The Normal on Implicit Surfaces

- The normal in a point ***x*** on the implicit surface *f*(***x***) :

$$\mathbf{n}(\mathbf{x}) = \nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x}(\mathbf{x}) \\ \frac{\partial f}{\partial y}(\mathbf{x}) \\ \frac{\partial f}{\partial z}(\mathbf{x}) \end{pmatrix}$$

$$\approx \begin{pmatrix} f(x+\varepsilon, y, z) - f(\mathbf{x}) \\ f(x, y+\varepsilon, z) - f(\mathbf{x}) \\ f(x, y, z+\varepsilon) - f(\mathbf{x}) \end{pmatrix}$$

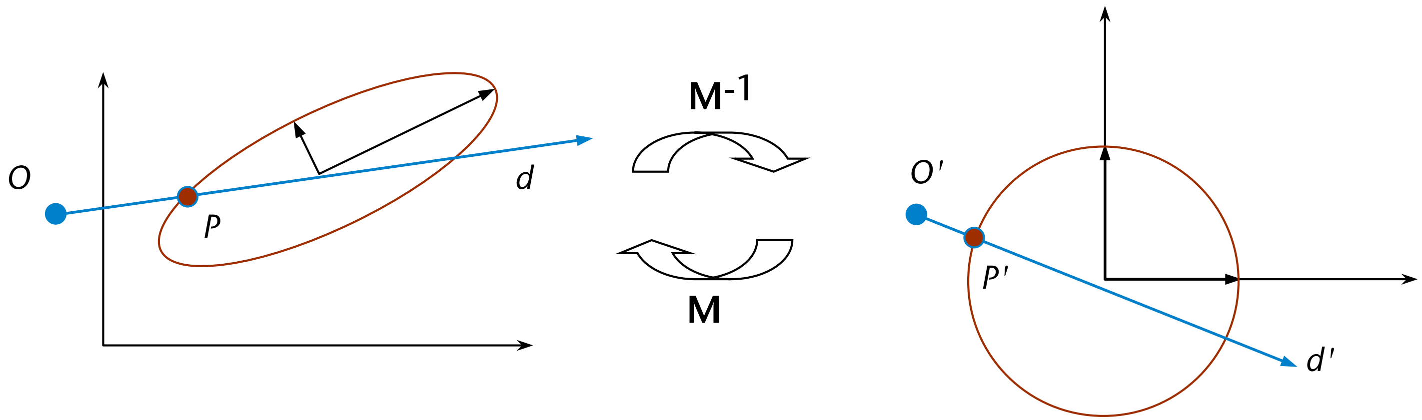
$$\approx \begin{pmatrix} f(x+\varepsilon, y, z) - f(x-\varepsilon, y, z) \\ f(x, y+\varepsilon, z) - f(x, y-\varepsilon, z) \\ f(x, y, z+\varepsilon) - f(x, y, z-\varepsilon) \end{pmatrix}$$

Instancing / Ray Transformation

- "Complex" (transformed) shapes can often be reduced to simpler & canonical shapes
- Idea:
  - Transform ray by inverse of shape's transformation
  - Compute intersection of ray and canonical shape
  - Transform intersection point (and normal) back

Example: Ray – Ellipsoid Intersection

- The back-and-forth transformations:



- The algorithm:

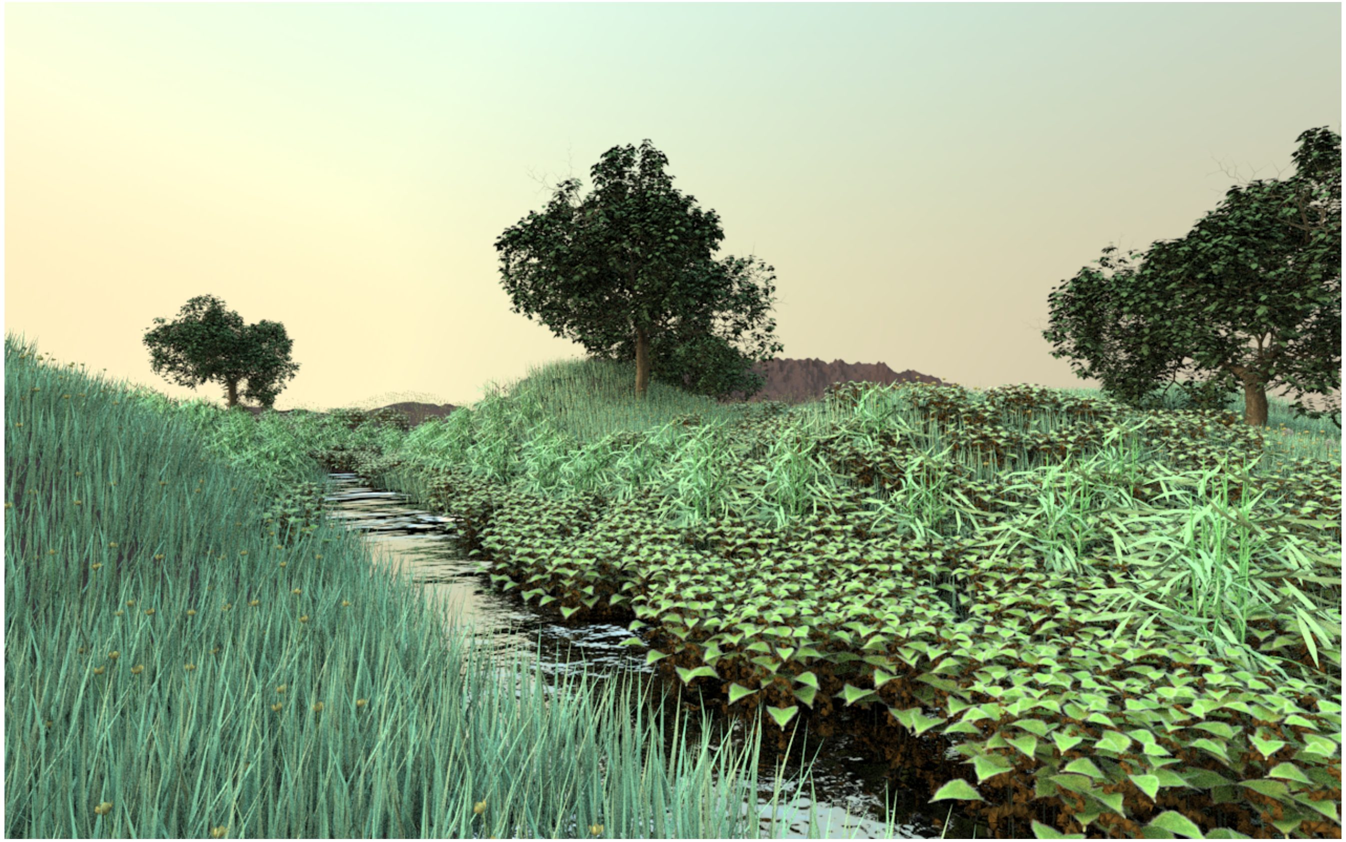
$$\text{berechne } P'(t) = \mathbf{M}^{-1}O + t\mathbf{M}^{-1}d$$

$$\text{schneide } P'(t) \text{ mit Einheitskugel } \rightarrow P' , \mathbf{n}' , t'$$

$$P := \mathbf{M} \cdot P' ; \mathbf{n} := (\mathbf{M}^{-1})^{\mathsf{T}} \cdot \mathbf{n}' ; t := ?$$

Another Reason for *Instancing*

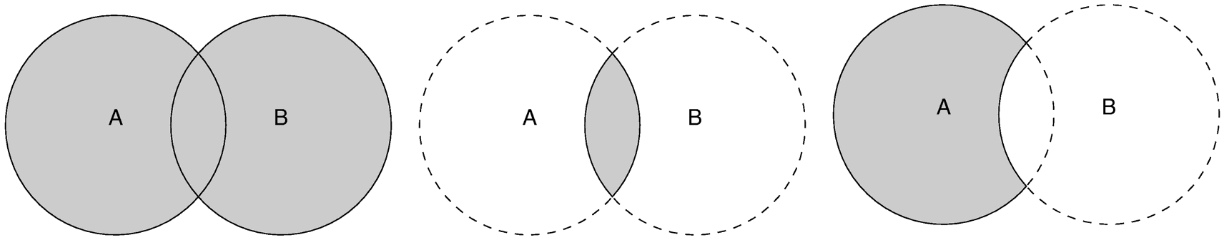
- Memory efficiency: only using instancing can you fit such huge scenes into main memory



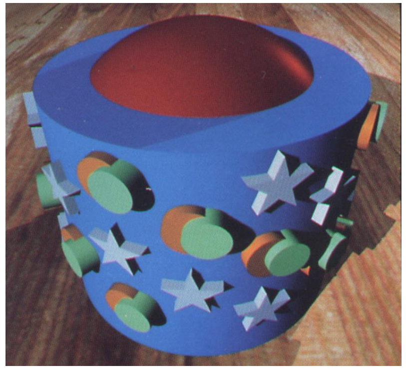
With instancing: 61 unique plant models, 1.1M unique triangles, 300MBytes
With explicit representation: 4000 instanced plants in the scene, 19.5M triangles

