Visualisation, Rendering and Animation

2 VO / 1 KU

Heinz Mayer, Franz Leberl & Andrej Ferko

ferko@icg.tu-graz.ac.at

Agenda

- Organisation
- Color
- Light-Material Interaction

Organisation

- *Timetable, schedule*
- *References - besides slide shows (electronic, auch Deutsch), web... Mount lectures* www.sccg.sk/~ferko
  - *A. & M.Watt:* Advanced Animation and Rendering Techniques
  - *J. Foley, et al.:* Computer Graphics, Principles and Practice
- *Class Assignments*
- *E-mail, graphics&vision student contest*
- *Andrej Ferko: Work Page...* *WELCOME = VITAJTE*
- *Better pages:*
- *SIGGRAPH EG SCCG CESCG SIGGRAPH2003 SIGGRAPH2002 CGpapers Citeseer Helwig's Calendar ACM SIGGRAPH Calendar IEEE Calendar Techexpo pg.netgraphics.sk Prusinkiewicz Webby Facedemo Helwig's SmileyDict. D.Mount's Book GOOOOOGLE*
  - *http://www.cs.umd.edu/~mount/427/Lects/427lects.pdf*

Jan 2004 days - blocked

Mo	Di	Mi	Do	Fr
12. 1.	13. 1.	14. 1.	15. 1.	16. 1.
i5	i5	i5	D1.10	D1.10, i11
8.00-12.00	10.00-11.30 ?	10.00-11.30 ?	11.00-13.00-	9.00-10.15-
13.00-16.00	12.00-14.00	12.00-14.00 midterm	15.00-17.00-	13.00-14.00

Jan 2004 days - blocked

Mo **19. 1.** ***i11***	**Di** **20. 1.** ***i11***			
Midterm?	*9.00-* *11.30* *?*			
13.00- *17.00*	*12.00-* *13.00* ***final***			

Neue Professur @ ICG

- Donnerstag, Professur "VR und Computergrafik"
- 11:00 Matthias Teschner on Collision Detection
- 13:00 Wolfgang Müller on Zukunft von CG & VR
- 15:00 Marc Ebner on Bidverarbeitung & Grafik
- 17:00 Dirk Bartz on Virtuelle Medizin
- --- Ort: Seminarraum IICM, Inffeldgasse 16c, EG, Raum Nr. D1.10 ---
- Freitag, Professur "VR und Computergrafik"
- 9:00 Wolfgang Stürzlinger TBA
- 13:00 Dieter Schmalstieg on Mobile AugmentedR
- --- All professorial lectures included into Vis&Anim/AKCG classes ---

2nd Unit - Content

4. Color (Foley, van Dam, Chap.13)

- Color metrics, color models
- Color reproduction

5. Light-Material Interaction (Watt)

- Categories of Lighting Models
- Local Illumination Models
- Lighting Optimization (Möller, Haines)
- Shadow Generation

What is a color?

- Physical description: wavelengths
- Psychological perception: stimulus
- Computer description: color models
- - different sets of basis and coordinates
- - dithering/halftoning
- Color correction: ensure that percepted colors are correct

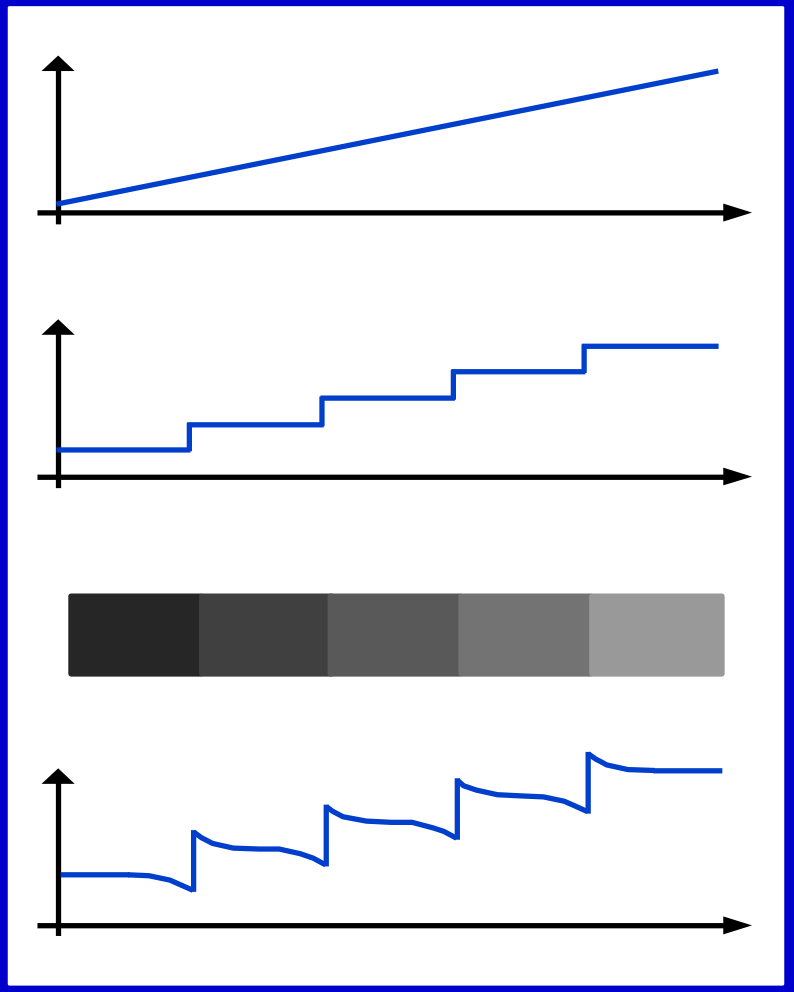
  - by N. Holzschuch, UCT 1996: Color fidelity and Color spaces

Color and Color metrics

- Meaning perception
- Metrics means measure relation of colors to each other
- Light rays from self-luminous object or reflected at the object surface (electromagnetic wave)
- Transition from color stimulus evaluation to the experience of color

Experience of Color

- Color stimulus (given by object properties and illumination)
- Properties of surrounding objects
- Visual system of the viewer
- „Mach-band“ effect



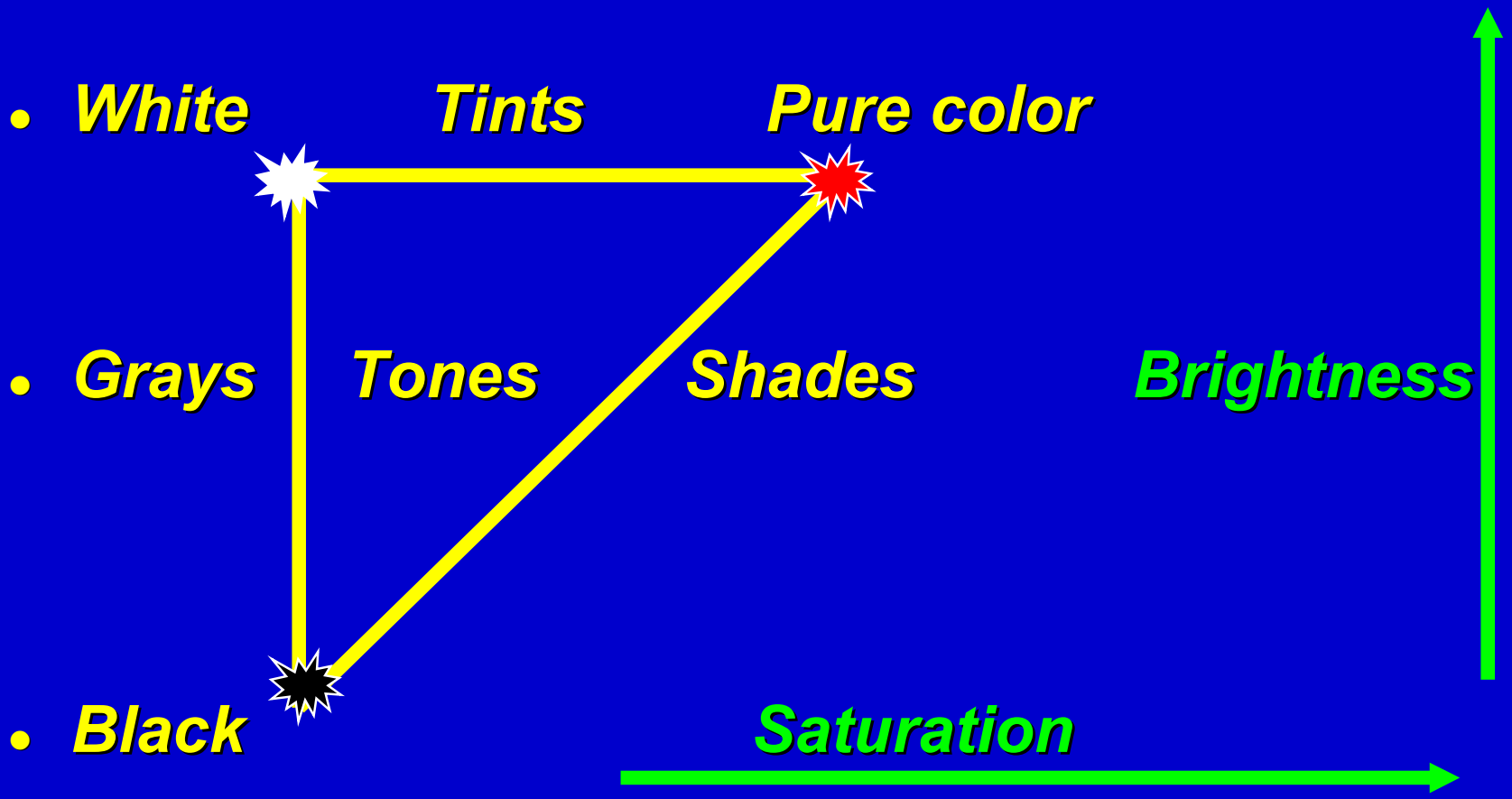
Perceptual Term vs. Colorimetry

- Hue — Dominant wavelength
- Saturation — Excitation purity

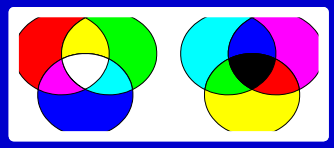
- Lightness (reflecting objects) — Luminance
- Brightness (self-luminous objects) — Luminance

- Wavelengths from 400 nm to 700 nm

Artist Perception

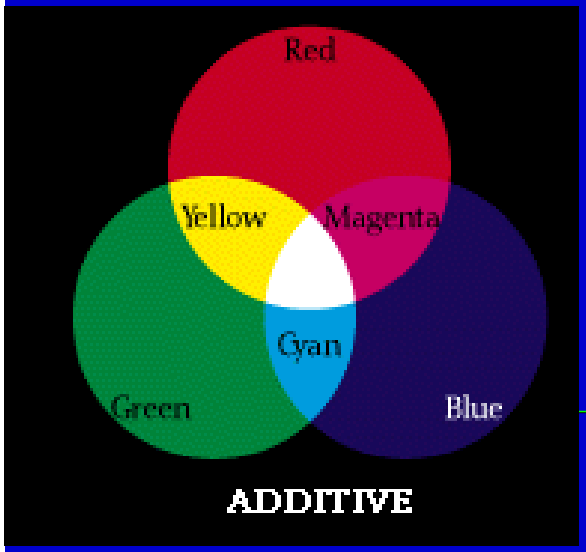


Mixing Colors

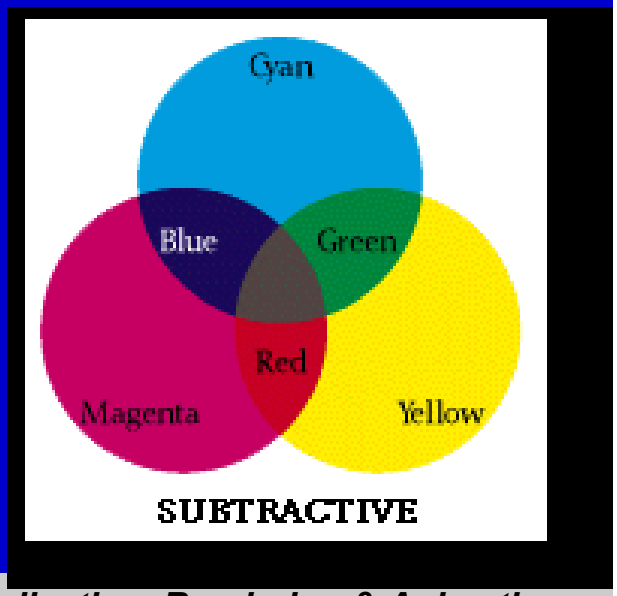


- **Additive color mixing** (pg.netgraphics.sk)
  - Combination of light rays
  - Examples: CRT, Video Beamer
- **Subtractive color mixing**
  - Combination of dye stuffs (pigments)
  - Examples: ink jet printer, -plotter

Goethe's triangle



http://www.cs.brown.edu/courses/cs092/VA10/HTML/start.html



Color Models

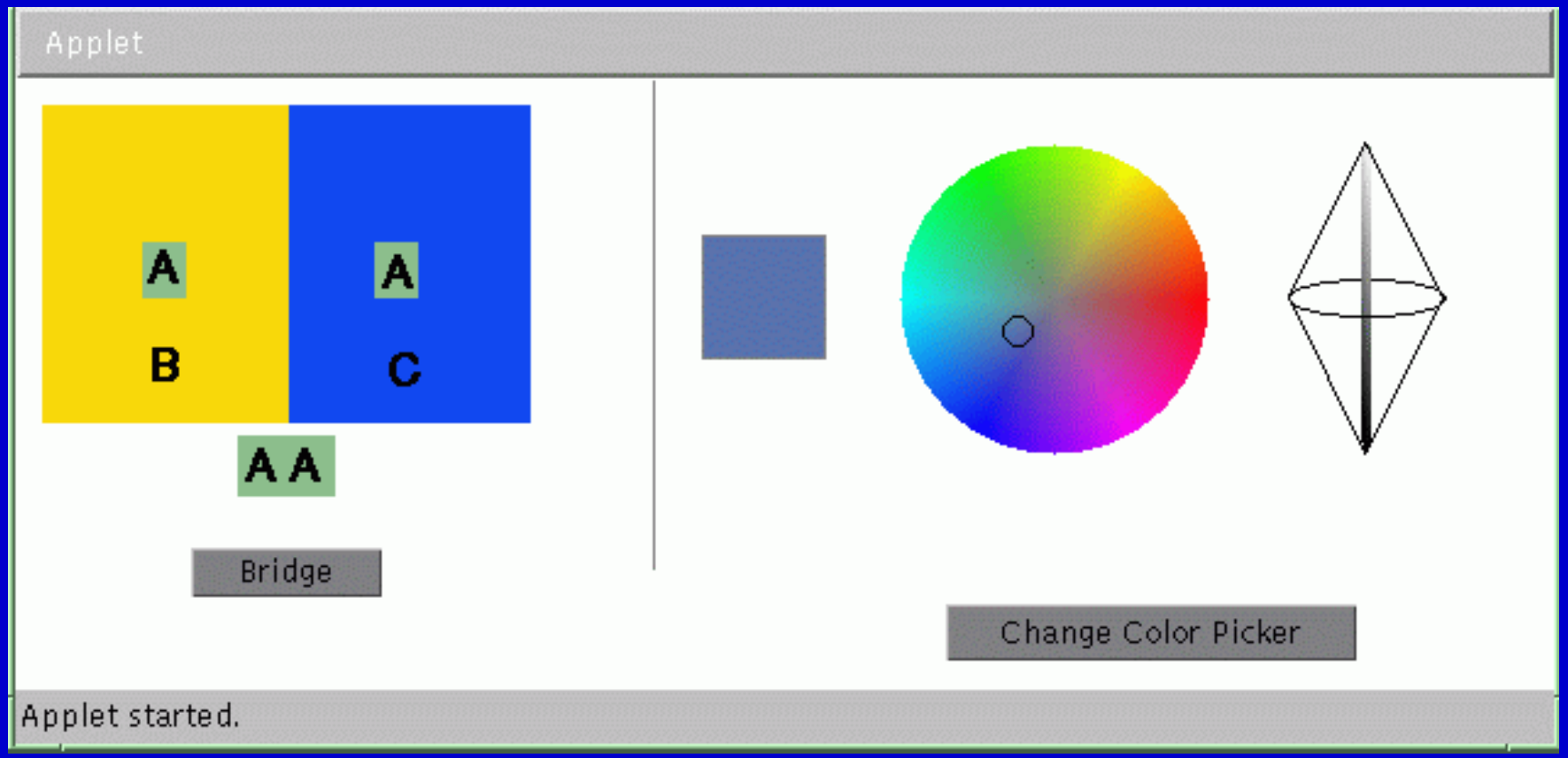
Comparison Criteria:

- Reference to human perception
- Representing of all colors
- Choosing colors of equal brightness
- Hardware-/user- oriented
- Intuitive or theoretic specification

Examples: CIE, CIE-LAB, CIE-LUV, RGB, CMY(K), HSV, HLS

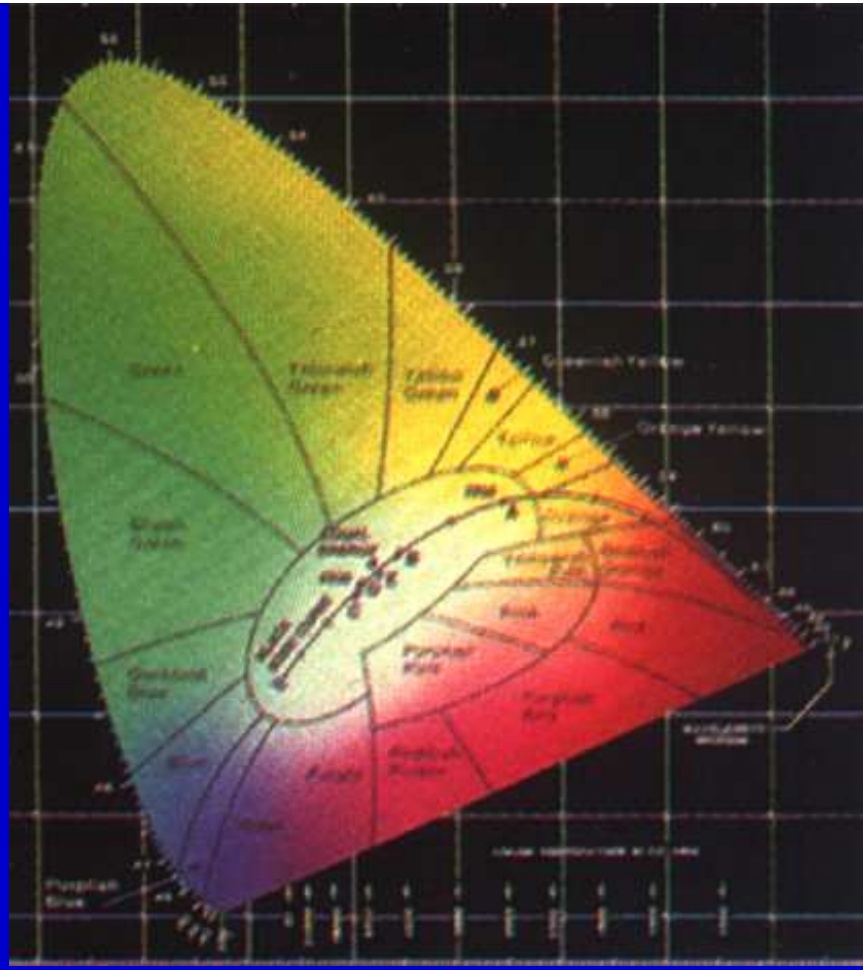
- http://www.cs.brown.edu/courses/cs092/VA10/HTML/start.html

One color appears as Two



CIE

- All visible colors defined (chromaticity)
- Colors of equal brightness
- Complement colors
- „ColorGamut“
- Different luminances with the same chromaticity
- Pure colors (curved part)



