & grades

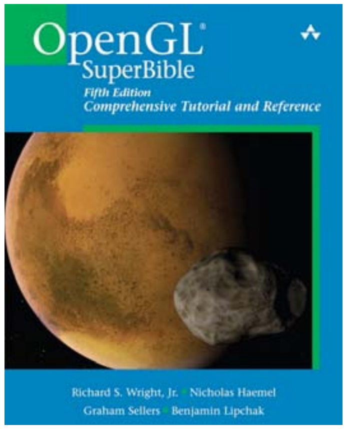
discussions on all of the above (on Piazza)

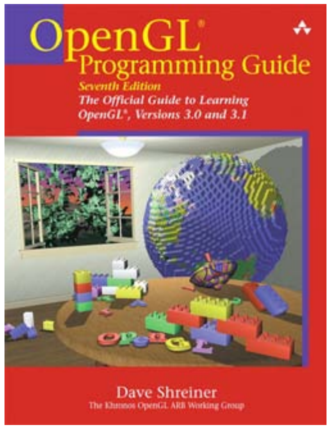
textbook (required)

- by Peter Shirley & Steve Marschner
- n.b. fourth edition: 2016
- available @ amazon, bookstore, library reserve



textbooks - optional
(earlier editions are available online)


OpenGL®
SuperBible
Fifth Edition
Comprehensive Tutorial and Reference
Richard S. Wright, Jr. • Nicholas Haemel
Graham Sellers • Benjamin Lipchak


OpenGL®
Programming Guide
Seventh Edition
The Official Guide to Learning OpenGL®, Versions 3.0 and 3.1
Dave Shreiner
The Khronos OpenGL ARB Working Group

prerequisites

CS skills:

- sorting / searching
- data structures
- programming in (basic) C++
- numerical methods
- basic software engineering skills - work easily with multiple files, use debugger, etc.

prerequisites - C++

class inheritence

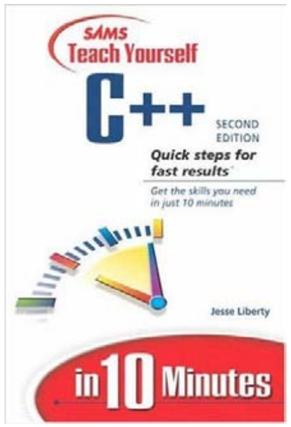
polymorphism

a **little** Standard Template Library (std::vector)

passing by reference vs. value, pointers, references, etc.

9

prerequisites - C++


SAMS Teach Yourself
C++
SECOND EDITION
Quick steps for fast results
Get the skills you need in just 10 minutes
Jesse Liberty
in 10 Minutes

available for free on CLIO…

10

prerequisites

math skills:

- trigonometry
- linear algebra (vectors, matrix methods)
- polynomials & their derivatives
- basic integration & differentiation

11

homework assignments

there are 2 programming themes…

with ***approximately*** 6 assignments in the first, and 5 in the second.

assignments are weighted equally.

12

homework assignments

done in (basic) C++ :C w/ classes, polymorphism, and templates

must be *compilable* on the **clic** machines

- you should get a CLIC/CS account (see www.cs.columbia.edu/~crf)
- best development setup: your own laptop for 90% of the work, then CLIC for the final testing.

CLIC lab

reserved just for us:

M - 10am - 12pm
T - 6:30pm - 8:30pm
W - 10am - 12pm
Th - 10am - 12pm
S - 1pm - 4pm
S - 1pm - 4pm

There are 21 machines with good real-time graphics capabilities (new nvidia GPUs) - these will be useful for the second theme of the class

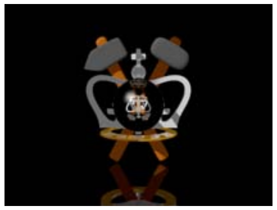
“creative submissions”

most assignments will allow “creative submissions” in addition to the expected implementation

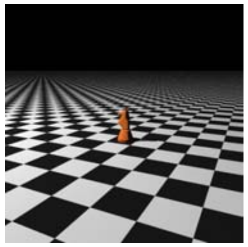
this is a chance to show your creative side!

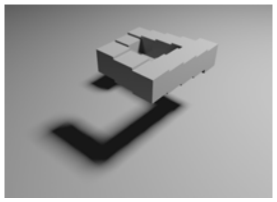
additional (small) points, class-wide voting for the best submissions, and end-of-semester recognition await!

