# Grafické systémy, vizualizácia a multimédiá

## 2 / 1

*Letný semester 2006*

# Agenda

- Course Outline
- Class Assignments (3D authoring) – March 15
- Data Structures and Data Formats
- Parametric Curves and Surfaces
- …

# Úvod do PG, M-MASZ-035

- *Počítačová grafika, definícia, konceptuálny rámec, história, základné problémy a ich riešenie. Transformácie v rovine.* ***Základy špecifikácie grafických systémov a noriem.*** *Grafické objekty, ich vytváranie, kódovanie, modifikácia. Rasterizácia. Orezávanie a prieniky. Základné algoritmy spracovania obrazu.*

# Grafické systémy, vizualizácia a multimédiá,

- *1. Referenčný model počítačovej grafiky, architektúra multimediálneho systému, metodika matematického modelovania a vizualizácie, aplikačné oblasti počítačovej grafiky, vizualizácie a multimédií. Medzinárodná standardizácia (ISO, Web Consortium, EU normy). Visual computing – stručná história, sociálne implikácie, ekonomické a autorskoprávne aspekty. Duševné vlastníctvo.*
- *2. Grafická komunikácia. Geometrická modelovanie (tvorba jednoduchých objektov). Zaklady renderingu. Použitie API (OpenGL).*
- *3. HCI. Zaklady komunikácie človek-stroj. Navrh jednoduchého interaktívneho grafického rozhrania (GUI).*
- *4. Fyzické a logické vstupné zariadenia. Programovanie GUI. Interaktívne aspekty multimediálnych systémov a komunikácie.*
- *5. Kódovanie grafickej a multimediálnej informacie. Princíp kompresie dát. Funkčné normy pre počítačovú grafiku a spracovanie obrazu. Web consortium. De facto štandardy (OpenGL, oknové systémy). Špecifikácia funkčnosti grafického systému a GUI.*

# Grafické systémy, vizualizácia a multimédiá

- 6. Súradnicové systémy. Homogénne súradnice. Afinné transformácie (škálovanie, rotácia, posunutie). Implementácia jednoduchého zobrazovacieho kanála. Orezávanie. Rasterizácia usecky (DDA, Bresenhamov algoritmus).
- 7. Implementácia základných 2D grafických výstupných prvkov: lomená čiara, výplňová oblasť, text.
- 8. Hierarchia obrazu a 2D počítačová animácia.
- 9. Jednoduché modely farieb (RGB, CMYK). Kultúrny význam niektorých farieb. Návrh web stránok. Používanie textu v obrázkoch. Web publishing.
- 10. Ludske vnimanie. Analogove a digitalne reprezentacie pre multimedia. Spracovanie obrazu a zvuku.
- 11. Zakladne funkcie pre vizualizaciu. Historia vizualizacie. Vizualizacne scenare. Interaktivne multimedialne tituly.

# Grafické systémy, vizualizácia a multimédiá

- *12. Uvod do 3D grafiky. Problem viditelnosti a z-buffer. Zdroje svetla. Parametre kamery. Graf sceny. Norma VRML.*
- *13. Interakcia svetla a objektov. Lokalny osvetlovaci model a tienovanie (konstatne, Gouraud, Phong). Textury. Fotorealisticke zobrazovanie.*
- *14. Modelovanie 3D scen. Parametricka a implicitna reprezentacia. CSG a B-rep. Proceduralne modelovanie (fraktaly a casticove systemy).*
- *15. Pocitacova animacia. Pocitacove hry a virtualna realita. Snimanie, modelovanie a zobrazovanie medicinskych dat.*
- *Lit. Ružický, E. 1995. Počítačová grafika a spracovanie obrazu. Bratislava: Sapientia 1995.*

# Web Pages via www.sccg.sk/~ferko

- 
- Andrej Ferko: Work Page... WELCOME = VITAJTE

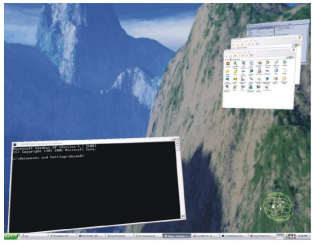
- Better pages:
- SIGGRAPH EG SCCG CESCG SIGGRAPH2003 SIGGRAPH2002 CGpapers Citeseer Helwig's Calendar ACM SIGGRAPH Calendar IEEE Calendar Techexpo pg.netgraphics.sk Prusinkiewicz Webby Facedemo Helwig's SmileyDict. D.Mount's Book GOOOOOGLE
    - http://www.cs.umd.edu/~mount/427/Lects/427lects.pdf

# Computer Graphics @ CU

- UPG resp. GSVMM Classes - Intro
-
- BSc. Work
- Diploma (MSc.) Work – Mgr.
- (rigorosum - RNDr.)
- PhD.

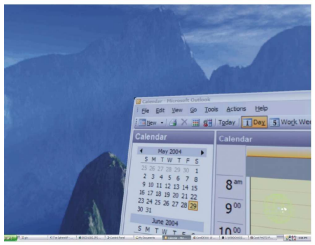
# 3D Desktop Enhancement for Windows

www.hamar.sk/sphere



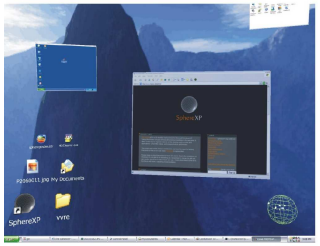
## Navigation

- movement is limited
- the viewport is always facing apart from the sphere center
- currently supported platform - Windows XP



## Use of the mouse

- almost as working with the regular desktop
- left button for manipulation with objects
- the scroll for moving the viewport
- right button for Z-movement



## The Sphere Comunity

- webpage
- newsletter
- forums
- almost 300 000 downloads

## Try it yourself

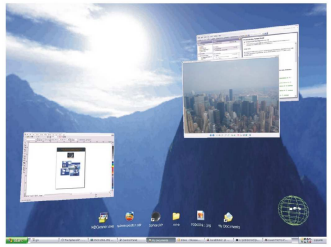
- www.hamar.sk/sphere


Faculty of Mathematics, Physics & Computer Science
Commenius University Bratislava, Slovakia
Dušan Hamar



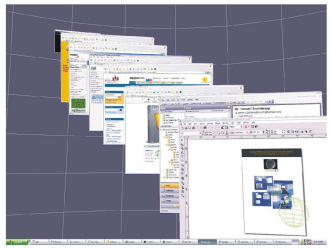
3D Desktop Enhancement for Windows

www.hamar.sk/sphere

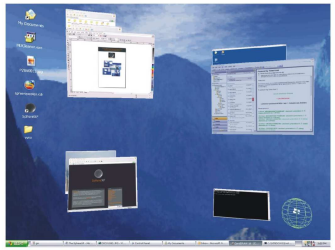
## What is the Sphere?

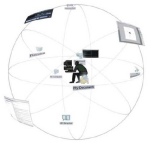
- a 3D desktop replacement
- a natural way of extending the classic desktop metaphor
- platform for 3D applications



## A simple idea

- display icons and windows inside a sphere
- provide easy manipulation and navigation instead of a flat desktop





## Used technologies

- Microsoft .NET 1.1
- OpenGL (Tao)
- Currently supported platform - Windows XP


Faculty of Mathematics, Physics & Computer Science
Commenius University Bratislava, Slovakia
Dušan Hamar

# Research Activities @ CU

## In the field of Computer Graphics

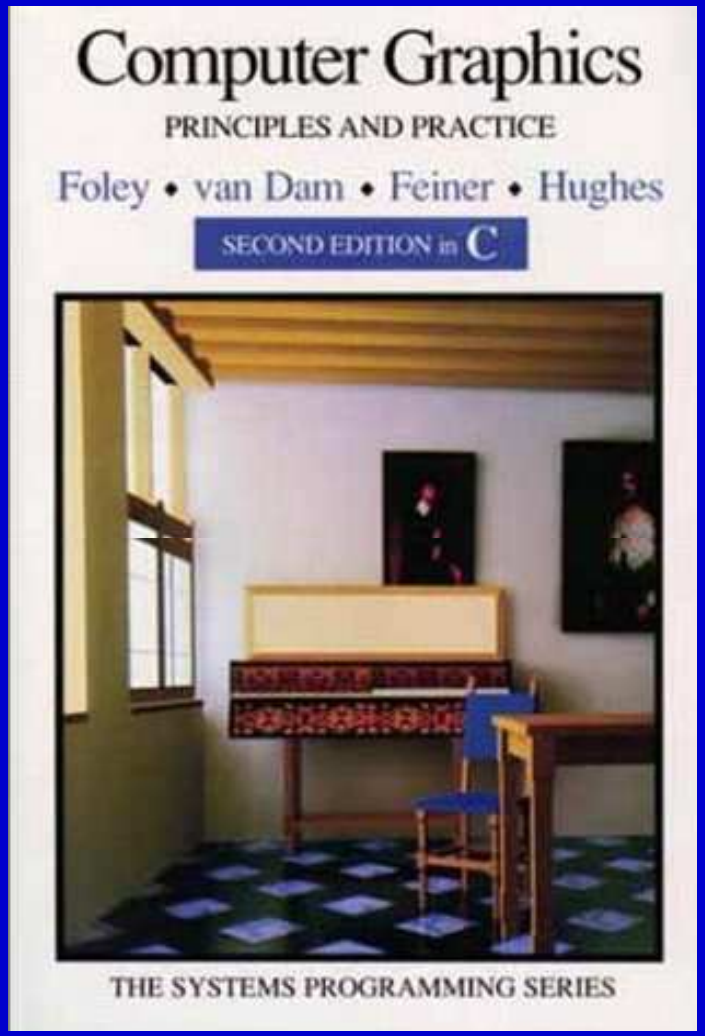
- **Virtual Environments**
  - Virtual Cities - outdoor
  - Virtual Museum (Žilina, Bratislava)
  - Others (in progress, too)... (Štiavnica)
  - MDPT: CD version
  - EPUD
- **Medical Volume Graphics, Rendering**
  - Doc. Šrámek, Doc. Ďurikovič
  - ...

# Virtual Heart of Central Europe

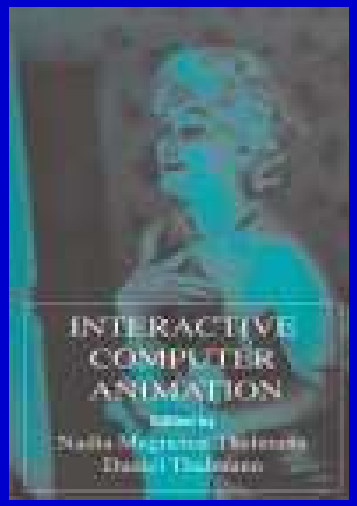
# Computer Graphics Motivation

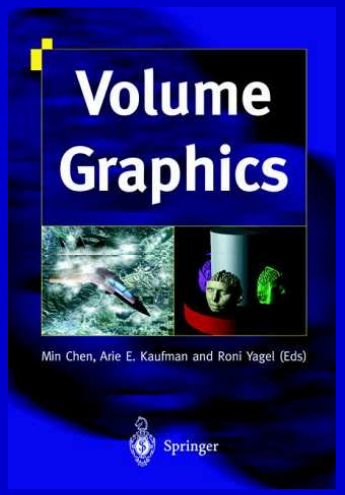
- no Marxistic mass of workers, but the technology
- three great discoveries
  - microprocessor
  - open system
  - global network
- problems
  - lack of investments and standards
  - byrocratic stupidity
  - low speed bandwith
  - enormous data extent...
- technology borders, knowledge and/or fantasy