COMMISSION INTERNATIONALE DE L'ECLAIRAGE

INTERNATIONAL COMMISSION ON ILLUMINATION

INTERNATIONALE BELEUCHTUNGSKOMMISSION

CIE Central Bureau Kegelgasse 27 A-1030 Wien Austria

Standard Color Models

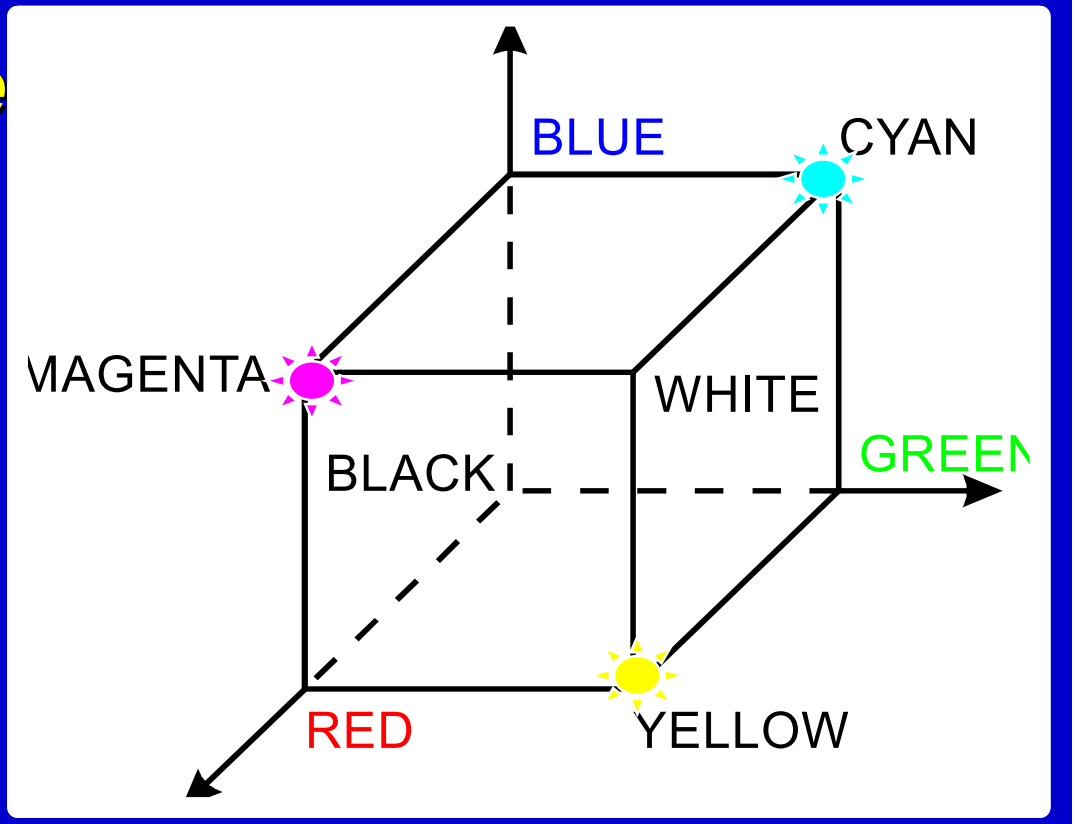
- Simple building of the model for the used hardware
- Based on additive or subtractive color mixing
- Output medium

Examples:

- RGB, CMY(K), YIQ, YCbCr - display based
- HSV, HLS - perception based

RGB – Color cube

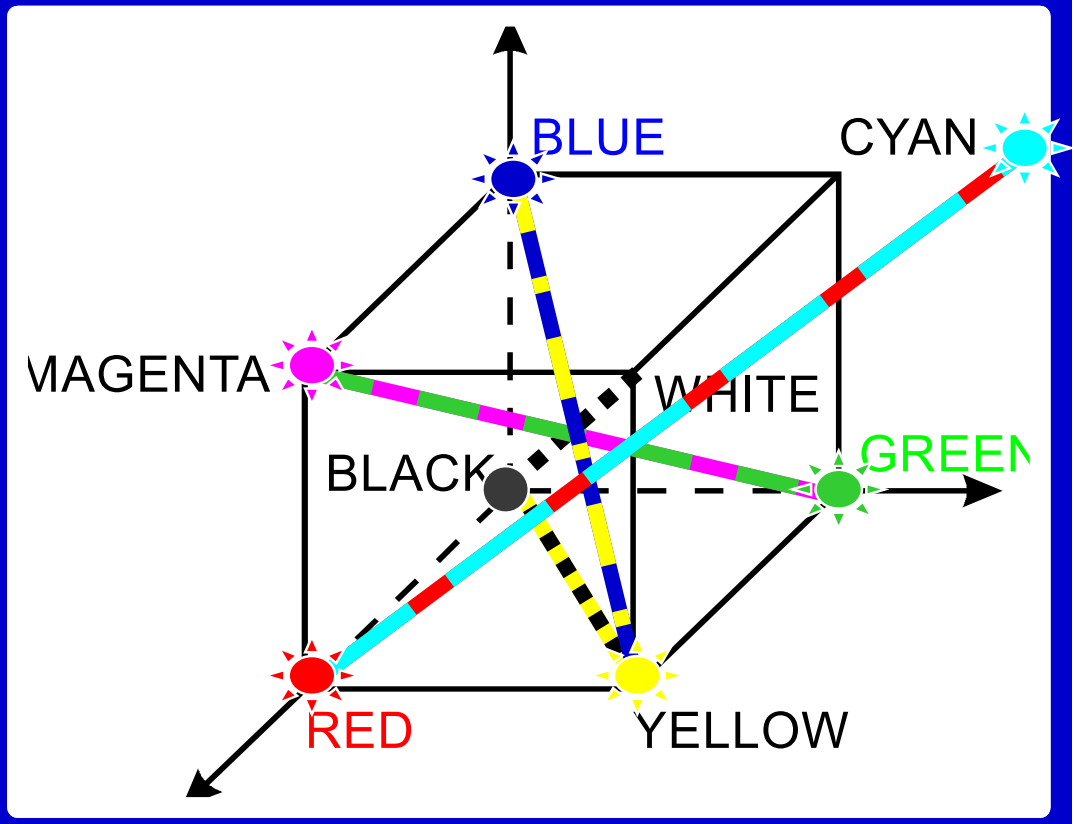
- System of 3 coordinates: red, green, blue
- Additive color
- mixing
- Usage: CRT



Complement color pairs

- R-C
- G-M
- B-Y
- K-Y
- W-K, black&white

- 5 EXTREMES

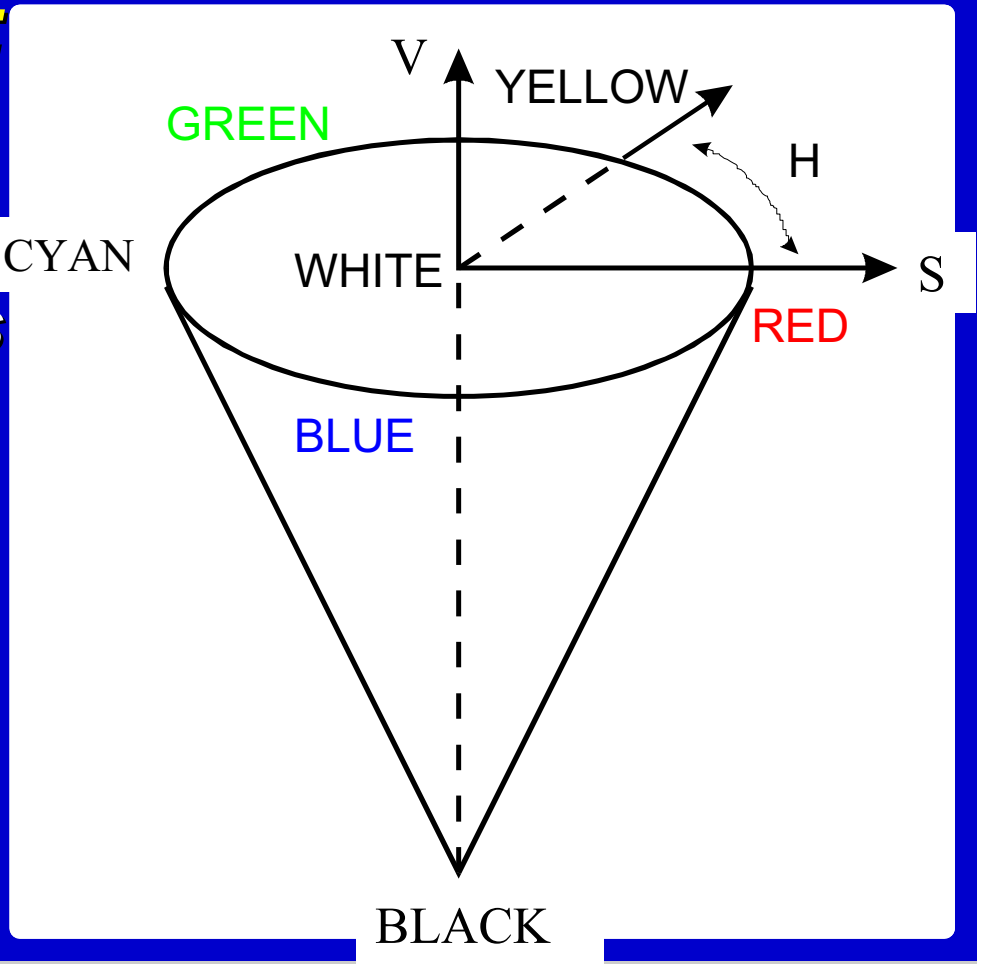


CMY/CMYK Color space

- 3D system: cyan, magenta, yellow
- Subtractive color mixing
- Usage: hard copy devices
- Conversion RGB -> CMY:
  R/G/B = 1 - C/M/Y
- CMYK: additional blacK component
  - K = min{C,M,Y}
  - C/M/Y = C/M/Y - K

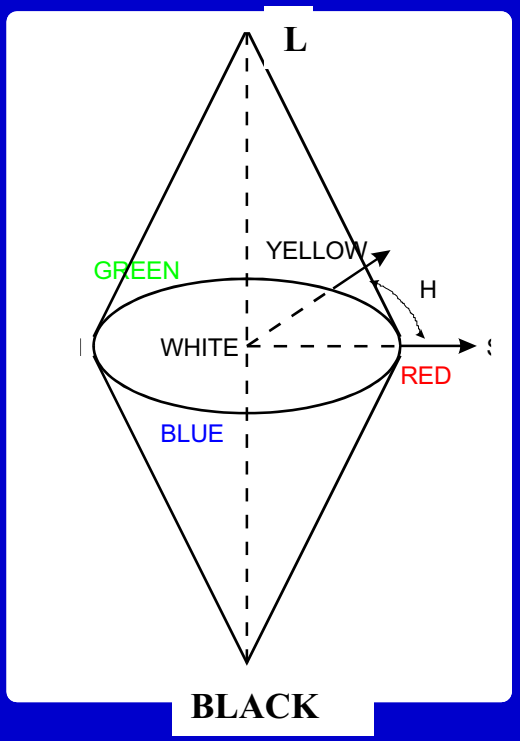
HSV

- **Cylindric coordinate system:**
  - Value: height
  - Saturation: distance to the axis
  - Hue: angle in the SV-plane
- **Representable colors create a cone**



Other Models

- HLS
  - Hue
  - Lightness
  - Saturation
  - Doubled pyramid



- NCS - Natural Color System
- CNS - Color Naming System
- YIQ - NTSC color system

Natural Color System (NCS)

Basic colors:

blue, red, yellow, green

Color mixtures:

<SS><CC>-<F1><%%><F2>

SS black contribution
CC color contribution
F1 color 1
%% mixing proportion
F2 color 2

Example:

4020-Y60R

Color Naming System (CNS) 1

Idea: A HLS relative, verbal color model with 7 basic colors:

red, green, blue, yellow, purple, orange, brown

Mixed colors:

Possible inbetweens for adjacent colors with the possible relationships:

- 25% / 75%: greenish-yellow
- 50% / 50%: green-yellow
- 75% / 25%: yellowish-green

Color Naming System (CNS) 2

Levels of brightness:

very dark, dark, medium, light, very light

additionally: black, white

(0, 0.16, 0.33, 0.5, 0.66, 0.83, 1.00)

Saturations:

grayish, moderate, strong, vivid

Example:

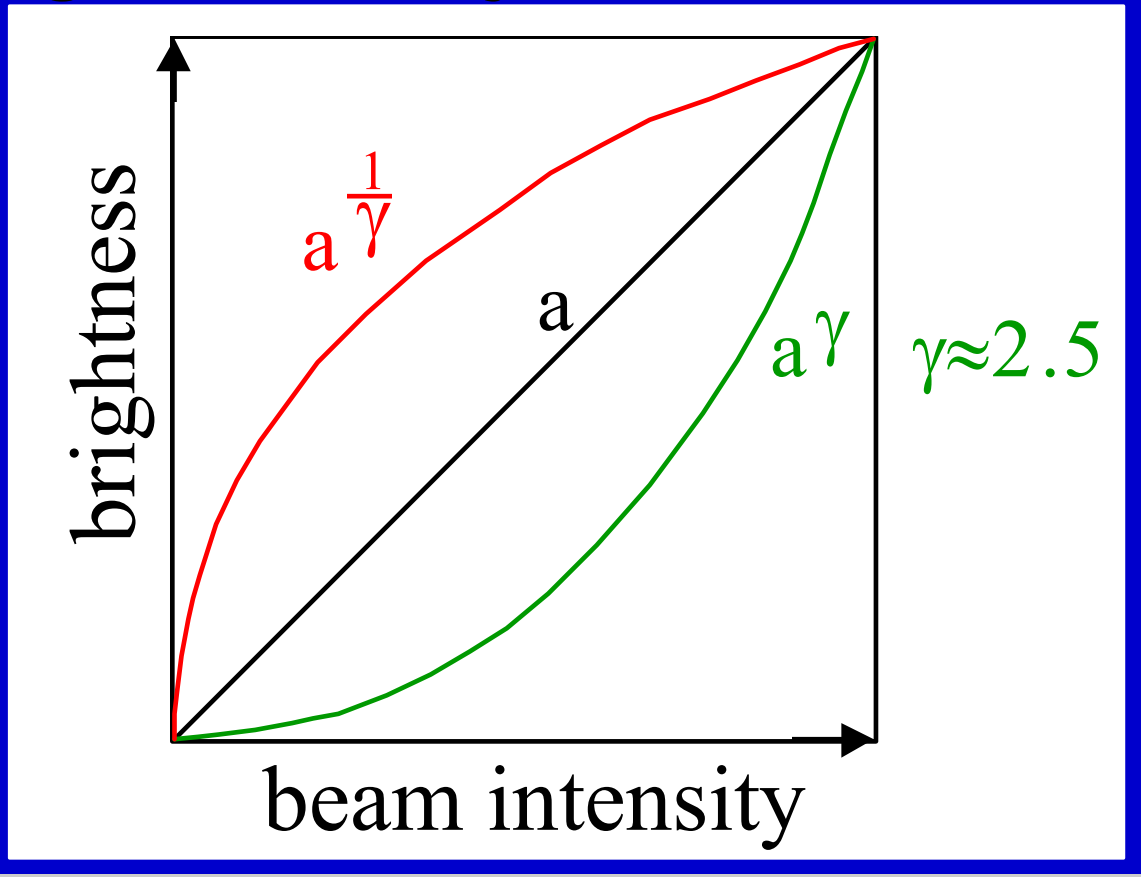
very dark vivid red

Gamma-correction

The connection between electron ray intensity and light intensity is not linear!

Formally:

$$I' = I^{1/\gamma}$$



Color Reproduction

- Representing of synthetic image at the display area or at the paper (Truecolor -> 2^n colors)
- Problem area 1 (raster displays):
  - geometric resolution satisfactory
  - radiometric resolution not fulfilled
- Problem area 2 (hardcopy methods):
  - geometric resolution more than satisfactory
  - radiometric resolution not fulfilled

Color Quantisation

Task specification: *Represent more colors using some appropriate colors with the help of color tables (LUT, look-up-table).*

Method:

- *Uniform quantisation*
- *Popularity method*
- *Median-cut method*
- *Octree-quantisation*

Uniform Quantising

TrueColor:

- 8 bit red (256 steps)
- 8 bit green (256 steps)
- 8 bit blue (256 steps)

16.7 million colors

256 color mode:

- 3 bit red (8 steps)
- 3 bit green (8 steps)
- 2 bit blue (4 steps)

8x8x4 = 256 colors

Popularity method

Idea: Find the K most frequent colors and use them in the LUT.

Realisation:

- Table with the frequencies
- K most frequent colors selection
- apply the closest color in the table

Pitfal:

minor details might be represented by strongly falsified colors

Median-cut Method

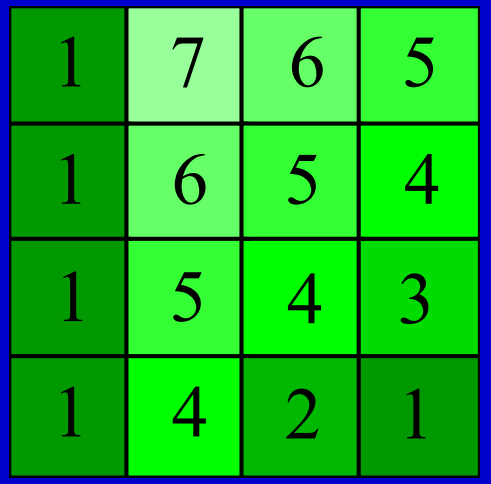
Idea: Each LUT entry should be represented by approximately equal number of pixels.

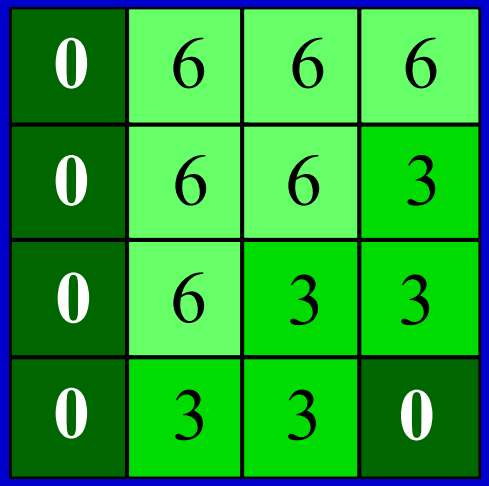
Realisation:

- Divide the color cube to obtain 2 parts with the required property
- Subdivide each kuboid with the most entries along the longest edge until K parts created
- each median is the representative

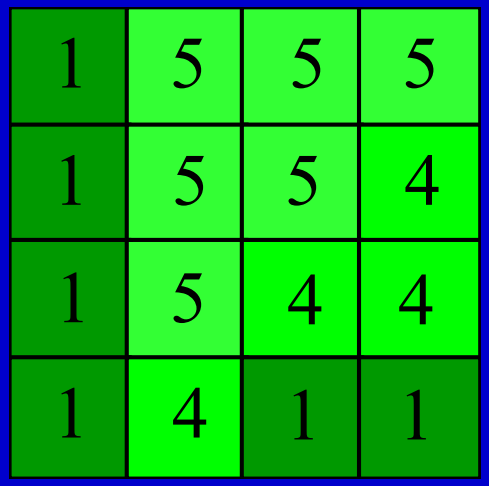
Comparison

Original

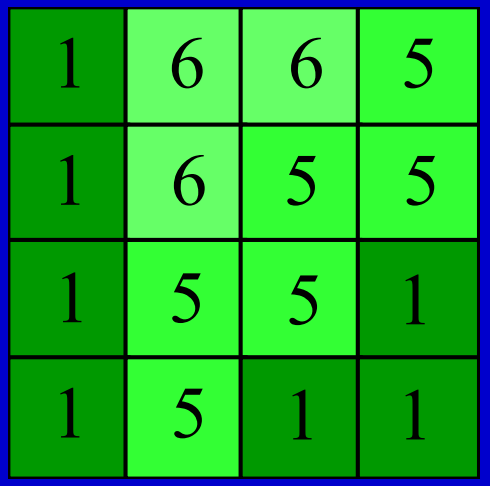




Uniform



Popularity



Median Cut

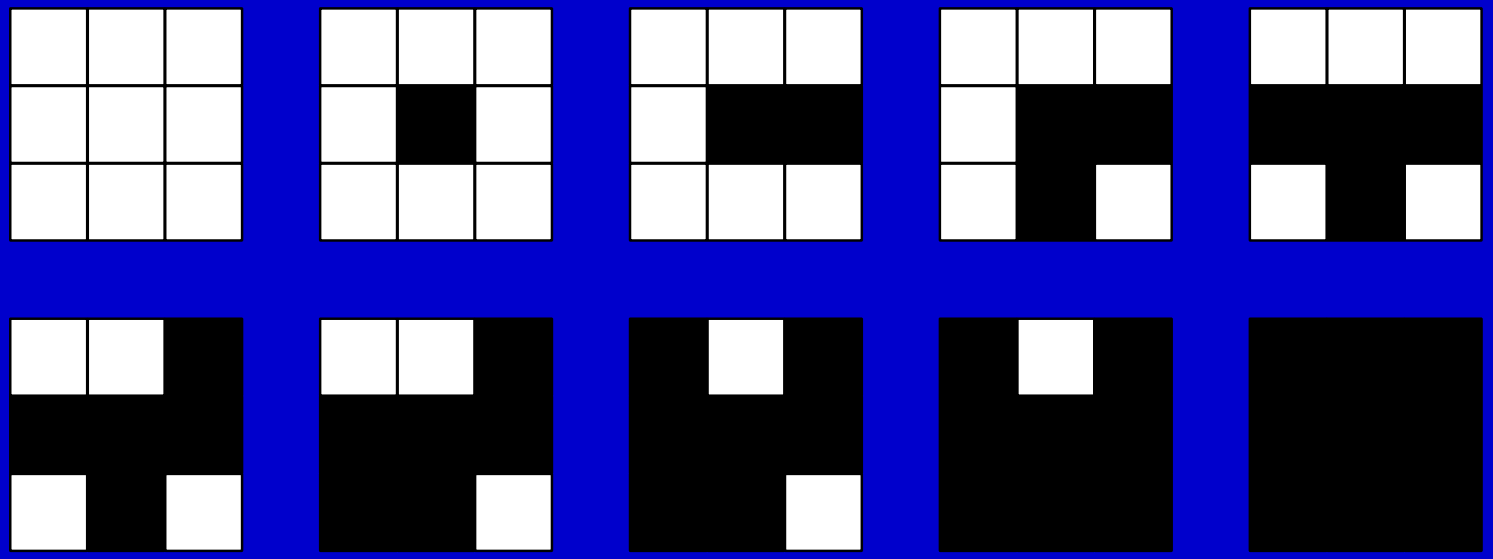
Dithering/Halftoning

- Simulate more color steps at expense of the geometric resolution
- Definition of dither matrices
  - grow the pattern outward from the center
  - N+1 level contains all points from level N
  - no structures produced
  - compact regions
- Alternative method: „Floyd-Steinberg“-error diffusion method

Dither matrices

Example for n=3:

- geometric resolution decreases by factor 3 (in each coordinate direction).
- radiometric resolution increases from 2 (monochromatic image) to 10.



