Computer Graphics Course

Three-Dimensional Modeling

Function representation

References

- "Function representation in geometric modeling: concepts, implementation and applications", A. Pasko, V. Adzhiev, A. Sourin, V. Savchenko, The Visual Computer, vol. 11, No. 8, 1995, pp. 429-446.
- "Implicit surfaces for geometric modeling and computer graphics", J. Menon and B. Wyvill (Co-chairs), SIGGRAPH'96 Course Notes No.11.
- "Implicit modeling techniques", J. Menon et al., Course Notes, Shape Modeling International '97, University of Aizu, 1997, 132 p. Files at: `/public/www/labs/sw-sm/SMI97/tut/`
- Ricci A. "A constructive geometry for computer graphics", The Computer Journal, vol. 16, No. 2, 1973, pp.157-160.
- Rvachev V.L. "On the analytical description of some geometric objects", Reports of Ukrainian Academy of Sciences, vol. 153, No. 4, 1963, pp. 765-767 (in Russian).

Contents

1. Motivation for F-rep introduction
2. F-rep concepts
   - Objects
   - Operations
   - Relations
3. Example of complex metamorphosis
4. Relations and collision detection
5. Advanced topics: polygons and sweeping

Motivation

View inside implicits:

- The idea of representation of an **entire** complex object by a single real function:
  - applied effectively in skeletal implicits;
  - min/max operations for CSG proposed by Ricci [1973] have not found wide acceptance (C^1 discontinuity as a reason);
  - existence of R-functions proposed by Rvachev [1963], surveyed by Shapiro [1988], and applied in multidimensional geometric modeling by Pasko [Ph.D. thesis, 1988].

More deep connection with *CSG* is possible.

- Theoretical possibility to derive an implicit description of a surface swept by a woving solid [Wang 1984]. Symbolic computations required to yield a formula for the implicit form.

More deep connection with *sweeping* is possible.

Constructive Solid Geometry (CSG)

Motivation

- Attention is paid in Computer Aided Geometric Design (CAGD) to implicit surfaces because of their closure under some important operations: *offsetting* and *blending* [Hoffman 1993].
- Generalization of the "implicit function representation" by introducing a *deformation* technique using a matrix of free vibrations [Sclaroff and Pentland 1991].
- Research on *collision detection* for implicit surfaces [Gascuel 1993].
- Similar polygonization algorithms stress the common nature of implicit and voxel models. Time-dependent transformation (*metamorphosis*) of skeletal implicits [Wyvill 1991] and scheduled Fourier volume morphing [Hughes 1992].

More deep connection with *voxel models* is possible.

Motivation

Conclusions from the above overview:

- Representations by real functions are widely used in geometric modeling and computer graphics in several forms.
- These models are not closely related to each other.
- These models are not closely related to such well-known representations as CSG, B-rep, sweeping, and spatial partitioning (voxel models).
- This obviously retards further research.
- A uniform *function representation* is needed to fill these gaps. It has to:

- unify all functionally based approaches;
- be convertible from other representations;
- be dimension independent;
- have as reach as possible a system of operations and relations.

F-rep concepts

Let us describe *geometric concepts* of a functionally based modeling environment as a triple:

$$(M, \Phi, W)$$

where
M is a set of *geometric objects*,
Φ is a set of *geometric operations*,
W is a set of *relations* on the set of objects.
Mathematically this triple is a sort of ***algebraic system***.

<u>Objects</u>
Geometric objects are considered as closed subsets of *n*-dimensional Euclidean space E^n with the definition:

$$f(x_1, x_2, ..., x_n) \geq 0$$

where *f* is a real continuous function defined on E^n. We call *f* a *defining function*.

This inequality is called a

function representation (or *F-rep*)
of a geometric object.

In 3D space, the boundary of such an object is a so-called *implicit surface*.
The function can be defined by:
1) analytical expression;
2) function evaluation algorithm;
3) tabulated values and an appropriate interpolation procedure.

The major requirement to the function is to have at least C^0 continuity.
There is a classification of points in E^n space associated with the closed n-dimensional object:

$f(\boldsymbol{X}) > 0$ - for points inside the object;
$f(\boldsymbol{X}) = 0$ - for points on the object's boundary;
$f(\boldsymbol{X}) < 0$ - for points outside the object.

