IV121 Vybrané aplikace informatiky v biologii

3D počítačová grafika

Katedra informačních technologií
Masarykova Univerzita Brno

Jaro 2012

Tento projekt je spolufinancován Evropským sociálním fondem a státním rozpočtem České republiky.

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

3D Počítačová Grafika (nebo Geometrie)

- modelování scén

SDL (scene description language)

- vizualizace scén (rendering)

- rasterizace
- „raytracing“

- zajímavé koncepty

CSG (constructive solid geometry)

skriptování scén

příklad generování realistických stromů a keřů

SDL – Scene Description Languages

VRML/X3D

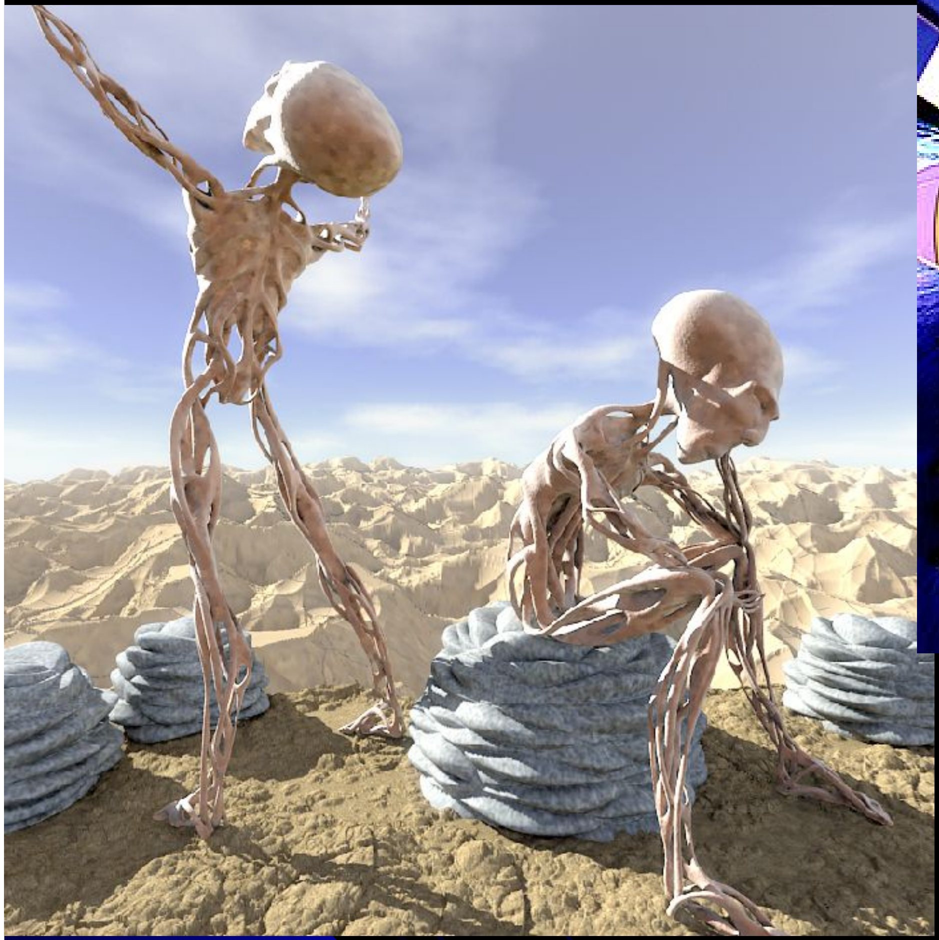
3DMLW

POV-Ray SDL

Renderman shading language

http://en.wikipedia.org/wiki/Scene_description_language

Co je RenderMan?



Entropy image contest winner by Claude Schitter

MonstersInc by Pixar

A Bug‘s Life by Pixar

Co je RenderMan?

Autorem je společnost Pixar (1987)

Něco jako PostScript pro 3D

- Scene Description Language

  není modelovacím programem

  není renderingovým programem

  Rozhraním mezi modelováním a renderingem

Příklad Bytestream kódu pro RenderMan Interface

```
Display "RenderMan" "framebuffer"
  "rgb"
Format 256 192 1
WorldBegin
Surface "constant"
Polygon "P" [0.5 0.5 0.5 0.5 -0.5
  0.5 -0.5 -0.5 0.5 -0.5 0.5 0.5]
WorldEnd
```

API

```
#include <ri.h>
RtPoint Square[4] = { {.5,.5,.5}, {.5,-.5,.5}, {-.5,-.5,.5},
   {-.5,.5,.5} };

main(void){
    RiBegin(RI_NULL);   /* Start the renderer */
        RiDisplay("RenderMan", RI_FRAMEBUFFER, "rgb", RI_NULL);
        RiFormat((RtInt) 256, (RtInt) 192, 1.0);
        RiWorldBegin();
            RiSurface("constant", RI_NULL);
            RiPolygon( (RtInt) 4,      /* Declare the square */
                RI_P, (RtPointer) Square, RI_NULL);
        RiWorldEnd();
    RiEnd();                /* Clean up */
```

RenderMan Shading Language

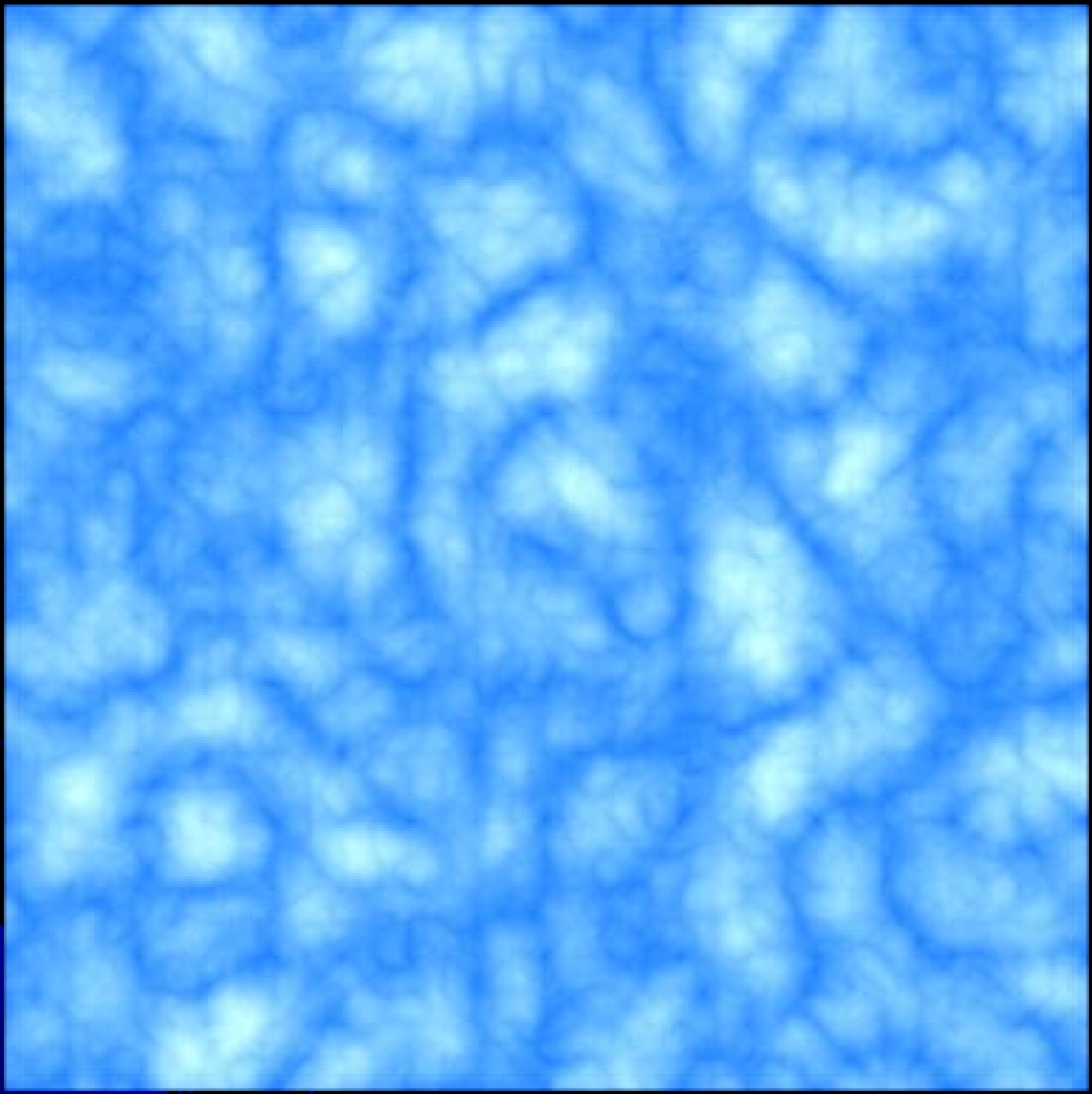
```
surface clouds(float vfreq = .8 )
{
    float sum ;
    float i;
    color white = color(1.0, 1.0, 1.0);
    point Psh = transform("shader", P);

    sum = 0;
    freq = vfreq;
    for (i = 0; i < 6; i = i + 1) {
        sum = sum + 1/freq * abs(.5 - noise(freq * Psh));
        freq = 2 * freq;
    }
    Ci = mix(Cs, white, sum*4.0);
    Oi = 1.0;              /* Always make the surface opaque */
}
```

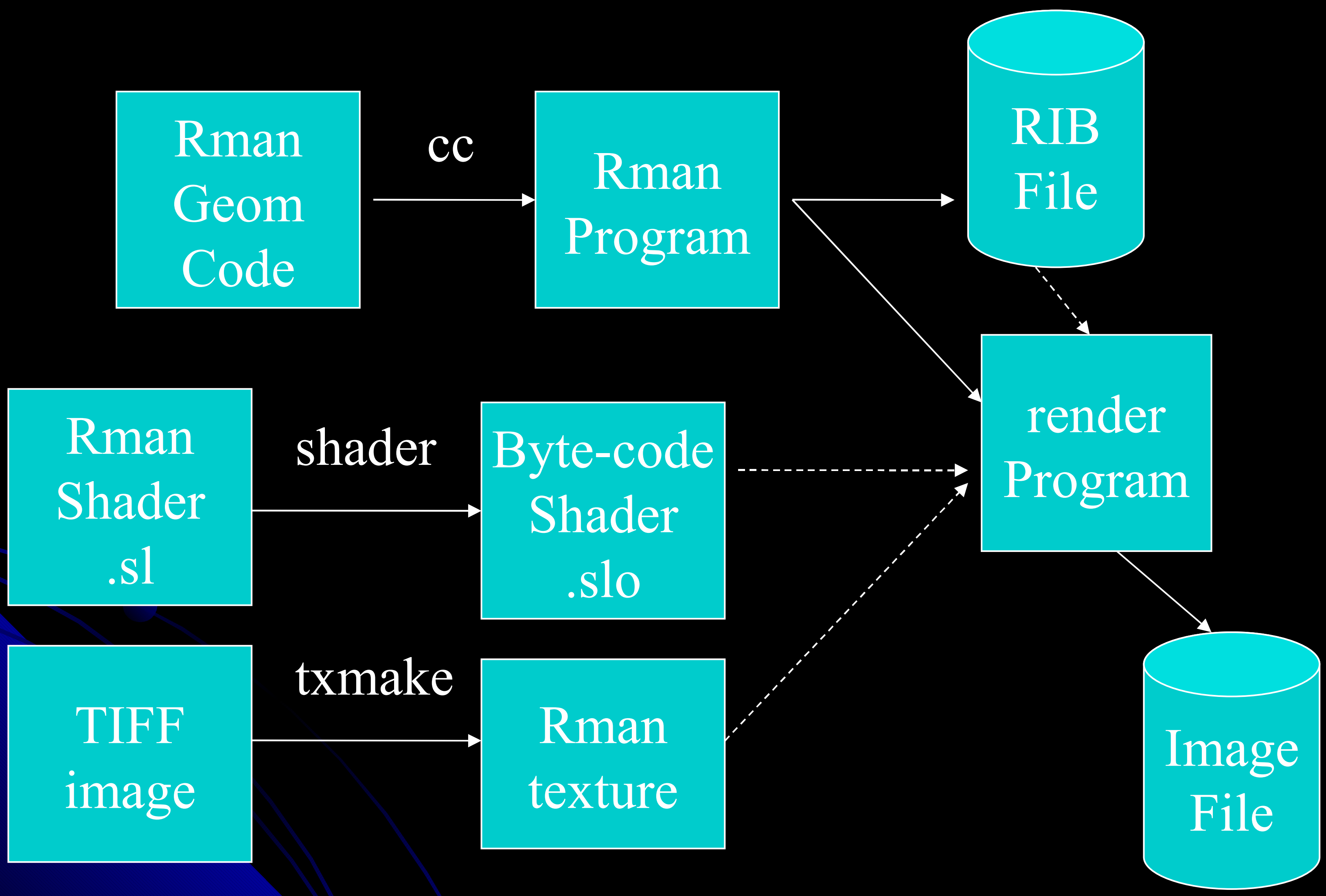
RIB s použitím “Shader” kódu

```
Display "RenderMan" "framebuffer"
  "rgb"
Format 256 192 1
WorldBegin
Surface "clouds"
Polygon "P" [0.5 0.5 0.5 0.5 -0.5
  0.5 -0.5 -0.5 0.5 -0.5 0.5 0.5]
WorldEnd
```

Result



Součásti systému RenderMan



„Rendering“

- rasterizace

Vlastnosti všech bodů v prostoru/modelu jsou lokálně definovány

- “raytracing“

Vlastnosti bodů jsou ovlivněny globálně všemi ostatními body ve scéně/modelu.

Dokáže správně vykreslit tansparetnost, refrakci světla a podobné efekty.

Rendering Pipeline - OpenGL

Aaron Bloomfield

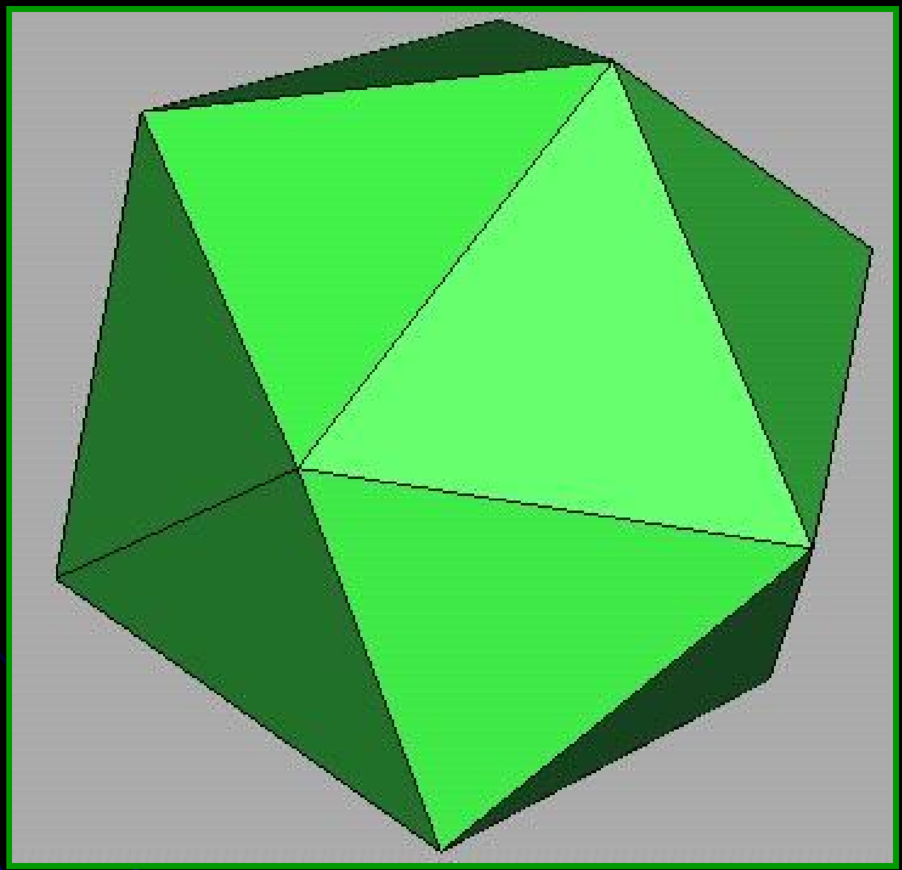
CS 445: Introduction to Graphics

Fall 2006

(Slide set originally by Greg Humphreys)

3D Polygon Rendering

Many applications use rendering of 3D polygons with direct illumination



3D Polygon Rendering

Many applications use rendering of 3D polygons with direct illumination

3D Rendering Pipeline

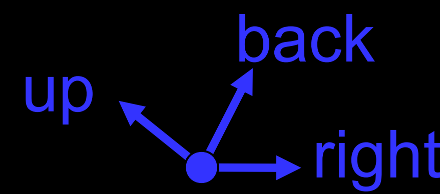
This is a pipelined sequence of operations to draw a 3D primitive into a 2D image

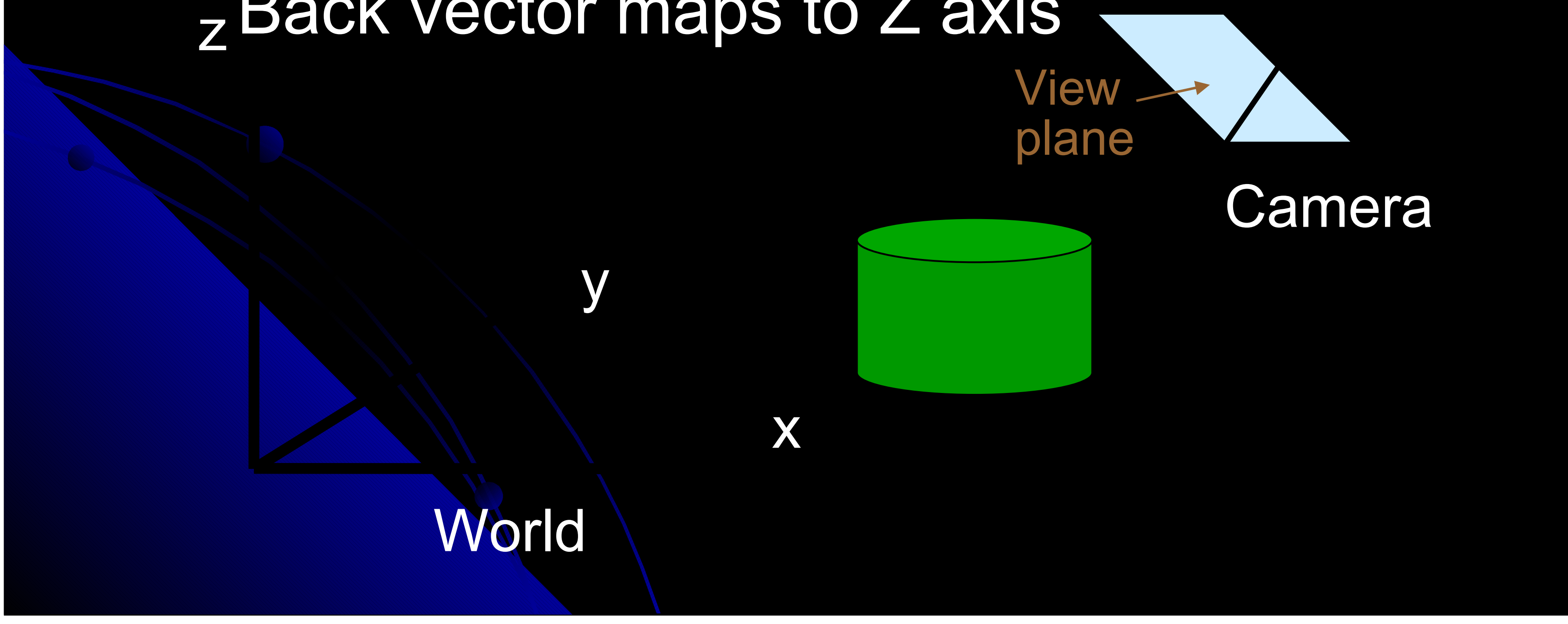
(this pipeline applies only for direct illumination)