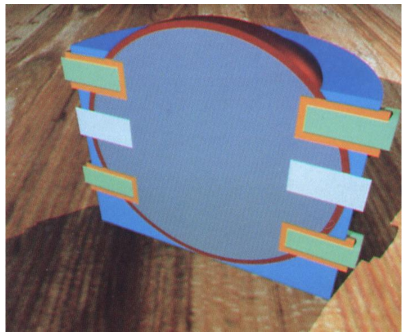
Constructive Solid Geometry (CSG)

- Very easy to render by ray-tracing
- Central idea: construct new a object by *set operations* performed on simpler, volumetric objects → *constructive solid geometry* (CSG)
- Simple primitives: all objects that can be described and ray-traced easily (e.g., sphere, box, ...)
- Set operations : union, intersection, difference



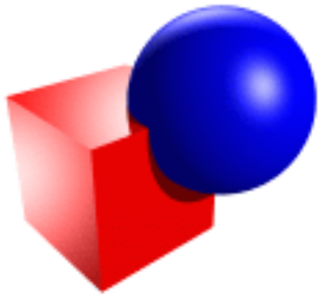
Union Intersection Difference





Examples

- The three set operations applied to sphere & box in 3D:



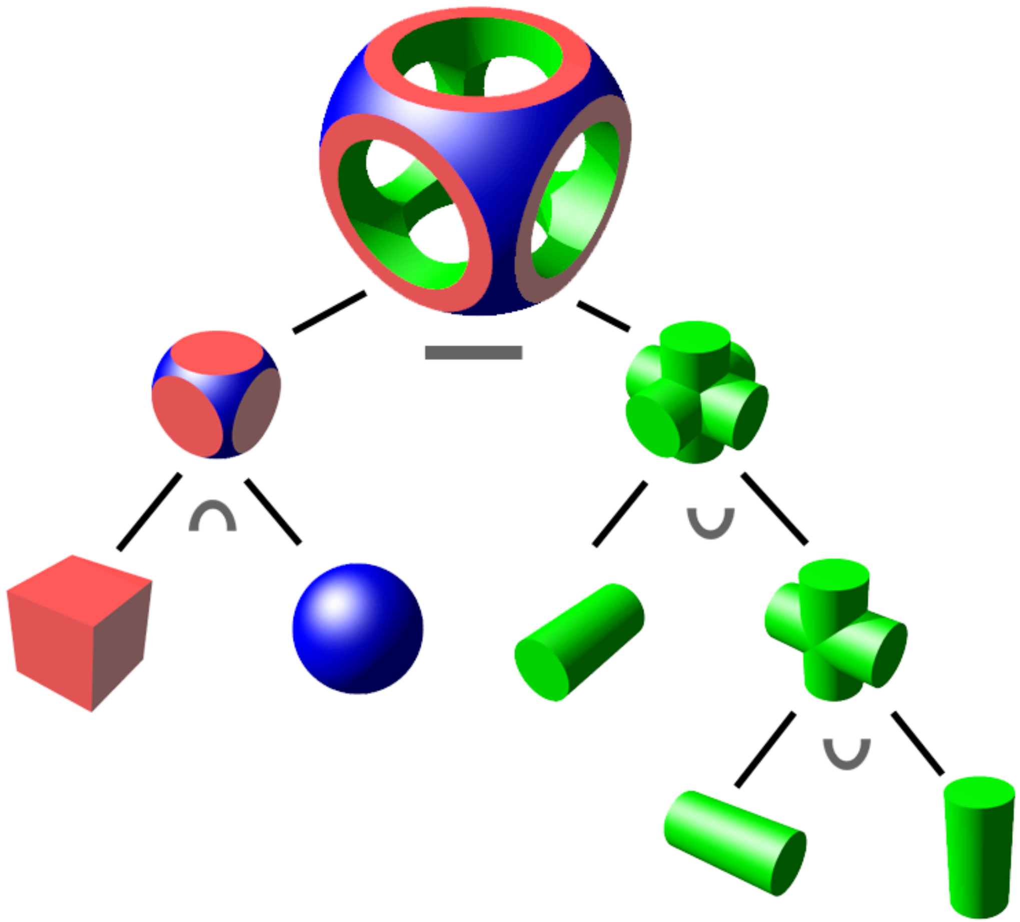
Union

Difference

Intersection

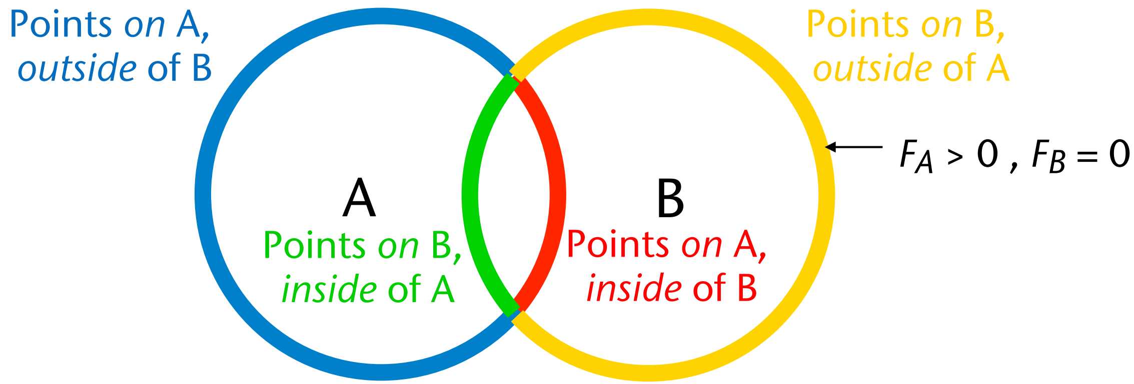
The CSG Tree by Way of an Example

- Recursive application of the set operations
  $\longrightarrow$ CSG tree = one CSG object
- Evaluation of CSG trees works similar to evaluation of arithmetic expression trees

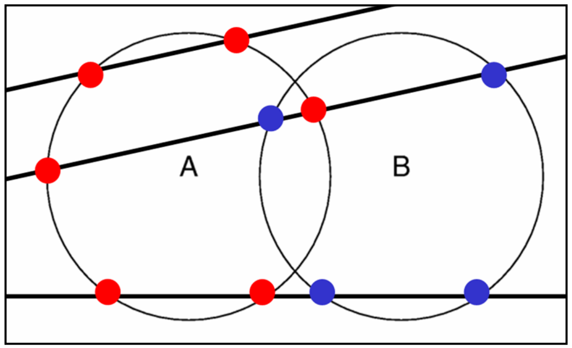