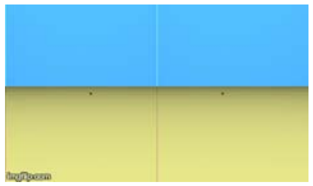
homework results from previous classes



homework results from previous classes







17

homework assignments

late policy:

homework assignments have a specified day and hour they are due

scores are reduced by an additional 20% during each 24-hour unit past the due date/time

18

homework assignments - collaboration

Discussion of algorithms is encouraged, as well as the sharing of drawings and other representations of problems.

Showing each other images to help debug - YES, FINE! (on Canvas too!)

Not permitted: the sharing or acquisition of code in any form.

Any material from an outside source must be explicitly acknowledged.

we use Moss to compare homeworks!

19

homework assignments - reading

there are reading assignments given every week

usually we cover the reading material in the next week's classes - so be prepared with questions

the class slides & discussions support the reading material, and add a different perspective

however, the class slides are really just notes - you must attend class!

20

exam

1 final exam

designed to be relatively easy for those that have attended class, done the reading and assignments

closed book

21

grading

final score =

.80 x (average of homeworks)
+
.20 x final

22

class exception!

I *may* not be in class Spetember 22

that class will be made up (prob 12/21)

attendance is expected.

23

about me

"Solid Model Acquisition from Range Imagery" - CUCS thesis, 1998

24

about me

currently: R+D group at Blue Sky Studios

25

what <u>is</u> computer graphics?

definition: **computer graphics** is the sub-field of computer science which studies methods for digitally synthesizing and manipulating visual content.

primary concerns:

modeling

rendering

animation

26

what <u>is</u> computer graphics?

modeling: the specification of shape (also: materials and lighting)

rendering: the generation of an image from a model

animation: combining modeling and rendering with the passage of time

27

computer graphics sub-fields

images and image processing

2D graphics: 2D modeling, rendering & animation

3D graphics: 3D modeling, rendering & animation

scientific visualization

user interfaces & virtual reality

real-time rendering, non-photoreal rendering, etc. etc.

...

28

why animation is such a good test of a graphics system

computer animation stresses the parts of a rendering system to the extreme, due to these requirements:

- very complex scene geometry and materials
- complex motion
- extreme geometry deformation
- motion blur
- camera effects (like depth-of-field)
- the need for everything to be directable

computer graphics - related fields

image processing

physics

applied math

simulation

human perception

algorithms

computer architecture

numerical methods

software development

this course: primarily rendering

we are going to focus on methods that can generate **realistic** images of 3-dimensional scenes

in particular:

ray tracing (and its variants)

pipeline rendering (through the OpenGL library)
(also called “scanline rendering”, “object-order rendering”, ...)

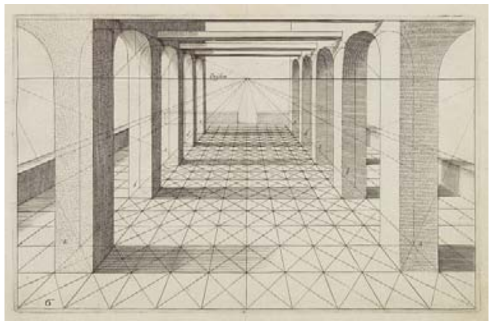
what is realism in an image?

realistic images combine 2 attributes:

- “correct” perspective
- “correct” color

33

what is realism in an image?



34

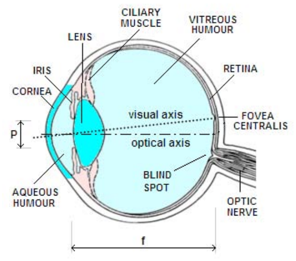
what is realism in an image?

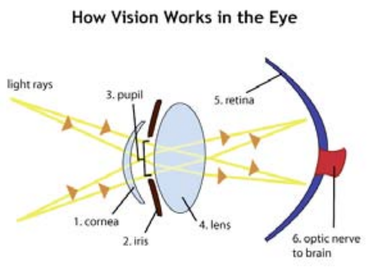
35

what is realism in an image?

