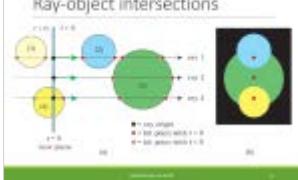
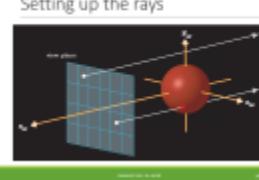


Fundamentals of Computer Graphics

PHILIP DUTRÉ

DEPARTMENT OF COMPUTER SCIENCE

Lecture Overview

<p>Admin: Course Organization</p> <p>13 lectures, 2 hours each</p> <p>No project for this course</p> <p>Final evaluation for entire course: written exam during exam period [examples on Moodle]</p>	<p>What is Computer Graphics?</p> <p>Computer graphics and its discipline has different facets, from computer-aided drafting, rendering in film/television, the use of graphics in games and computer games, and many applications in scientific visualization. This course will focus on the latter two. It will introduce the basic concepts of computer graphics and how they are used in computer games and scientific visualization. It will also introduce the basic concepts of rendering, including shading, lighting, and materials. The course will cover the basics of ray tracing, including how it works, how it can be used to render images, and how it can be used to solve other problems in computer graphics.</p>	<p>What will we learn in this course?</p> <p>Fundamentals of computer graphics rendering algorithms Basic algorithms applied here: 3D model as 2D images Ray tracing, anti-aliasing, ray-tracing applications (game design, movies, ...)</p> 
<p>Preliminaries: Images</p>	<p>Preliminaries: Colors</p>	<p>Ray Tracing: Basics</p>
<p>Ray-object intersections</p> 	<p>Polygon Mesh</p> <p>Polygon mesh = collection of connected polygons (usually triangles)</p>  <p>"Utah Teapot" "Stanford Bunny"</p>	<p>Setting up the rays</p> 

Relevant sections in book: Chapters 3, 19

(Illustrations from Ray Tracing From The Ground Up, Physically-Based Rendering, Fundamentals of Computer Graphics)

(page numbering might skip some slides due to 'hidden' slides in my presentation.)

Admin: Course Organization

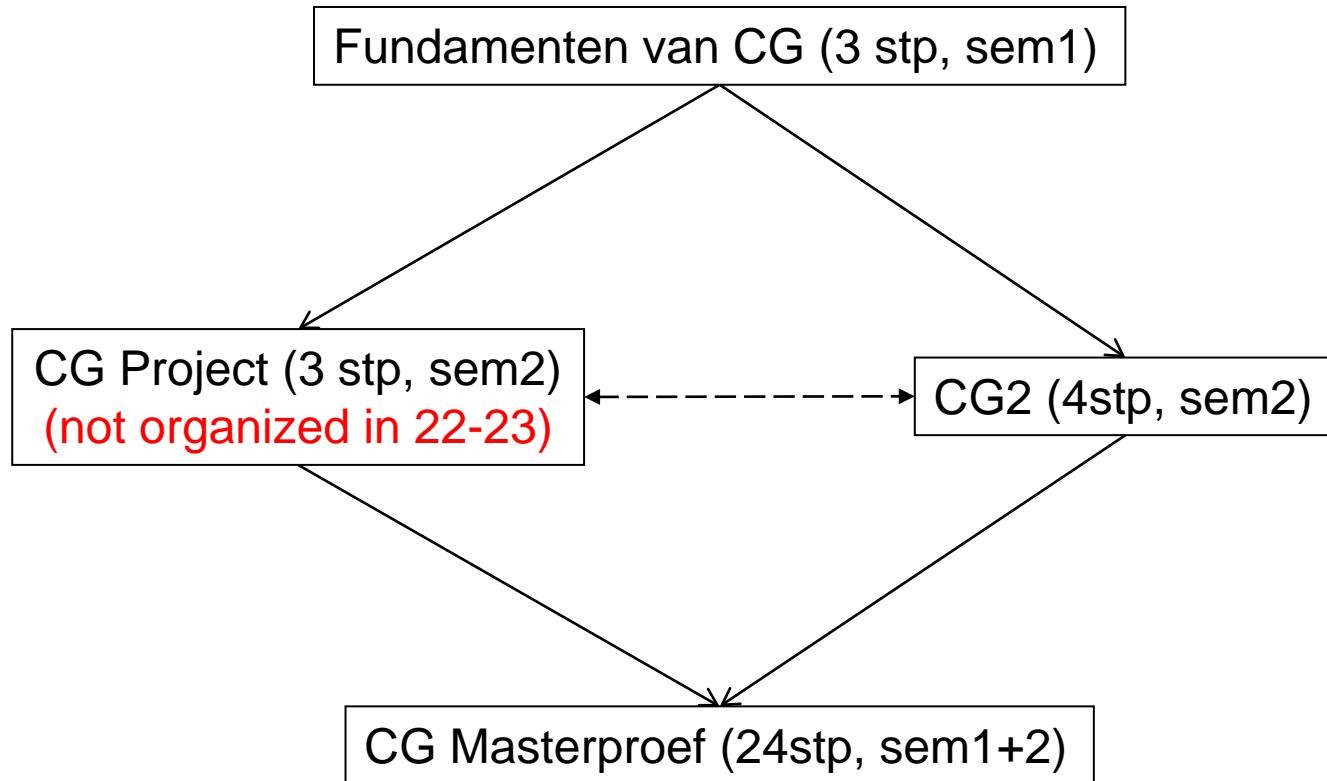
10 lectures, 2 hours each

No project for this course

Final evaluation for entire course

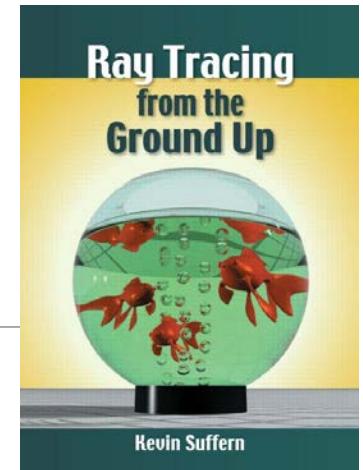
- Written exam during exam period (examples on TOLEDO)
- Closed book

Admin: Other Graphics Courses



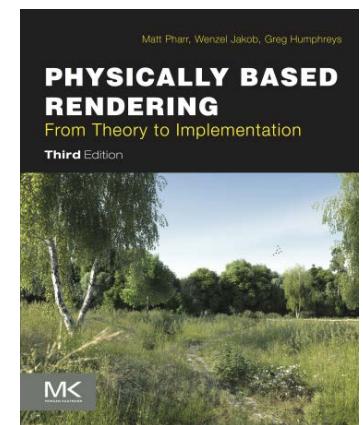
Admin: Course Materials

Ray Tracing from the Ground Up



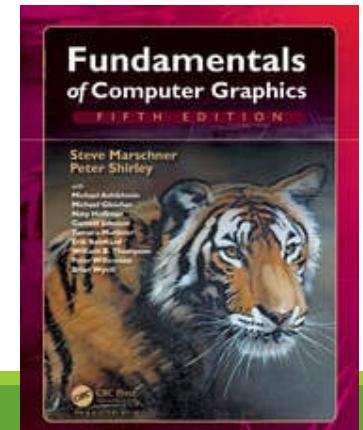
Other books are also usable:

- Physically Based Rendering
- Fundamentals of Computer Graphics
- ...
- (some study material is from these books)



Slides + additional course material available online

Course summary on Google Docs

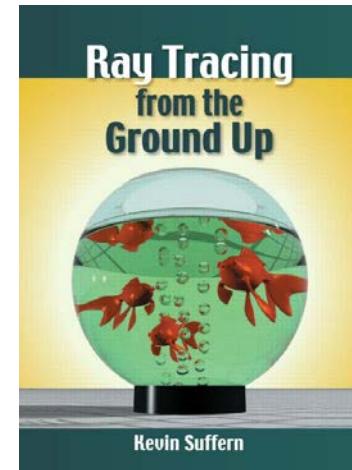


Admin: Course Materials

How will we use this book?

Lectures:

- Overall motivation, how things fit together
- Book: detailed explanation + code



What is Computer Graphics?

????

Computer graphics deals with generating [images](#) with the aid of [computers](#). Today, computer graphics is a core technology in digital photography, film, video games, cell phone and computer displays, and many specialized applications. A great deal of specialized hardware and software has been developed, with the displays of most devices being driven by [computer graphics hardware](#). It is a vast and recently developed area of computer science. The phrase was coined in 1960 by computer graphics researchers Verne Hudson and William Fetter of Boeing. It is often abbreviated as CG, or typically in the context of film as [computer generated imagery](#) (CGI). The non-artistic aspects of computer graphics are the subject of [computer science](#) research.^[1]

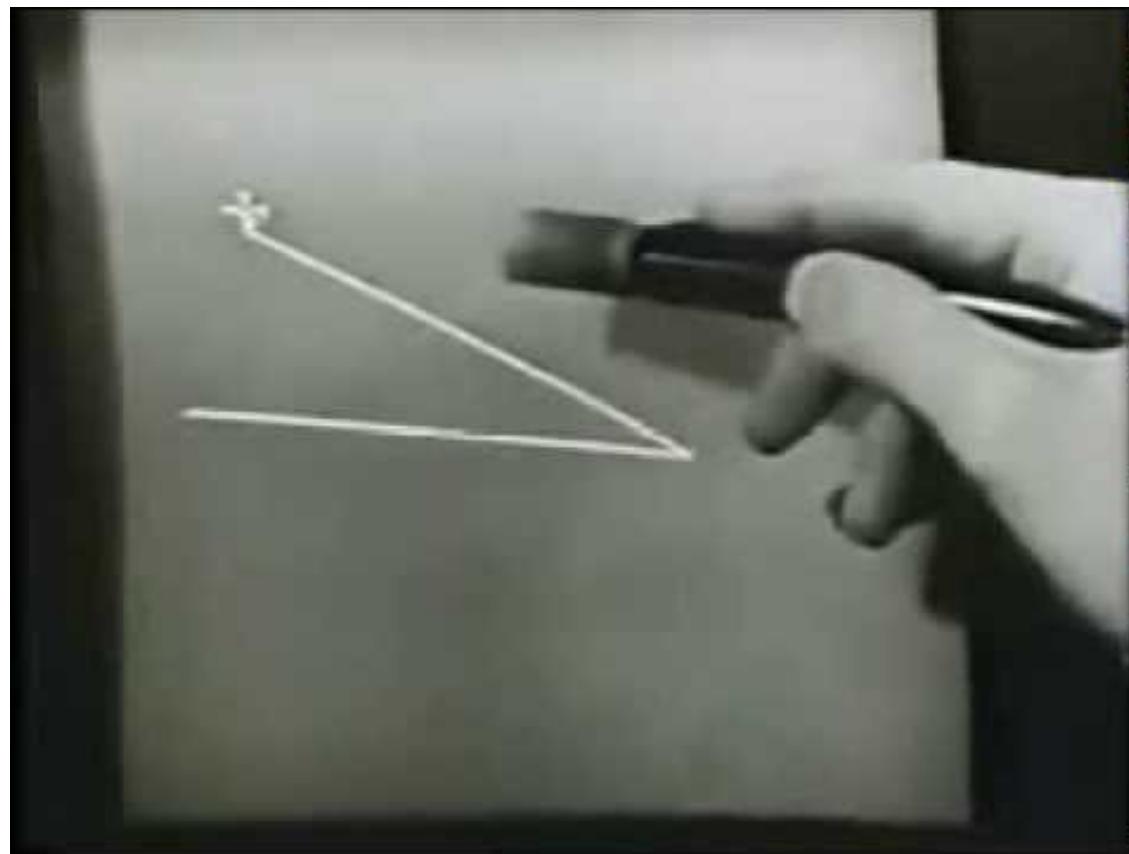
Some topics in computer graphics include [user interface design](#), [sprite graphics](#), [rendering](#), [ray tracing](#), [geometry processing](#), [computer animation](#), [vector graphics](#), [3D modeling](#), [shaders](#), [GPU design](#), [implicit surfaces](#), [visualization](#), [scientific computing](#), [image processing](#), [computational photography](#), [scientific visualization](#), [computational geometry](#) and [computer vision](#), among others. The overall methodology depends heavily on the underlying sciences of [geometry](#), [optics](#), [physics](#), and [perception](#).

(Wikipedia, Sept 2022)

What is Computer Graphics?

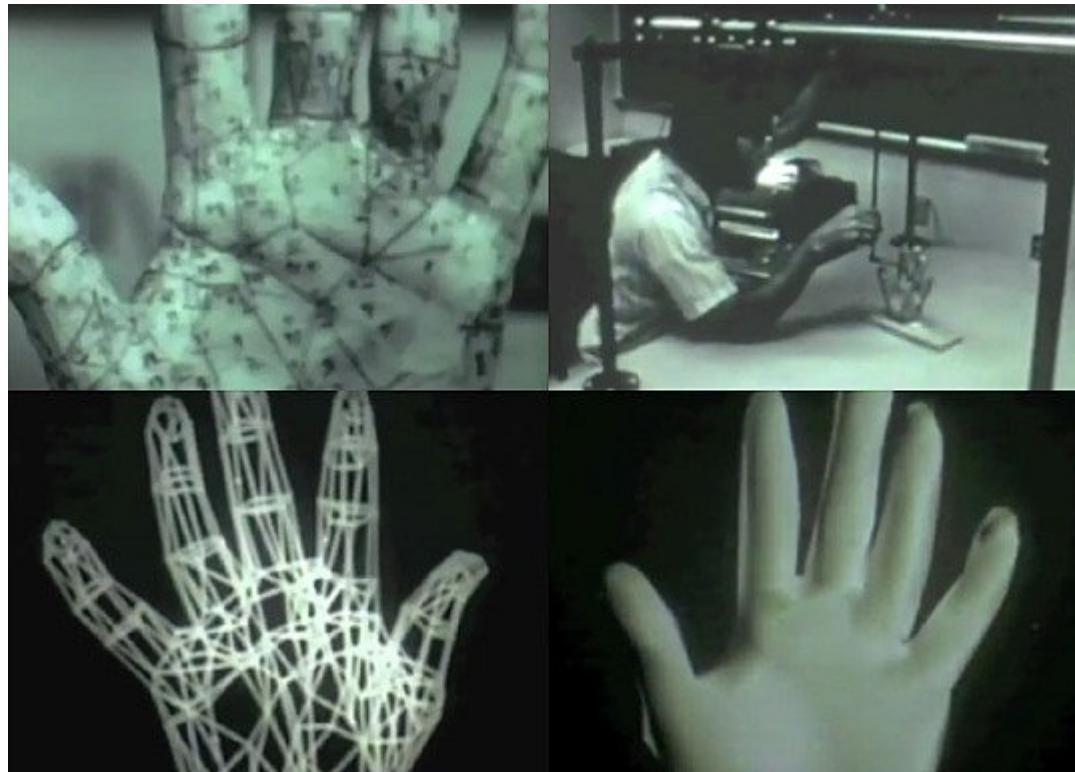
“Computer graphics enhances the human imagination.”

“Computer graphics enhances
the human imagination”



(Sketchpad, 1963)

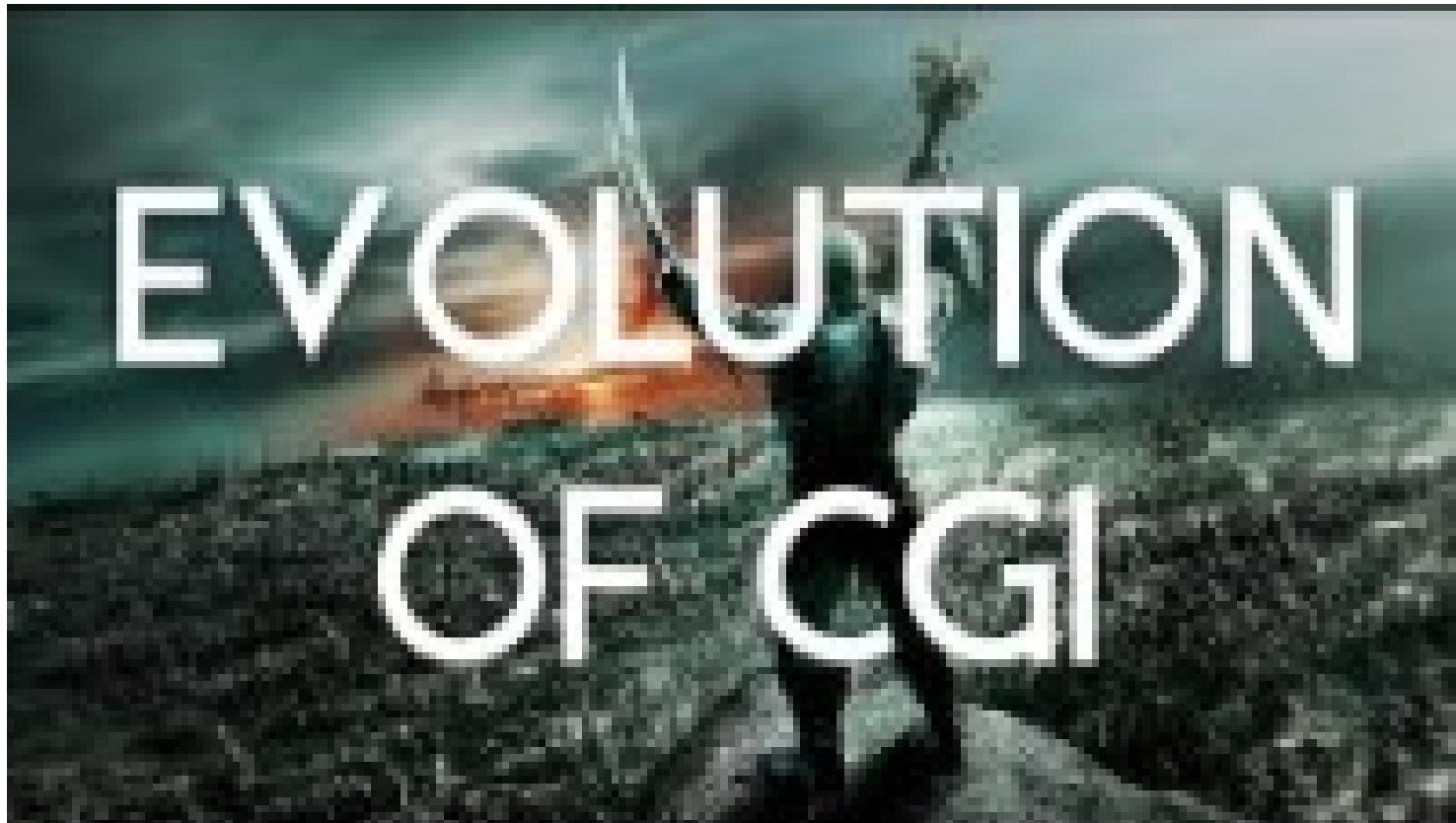
“Computer graphics enhances the human imagination”



(Ed Catmull, 1972)

(image sequence inspired by Nvidia launch event August 2018)

“Computer graphics enhances
the human imagination”



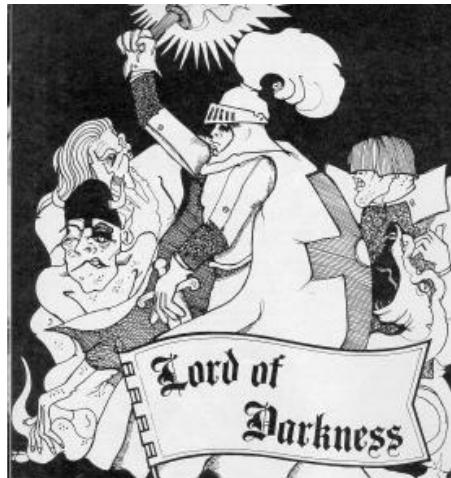
<https://youtu.be/pWqT8i4GaMo>

Games



<https://youtu.be/vUpv-qyxr3A>

Personal History ...



