we did not deal with, is minimization of number of rays totally missing the object. The minimal CD of voxels such ray passes through can be used to estimate a distance in which it is sufficient to generate next ray without missing any object voxel.

References

[Bor86] Gunilla Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34(3):344–371, 1986.

[CW88] John C. Cleary and Geoff Wyvill. Analysis of an algorithm for fast ray tracing using uniform space subdivision. *The Visual Computer*, 4(2):65–83, July 1988.

[LC87] W.E. Lorensen and H.E. Cline. Marching cubes: A high-resolution 3D surface construction algorithm. *Computer Graphics*, 21(4):163–169, July 1987.

[OM87] Matasaka Ohta and Mamoru Maekawa. Ray coherence theorem and constant time ray tracing algorithm. In T. Kunii, editor, *Computer Graphics 1987 – Proceedings of CG International '87*, pages 303–314. Springer–Verlag, 1987.

[SW91] John Spackman and Philip Willis. The smart navigation of a ray through an oct-tree. *Comput.&Graphics*, 15(2):185–194, 1991.

[Š92] Miloš Šrámek. Ray tracing volume data with subvoxel precision. In Václav Skála, editor, *Proceedings of the Winter School on Computer Graphics and CAD systems*, pages 47–65, Plzeň, Technical University, ČSFR, 1992.

[Š93] Miloš Šrámek. Interactive segmentation of tissues for medical imaging. In Václav Hlaváč and Tomáš Pajdla, editors, *Czech Pattern Recognition Workshop '93*, pages 164–171, Temešvár u Písku, Czech Republic, November 4^{th}-6^{th}, 1993.

[Š94] Miloš Šrámek. Cubic macro-regions for fast voxel traversal. In *GKPO'94*, Poland, June 1994. accepted.

[YCK92] Roni Yagel, Daniel Cohen, and Arie Kaufman. Discrete ray tracing. *IEEE Computer Graphics and Applications*, 12(5):19–28, September 1992.

Generating Plants for Computer Graphics

Bedrich Benes CTU, Fac. of Electrical Eng.,
Dept. of Computer Science
Karlovo nám. 13, 121 35 Praha 2
BENES@CSLAB.FELK.CVUT¿CZ

keywords: Tree, L-system, Rendering, Computer Graphics

Computer Graphics focuses its activites mostly on the modelling of the real world. It means in practice to find a model of the modelled object which can be interpreted by means of computer graphics. There has been a lot of progress in this research. We have fundamental models of light of reflecting, and are able to simulate real surfaces (e.g. an orange, a metal, a wood and so on). To visualise these models we can use on of two principal methods: **ray tracing** or **radiosity** [ZARA92].

There are several open problems in Comupter Graphics. One of these is the general problem of time. These methods are really strongly time consuming. This time depends on the modeled scene. The time needed for visualisation increase with the number of modeled objects and can be strongly decreased with a smart algorithm or a good data representation or speed hardware.

Lindenmayer [LIND68] showed a structure of a parallel string rewriting system for the growth of living organisms in 1968. This method describes the geometry of a tree, but it was used for these purposes for the first time in 1984 in the work of Aono and Smith [SMIT84]. This work continues in the work of Lindenmayer and Pruzsinkiewicz at present. A focus of these works lies in the topology of trees, a geometry of these objects is showed in works of [BLOO85], [OPPE86], [KAWA82].

The generation of topology is based on string rewriting systems (called by Lindemayer [LIND68] L-systems).

Let V [PRUS86] equal a set called alphabet whose members are symbols. Let V^* equal a set of words over V and V^+ is a set of non-empty words over V.

The L-system is a triple [LIND68] $G = (V, \omega, P)$, where

- V is an alphabet

- $\omega \in V^+$ is a word called axiom
- $P \subset V \times V^+$ is a set of production rulese where
  - $a \in V$ is predictor
  - $\alpha \in V^+$ je successor and
  - $p = [a, \alpha] \in P$, shall be written as $a \Rightarrow \alpha$.

We shall suppose that $\forall a \in V$ at least one $\alpha \in V^*$ exists so $a \Rightarrow \alpha$.

We add symbols [LIND68] [a] to the alphabet of the system. These symbols define a branch. If any string is derived from an axiom of the L-system we can construct a geometrical model. The generation of geometry means interpretation of the symbols of the alphabet by **turtle graphics**.

The turtle is represented by its position and orientation. Orientation is defined by three vectors: $\vec{H}$–head,$\vec{L}$–left, $\vec{U}$–up. The position is defined by point in 3D space (x, y, z).

The link between topology and geometry is to be made by the relation which links to each symbol in the alphabets an action of the turtle. This relation is described by tabular 1.

Topology	Geometry
F	go ahead in direction of $\vec{H}$, length of d
+	turn left, angle δ around $\vec{U}$
-	turn right, angle δ around $\vec{U}$
^	turn up, angle δ around $\vec{L}$
&	turn down, angle δ around $\vec{L}$
\	turn left, angle δ around $\vec{H}$
/	turn right, angle δ around $\vec{H}$
[	push position onto stack
]	pop position from stack

Tabulka 0.1: A realtion between topolgy and geometry

The output of the action of turtle is a set of points in 3D. These points define the geometry of tree. Each action of "go ahead" means to put a new point of branch into the model, each "push" and "pop" means to start and finish a branch. Points generated in this way can be interpreted as a 3D spline

(e.g Catmull-Room's or Coons's) and a circle sweep around this spline defines a shape of this branch. These data can be converted into 3D triangles with normal vectors, which are good input data for any ray tracer or radiosity program.

- [BLOO85], Bloomenthal J., Modelling the mighty maple, **ACM SIGGRAPH Vol.19, 1985**
- [KAWA82] Kawaguchi Y. A morphological study of the form of nature. **ACM SIGGRAPH 82 In CG 1982**
- [LIND68], Lindenmayer A. Mathematical model for cellural interaction in developement, Parts I and II, **Journal of theoretical biology 1968**
- [OPPE86], Oppenheimer P, Real time design and animation of fractal plants and trees, **ACM SIGGRAPH Vol.20, No4. 1986**
- [PRUS86], Prusinkiewicz P, Graphical application of L-systems, **Vision interface proceedings 1986**
- [SMIT84], Smith A.R Plants, fractals and formal languages, **ACM SIGGRAPH 84 CG, 18,3, 1984**
- [ZARA92], Žára J, Limpouch A, Beneš B, Werner T, Počítačová grafika, pricipy a algoritmy, **Grada 1992**