36

what is realism in an image?





class topics

<u>goal</u>: understanding realistic rendering in 3 dimensions, including:

- how images are represented, created and modified
- how geometry is represented
- how lighting and material models are combined to determine appearance
- how to combine all of the above into a renderer

implicit class topics

high-performance programming (good graphics code is high performance code!)

software engineering fundamentals (i will frequently discuss my code, and some of your code)

class schedule - VERY roughly

first 2/3: (assignments based on raytracing)

- camera models
- geometric representation
- transformations
- materials
- lighting
- efficiency structures

last 1/3: pipeline rendering (OpenGL)

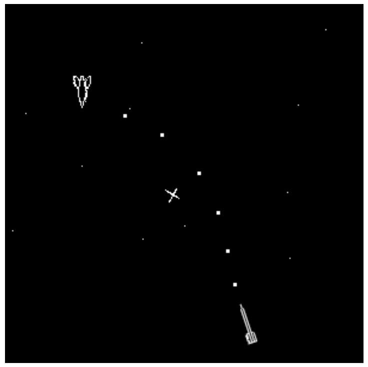
- all of the above, in OpenGL context

a (brief) history of computer graphics

41

ivan sutherland 1963

42



steve russell - spacewar

43

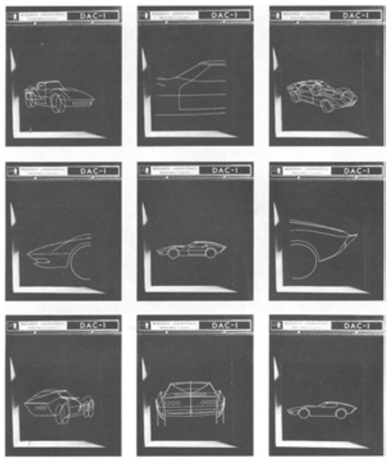


FIG. 1-8 Scale expansion, rotation, and partial views in a DAC-1 design exercise.

GM & IBM “Design Augmented by Computer”

44

also in the 1960s

research into raster devices: Breshenham
geometric representations: Coons, Bezier
rendering algorithms: Appel
hardware & systems:
mouse
hypertext
shared-screen collaboration

“computer graphics” term coined

45

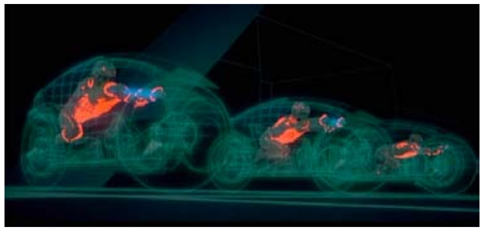
1970s

“pong” & home video games
Apple 2
research in:
 shading (Phong)
 surfaces & texture mapping (Blinn)
 geometry & rendering (Catmull)

46

1980s

MAGI + others: Tron
AutoCAD design software
Pixar formed (from ILM)
hardware: VGA introduced
computers for education



47

1990s

Pixar: introduces renderman (’90), Toy Story (95)
SGI introduces OpenGL
NVIDIA, 3DFX, ATI are all profitable companies
resurgence of the computer gaming industry

48

21st century

animated film industry no longer novel (ditto FX)
every computer has excellent graphics capabilities
every cellphone has advanced graphics capabilities

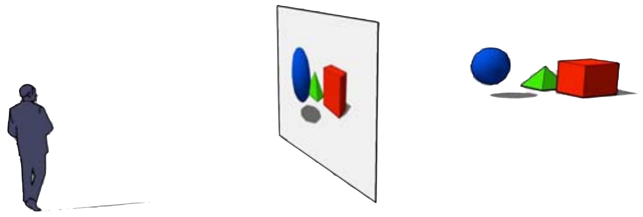
49

rendering

50

50

rendering



51

51

other types of rendering than “3d”

font display

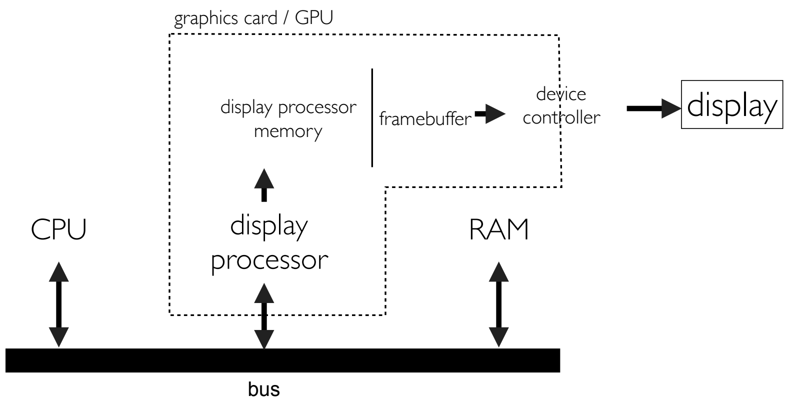
window & mouse

images (e.g. taken with camera, or precomputed)

52

52

how do images get displayed?