5.
Light-Material Interaction

Illumination models

„Computer Graphics...

- *… can be formulated as a radiometrically „weighted“ counterpart of computational geometry…*
- *… rendering is done through the application of a simulation process to quantitative models of light and materials to predict/synthesize appearance“*
-
  - *D. Dobkin & S. Teller, 1999*

Computer Graphics...

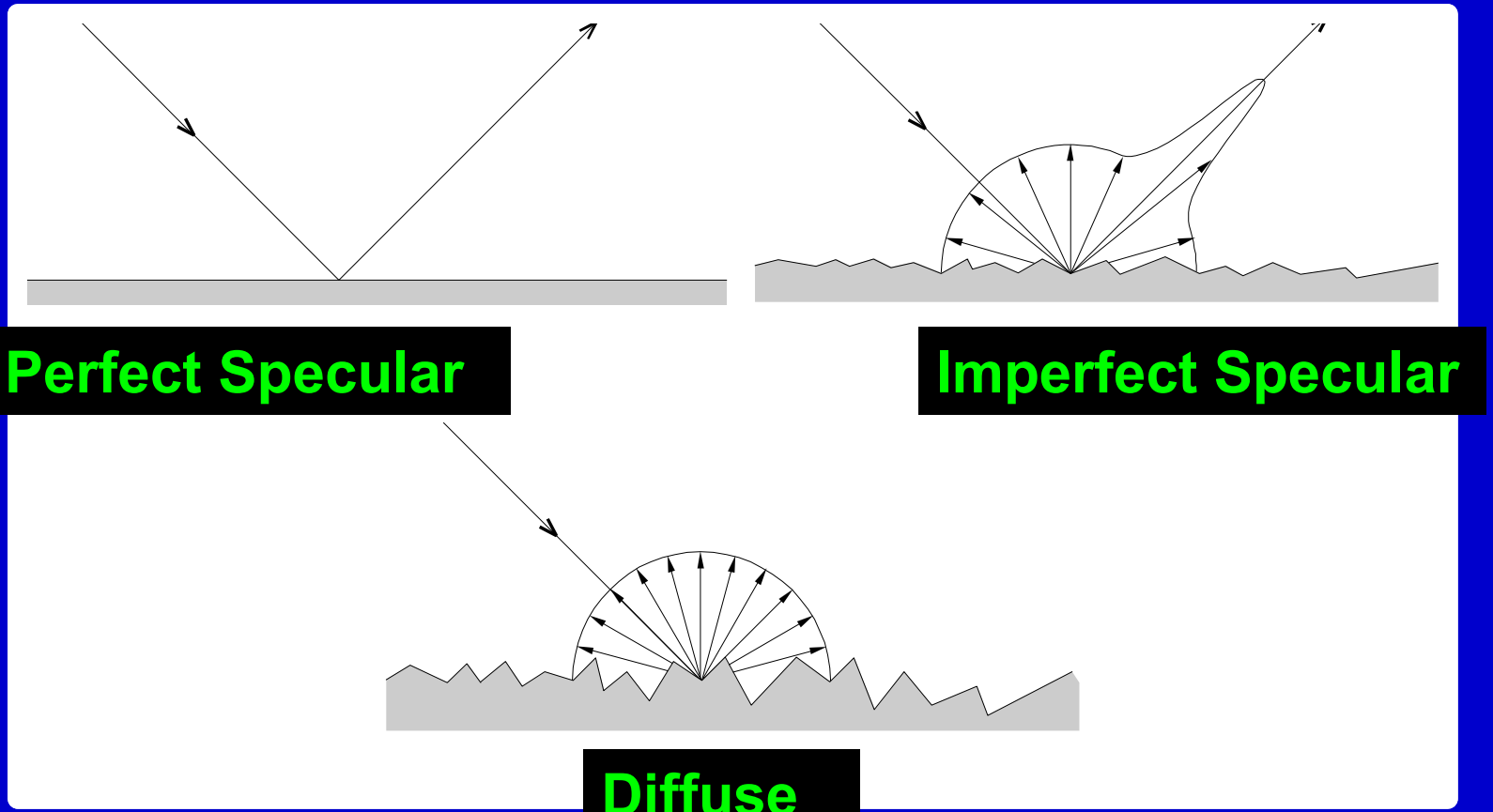
- ... must account geometry
- material properties: reflectance/color, refractive index, opacity, and (for light sources) emmisivity
- radiometry
- output for viewing: explicitly or implicitly psychophysics

- by D. Dobkin & S. Teller

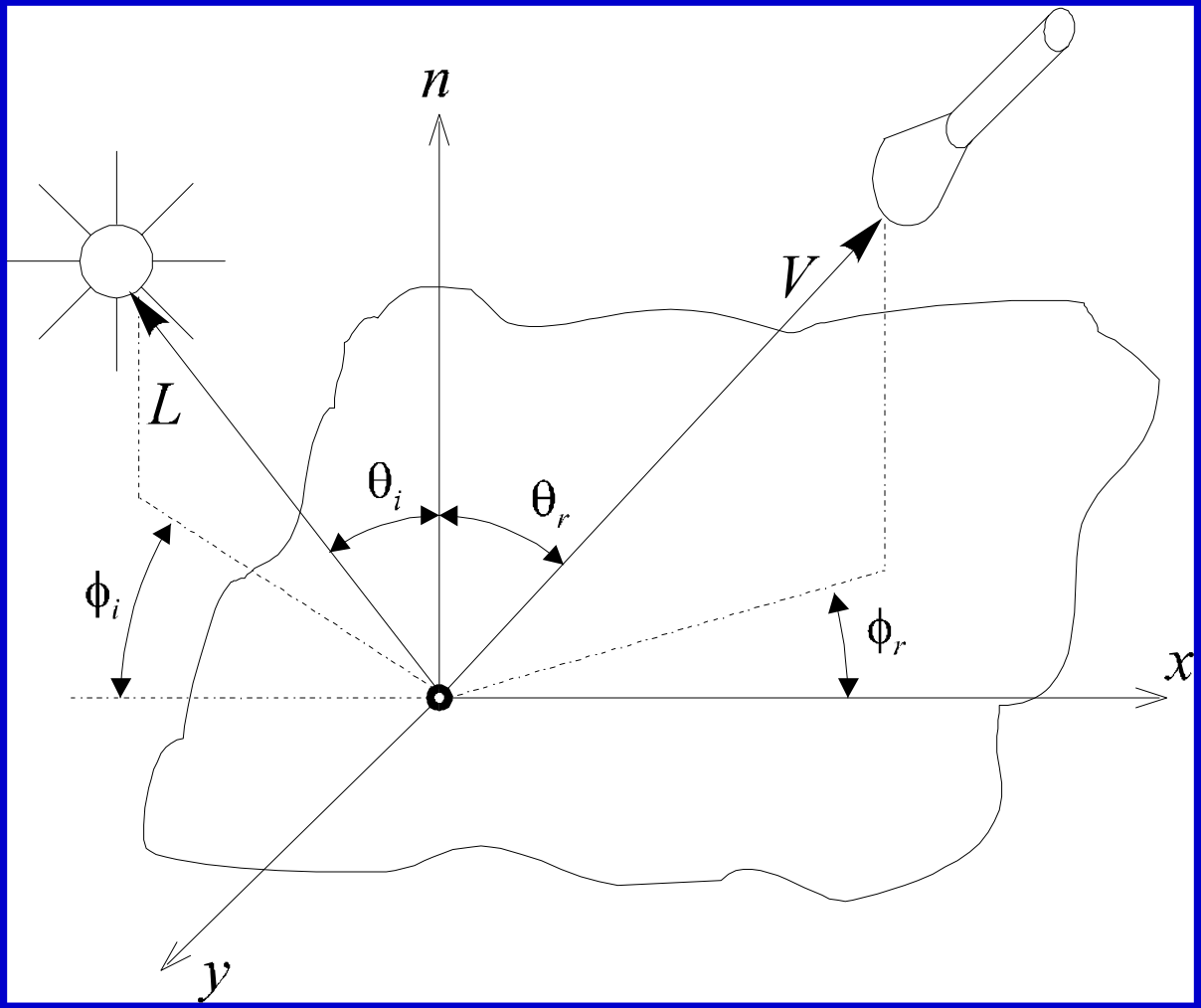
Illumination Models

- **Local Illumination Models (first order)**
  - Empiric Models (feasible)
  - Physical Models (possible, but expensive)
- **Globale Illumination Models (second order)**
  - Ray-Tracing (photons)
  - Radiosity (waves, „key is the light“)

Reflexion Properties



BRDF



Ambient Light

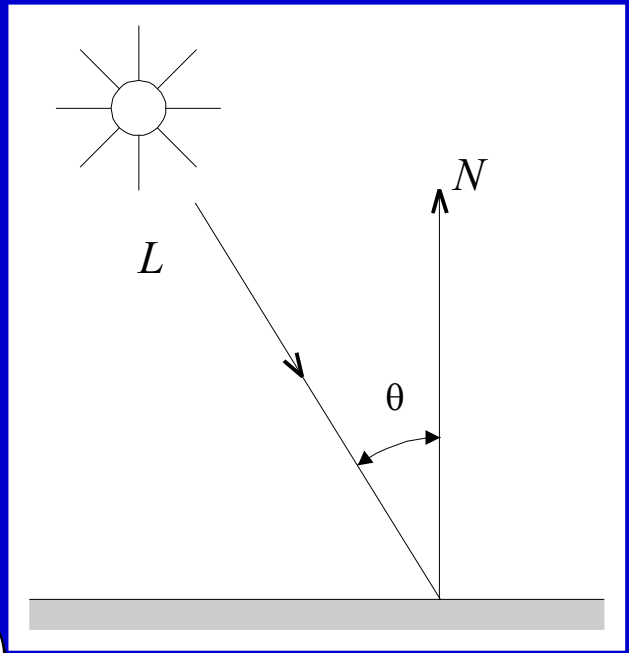
- Daylight (diffuse, undirected) lightsource
- Intensity in the given scene constant
- Multiple reflections on surfaces in the scene
- Trivial Illumination Model: $I = I_a k_a$

I_a intensity of ambient light
k_a ambient reflection coeficient

Lambertian Illumination Model

- Directional lightsource(s) added
- Diffuse reflection: independent from the camera position
- Illumination Model:

$$I = I_p\, k_d \cos\theta = I_p\, k_d\, (N \bullet L)$$



I_p Intensity of directional lightsource, point

k_d diffuse reflection coeficient

Intensity attenuation

- **Intensity contribution:**
  d_L lightsource distance

$$f_{att} = \frac{1}{d_L^2}$$

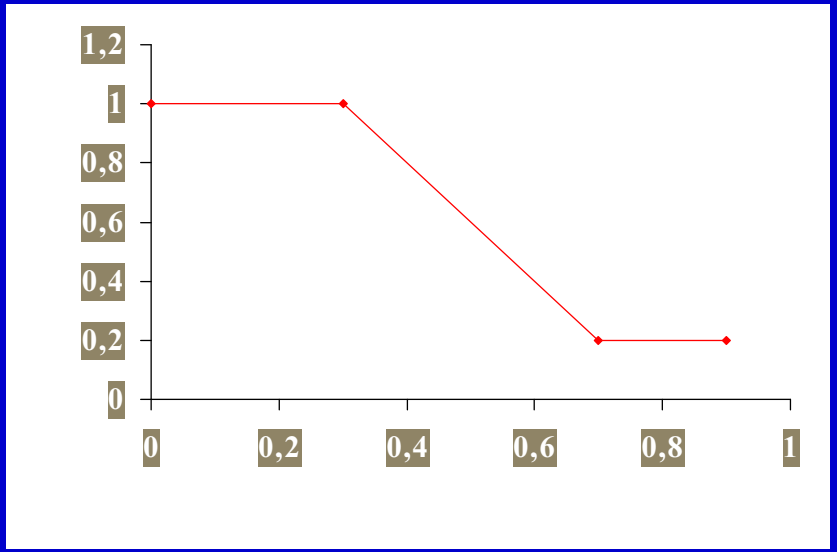
- **Alternative representation:**

$$f_{att} = \min\left(\frac{1}{c_1 + c_2 d_L + c_3 d_L^2}, 1\right)$$

- **Lighting model:** $I = I_a k_a + f_{att} I_p k_d (N \bullet L)$

Depth-cueing

- Distant objects appear darker (optionally „color-shift“, too)
- „Athmospheric perspective“



- Linear interpolation: $I' = s_0 I_f + (1- s_0) I_b$
- Scaling between „front/backplane“

Shaders, shading models

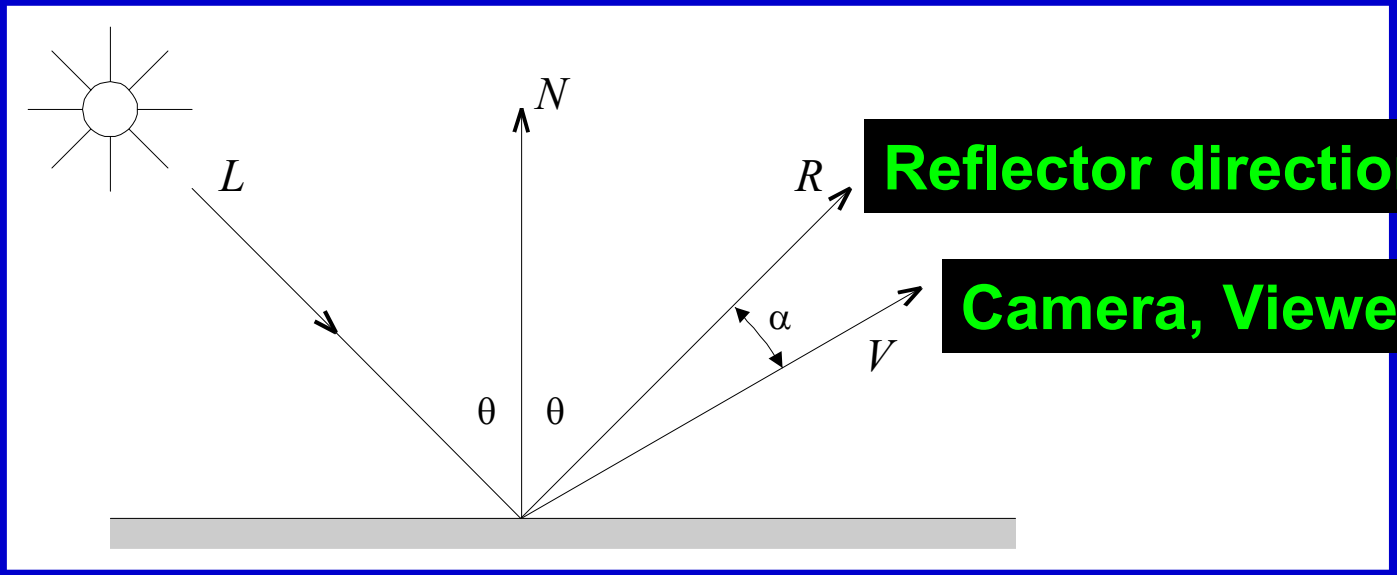
- Fill polygons after transformations and rasterization by color values
- Flat-Shading:
  - Lamberts illumination model
  - single color value for each polygon/triangle
  - advantage: very fast
  - drawbacks: Mach-bands, causing nonrealistic appearance
- Better ones: Gouraud-, Phong-Shading

Phong Illumination Model

- **Adding specular reflection (depends on camera position)**
- **New Illumination Model:**

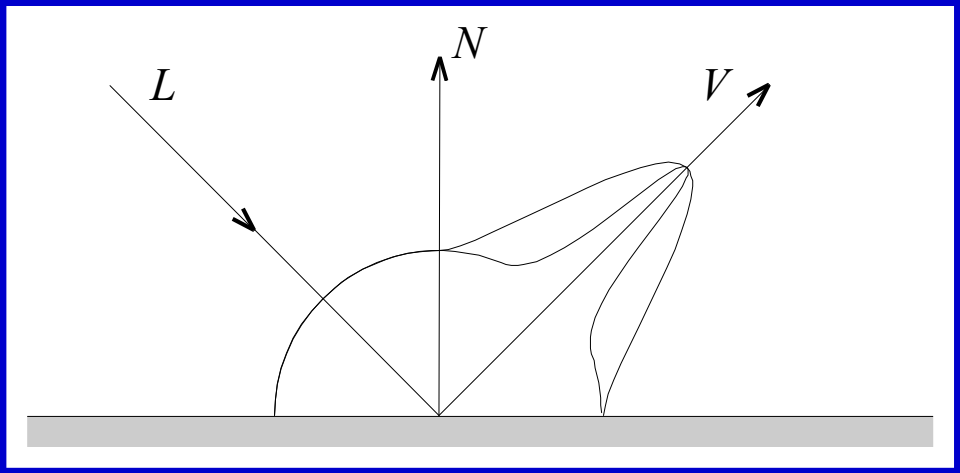
$$I = I_a k_a + f_{att} I_p (k_d \cos\theta + k_s \cos^n \alpha) =$$
$$I_a k_a + f_{att} I_p [k_d (N \bullet L) + k_s (R \bullet V)^n]$$