Here, $\boldsymbol{X} = (x_1, x_2, ..., x_n)$ is a point in E^n.
Note that the definition of an object is the inequality with the explicit function of n variables $f = f(x_1, x_2, ..., x_n)$ but not the implicit function of n-1 variables $f(x_1, x_2, ..., x_n) = 0$.

⇨ The objects are defined in multidimensional space for choosing a space of arbitrary dimension in each specific case. For example, if $n = 4$ then (x_1, x_2, x_3) can be space coordinates and x_4 can be interpreted as time.

⇨ Two major types of elements of the set *M* are

1) *primitives* (simple geometric objects);
   Each geometric primitive is described by the concrete type of a function chosen from the finite set of such types.
2) *complex geometric objects*.
   A complex geometric object is a result of operations on primitives.

⇨ In the modeling system, the finite set of primitives can be defined. However, the possibility of the *extension* of this set in a symbolic manner should be provided. Actually, this approach allows the modeling system to be initially "*empty*" and make the user to be responsible for an application oriented filling of the primitive set.

Operations

The set of geometric operations Φ includes such operations as:

$$\Phi_i: M^1 + M^2 + \ldots + M^n \rightarrow M$$

where n is a number of operands of an operation. The result of each operation is also an object from the set M that ensures the closure property of the function representation.

⇨ Two main classes:

1) ***unary*** ($n=1$) operations

Let object G_1 has the definition $f_1(X) \geq 0$. The term "unary operation" on the object G_1 means the operation $G_2 = \Phi_i(G_1)$ with the definition

$$f_2 = \Psi(f_1(X)) \geq 0,$$

where Ψ is a continuous real function of one variable. The examples of unary operations are the bijective mapping, affine mapping, projection, offsetting.

2) ***binary*** ($n=2$) operations.

The binary operation on objects G_1 and G_2 means the operation $G_3 = \Phi_i(G_1, G_2)$ with the definition

$$f_3 = \Psi(f_1(X), f_2(X)) \geq 0,$$

where Ψ is a continuous real function of two variables.

The examples of binary operations are the set-theoretic operations, blending operations, Cartesian product, metamorphosis.

⇨ As with the objects, the user of the F-rep based modeling system is able to introduce any desired operation by its analytical or procedural description in symbolic form and thus extend the list of operations.

Relations

A binary relation is a subset of the set $M^2 = M \times M$. It can be defined as

$$S_i: M \times M \rightarrow I$$

The examples of binary relations are inclusion, point membership, interference or collision.

Types of R-functions

Correspondence between set-theoretic operations on geometric objects G_i and operations on their defining functions f_i:

Union $G_3 = G_1 \cup G_2 \rightarrow f_3 = f_1 / f_2$
Intersection $G_3 = G_1 \cap G_2 \rightarrow f_3 = f_1 \& f_2$
Subtraction $G_3 = G_1 \setminus G_2 \rightarrow f_3 = f_1 \setminus f_2$

|,& ,\ are signs of R-functions.

One of the possible analytical descriptions of the R-functions is as follows:

$$f_1 | f_2 = \frac{1}{1+\alpha}(f_1 + f_2 + \sqrt{f_1^2 + f_2^2 - 2\alpha f_1 f_2})$$

$$f_1 \& f_2 = \frac{1}{1+\alpha}(f_1 + f_2 - \sqrt{f_1^2 + f_2^2 - 2\alpha f_1 f_2})$$

where $\alpha = \alpha(f_1, f_2)$ is an arbitrary continuous function satisfying the following conditions:

$$-1 < \alpha(f_1, f_2) \le 1,$$

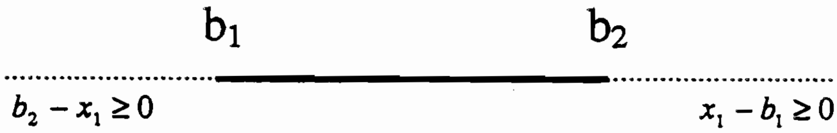
$$\alpha(f_1, f_2) = \alpha(f_2, f_1) = \alpha(-f_1, f_2) = \alpha(f_1, -f_2)$$