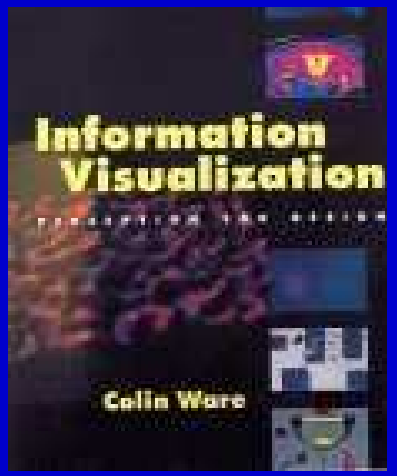
# Computer Graphics Bible

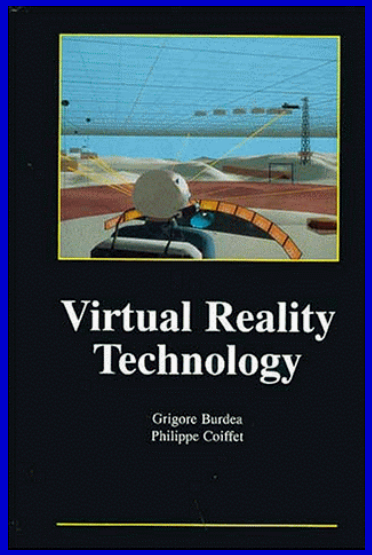


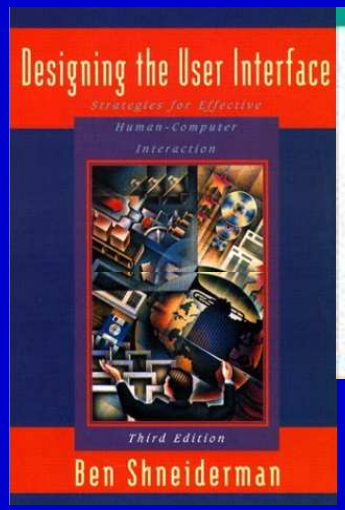
# Graphics Sub-comunities



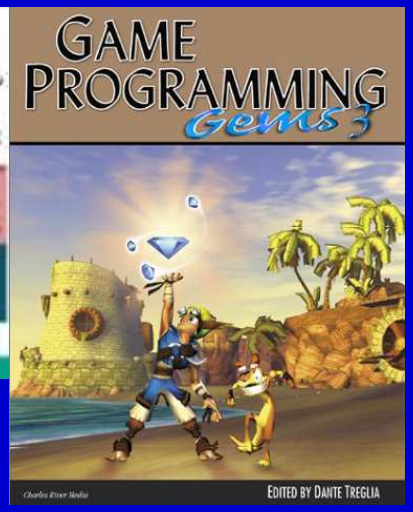



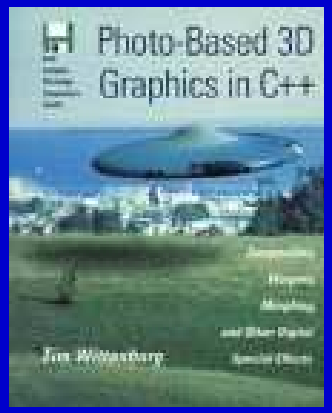



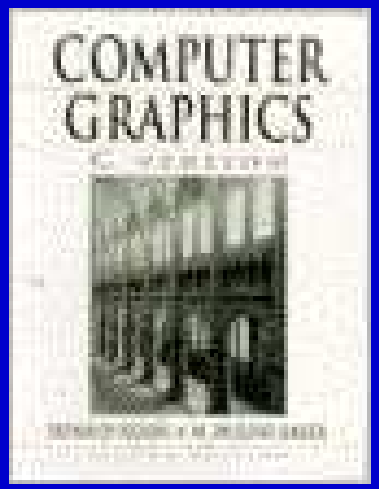



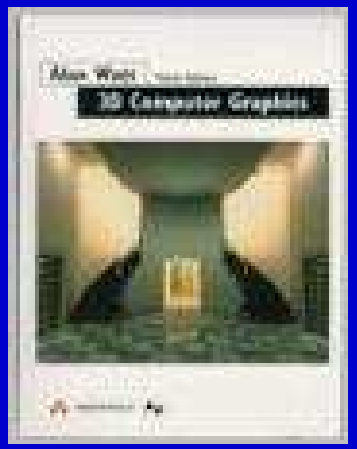


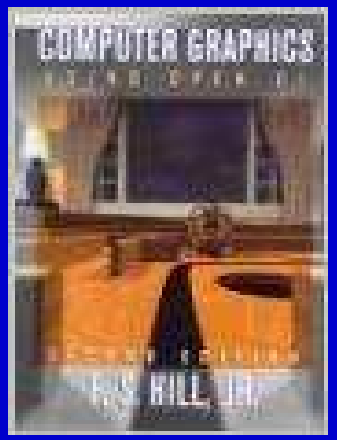


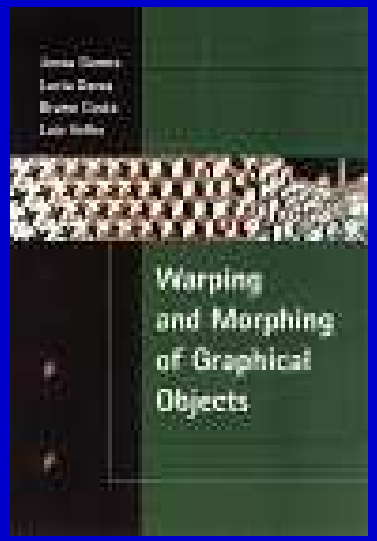



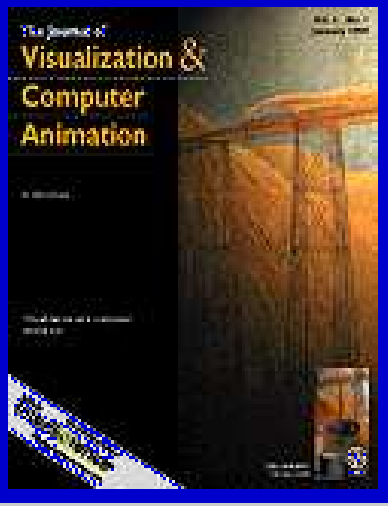

# Graphics Sub-comunities



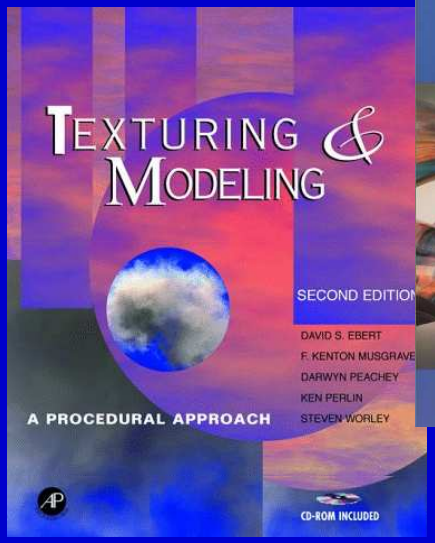



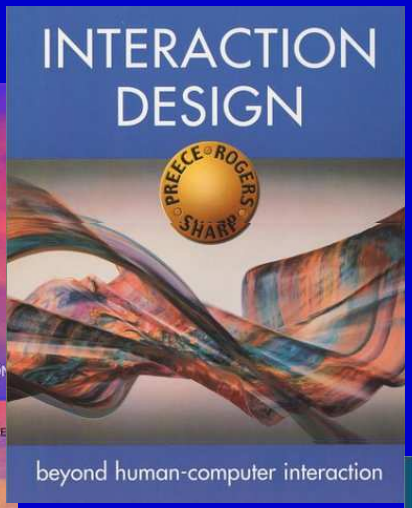



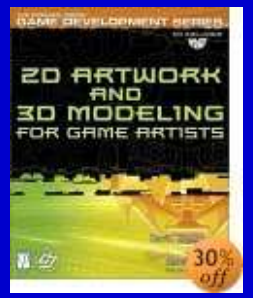



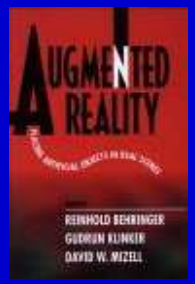
# Graphics Sub-comunities



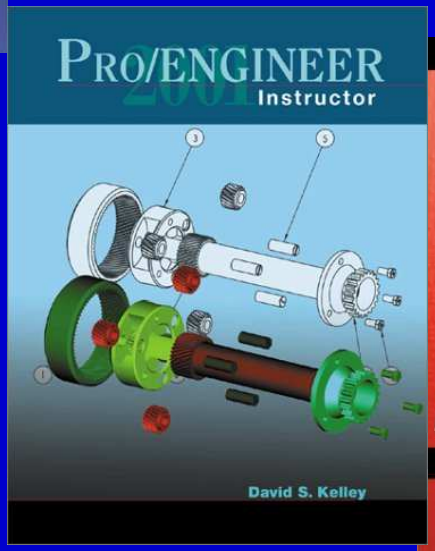



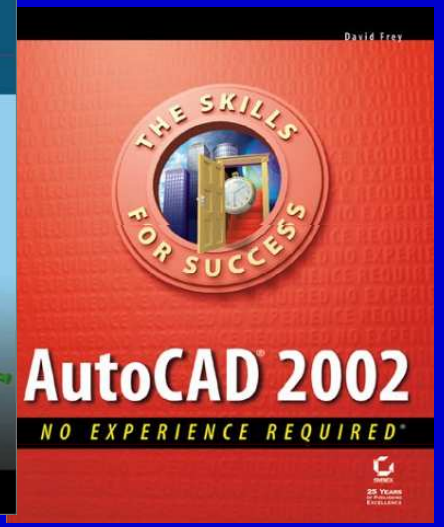











# Hot CG Events

# Conference Calendars

- **IEEE Calendar**
- **http://www.computer.org/conferences/calendar.htm**

- **ACM SIGGRAPH Calendar**
- **http://www.siggraph.org/calendar/calendar.html**

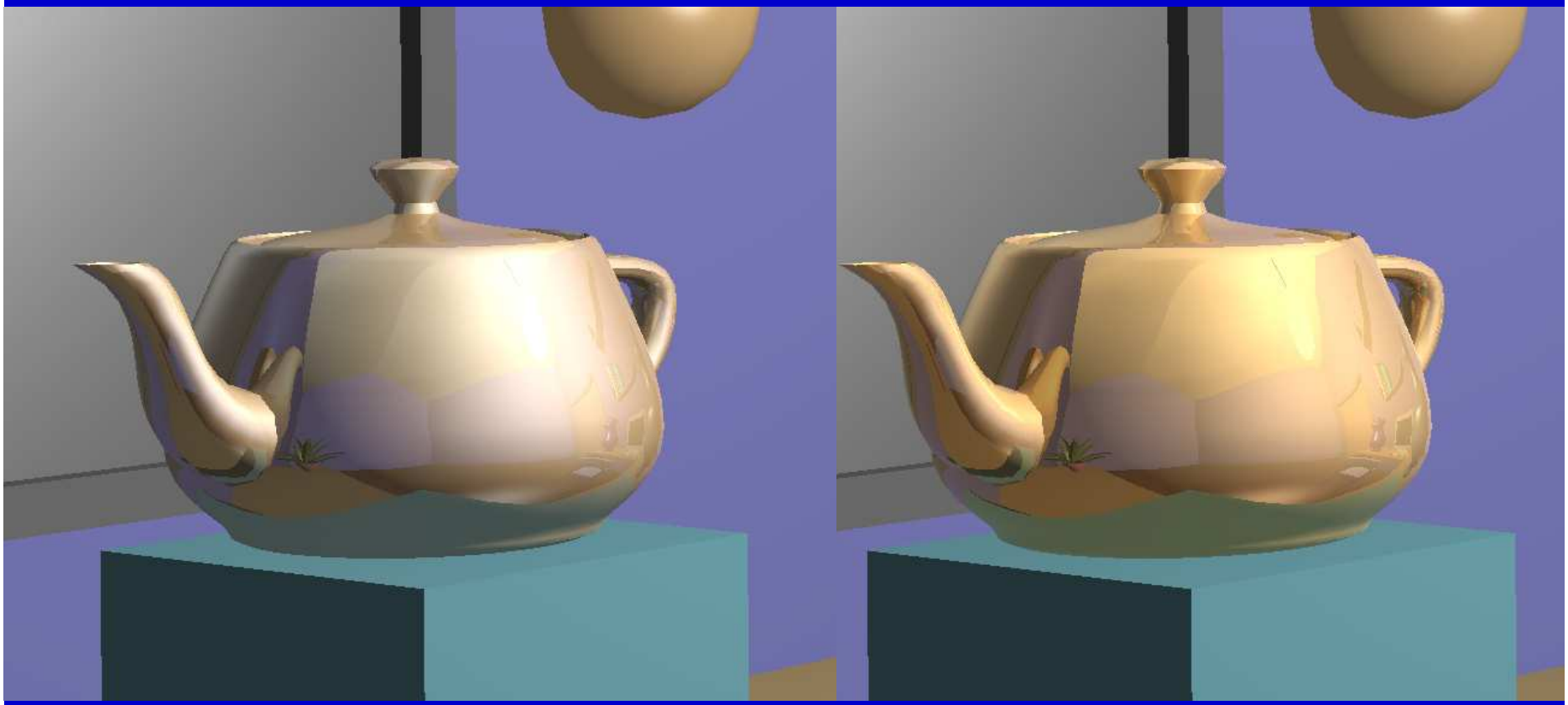
# Compare Reality - Synthesis

Photograph

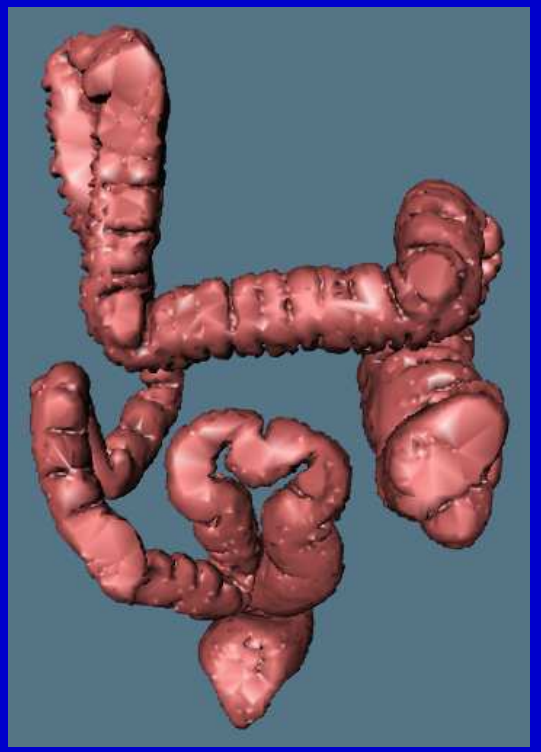
Rendering using the [illegible] method

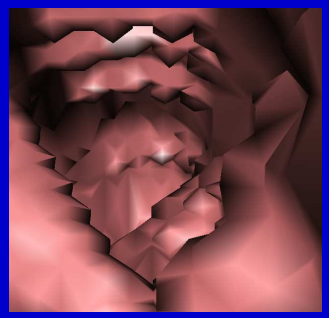
# Simulating the Office Scene

# Difference of polished brass & gold

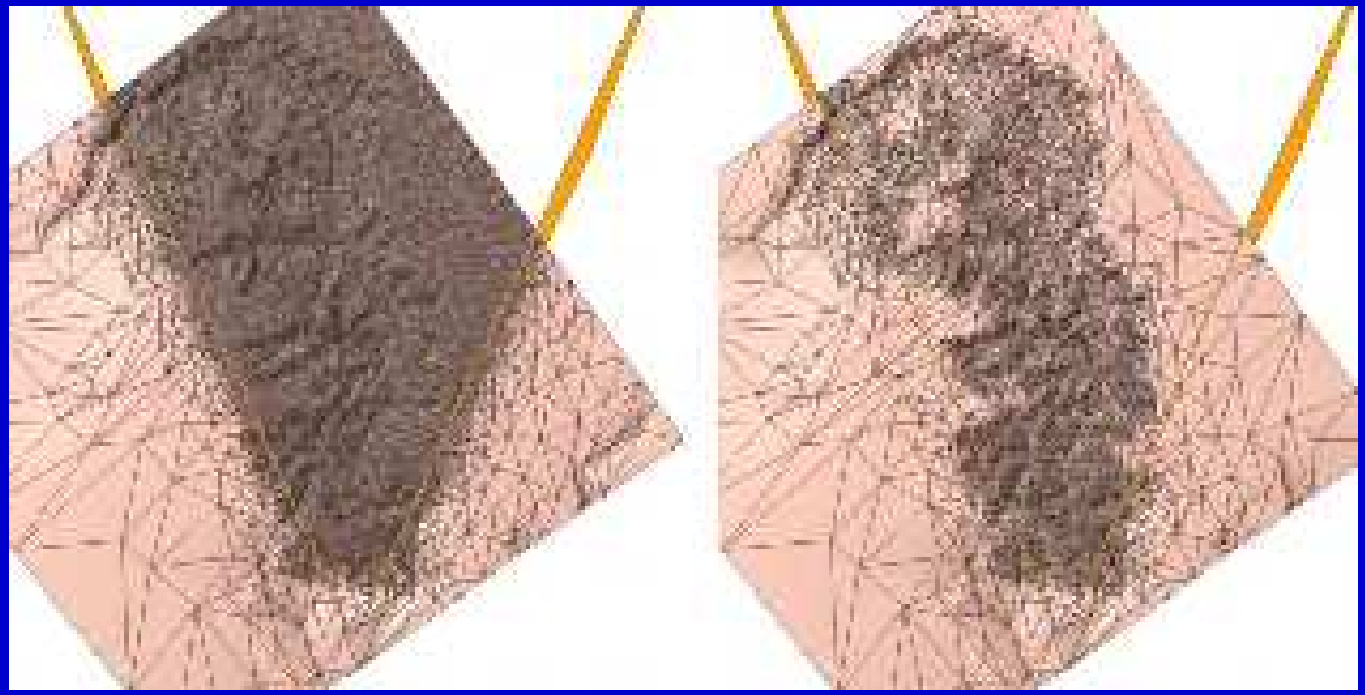


# Medical VolVis





# Multiresolution Analysis



# What means Photorealism?


- *2001* - ***Final Fantasy*** *( The Spirits Within )*
  - *all characters animated, photorealism, hair for millions dollars*


U.S.M.C


USMF


FINAL FANTASY


FINAL FANTASY


FINAL FANTASY

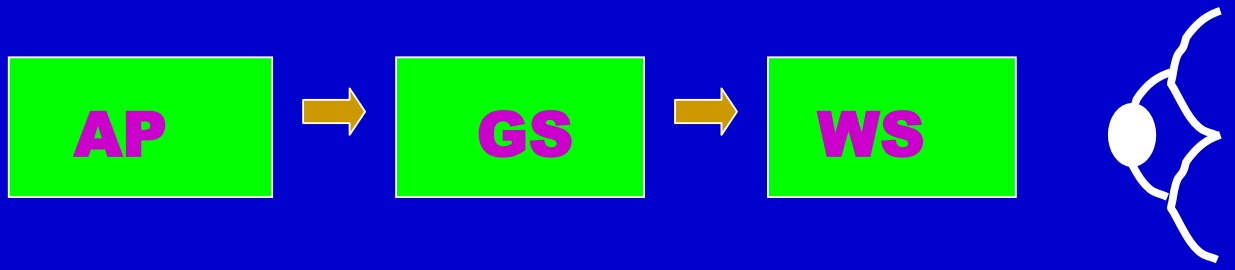

FINAL FANTASY


FINAL FANTASY


FINAL FANTASY

# CG Reference Model

- *Problem – Application Program - Solution*




GUI HVS

# The Graphics Pipeline Steps

input devices

modeler

*Modeling*

transformations

projection

clipping

visibility

*Rendering*

shading

rasterization

*Output*

device drivers

output devices

# The Three Brains/Heads

input devices

modeler

*Modeling*

transformations

projection

clipping

visibility

shading

*Rendering*

rasterization

*Output*

device drivers

output devices

# Triple Schizofreny

**Application Programmer**

input devices → modeler

*Modeling*

**GS author**

transformations → projection → clipping → visibility → shading → rasterization

*Rendering*

*Output*

device drivers → outp

**User, bonnie, ISO: operator, customer, boss...**

# 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…)

# Computer Science Context

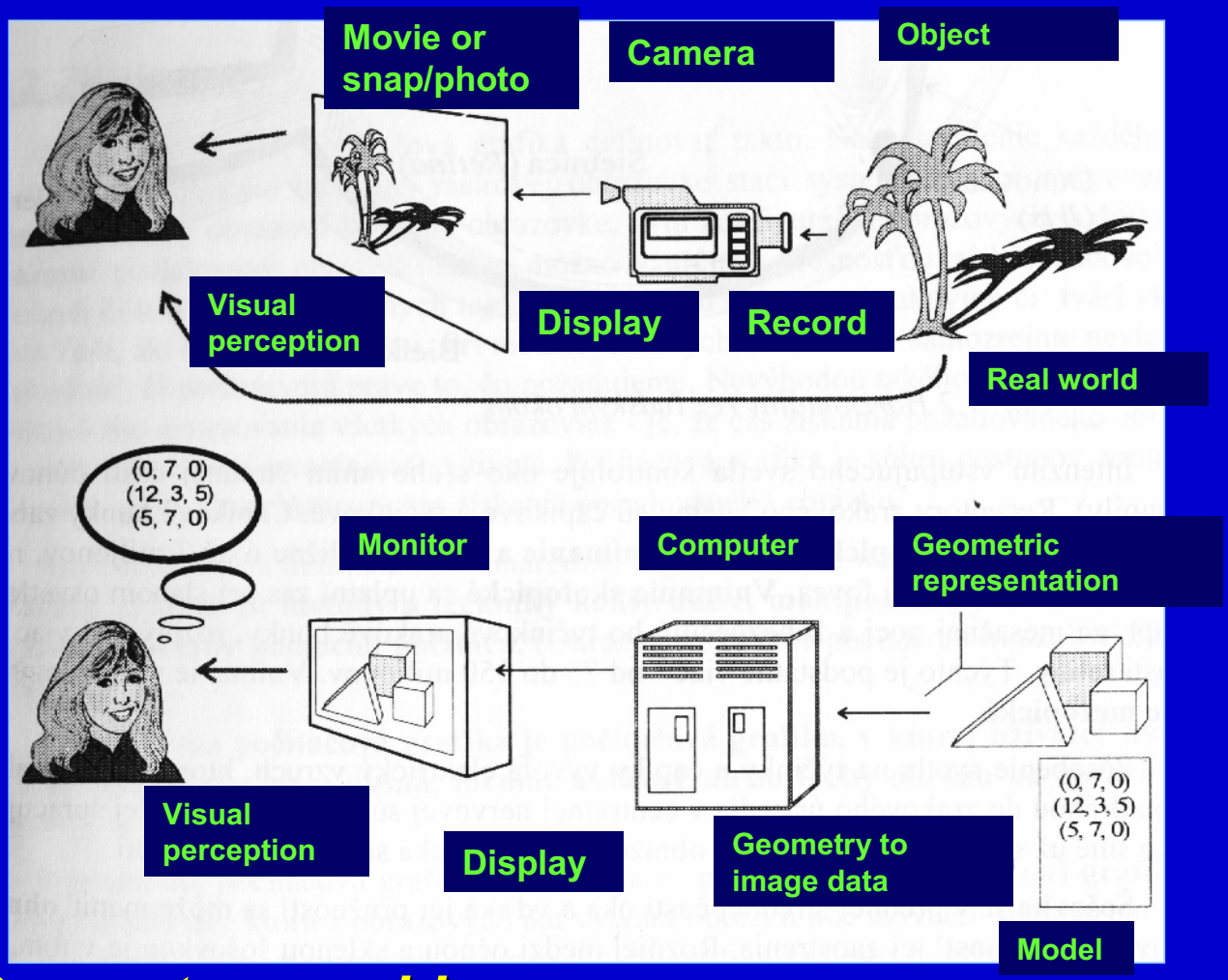
## Reference Models:

- Simple
- ISO (GKS, PHIGS, PREMO, ...)
- Others… (OpenGL, DirectX…)

