Using Real Functions with Application to Hair Modelling

A.Sourin[1], A.Pasko[2], V.Savchenko[2]

[1]School of Applied Science
Nanyang Technological University
Nanyang Avenue, Singapore 2263

[2]Laboratory of Shape Modelling
The University of Aizu
Aizu-Wakamatsu City, Fukushima, Japan

Abstract

In this paper, we define complex solids by real functions. Just from relatively small formulae we produce highly detailed complex objects and are able to manipulate and transform them producing more complex ones.

We show how complex static and time-dependent objects can be created with using of so-called *R*-functions. Then, we consider just one long-standing problems, hair modelling, and show how our functionally based model can be applied there.

Modelling hair we represent it with solid noise and subsequently unify it with the solid being made hairy. The hair and the solid are defined by real functions and the resultant hairy solid is in turn functionally defined and can be an argument for other operations. We are able to control length, thickness and curliness of hair and to obtain different hairstyles varying defining functions and applying set-theoretic operations to solid hair.

Using Real Functions with Application to Hair Modelling

A.Sourin[1], A.Pasko[2], V.Savchenko[2]

[1]School of Applied Science
Nanyang Technological University
Nanyang Avenue, Singapore 2263

[2]Laboratory of Shape Modelling
The University of Aizu
Aizu-Wakamatsu City, Fukushima,
Japan

1. Introduction

By now, the textured polygon mesh representation is basically used for rendering complex 3D scenes. The high processing speed of modern workstations allows them to generate thousands of textured antialiased polygons per pipeline, producing appearance of high complexity and detail. Basically, the database of a complex 3D scene consists of 3D polygon mesh and appropriate 2D textures for mapping onto the polygons. This approach achieves the appearance of the real objects but it is, nevertheless, still drawing a texture on an object's surface instead of real 3D texture modelling. The drawback of this approach is obvious when the observer approaches closer to the textured object in a 3D scene or when, for example, such as fur or woven materials have to be rendered. Real 3D texture and tiny details but not painted smooth surfaces are expected to be seen here. The solution to this problem is modelling 3D textures instead of painting smooth surfaces. Polygon mesh representation of 3D textures, obviously, involves tremendous data base increase. To avoid it, another methods of modelling should be applied. By now, the most successful

representatives of these methods are hypertextures and particle systems (Perlin[1], Reevs[2], Reevs[3], Kajiya[4]).

We consider functional approach to solid modelling as another alternative to polygon representation. This approach assumes producing highly detailed complex solids from a 'small formula'. In contrast with thousands of polygons, these relatively small formulae completely define objects and might be thought as DNA constituting the nature for organisms. An increasing interest in functional modelling has being observed in different fields of computer graphics for the last few years. The reason for this is that certain tasks in the process of geometric and solid modelling can be solved only with considerable effort when using polygon mesh representation, while they are solved relatively easily using the functional form. The typical examples of such tasks are constructing blending solids and an offset to a given solid (Woodwark[5], Hoffmann[6]).

This paper doesn't aim to cover the problem of functional solid modelling in general. On the example of our functional model, we study how to represent geometric solids with real functions and how to grow solid hair on complex solids.

2. Function representation of solid objects

In further considerations, we will represent solid objects as closed sets of 3D points with the defining function G: $f(x,y,z) \geq 0$, where G is some solid object, f is a real continuous function of points coordinates and $f > 0$ for points inside the solid, $f = 0$ for points on its boundary, and $f < 0$ for points outside the solid. Such representation is well-known in CAD (Hoffmann[6]), computer graphics and animation (Blinn[7], Bloomental[8], Essa[9]). To transform or deform the solid we just change its defining function. In other words, we consider these defining functions and operations on these functions when modelling geometric objects and operations on these objects. Defining functions of complex solids

can be created from a finite set of solid primitives or given defining functions with set-theoretic operations by CSG-like scheme. Generally speaking, solids defined in this manner are not regularized as it is required in Constructive Solid Geometry. One cannot distinguish between points on the boundary and internal points with f=0 which can arrive after applying set-theoretic operations. On the other hand, it gives more freedom for designing operations on objects. The exact analytical definitions of set-theoretic operations have been proposed by Rvachev in the theory of *R*-functions (Rvachev[10]) and applied for solving problems of mathematical physics. Applications of *R*-functions in solid modelling are described in Shapiro[11], Pasko[12], Pasko[13], Shapiro[14], Pasko[15]. In following considerations, $f_1 \wedge f_2$ stands for *R*-intersection, $f_1 \vee f_2$ means *R*-union, $f_1 \setminus f_2$ stands for *R*-subtraction and -*f* for *R*-complement.