The expression for the subtraction operation is

$$f_1 \setminus f_2 = f_1 \& (-f_2)$$

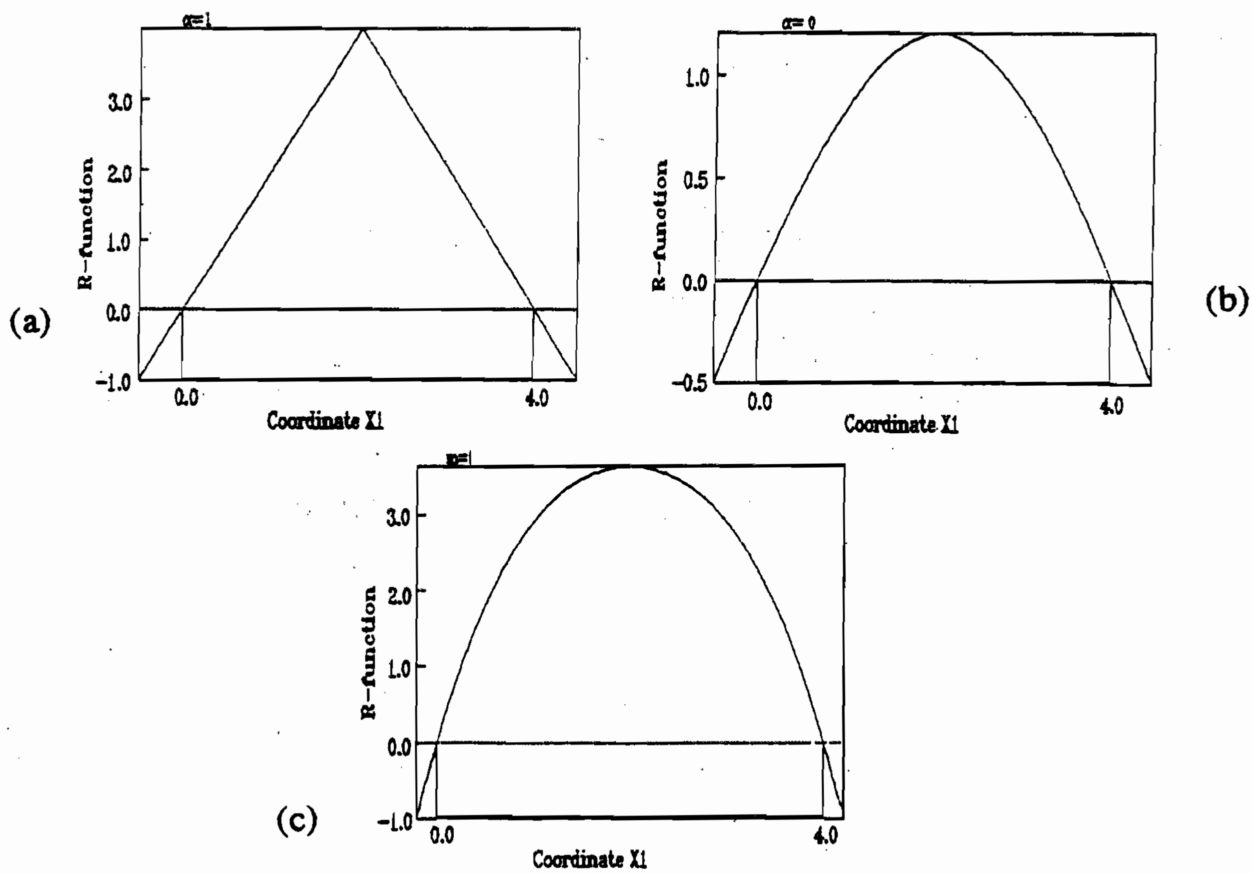
1D example

The description of a segment in E^1 can be obtained from the descriptions of two rays as follows:



$$f(x) = (x_1 - b_1) \& (b_2 - x_1)$$

The plot of this function for the min/max functions is shown in Fig. 1a with α=1, Fig. 1b corresponds to α=0, and Fig. 1c corresponds to C^m functions with m=1. It is important to point out that the function in Figs. 1b and 1c does not have points in its domain where the derivative is discontinuous.



Note that with this definition of the subtraction, the resultant object includes its boundary.

If α=1, the above functions become

$$f_1 | f_2 = \min(f_1, f_2)$$
$$f_1 \& f_2 = \max(f_1, f_2)$$

This is a particular case described by Ricci [1973]. The *min/max* functions are very convenient for calculations but have C^1 discontinuity when $f_1 = f_2$.