By Philip Dutre

Prepare to combat the dark forces! Philip Dutre has written us a 3D graphics adventure that combines animated graphics, menu driven controls and traditional text input — all this in a quest to seek out and destroy the Lord of Darkness before he gains control of Middle Earth. It's a long program, but definitely worth the effort to type it in.

Listing 1

This is the master Loader program which will load the parts of the finished game. Enter it and save it onto your game tape with the command: **SAVE "Lord"** LINE 1.

Listing 2

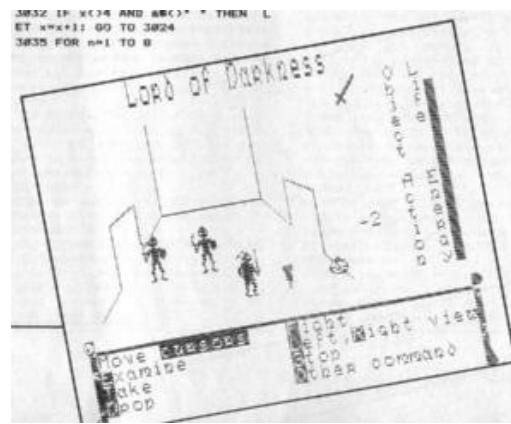
Type **NEW**, then enter this listing which contains the code for the new character. **RUN** it and when prompted **SAVE** the code onto your game tape. It's a good idea to make a separate copy of this program just in case you make a mistake in the data statements.

Listing 3

Type **NEW** again, and enter this listing. This contains data for 26 UDIST. **RUN** it, and save the code onto your game tape. You should also make a separate copy of this one too.

Listing 4

Now the Spectrum is all set, buy some sandwiches, and get ready to enter this listing. Don't **BURN** it, but **SAVE** it onto your game tape with: **SAVE "Lord"** LINE 1.



(ZX Computing, 1985)

Personal History ...

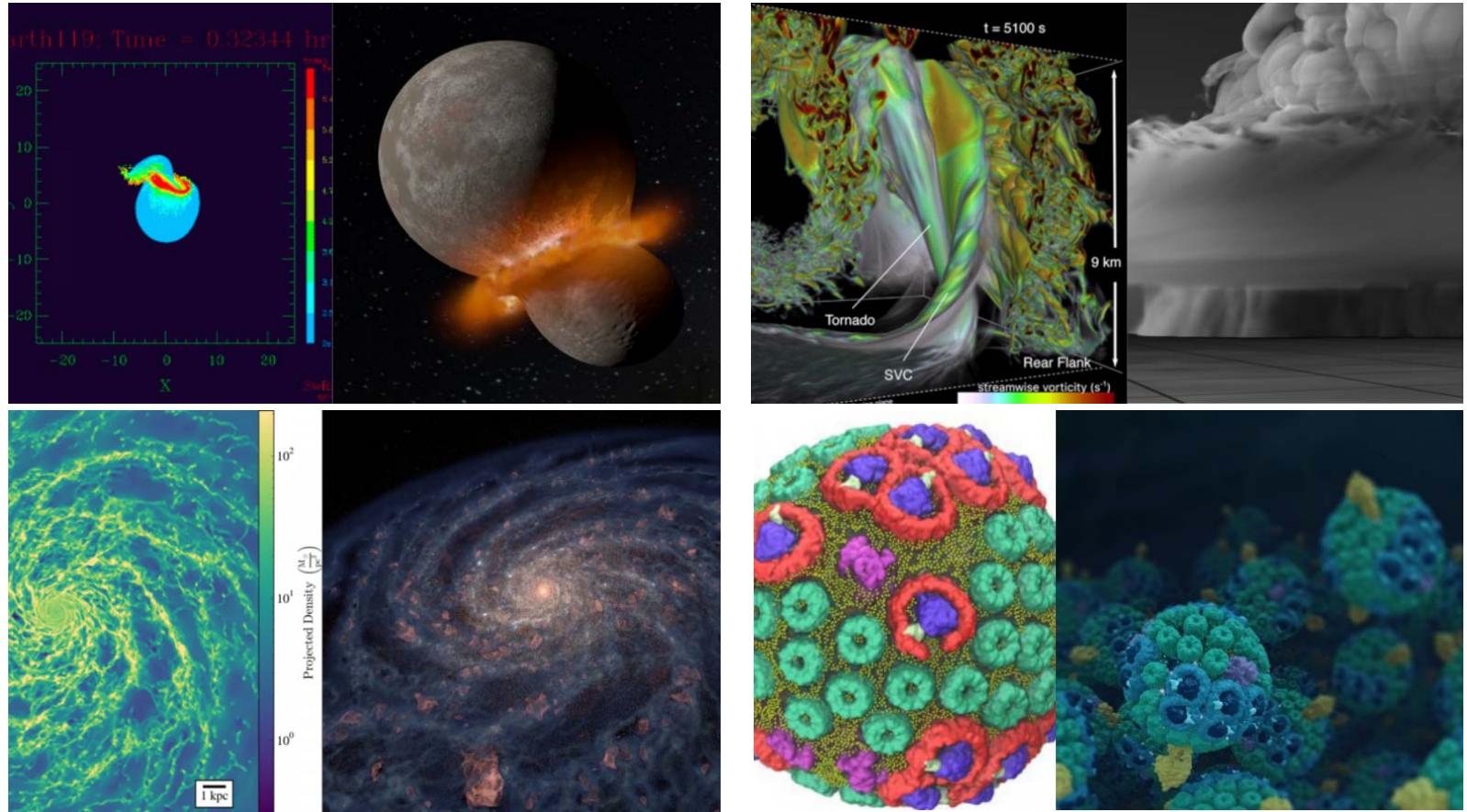


Medical Imaging



<https://youtu.be/QoWSgzrtjU>

Scientific Visualization



<https://blog.siggraph.org/2019/07/why-cinematic-scientific-visualization-is-more-important-than-ever.html/>

Architecture



Interior Design

When You Flip Through an IKEA Catalog, 75% of the 'Photography' You See is CGI

Aug 28, 2014

DL CADE

Share 255

Tweet

19 COMMENTS

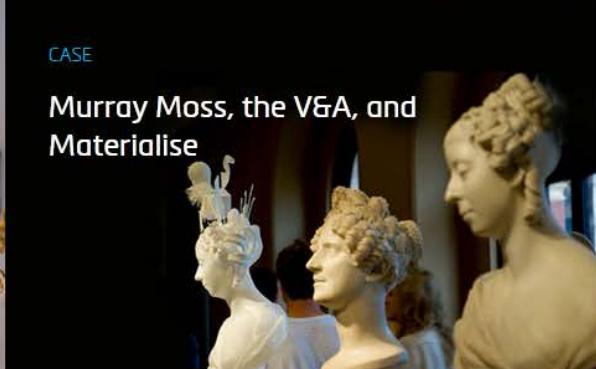


Digital Art



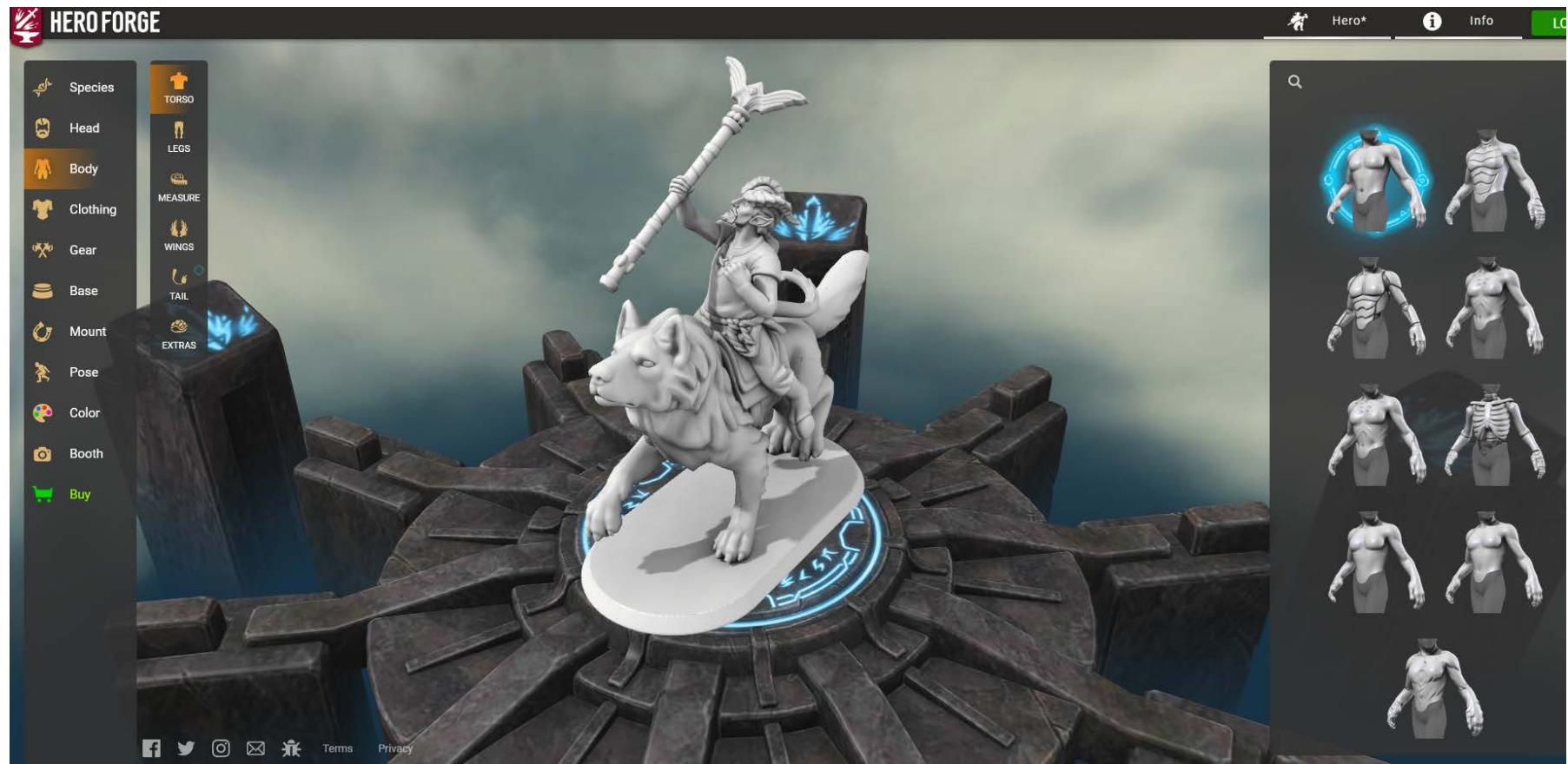
(SIGGRAPH 2019 Art Gallery Preview)

Art & 3D printing



(www.materialise.com)

Online Design



<https://www.heroforge.com/>

Some Recent Research



What will we learn in this course?

Fundamentals of computer graphics rendering algorithms

- Entire algorithmic pipeline from 3D model to 2D image
- Not: Specific APIs; Graphics applications (game design, Photoshop, ...); artistic skills; ...



What will we learn in this course?

Fundamentals of computer graphics rendering algorithms

- Entire algorithmic pipeline from 3D model to 2D image
- Focus on mathematics and physics
- Strong focus on modeling & generating images
- Less (or no) focus on animation

- Main algorithmic approach: ray tracing

3D Computer Graphics

Modeling

Rendering

Animation

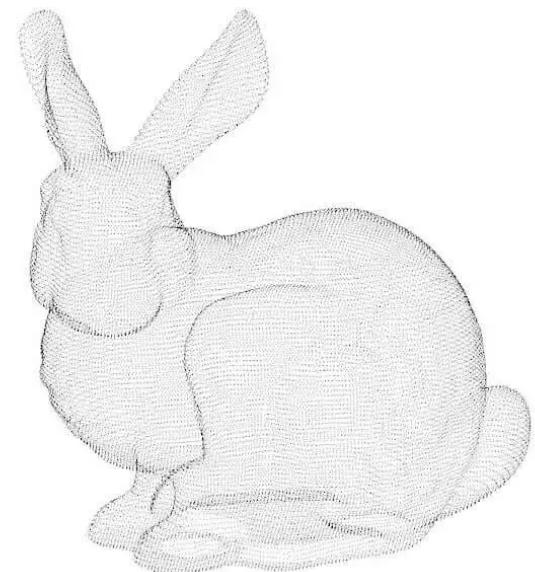
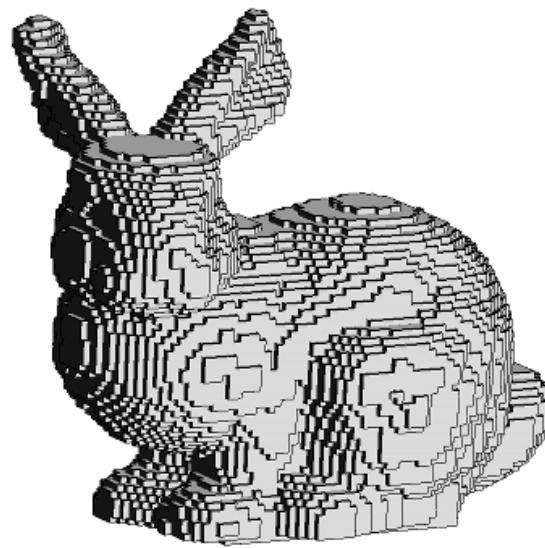
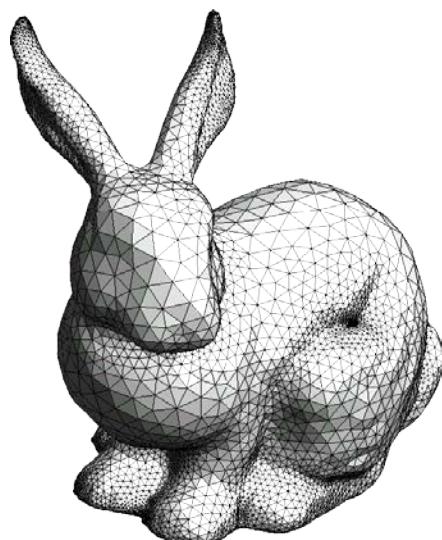
3D Computer Graphics

Modeling

- Representation of 3D models, capturing, manipulation, scanning, ...

Rendering

Animation



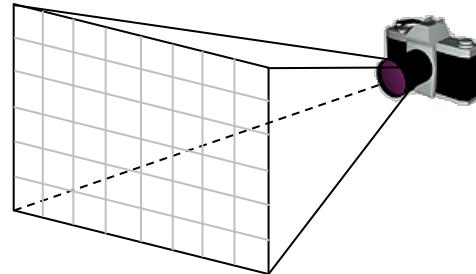
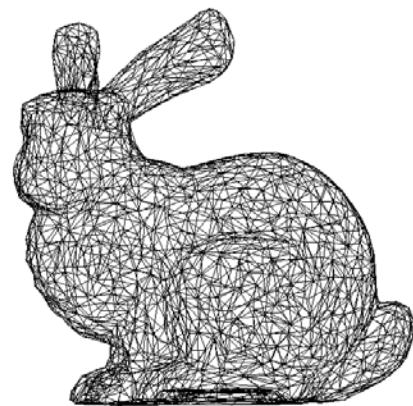
3D Computer Graphics

Modeling

Rendering

- 3D-to-image, shading, textures, materials, ...

Animation



3D Computer Graphics

Modeling

Rendering

- 3D-to-image, **shading**, textures, materials, ...

Animation



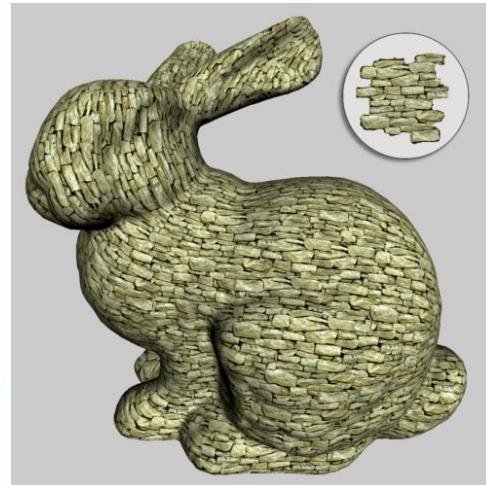
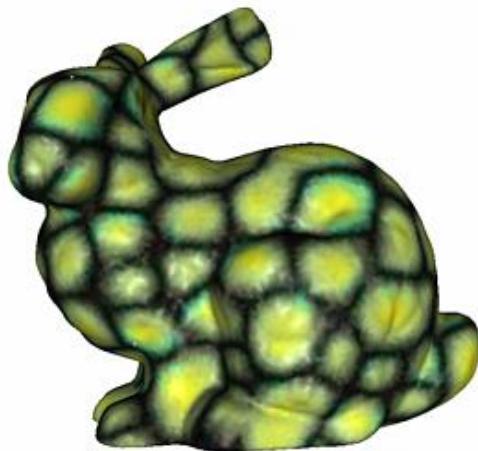
3D Computer Graphics

Modeling

Rendering

- 3D-to-image, shading, **textures**, materials, ...