## Analogies:

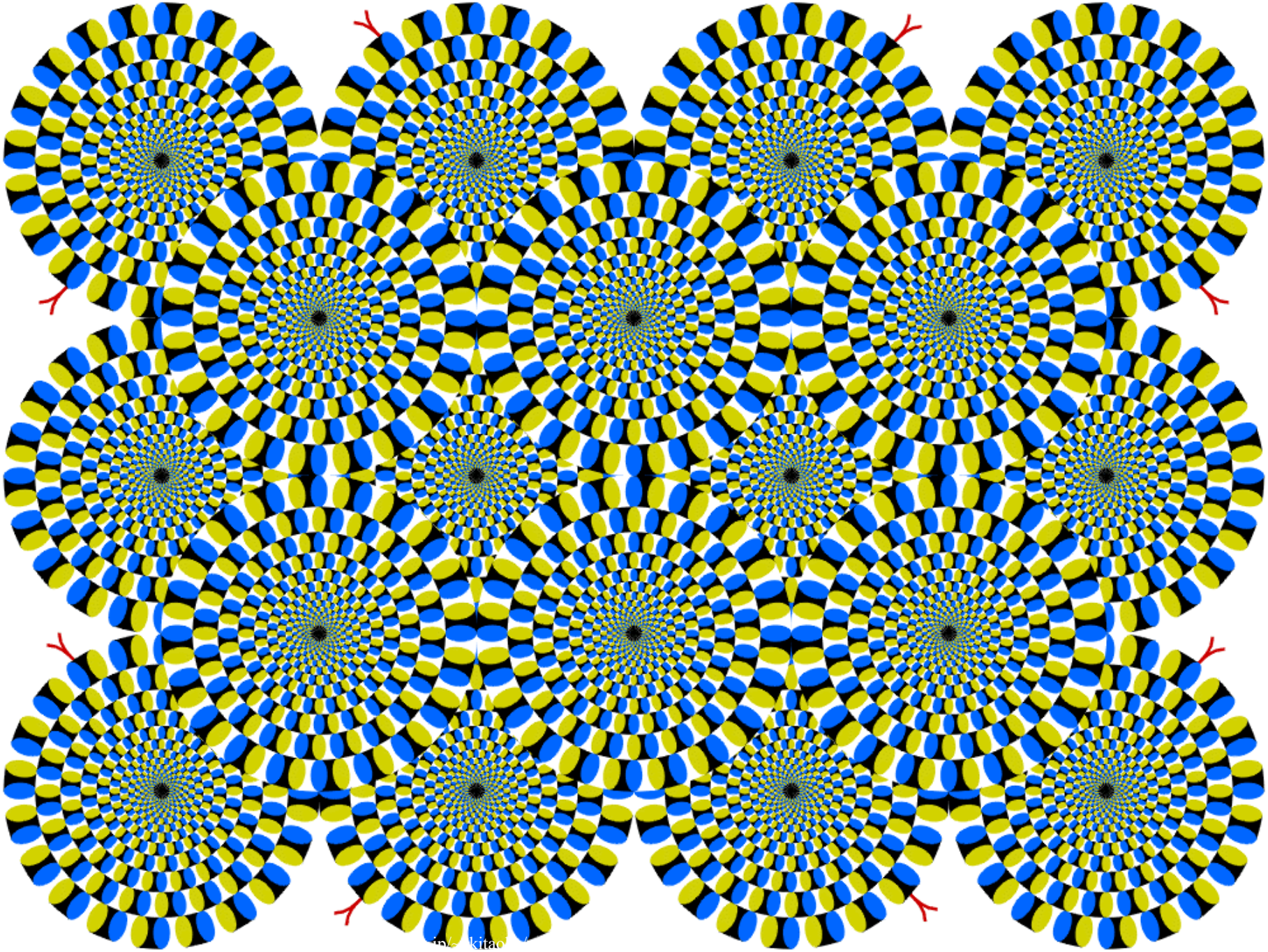
- Photography

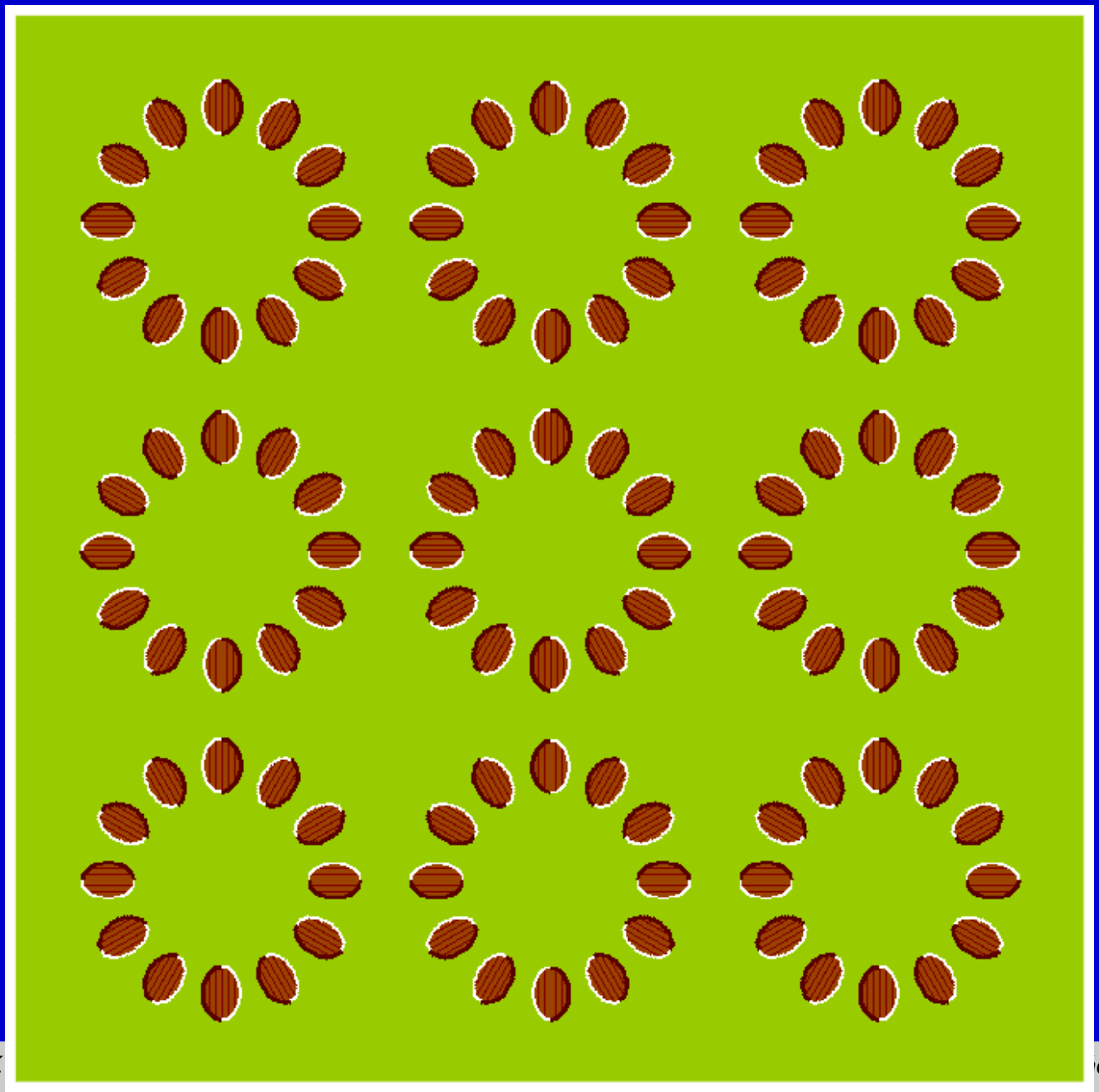
# Analogy: photography & computer graphics




- **ISO: Computer graphics:** *methods & techniques for construction, manipulation, storage and displaying pictures using computer.*

# ESCHER





# CG - Modeling

# Data Structures & Data Formats

*Ways of object representation*

# Object Representations

- Curves and Surfaces
- Solid Modeling
  - Boundary Representation
  - Spatial Enumeration Models
    - Spatial-Occupancy Enumeration (Voxel)
    - Binary Space Partitioning (BSP) Trees
    - Octrees
  - Constructive Solid Geometry (CSG)
  - Function Representation (F-rep)

# Criteria

**Modeling:**

- Representation Power
- Transformation / Combination
- Interactivity Support
- Multiple Use, Generality

**Rendering:**

- Representation Precision
- Memory Requirements

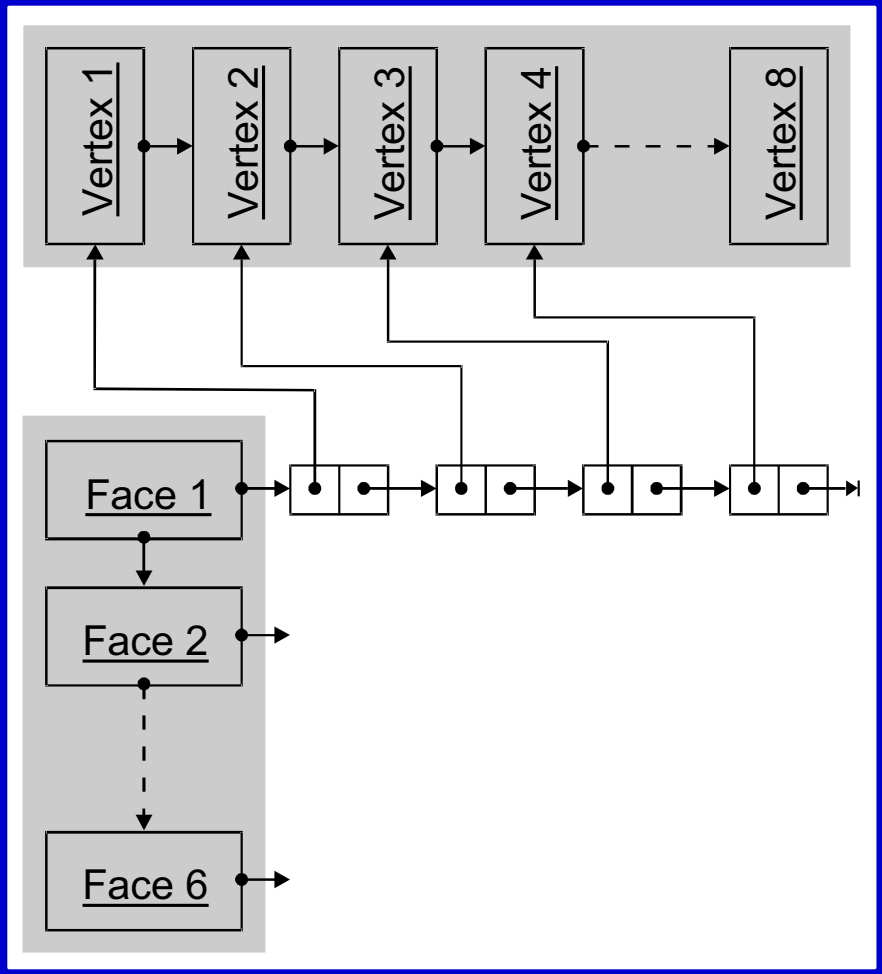
# Boundary Representation, B-rep

## Point List

- Element: 3D-coordinates
- Linked Lists

## Face Lists

- Element: Index List to the 3D-points
- Linked Lists



# B-rep Advantages/Disadvantages

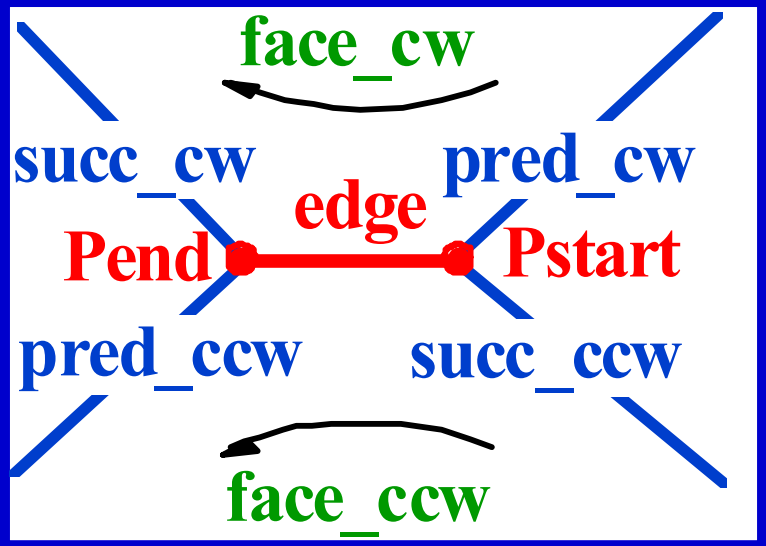
- **Pros**
  - Simple transformations
  - General representation
  - Supported by many graphics libraries
- **Cons, drawbacks**
  - Higher memory requirements
  - Combinations necessary and non-robust
  - Curved objects – approximation

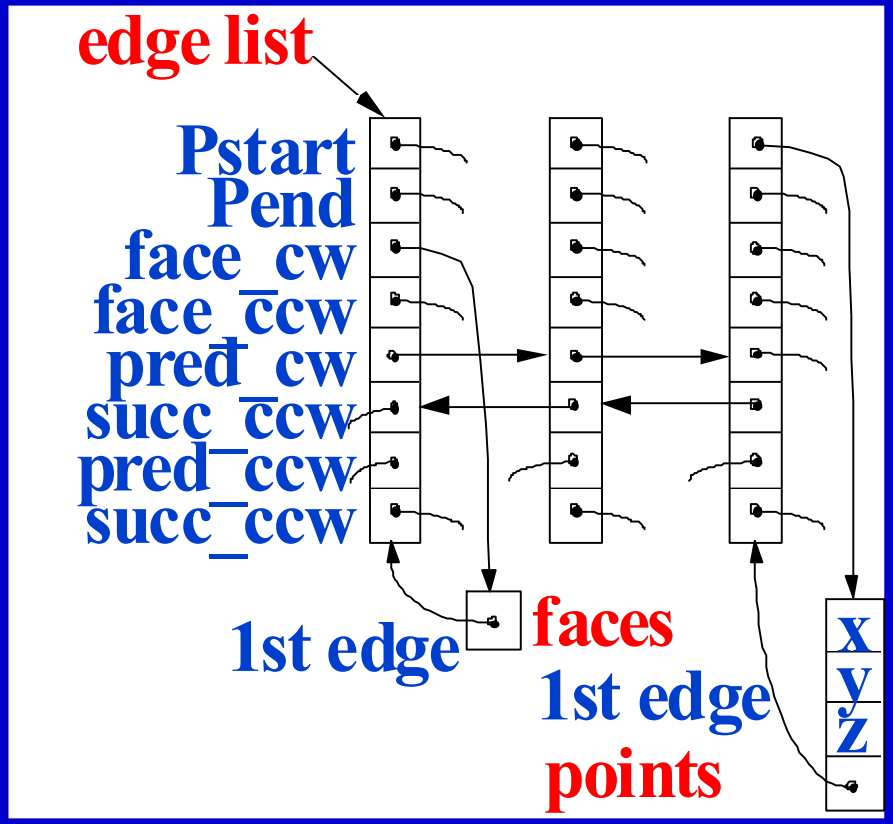
# IndexedFaceSet (VRML97)

```
Shape {
  geometry IndexedFaceSet {
    coord Coordinate {
      point [
        x0 y0 z0,     # vertex 0
      ]
    }
    coordIndex [
      0, 1, 4, 2, -1, # face 0
    ]
  }
}
```

# Winged Edge Data Structure

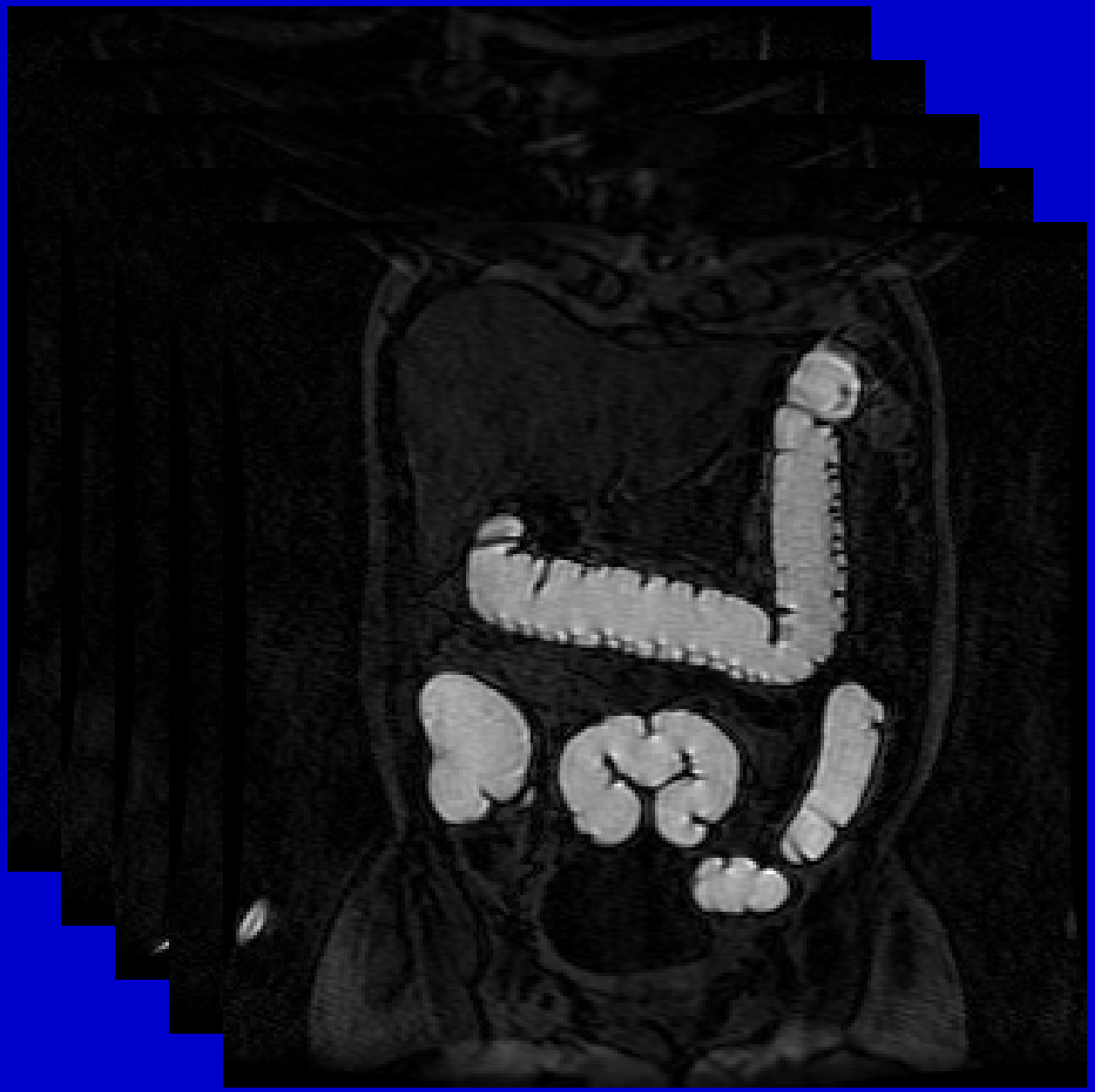
- Alternative to hierarchic B-Rep.
- Central element is the edge:



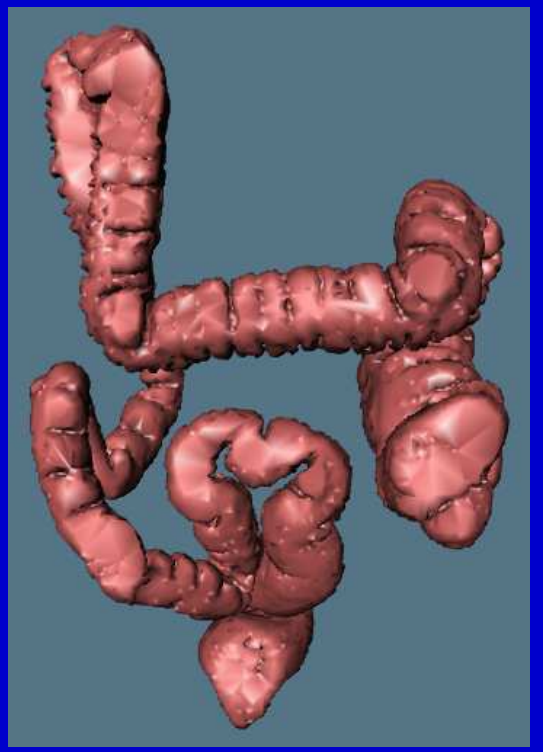



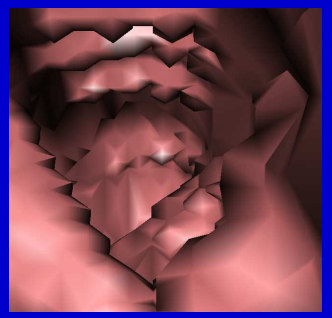
# Spatial-Occupancy Enumeration

- Dividing into the regular grid
- Elements – equally sized cubes (or quaders)
- Each element has density (Hounsfield)
- Applications:
  - Scientific Visualization
  - Medical data sets (CT, MR, US, SPECT)



# Medical VolVis





# Voxels Pros/Cons

- **Advantages**
  - Combination simple (in the same grid)
  - Direct utilisation of medical data
  - ...
- **Disadvantages**
  - Extreme memory consumption
  - Representation necessary
  - Curved object has to be approximated
  - ...

# BSP Trees

- Subdivisions in halfspaces in/out - Test
- Generating the binary tree sequentially - from the start polygon until all polygons processed
- Nodes represent polygons
- Node attributes:
  - Plane equation
  - Normal vector

# Generating BSP-tree

1. For convex objects trivial - lists
2. B-Rep traversal:
   - Polygon search, with least points of intersection
   - Subdivide the face list: 2 parts (in/out)
   - Insert polygon into tree, recursive subtrees processing

# Visibility

## Recursion:

- Eye point in positive halfspace (A+)
  - Output of A-
  - Draw A
  - Output of A+
- Eye point in negative halfspace (A-)
  - Output of A+
  - Draw A
  - Output of A-

## Hint: Backface-Culling possible

# BSP-tree Pros/Cons

- **Pros**
  - Easy transformable
  - Fast rendering with correct visibility
  - Universal like B-Rep
- **Cons**
  - Higher memory consumption
  - Curved objects to be approximated

# BSP FAQs

- ...
- 7.HOW DO YOU BUILD A BSP TREE?
- 8.HOW DO YOU PARTITION A POLYGON WITH A PLANE?
- 9.HOW DO YOU REMOVE HIDDEN SURFACES WITH A BSP TREE?
- 10.HOW DO YOU COMPUTE ANALYTIC VISIBILITY WITH A BSP TREE?
- 11.HOW DO YOU ACCELERATE RAY TRACING WITH A BSP TREE?
- 12.HOW DO YOU PERFORM BOOLEAN OPERATIONS ON POLYTOPES WITH A BSP TREE?
- 13.HOW DO YOU PERFORM COLLISION DETECTION WITH A BSP TREE?
- 14.HOW DO YOU HANDLE DYNAMIC SCENES WITH A BSP TREE?
- 15.HOW DO YOU COMPUTE SHADOWS WITH A BSP TREE?
- 16.HOW DO YOU EXTRACT CONNECTIVITY INFORMATION FROM BSP TREES?
- 17.HOW ARE BSP TREES USEFUL FOR ROBOT MOTION PLANNING?
- 18.HOW ARE BSP TREES USED IN DOOM?
- 19.HOW CAN YOU MAKE A BSP TREE MORE ROBUST?
- 20.HOW EFFICIENT IS A BSP TREE?
- 21.HOW CAN YOU MAKE A BSP TREE MORE EFFICIENT?
- 22.HOW CAN YOU AVOID RECURSION?
- 23.WHAT IS THE HISTORY OF BSP TREES?