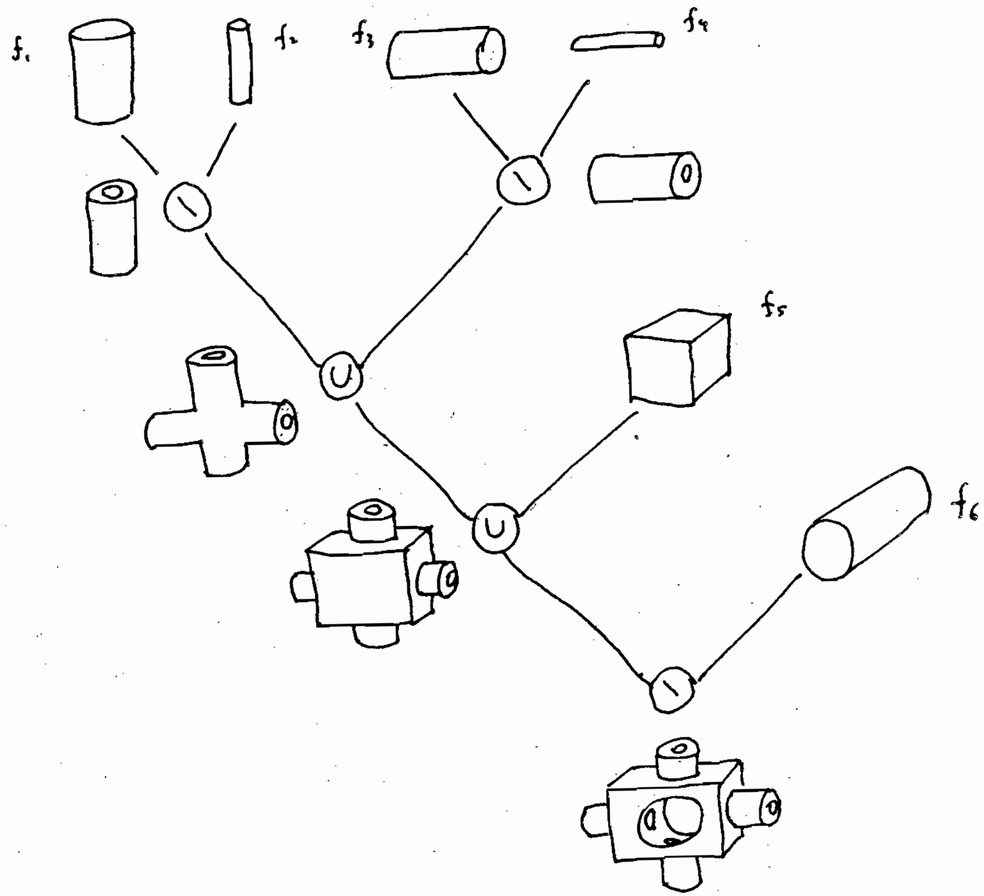
If α=0, the above functions take the most useful in practice form:

$$f_1 | f_2 = f_1 + f_2 + \sqrt{f_1^2 + f_2^2}$$
$$f_1 \& f_2 = f_1 + f_2 - \sqrt{f_1^2 + f_2^2}$$

These functions have C^1 discontinuity only in points where both arguments are equal to zero. If C^m continuity is to be provided, one may use another set of R-functions:

$$f_1 | f_2 = \left(f_1 + f_2 + \sqrt{f_1^2 + f_2^2}\right)\left(f_1^2 + f_2^2\right)^{\frac{m}{2}}$$
$$f_1 \& f_2 = \left(f_1 + f_2 - \sqrt{f_1^2 + f_2^2}\right)\left(f_1^2 + f_2^2\right)^{\frac{m}{2}}$$

3D example



$$F = (((f_1 \setminus f_2) \,/\, (f_3 \setminus f_4)) \,/\, f_5) \setminus f_6$$

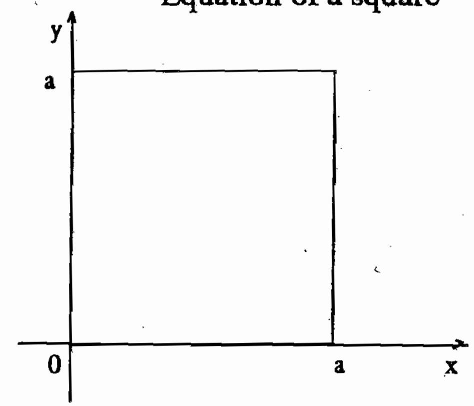
2D example

$$2a - \sqrt{2x^2+a(a-2x)} - \sqrt{2y^2+a(a-2y)} -$$

$$- \sqrt{4a^2-2a(\sqrt{2x^2+a(a-2x)}+\sqrt{2y^2+a(a-2y)})+2x(x-a)+2y(y-a)} = 0$$

?