Animation



3D Computer Graphics



<https://blenderartists.org/t/bunny-stanford-stress-test/698581>

3D Computer Graphics

Modeling

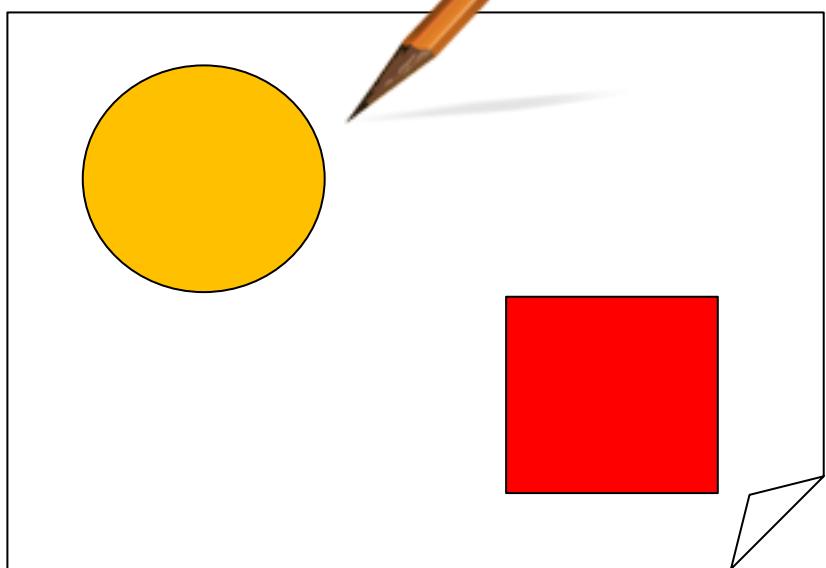
Rendering

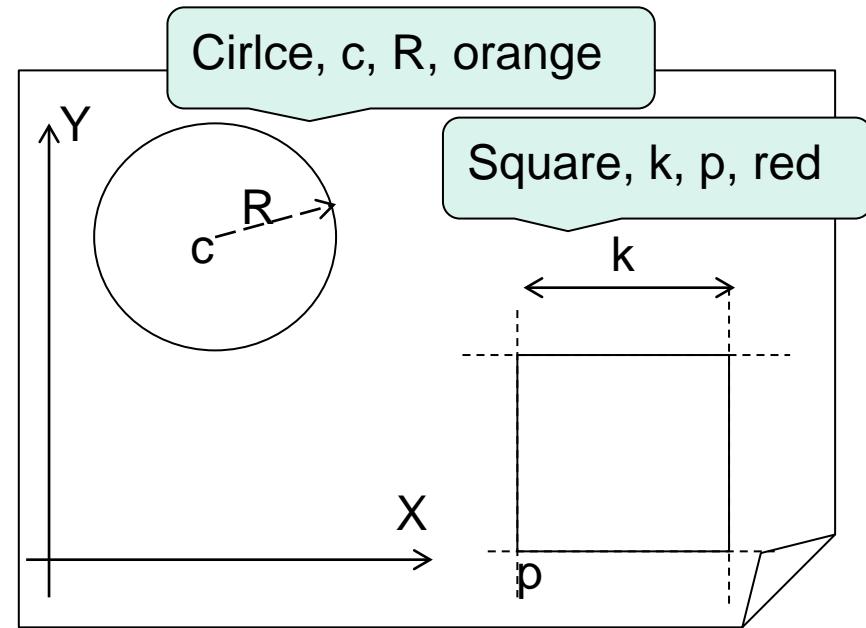
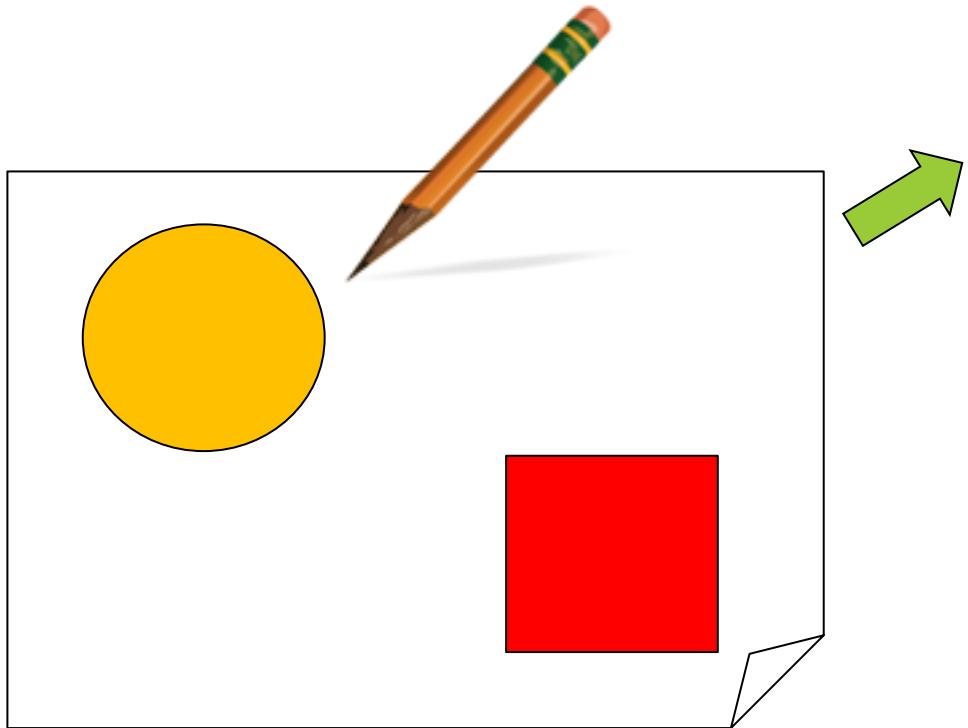
- 3D-to-image, shading, textures, materials, ...

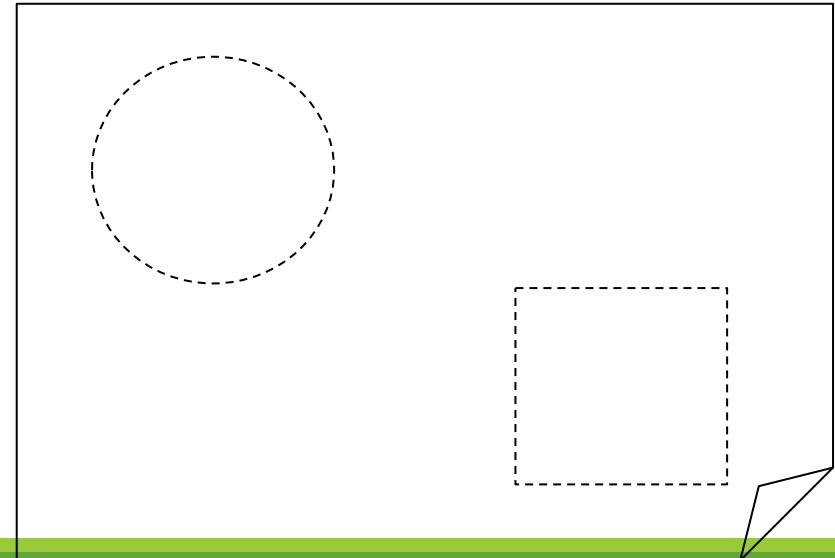
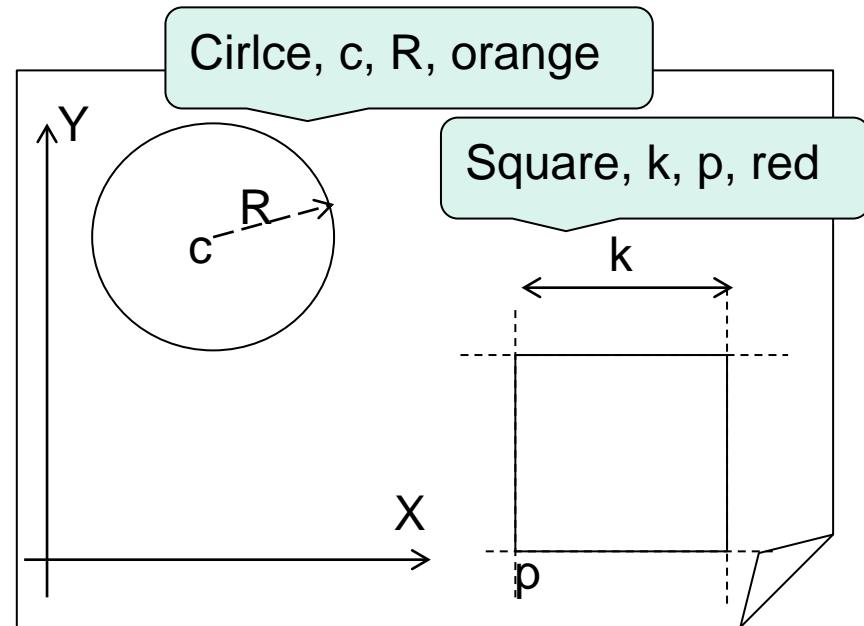
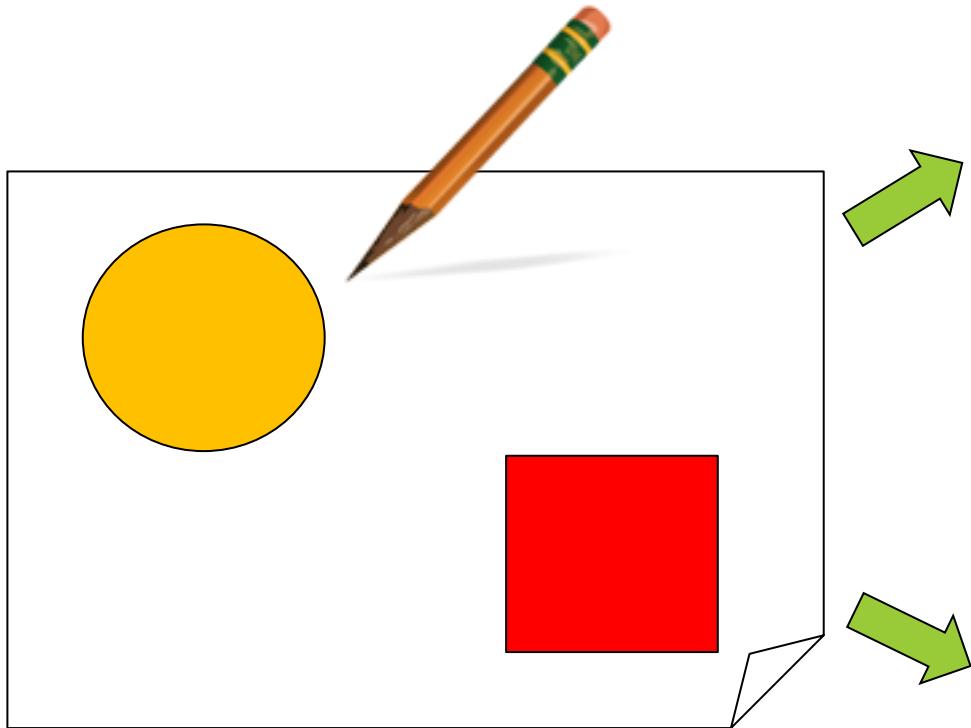
Animation

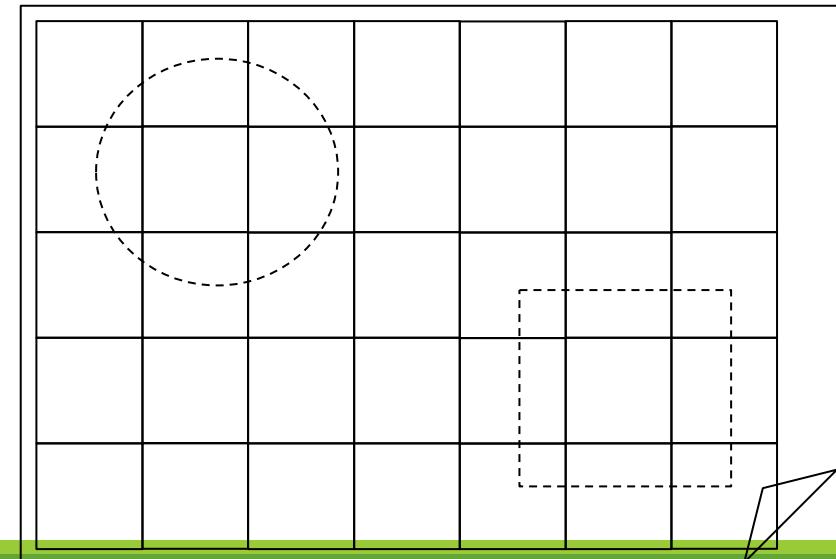
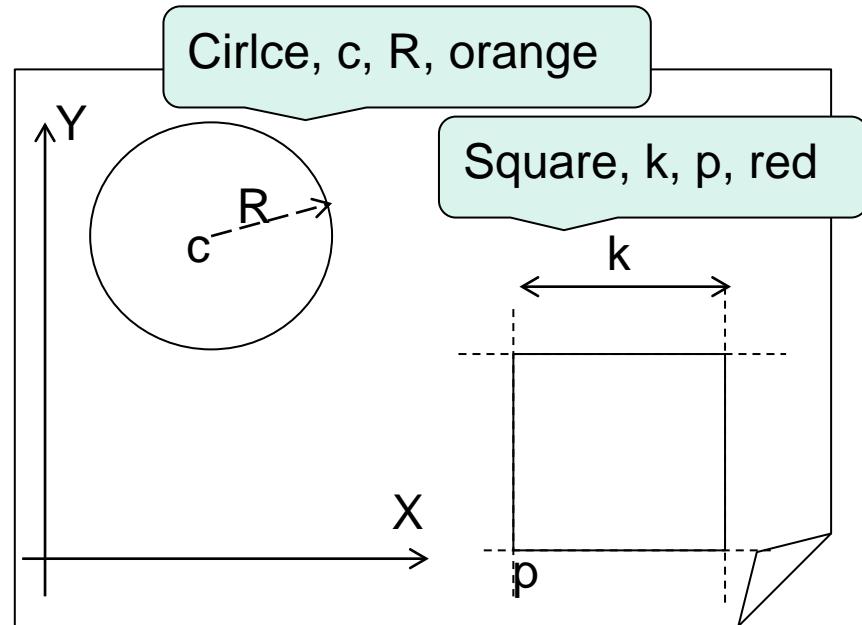
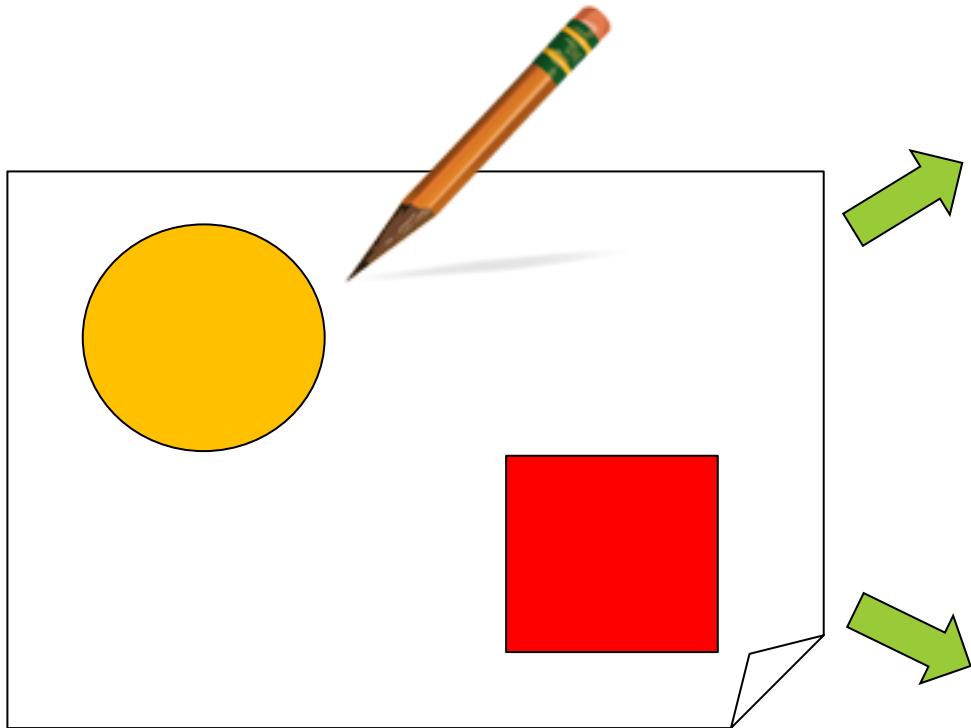


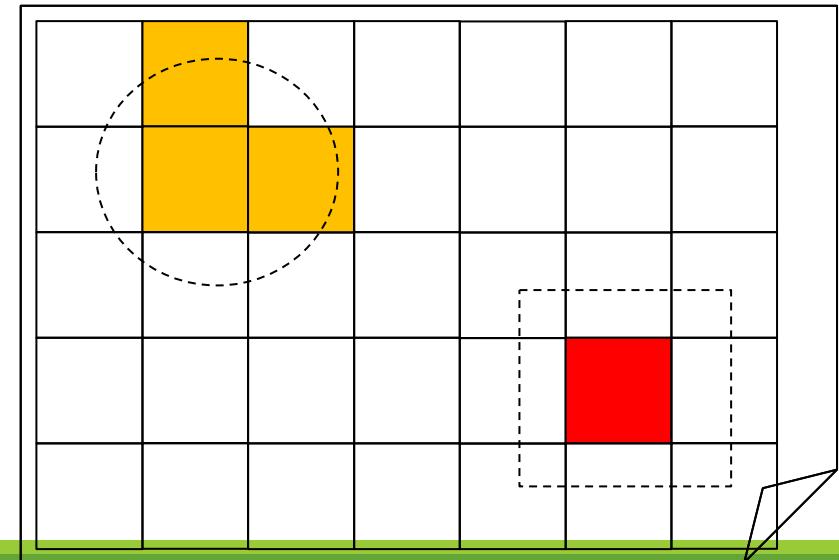
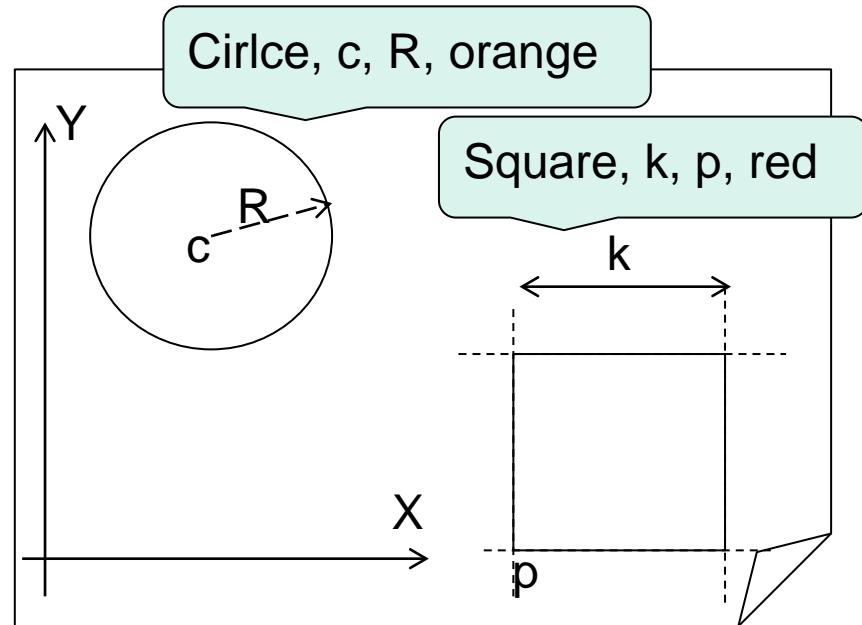
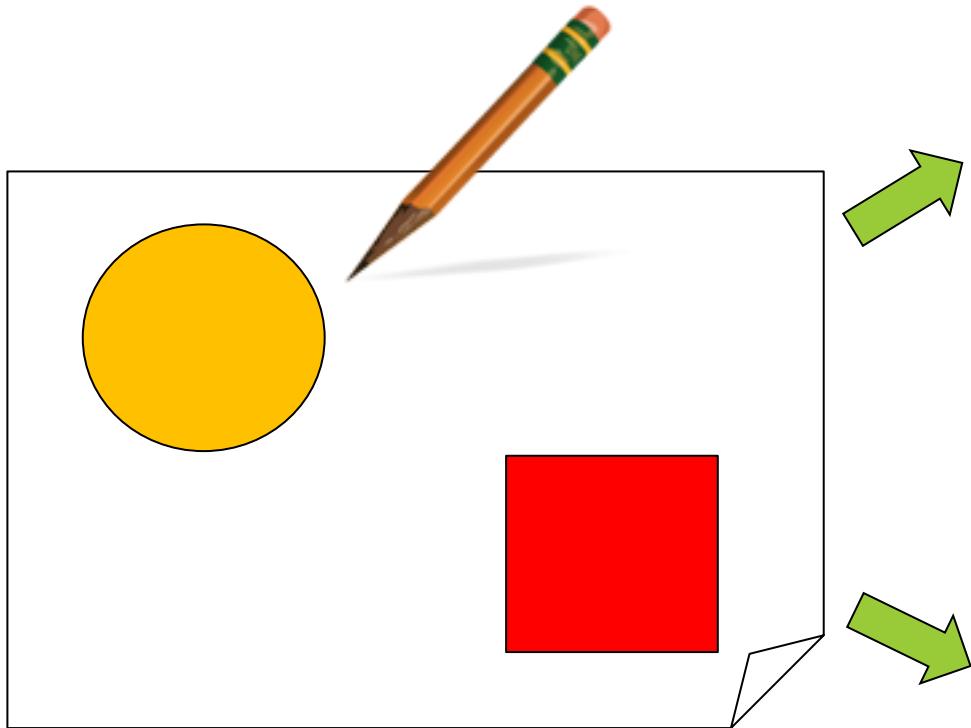
Preliminaries: Images