- 24.WHERE CAN YOU FIND SAMPLE CODE AND RELATED ONLINE RESOURCES?
  - ftp://ftp.sgi.com/other/bspfaq/faq/bspfaq.html#6.txt

# Octrees

- Iterative space subdivision
- Nodes empty, full, or partial: E, F or P
- Subdivide until the approximation satisfies required quality 512*512*512
- Dividing planes not object dependent (differs from BSP tree!)
- Analogy in 2D: „Quad Tree“

# Visibility and Octrees

## Observations:

- Topologic structure from data strukture
- maximum 3 sides of a cube visible

## Algorithm:

- Identify the rendering order for sub-cubes for given eye point (camera position)
- Recursive rendering of octree with identified sequence

## Sequence:

1. Distant octants ($O_E$)
2. Each 3 octants having the face with $O_E$ parts
3. Each 3 octants having the face with $O_V$ parts
4. Front-side octants ($O_V$)

## Rendering:

– Sequence from all nodes usage

– Sequence depends only from observer direction

– There are  8 important sequences

- **Advantages**
  - Simple combination-operators
  - Fast rendering
  - Spatial queries/search very fast
- **Disadvantages**
  - Higher memory consumption for well approximated objects
  - Limited transformability
  - Approximation (criterion)

# Extended Octrees

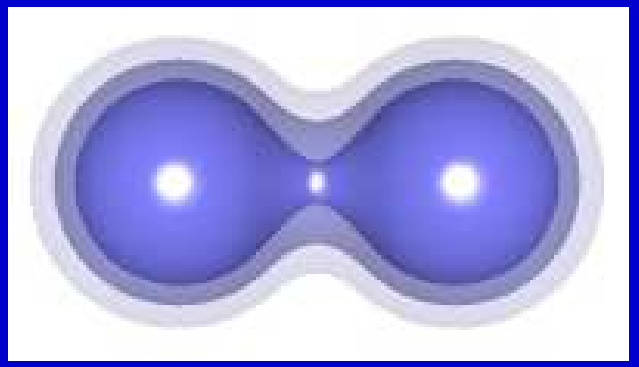
## Extension by new node types:

- Surfaces
- Edges
- Points, vertices

## Generation:

- B-Rep production
- Lists in 8 parts subdivision
- Processing of additional node types

# Volume Graphics



**Gallery at :**

- http://vg.swan.ac.uk/gallery/index.html

**Book:**

– Chen et al. London: February 2000

**vlib - GNU library at :**

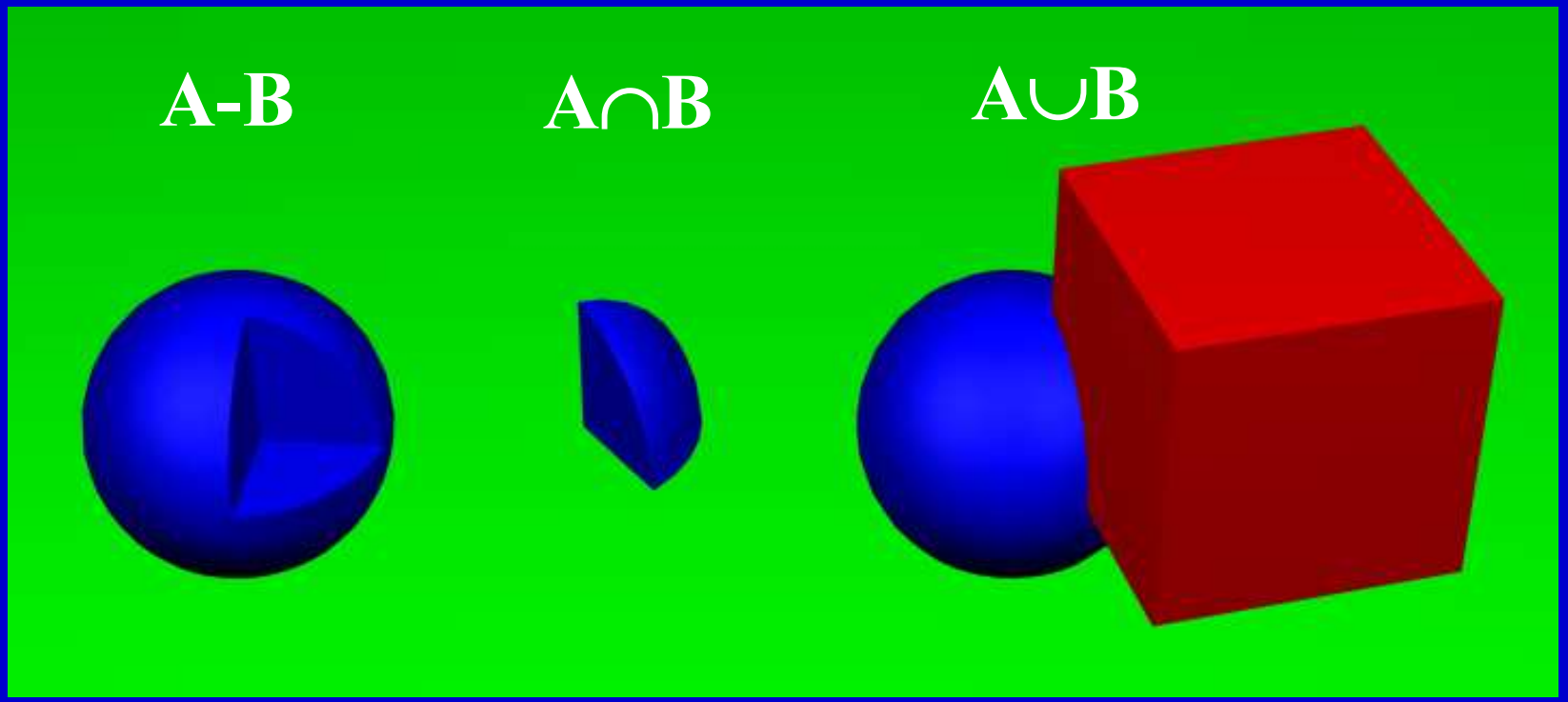
- http://vg.swan.ac.uk/

# Constructive Solid Geometry

- Composition of primitives
- Primitives: sphere, cone, cube, cylinder, ...
- Operations: +, -, ∩, ∪, ...
- Primitives in the leafs and operations build the rest nodes of the CSG-tree

# Boolean Operations

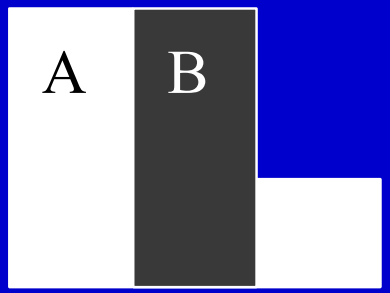
## Using 3 operators enables for all possible combinations - not uniquely



# Regular Boolean Operations

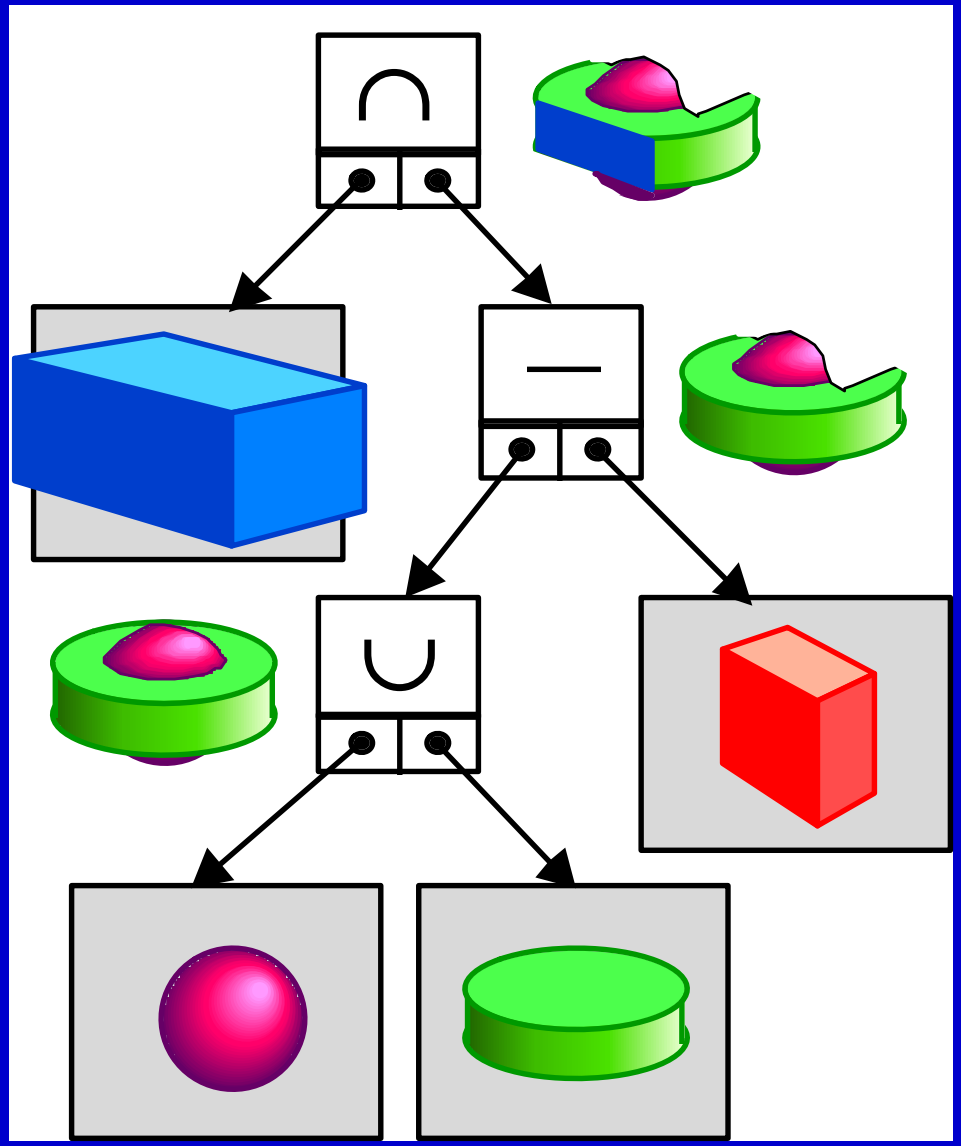
## Basic-Operations:

- Union $\cup^*$
- Intersection $\cap^*$
- Subtraction $-^*$




$A \cap B$

$A \cap^* B$



# CSG Rendering

- **Rendering Algorithms**
  - ray casting (tracing)
  - extended depth buffer algorithm
- **Evaluation/Conversion Algorithms**
  - boundary representation
  - octree
  - constructive cubes

# CSG Pros/Cons

- Pros
  - Low memory consumption
  - Simple combining
  - Exact representation of complex surfaces
  - 12 Mantylas operators and his proof
- Contras
  - Slow and difficult processing
- ?
  - Not unique representation

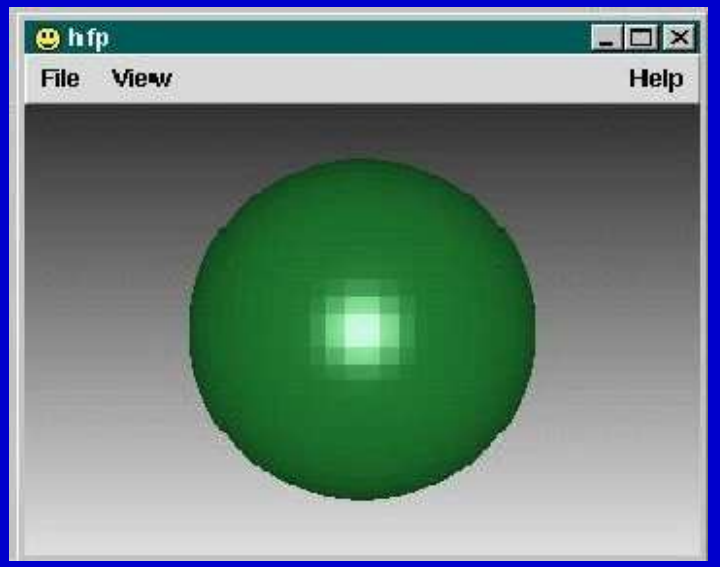
# CSG Generalisation - F-rep

- *HyperFun is a simple geometric modeling language. It is intended for modeling geometric objects described in the form:*
- *F(x1, x2, x3, ..., xn) >= 0*
- *This language is applicable to modeling algebraic and skeleton-based "implicit" surfaces, convolution surfaces, distance-based models, voxel objects, constructive solids, and more general F-rep objects.*

- *The model in HyperFun is interpreted by the modeling and visualization software tools.*

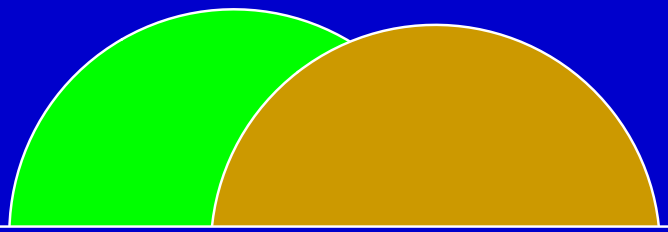
# F-rep sphere

```
my_model(x[3], a[1])
{
   my_model = 5^2 - (x[1]^2 + x[2]^2 + x[3]^2);
}
```



# CSG Generalisation - F-rep

- *Set theoretic operations are expressed as functions!!!*
- *A* *B*



**Union(A,B) = max(A,B)**

**Intersection(A,B) = min(A,B)**

# Spreadsheet Rendering & Inbetween Statues in 2D Time

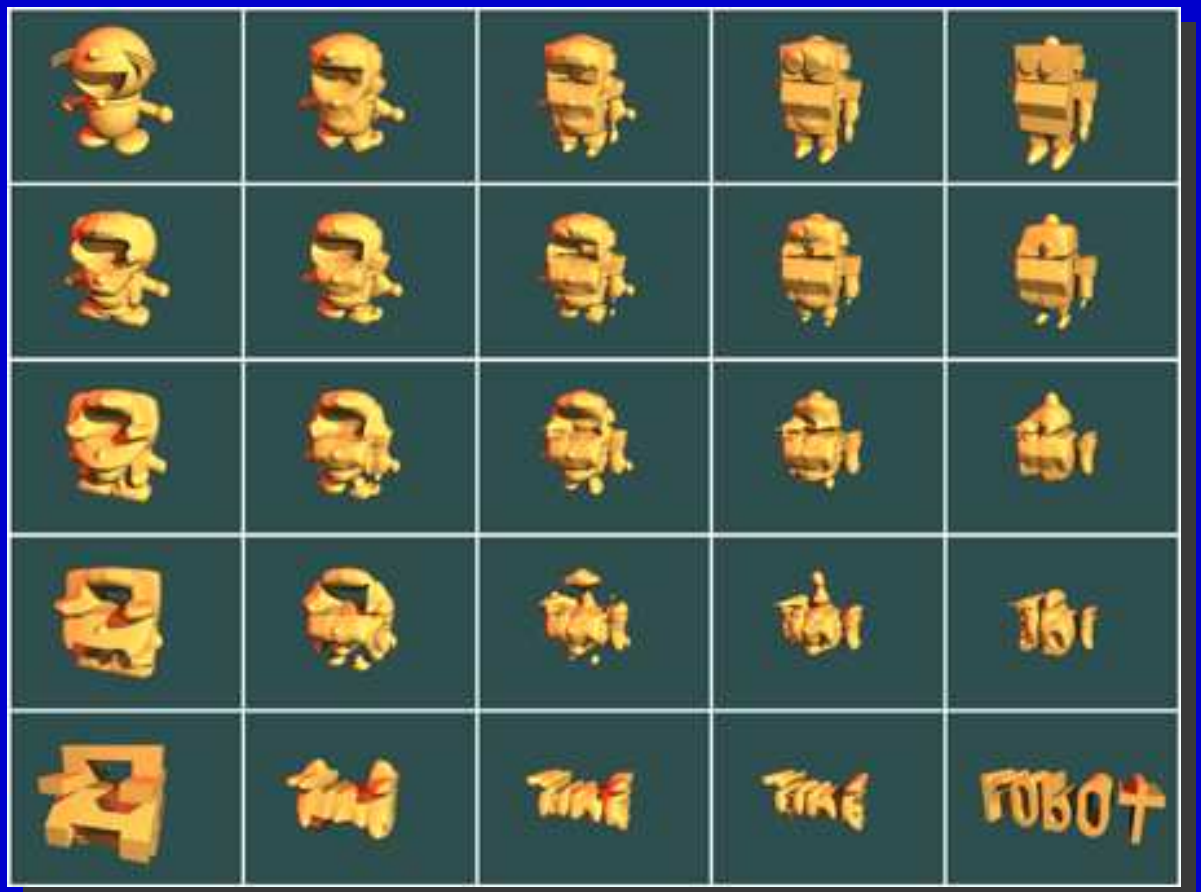
- spreadsheet animation images "Homotopic Fun in 5D space" are by E. Fausett, A. Pasko, V. Adzhiev, see http://wwwcis.k.hosei.ac.jp/~F-rep/Homotopic.html