Viewing Transformation

Mapping from world to camera coordinates

- Eye position maps to origin
- Right vector maps to X axis
- Up vector maps to Y axis
- Back vector maps to Z axis





Projection

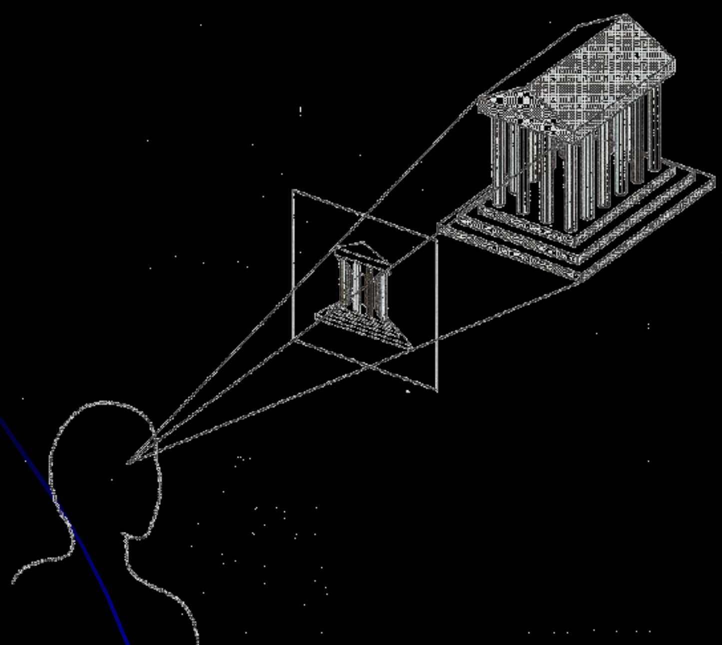
General definition:

- Transform points in *n*-space to *m*-space ($m<n$)

In computer graphics:

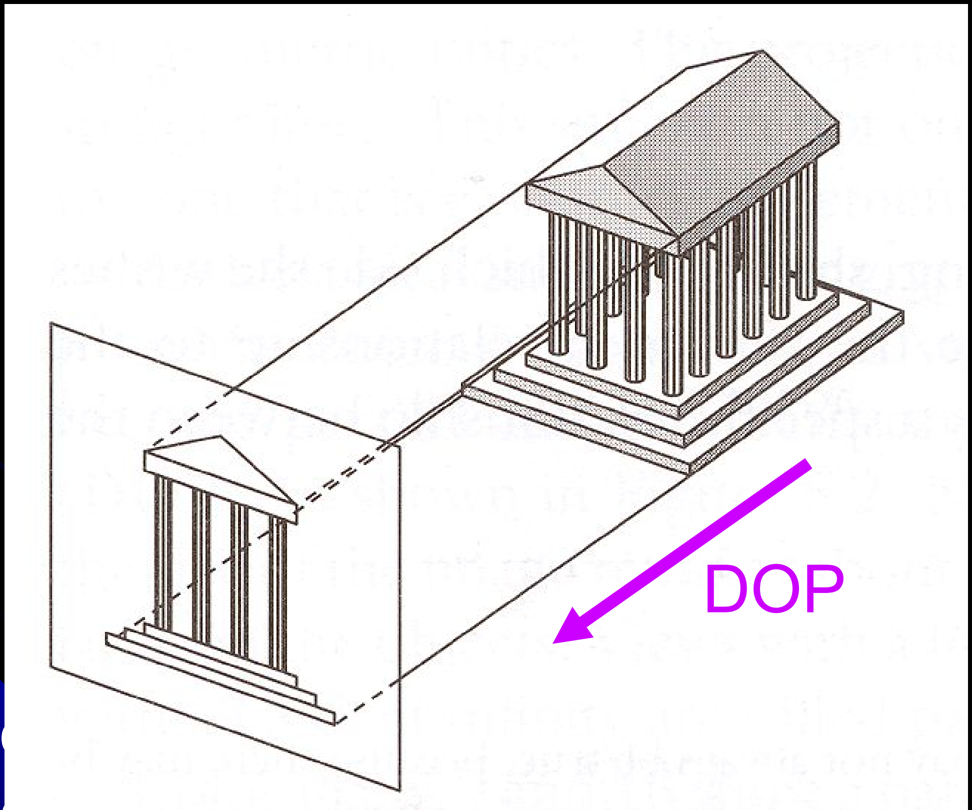
- Map 3D camera coordinates to 2D screen coordinates

For perspective transformations, no two “rays” are parallel to each other



Parallel Projection

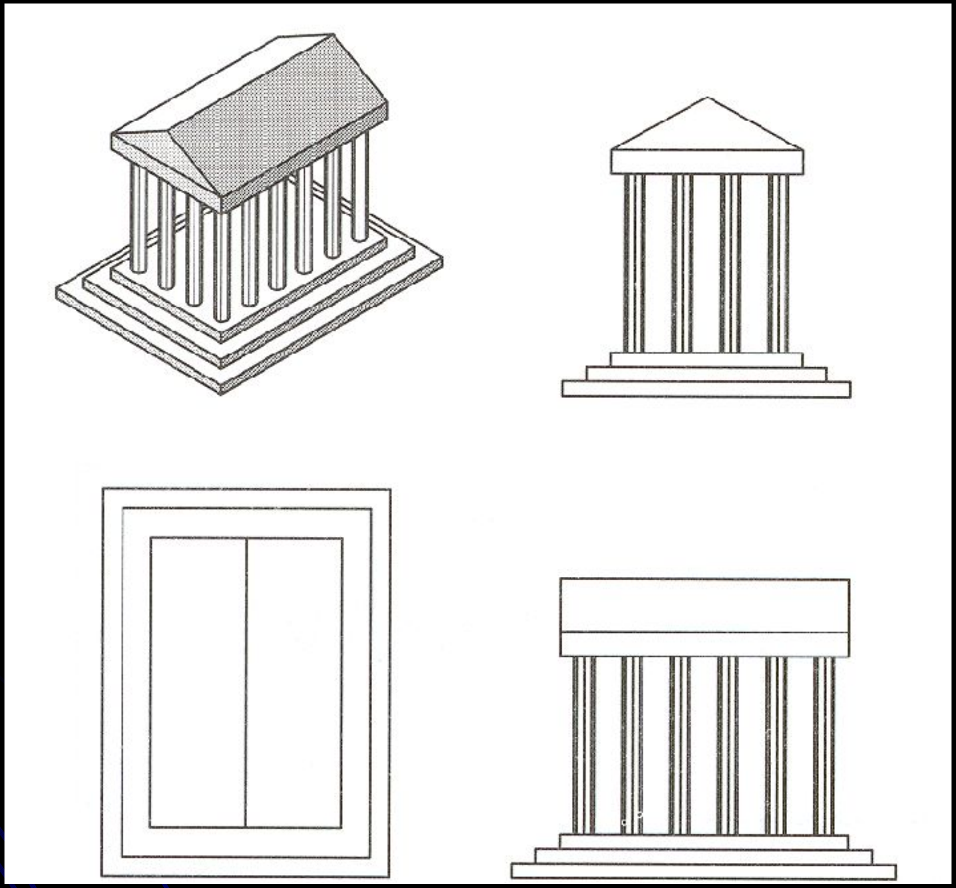
- Center of projection is at infinity
  - Direction of projection (DOP) same for all points



Angel Figure 5.4

Orthographic Projections

DOP perpendicular to view plane



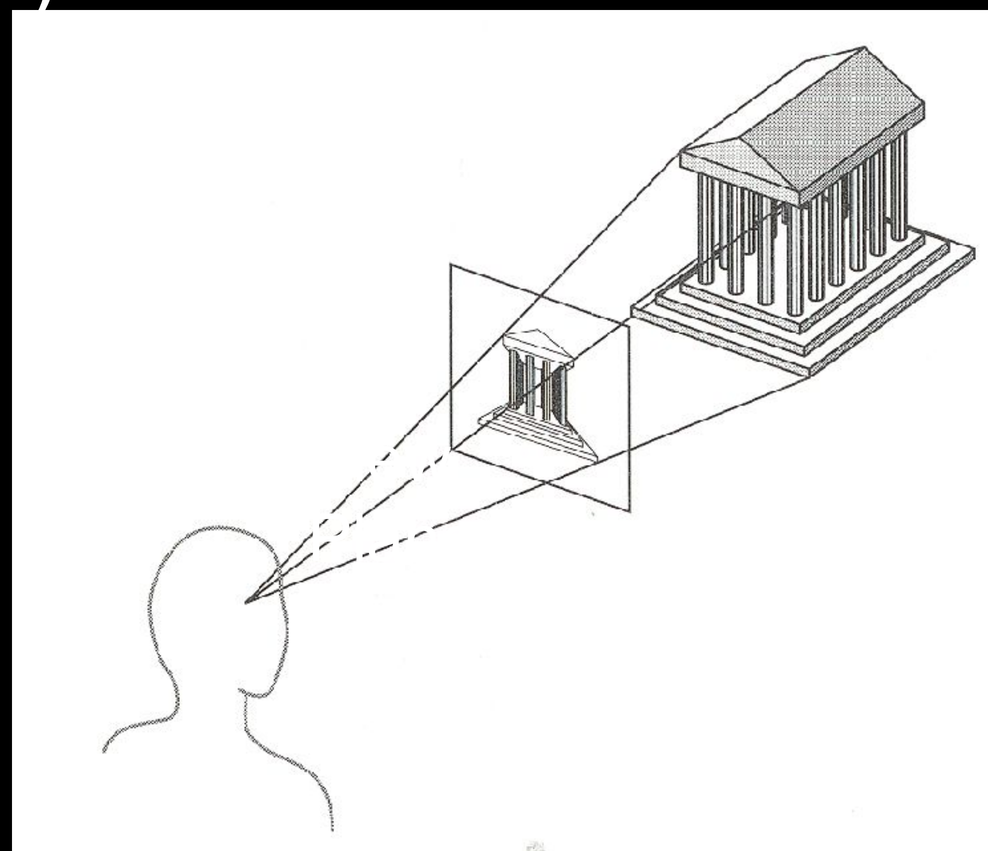
Top Side

Angel Figure 5.5

Perspective Projection

Map points onto "view plane" along "projectors" emanating from "center of projection" (COP)

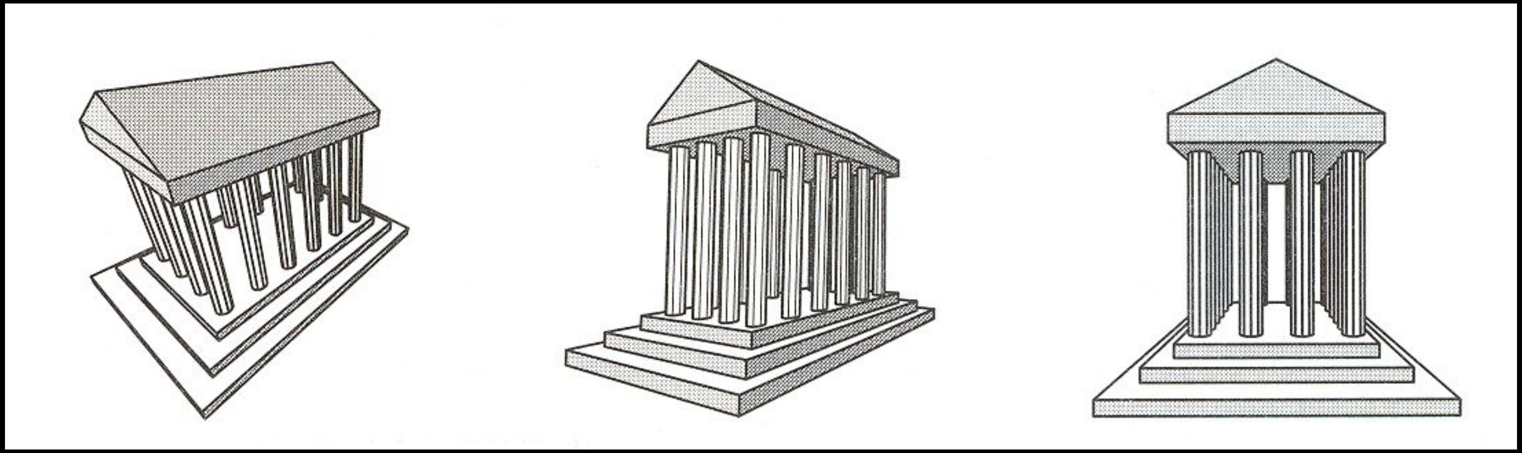
Center of Projection



re 5.9

Perspective Projection

How many vanishing points?

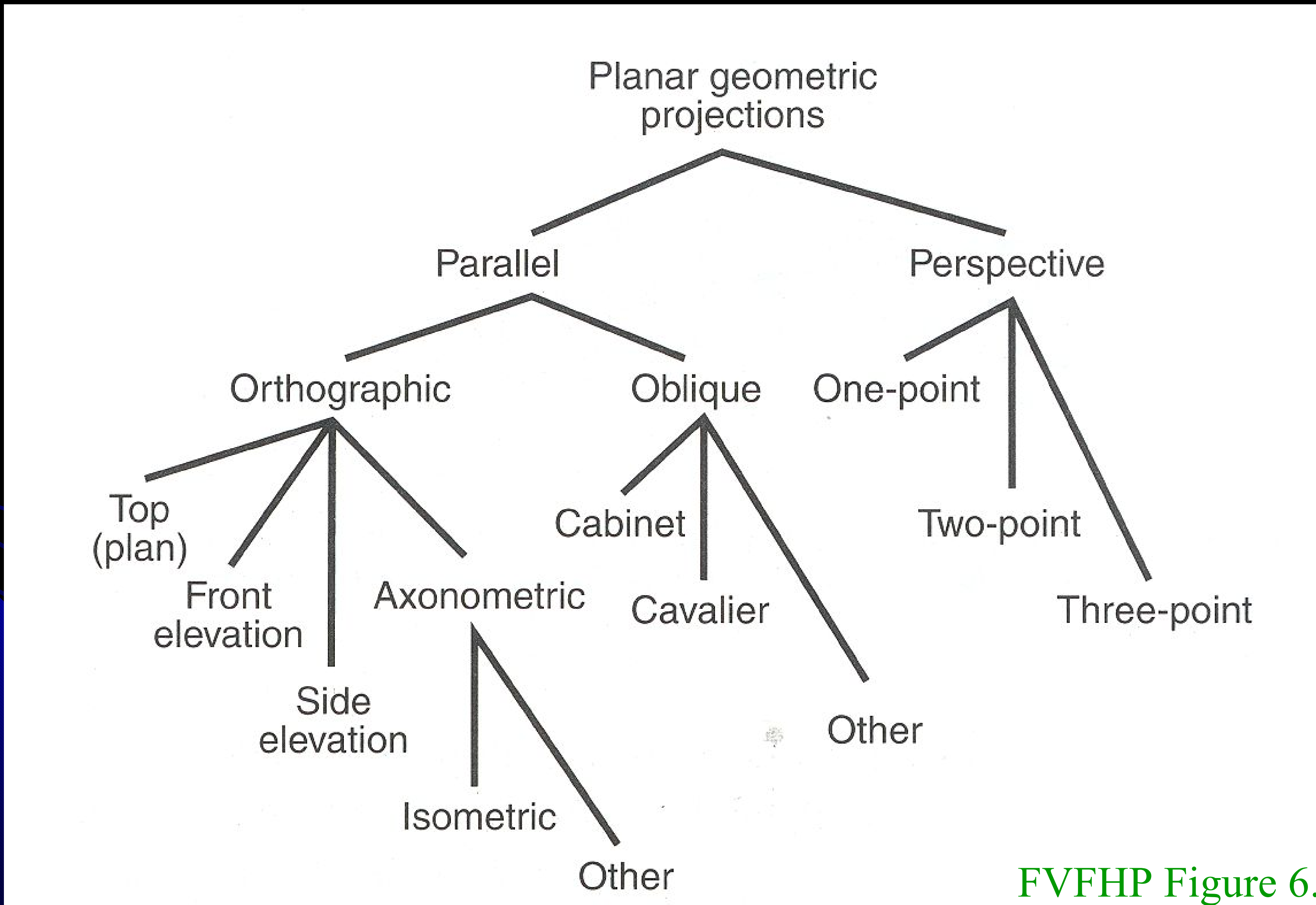


3-Point Perspective | 2-Point Perspective | 1-Point Perspective

- The difference is how many of the three principle directions are parallel/orthogonal to the projection plane

Angel Figure 5.10

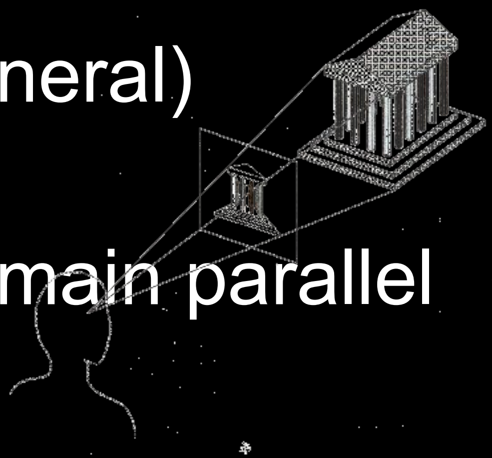
Taxonomy of Projections



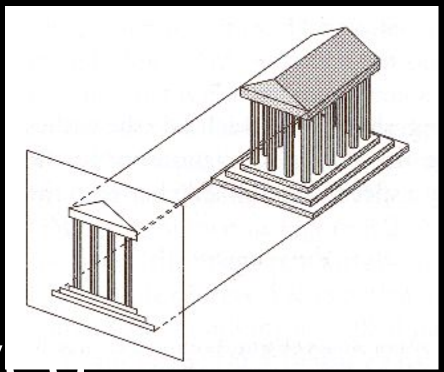
FVFHP Figure 6.10

Perspective vs. Parallel

- Perspective projection
  - + Size varies inversely with distance - looks realistic
  - – Distance and angles are not (in general) preserved
  - – Parallel lines do not (in general) remain parallel