For example, if geometric object G_1 is defined as $f_1(x,y,z)\geq 0$ and geometric object G_2 is defined as $f_2(x,y,z)\geq 0$ then the following *R*-functions are the most useful in practice.

Intersection $G_3 = G_1 \cap G_2$: $f_3 = f_1 \wedge f_2 = f_1 + f_2 - \sqrt{f_1^2 + f_2^2}$

Union $G_3 = G_1 \cup G_2$: $f_3 = f_1 \vee f_2 = f_1 + f_2 + \sqrt{f_1^2 + f_2^2}$

Complement $G_3 = \neg G_1$: $f_3 = -f_1$

Subtraction $G_3 = G_1 \setminus G_2$: $f_3 = f_1 \setminus f_2 = f_1 - f_2 - \sqrt{f_1^2 + f_2^2}$

These *R*-functions have C^1 discontinuity only in points where both arguments are equal to zero. Any desirable C^m continuity can be provided with another set of *R*-functions:

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

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

Another implementation of set-theroretic operations has been proposed by Ricci[16]:

$$f_1 \vee f_2 = \max(f_1, f_2)$$

$$f_1 \wedge f_2 = \min(f_1, f_2)$$

These functions are very simple and convenient for calculations but they have C^1 discontinuity when $f_1 = f_2$.

Considering defining functions of solids, we can easy make these solids moving introducing the forth variable - time t. Thus, general defining function of a moving solid is becoming $f(x,y,z,t) \geq 0$. We don't stipulate the user and allow him to choose any possible form for the representation of $f(x,y,z,t)$. But often, it is more convenient to use time-dependent affine transformations as a very familiar mathematical representation. And besides, these affine transformations can also be applied to the forth coordinate t. For example, time translation can be treated as bringing forward and postponing, time scaling - as acceleration and deceleration. More unusual operations with hardly predictable results can be created if we consider, for example, 4D rotation about 3D plane or scaling relative to some 4D point.

Elsewhere in Pasko[13], Pasko[15], different classes of functionally represented solids and operations on them are defined. These operations are common bijective mapping, affine and projection transformations, offsetting, blending, and metamorphosis. In this paper we consider just one class of advanced operations on functionally defined geometric solids - hair modelling.

3. Hair modelling

In this section we show how to grow solid texture on the solid object to make its look more natural. We have chosen hair modelling which is one of the most challenging problem in today's computer graphics. Just few researchers have published methods for rendering hair or fur. Examples of them can be found in Poulin[17], Chen[18], Watanabe[19], Anjyo[20], Daldegan[21]. In general, the problem of hair rendering comprises four problems to solve in order to produce realistic hair objects. They are hair

modelling, hair collision detection, hair motion and hair rendering. In this paper we basically consider only the first problem. Our method of hair modelling presented in this section was inspired by the furlike hypertextures (Perlin[1]).

Let us consider the continuous function **noise**(*x, y, z*) producing real values in some interval [*a*, *b*], where *a* and *b* are some real numbers and $a<0$, $b>0$. According to the previous section, the inequality **noise**(*x, y, z*)≥0 defines some geometric object in 3D Euclidean space. Thus, a geometric object defined by the noise function can be treated as any other object with the function representation. The examples of such a noise function can be found in Gardner[22], Perlin[1], Lewis[23]. Note that the set-theoretic operations defined by *R*-functions can be applied to this solid object in the case of the noise function continuity.

Suppose we want to grow fur or hair on some solid G_1 defined by functions $f(x, y, z)\geq 0$. In other words, we want to find out some function $\mathbf{hair}(x, y, z)$ defining the solid hair which then can be unified with the solid G_1. Thus, hairy solid G_3 is defined as follows

$$G_2 = \mathbf{hairstyle}(\mathbf{hair}(x, y, z));$$
$$G_3 = G_1 \cup G_2;$$

where **hairstyle** can be some complex set-theoretic or any other operation over the solid hair.