- inbetween statues in the triangle are by M. Kazakov, A. Pasko, V. Adzhiev

F-rep completization of CG theory & practice

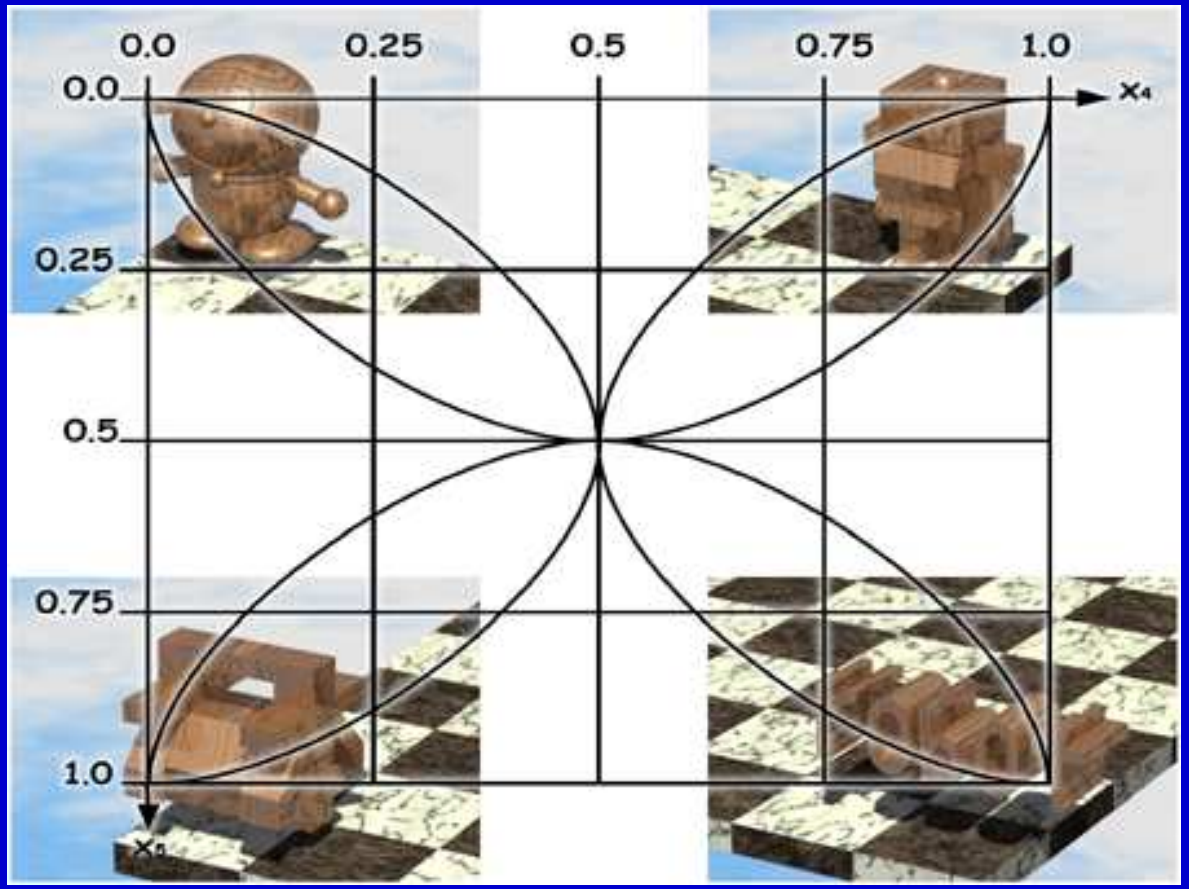
# Spreadsheet Rendering

## by Alexander PASKO, www.hyperfun.org



# Animation Path in $t_1t_2$ Plane

## by Alexander PASKO, www.hyperfun.org

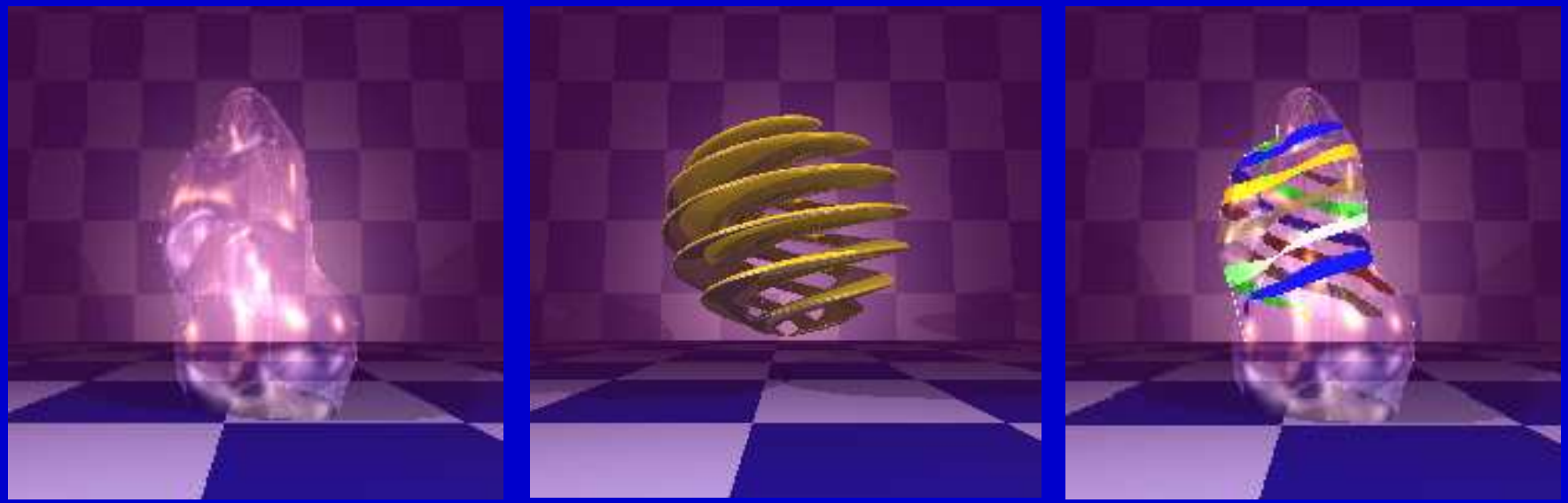


# F-rep Hypertextures

- F-rep
- Unified language for objects and attributes, Schlick et al. 2001
- This historical value result unifies different dialects of mathematical language for objects, relationships, and operations AND ATTRIBUTES...
- ... DENSITY, TOO (SIMULATION)
- WWW.HYPERFUN.ORG

# Constructive Hypervolume Texturing

## by Alexander PASKO, www.hyperfun.org

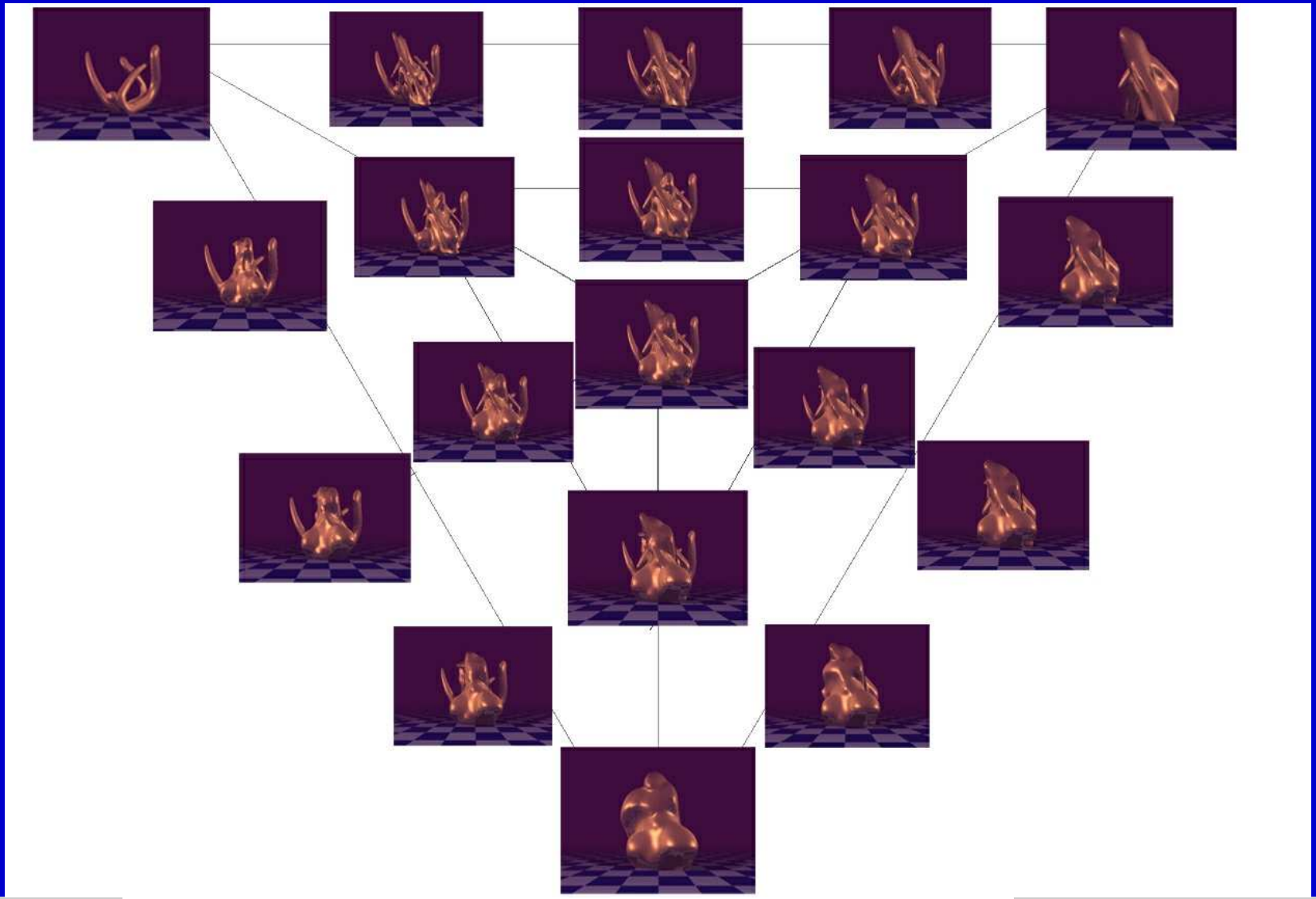


*implemented by B. Schmitt*

# F-rep Importance

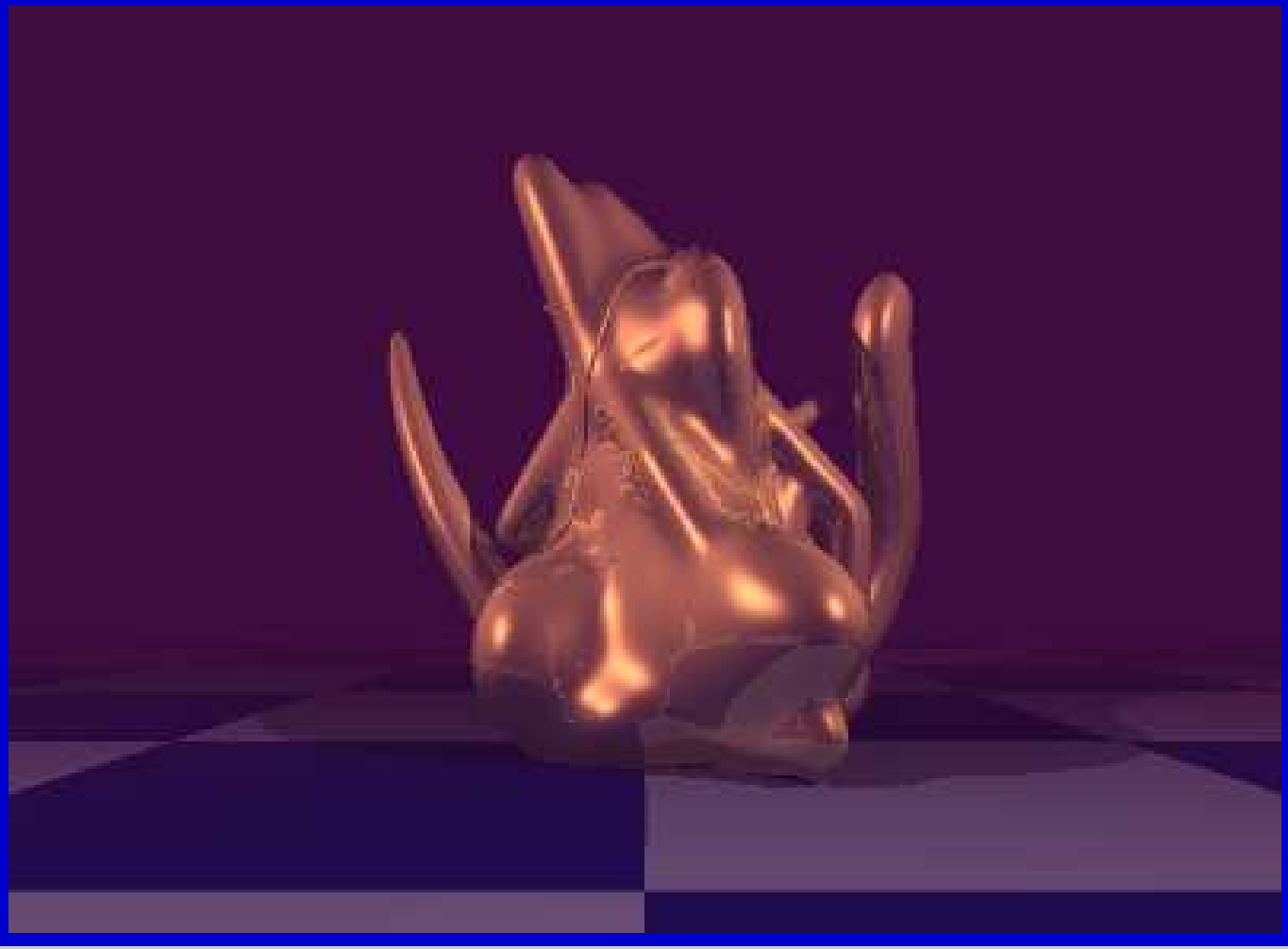
- F-rep
- Unified language for both CSG tree (or scene graph) and subsequent primitives
- This historical value result unifies different dialects of mathematical language for objects, relationships, and operations
- Modeling "esperanto"

*by Alexander PASKO, www.hyperfun.org*



# The Shape in the Triangle Center

**by Alexander PASKO, www.hyperfun.org**



# F-rep Summary

- F-rep (function representation)
- Implicit surfaces using the functional form of set operations (union, intersection, difference)
- Discovery by Rvachev [Rvac63] thus enables for unified language for both CSG tree (or scene graph) and subsequent primitives
- Problems: costly, interpolation

# Criteria for Data Structure

**Modeling:**

- Complexity of Representation
- Transformation / Combination
- Interactivity
- Reusability...

**Rendering:**

- Precision of given representation
- Memory Requirements, Real-time...

# Easy Modeling Methods

# Easy Modeling Methods

**Task:**

Create the Object Description for later Processing within the Rendering- and Output-Modul.

**Generated via:**

- User Interaction
- Automatically (eg „Object-Scanner“, range images, ...)

- **Elementary Objects**
  - Primitives, regular polyhedra, ...
  - Sweeps
  - Free-form patches
  - (Super-)Quadrics
  - Terrain (DTM, DEM)
  - Fractal Mountains
  - Soft Objects
  - Particle Systems
  - Natural Phenomena...
- **Transformations**
  - linear ones
  - twist, blending ... (Verbiegeoperationen)
  - local operations
- **Combining methods**
  - Boolean Operations with Elementary Objects (CSG)
  - F-rep
  - (Solid Modeler UI)

# Sweeps

## Idea:

Move a 2D-Object (Contour) in the space. All enclosed points generate the object swept.

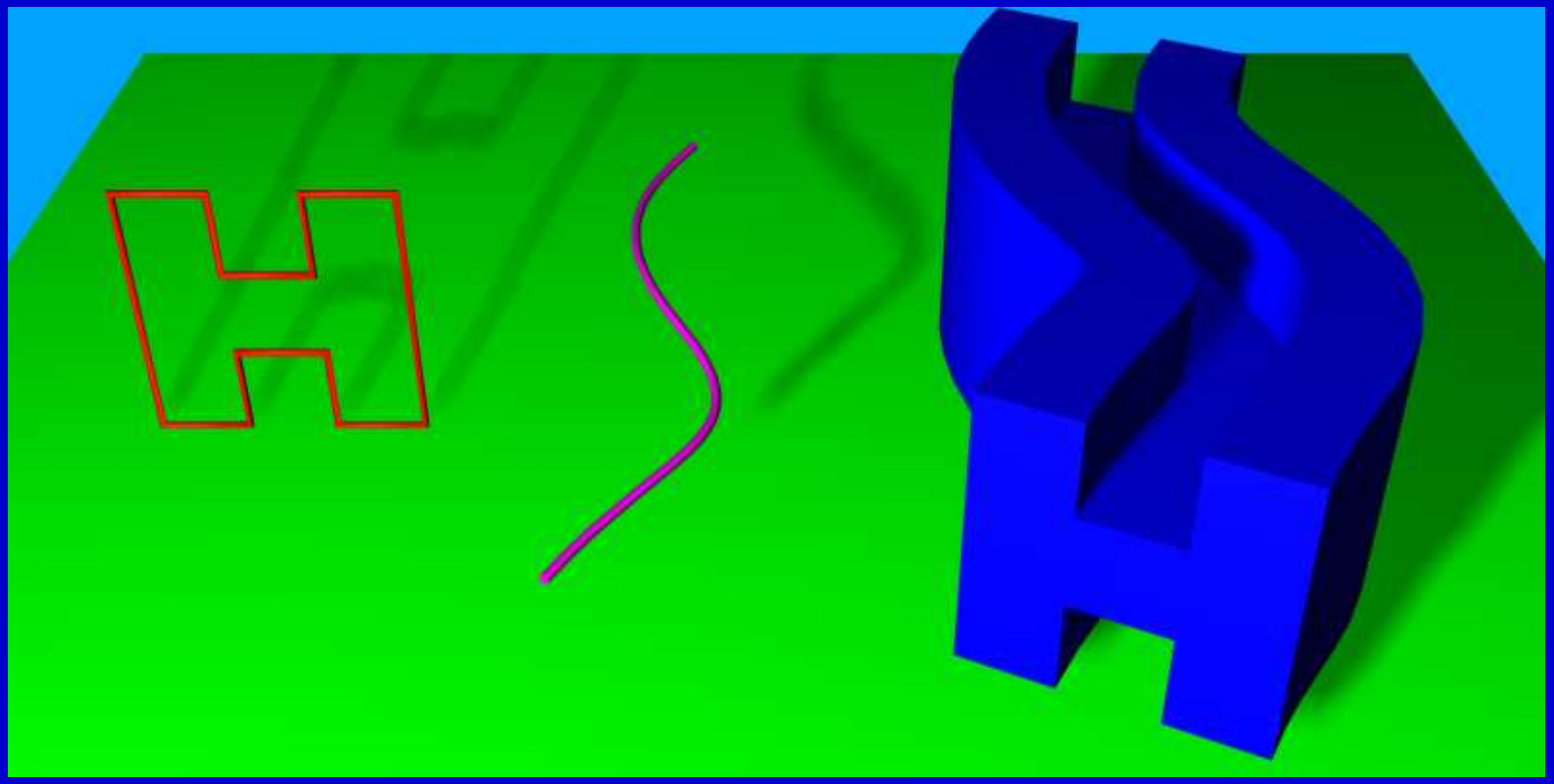
## Forms:

- translational sweep
- rotational sweep
- conical sweep
- sphere sweep
- general cylinder, …
- NOTE: Parametrisation

# Translational Sweep

## Method:

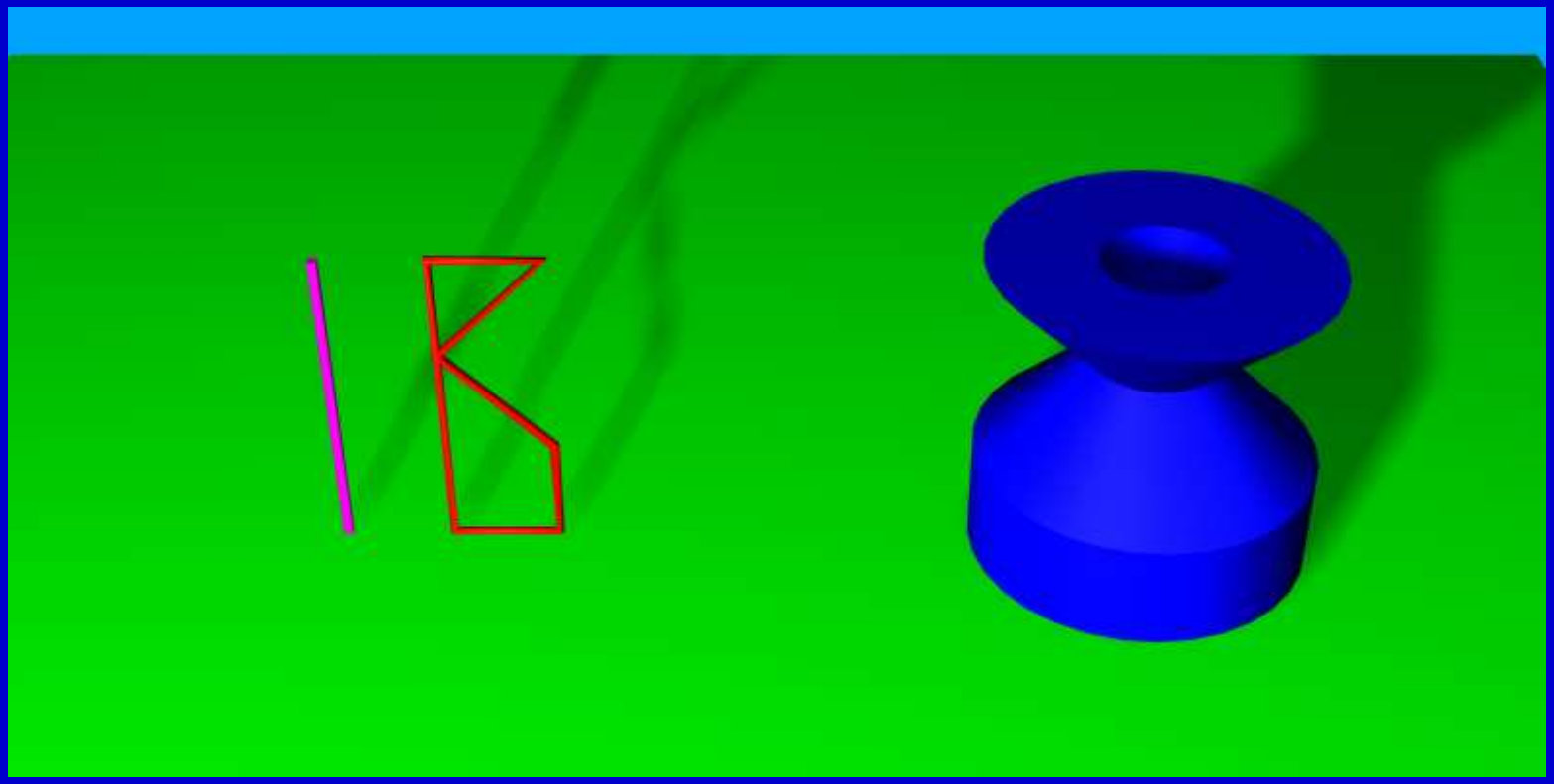
Define the object with the contour (2D) and the path.