53

53

image "rendering"

(more accurately: "image display")

input: image (2D array of color values)

output: display

process: resampling

speed: interactive

54

54

window/viewport rendering

input: 2D geometric data (e.g. fonts and windows, represented mathematically)

output: display

process: 2D render - geometry used to create appropriate image at appropriate scale in graphics card

speed: interactive

notes: simple materials, lighting and transparency; renderer: quartz on mac, direct2D on MS

55

55

interactive 3d rendering

input: 3D geometric data (shapes), & **simple** lighting, materials & camera model

output: display

process: 3D rendering (pipeline renderer) - possibly sped up with hardware (through openGL or directX)

speed: interactive

56

56

“off-line” 3d rendering

input: 3D geometric data (shapes), & any lighting & camera model

output: image file (jpeg, png, etc.)

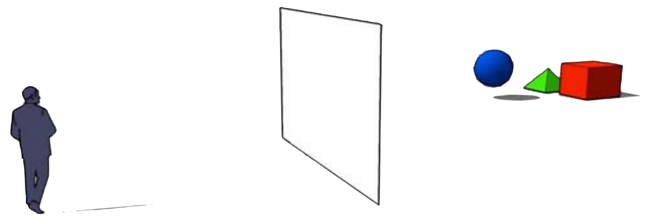
process: 3D rendering (ray tracing or pipeline renderer) - mostly all in software

speed: slow to very, very slow (hours per image)

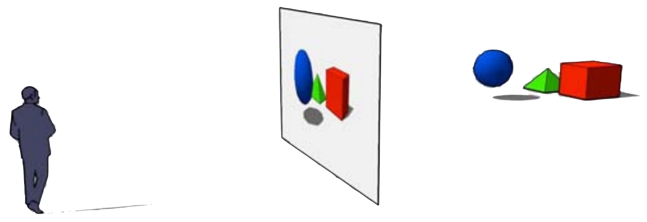
later: image “rendered” to display via image viewer

rendering

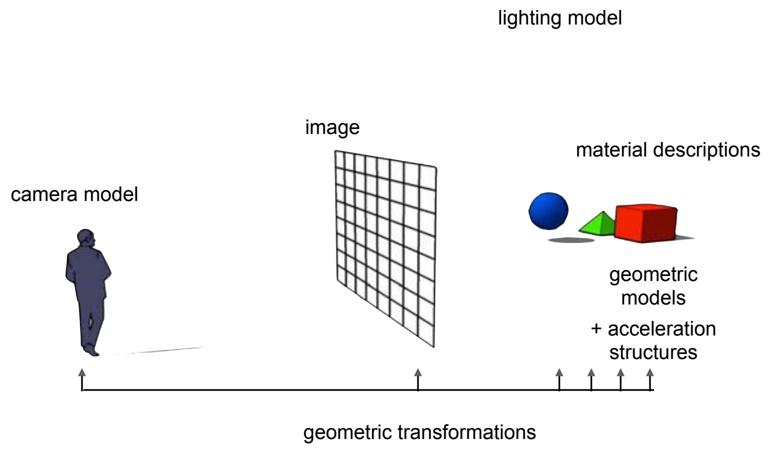
rendering



rendering



rendering overview

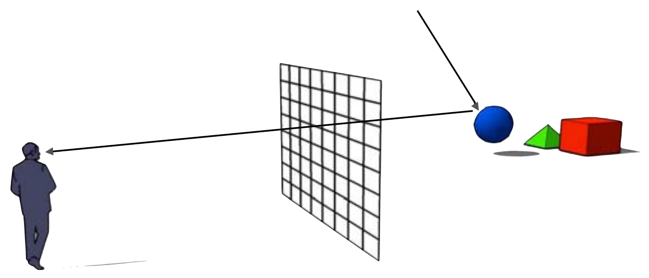


rendering

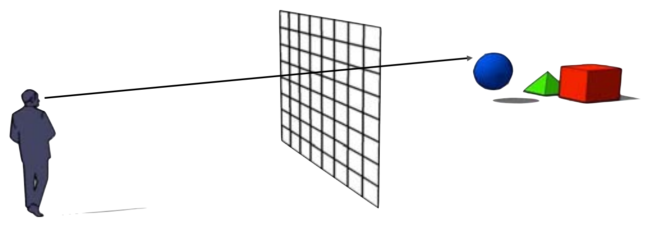
2 common methods:

- raytracing
  - examples: Mental Ray, Blue Sky's CGIStudio, etc. etc.
- pipeline rendering
  - called “object-order rendering” in the book
  - often called “scanline rendering”
  - examples: Pixar's PRMan, Maya's default renderer, most commercial renderers, 99.99% of hardware renderers

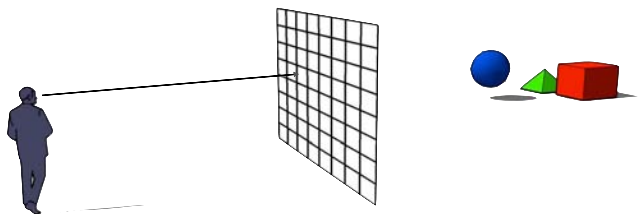
ray tracing



ray tracing



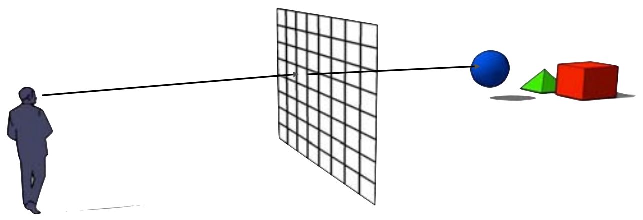
ray tracing



65

65

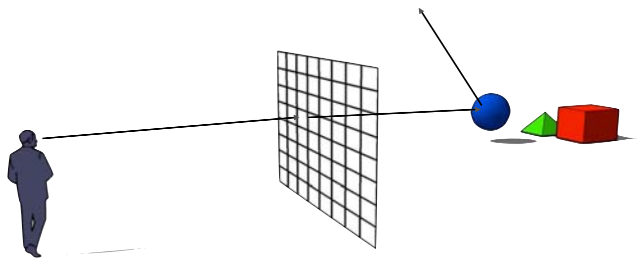
ray tracing



66

66

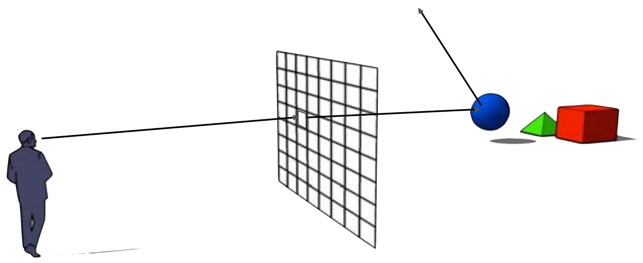
ray tracing



67

67

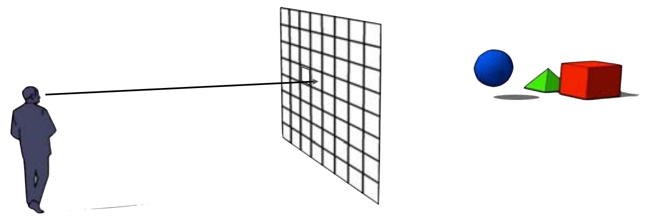
ray tracing



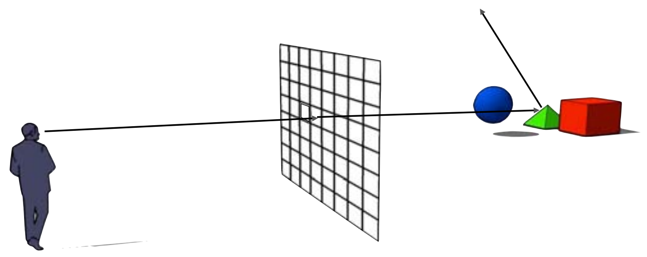
68

68

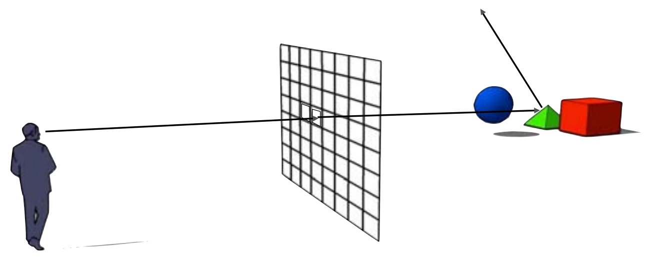