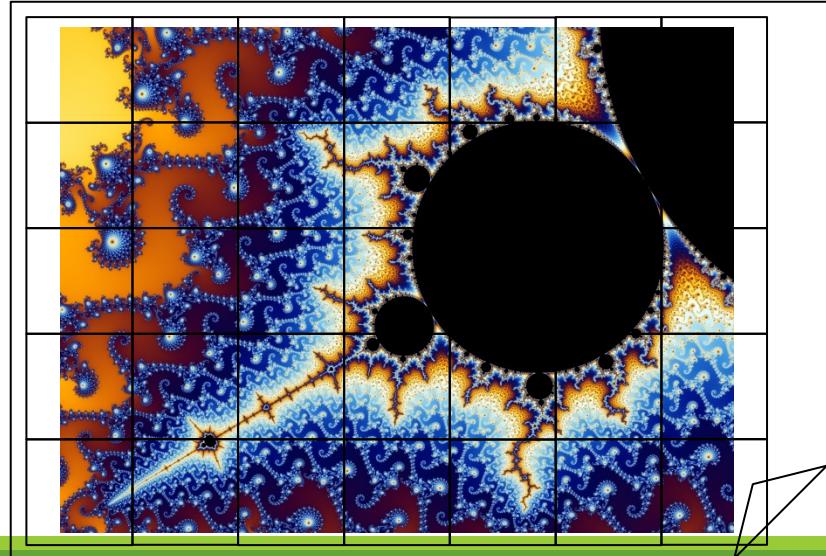
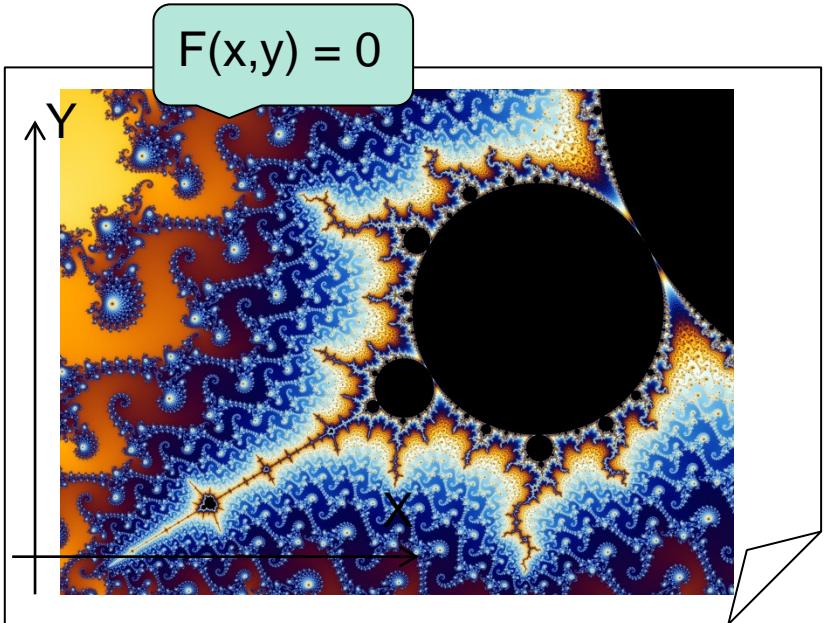
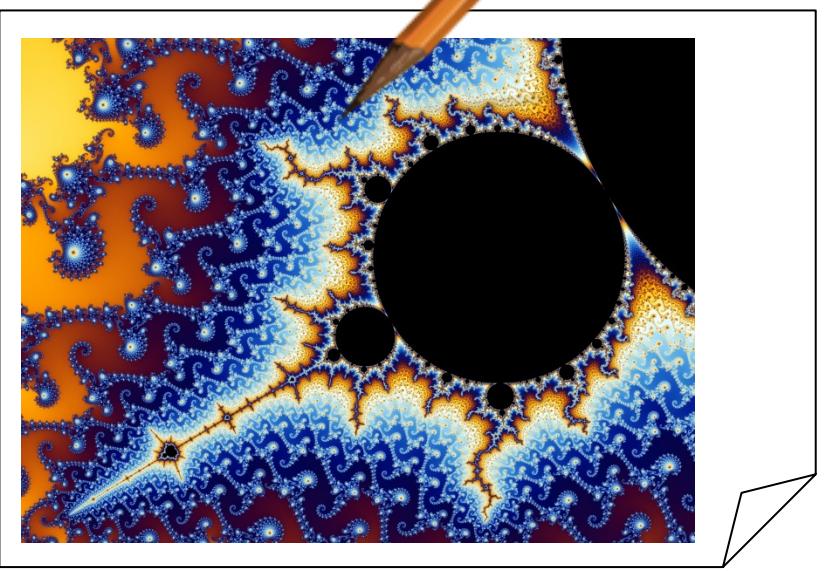


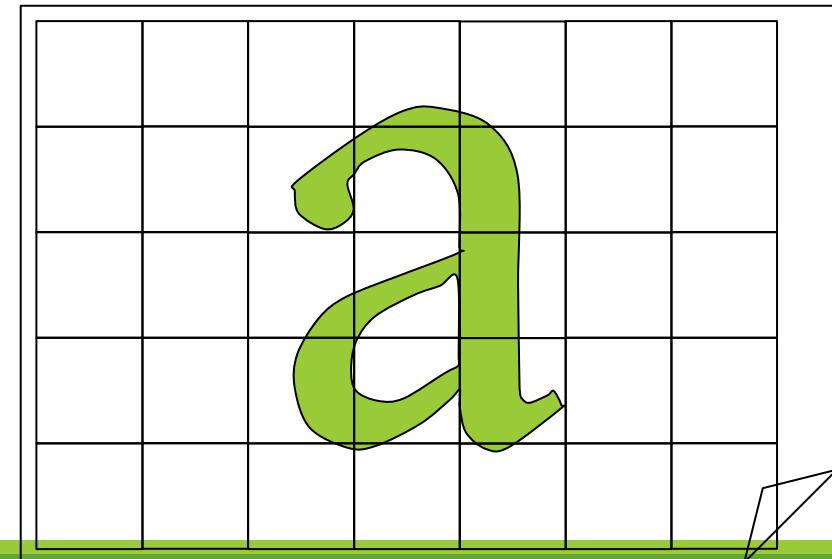
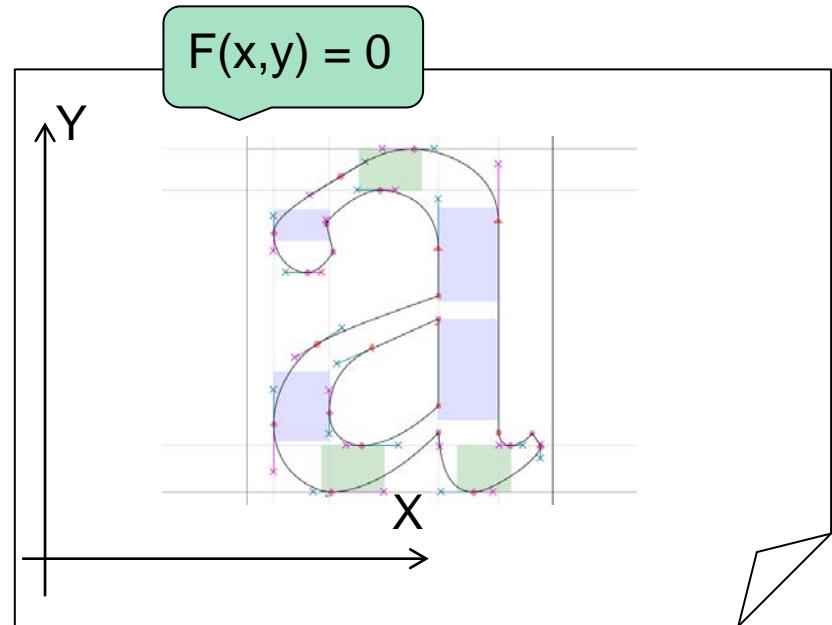
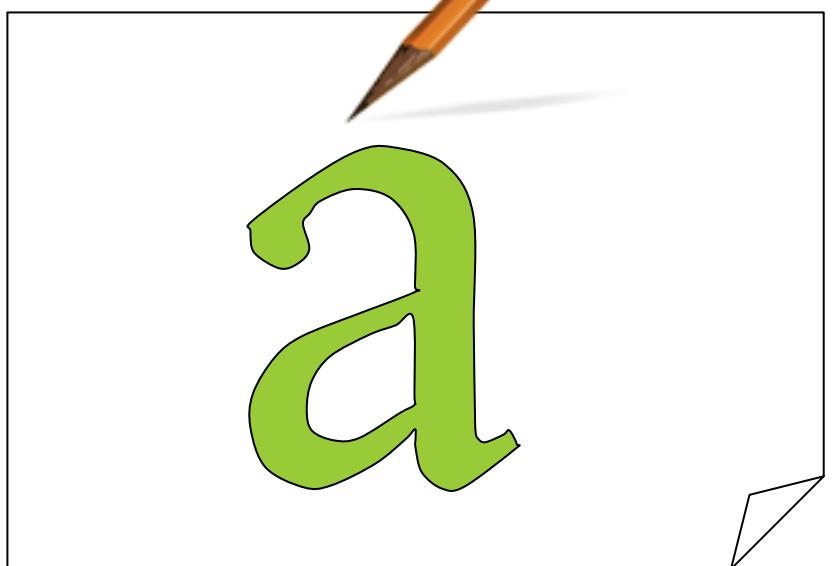












Raster image representation

2D array of numbers

Major advantage: arbitrary content

- approximate arbitrary functions with increasing resolution
- works because memory is cheap (brute force approach!)

Disadvantage: some operations are difficult

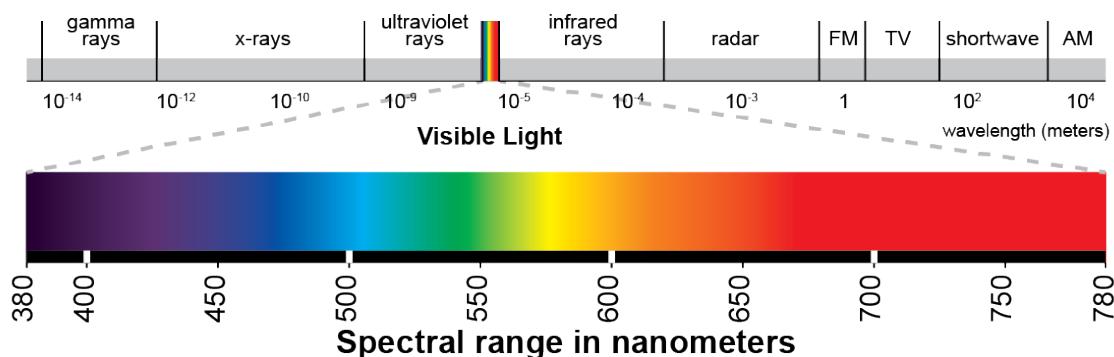
- E.g. scaling / zooming

Preliminaries: Colors

Colors

Color science

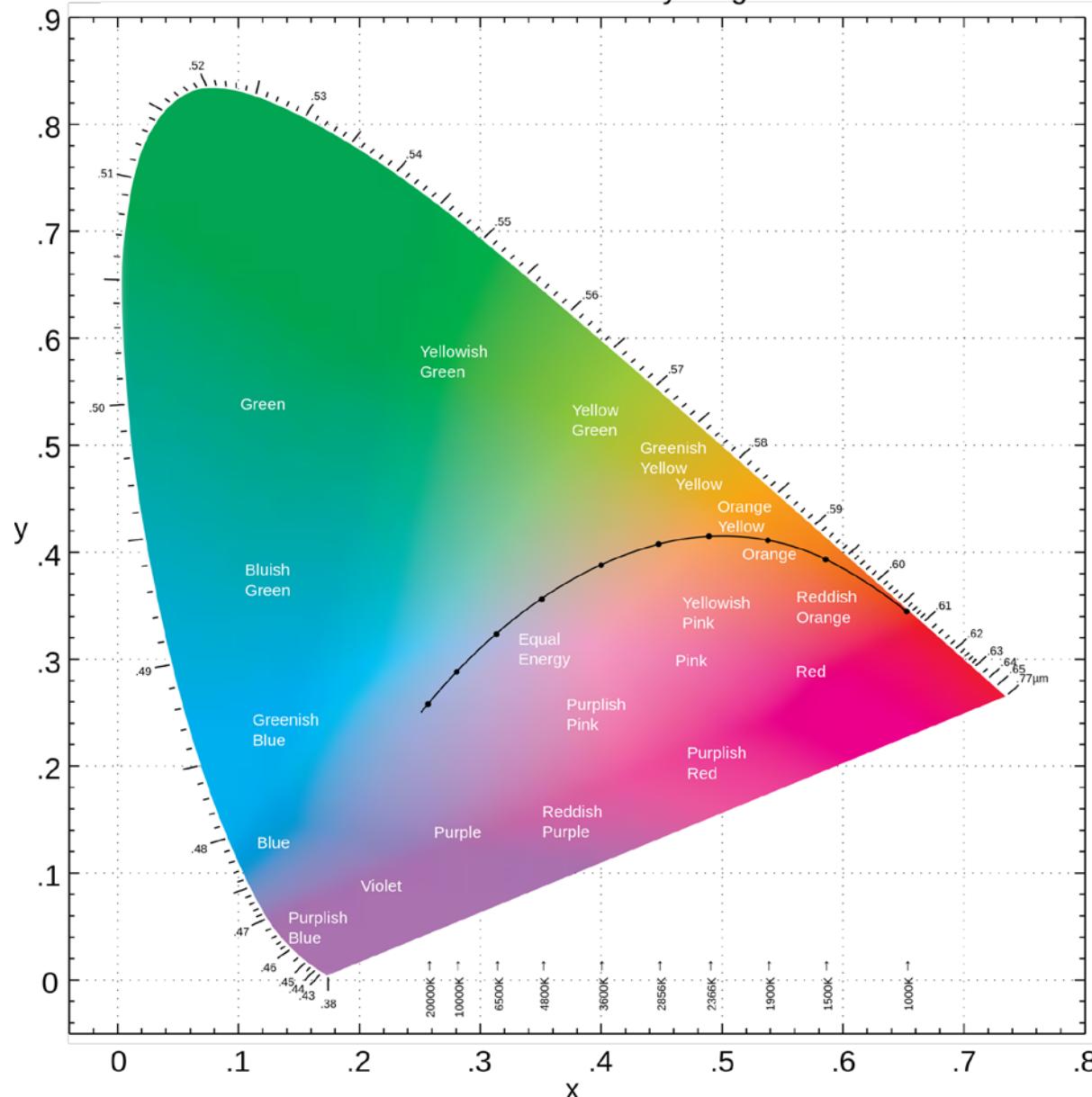
- Physics
 - Radiometry
 - Light reflection
- Perception
 - (Human) Eye
- Psychology
 - Meaning, mood, ...
- ...



color EMOTION GUIDE

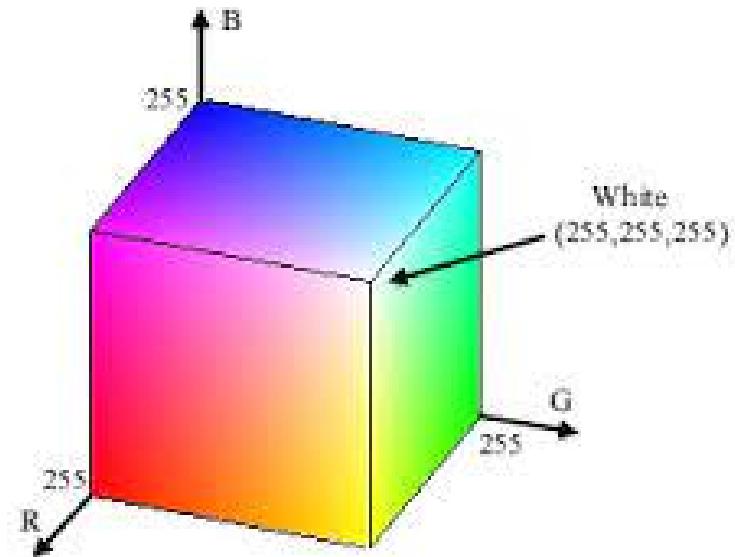
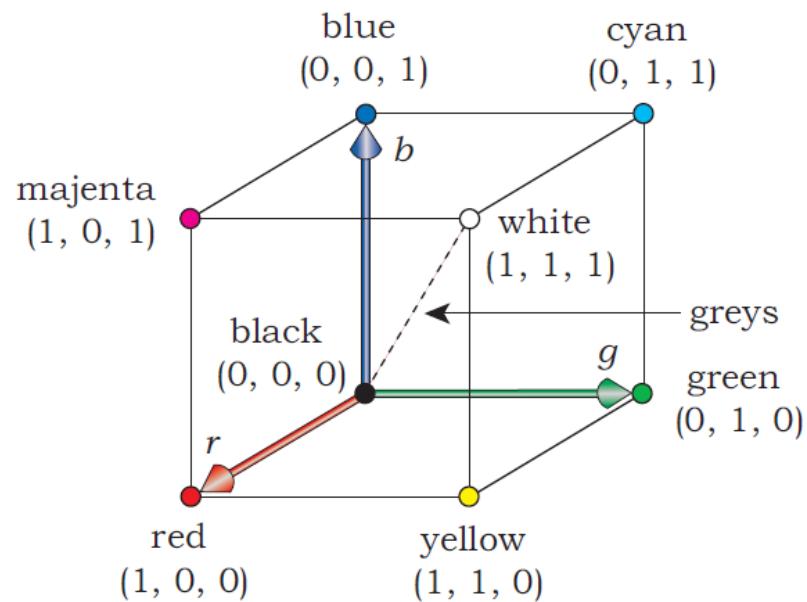
RED	EXCITEMENT, PASSION, STRENGTH, ENERGY, BOLD
ORANGE	CHEERFUL, ENTHUSIASTIC, FUN, CONFIDENCE, PLAYFUL
YELLOW	HOPE, FRIENDSHIP, WARMTH, JOY, OPTIMISM
GREEN	GROWTH, HARMONY, HEALTH, LUCK, PEACEFUL
BLUE	TRUST, LOYALTY, DEPENDABLE, RELAXED
BROWN	STABLE, COMFORTABLE, NATURE
BLACK	ELEGANCE, POWER, FORMAL, SOPHISTICATED, STRONG
GREY	BALANCE, NEUTRAL, CALM, SUBTLETY

C.I.E. 1931 Chromaticity Diagram



Representing colors

Red-green-blue additive color model



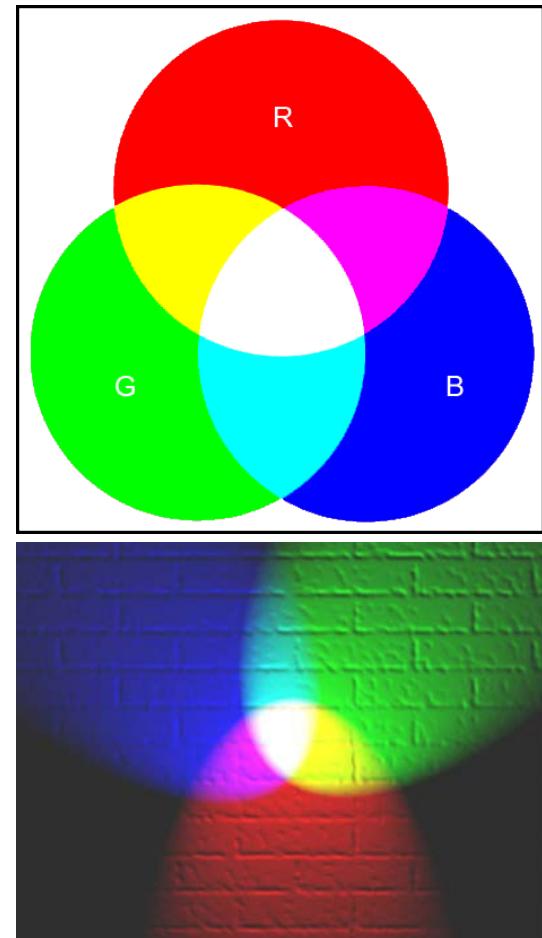
Not all visible colors can be represented by (positive) r,g,b triplets

Representing colors

Additive RGB model:

- red + green = yellow
- green + blue = cyan
- blue + red = magenta
- red + green + blue = white

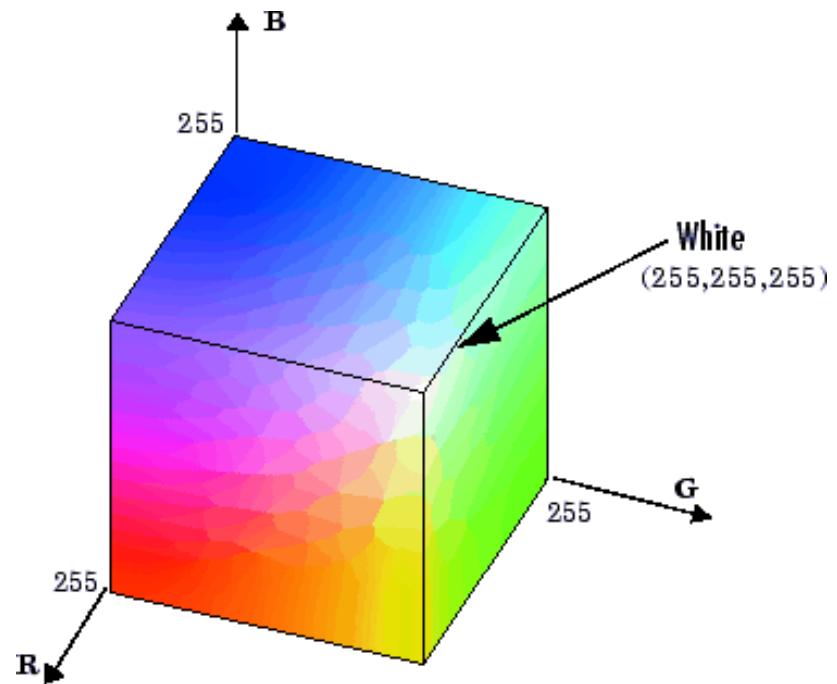
Approximation of true wavelength model of colors



Representing colors

Per pixel in an image:

- 1 byte for each color channel:
 - 256 different values each for red, green, blue
 - 16,777,216 different colors
 - ➔ “24 bit color”
- More bits / color channel are possible
 - Some file formats use floats ...