# Rotational Sweep

## Method:

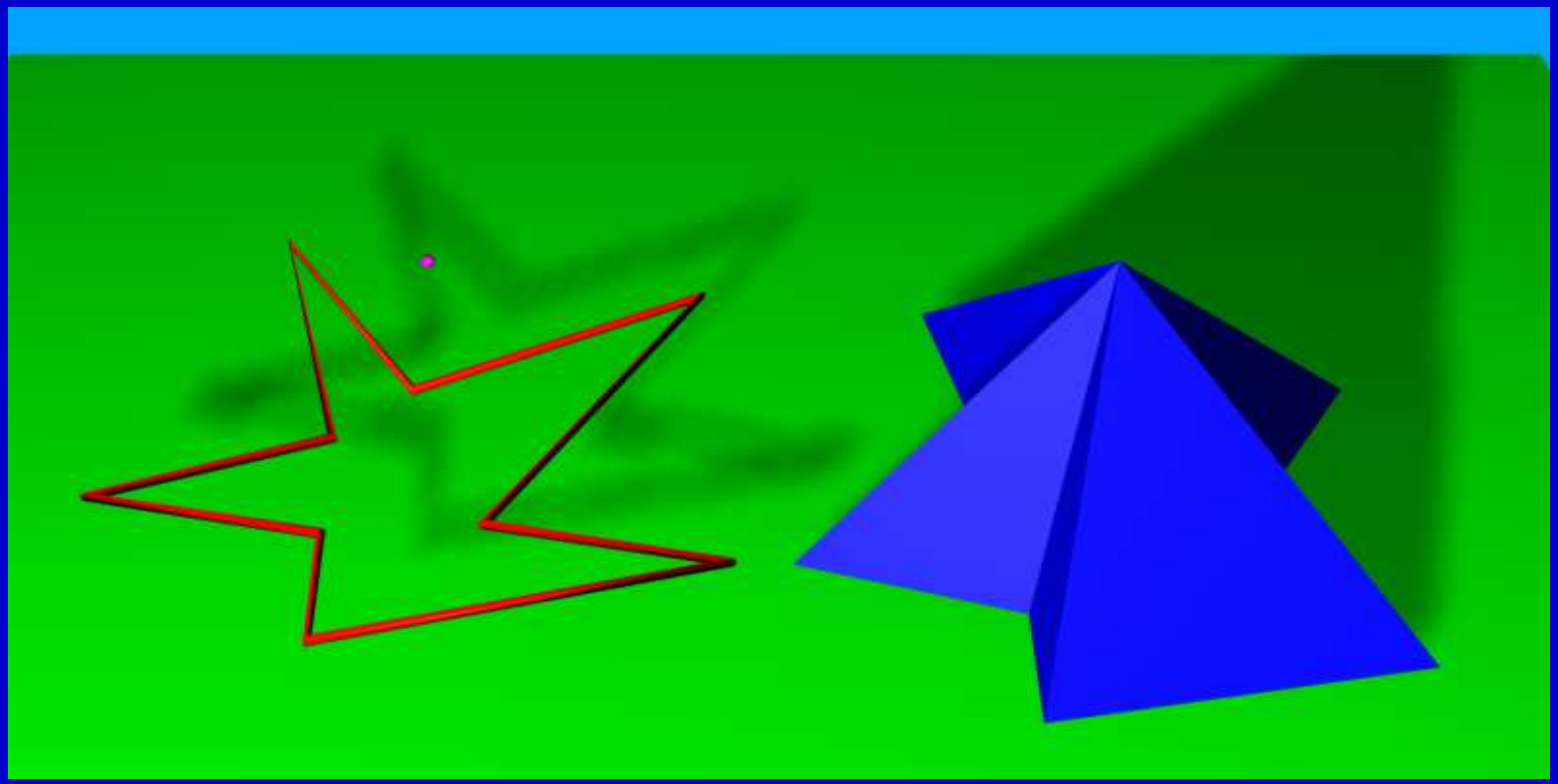
*Define the object by rotating of the contour (2D) with the arbitrary axis.*



# Conical Sweep

## Method:

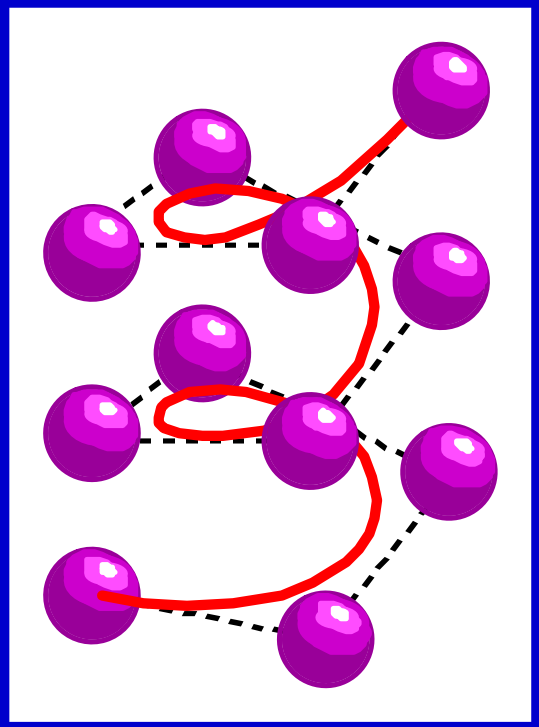
Define the object by the contour (2D) and a 3D-point (top of the pyramide).

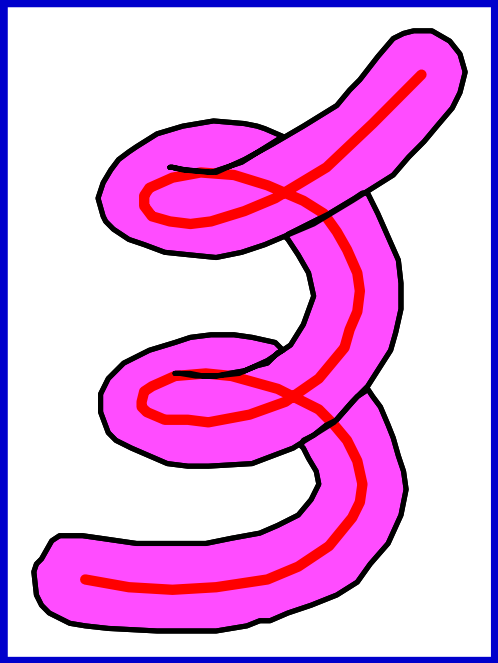


# Sphere Sweep

## Method:

*Define the object by the sphere with the varying radius and a path.*

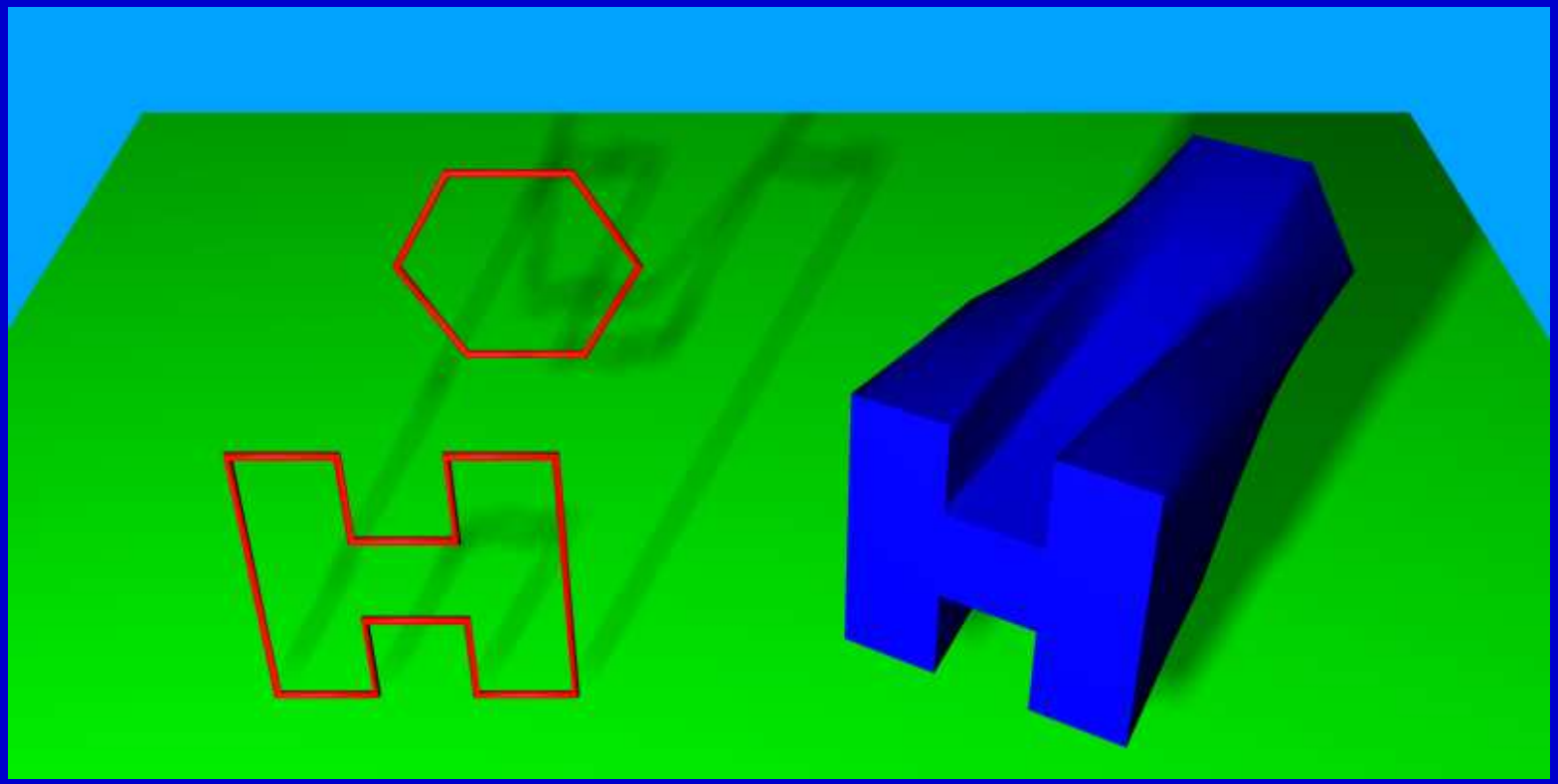




# General Cylinder

## Method:

*Define the object by the set of „control“-contours and by the path.*



# Quadrics

## Idea:

Quadrics are all objects, which is possible to describe using quadratic functions (polynomials).

## Definition:

– explicitly : $x^2 + y^2 + z^2 = r^2$

– parametric:

$$x = r \cos\alpha \cos\beta$$

$$y = r \cos\alpha \sin\beta$$

$$z = r \sin\alpha$$

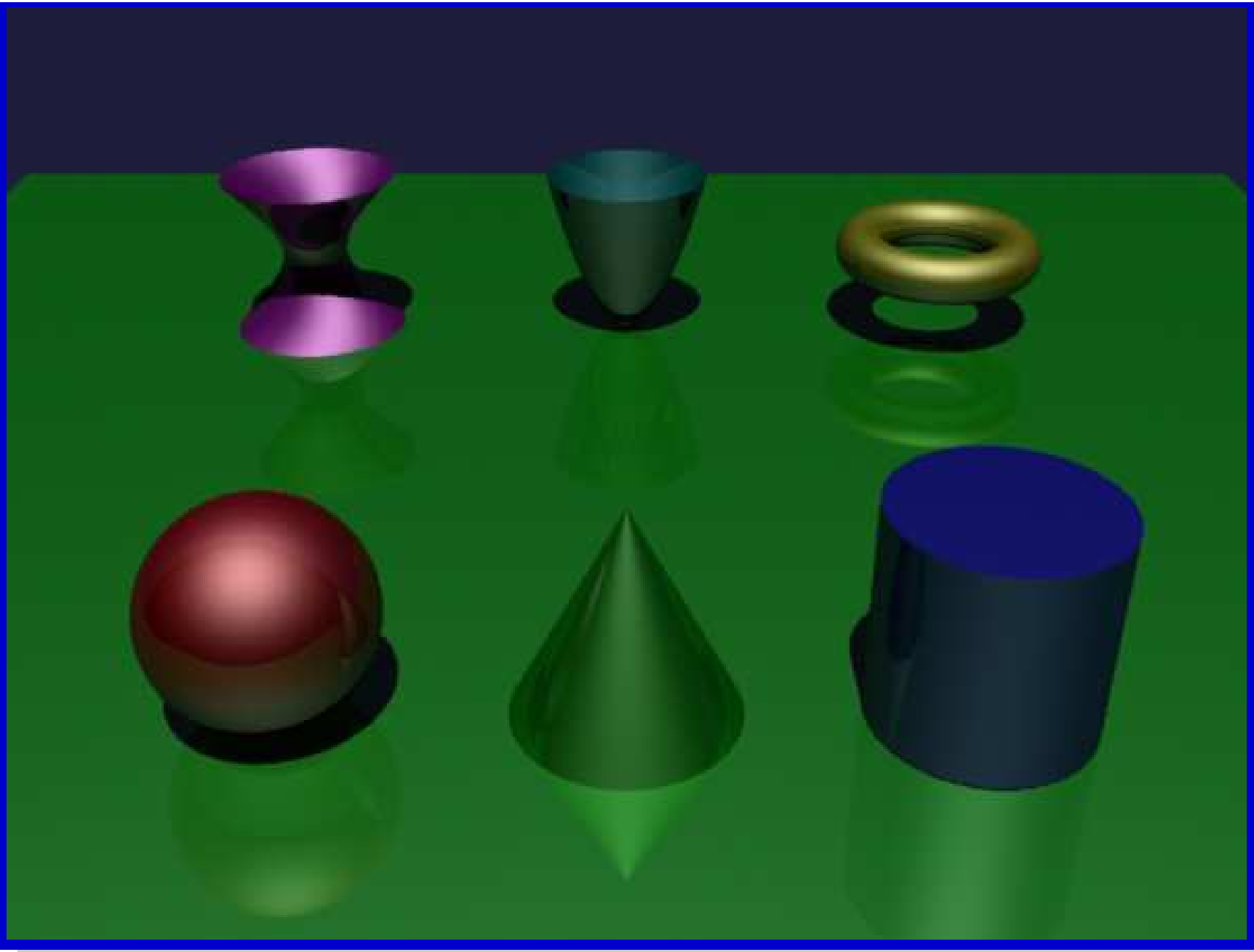
# Quadrics (examples)

- Sphere: $x^2 + y^2 + z^2 = r^2$
- Cylinder: $x^2 + y^2 = r^2 \quad 0 \le z \le height$

  $x^2 + y^2 \le r^2 \quad z = 0 \text{ and } z = height$
- Conic: $x^2 + y^2 = z^2 \quad 0 \le z \le height$

  $x^2 + y^2 \le z^2 \quad z = height$
- Torus: $\left(x^2 + y^2 + z^2 + R^2 - r^2\right)$

  $-4\left(x^2 + y^2\right) = 0$



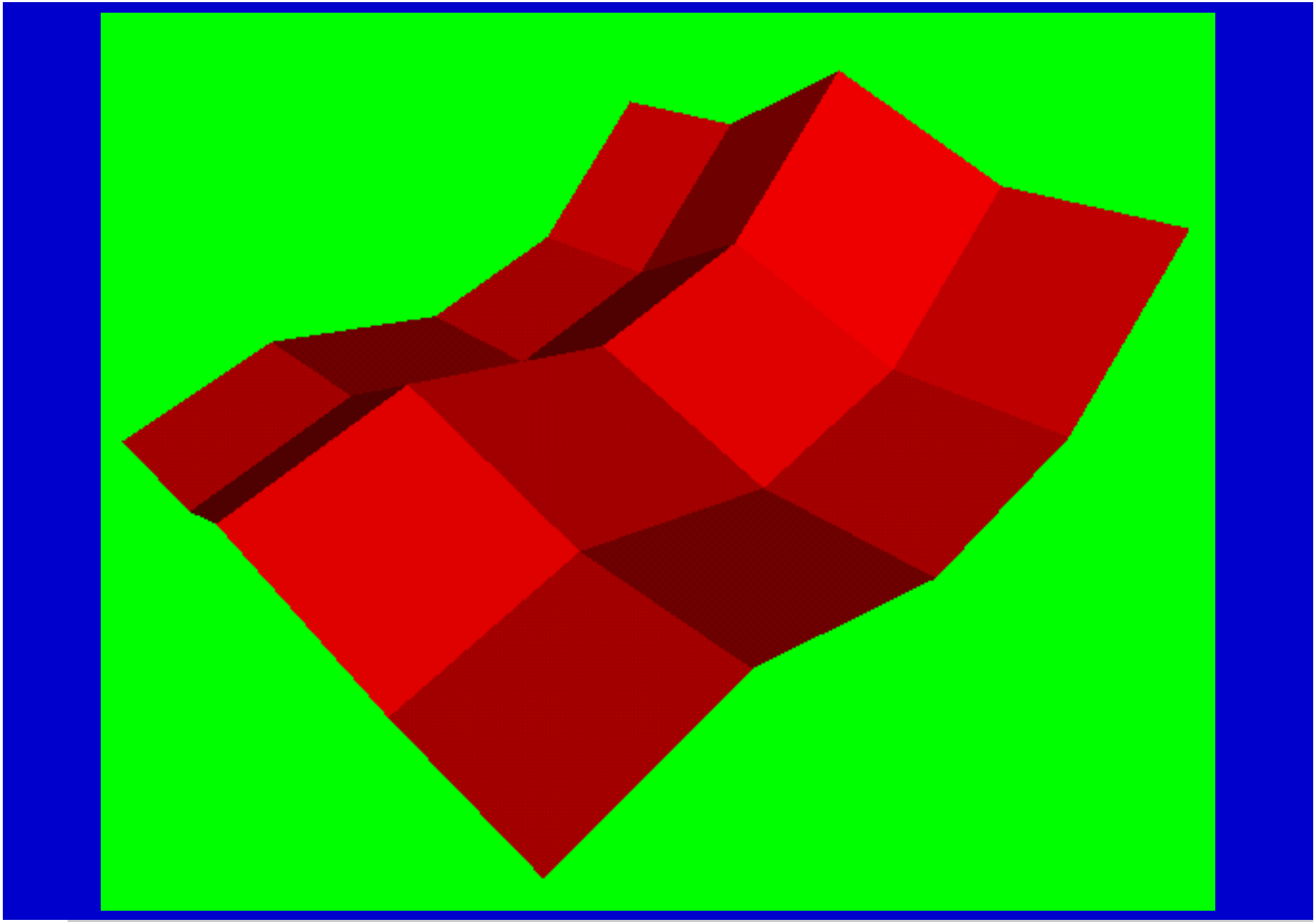
# Terrain

## Definition:

– Given the equidistant grid in the plane and in gridpoints the heigths (Z-coordinates). **DEM** (digital elevation model, field of heights).