ray tracing



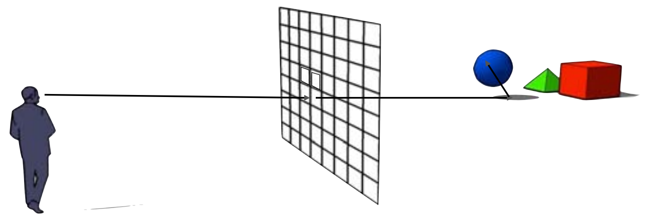
ray tracing



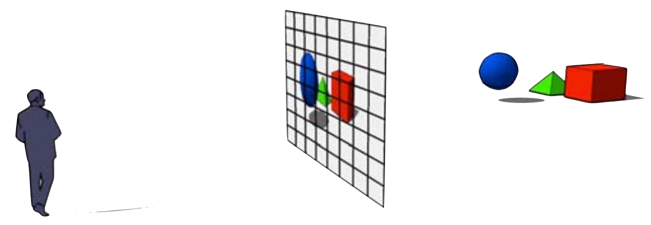
ray tracing



ray tracing



ray tracing



73

73

ray tracing

operations:

- ray-object intersection
- shading: lighting & materials calculation
- geometric transformations

74

74

ray tracing

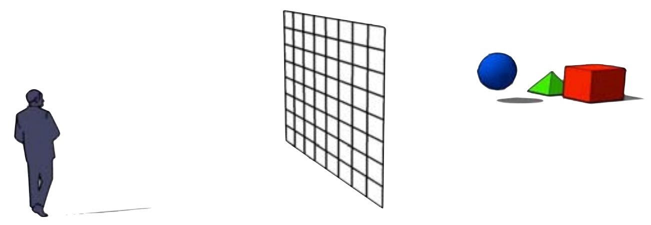
“image-order rendering”

```
for i = 1 to image_width {
  for j = 1 to image_height {

    generate “ray” from focal point through pixel (i,j)
    which_obj = first object ray intersects in scene
    calculate the shading on which_obj at intersection
    pixel (i,j) = result of shading calculation

    }
}
```