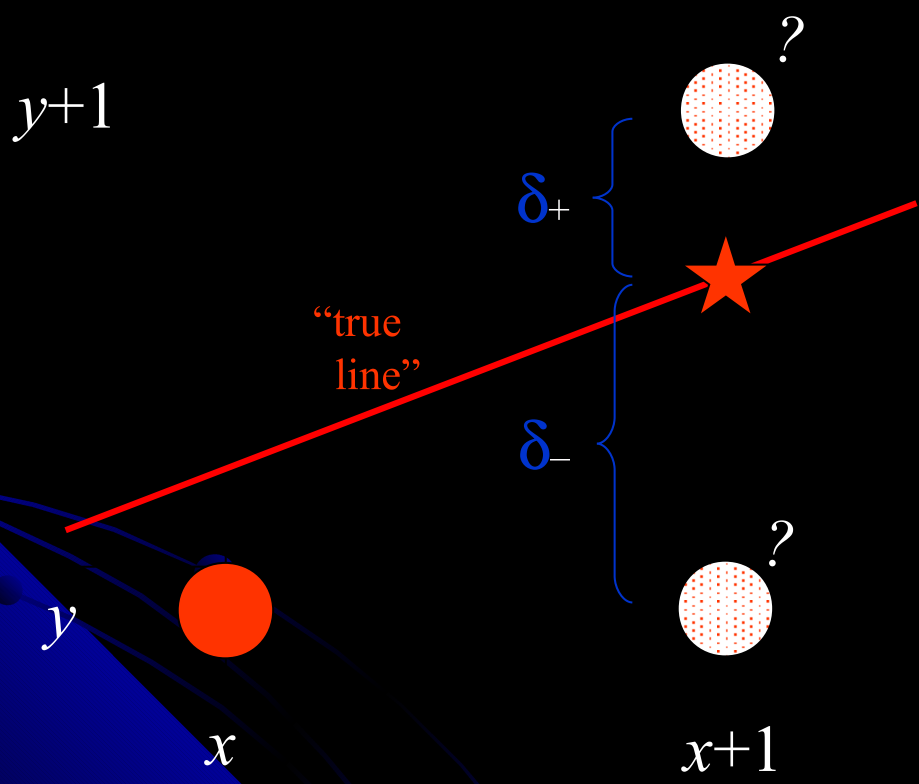


- Parallel projection
  - + Good for exact measurements
  - + Parallel lines remain parallel
  - – Angles are not (in general) preserved



Scan-converting Lines - Toward the Bresenham Algorithm

Special case: $0 < m < 1$, $\Delta x > 0$



imagine pixel centers at grid vertices

At each step:

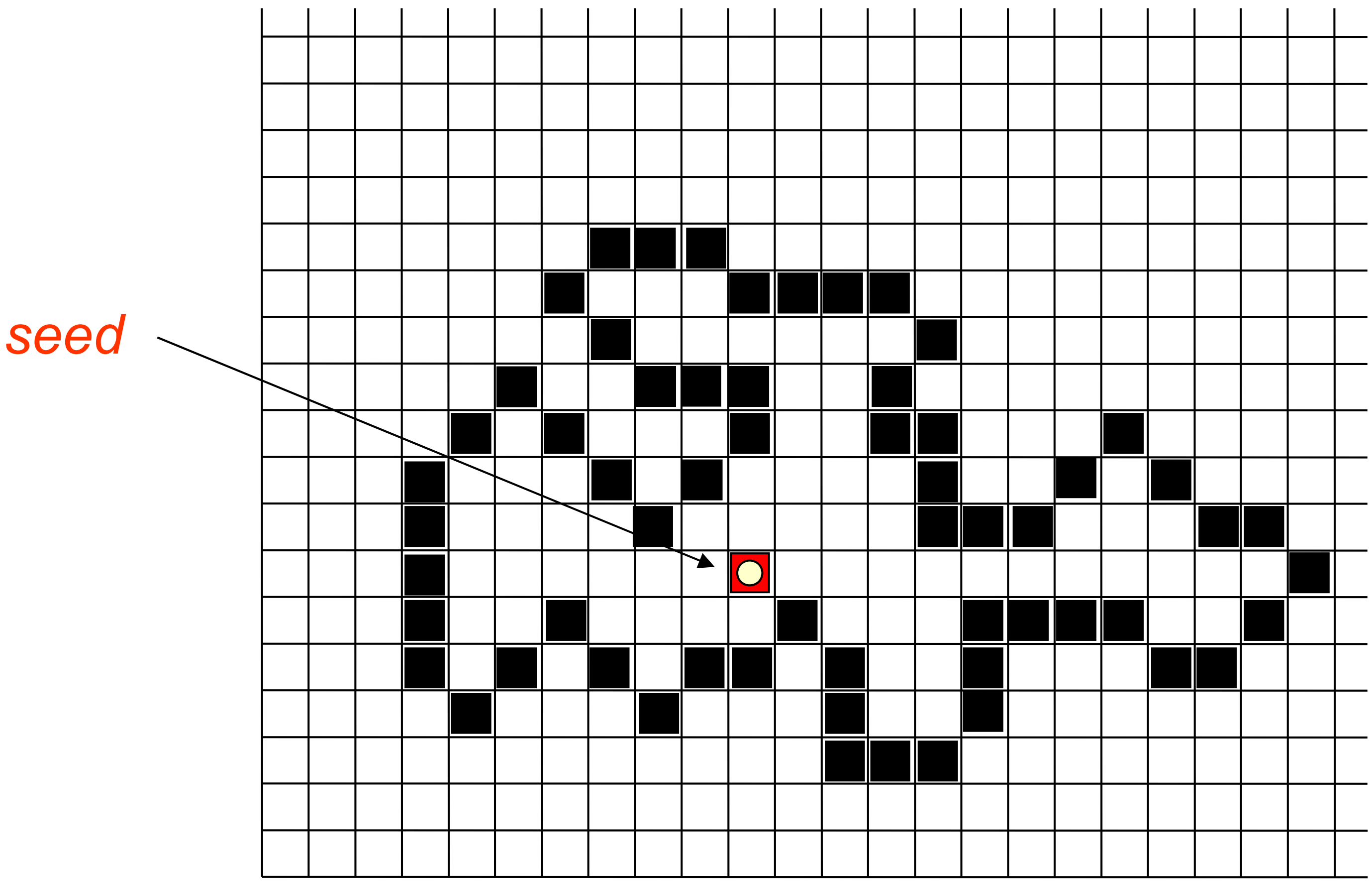
```
x++ ;
if ( δ- > δ+)
   y++ ;
```

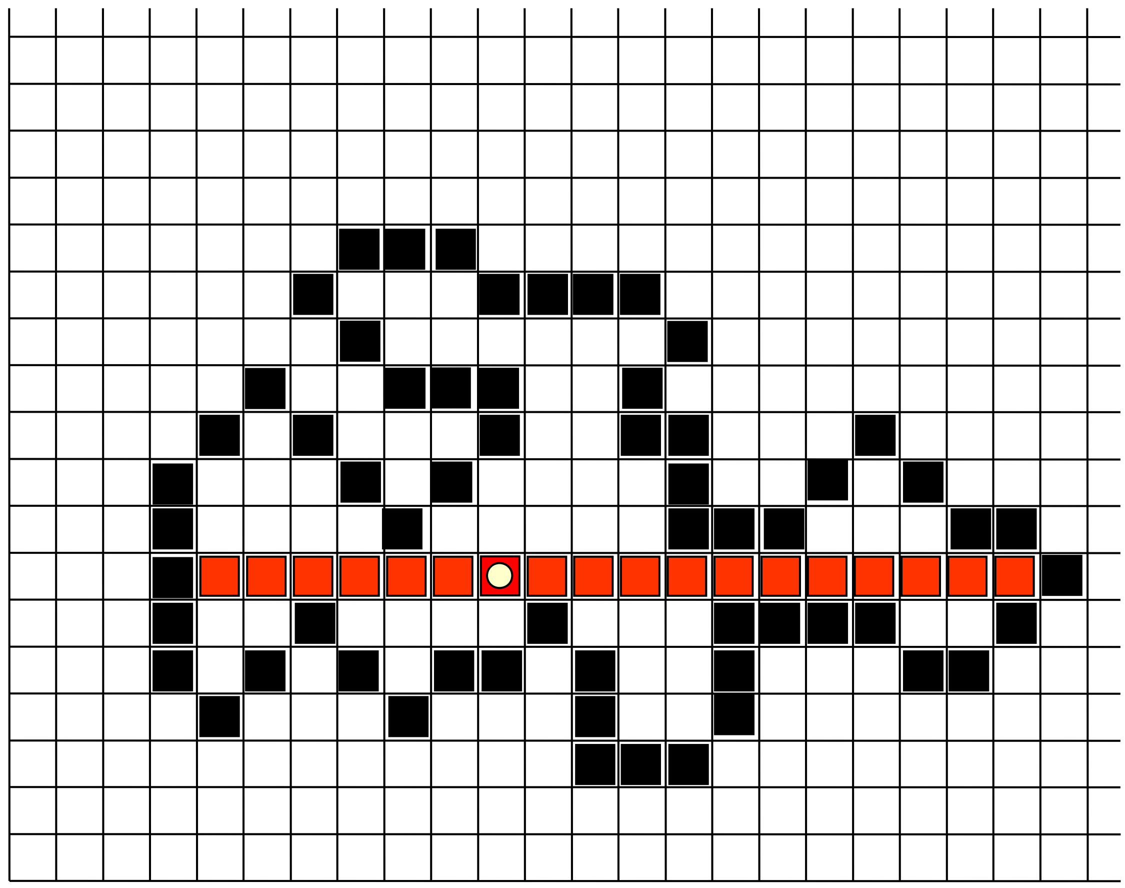
Let $p = \Delta x\,(\,\delta^- - \delta^+\,)$. Then:

At each step:

```
x++ ;
if ( p > 0 )
   y++ ;
update p ;
```

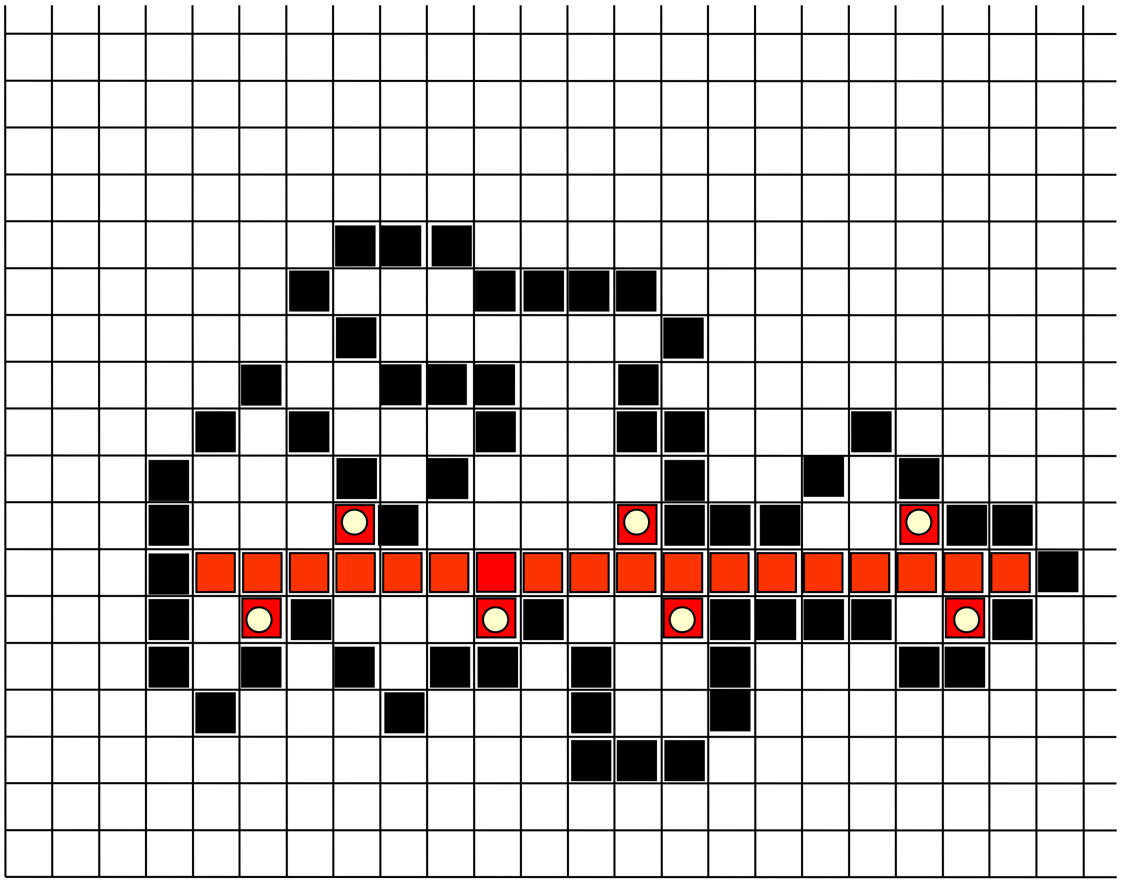
Can we do this with simple all-*int *arithmetic? Yes!