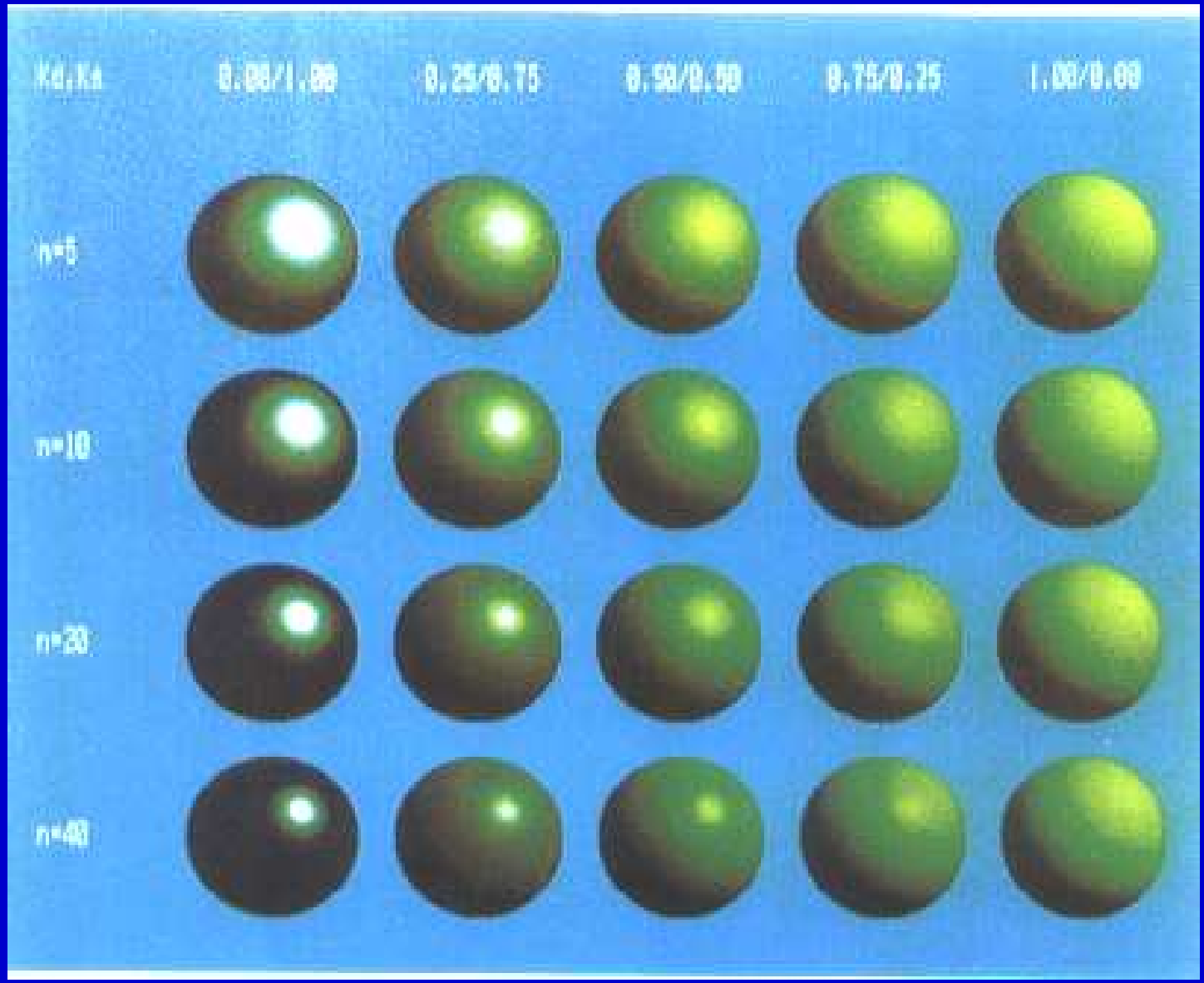
k_d **diffuse reflection coeficient**
n *(Spiegelneigung)*, **„shininess“ parameter**
R **Reflected photon direction vector**
V **Viewer/Camera direction vector**



Reflector direction

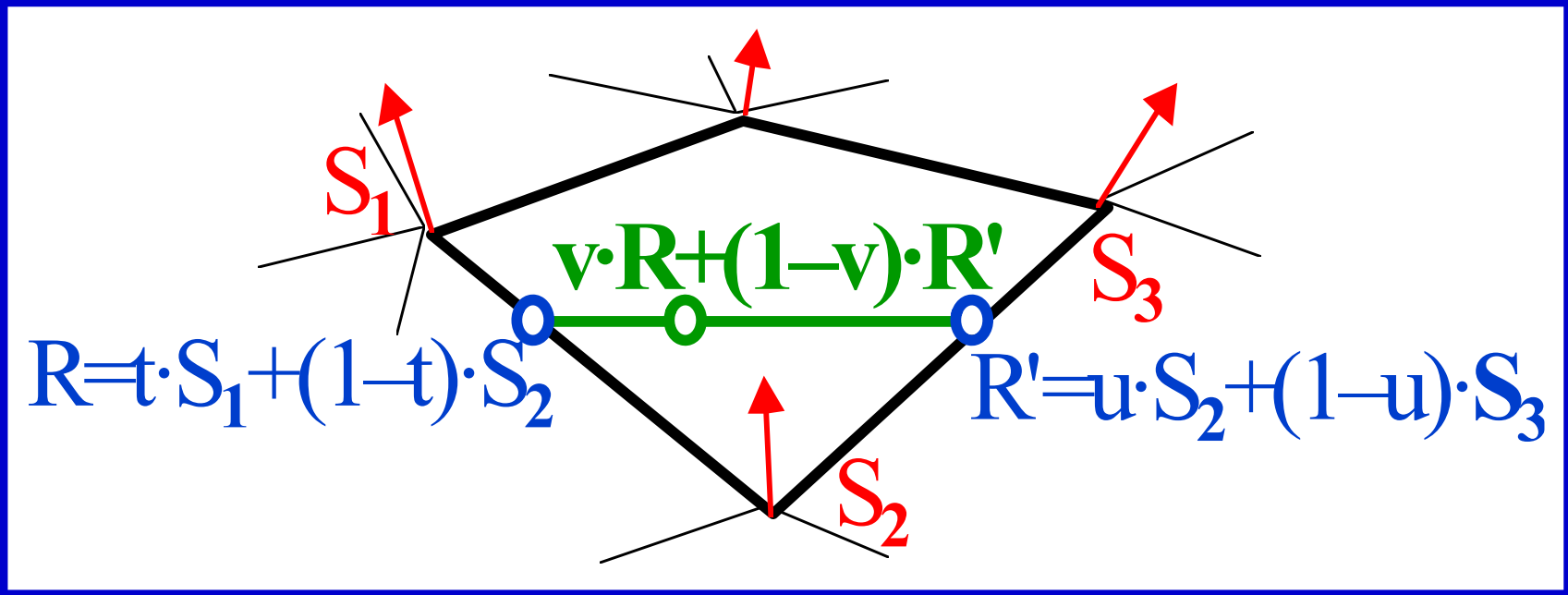
Camera, Viewer




Kd,Ks
0.00/1.00
0.25/0.75
0.50/0.50
0.75/0.25
1.00/0.00
n=5
n=10
n=20
n=40

Gouraud Shader

- Lambert / Phong Illumination Model
- Color values in the vertices
- Normal vectors given by:
  - Face normals
  - Face normals averaging
- Linear Interpolation of Color:
  - Along the edges
  - Along the „scan-lines“
- Drawback: quality
- Advantage: speed

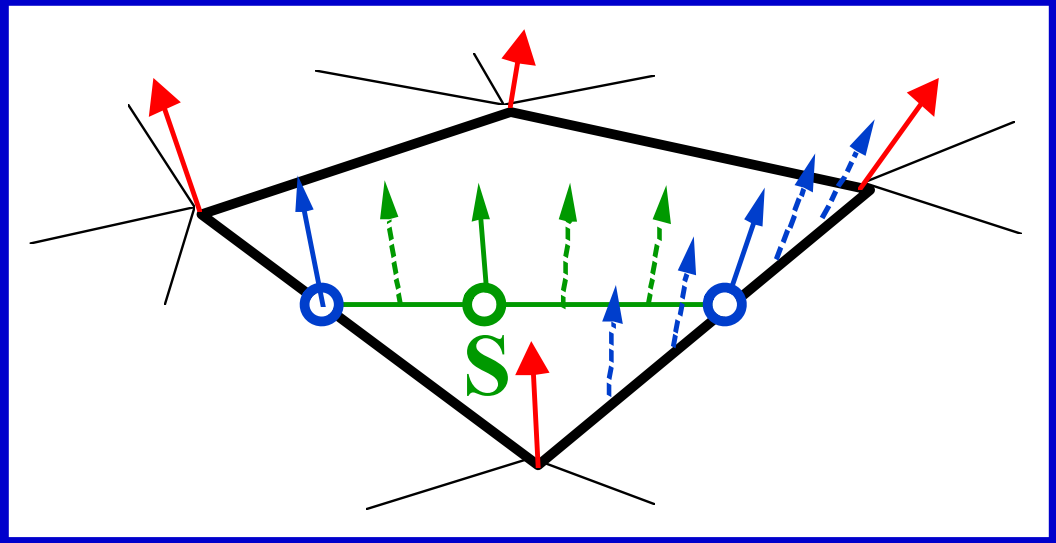


1. *find normal vectors at corners and calculate shading (intensities) there*
2. *interpolate intensities along the edges linearly*
3. *interpolate intensities along scanlines linearly*

Phong Shader, phong

- Normal vectors like Gouraud
- Linear interpolation of normal vectors instead of color value
- Color computation per pixel
- Pro: specular highlights in the given polygon rendered correctly
- Con: computationally expensive

phong



1. normal vectors at vertices
2. interpolate normal vectors along the edges
3. interpolate normal vectors along scanlines and calculate shading (intensities) for every pixel

Interpolation Problems

- Polygon-Silhouette
- Interpolations artefacts from the given „scanline“ - direction
  - Orientation dependent
  - Perspective dependent
- Not representative vertex no
- Hint: refine the triangulation (more complex model)

Rendering Polygonal Scene

- 1. Extract polygons from the database
- 2. Transform to WC and VRC
- 3. Backface culling and visibility
- 4. Clip against the visible volume
- 5. Projection of clipped polygons
- 6. Shading by incremental shader:
  - 1. Rasterize,
  - 2. Depth and visibility, (z-buffer)
  - 3. Shading (constant, Gouraud, Phong…)

Local Illumination Summary

- Empirical shading models
  - constant, Gouraud, Phong...
- Ambient, diffuse and specular reflection
- Light rays only
- Polygonal scenes
- Rendering summary (polygonal case)
- More: transparency, bumpy surfaces, textures, global illumination, animation...

Local Illumination Online

- ***Applet by Patrick Min at***

  http://www.cs.princeton.edu/~min/cs426/classes/light.html

- *http://www.siggraph.org/education/materials/HyperGraph/illumin/illum0.htm*
- *http://www.siggraph.org/education/materials/HyperGraph/illumin/vrml/pellucid.html*

Definition of Light Sources

- Point light source
- Multiple point sources… area
- 4 abstract lightsources - ambient, directional, point, flood
- intensity/fog = I/(a*d*d…*d + b)
- flood: powers of cosine (Phong)

Lighting Optimization

(polygonal model, light sources)

In: Real-Time Rendering [Moeller-Haines, 1999, pp. 248n]

www.realtimerendering.com, 2nd edition in the year 2002

– 1. Is lighting needed?

- Sometimes, higher constant for ambient light helps
- Alternatives: texturing,
- texturing with colors at the vertices,
- colors at the vertices

- Example: background polygon

Lighting Optimization 2

- *Light speed constant, unlike lighting speed :-)*

- *Directional light sources, point lights, spot lights*
- *Directional light source gives normalised vector*
- *Spot light: in-cone test for vertices, exponential fall-off from the center of the light cone*
- *Gray-scale light source is faster than a colored light source*
- *The number of light sources*

  (some graphics accelerators are optimized for lighting calculations with only one source)

Lighting Optimization 3

- **Non-local viewer trick (camera at infinity)**

- Simplification of specular highlight computation and environment mapping calculations
- Normally, the vector from the eye to the vertex being illuminated is computed and normalised
- This is replaced by (0,0,-1) or (0,0,1) - if the viewer is looking along the positive z-axis
- With directional lights this means that the halfway vector h is computed only once for each scene
- The error (slight shift in the locations of the specular highlights) of this simplification is not usually not detectable

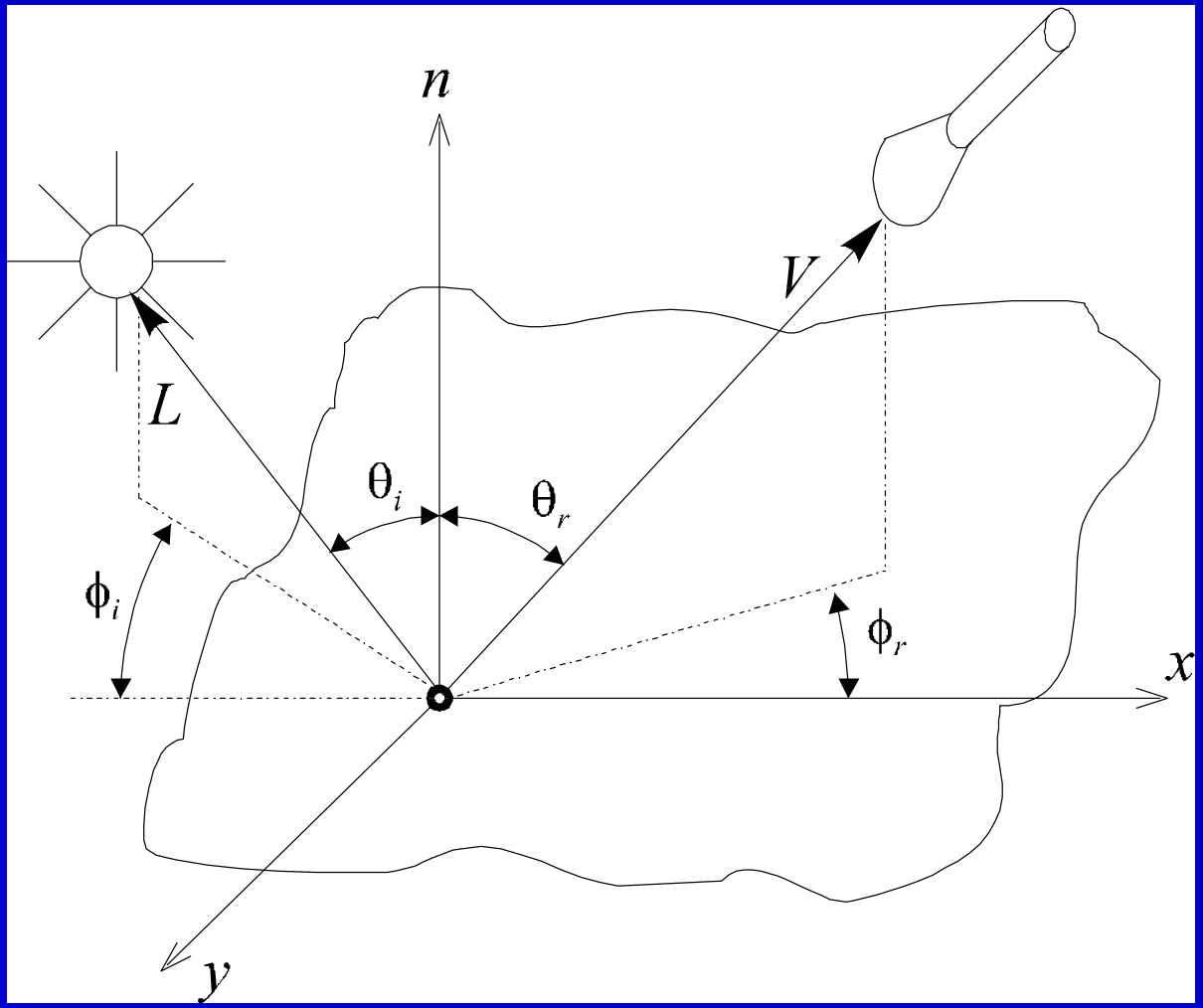
Lighting Optimization 4+

- 4. If the model will be not scaled during rendering all normals should be normalised as a preprocess
- 5. Flat shading is faster than the Gouraud one
- 6. If a non needed lighting is computed for both sides of a polygon then turn this feature off
- 7. If possible, set the material specular component to (0,0,0) to supress the expensive higlight calculations

Lighting Optimization 8

8. If the light sources are static with respect to geometry, and the polygonal data is attached to a material without specular parameters, then the diffuse and ambient lighting can be precomputed and stored as colors at the vertices. Simple shadows can be captured by direct visibility tests between lights and vertices. Even the radiosity can also be precomputed and stored as colors at the vertices or as light maps. Specular (view dependent) contributions can be added in (no highlights in areas which are in shadow)…

BRDF



Physically based models

- **Modeling of BRDF:**

$$\rho_{brdf}\left(\theta_i, \phi_i, \theta_r, \phi_r\right) = \frac{I_r}{E_i}$$

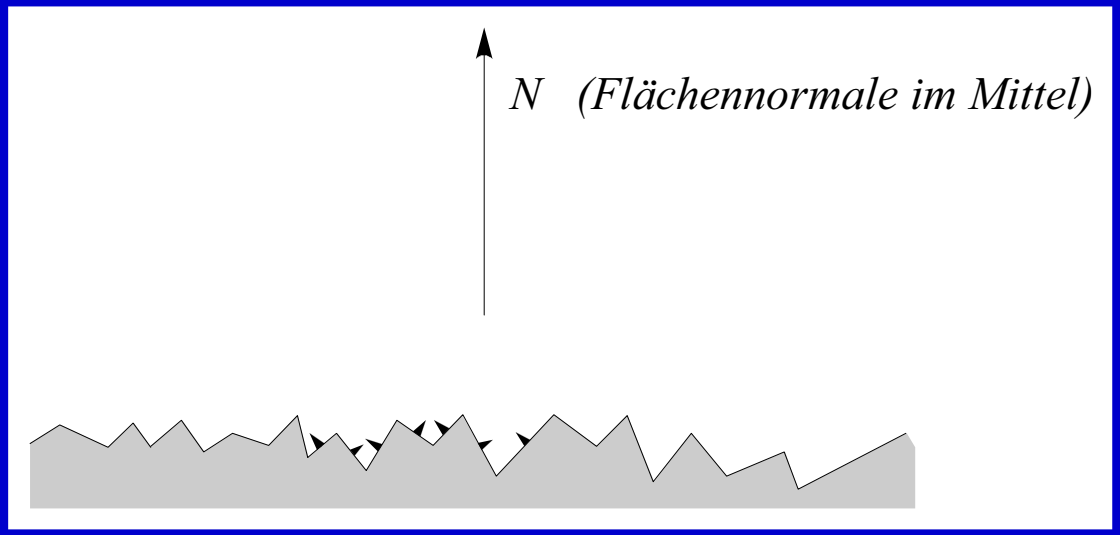
- **Classic Wave theory**
  - Maxwell's Equations
  - For reflective surfaces

  - Rayleigh-criterion:

$$h < \frac{\lambda}{8 \sin \beta}$$

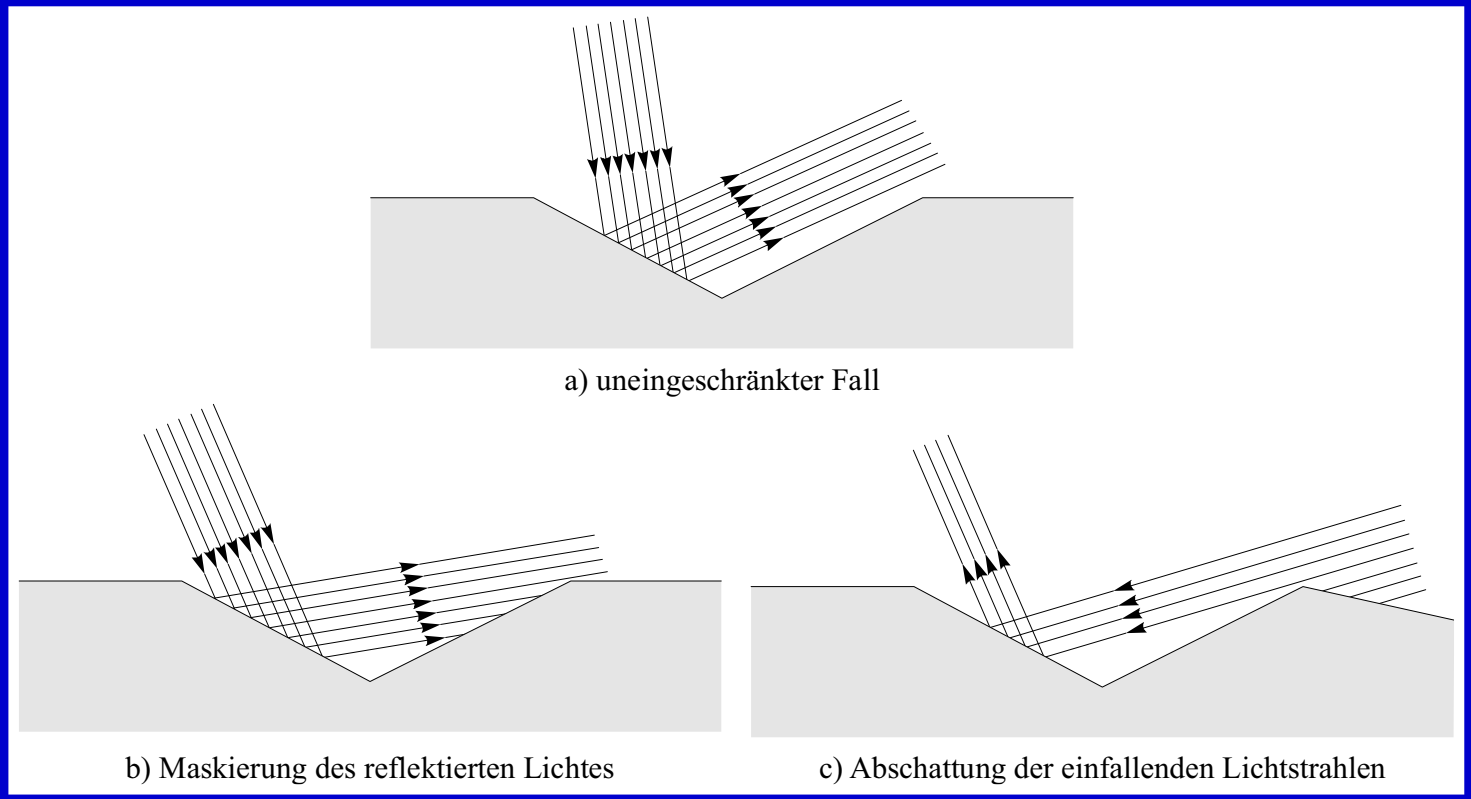
Micro-facets model

- **Idea:** Distribute many small mirrors at the object surface
- Simulation of diffuse ... oriented-diffuse material properties
- $\rho_{brdf} = DG\rho_s$