75

75

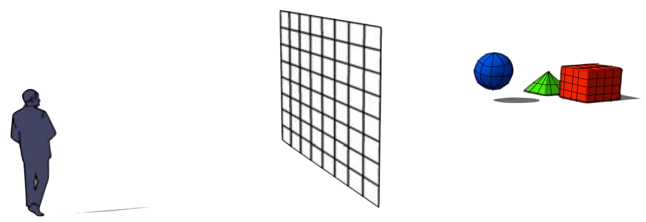
pipeline rendering



76

76

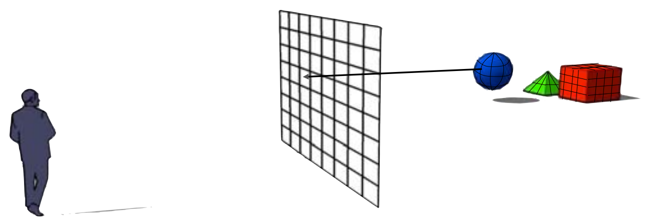
pipeline rendering



77

77

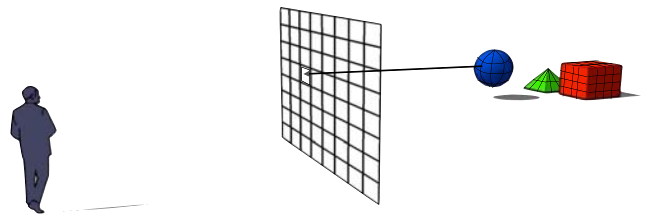
pipeline rendering



78

78

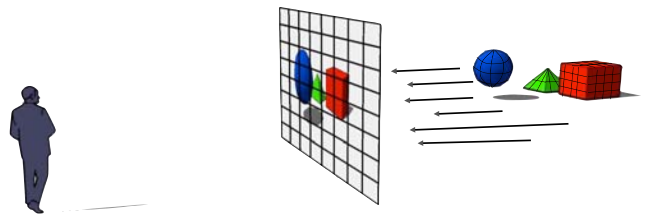
pipeline rendering



79

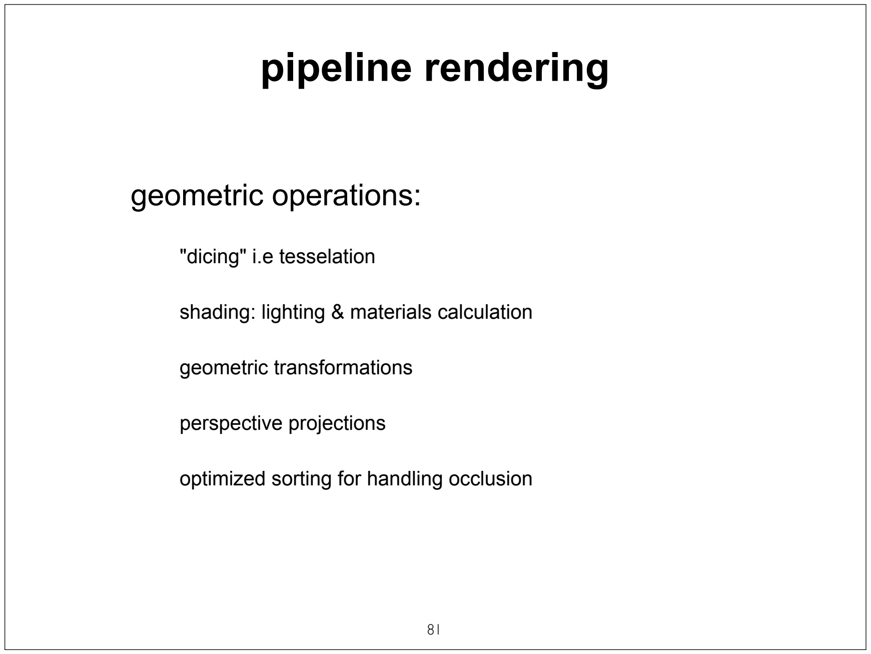
79

pipeline rendering



80

80



pipeline rendering

geometric operations:

"dicing" i.e tesselation

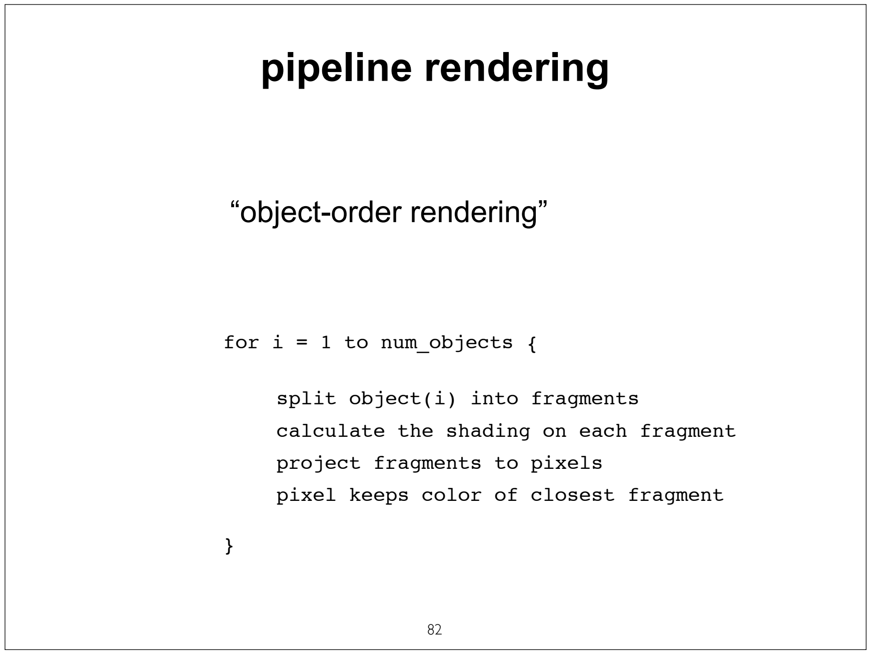
shading: lighting & materials calculation