seed

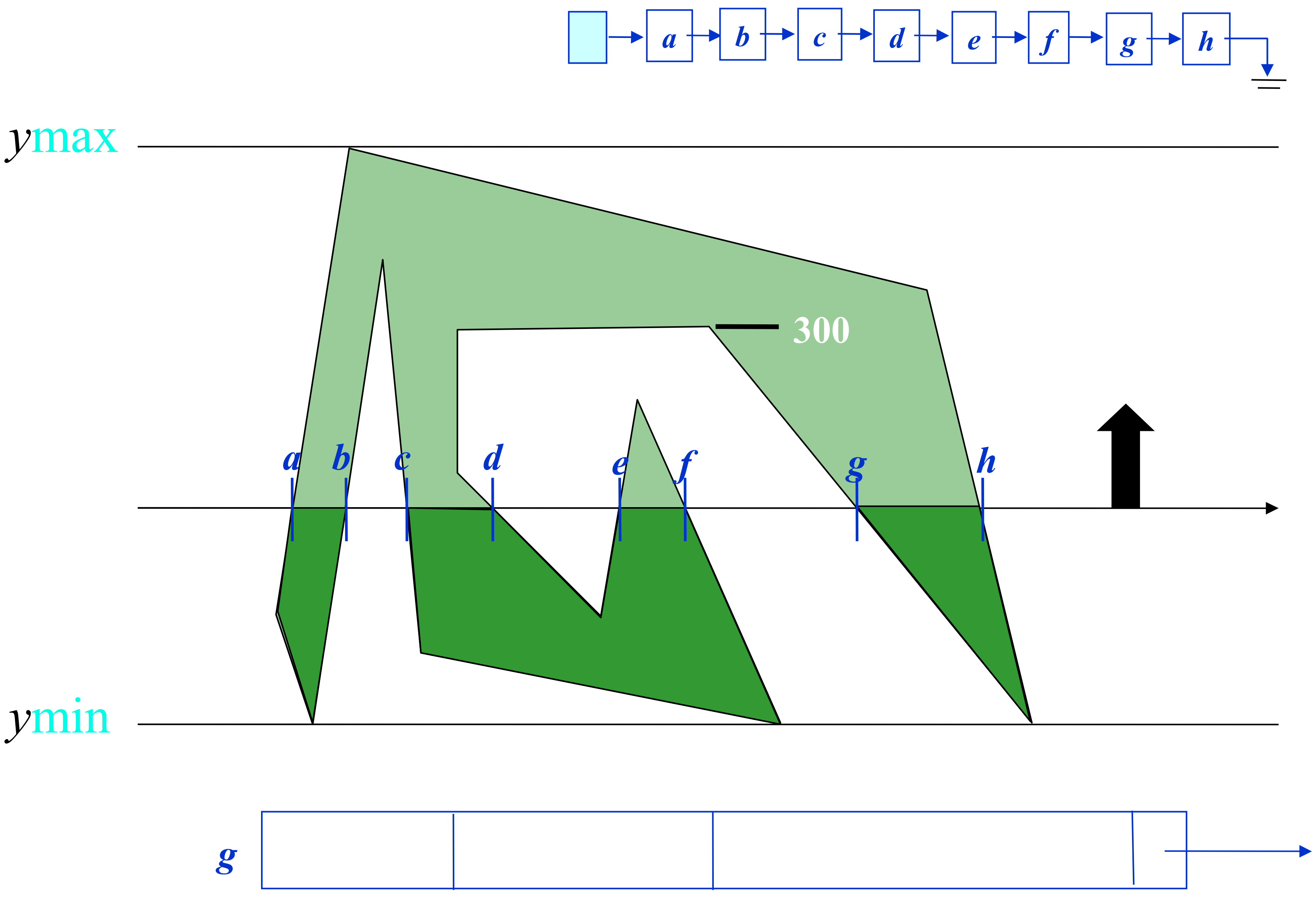


Fill the
scan line

Push seeds for the next step



rightmost left pixels, above & below

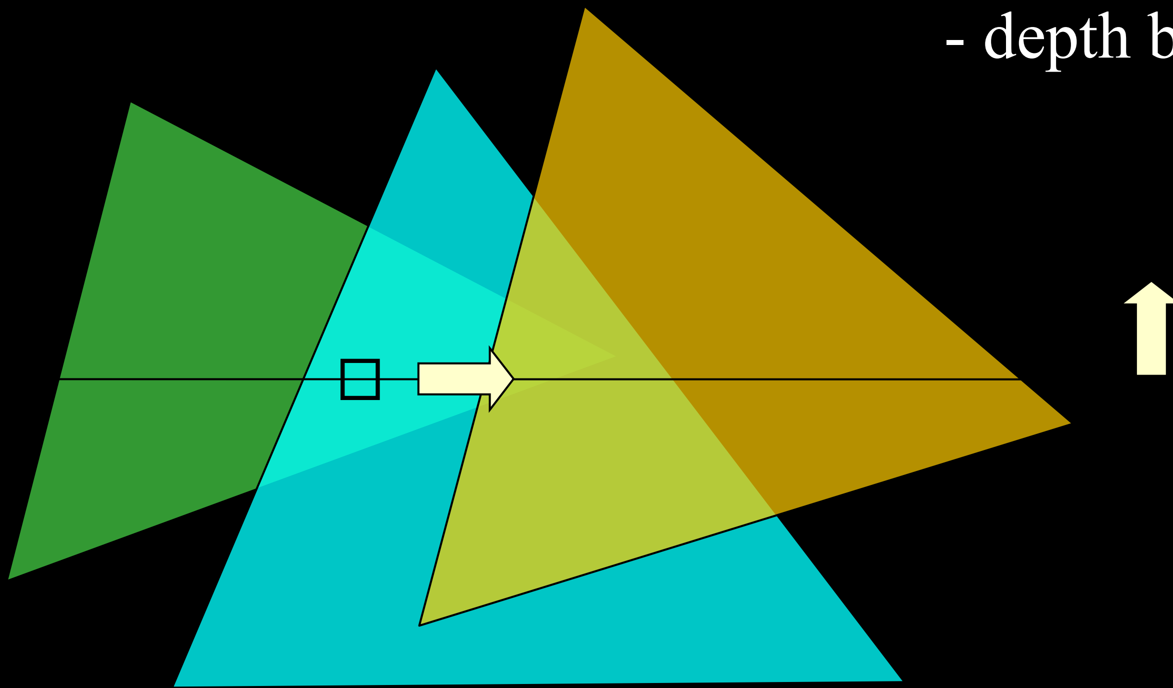


a
b
c
d
e
f
g
h
ymax
300
a
b
c
d
e
f
g
h
ymin
g

Z-Buffering

Two buffers

- screen buffer (color)
- depth buffer

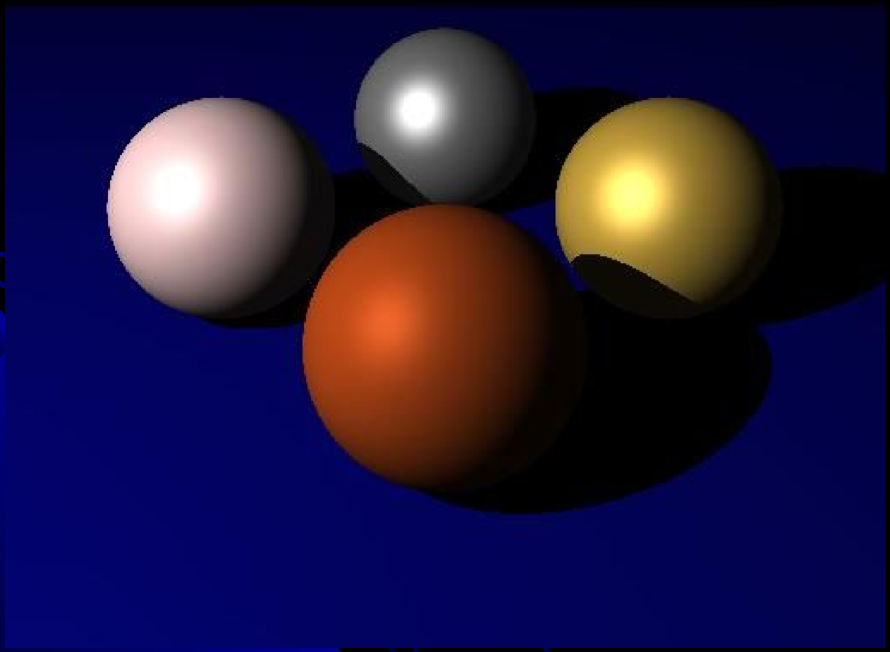


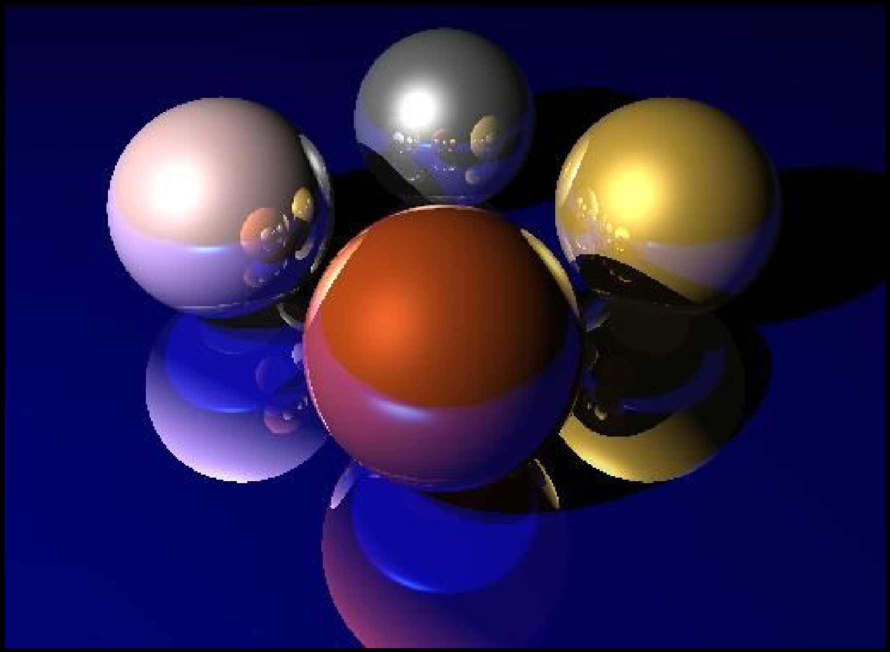
```
for each polygon
    for each scanline
        for each pixel in scanline
            update depth at pixel
            if pixel depth < buffered depth
                write to screen & depth buffers
```

Ray Tracing

Mani Thomas
CISC 440/640
Computer Graphics

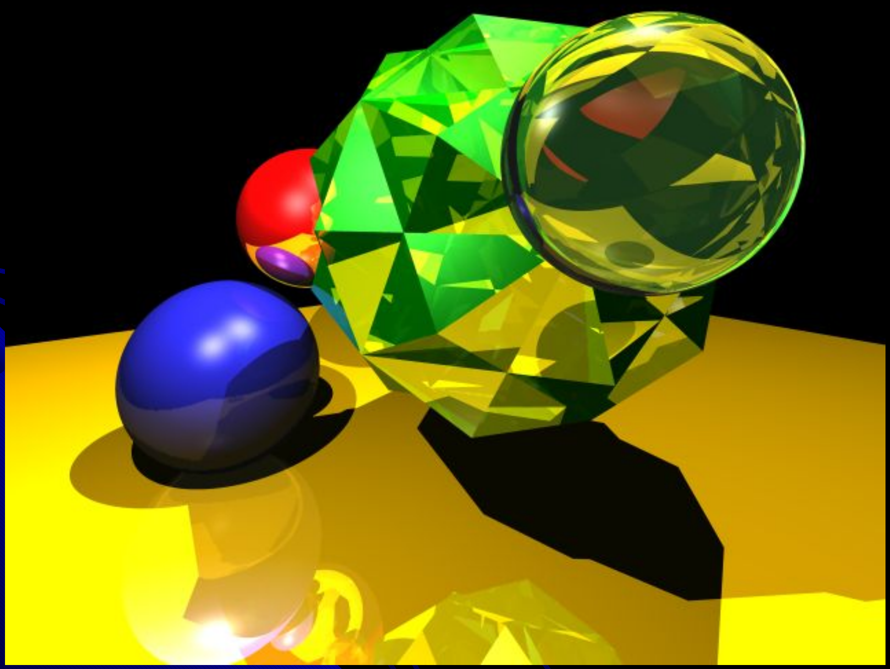
Fotorealismus

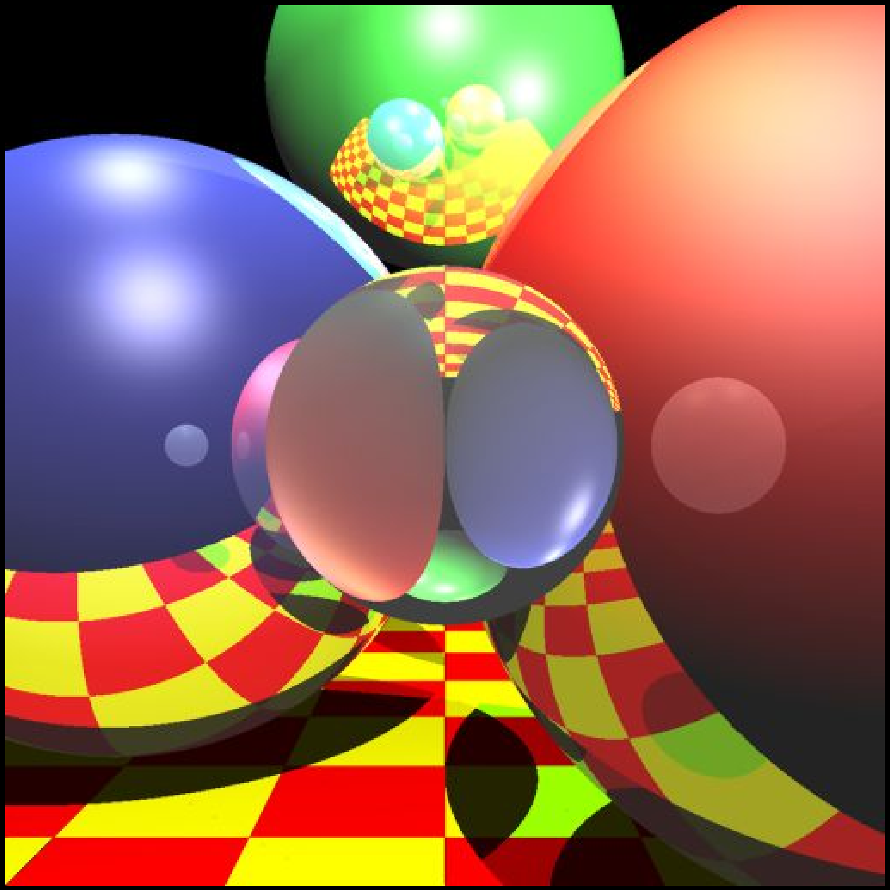




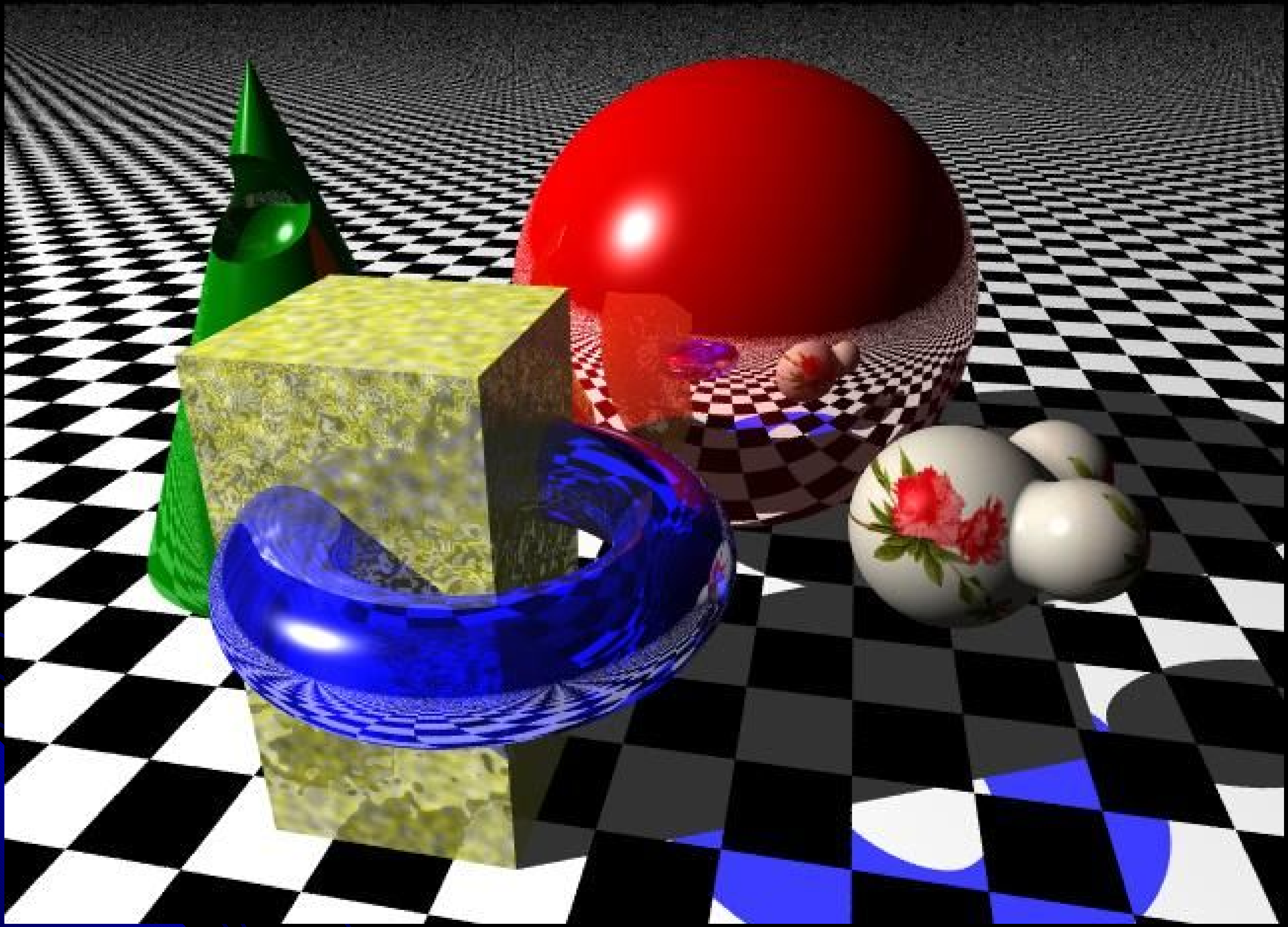
Created by David Derman – CISC 440

Fotorealismus





Created by Jan Oberlaender – CISC 640

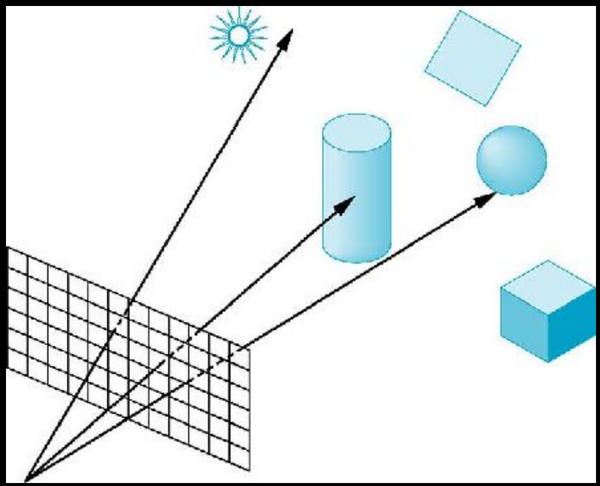


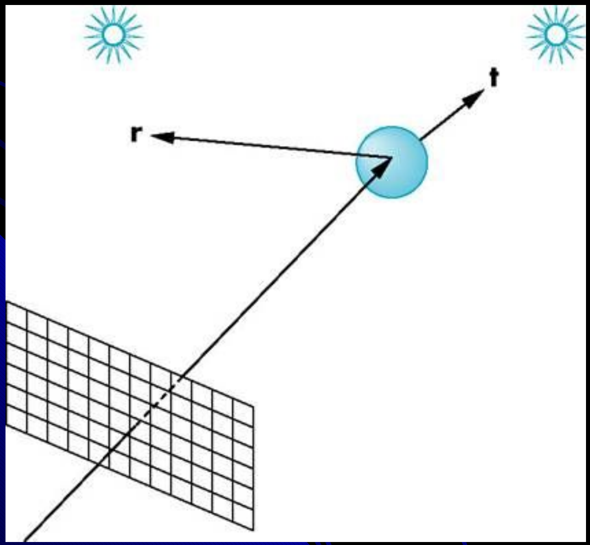
Created by Donald Hyatt
http://www.tjhsst.edu/~dhyatt/superap/povray.html

Úvod

- Co je Ray Tracing?
  - Ray Tracing je renderingová metoda založena na globálním osvětlení scény, která generuje realistické obrazy pomocí počítače.
  - V ray tracing-u, paprsek světla je sledován podél své dráhy v opačném směru.
    - Začínáme od kamery směrem ke zdroji světla a zjišťujeme stav objektů protínajících dráhu paprsku
    - Daný obrazový bod je nastaven na barvu odpovídající danému paprsku.
    - Pokud paprsek nenarazí na žádný předmět je bod nastaven na barvu pozadí.