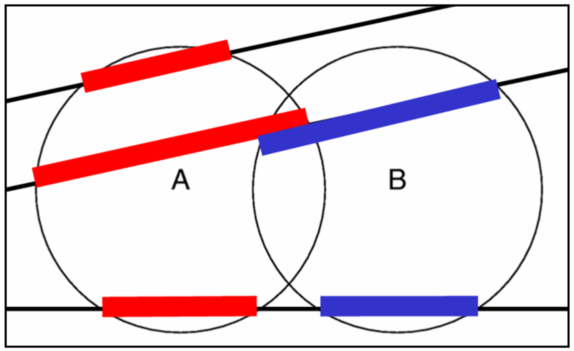
Raytracing of CSG Objects

- Use implicit or explicit representation of the primitives:

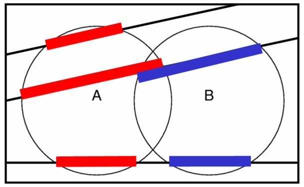


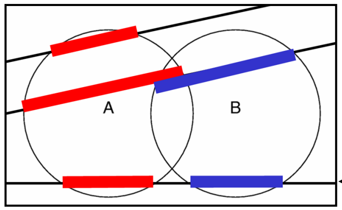
- Determine **all** intersection points of a ray with the 2 primitives
- If primitives are convex → one interval where ray is inside primitive



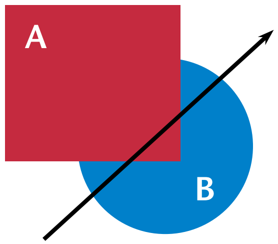


- Apply the CSG operation on the intervals
- Propagate intervals up through the CSG tree
- If interval is empty when reaching root → no intersection
- Else: choose closest interval and point closest to viewpoint
- Warning:
  - During CSG operations on intervals, the resulting interval can be non-contiguous (i.e., several intervals need to be maintained during tree traversal)!





With A∪B, we get "one" non-contiguous interval in case of this ray!



Dito in this case with B - A !