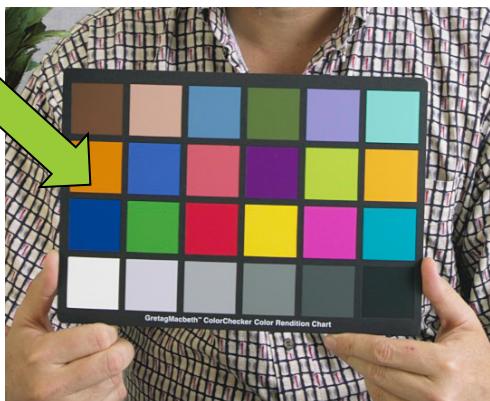
Representing colors

Be careful!

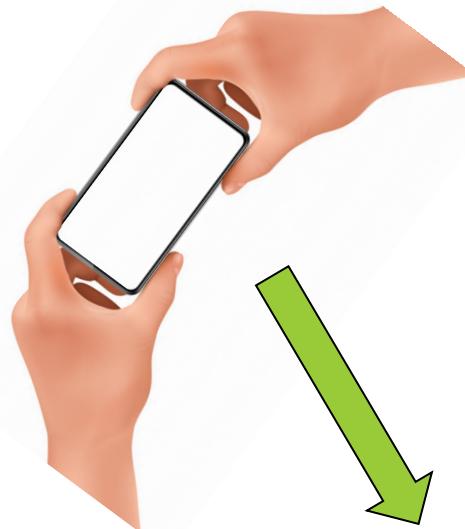
- Storing colors <> displaying colors <> capturing colors ...
- “What color do I see when I look at a pixel with color (255,0,0) on my laptop? On my smartphone? On a printer? On a projector screen?”
- “What light spectrum did my camera detect when it attributes (255,0,0) to a pixel? Is this the same color when displaying (255,0,0) on a screen?”



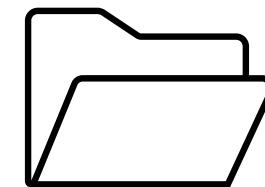
1. Light reflects on colored object



2. Capture colors with camera



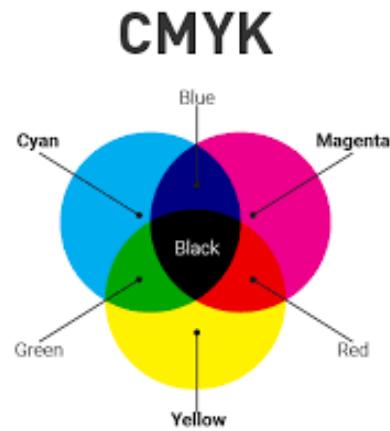
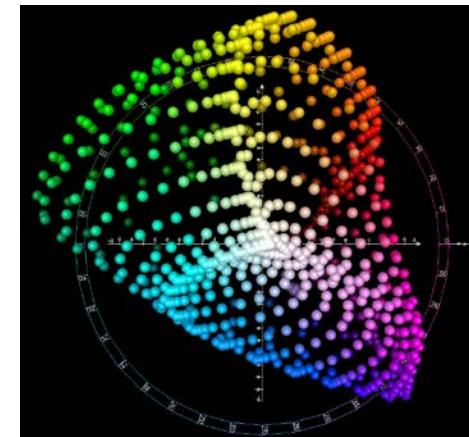
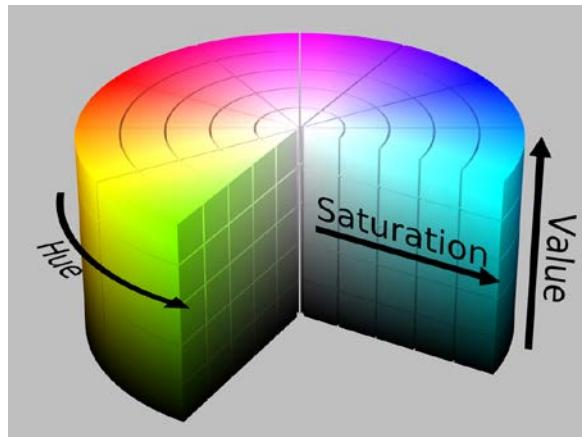
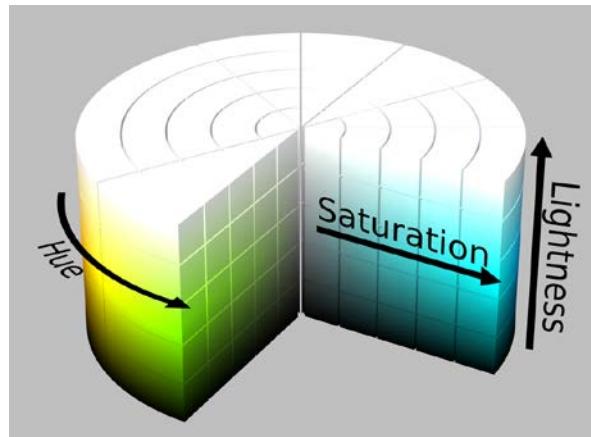
Do I see the same colors?



3. Store colors as numbers on file

4. Show colors on screen

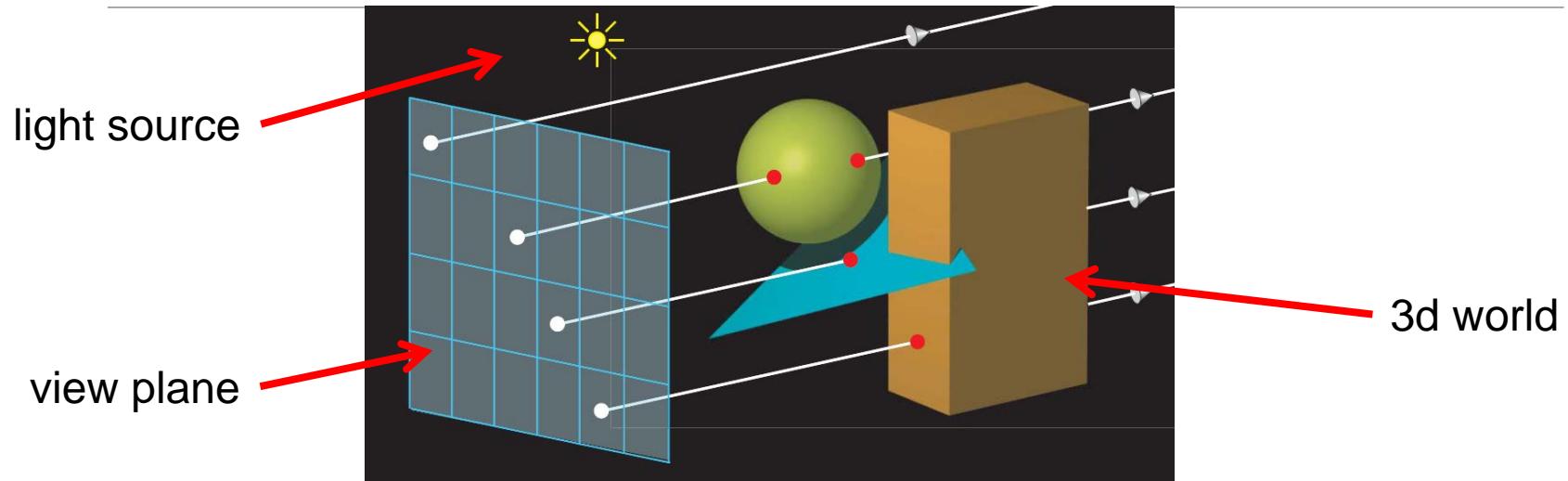
Many, many color models ...



1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010
1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023
1024	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035
2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010
2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
2029	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020
3022	4004	4005	4006	4007	4008	4009	4010	4011	4012	4003
4004	4005	4006	4007	4008	4009	4010	4011	4012	4009	4011
5001	5002	5003	5004	5005	5007	5008	5009	5010	5011	5012
5012	5013	5014	5015	5017	5018	5019	5020	5021	5022	5023
5023	5024	5025	5026	5029	5031	5032	5033	5034	5035	5036
6006	6007	6008	6009	6010	6011	6012	6013	6014	6015	6016
6016	6017	6018	6019	6020	6021	6022	6023	6024	6025	6026
6027	6028	6029	6032	6033	6034	6035	6036	7009	7011	7012
7003	7004	7005	7006	7008	7009	7010	7011	7012	7013	7014
7013	7015	7019	7021	7022	7023	7024	7026	7030	7031	7032
7032	7033	7034	7035	7036	7037	7038	7039	7043	7042	7043
7043	7044	7045	7046	7047	7048	8000	8001	8002	8003	8004
8004	8007	8008	8011	8012	8014	8015	8016	8017	8019	8020
8022	8023	8024	8025	8028	8029	9001	9002	9003	9004	9023
9005	9006	9007	9010	9011	9015	9017	9018	9022	9023	9023

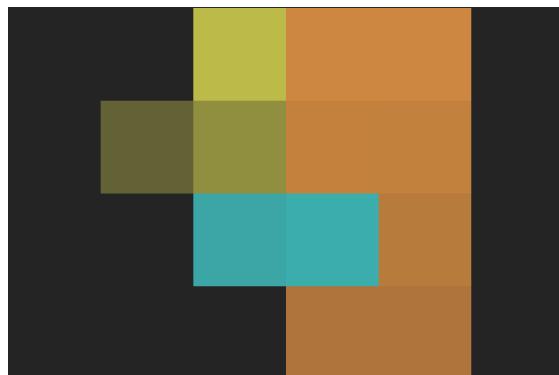
Ray Tracing: Basics

How Ray Tracing Works

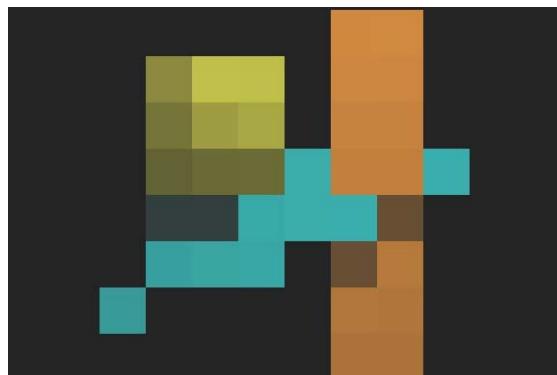


for each pixel
shoot ray through pixel
compute nearest hitpoint
compute color of hitpoint
pixel \leftarrow color

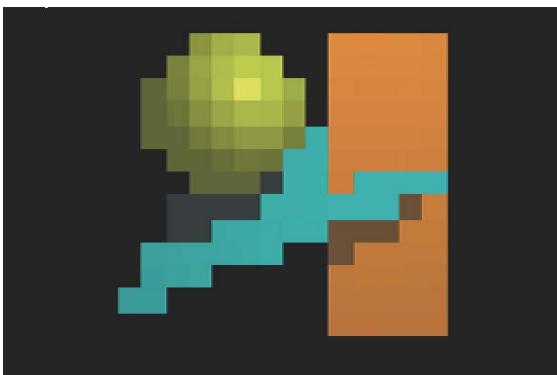
How Ray Tracing Works



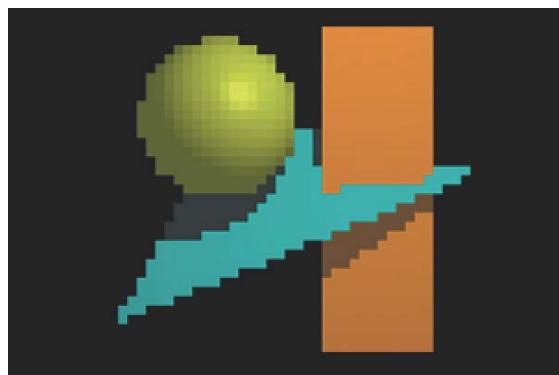
6x4



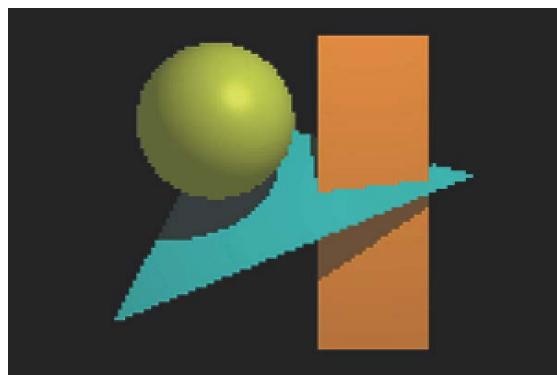
12x8



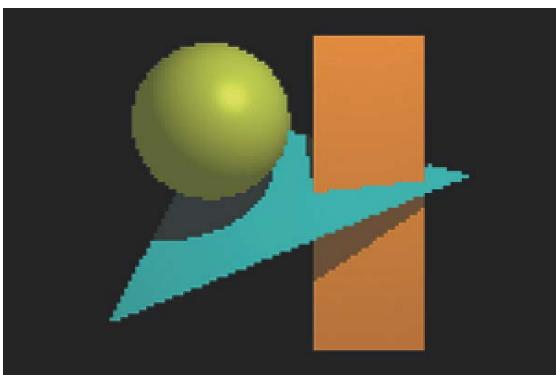
24x16



60x40



120x80



150x100

Ray Tracing by Dürer

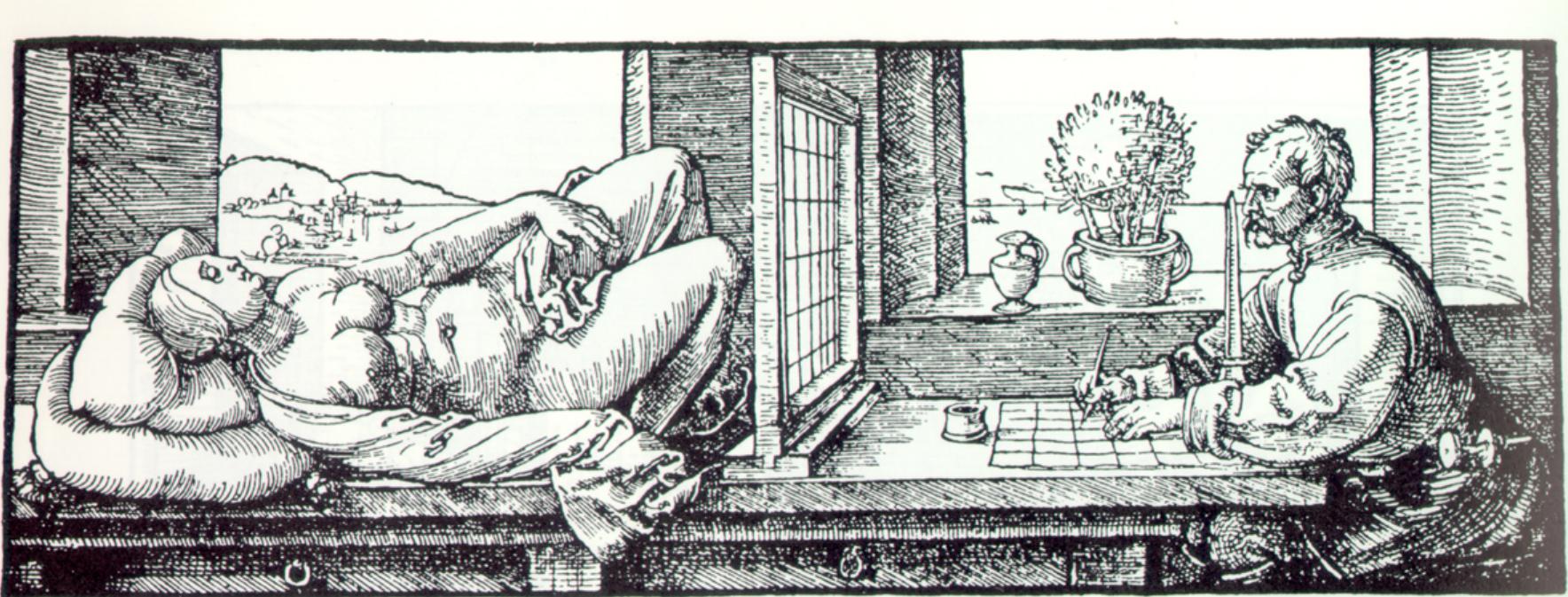
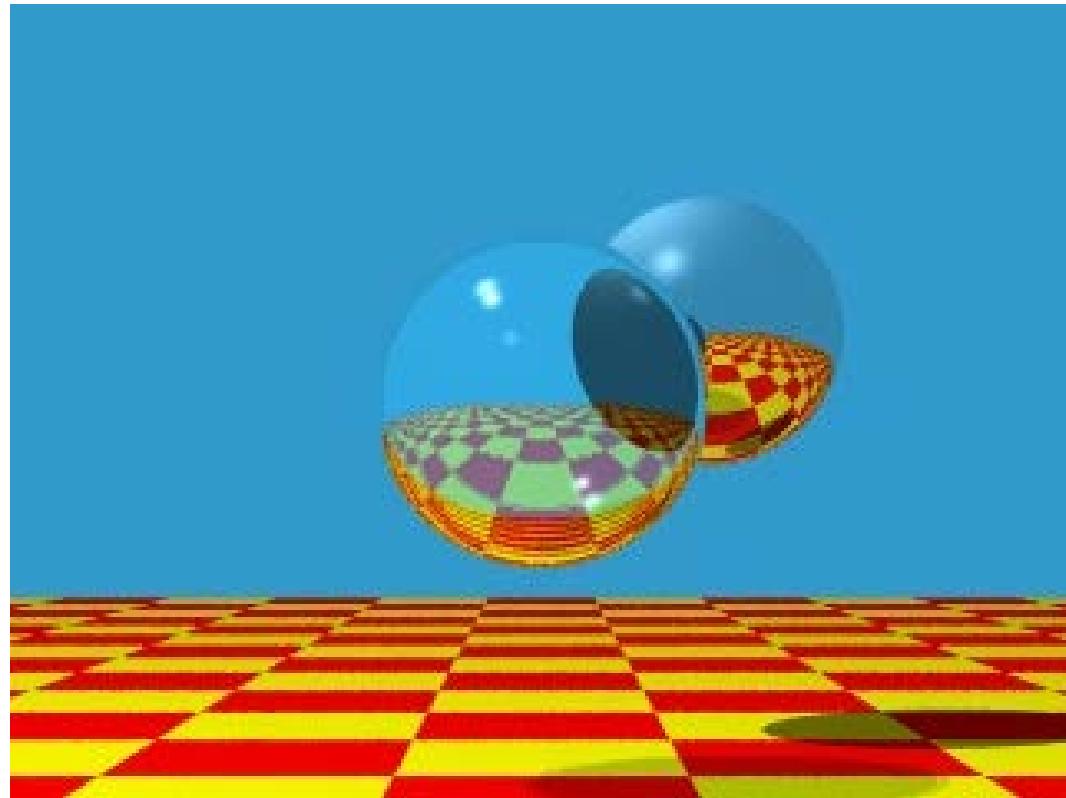
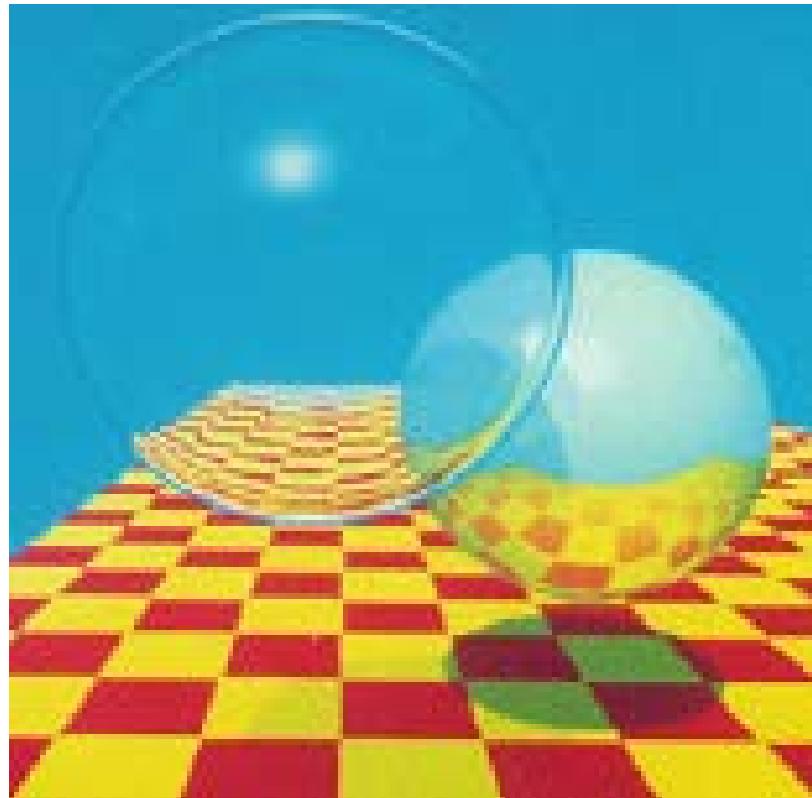