Geometric Attenuation

$$G = \frac{1}{\sqrt{\cos\theta_i \cos\theta_r}}$$



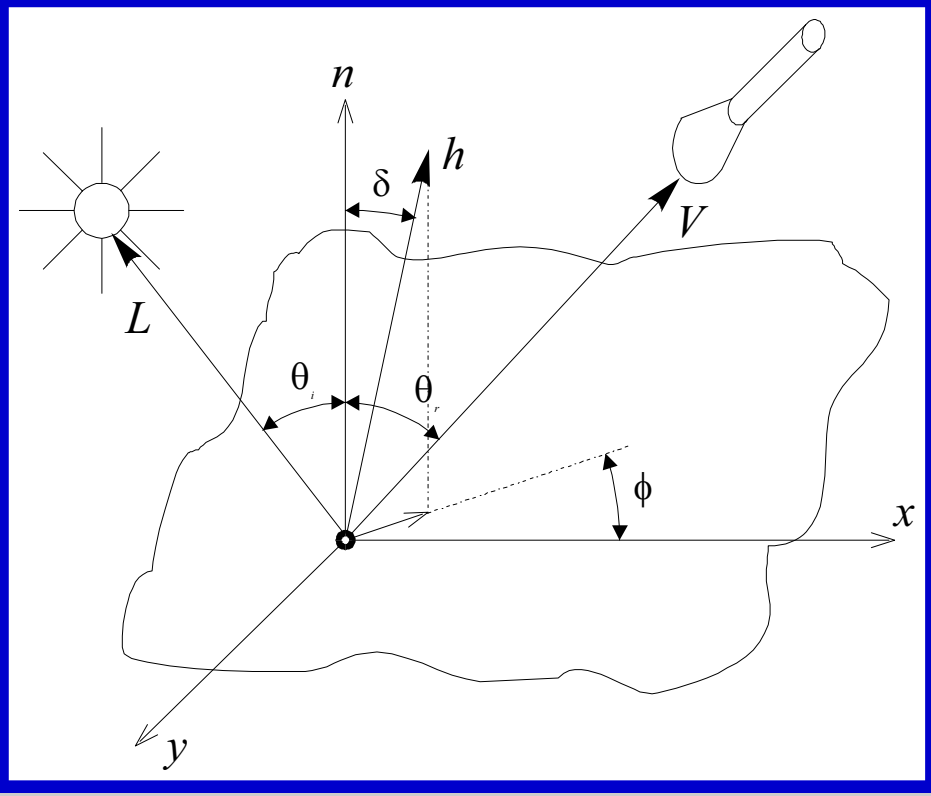
a) uneingeschränkter Fall

b) Maskierung des reflektierten Lichtes

c) Abschattung der einfallenden Lichtstrahlen

Gauss Distribution

$$D = \frac{1}{4\pi\alpha^2} \exp\left(-\frac{\tan^2 \delta}{\alpha^2} \right)$$



Closed Form

- $$\rho_{bd,iso} = \frac{\rho_d}{\pi} + \rho_s \frac{1}{\sqrt{\cos\theta_i \cos\theta_r}} \frac{\exp\left(-\tan^2 \delta / \alpha^2\right)}{4\pi\alpha^2}$$

  δ angle between N and H

- **For anisotropic materials:**
  - additional model parameter: $\alpha \rightarrow \alpha_x, \alpha_y$
  - additional variable: ϕ

Illumination Models

- **Local Illumination Models (first order)**
  - Empiric Models (feasible)
  - Physical Models (possible, but expensive)
- **Global Illumination Models (second order)**
  - Ray-Tracing (photons)
  - Radiosity (waves, „key is the light“)

Image-Based Rendering

Heinz Mayer, Andrej Ferko

Institut für Maschinelles Sehen und Darstellen

Movie from Stills

Movie synthesis from still pictures
Layer-structured 3D model
rendering with view-dependent
texture mapping

http://www-sop.inria.fr/robotvis/
personnel/fabad/PhD/index.html

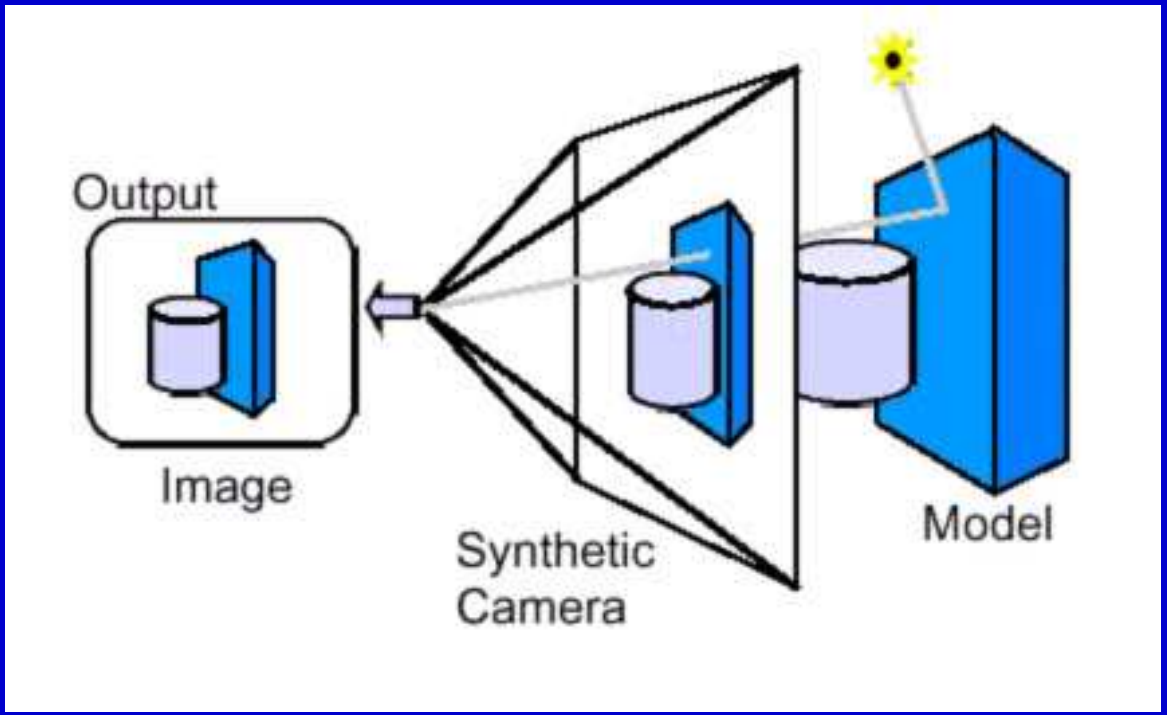
CREDITS

Frederic Abad (the PhD-student who made the film)
Inria (his institute)
Realviz (a company who sponsored him)
MainStreet (another company who provided the reference images, I think this is a film production company)

Tracking Technologies

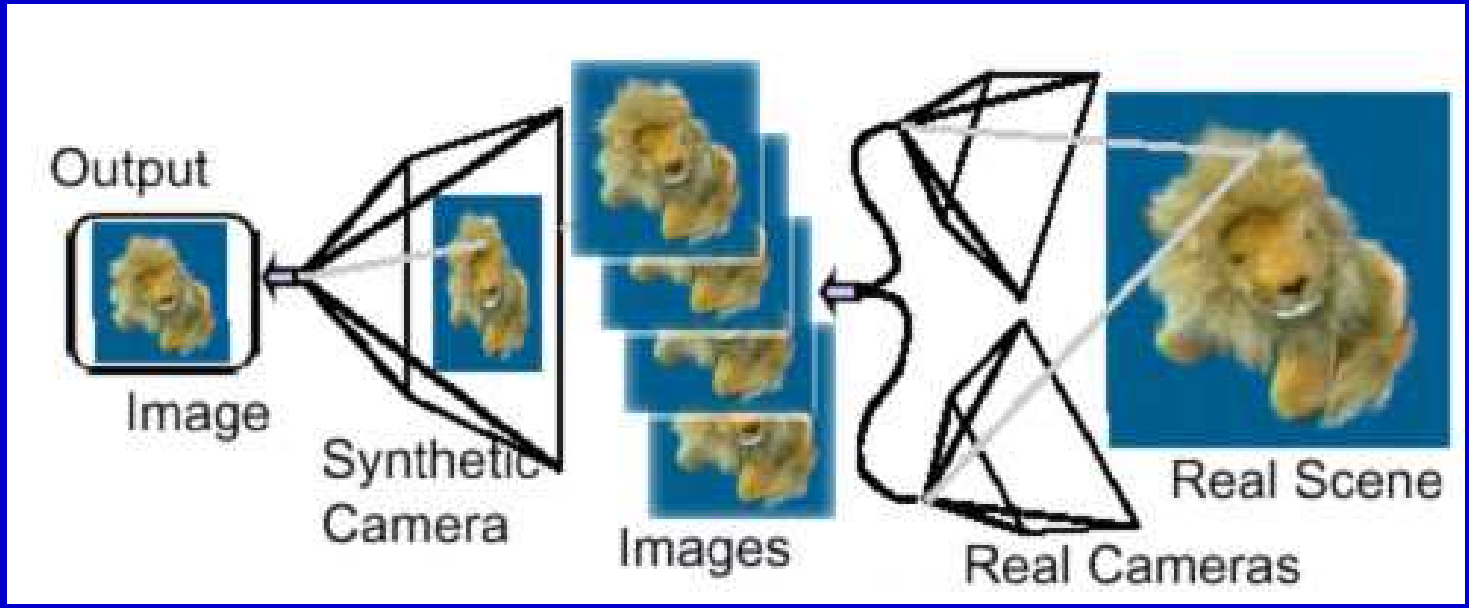
- Mainly position & orientation
- Movement freedom
- Important evaluation criteria
  - measuring rate
  - latency time
  - precision
  - resolution
  - operation extent (& appropriate precision)

Model-Based-Rendering



The real scene built with geometric objects

Image-Based-Rendering



Varied views on real scene combined to the new one

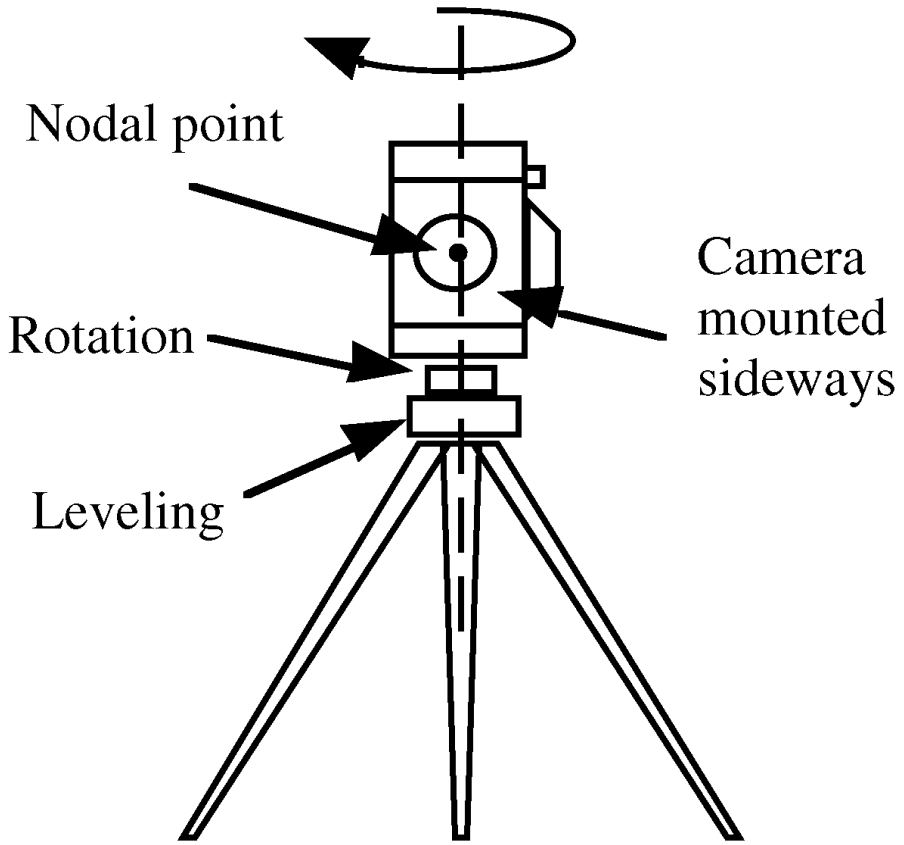
Comparison

Model-Based-Rendering	Image-Based-Rendering
Based on 3D model	Based on photos/stills
Expenditure strogly depends on scene complexity	Expenditure independent from scene complexity
Requires expensive SW for realistic results	Realism depends on input-data only
Special HW necessary	Processor suffices
Conventional Rendering-Pipeline	Pixel projection and Pixel interpolation


View Interpolation

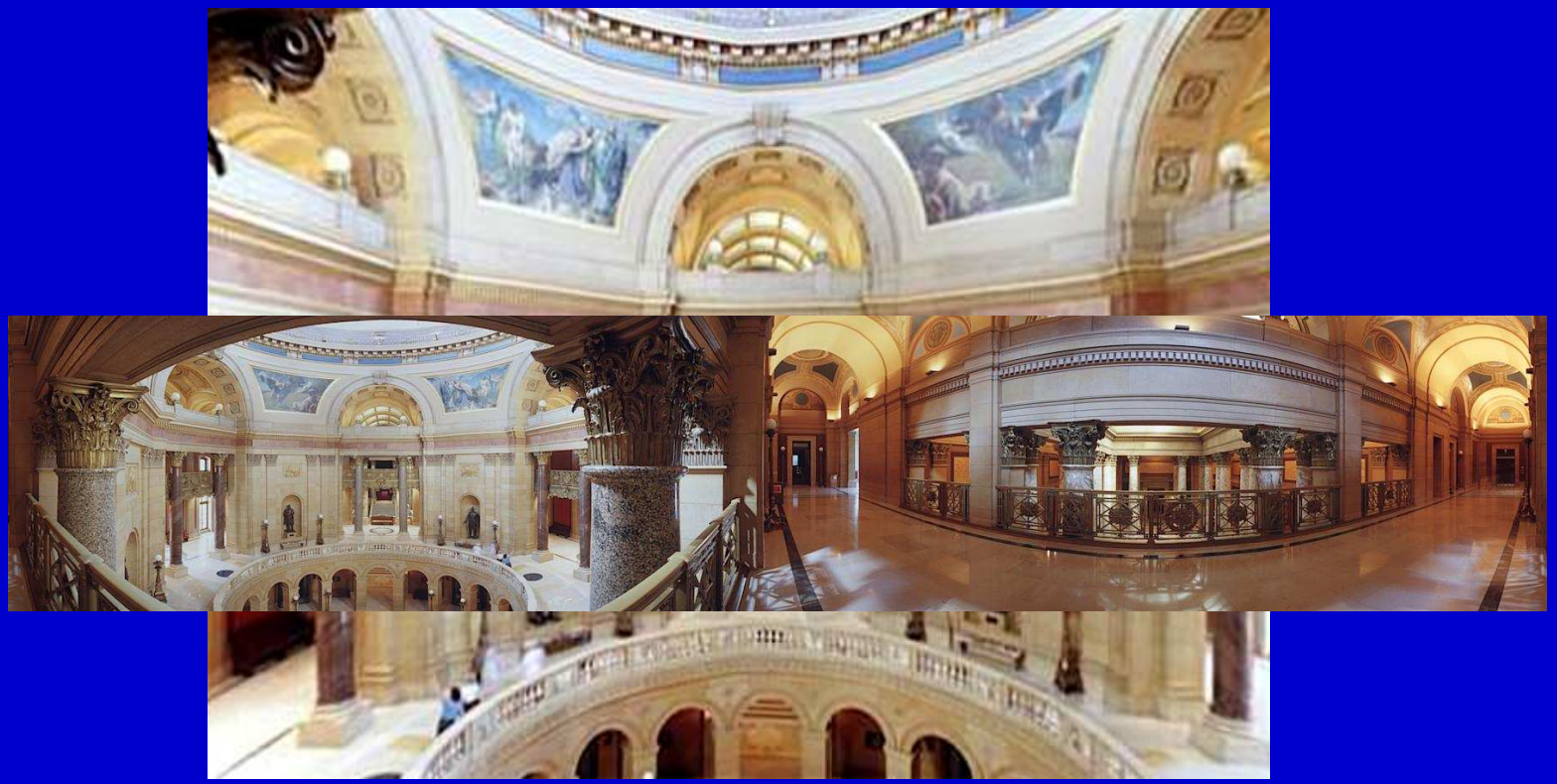
Recording Systems

- *Rotating Platform*
  - *CCD-lines*
  - *CCD-camera*
  - *Stereocameras pair*



- *Panoramas from*
- *exponed positions*

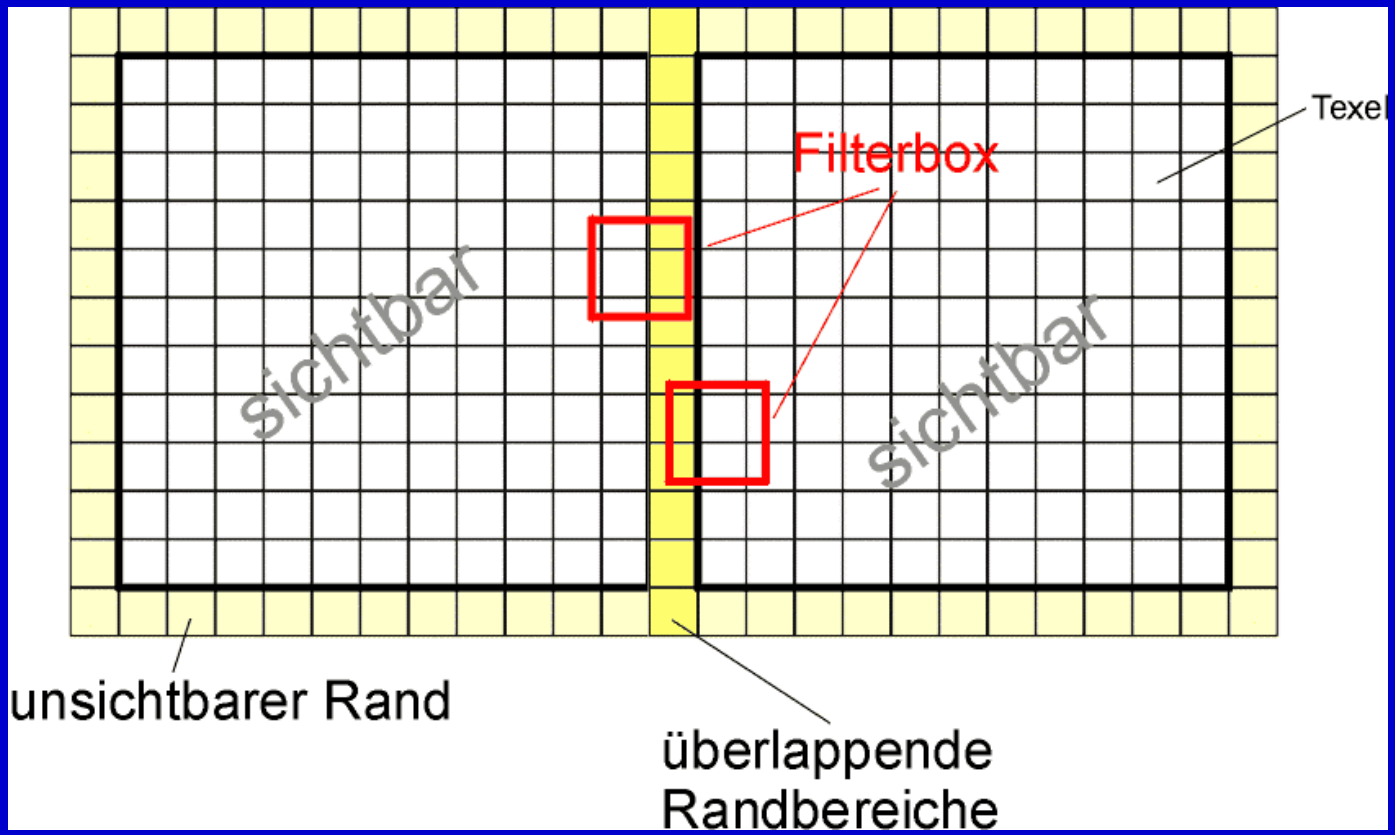
From Panoramic Images to Image Synthesis