Equation of a square



G: $f(x,y) \geq 0$

G_1: $x \geq 0$, $f_1(x,y) = x$
G_2: $x \leq a$, $f_2(x,y) = a - x$
G_3: $y \geq 0$, $f_3(x,y) = y$
G_4: $y \leq a$, $f_4(x,y) = a - y$

$f = f_1 \,\&\, f_2 \,\&\, f_3 \,\&\, f_4$

Polygon-to-function conversion

Data: a 2D simple polygon

Find:

$$F(x,y) \begin{cases} = 0 & \text{at polygon edges;} \\ > 0 & \text{inside the polygon;} \\ < 0 & \text{outside the polygon.} \end{cases}$$

Requirements to the conversion algorithm:

- It should provide exact polygon description as a zero set of a real function;
- No points with zero function value should be inside or outside a polygon;
- It should allow processing a simple arbitrary polygon without any additional information.

Monotone formula

Rvachev [1974] and Peterson [1984] - monotone formula.
Dobkin et al. [1988] - an efficient algorithm for deriving it.

A set-theoretic formula where each of the half-planes appears exactly once and no additional half-plane is used.

Elements and notation:

- Polygon vertices $A_1(x_1, y_1), A_2(x_2, y_2), \ldots, A_n(x_n, y_n)$
- Line passing through points $A_i(x_i, y_i)$ and $A_{i+1}(x_{i+1}, y_{i+1})$:

$$f_i \equiv -x(y_{i+1} - y_i) + y(x_{i+1} - x_i) - x_{i+1}y_i + x_i y_{i+1} = 0$$

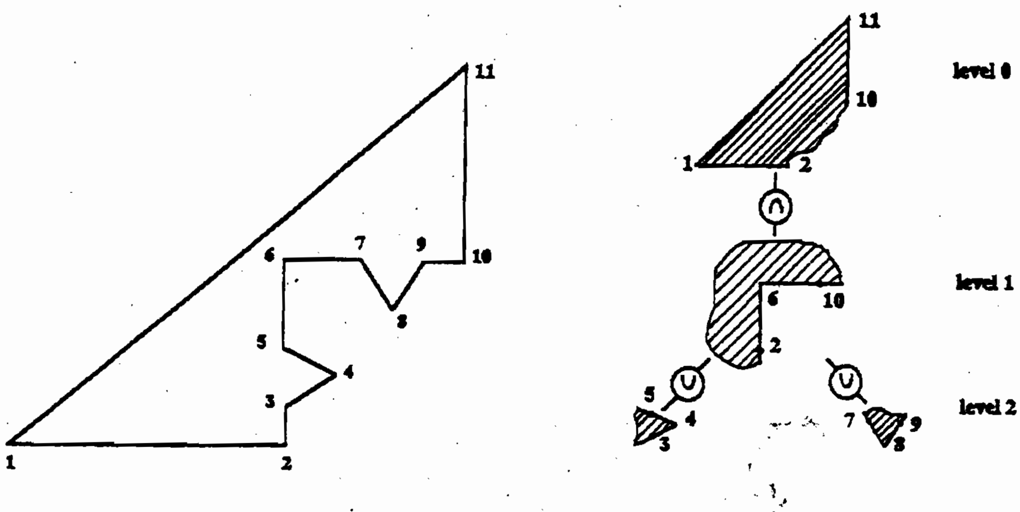
- f_i is a function positive in an open region Ω_i^+ and negative in an open region Ω_i^-, half-plane.
- Internal region of a convex polygon:

$$\Omega^+ = \Omega_1^+ \cap \Omega_2^+ \cap \ldots \cap \Omega_n^+.$$

- External region of a convex polygon

$$\Omega^+ = \Omega_1^+ \cup \Omega_2^+ \cup \ldots \cup \Omega_n^+.$$

Deriving a formula



Level 0: $A_1A_2A_{10}A_{11}$
Level 1: $A_2A_6A_{10}$
Level 2: $A_3A_4A_5$, $A_7A_8A_9$