Fig. 6.9 Albrecht Dürer, illustration showing a ‘veil’ being used to draw a perspective image of a naked woman. From his *Underweysung der Messung mit dem Zirkel und Richtscheit* (Nuremberg, 1525), Book 3, Figure 67.

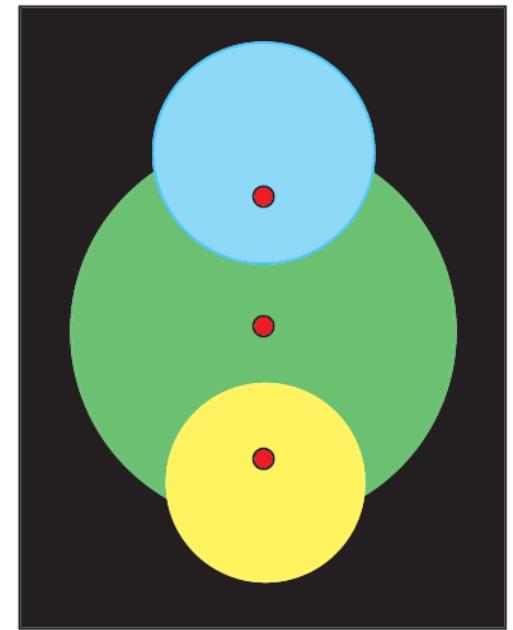
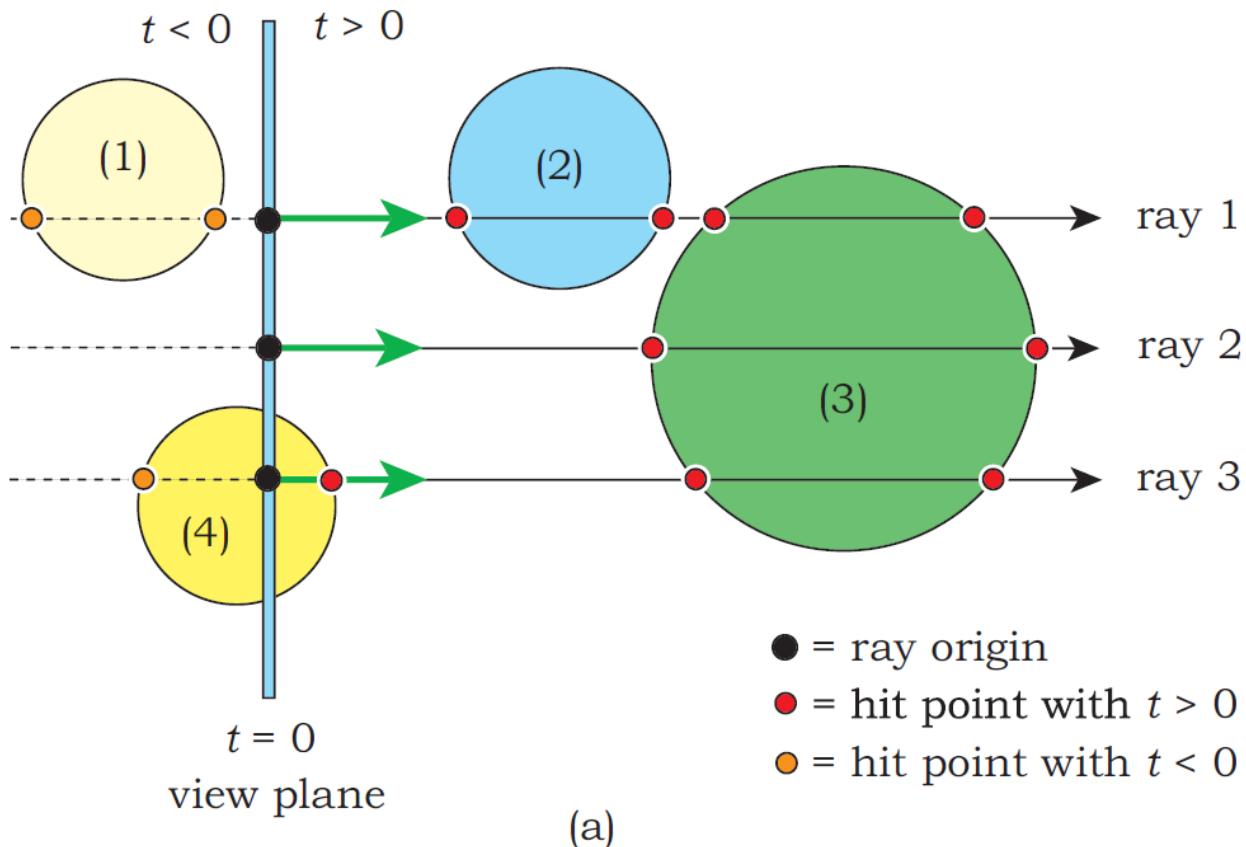
Ray Tracing by Whitted



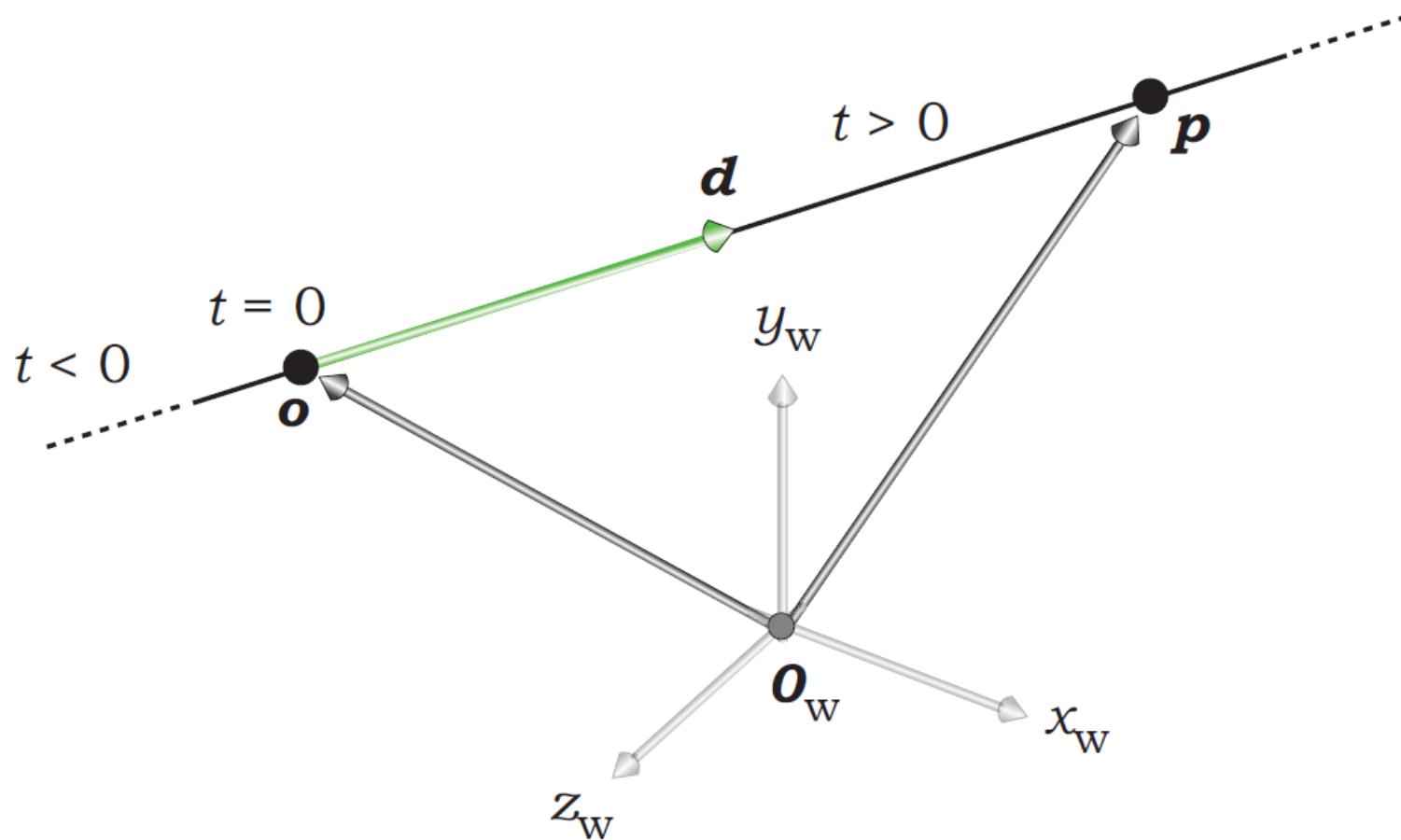
(Whitted 1980)

<http://www.youtube.com/watch?v=WV4qXzM641o>

Ray-object intersections

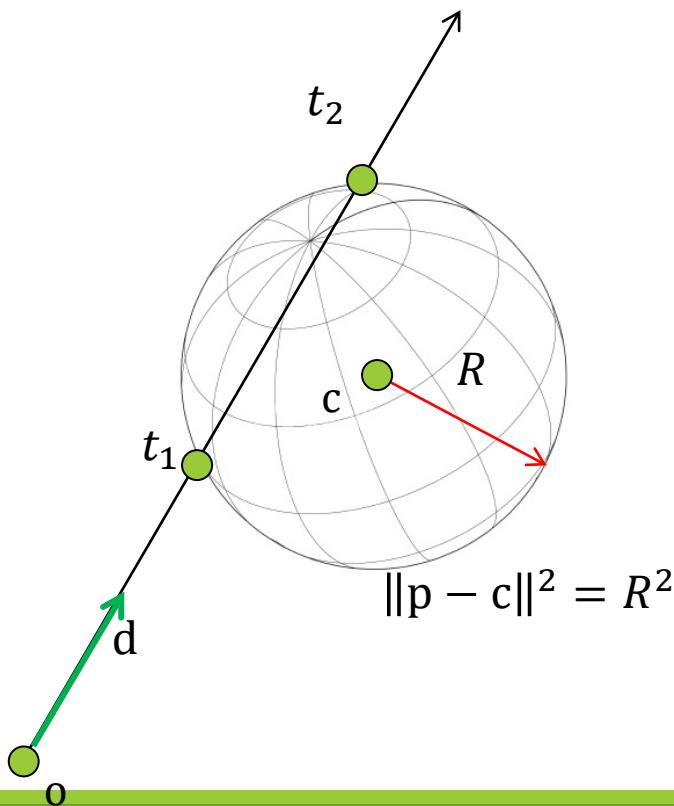


Ray = origin + direction



Ray-Sphere intersection

$$p(t) = o + td$$



$$\|p - c\|^2 = R^2$$

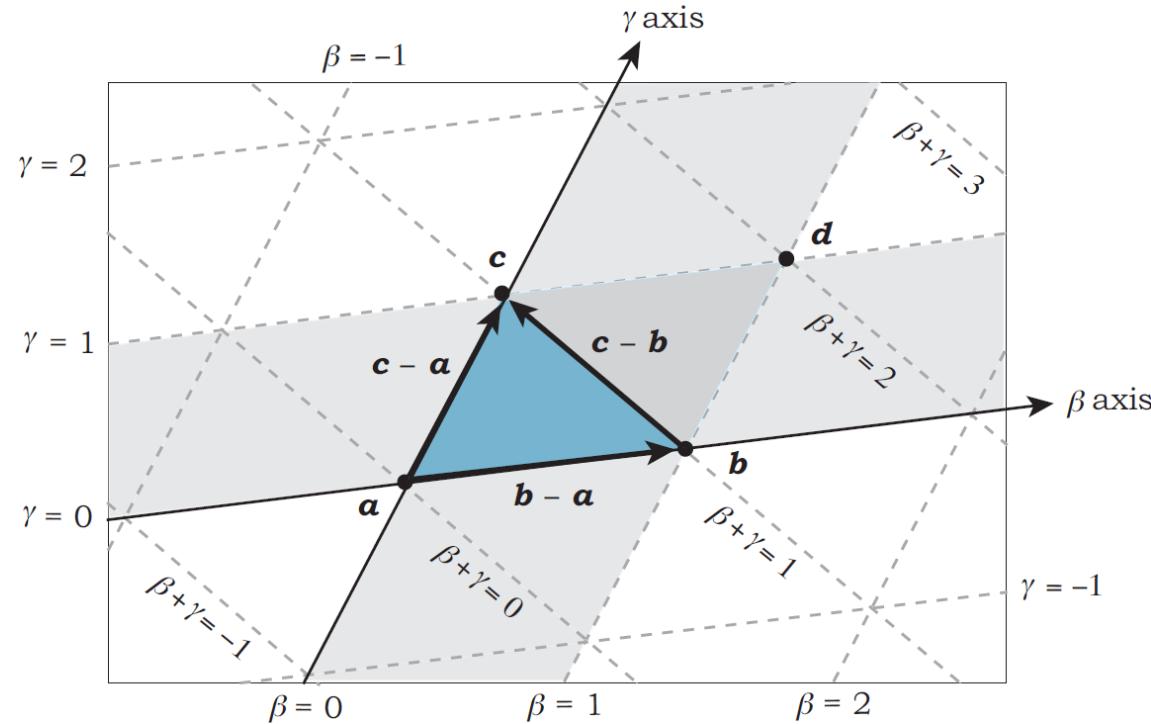
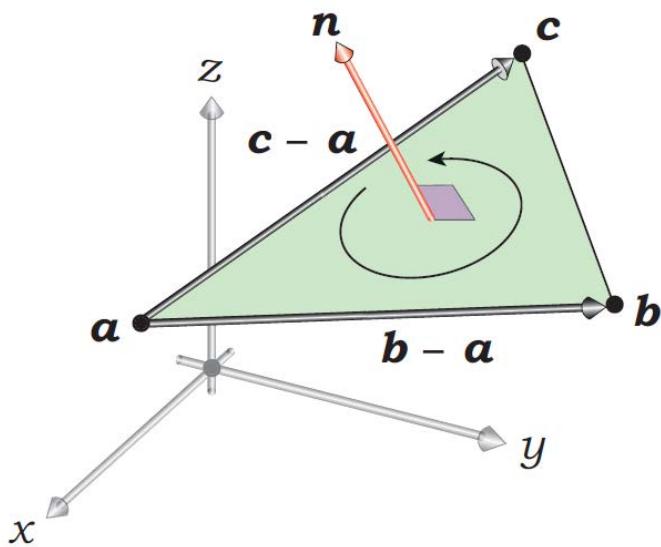
$$(p - c) \cdot (p - c) - R^2 = 0$$

$$(o + td - c) \cdot (o + td - c) - R^2 = 0$$

$$(d \cdot d)t^2 + 2d \cdot (o - c)t + (o - c) \cdot (o - c) - R^2 = 0$$

- 0, 1 or 2 solutions
- Keep smallest t -value > 0
- Ray might start inside sphere!