Functionality

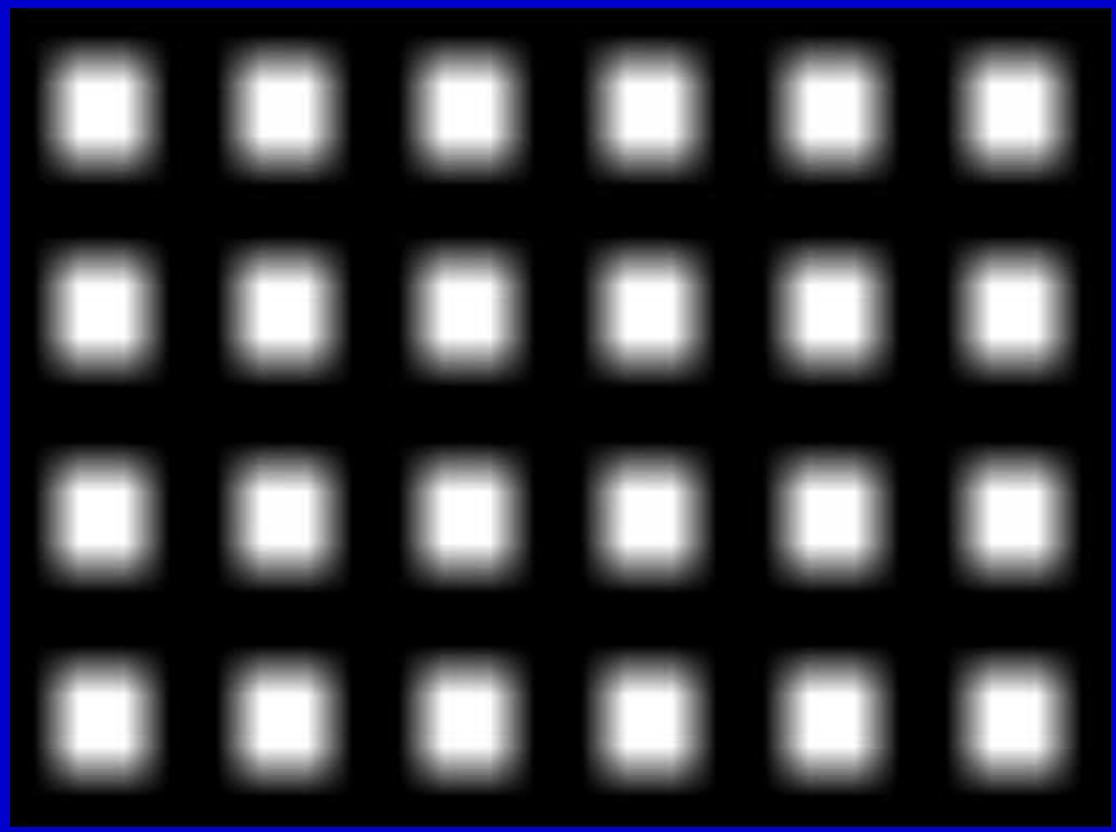
- Panorama image equalize
- Inner side of a cylinder panorama texturing
- Look up from the central axis
- Camera rotation: turn and declination
- Zoom

Partial Images Overlap

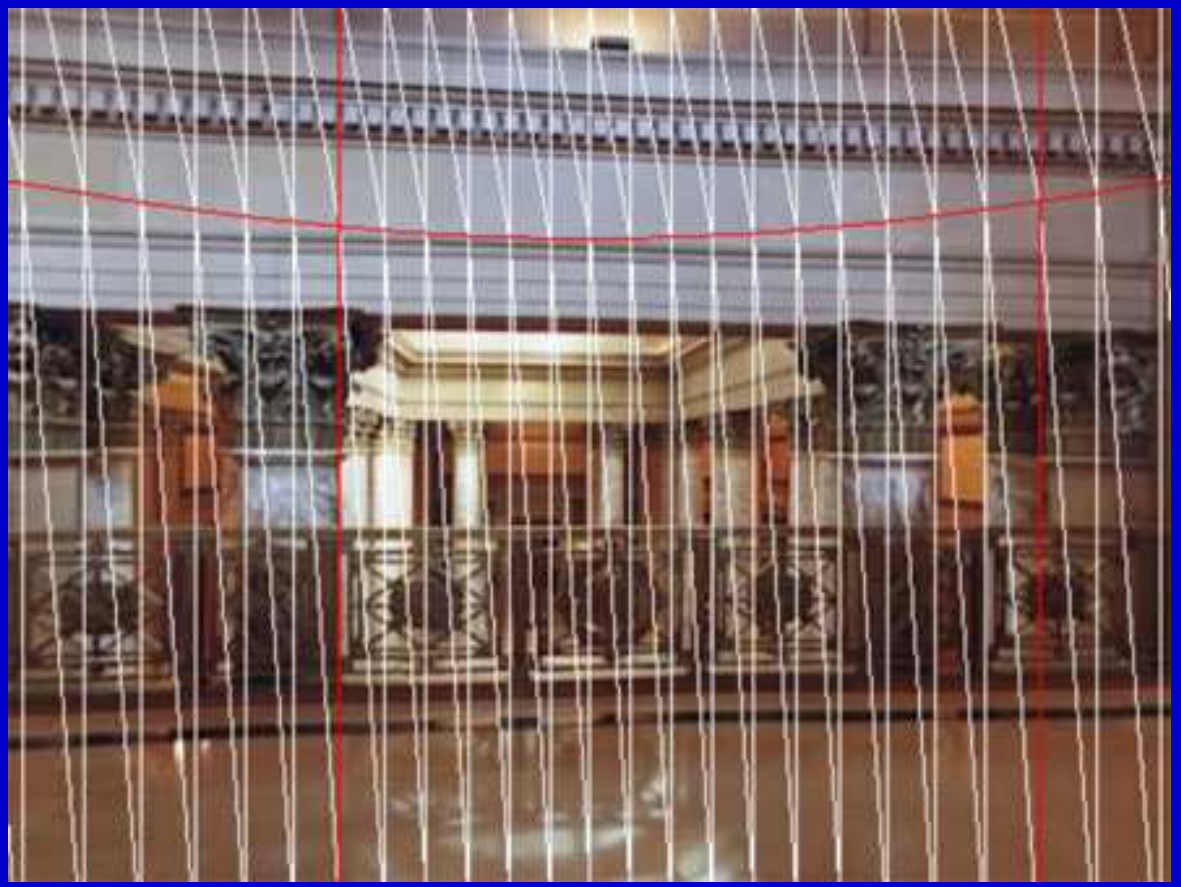


Filtering-Artifacts 1

Filtering-Artifacts 2



Results



View straightened out **Polygon boundary**

Panoramic Stereo Imaging

- Utilize a rotating stereo-camera pair for image acquisition
- Method:
  - image input (doubled)
  - projection warping
  - epipolar correction
  - displacement correction
- Stereoscopic visualisation

IBR-like Idea

- Use photographs of lightsources
- 2001: SIGGRAPH Award for Paul Debevec
- IMAGE-BASED LIGHTING
- www.debevec.com
- movies

IMAGE-BASED LIGHTING

- 2001: Paul Debevec, CVPR 2001 Short Course, 3.5 hours
- IMAGE-BASED LIGHTING:
- „integrating computer-generated imagery with live action photography that use measurements of real-world lighting to illuminate CG objects“

IBL Survey

- High-dynamic range images HDRI
- lighting acquisition (M. Gross)
- IBL and compositing
- real-time techniques
- software (Radiance, Maya...) and research

Rendering & Lighting Simulation Summary

- Point lightsource .. Photographs
- Radiance approach
- IBL idea
- Out of standard textbooks .. IBL
- Computationally very expensive
- http://www.debevec.com
- HOT research topic: conference papers

Visibility & Illumination

in images by Alan Watt

Compiled by permission by A. Ferko
Institute for Computer Graphics and Vision
TU Graz, Wintersemester 2001/2002

Reference and Permission

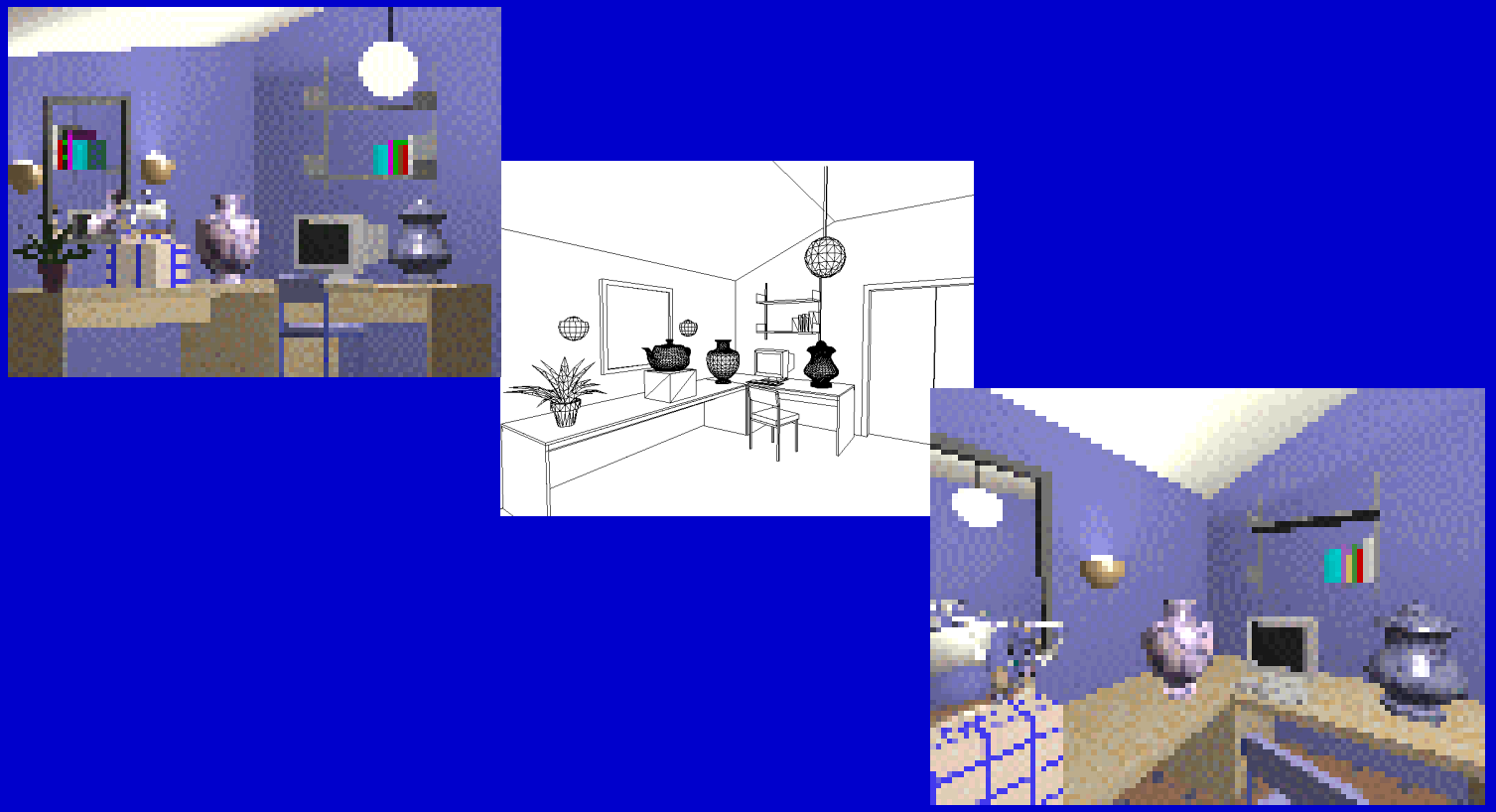
- [Watt00] WATT, A. 2000. *Three-Dimensional Computer Graphics.* Third edition. New York: Addison-Wesley 2000. - Accompanying CD by Pearson Education Ltd. 2000. ISBN 0-201-398559.
- Copyright 1997 A. Watt and L. Cooper
- Permission to copy without fee any or all of this material is granted, provided that the copies are not made or distributed for direct commercial advantage, the copyright notice and title of the image(s) appear and notice is given that copying is by permission of the copyright holder. To copy otherwise, or to republish, requires specific permission.

Basic Rendering Options

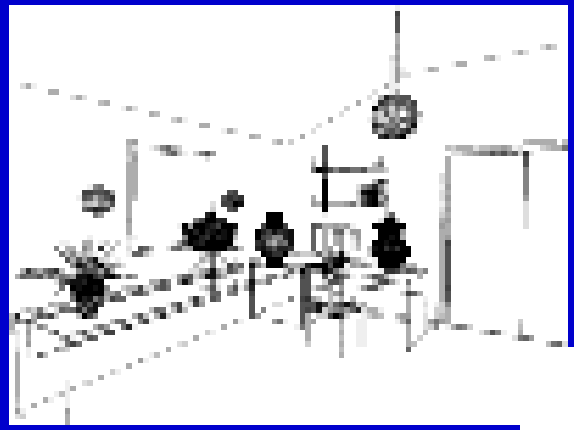
by A. Watt (2000)
selection and layout A. F. (2002)

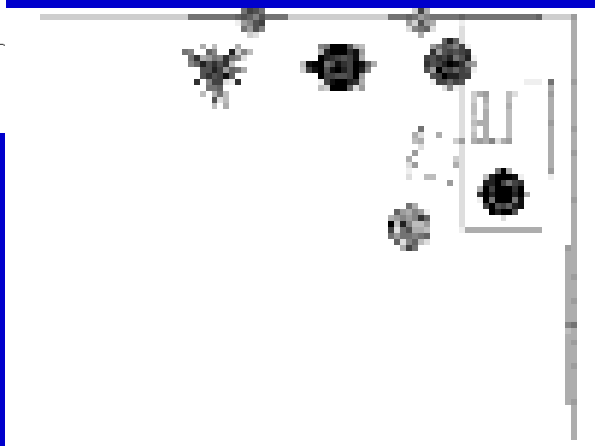
Main Scene

- ***Orthographic projection and the nearest perspective view***



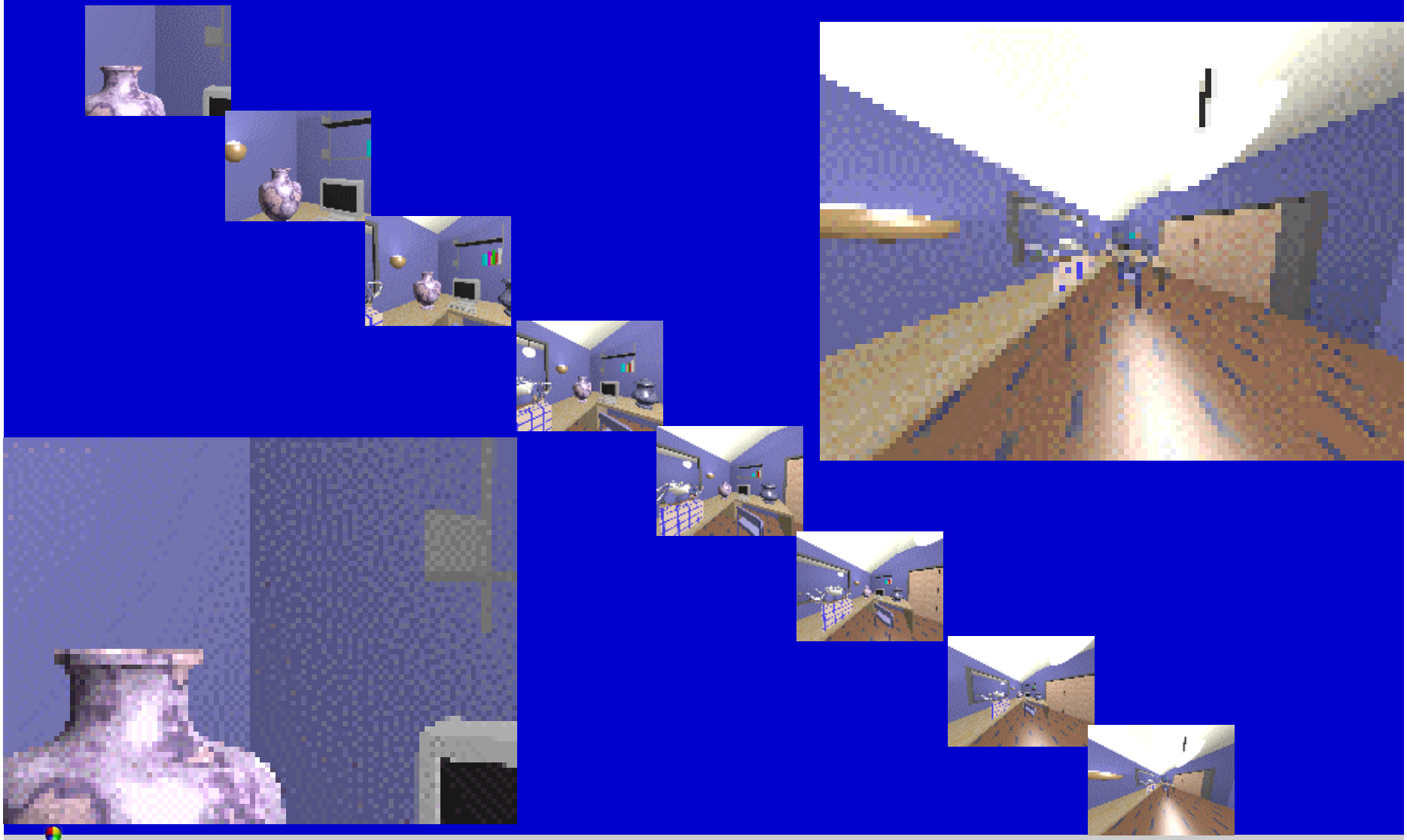
- ***Wireframe, hidden line removal, and hidden line from above***