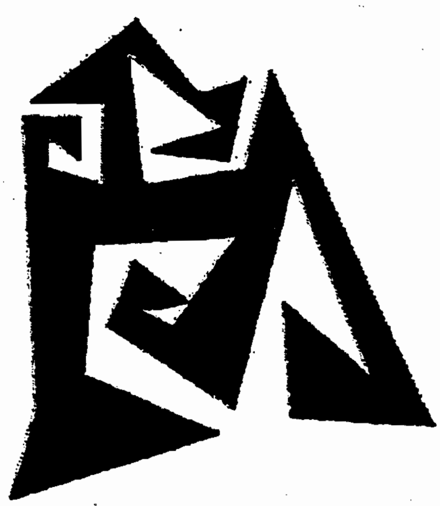
$$\Omega^+ = \Omega_1^+ \cap \left(\Omega_2^+ \cup \left(\Omega_3^+ \cap \Omega_4^+\right) \cup \Omega_5^+ \cup \Omega_6^+ \cup \left(\Omega_7^+ \cap \Omega_8^+\right) \cup \Omega_9^+\right) \cap \Omega_{10}^+ \cap \Omega_{11}^+$$

- Each region is presented only once.
- Set-theoretic operation applied to a region is determined by the tree level which this region belongs to.

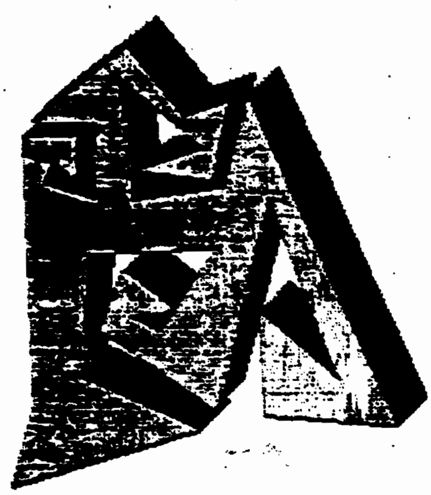
$\Omega_2^+ \equiv \Omega_5^+$ and $\Omega_6^+ \equiv \Omega_9^+$:

$$\Omega^+ = \Omega_1^+ \cap \left(\Omega_2^+ \cup \left(\Omega_3^+ \cap \Omega_4^+\right) \cup \Omega_6^+ \cup \left(\Omega_7^+ \cap \Omega_8^+\right)\right) \cap \Omega_{10}^+ \cap \Omega_{11}^+.$$

Sweeping with converted polygons



$f_1(x,y) \geq 0$



$f_3(x,y,z) \geq 0$

Goal: Sweeping operation to be closed on F-rep.

- $f_1(x,y)$ defines some polygon on the xy-plane;

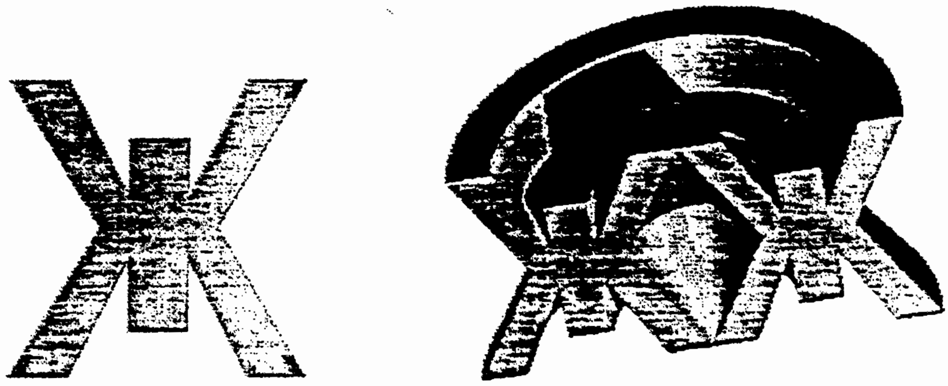
- $f_2(z) = z \wedge (a-z)$ defines the segment [0,a] on the z-axis;

- definition of a 3D translational sweep:

$$f_3(x,y,z) = f_1(x,y) \wedge f_2(z)$$

- intersection of the infinite cylinder with the boundary $f_2(x,y) = 0$ and two halfspaces $z \geq 0$ and $a \geq z$.