Ray Casting/Tracing

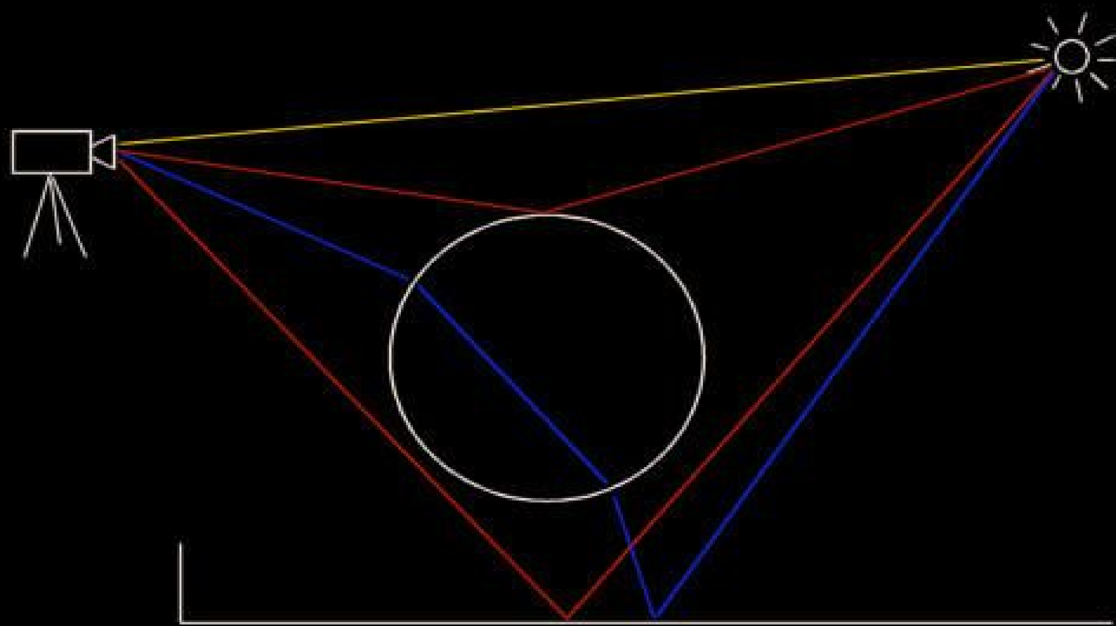




Courtesy: Angel

- Ray Casting
  - Paprsky se zastaví na prvním objektu
- Ray Tracing
  - Rekurze předcházejícího principu

Šíření světla



Typy paprsků

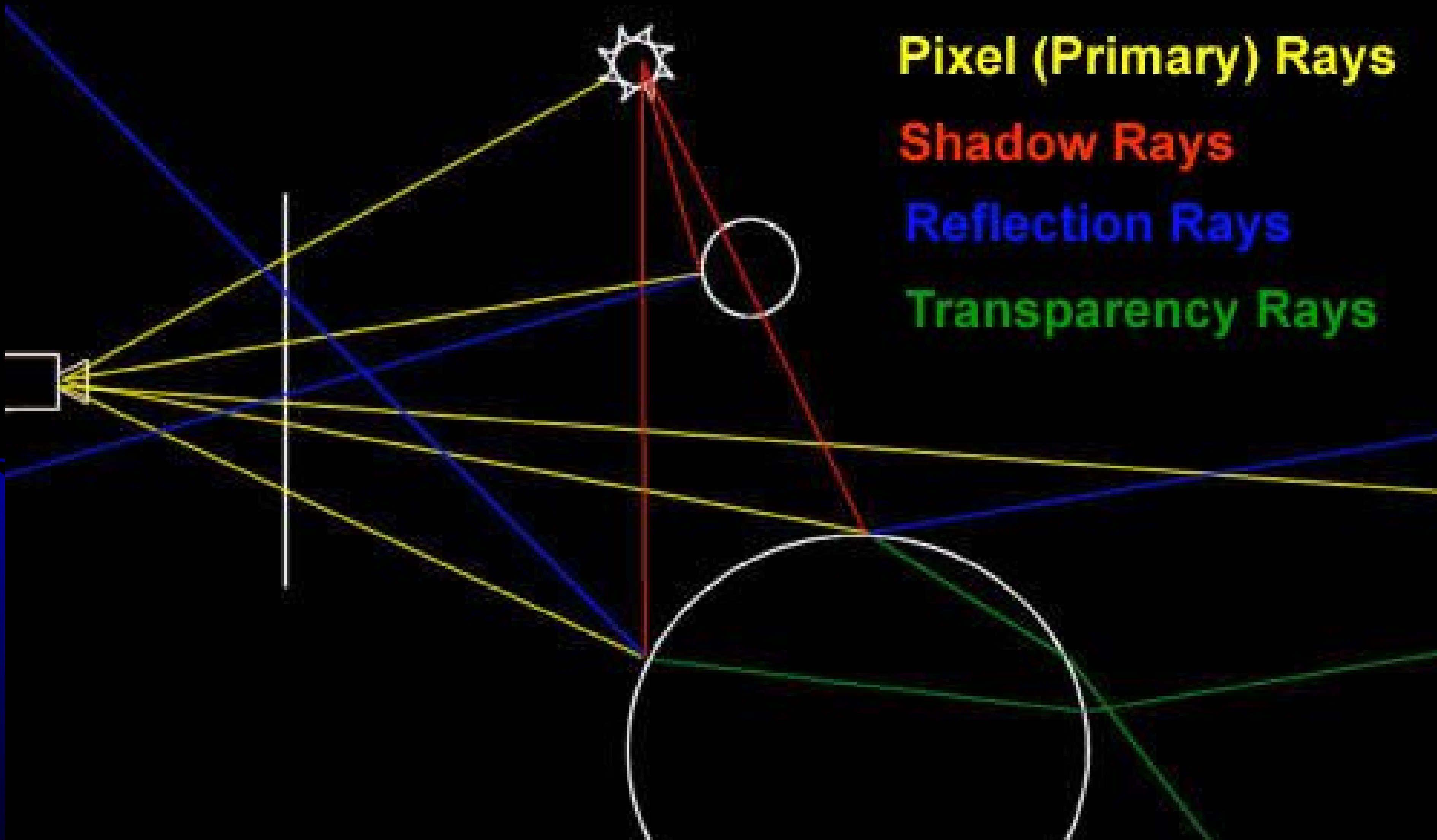


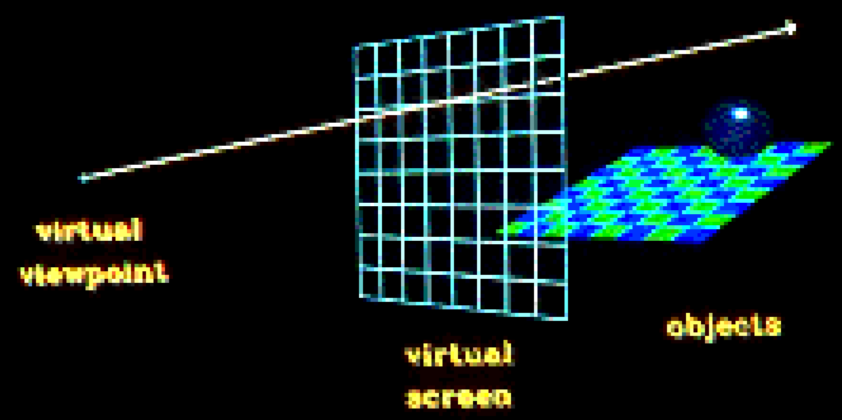
Image copyright Jacco Bikker.

Algoritmus – Ray casting

```
define the objects and light sources in the scene
set up the camera
for(int r = 0; r < nRows; r++)
  for(int c = 0; c < nCols; c++)
  {
   1. Build the rc-th ray
   2. Find all intersections of the rc-th ray with objects in the scene
   3. Identify the intersection that lies closest to, and in front of, the eye
   4. Compute the "hit point" where the ray hits this object, and the normal vector at that point
   5. Find the color of the light returning to the eye along the ray from the point of intersection
   6. Place the color in the rc-th pixel.
  }
```

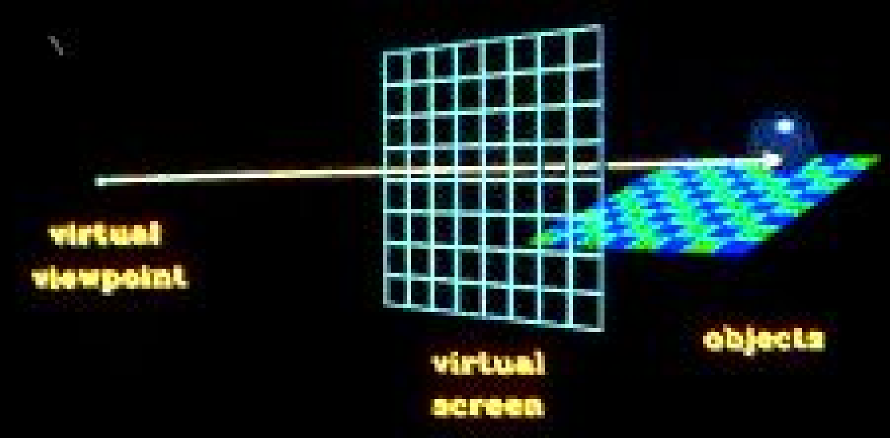
Courtesy F.S. Hill, "Computer Graphics using OpenGL"

Ray Tracing



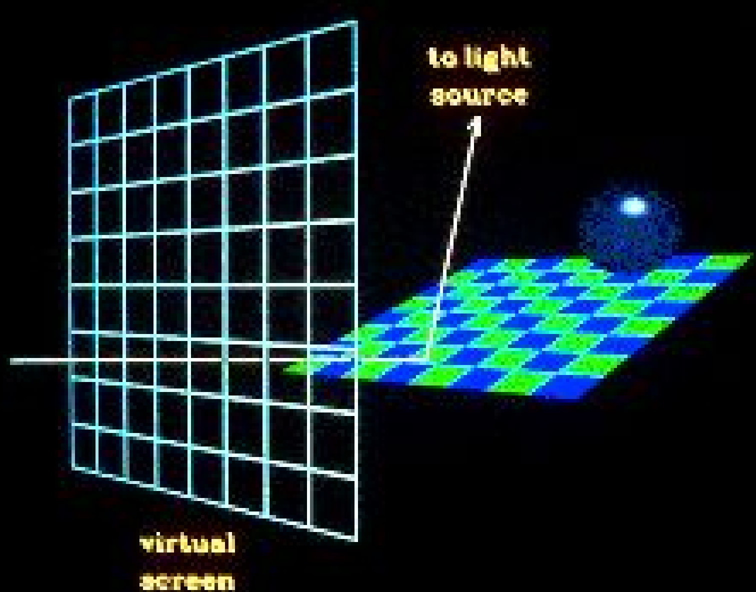
created by Michael Sweeny, et al for ACM SIGGRAPH Education slide set 1991

Ray Tracing



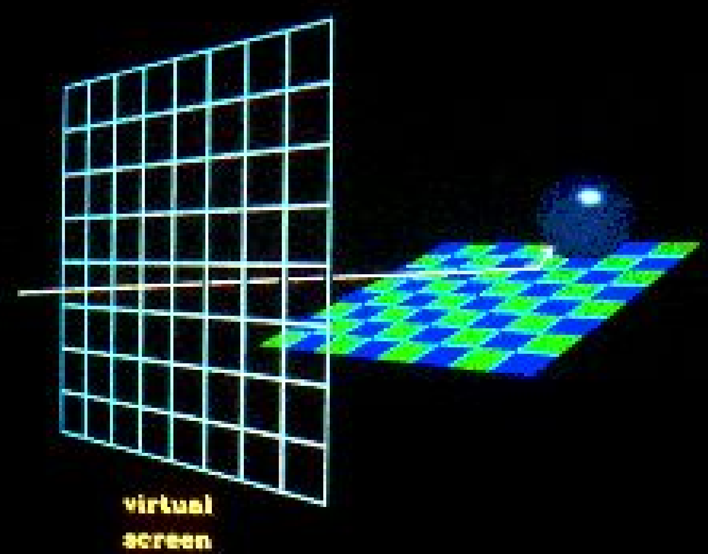
created by Michael Sweeny, et al for ACM SIGGRAPH Education slide set 1991

Ray Tracing



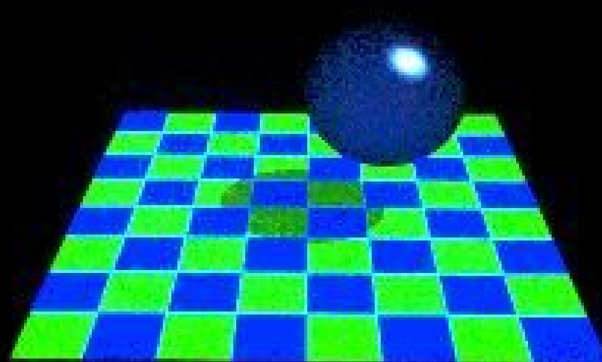
created by Michael Sweeny, et al for ACM SIGGRAPH Education slide set 1991

Ray Tracing



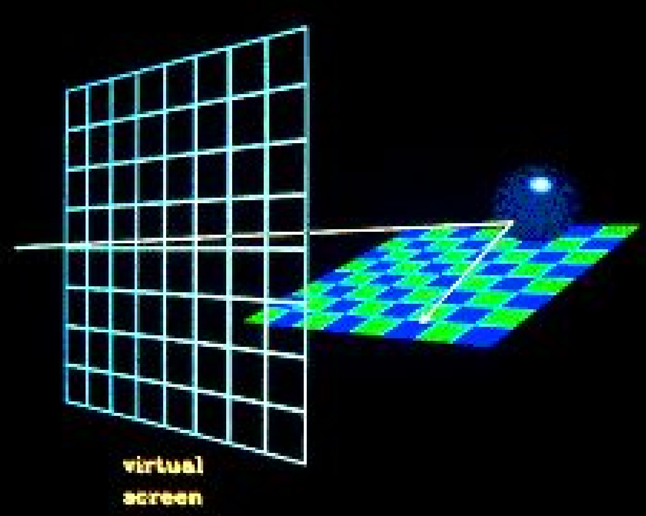
created by Michael Sweeny, et al for ACM SIGGRAPH Education slide set 1991

Ray Tracing



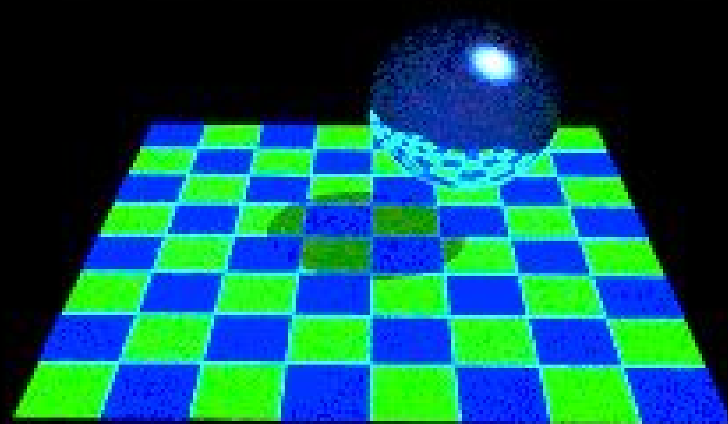
created by Michael Sweeny, et al for ACM SIGGRAPH Education slide set 1991

Ray Tracing



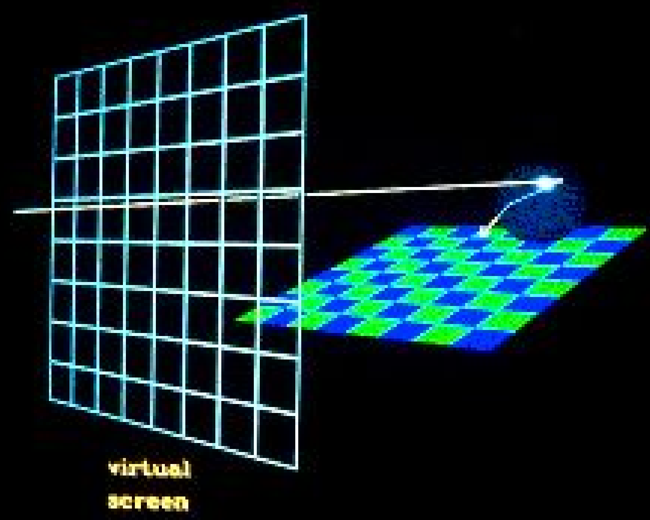
created by Michael Sweeny, et al for ACM SIGGRAPH Education slide set 1991

Ray Tracing



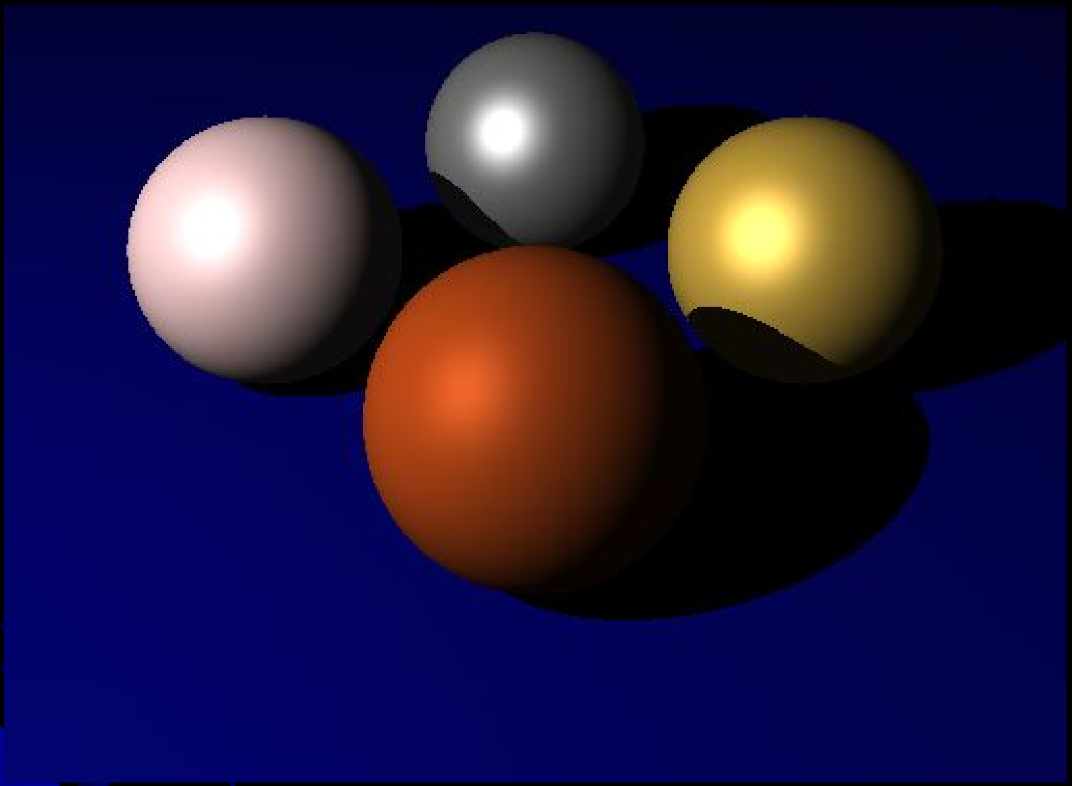
created by Michael Sweeny, et al for ACM SIGGRAPH Education slide set 1991