geometric transformations

perspective projections

optimized sorting for handling occlusion

81

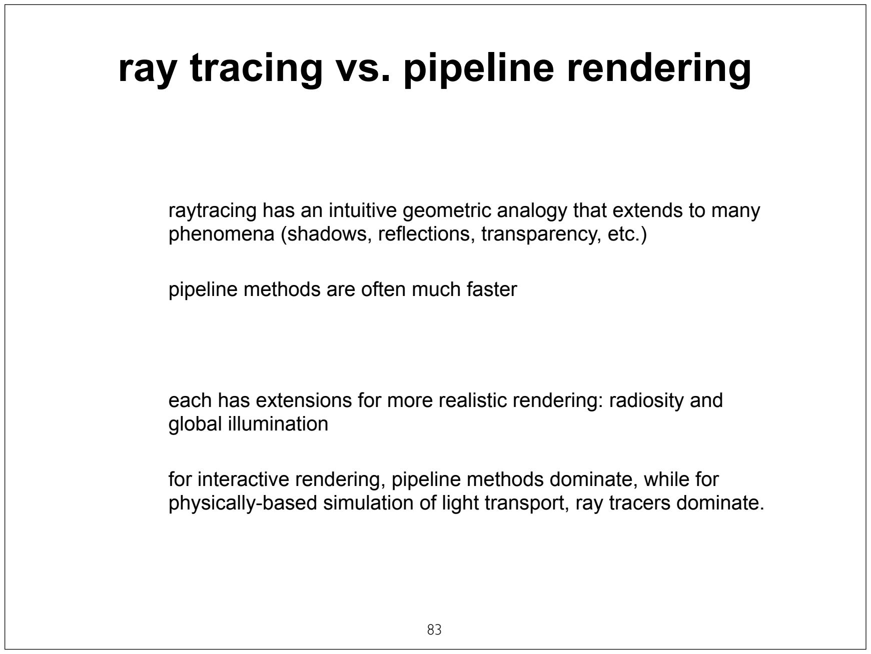


pipeline rendering

“object-order rendering”

```
for i = 1 to num_objects {

    split object(i) into fragments
    calculate the shading on each fragment
    project fragments to pixels
    pixel keeps color of closest fragment

}
```

82



ray tracing vs. pipeline rendering

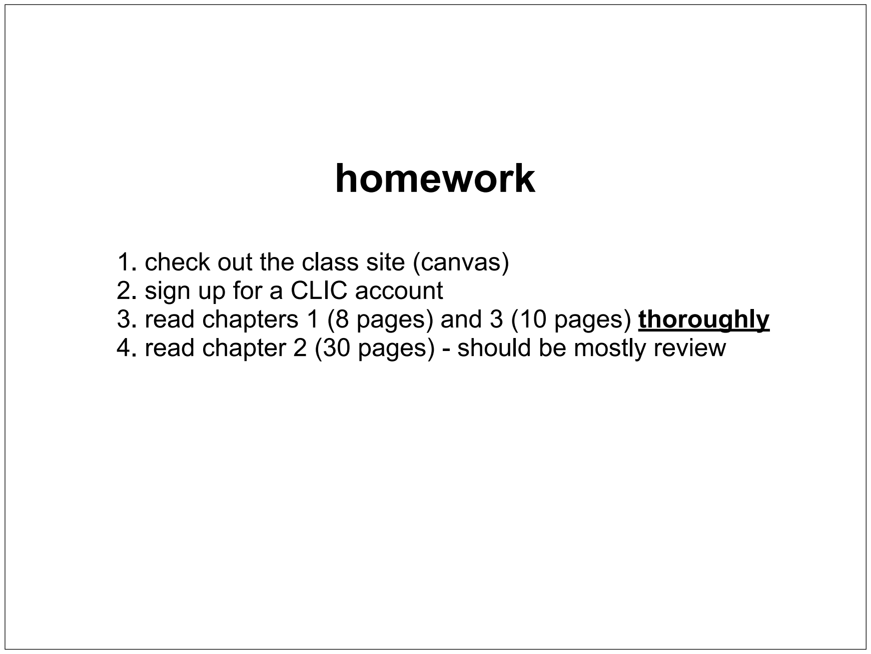
raytracing has an intuitive geometric analogy that extends to many phenomena (shadows, reflections, transparency, etc.)

pipeline methods are often much faster

each has extensions for more realistic rendering: radiosity and global illumination

for interactive rendering, pipeline methods dominate, while for physically-based simulation of light transport, ray tracers dominate.

83



homework

1. check out the class site (canvas)
2. sign up for a CLIC account
3. read chapters 1 (8 pages) and 3 (10 pages) **<u>thoroughly</u>**
4. read chapter 2 (30 pages) - should be mostly review

84