## Extension:

– More properties given for the terrain surface (eg Color), the DEM enriched by texturing **DTM** (digital terrain model).



# Note on 2.5D Objects

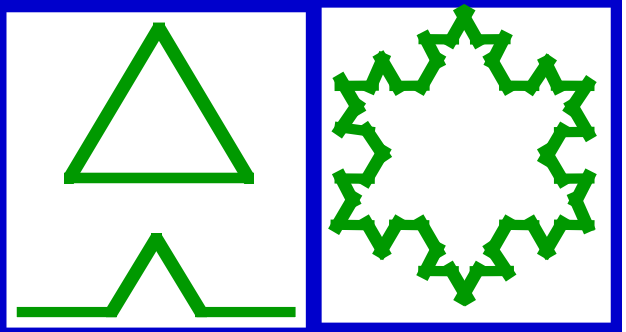
**Definition:**

– If the surface of terrain or sweeped solid can be adressed by 2 parameters, we speak about 2.5D objects.

**Question:**

– Which solids are 2.5D ones?

# Fractal Mountains



## Fractal - Koch's Curve:

- Initiator: Start with the polygon.
- Generator: Replace each line segment.

## Fractal Mountains:

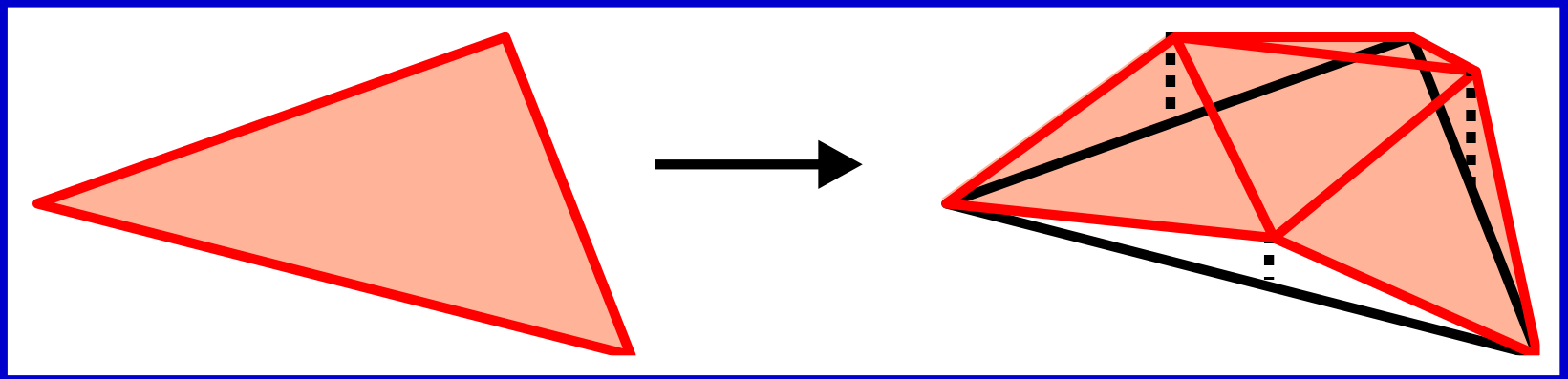
- Generator no more regular, but random one within the given borders Grenzen.
- Initiator: 1 triangle (or 2)
- Generator: each egde divide by a randomly generated point.

# Fractal Mountains (algorithm)

**Each Step:**

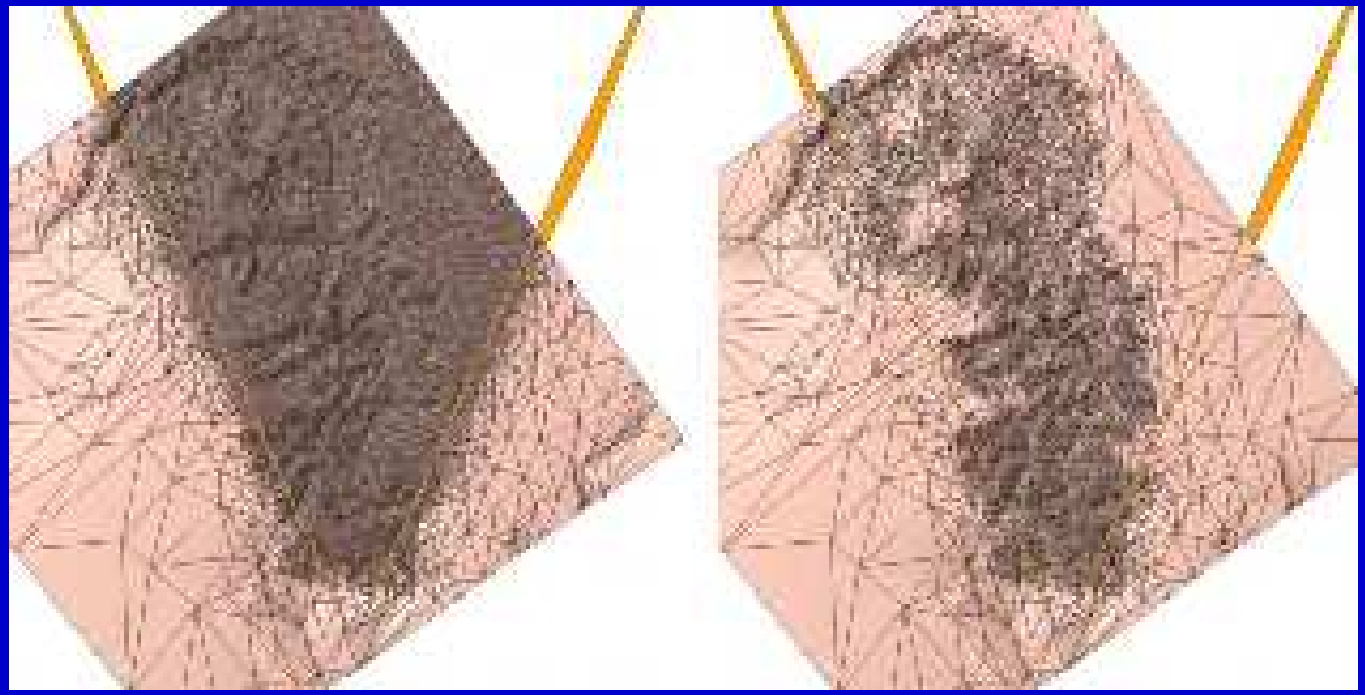
from 1 triangle generate 4 new ones.

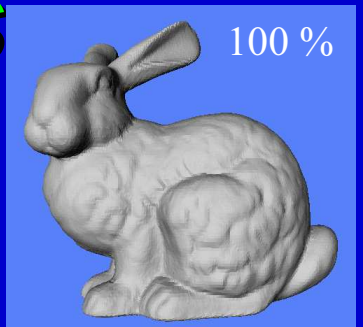
**Subdivide** until the Quality suffices.

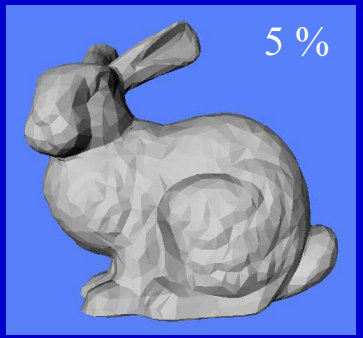


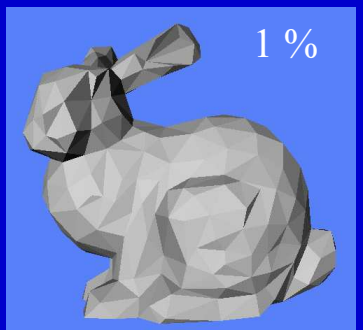
# Multiresolution, Area Subdivision

## Oscar Winning Animations

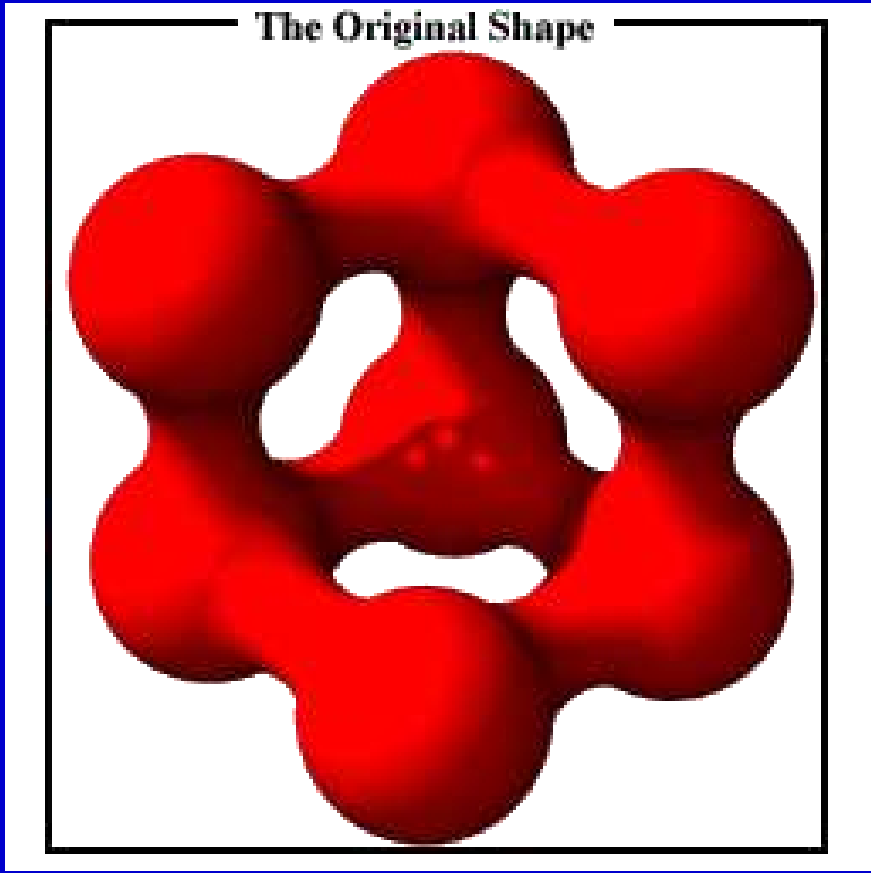


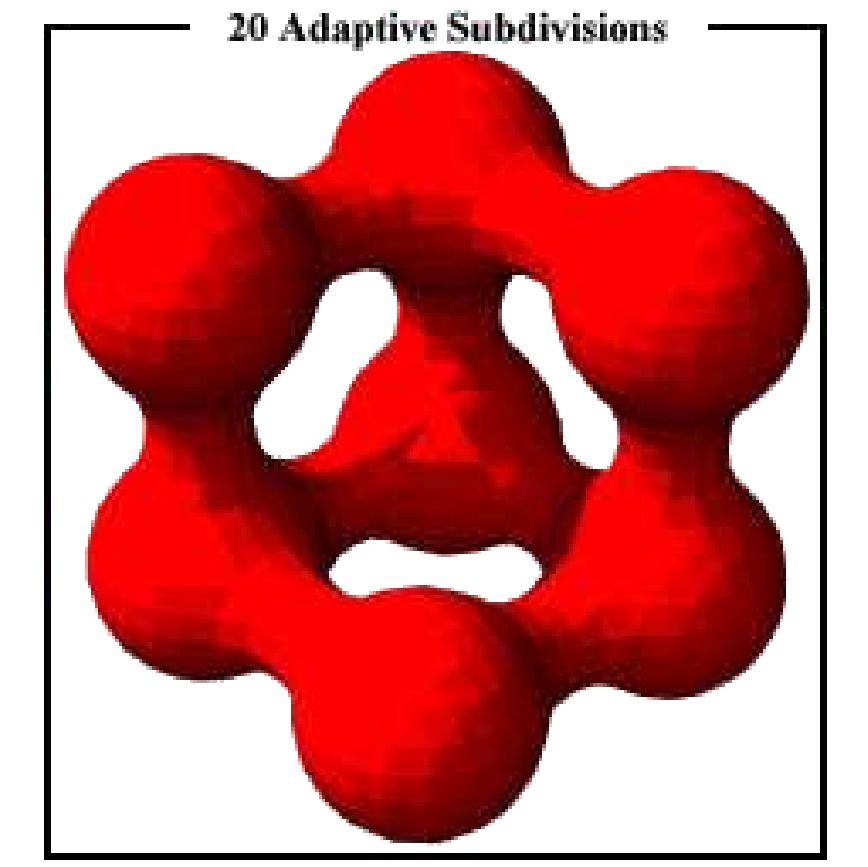


# Example

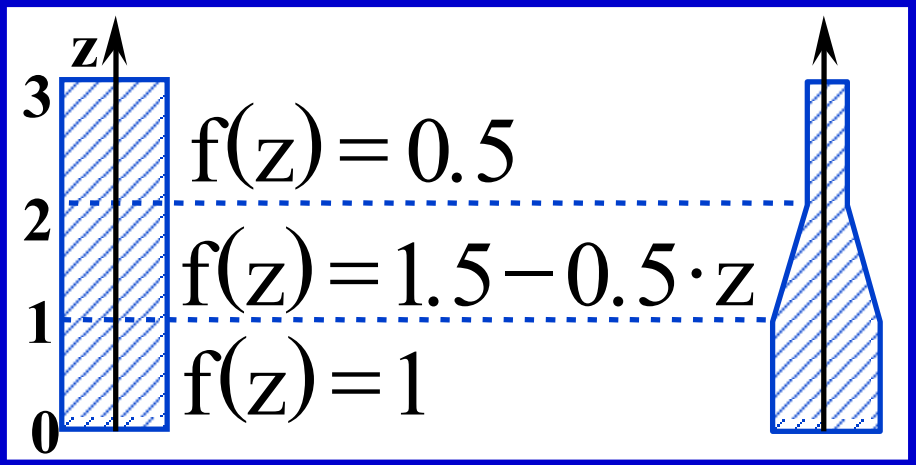


# Tapering

**Position dependent functions work like the scaling factors.**

$$\vec{X} = \begin{pmatrix} f_1(\vec{x}) & 0 & 0 \\ 0 & f_2(\vec{x}) & 0 \\ 0 & 0 & f_3(\vec{x}) \end{pmatrix} \cdot \vec{x}$$

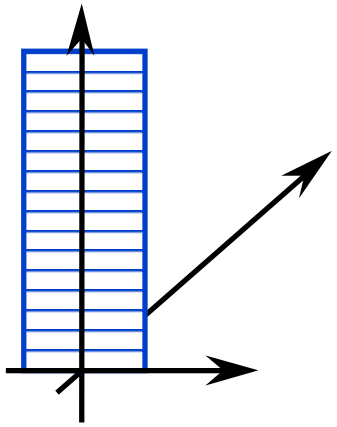


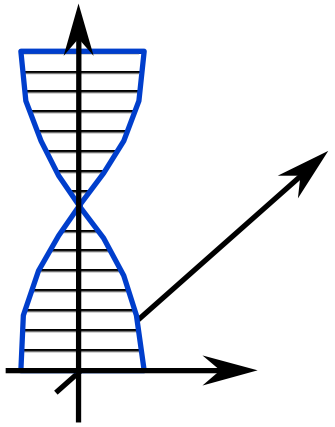
# Twist

**Position dependent functions work for twisting of the object, too.**

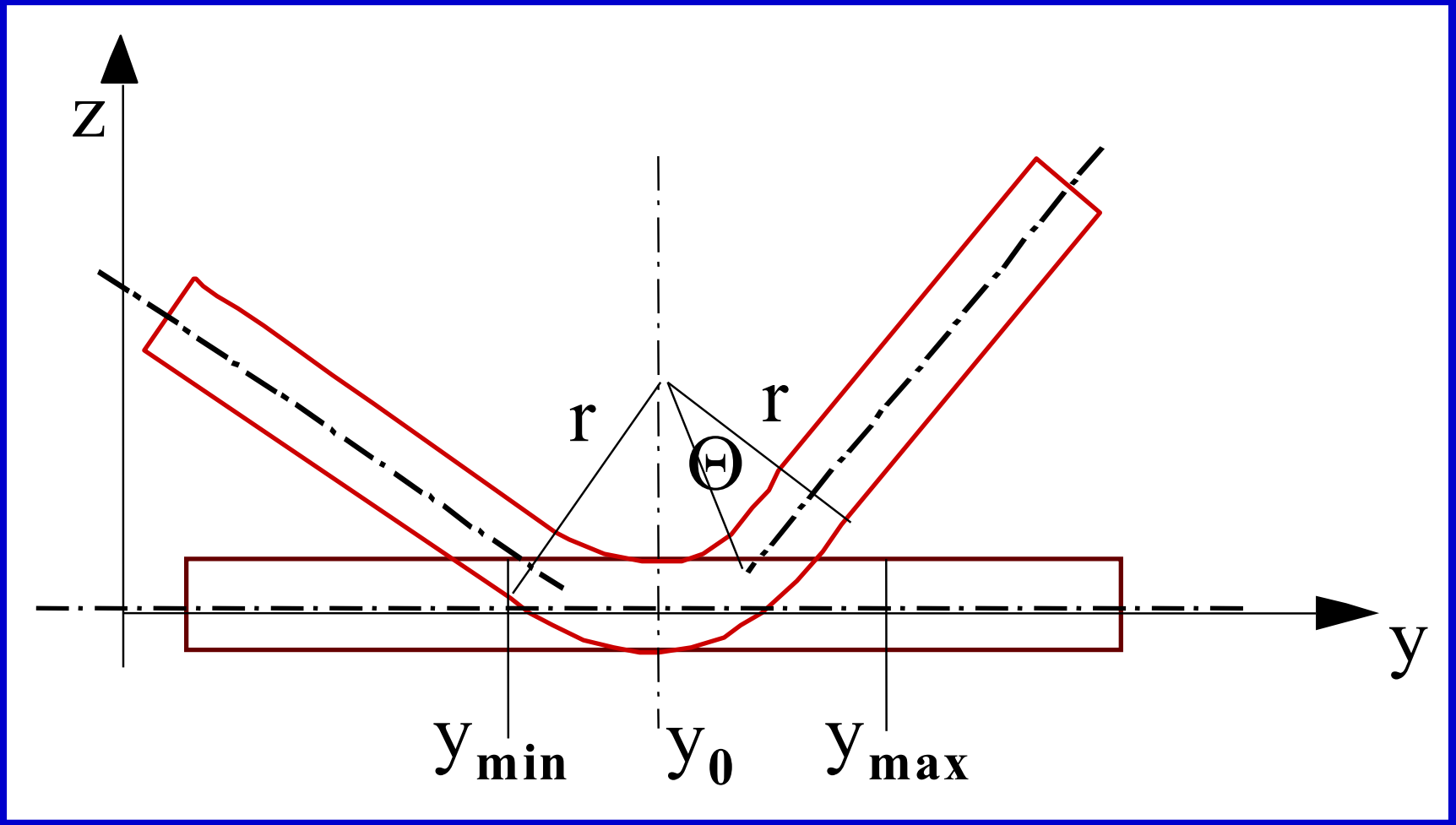
$$\vec{X} = \begin{pmatrix} \cos f(\vec{x}) & -\sin f(\vec{x}) & 0 \\ \sin f(\vec{x}) & \cos f(\vec{x}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \vec{x}$$

$$\mathrm{f}(\vec{\mathrm{x}}) = \begin{pmatrix} 1 \\ 1 \\ \mathrm{z} \end{pmatrix}$$





# Bend



# Note on valid solids

***Definition:*** *The valid solid surface is locally topological equivalent with the Euclidean plane.*

***Question:***

- *Which solids are not valid?*

***Hints:***

- *Sphere is equivalent with the plane. Zero thickness is invalid.*

# Solid Modeling Summary

- Elementary Objects
  - Sweeps, Quadrics,Terrain, Soft Objects, ...
- Transformations
- Combining Objects (CSG, F-rep)
- Volume Representation and others
- Rendering Summary (Polygonal Case)
- Future work: transparent objects, bumpy surfaces, textures, global illumination, animation...