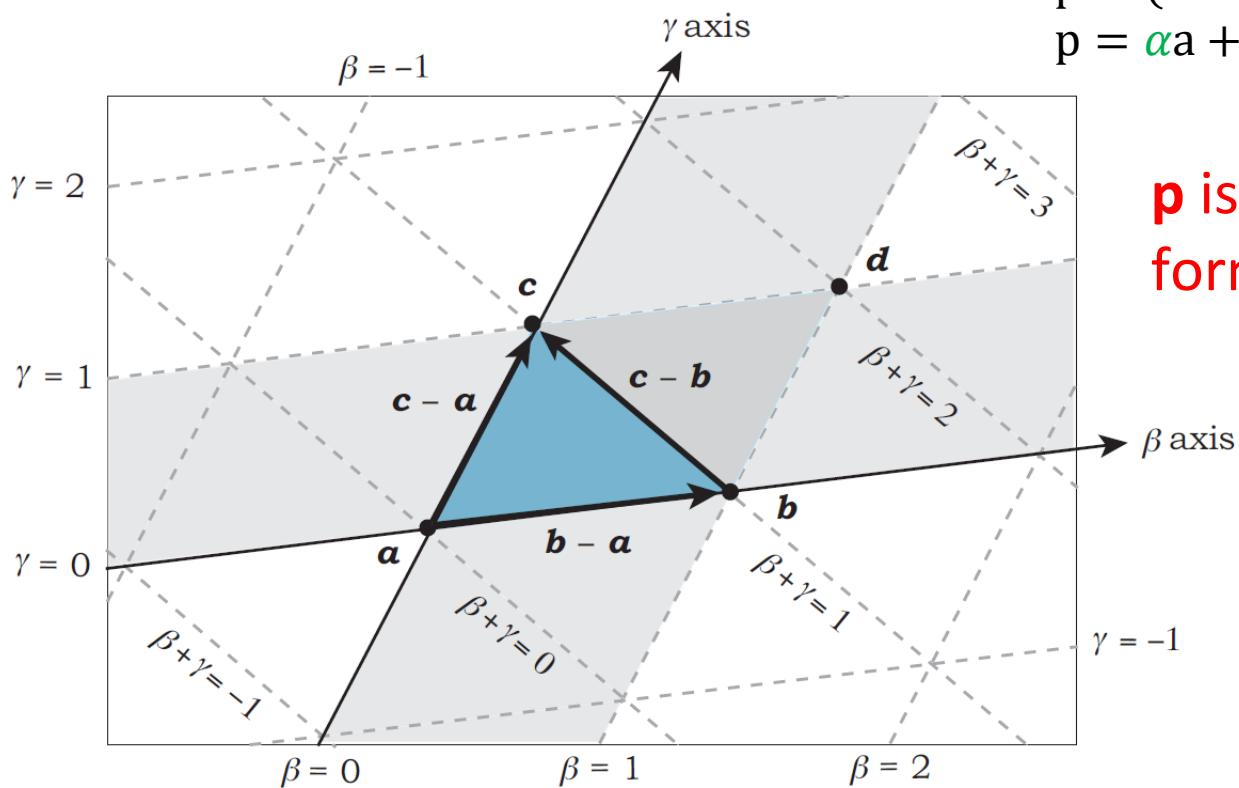
Ray-Triangle intersection



Given 3 points a, b, c

Each point p in the plane (a, b, c): $p = a + \beta(b - a) + \gamma(c - a)$

Ray-Triangle intersection

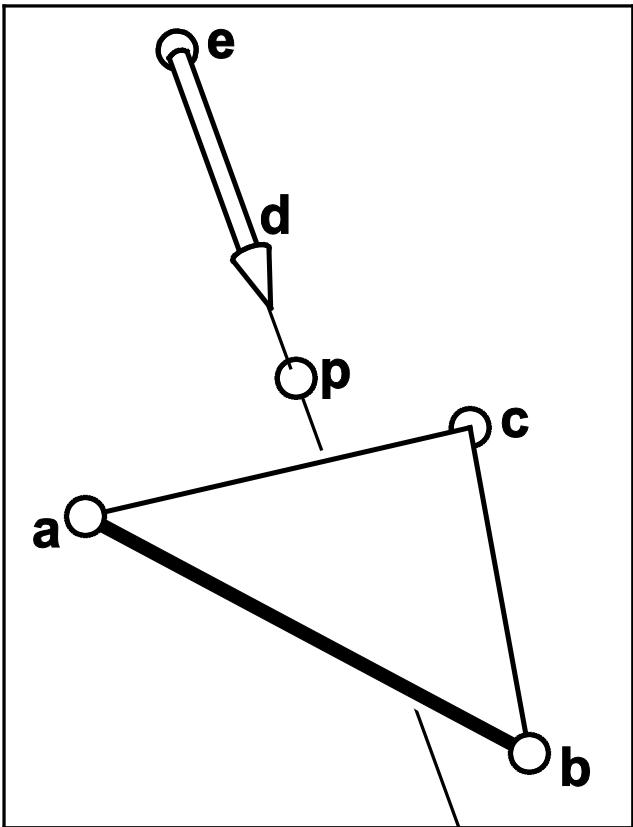


$$\begin{aligned} p &= a + \beta(b - a) + \gamma(c - a) \\ p &= (1 - \alpha)a + \beta b + \gamma c \\ p &= \alpha a + \beta b + \gamma c \quad (\alpha + \beta + \gamma = 1) \end{aligned}$$

p is inside the triangle formed by (a,b,c) if:

$$\begin{aligned} 0 < \alpha < 1 \\ 0 < \beta < 1 \\ 0 < \gamma < 1 \end{aligned}$$

Ray-Triangle intersection



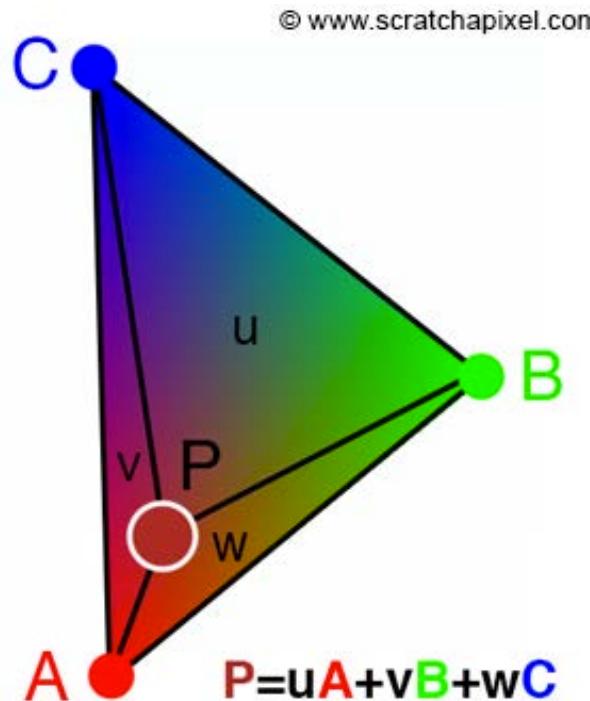
$$p(t) = e + td = a + \beta(b-a) + \gamma(c-a)$$

$$\begin{bmatrix} x_a - x_b & x_a - x_c & x_d \\ y_a - y_b & y_a - y_c & y_d \\ z_a - z_b & z_a - z_c & z_d \end{bmatrix} \begin{bmatrix} \beta \\ \gamma \\ t \end{bmatrix} = \begin{bmatrix} x_a - x_e \\ y_a - y_e \\ z_a - z_e \end{bmatrix}$$

Valid intersection if:
 $t > 0$ and point p inside triangle

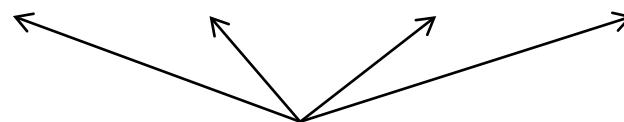
(Other) practical uses of barycentric coordinates

Attributes (e.g. color rgb) linked to triangle vertices can be interpolated



$$p = \alpha a + \beta b + \gamma c \quad (\alpha + \beta + \gamma = 1)$$

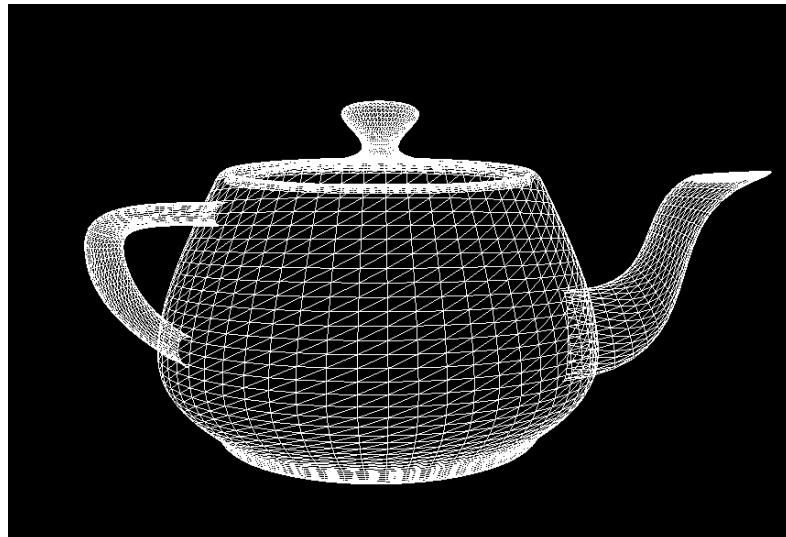
$$p_{attr} = \alpha a_{attr} + \beta b_{attr} + \gamma c_{attr}$$



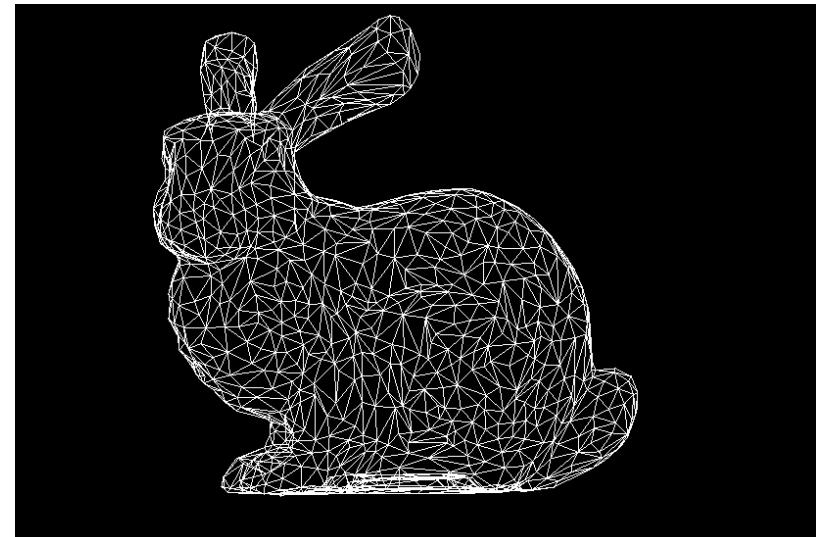
Could be a color, a normal vector, ...

Polygon Mesh

Polygon mesh = collection of connected polygons (usually triangles)



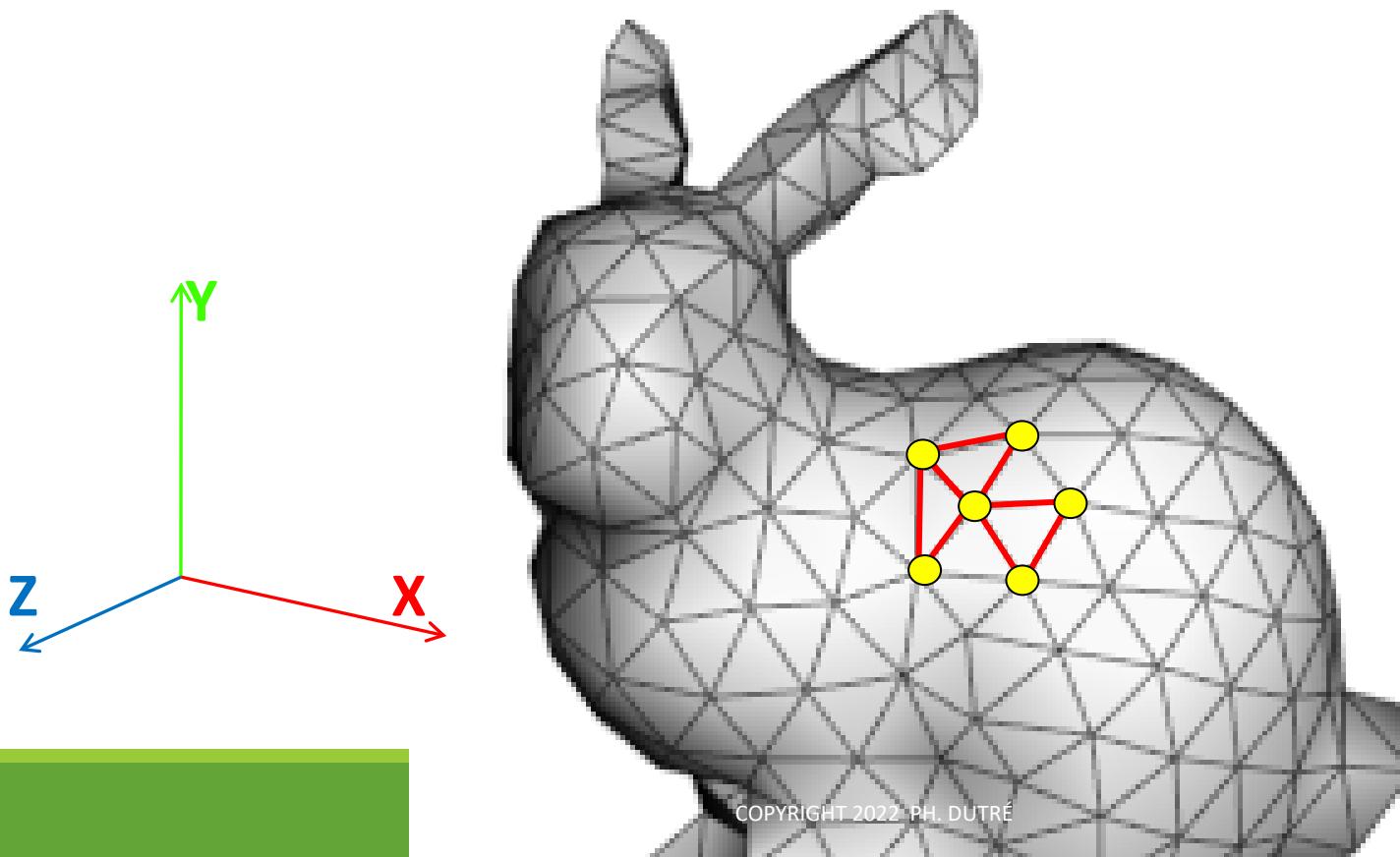
“Utah teapot”



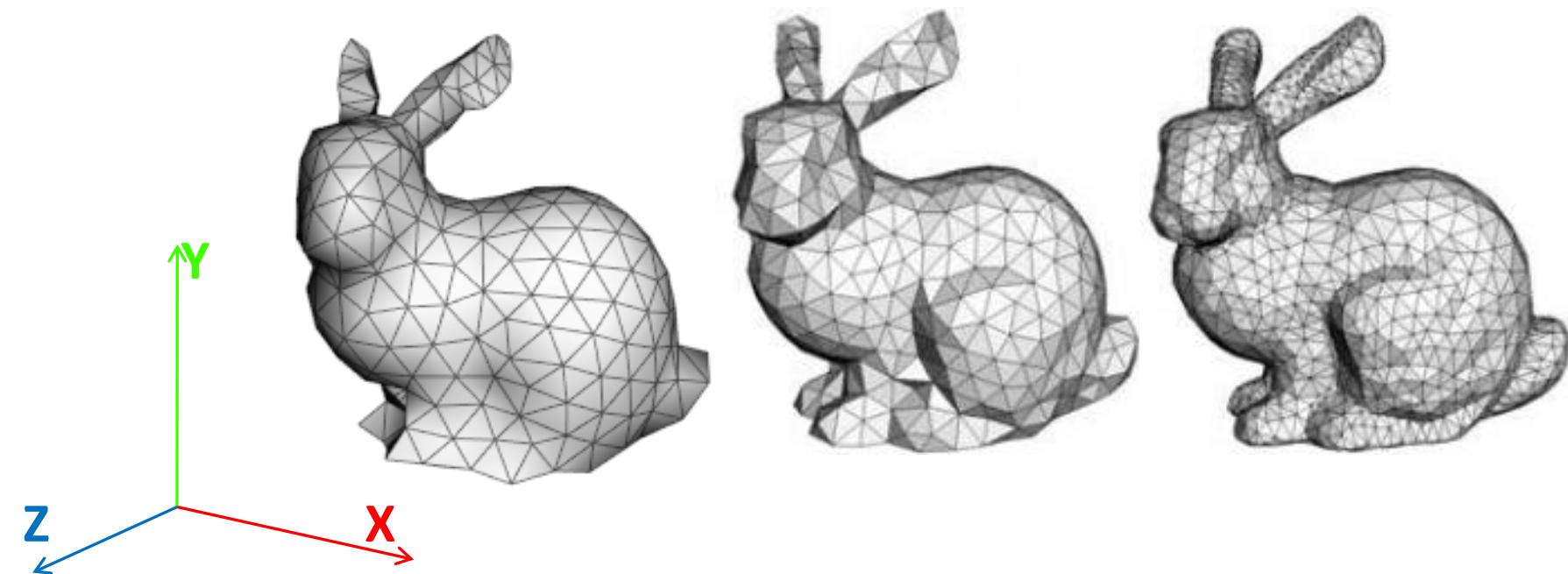
“Stanford Bunny”

Polygon Mesh

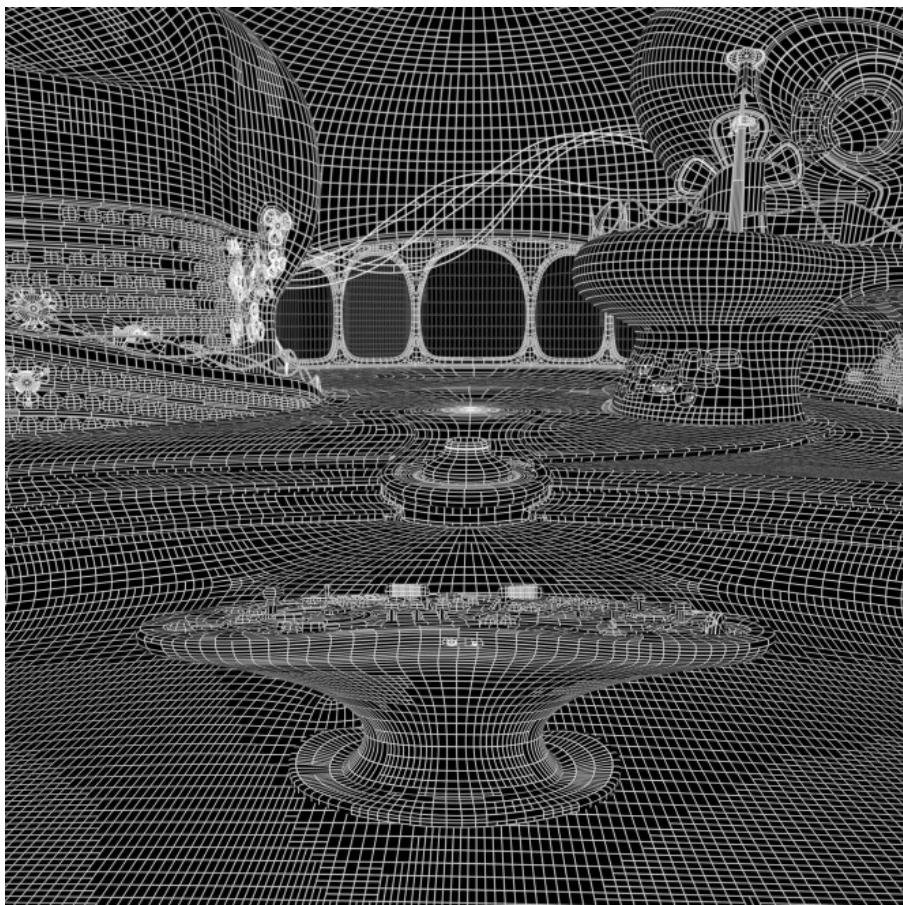
Polygons share edges and vertices, and form a larger 3D surface