  - Also, pay attention to numerical robustness (e.g., kill too small intervals)

CSG in Design

Ferruccio Laviani

Solution to the Quizz

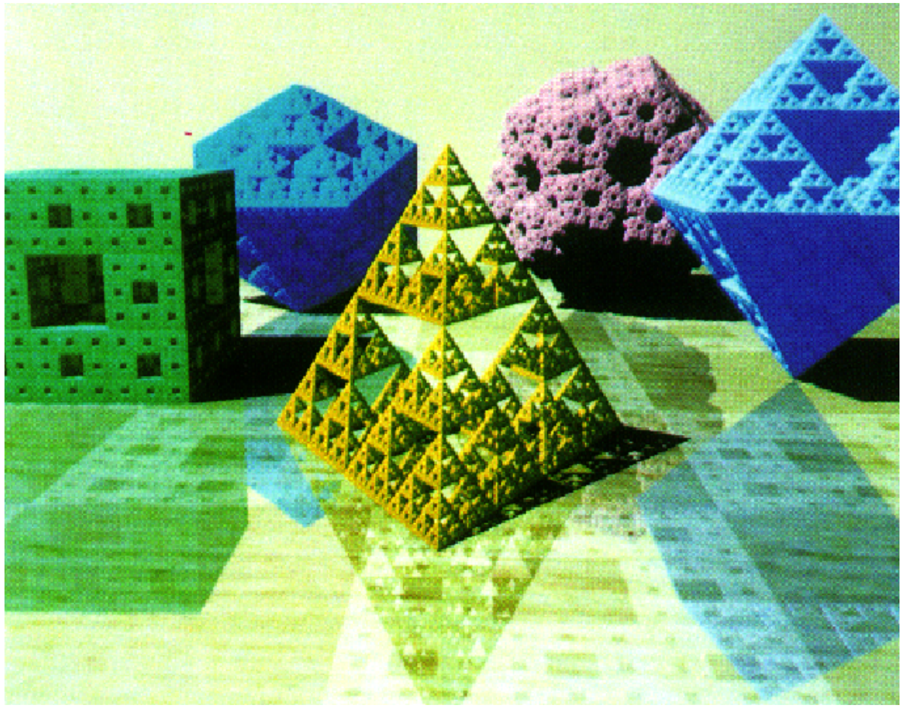
"Villarceau Circles" by Tor Olav Kristensen (2004)

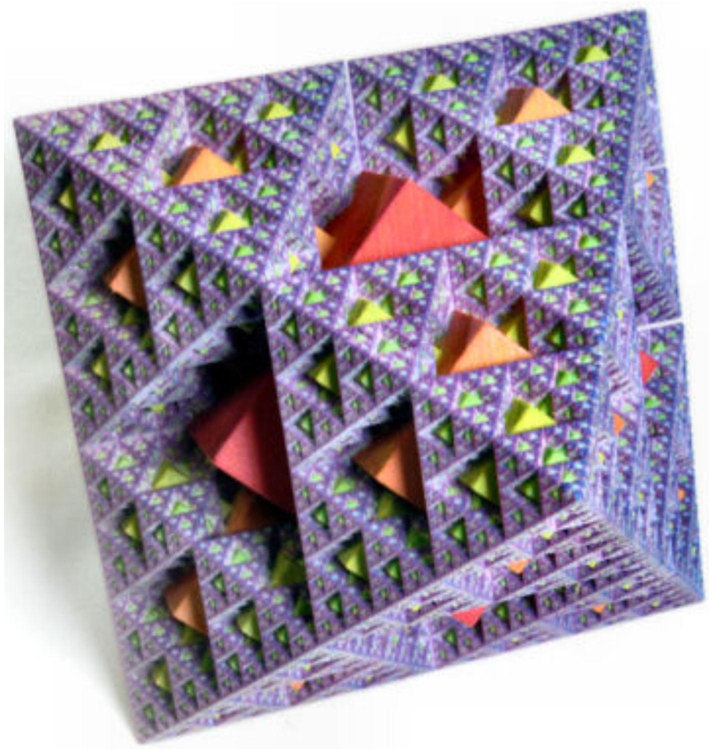
For every point on a torus one can draw four different circles through it that all lie on the surface of the torus. Two of these four circles are called Villarceau circles. The four narrow pairs of bands in this image follow such Villarceau circles.
All the shapes in this image are made with Constructive Solid Geometry operations with tori only (except for the ground plane of course).

Fractals

- Fractals can be ray-traced trivially
  - At least the simple fractals as shown below
- Just recurse "on demand" up to some predefined depth

→ Prozedural objects





An Early Application of Fractals

- Fractals are very well-suited for modeling terrain:

Loren C. Carpenter: Vol Libre. 1980