Rotational sweeping



The mapping from the **Cartesian coordinate system**
to the **cylindrical coordinate system:**

$$x' = \sqrt{x^2 + y^2}$$

$$y' = arctan(y / x)$$

Variable generator

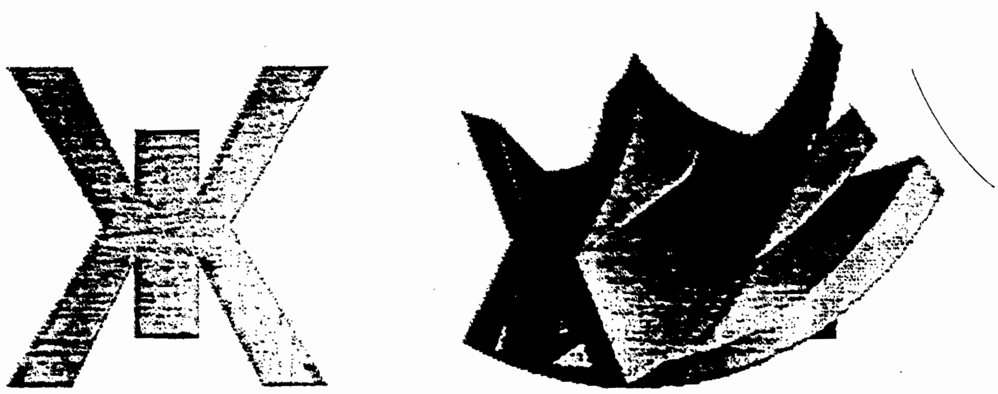
More general sweeping: a polygon generator dependent on z.

1) Sweeping by the polygon moving in xy directions:

$$f_3(x,y,z) = f_1(x-\phi_1(z), y-\phi_2(z)) \wedge f_2(z)$$

2) polygon changing its position, orientation and shape:

$$x' = \phi_1(x,y,z),$$
$$y' = \phi_2(x,y,z),$$
$$f_3(x,y,z) = f_1(\phi_1(x,y,z), \phi_2(x,y,z)) \wedge f_2(z)$$



$$\theta = kz, c = cos(\theta), s = sin(\theta),$$
$$x' = xc + ys$$
$$y' = -xs + yc$$

Inclusion and point membership

Inclusion relation

This relation is described as $G_2 \subset G_1$ and means that the object G_2 is a subset of G_1. If G_2 is a point P, the relation can be described by the following bivalued predicate:

$$S_2(P,G_1) = \begin{cases} 0, \text{ if } f_1(\mathbf{X}) < 0 & \text{for } P \notin G_1 \\ 1, \text{ if } f_1(\mathbf{X}) \geq 0 & \text{for } P \in G_1 \end{cases}$$

Point membership relation

Let iG_1 be the interior of G_1 and bG_1 be the boundary of G_1. The point membership relation is described by the 3-valued predicate:

$$S_3(P,G_1) = \begin{cases} 0, \text{ if } f_1(\mathbf{X}) < 0 & \text{for } P \notin G_1 \\ 1, \text{ if } f_1(\mathbf{X}) = 0 & \text{for } P \in bG_1 \\ 2, \text{ if } f_1(\mathbf{X}) > 0 & \text{for } P \in iG_1 \end{cases}$$

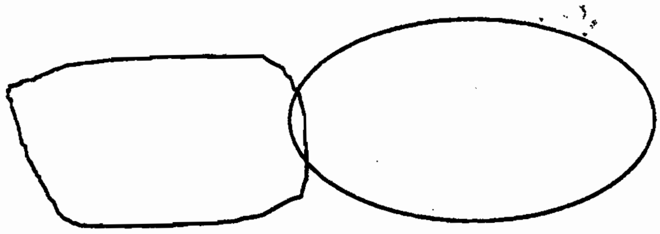
This predicate can be correctly evaluated for G_1 without "*internal zeroes*" (internal points with $f_1(X) = 0$).

Intersection relation

The intersection (interference, collision) relation is defined by the bivalued predicate:

$$S_c(G_1, G_2) = \begin{cases} 0, \text{ if } G_1 \cap G_2 = \varnothing \\ 1, \text{ if } G_1 \cap G_2 \neq \varnothing \end{cases}$$

$G_1: f_1(\mathbf{X}) \geq 0$ $\quad\quad$ $G_2: f_2(\mathbf{X}) \geq 0$



Is intersection empty?