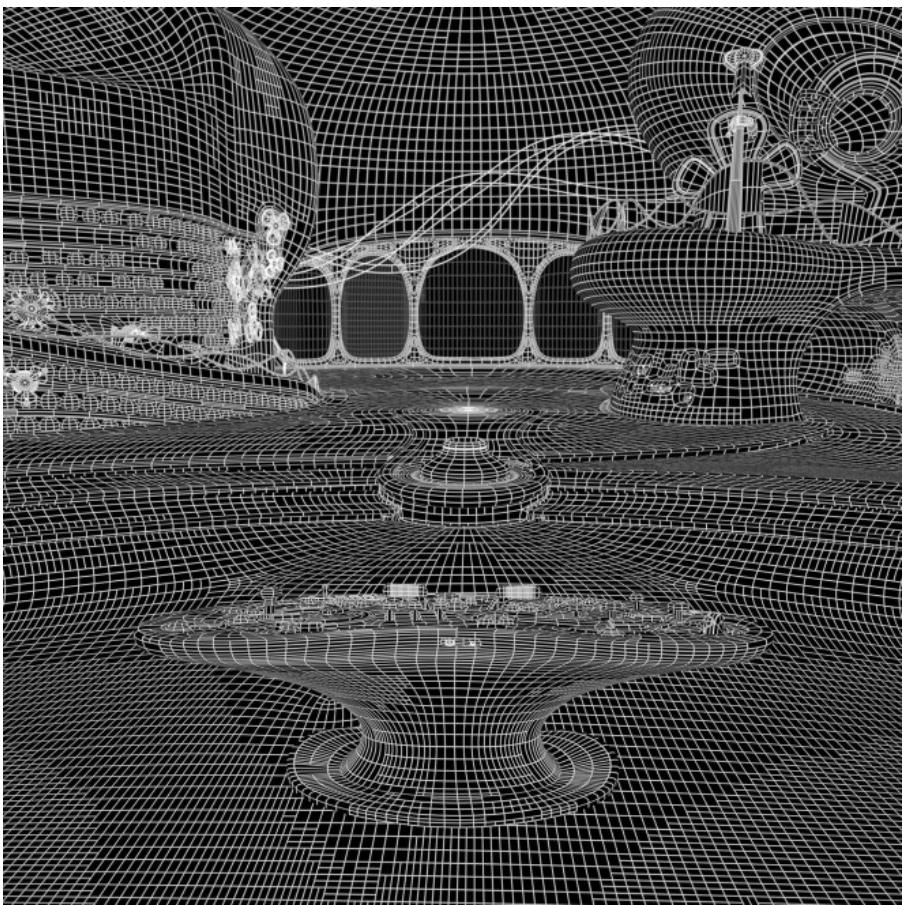
Polygon Mesh





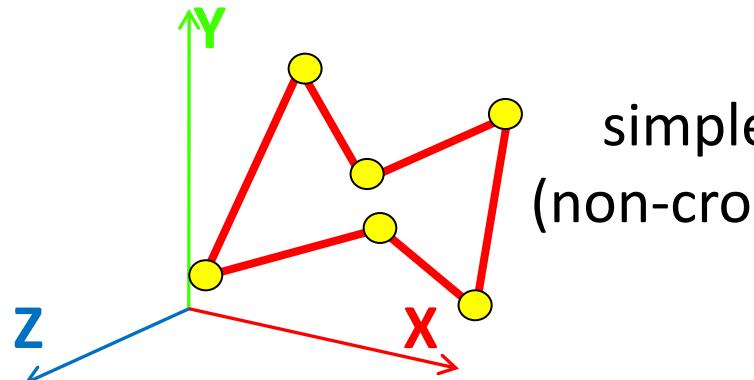


(Pixar, Inside Out, 2015)

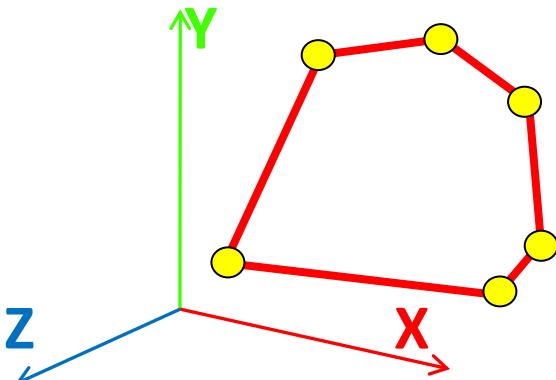


(Pixar, Inside Out, 2015)

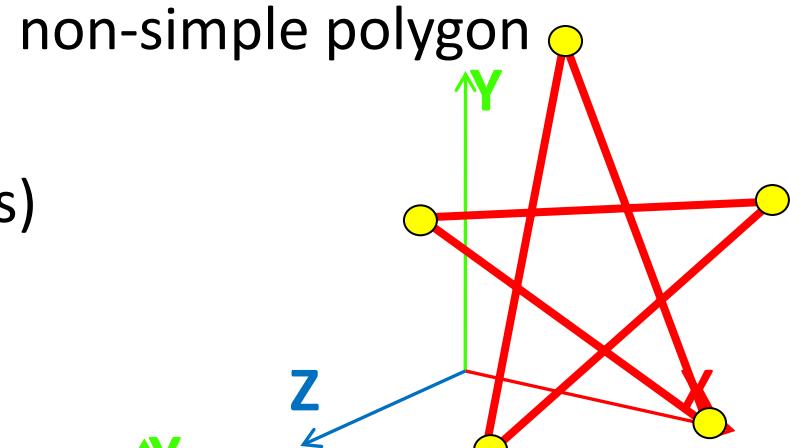
Polygon Mesh



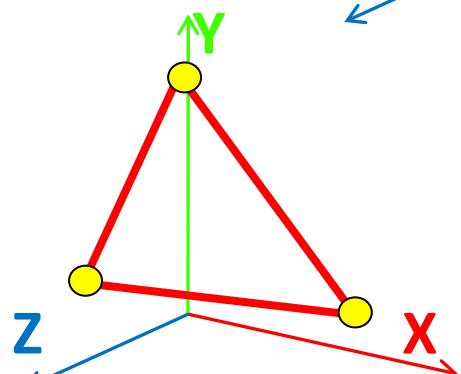
simple polygon
(non-crossing edges)



simple convex polygon
(internal angles $< \pi$)



non-simple polygon



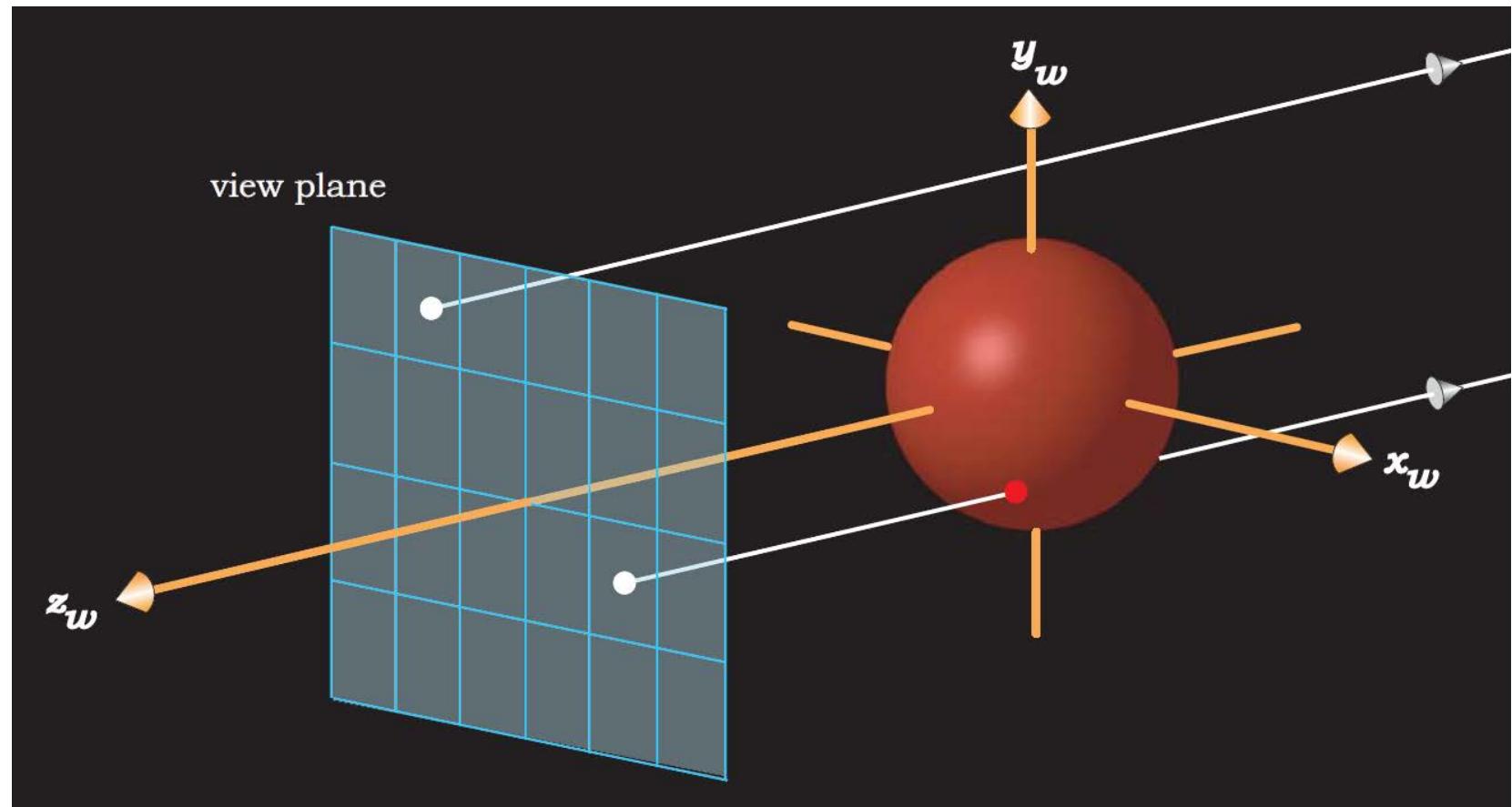
triangle

Polygon Meshes

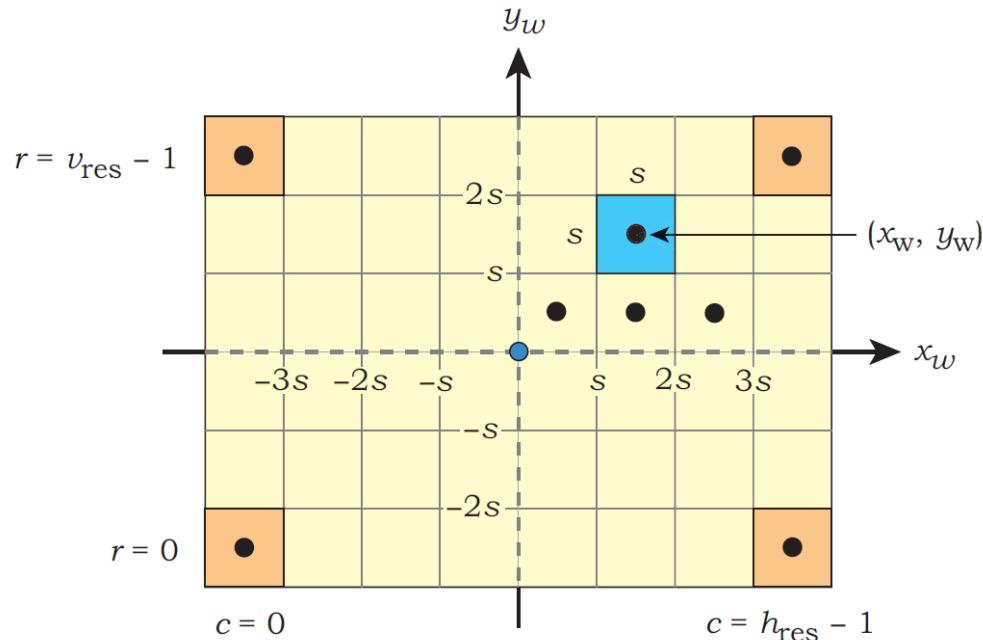
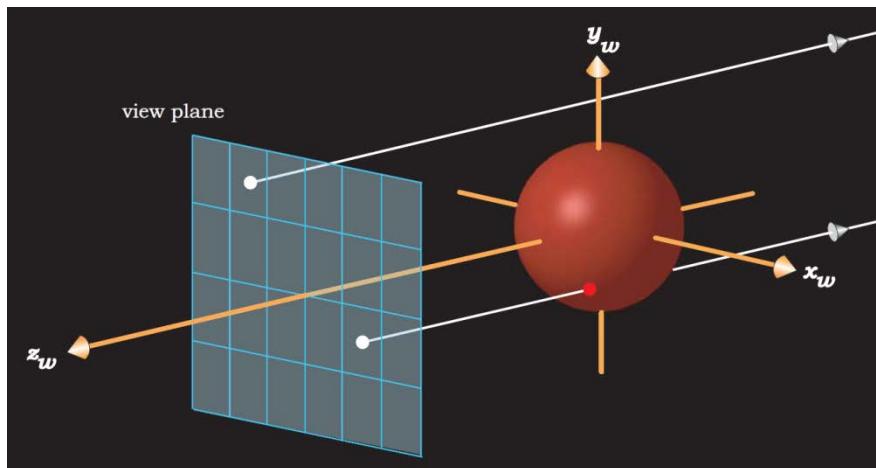
How to intersect a ray with a polygon mesh?

```
init nearest_intersection (distance = infinity)
for each polygon
    compute new_intersection t of ray with polygon
    if t < nearest and t > 0
        nearest_intersection = t
return nearest
```

Setting up the rays



Setting up the rays

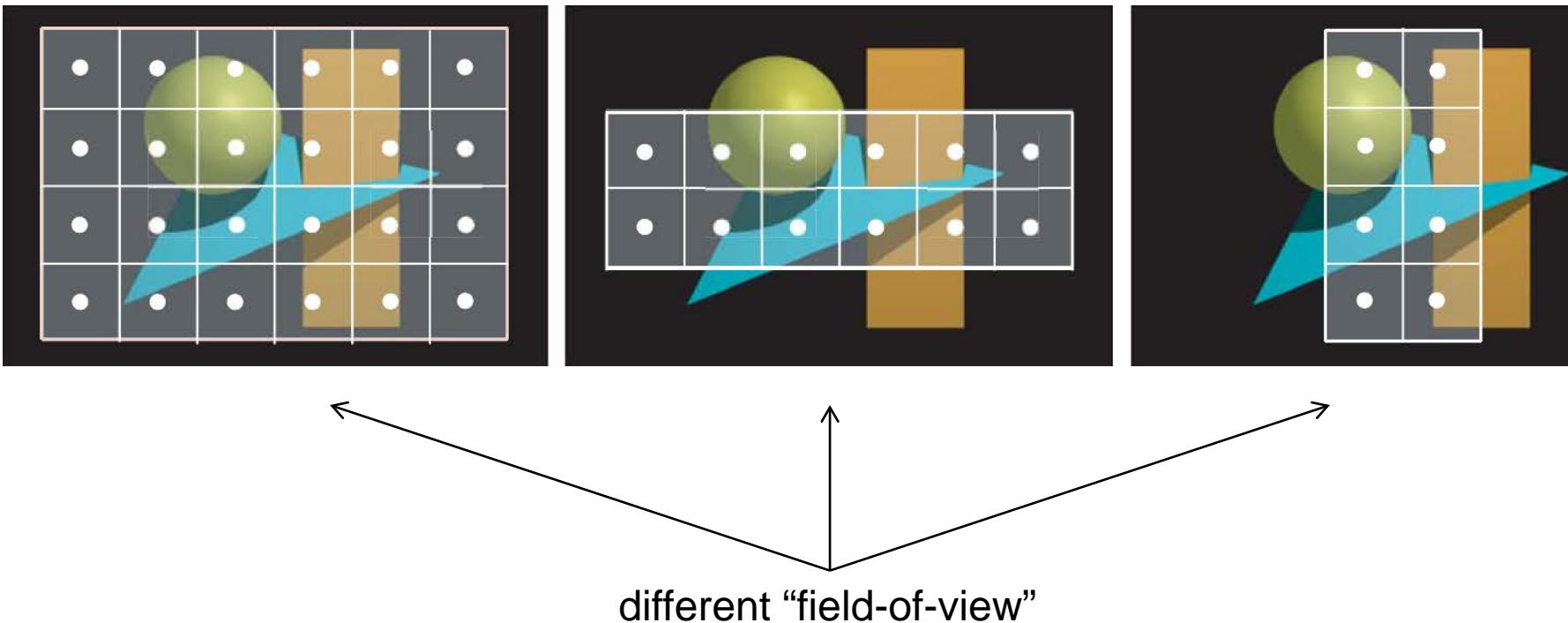


$$x_w = s(c - h_{\text{res}}/2 + 0.5)$$

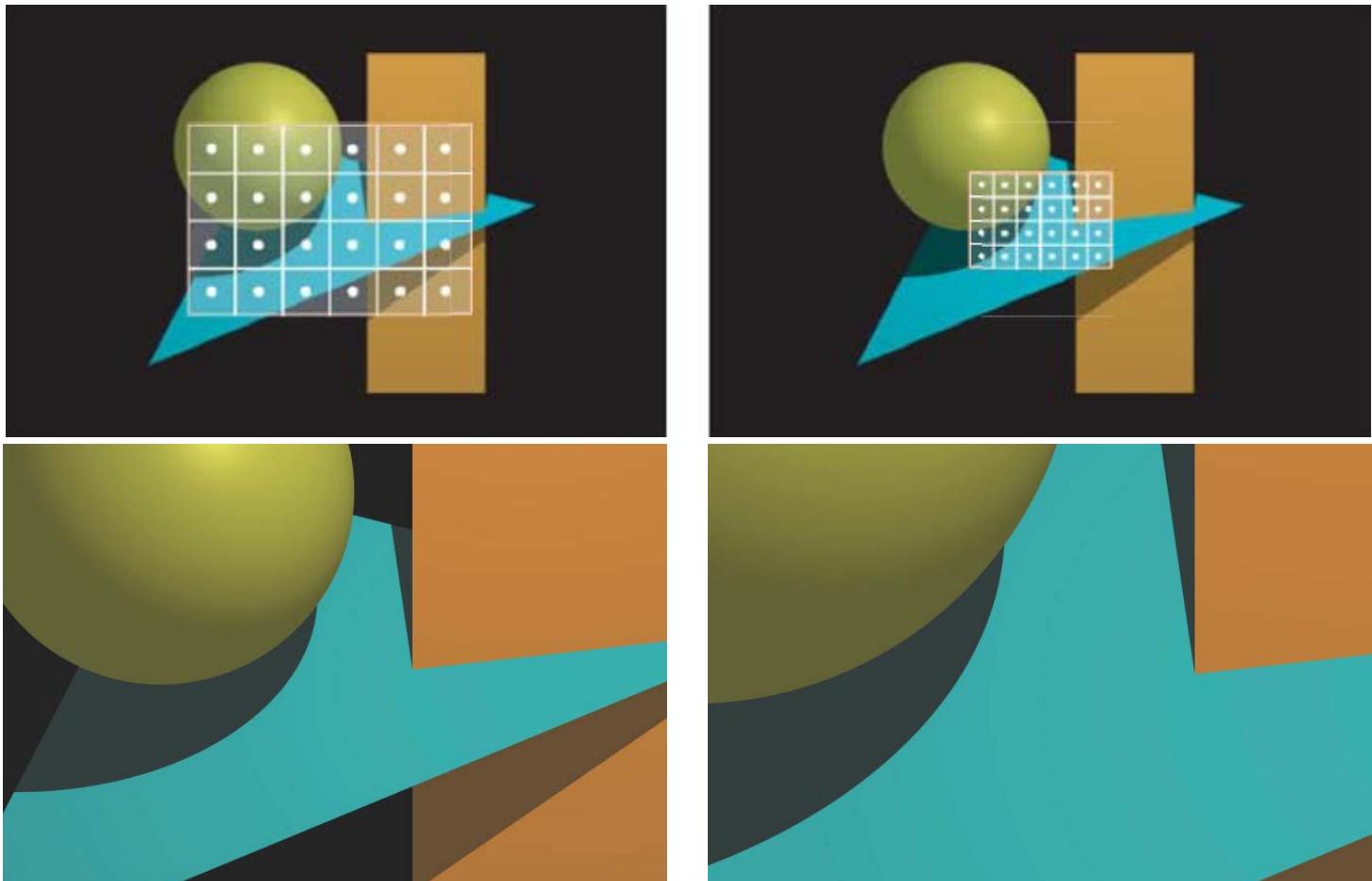
$$y_w = s \left(r - \frac{v_{\text{res}}}{2} + 0.5 \right)$$

$$d = (0, 0, -1)$$

Setting up the rays



Setting up the rays



Setting up the rays