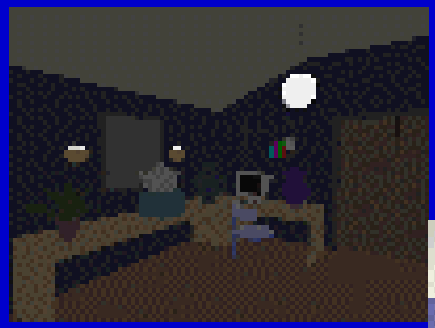


- ***Front and back clip planes, using front clip to see inside geometry***

- ***FOV varying from 20 to 160 degrees (8 images)***

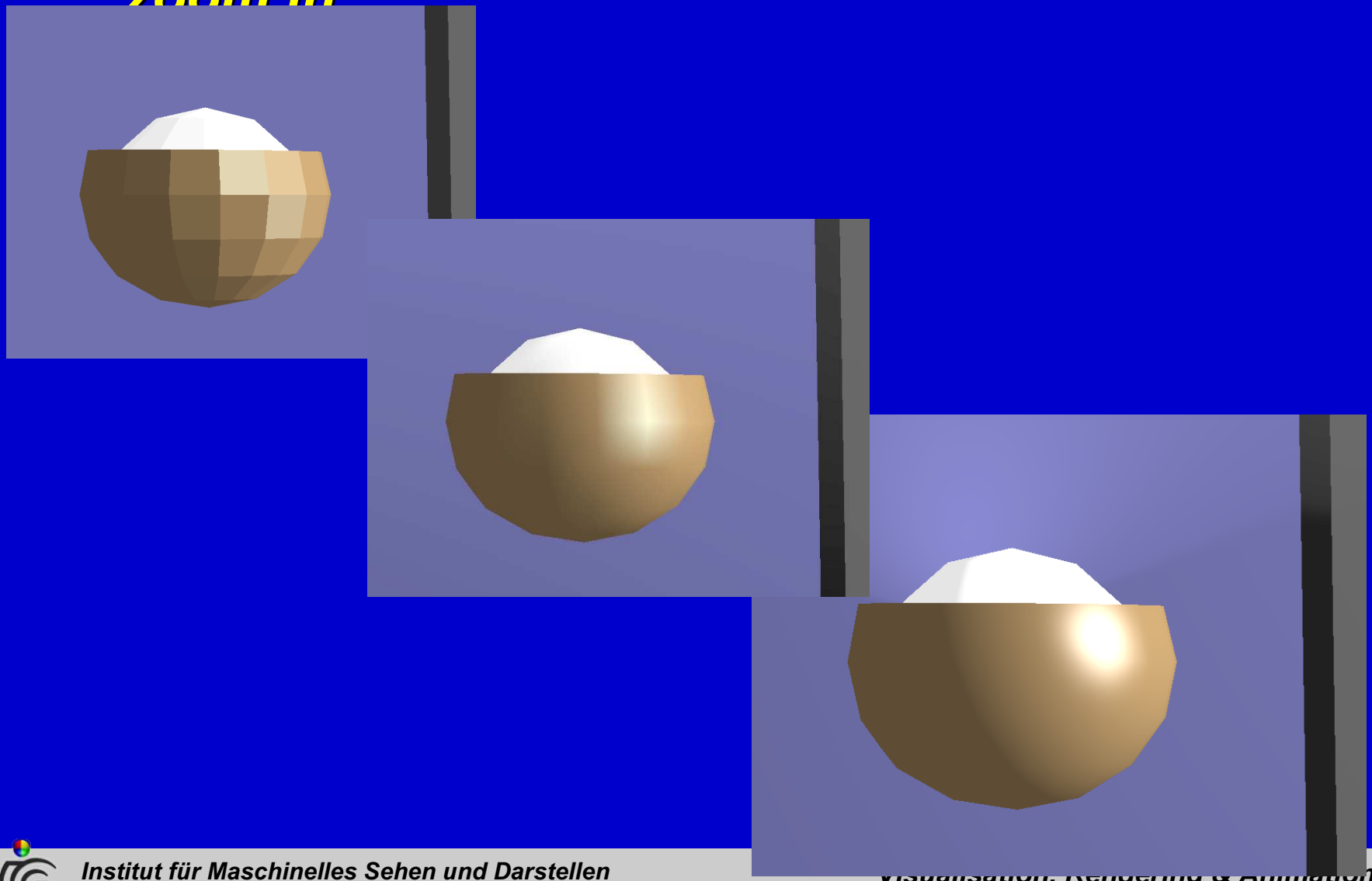


- ***Phong shaded scene: ambient, amb.+diffuse, amb.+diff.+specular***



- ***Flat, Gouraud, and Phong shaded scene***

- ***Flat, Gouraud, and Phong shaded scene - zoom in***



Material Realism

by A. Watt (2000)

selection and layout A. F. (2002)

- ***Materials used (5*5 array)***

- iron steel stainless steel machine steel antique brass

- polished brass copper bronze nickel zinc

- lead cast aluminiummachined aluminium magnesium gold

-
- burnished gold polished gold silver silver plate tungsten

- platinum chromium chromium plate graphite mercury

- *Difference of polished brass & gold ...*
  - *… hard to achieve by Phong shading*

5.
Light-Material Interaction

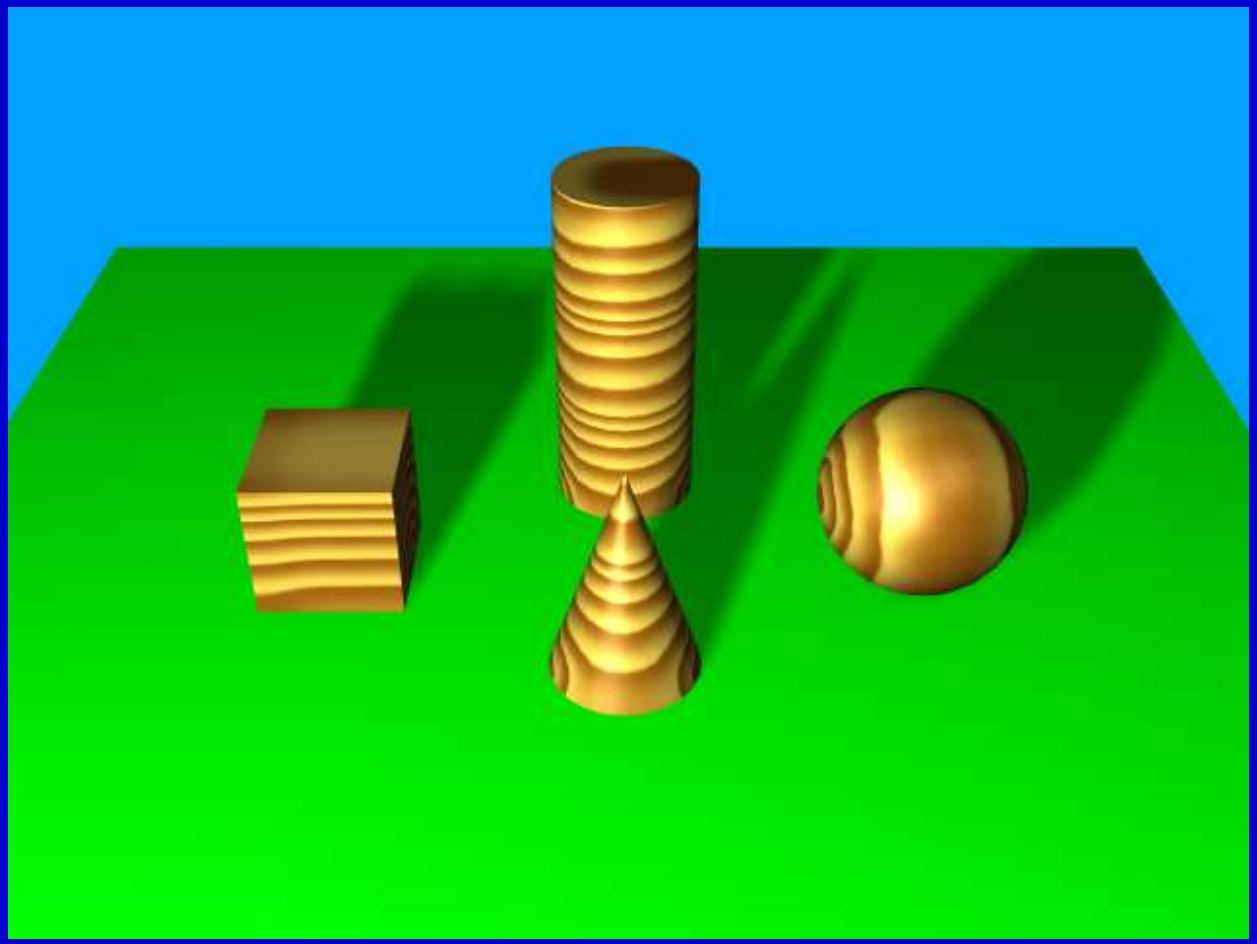
Shadow generation

Contents

5. Light - Material Interaction

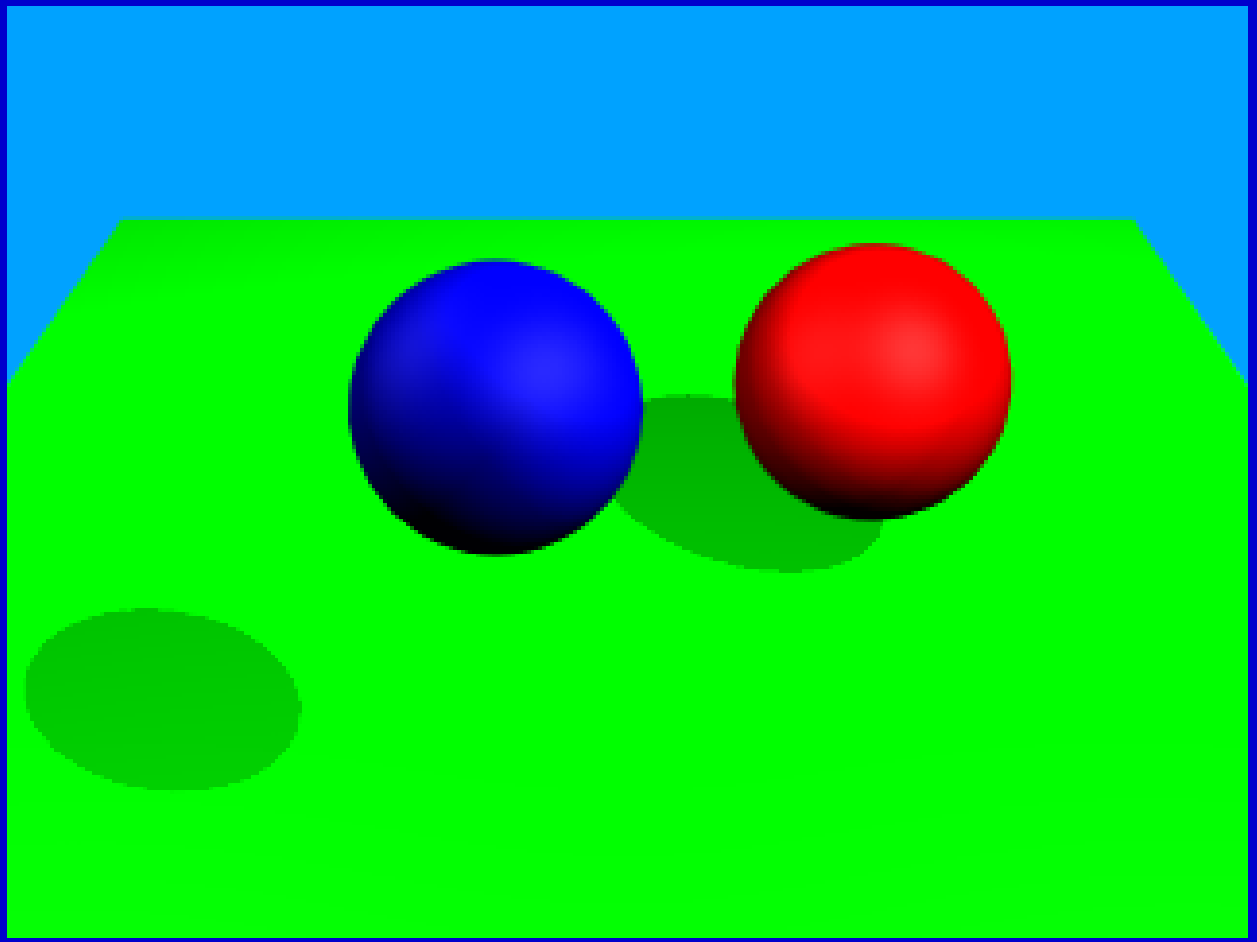
- Shadow generation
  - General
  - Image Precision Methods
  - Object Precision Methods
  - Shadow Volumes

- „Shadows play a subtle & vital role in our visual perception of an environment“ (Watt&Watt, p. 155)

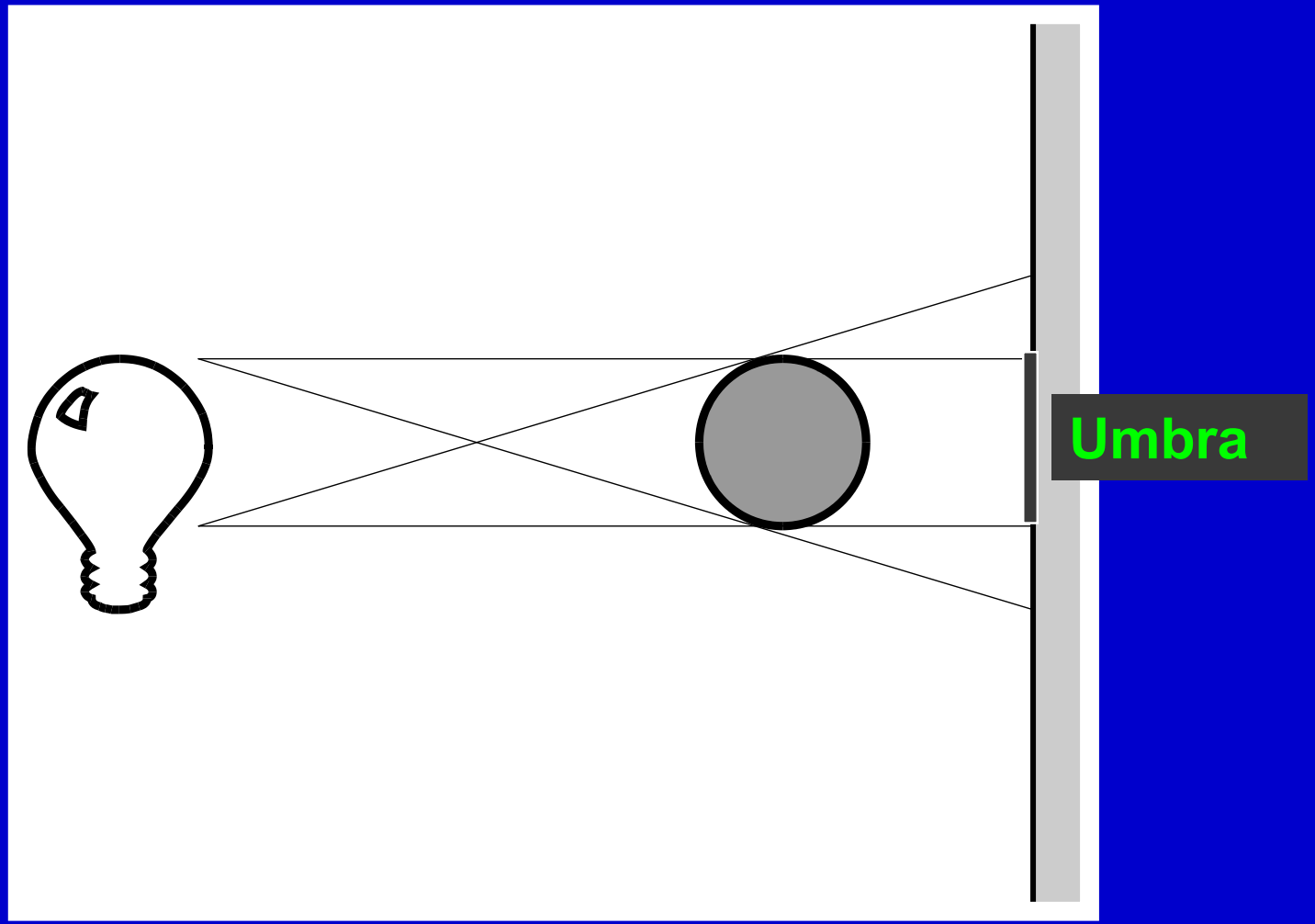


Shadow Generation

- Next improvement of realism
- Local illumination models
  - Visibility algorithms for lightsources
  - Object/Image precision
  - Point/Area light sources
  - Modification of the illumination model:
    $I = I_a k_a + \sum S_i f_{att} I_p [k_d (N \bullet L) + k_s (R \bullet V)^n]$
- Global illumination models: shadows solved implicitly



Umbra/Penumbra Effect



Methods Classification

- **Explicit shadow generation:**
  - Scanline Integration
  - Transformation & Clipping based method
  - Shadow Volumes
- **Implicit shadow generation:**
  - Ray Tracing
  - Radiosity

Scanline Integration

Principle:

- Standard „Scanline“ algorithm
- Projection of all „shadow polygons“
- Solving intersections:
  Scanline - shadow polygons

Pitfals:

- point light sources only
- polygonal scene model only

Recall: z-Buffering

- Object visibility
- Store the depth information per pixel (additional buffer)
- Compare the z-value against the actual one ($z_0 < z_D$):
  - $z_0 \leq z_D$: rewrite the pixel in „color-buffer“ and depth information in „depth-buffer“
  - $z_0 \geq z_D$: pixel/color given by another scene point

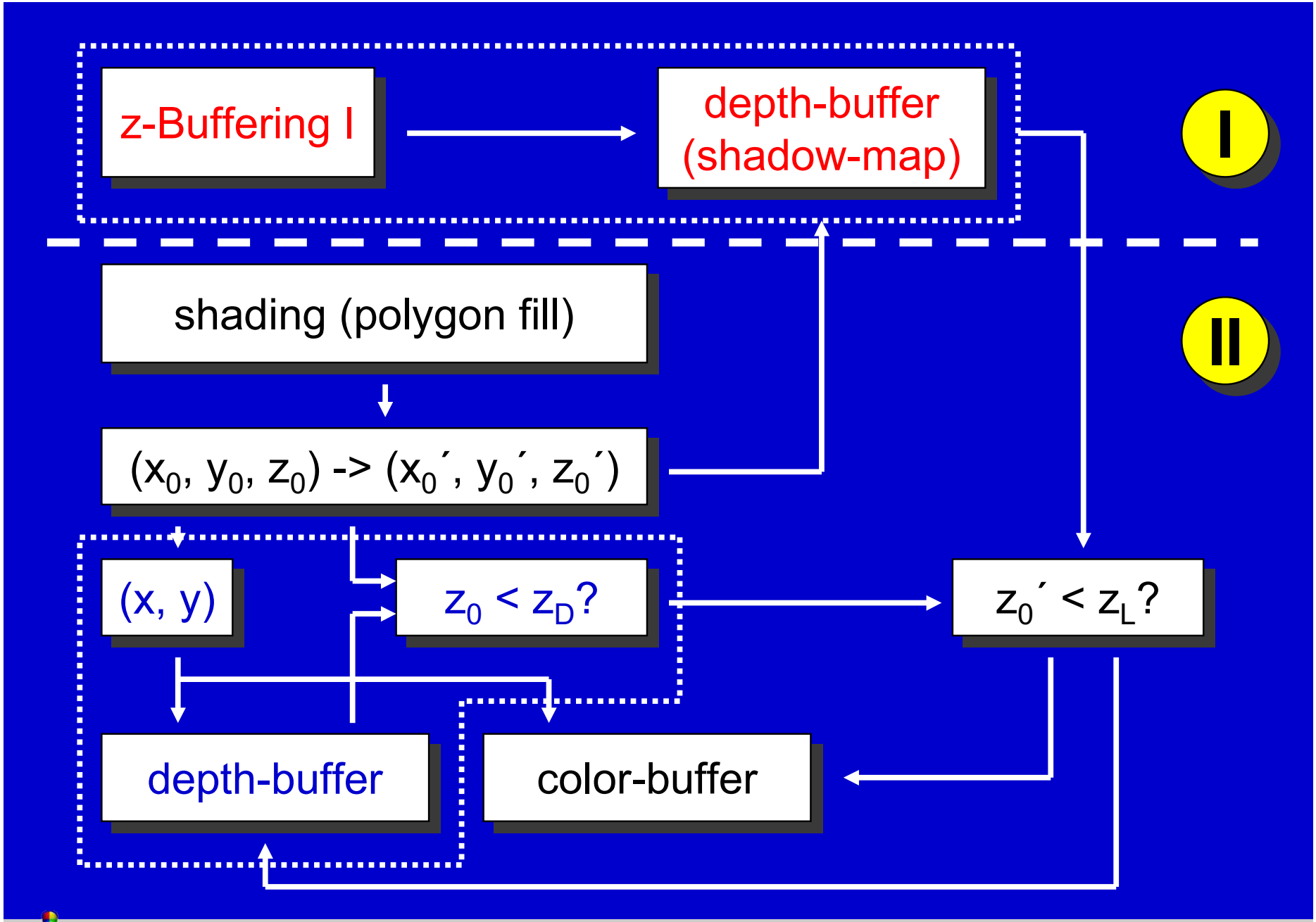
2-step z-buffer Method

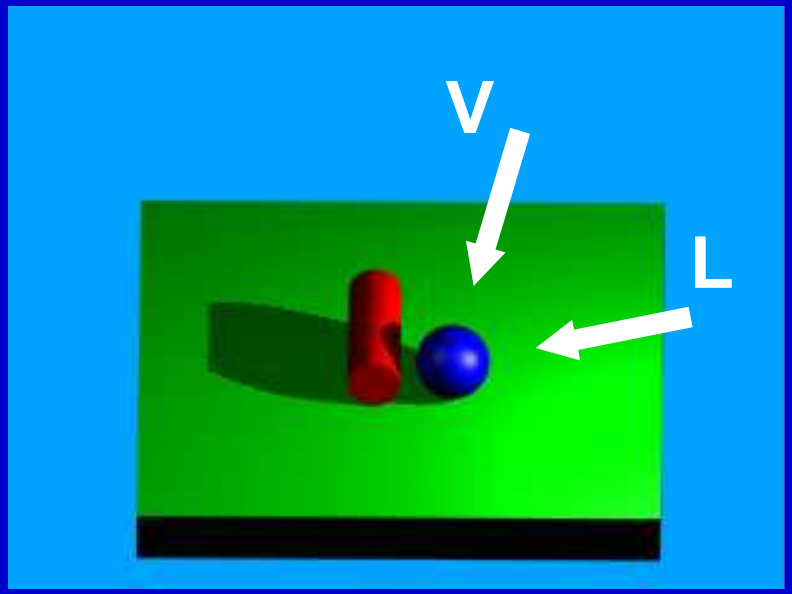
Algorithm:

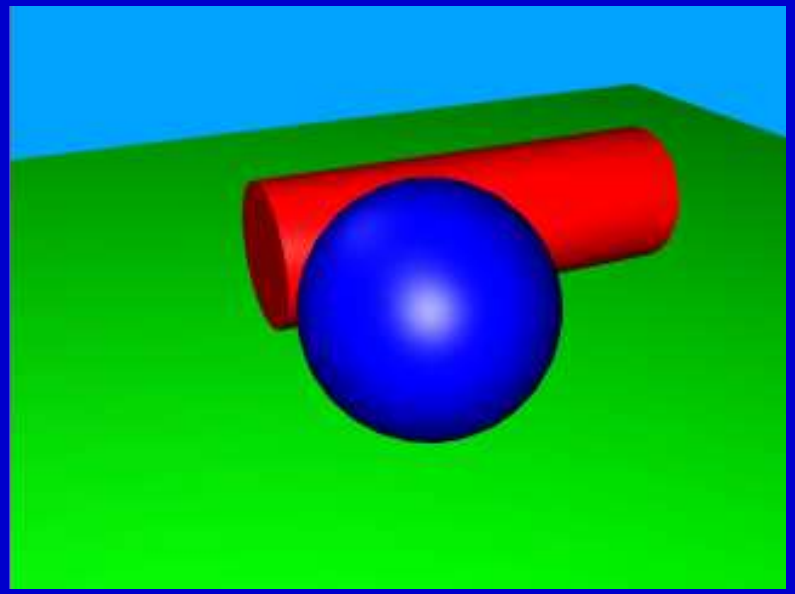
1.) z-Buffer algorithm with position of the lightsource (z-buffer = shadow-map)

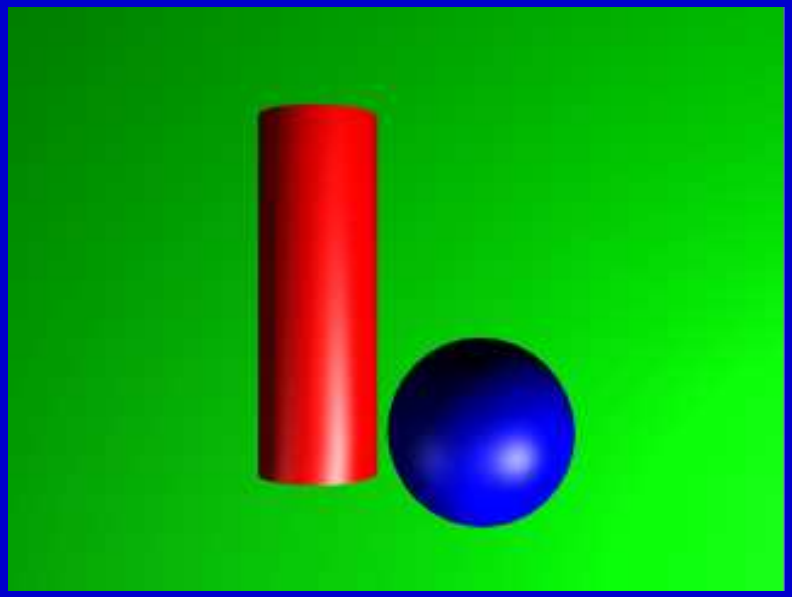
2.) Scene rendered from camera position

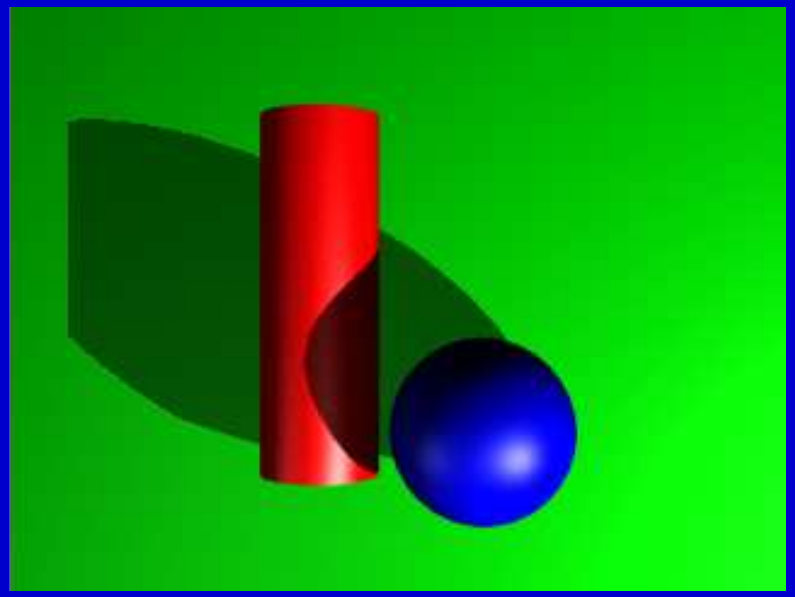
- Shading with the known method (i. e.: Phong)
- Transform to the lightsource coordinate system (x_0, y_0, z_0 -> x_0', y_0', z_0')
- z_L (shadow-map) < z_0' => the ambient term
  else => cummulative model
- Visibility using the z-buffer (z_0), as well


z-Buffering I
depth-buffer (shadow-map)
I
shading (polygon fill)
II
$(x_0, y_0, z_0) \rightarrow (x_0', y_0', z_0')$
(x, y)
$z_0 < z_D$?
$z_0' < z_L$?
depth-buffer
color-buffer


V
L

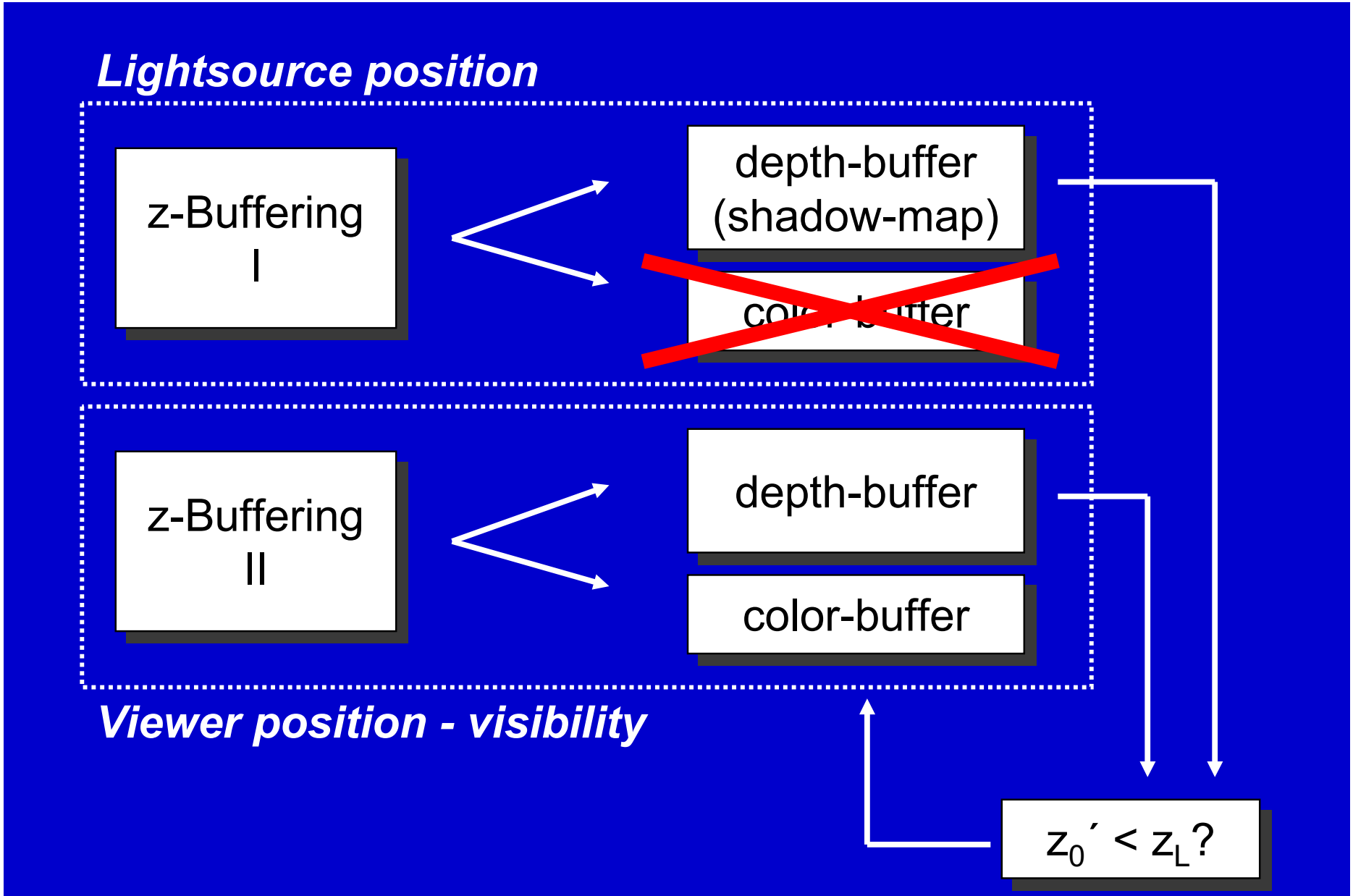


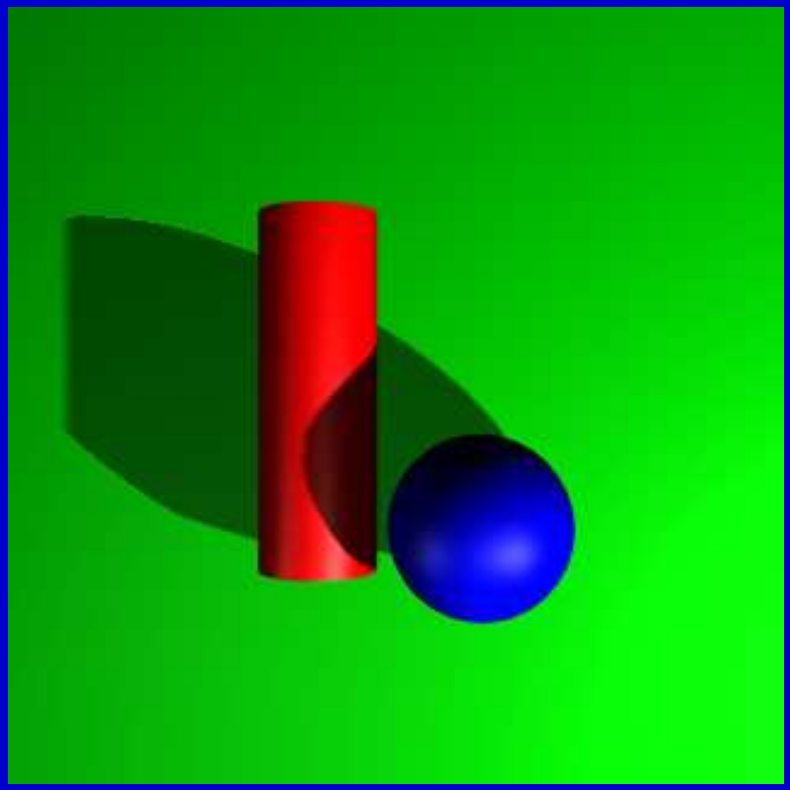


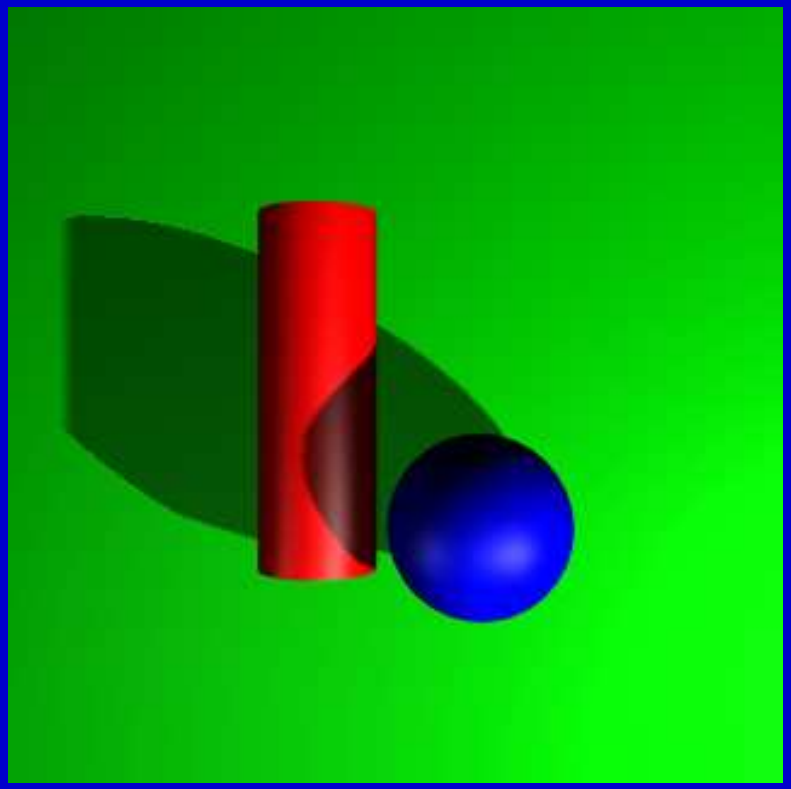


Postprocessing Alternative

- **Algorithm:**
  - execute z-buffer algorithm with lightsource position (z-buffer = shadow-map)
  - run z-buffer with camera position
  - for all pixels:
    z_L (shadow-map) < z_0‘ => darken color value
- **Pro: HW implementation possible**
- **Con: Highlights-artifacts**


Lightsource position
z-Buffering I
depth-buffer (shadow-map)
color-buffer
z-Buffering II
depth-buffer
color-buffer
Viewer position - visibility
$z_0' < z_L$?





Summary

Problems:

- numeric instability, e. g. with the postprocessing alternative
- no self-shadowing enabled
- Hint: „percentage closer filtering“
- multiple shadows computations per pixel

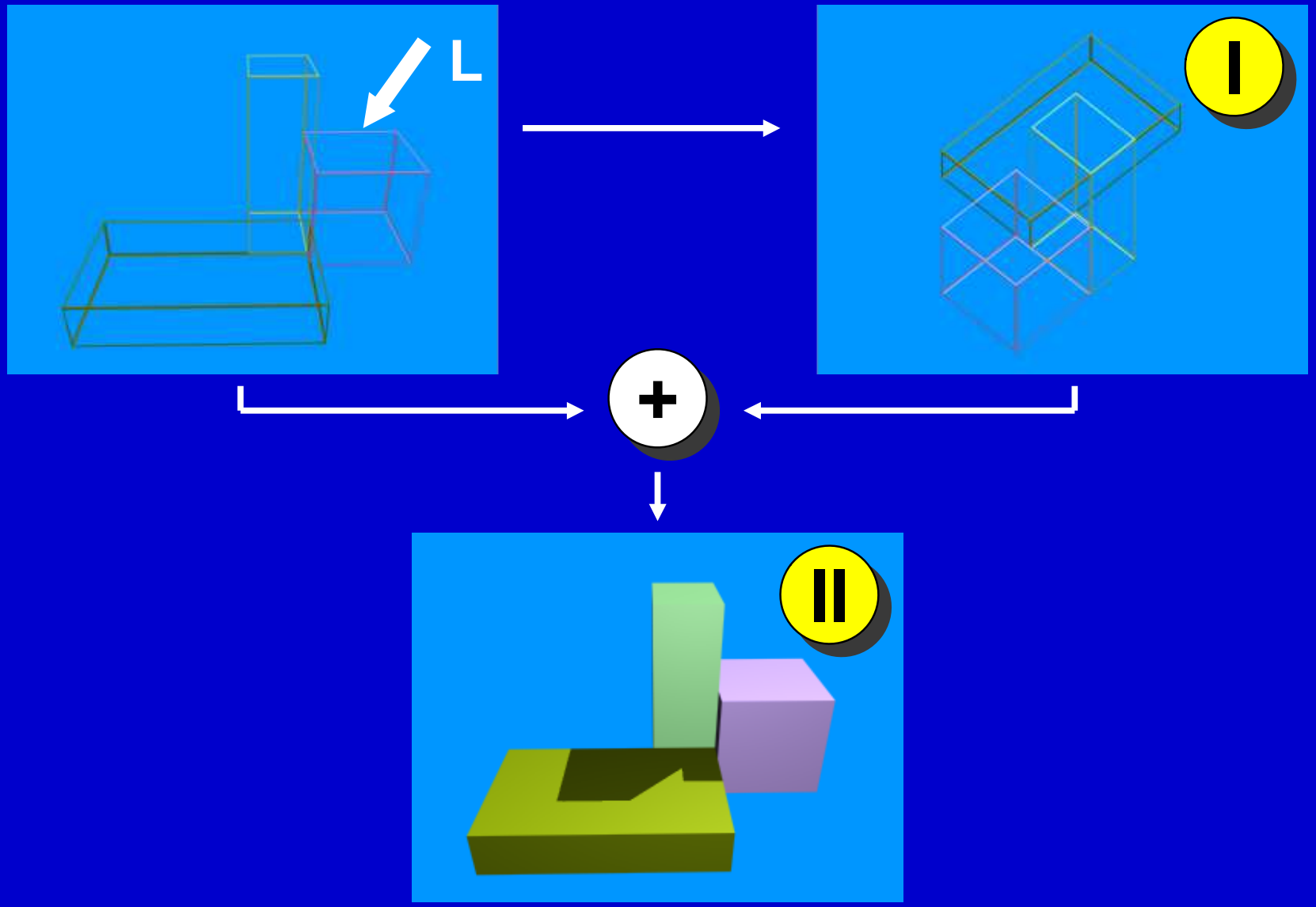
More light sources:

- multiple „shadow-maps“

Object-precision Method

Algorithm:

- Transform to the lightsource coordinates
- Object-precision visibility algorithm
- Combination of the geometries
  - Original model for ambient light
  - Visible geometry for cummulative model
- Scene rendering
  - any camera position (no new computation)
  - visibility either as shadow-computation or HW-oriented (z-buffering)


L
I
+
II

Confrontation

- **Image precision approach**
  - HW-oriented (z-Buffering)
  - simple (standard) comutation
  - Aliasing doubled
- **Object precision approach**
  - partly HW-oriented
  - complex computations
  - no new cost for moving camera/viewer

Shadow Volumes

Principle:

- *Silhouette of the object stretches with the rays from the lightsource a volume*
- *Shadows are intersections of polygons with the „shadow volume“*
- *Mixed image-/object-precision methods for practical conversion*

Stencil Buffer Realisation

Algorithm:

- 1. Polygon representation:
  - color-buffer: ambient part
  - z-buffering
- Silhouette-polygon representation
  - stencil-buffer: front-faces increments, back-faces decrements (stencil = die Schablone)
  - representation using z-buffer comparison
- 2. Polygon rendering
  - color-buffer: overlay of remaining term
  - rendering with stencil-buffer comparison

More Shadows

Shadows from transparent objects

Antialiased soft shadows

Watt & Watt (1992), pp. 155-177

5.
Light-Material Interaction

Shadow generation