Ray Tracing

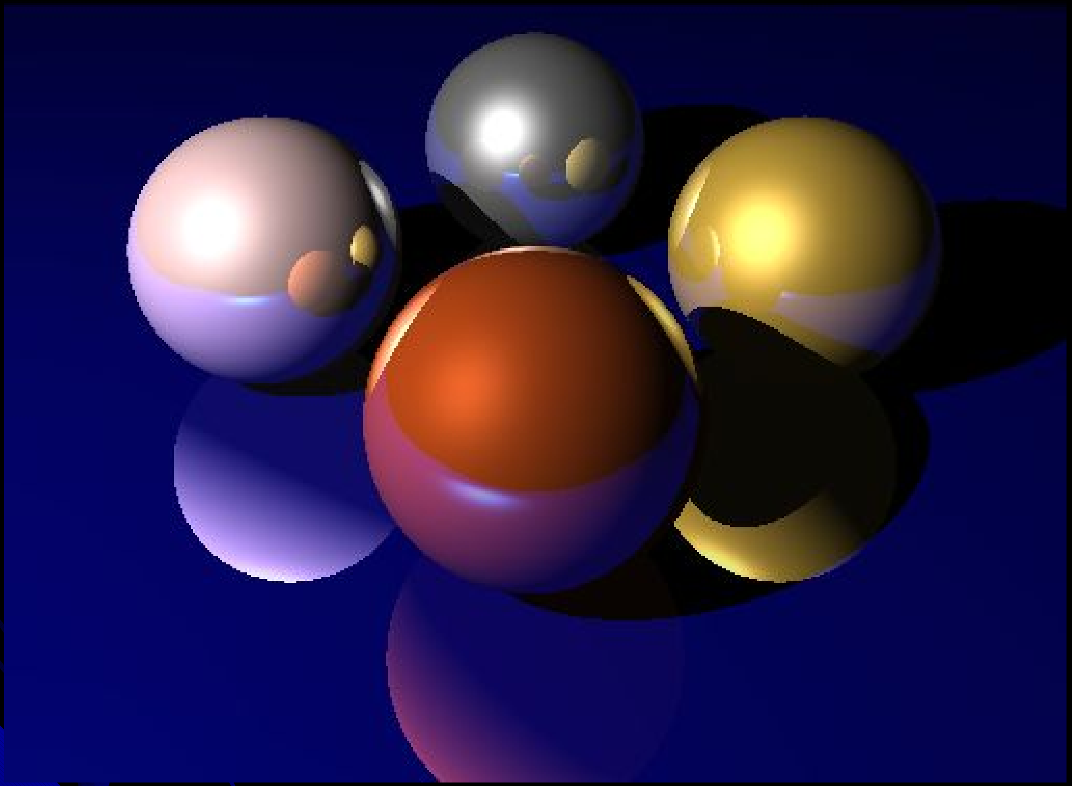


created by Michael Sweeny, et al for ACM SIGGRAPH Education slide set 1991

Odraz

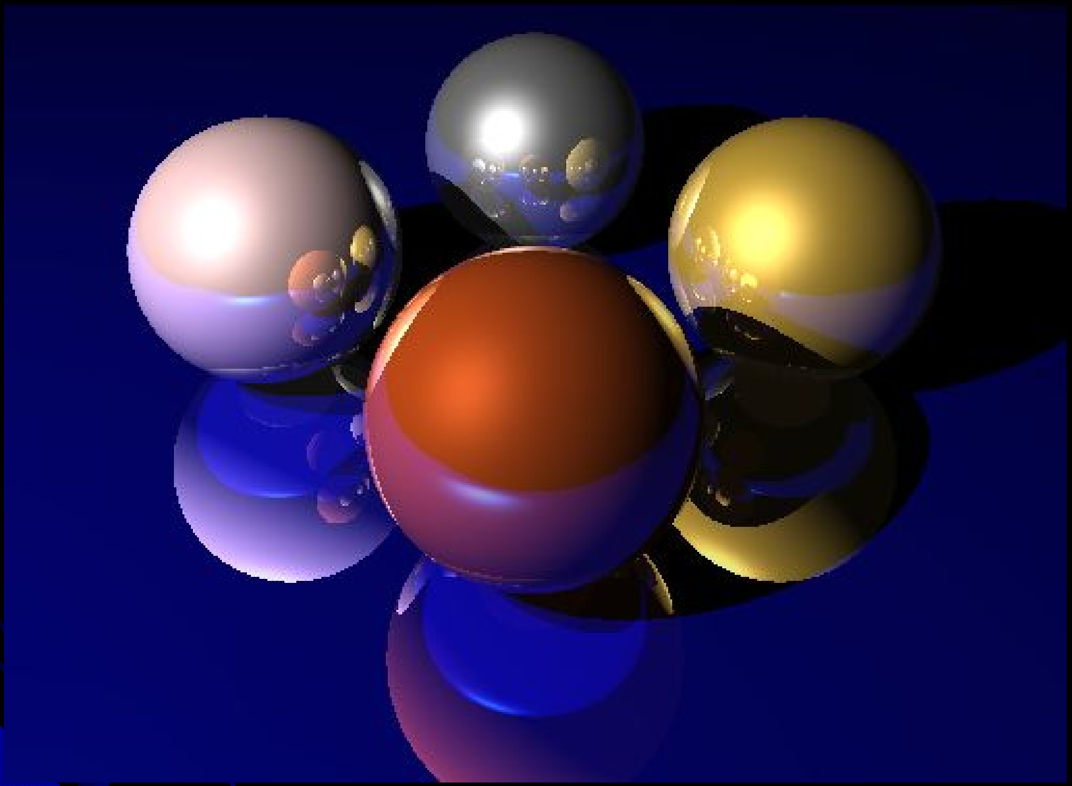


Odraz



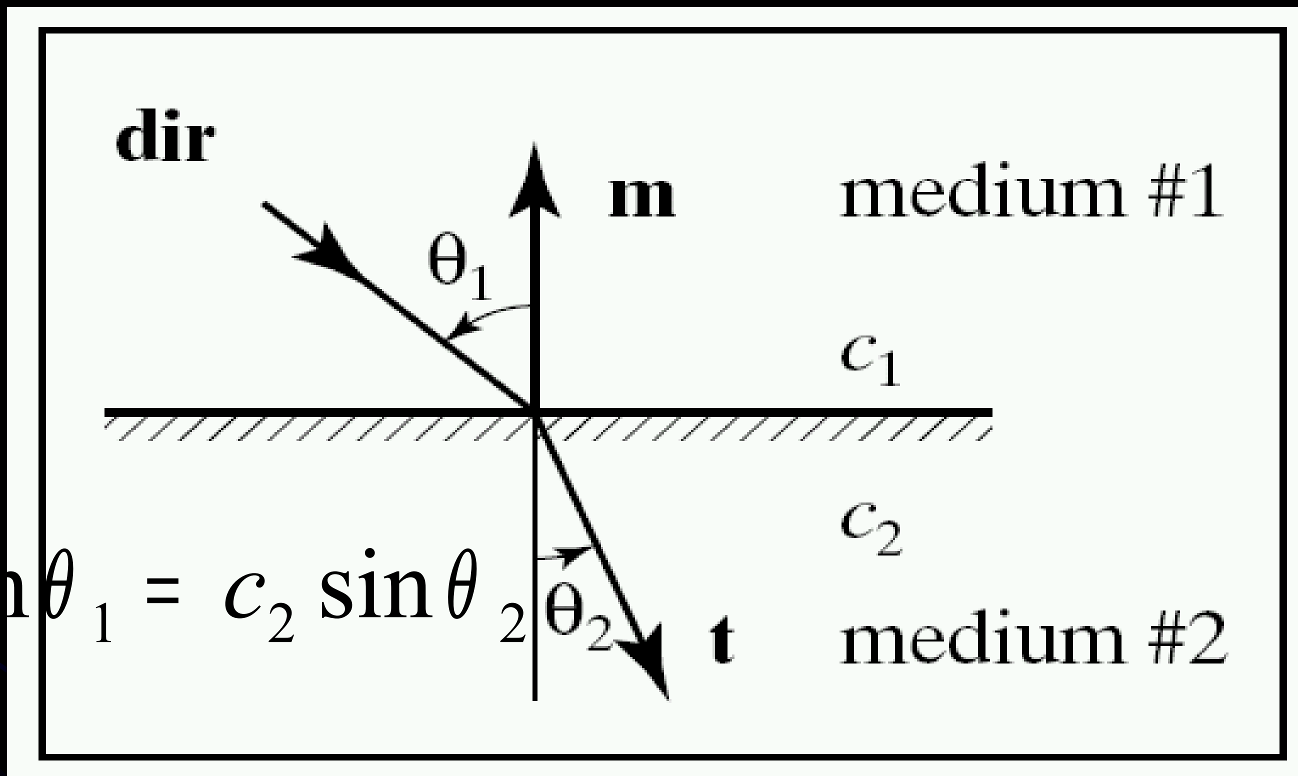
Created by David Derman – CISC 440

Odraz



Created by David Derman – CISC 440

Lom světla



$$\ldots n\theta_1 = c_2 \sin\theta_2$$

Courtesy F.S. Hill, "Computer Graphics using OpenGL"

Jiné efekty

Hloubka ostrosti

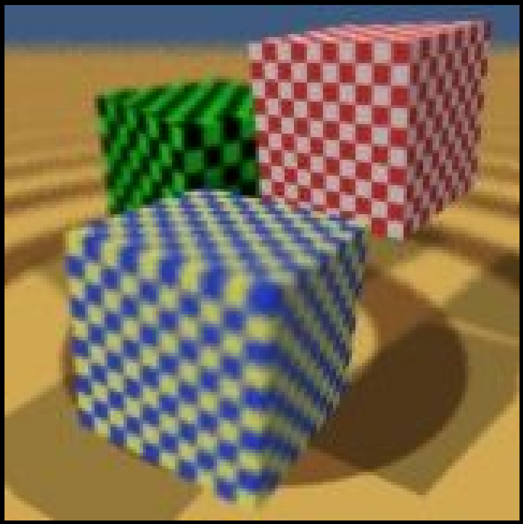


Image copyright
Josef Pelikan
http://cgg.ms.mff.cuni.cz/gallery/

Jiné efekty

Rozmazání pohybem

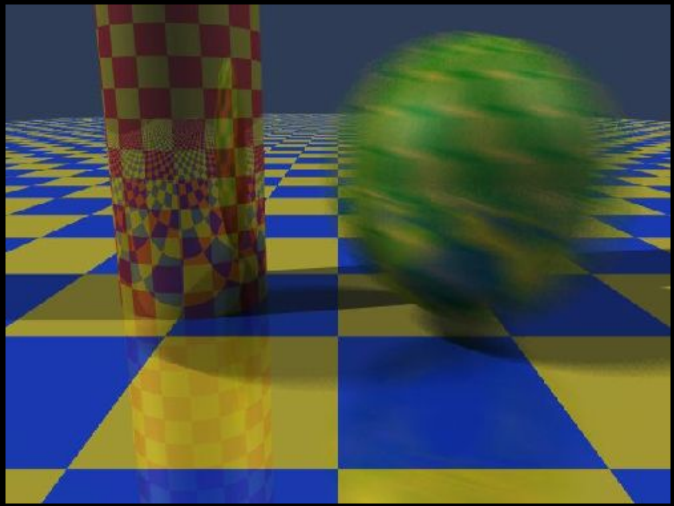
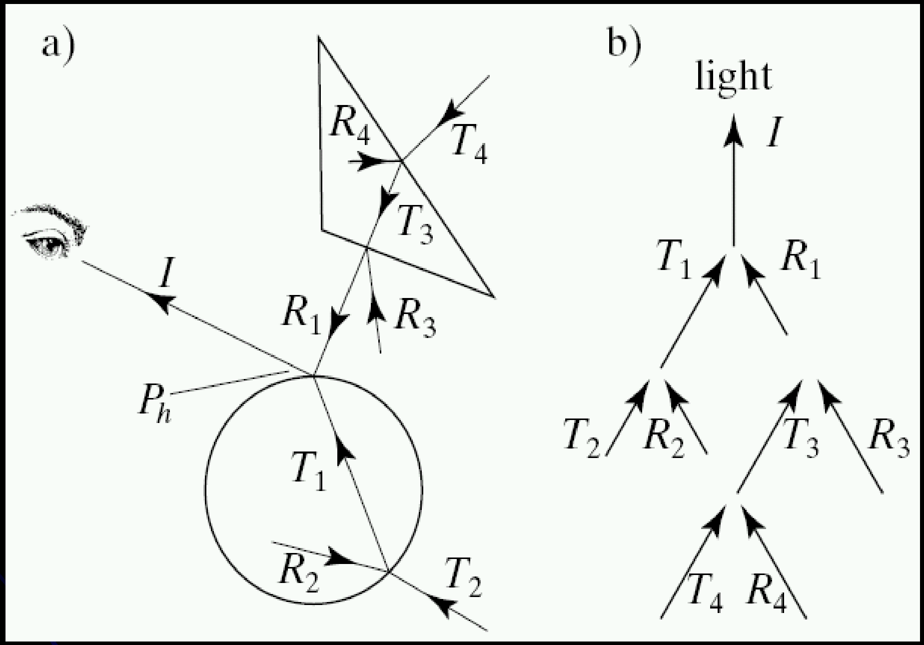