# Data Formats in CG

- **Normalised CAD data formats**
  - IGES
  - AutoCAD DXF, ...
- **Data formats for image/document exchange**
  - CGM, PNG, Postscript, ...
  - HyperODA, SGML: HTML, XML, ...
- **Data formats of 3D graphics**
  - SEDRIS, VRML - www.sedris.org, www.w3c.org
  - RenderMan Interface
  - MGF (Radiance), POV (POV-Ray), ...
  - ISO, EUN, ANSI, MIME... http://www.w3.org/Amaya/
  - .avi, .qt, .mpg, … DivX…

# Contents Overview

1. Introduction and Motivation - OK
2. Data Structures & Data Formats – OK
3. Parametric Curves & Surfaces
4. Color
5. Light-Material Interaction
   - Local Illumination Models
   - Antialiasing
   - Shadow Generation…

# 3. Free-form Curves/Surfaces

- *2001* - ***Final Fantasy*** *( The Spirits Within )*
  - *all characters animated, photorealism,* ***hair*** *for millions dollars*

# Content

## 3. Parametric representation of curves and surfaces

- Free-form curves
  - Bézier-curves
  - Rational Bézier-curves
  - B-Splines
  - NURBS - industrial standard, Maya...
- Free-form surfaces

# Motivation

- **Construction, CAGD, CAD/CAM**
  - Modeling of ship hulls, terminology
  - Design of cars and airplanes
- **Computer Graphics**
  - Simple modeling of smooth surfaces and solids
  - Definition of motion trajectories for animated objects

# Three forms of expression

- ***Analytic*** $y = \mathrm{sqrt}(r^2-x^2)$
- ***Implicit*** $x^2 + y^2 = r^2$
- ***Parametric*** $x = \cos t;\ y = \sin t;\ t \in \langle 0, 2\pi \rangle$
  ***travel along the curve***
  - *continuity - geometric G, parametric C*

# Parametric Blending

- Numbers, a, b, 0.5*(a + b)
- Points A, B, 0.3*A + 0.7*B
- Rotations, 4-tuples, quaternions
- Curve construction as weighted sum of points
- Surfaces
- Images… brightness, contrast, saturation, sharpening… image analogies, SIGGRAPH 2001
- … Morphing, Caricatures … state spaces

# PG.NetGraphics.sk and other applets

- Parabola
- Cubic Bézier curve
- B-spline (basis spline)
- Surface editing

- Free, © Juraj STUGEL

# Parametrically rep. curves

- Euclidean plane/space $E_2$, $E_3$ {O, x, y, z}
- Parametric representation of a curve in $E_3$ :

$$\vec{p}(t) = \begin{pmatrix} x(t) & y(t) & z(t) \end{pmatrix}^T \qquad \dot{\vec{p}}(t) \neq \vec{0}$$

Tangenta has direction vector: $\dot{\vec{p}}(t_0)$

- Curvature:

$$\kappa(t_0) = \frac{\left| \dot{\vec{p}}(t_0) \times \ddot{\vec{p}}(t_0) \right|}{\left| \dot{\vec{p}}(t_0) \right|^3}$$

# Lagrange Interpolation

**Given:** Point set $\{a_0, \ldots a_n\}$ and appropriate parameter values $\{t_0 < \ldots < t_n\}$

**Task:** Curve through $a_i$ for $t_i$

**1. solution:**

$$\vec{p}(t) = f_0(t)\vec{a}_0 + \ldots f_n(t)\vec{a}_n$$

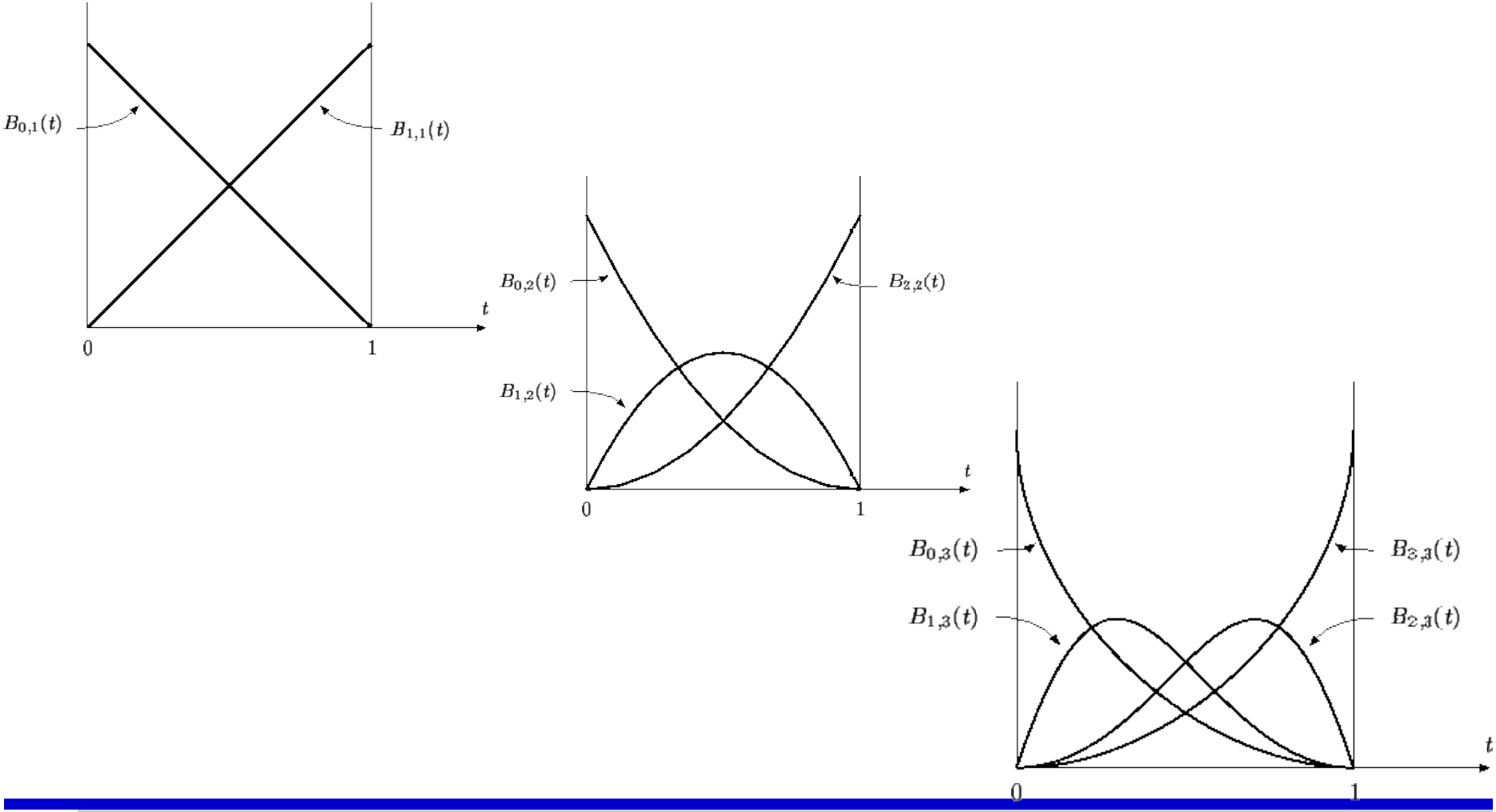
**Lagrange -Polynomial:**

$$f_i(t) = \frac{(t-t_0)(t-t_1)\ldots(t-t_{i-1})(t-t_{i+1})\ldots(t-t_n)}{(t_i-t_0)(t_i-t_1)\ldots(t_i-t_{i-1})(t_i-t_{i+1})\ldots(t_i-t_n)}$$

# Bernstein Polynomials

- $B_i^n(t) = \binom{n}{i}(1-t)^{n-i}\, t^i \qquad i = 0 \ldots n$
- **Properties:**
  - **Polynomials of order n**
  - $1 = \sum_{i=0}^{n} B_i^n(t)$

# Bernstein Polynomials: (C) Ken JOY



# Next Properties

- $$\binom{n}{i} = \frac{n!}{(n-i)!\,i!} = \binom{n}{n-i}$$

$$B^n_{n-i}(t^*) = B^n_i(t) = B^n_i(1-t^*) \qquad t^* = (1-t)$$

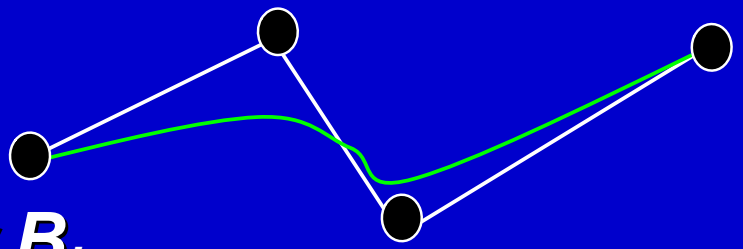
- *{1, t, ... $t^n$} is basis for polynomials the same way is {$B_0$, ... $B_n$} another basis*

# Bézier curve

- **Definition:**

$$\vec{p}(t) = \sum_{i=0}^{n} B_i^n(t)\vec{b}_i \qquad t \in [0,1]$$

- **Basic notions:**
  - Base points $B_i$
  - $B_i$ limited by base polygon
  - $b_i$ position vector for $B_i$



# Properties

- ***t = 0:*** $B_0^n(0) = 1$

  $B_i^n(0) = 0, \quad i = 1, \ldots n \quad \Rightarrow \vec{p}(0) = \vec{b}_0$

- ***t = 1:*** $B_n^n(1) = 1$

  $B_i^n(1) = 0, i = 0, \ldots n-1 \quad \Rightarrow \vec{p}(1) = \vec{b}_n$

- ***k-th derivative: depends on location t = 0 only of knot points $B_0$, ... $B_k$ . Analogously for t = 1.***

# Alternative Representation

- **Description with Shift operator: E**

$$E\left(\vec{b}_i\right) = \vec{b}_{i+1} \qquad E^0 = \mathrm{id}$$

$$E^k\left(\vec{b}_i\right) = \vec{b}_{i+k}$$

$$E^k E^j\left(\vec{b}_i\right) = E^{k+j}\left(\vec{b}_i\right)$$

- **Bézier-curve with Shift operator:**

$$\vec{x}(t) = \left[(1-t)E^0 + tE\right]^n \vec{b}_0$$

# DeCasteljau Algorithm

- ***Given:*** **Base points $\{b_0, \ldots b_n\}$, $t$**

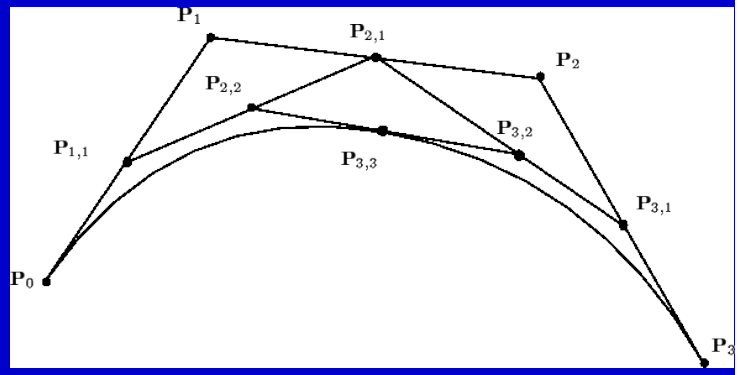



Fig. © by Ken Joy

- ***n iterations***

$$\vec{q}[i,0] = \vec{b}_i$$

$$\vec{q}[i, j+1] = (1-t)\vec{q}[i, j] + t\vec{q}[i+1, j]$$

$$\vec{q}[0, n] = \vec{x}(t)$$

- ***Version using one-dimensional array possible***

# Parameter transformations

- $t \in [0, a] \qquad a \leq 1$

  ***part of the original curve***

- $t \in [0, a] \qquad a > 1$

  ***original curve is the part - plus continuation***

# Spline curve

- Def.: „Spline curve“ consists of partial segments (Subsplines), combined by tangential or curvature preserving conditions
- Example: Bézier spline curve, binded from 2 Bézier curve pieces using the continuation.

# Rational Bézier curves

- Introduction of weights $\{w_0, \ldots w_n\}$ with the base points
- New form of the base polygon:

$$\vec{b}_i \rightarrow \vec{B}_i = \begin{cases} \begin{pmatrix} \omega_i \\ \omega_i \vec{b}_i \end{pmatrix} & \omega_i \neq 0 \\ \begin{pmatrix} 0 \\ \vec{b}_i \end{pmatrix} & \omega_i = 0 \end{cases}$$

- Representation via projection in the image plane

# Properties

- $w_0 = \ldots w_n = 1$ -> standard Bézier
- $w_0 = \ldots w_n <> 0$ -> rational Bézier

- Changing single weight:
  - $w_i$ increase: the curve goes closer
  - $w_i$ decrease: the curve goes far

- Modeling in higher dimensions, followed by projection

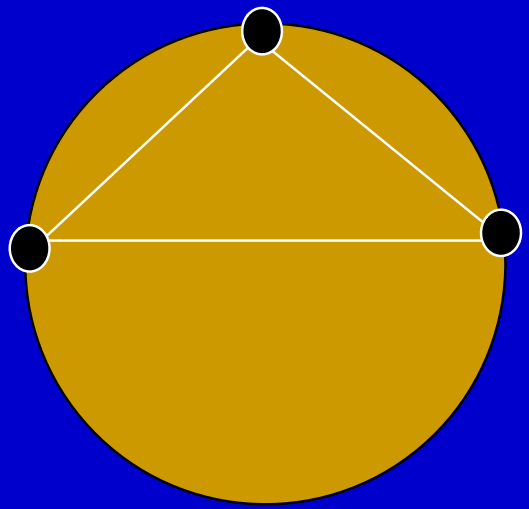
# Computation

- $B_i$ from input data computation
- Curve points $\vec{X}(t)$
- Projection to v0 =1:

$$\vec{x}(t) = \frac{\sum_{i=0}^{n} \omega_i \vec{b}_i B_i^n(t)}{\sum_{i=0}^{n} \omega_i B_i^n(t)} + \frac{\sum_{i=0}^{n} \vec{b}_i B_i^n(t)}{\sum_{i=0}^{n} \omega_i B_i^n(t)}$$

# Special Case - Circle

- Bézier curves cannot represent!
- Suitable setting of weights works for rational Bézier curves



# B-Splines

- *Idea:* Constant curvature setting of Bézier curves with Grad = 3.
- *Given:* Basis point $b_0$-$b_5$ define 3 Bézier curves with n = 3, when the subintervals are known (Design parameter)
- *Then:* The basis points of partial curves can be reconstructed.

# Curve Computation

- **Def.:** Giving the knot points $b_i$ and respective subintervals defines the Bézier spline curve of given order in a unique way.
- Method 1:
  - Find basis points of partial curve
  - Partial curves with DeCasteljau
- Method 2:
  - Both in short form: Cox / DeBoor
  - Own basis functions $N_{i,k}$

# Additional Properties

- „Local Control“: Control points influence the curve in one position t
- Implication:
  - curve can be modified using one control point
  - 2 different tangents in one point possible!

# Parameterisation Choice

- **Equidistant:** $u_i = i \qquad \forall i$
- **Chordal parameterisation:**

$$\frac{u_i - u_{i-1}}{u_{i+1} - u_i} = \frac{\left|\vec{b}_i - \vec{b}_{i-1}\right|}{\underbrace{\left|\vec{b}_{i+1} - \vec{b}_i\right|}_{*)}}$$

- **Centripedal parameterisation:**

$$\frac{u_i - u_{i-1}}{u_{i+1} - u_i} = \sqrt{*)}$$

# NURBS

- **Given:** $\{\vec{b}_0, \ldots \vec{b}_n\}$ $\{u_0 \leq \ldots \leq u_n\}$ $\{\omega_0, \ldots \omega_n\}$
- 1) Transform control points
- 2) B-Spline for given knot vector evaluation
- 3) Projection into the plane v0=1
- Non-Uniform-Rational-B-Spline curves

# Bézier Surfaces

- **Basis points:** $\{\vec{b}_{00}, \ldots \vec{b}_{mn}\}$

- **Bézier surface for given base points:**

$$\vec{x}(u,v) = \sum_i \sum_j \vec{b}_{ij} B_i^m(u) B_j^n(v) \quad (u,v) \in [0,1]^2$$

# Rational Bézier-Surfaces, B-Spline Surfaces

- Transformation into 4D space, compute there and project backwards, invariant!
- Rotational surfaces parametric exactly representable

- Advantage: Representation for general surfaces and surfaces of revolution

# Representation of FFC/FFS etc.

- Watt…
- Farin… and Web Notes by Ken JOY:
- http://muldoon.cipic.ucdavis.edu/~joy/
- Famous Curves Index: http://www-groups.dcs.st-and.ac.uk/~history/Curves/Curves.html - Astroid, Bicorn, Cardioid, Cartesian Oval, Cassinian Ovals, Catenary, Cayley's Sextic, Circle, Cissoid of Diocles, Cochleoid Conchoid, Conchoid of de Sluze, Cycloid, Devil's Curve, Double Folium, Dürer's Shell Curves, Eight Curve, Ellipse, Epicycloid, Epitrochoid, Equiangular Spiral, … Newton's Parabolas,Parabola… Witch of Agnesi

# Math Language Ruptures

- Elementary Arithmetics
- Algebra
- Infinitesimal Calculus
- Predicate Calculus
- Synthetic Geometry
- Analytic Geometry
- Iterative Geometry
- Set Theory

- (based on Kvasz´s epistemologic research, 1996)

# Analytic Geometry

- *Rene Descartes discovered the method how to assign to a given algebraic formula THE SHAPE.*
- *This visualization was so important that this new language was given a new name: analytic geometry.*
- *Constructive geometry using ruler and compass was difficult - for any object requires a specialised method and is limited to quadrics.*
- *Descartes method deconstructed each shape to points and enables us for constructing any shape POINT BY POINT.*
- *Therefore the curves are UNIVERSAL MODELING TOOL: car industry, flight simulations, caustics... Never ending story of applications.*
  - *(based on Kvasz´s epistemologic research, 1996)*

# Contents Review

1. Introduction and Motivation - OK
2. Data Structures & Data Formats - OK
3. Parametric Curves & Surfaces - OK
4. Color
5. Light-Material Interaction
6. etc.

# Thank You...

*… for Your attention.*

# Graficke systemy, vizualizacia a multimedia

## 2 / 1

*Letny semester 2006*