A function $f_3(\mathbf{X}) = f_1(\mathbf{X}) \& f_2(\mathbf{X})$ defining the result of the intersection can be used to evaluate S_c. It can be stated that $S_c = 0$ if $f_3(\mathbf{X}) < 0$ for any point of E^n (Rvachev 1967).

Collision detection algorithm

1. Calculate the admissible domain D for given two objects.

2. Find p^* where $f(p^*)=\max(f(p))$ in D.

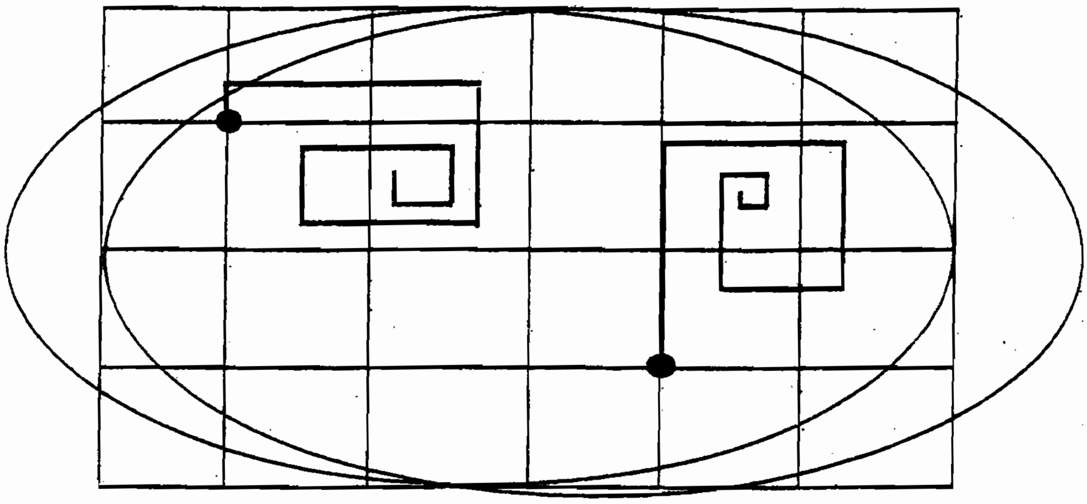
3. If $f(p^*) < 0$, no collision is detected.

4. If $f(p^*) \geq 0$, p^* gives coordinates of the collision point.

Admissible domain D :

1. Bounding boxes are projected onto three coordinate planes;
2. Projections intersections are detected in each plane;
3. Rectangular domain is detected in the space.

Search for an extremal point



- **Generate quasi-random LPτ points (or Sobol's sequences)**
  - points are placed randomly in the nodes of a rectangular grid;
  - N points guarantee $N^{-1/3}$ accuracy of point detection;

- **Start the spiral quadratic search from a random point**
  - successive one-dimensional quadratic searches;
  - a quadratic interpolant by three uniformly spaced points;
  - f(p) is required to be C^1 continuous;

- **Stop trials in the following cases:**
  - zero or positive value of function f(p) is found;
  - assigned number N of trials is exceeded;
  - p* is found with given accuracy.

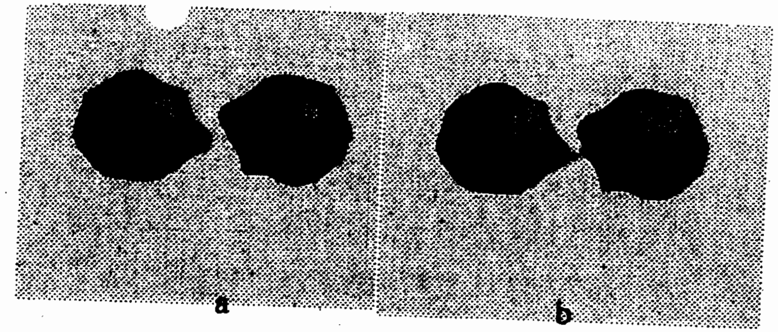
Internal zeroes

Internal zeroes are not acceptable for several reasons:

- Points of internal zeroes can be incorrectly classified as boundary points;

- Some operations on F-rep objects suppose distance-like behavior of the defining function and internal zeroes can cause incorrect results of these operations;

- The surface defined by the resulting function can be used in some applications including aesthetic design but a creased surface is not satisfactory.



Collision between two deformable objects: (a) worst case; (b) detected collision event.

Collision between fur strands