Image copyright
Josef Pelikan
http://cgg.ms.mff.cuni.cz/gallery/

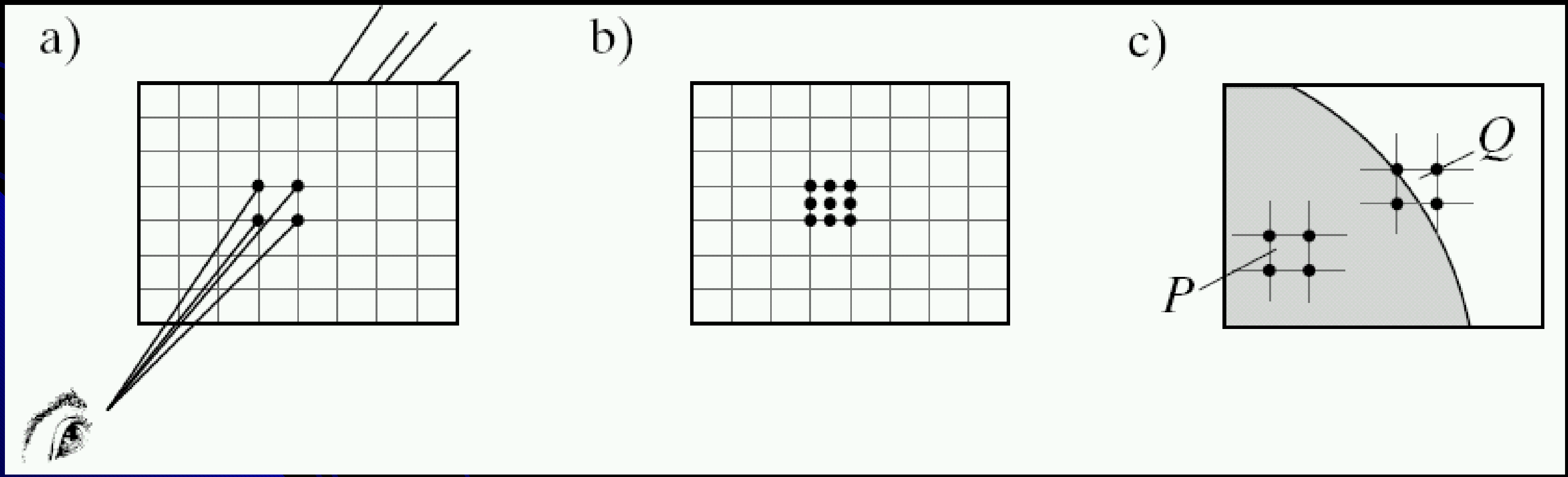
Strom světla

- Informace o paprsku sčítají



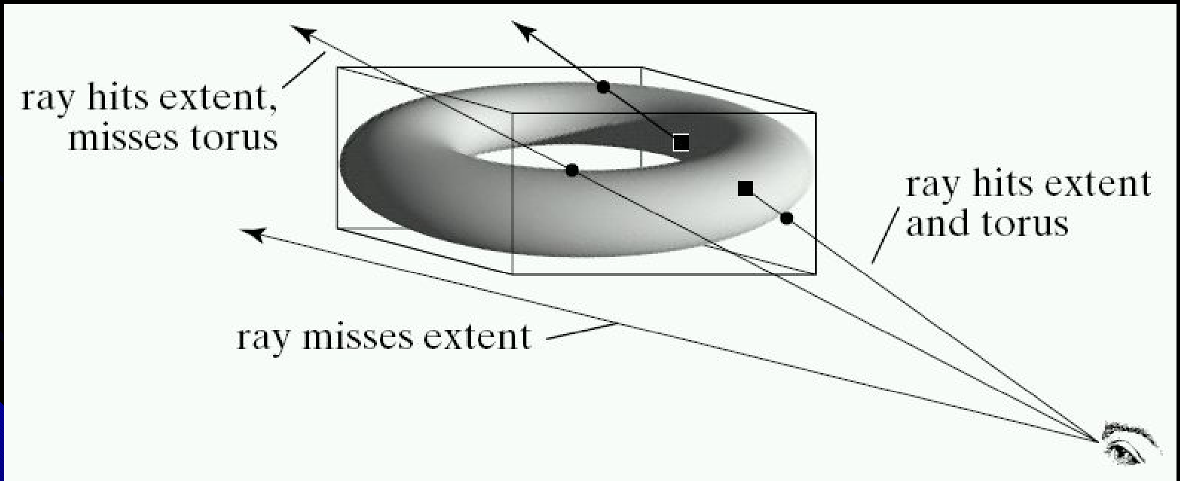
Super-sampling

- Vyhlazení hran

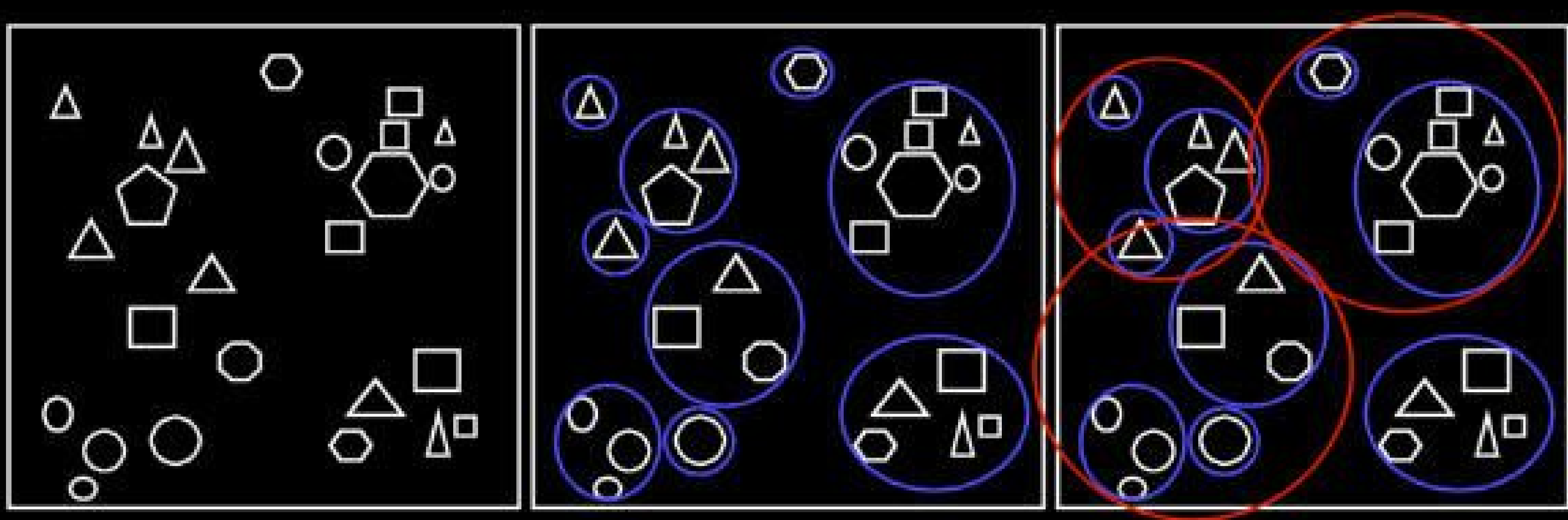


Courtesy F.S. Hill, *"Computer Graphics using OpenGL"*

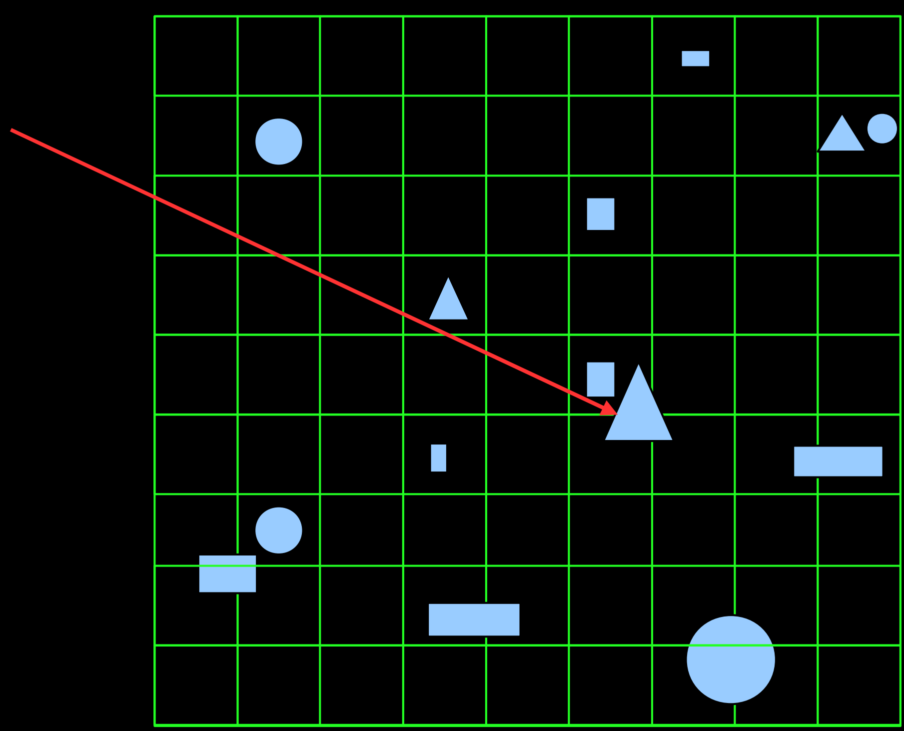
Obal


ray hits extent,
misses torus
ray hits extent
and torus
ray misses extent

Obal skupiny objektů



Prostorové rozdělení úloh



Nerovnoměrné rozdělení na podprostory

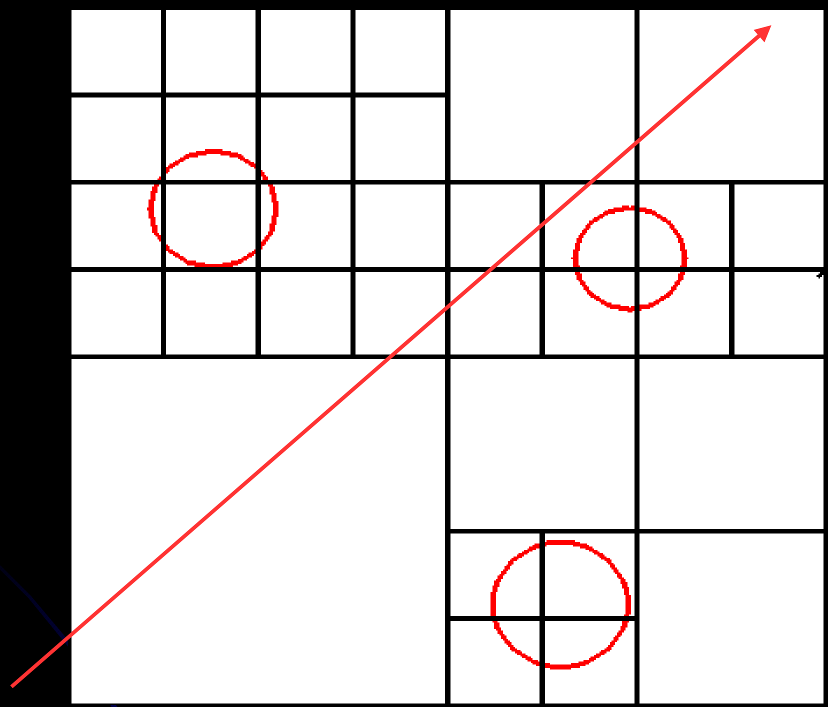


Image copyright
Worcester Polytechnic Institute

References

- Textbooks
  - F. S. Hill, “Computer Graphics Using OpenGL”
- Commonly used ray tracing program (completely free and available for most platforms)
  - http://www.povray.org/
- Interesting Links
  - Interactive Ray Tracer – Alyosha Efros
    - http://www.cs.berkeley.edu/~efros/java/tracer/tracer.html
- Ray Tracing explained
  - http://www.geocities.com/jamisbuck/raytracing.html
  - http://www.siggraph.org/education/materials/HyperGra

Structure Visualization Tools

Written by James Coleman

Presented by Xiang Zhou

Structure Visualization

- One of the primary activities in proteomics R&D is determining and Visualizing the 3D structure of proteins in order to find where drugs might modulate their activity.
- Other activities include identifying all of the proteins produced by a given cell or tissue and determining how these proteins interact.

- BIOINFORMATICS COMPUTING, p.186, Bryon Bergeron, M.D., Prentice Hall 2002

Some Common Tools

- 100’s of visualization tools have been developed in bioinformatics.
- Many are specific to hardware such as microarray devices.
- Shareware utilities for PC’s
  - PDB Viewer, WebMol, RasMol, Protein Explorer, Cn3D
  - VMD, MolMol, MidasPlus, Pymol, Chime, Chimera

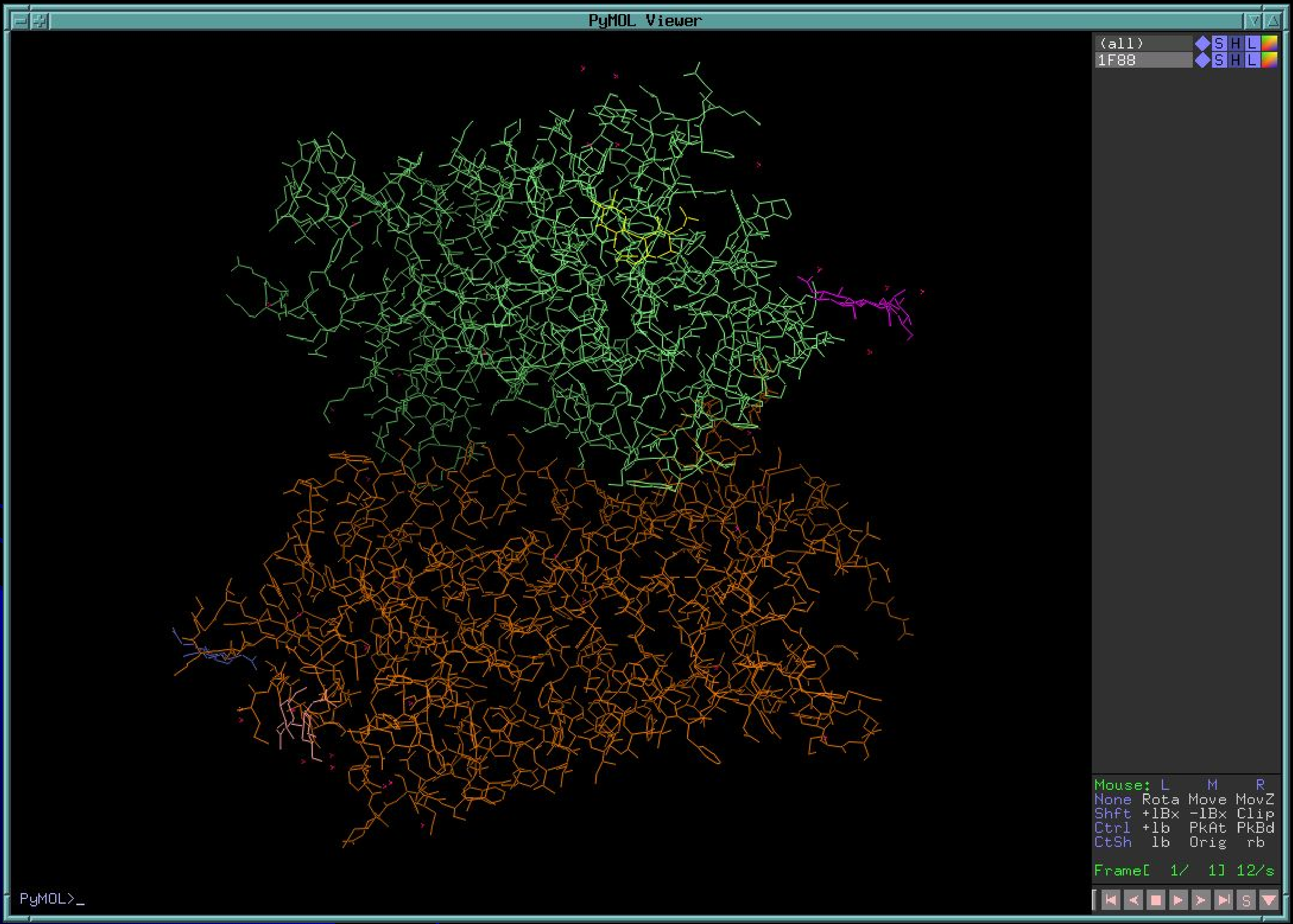
Application Feature Summary

Feature	RasMol	Cn3D	PyMol	SWISS-PDBViewer	Chimera
Architecture	Stand-Alone	Plug-in	Web-Enabled	Web-enabled	Web-enabled
Manipulation Power	Low	High	High	High	High
Hardware Requirements	Low/Moderate	High	High	Moderate	High
Ease of Use	High; command line	Moderate	Moderate	High	Moderate;GUI +command line
Special Features	Small Size; easy install	Powerful GUI	GUI; ray tracing	Powerful GUI	GUI; collaboration
Output Quality	Moderate	Very high	High	High	Very high
Documentation	Good	Good	Limited	Good	Very good
Support	Online; Users groups	Online; Users groups	Online; Users groups	Online; Users groups	Online; Users groups
Speed	High	Moderate	Moderate	Moderate	Moderate/Slow
OpenGL Support	Yes	Yes	Yes	Yes	Yes

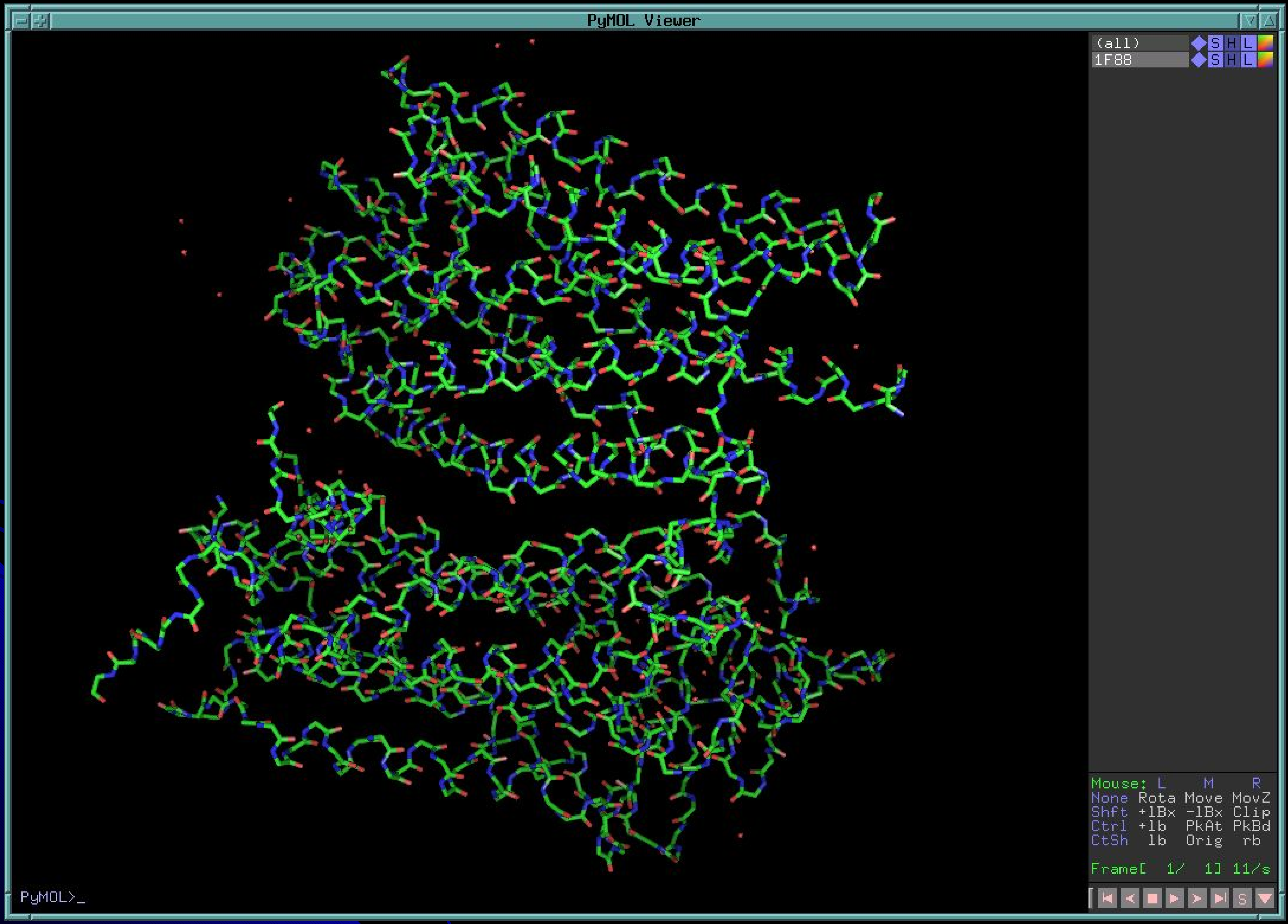
Molecule Representations

Wireframe	Bonds and Bond Angles	
Ball and Stick	Shows Atoms, Bonds and Bonds Angles	
Ribbon diagrams	Shows Secondary Structure	
Van der Waals surface Diagram	Shows Atomic Volumes	
Backbone	Shows Overall Molecular Structure	

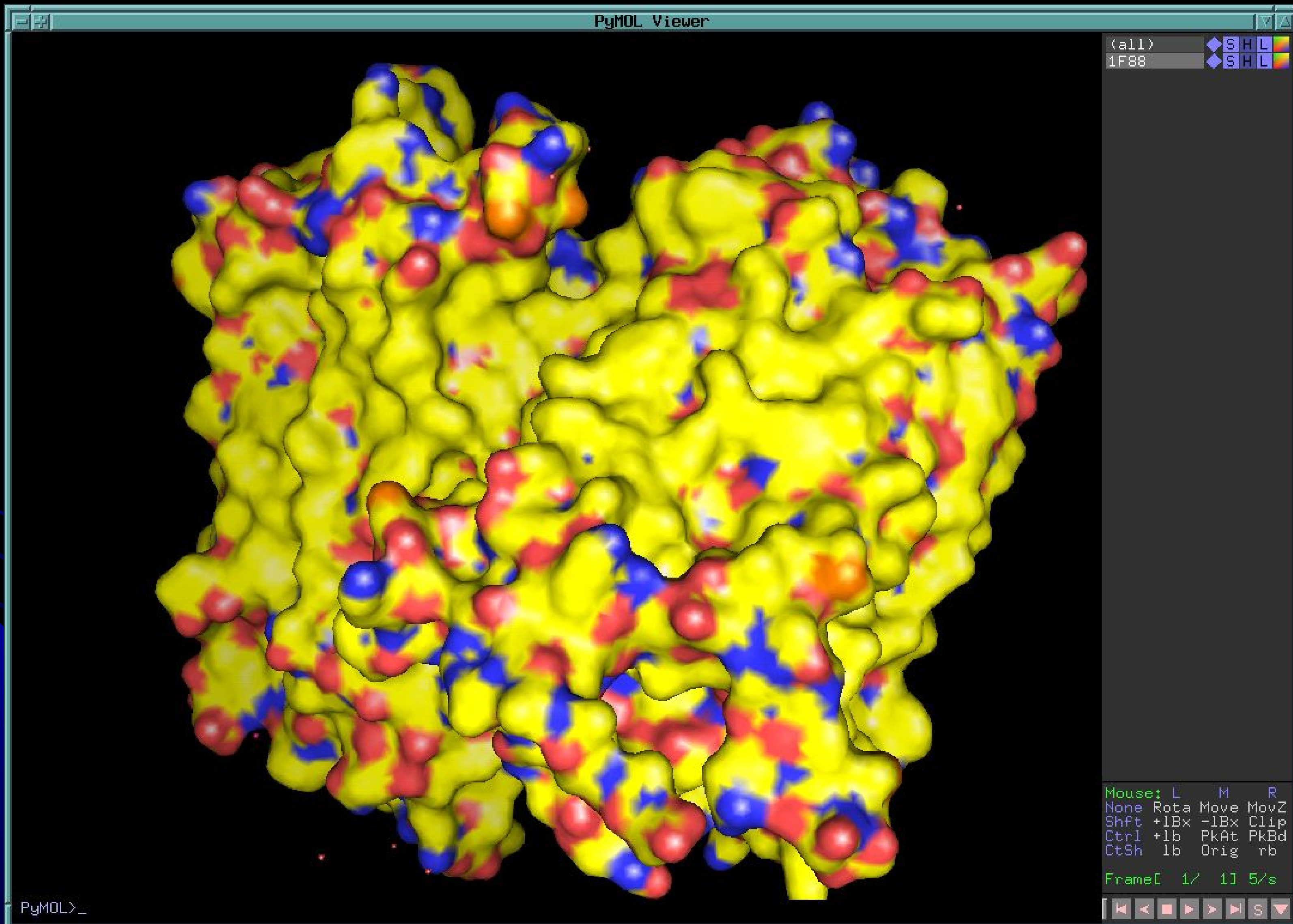
Wireframe used to show individual chains:



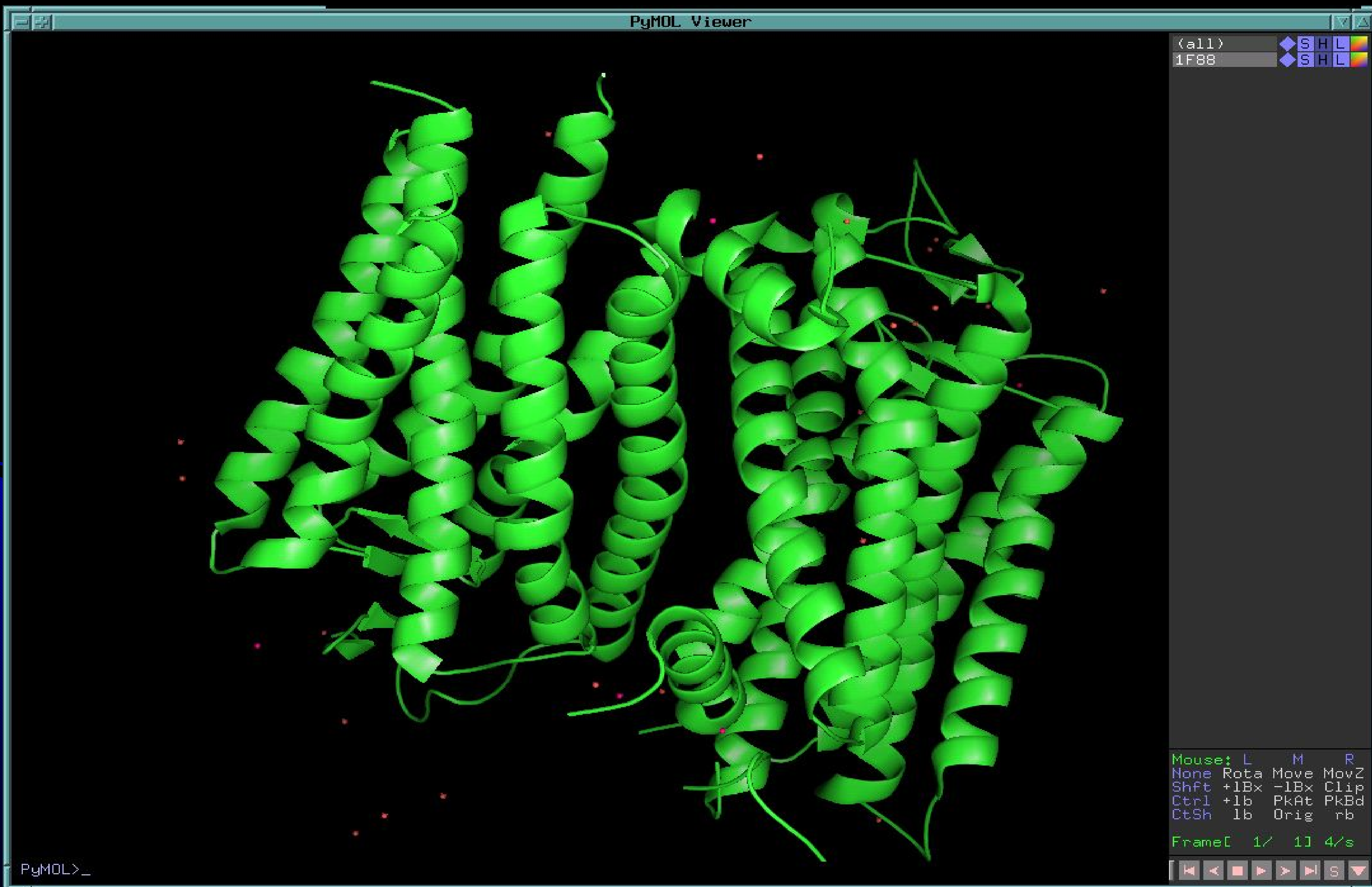
Stick view showing atoms and bonds:



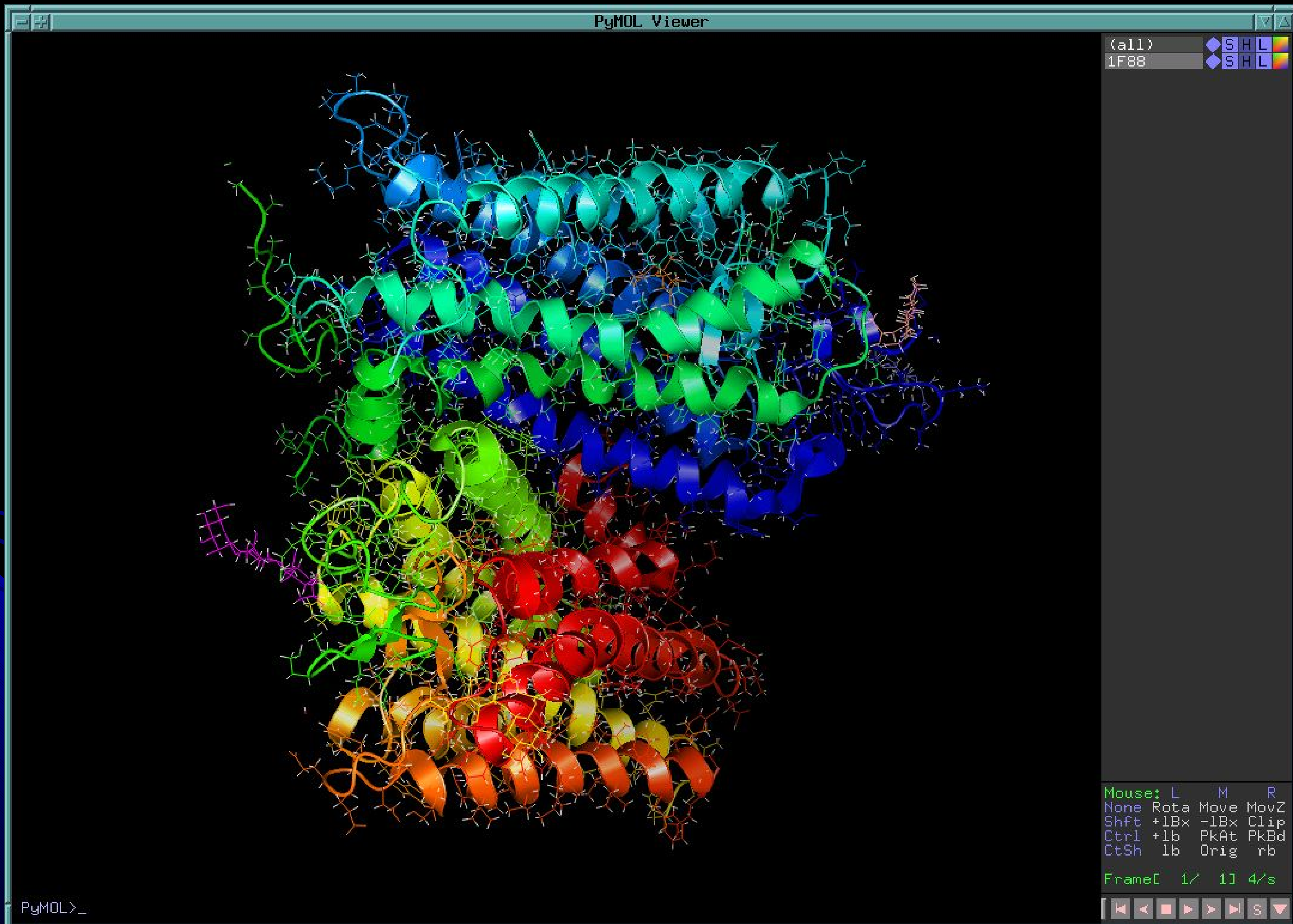
Surface View showing surface fields:



Ribbon view of secondary structure:



Distinct geometrical features by color:



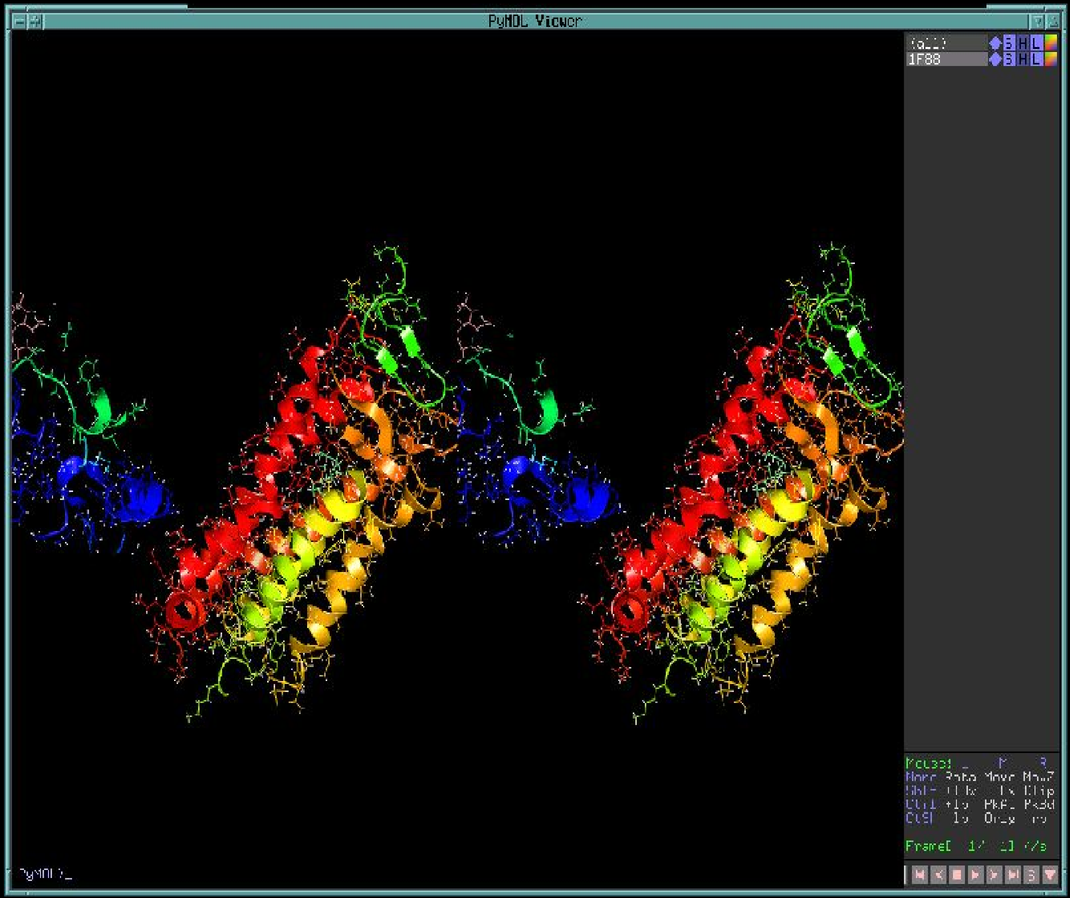
Other properties that can be Visualized

- MolMol supports the display of electrostatic potentials across a protein molecule.
- MidasPlus (a predecessor of Chimera) allows for the editing of sequences visually to see the effects of point mutations.

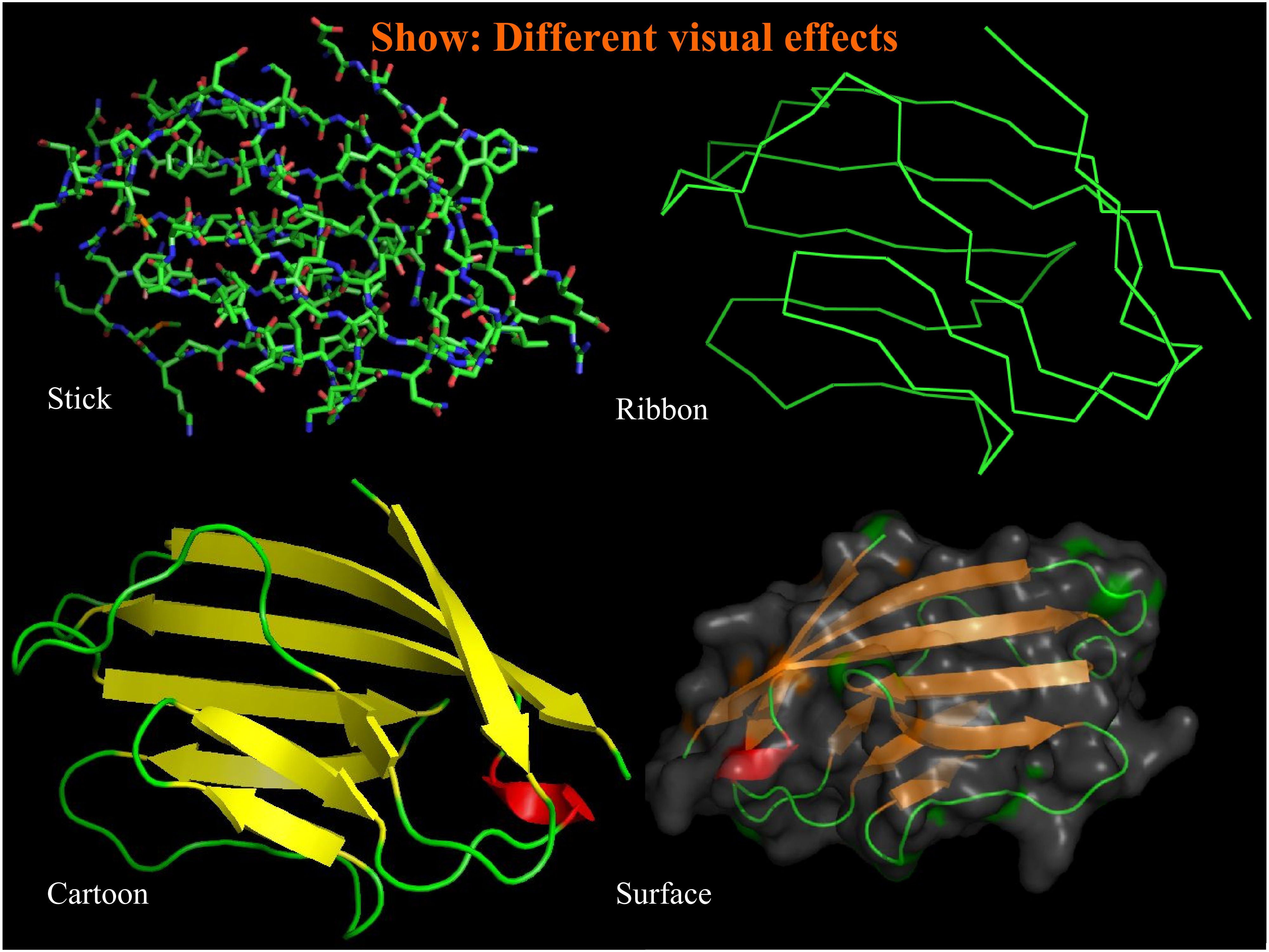
For Protein interactions, we need a metaphor that reveals dynamics

- Haptic Joystick: Provides force feedback when user manipulates a molecule near another one.
- 3D Goggles combined with haptic gloves to feel electrostatic potentials and see tertiary structure dynamics.
- PyMol provides scripting that can produce a movie in 3D of the geometrical relationship between multiple proteins.

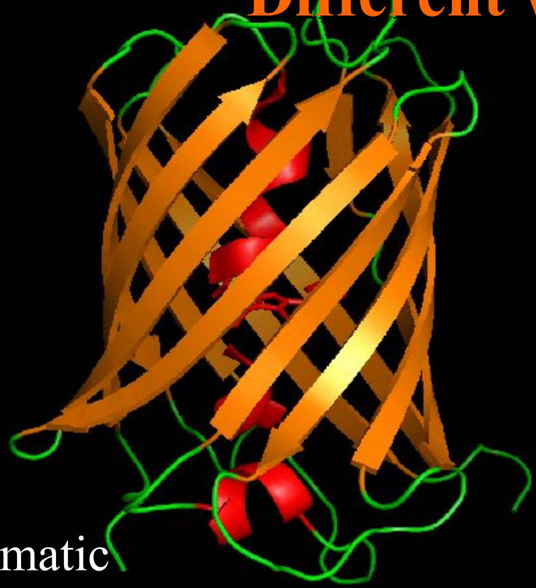
Stereo view of interaction of two proteins. Scripting allows for the movement of individual molecules creating a movie.



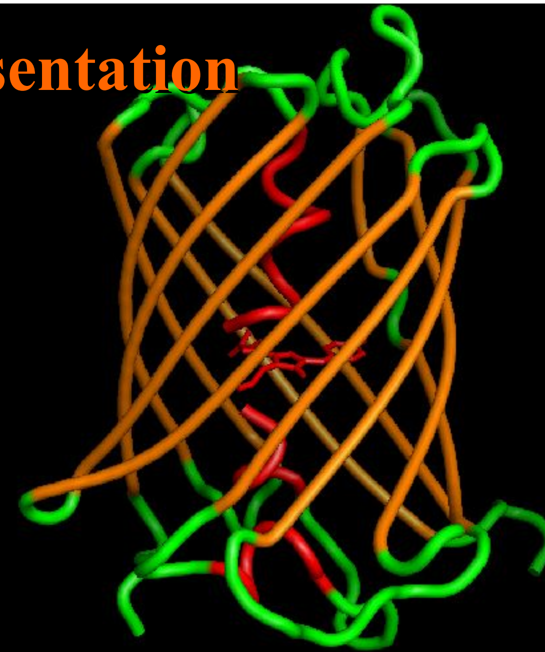
Show: Different visual effects



Different way of cartoon presentation

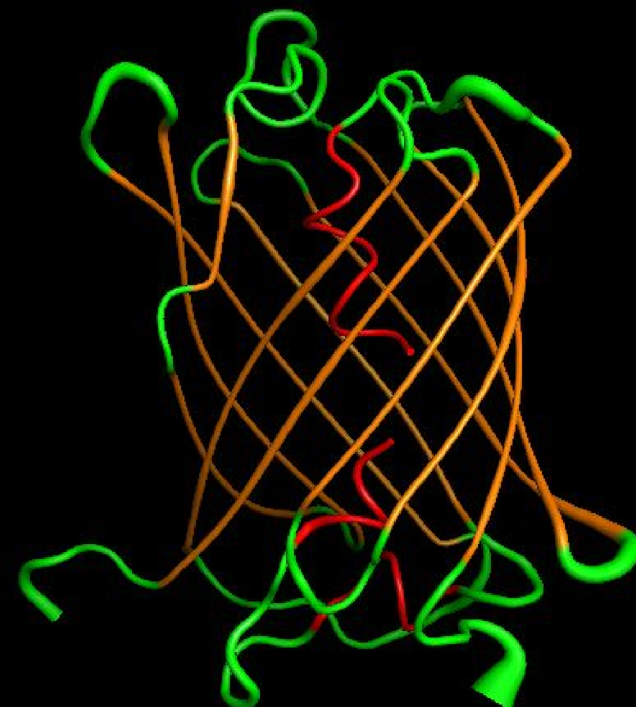


automatic



tube

loop



putty