First we define some simple boundary volume to evaluate the function **noise** on it. It can be, for example, the sphere with the center at (x_0, y_0, z_0) located somewhere inside or even outside the object G_1. The radius of the sphere can be arbitrary but should be taken into account in further considerations. Then, for every given point (x, y, z) we find the intersection point (x', y', z') between the sphere and the ray cast from (x_0, y_0, z_0) towards (x, y, z) (see Fig. 1a). Then, the function $\mathbf{noise}(x', y', z')$ is evaluated and its value is considered as the value of the function **hair** in point (x, y, z) (see Fig. 1b). By now, we

have got solid noise defined for every given point (x, y, z) and for every point belonging to the same ray the same value of the function **noise** is assigned. Thus we actually obtained individual solid hairs or strands which are defined where the function **noise** is not negative. To thin the hair out, we apply some negative offset d to this function

$$\mathbf{hair}(x, y, z) = \mathbf{noise}(x', y', z') - d$$

which makes the noise more fragmentary because for some points this function **hair** is getting negative after the offset. Then this semi-infinite hair must be cut that may be done as R-intersection of the function **hair** and a function defining some solid with the desired boundary. And eventually we apply R-union to the result of this intersection and the function defining the solid head G_1. Thus we treat an individual hair as a thin long piece of the solid noise that is illustrated in Figures 1c,d. Some minor enlargements of the hairs' ends are caused by the spherical projection and can be cleverly avoided if variable offset d is used. Growing an individual hair strand can be also thought as extrusion or sweeping by a surface patch resulting from intersection of a bounding sphere with a solid noise.

In our implementation, we basically used the Gardner series (Gardner[22]) to define the function **noise**:

$noise(x, y, z)$
{
$Ser_x = a_1 \sin(a_2 x) + a_1 \sin\left(a_2 x + a_3 \sin(a_2 z)\right)$
$Ser_y = a_1 \sin(a_2 y) + a_1 \sin\left(a_2 x + a_3 \sin(a_2 x)\right)$
$Ser_z = a_1 \sin(a_2 z) + a_1 \sin\left(a_2 x + a_3 \sin(a_2 y)\right)$
$noise = Ser_x Ser_y Ser_z$
}

where parameter a_2 defines the thickness of every individual hair and parameters a_1 and a_3 are responsible for the overall shape of the solid noise. Varying value of the offset d one can obtain different number of hairs - from almost 'bald' scalp to very thick hair. Choice of the parameters a_1, a_2, a_3 is determined by the radius R of the bounding sphere. With increase of the radius R the number of individual hairs increases as $\sim R^2$ and they get thinner as $\sim 1/R$.

Applying spherical projection described above, just simple brush-like fur can be simulated. All hairs in that case are straight and point out from the center of the bounding sphere. It is illustrated in Figure 1d. Another projection by normal to the plane surface is used for simulation of the carpet (Figure 2a). In that case, to achieve the regular pattern and equal length of individual hairs, we applied the following form of the function **noise**

$$noise(x, y, z) = a_1 \sin(a_2 x) \sin(a_2 y) \sin(a_2 z)$$

The same projection by normal can be used to achieve better appearance of hairy concave surfaces. Actually, some physical or quasi-physical laws should be used to achieve the naturally-looking behavior of the long hair. We model this hair by more complex projection function. Thus, curly or wavy hair can be obtained if coordinates *x,y,z* are initially displaced before they are projected and substituted to the function **noise**:

$$x' = x + d_x(x, y, z)$$
$$y' = y + d_y(x, y, z)$$
$$z' = z + d_z(x, y, z)$$

For example, we used the following function for displacement of coordinates to get wavy hair (see Fig. 2b):

$$x' = x + a\sin(by)$$
$$y' = y + c\sin(dy)$$
$$z' = dz$$

In Figure 2c curly hair is obtained with increasing the frequency and reducing the amplitude of the displacement functions.

Long hair in Figure 2d is obtained by another displacement functions depending on the length of every individual hair.

Figures 2e-f show different examples of hairstyles obtained as a result of set-theoretic operations over solid hair. Thus, short cut style in Figure 2e is got by intersecting the solid hair with the object *f=f+offset*, where *f* is a function defining the solid head and *offset* controls the length of the hair. The letters cut are obtained as set theoretic subtraction of solid 3D letters from the hair before it was unified with the head.

The hairstyle in Figure 2f is done by intersection with the offset followed by subtracting the left, right and bottom parts of hairs followed by unifying with the head. The full function definition of this object can be found in Figure 3. *R*-functions mentioned in Section 2 were used there to implement set-theoretic operations. To obtain more sophisticated hairstyles we can subtract from the hair, for example, a sweep defined by a cutter moving around the head like it was done in Figure 4a and defined in Figure 4b where the hair G_1 was grown on the cube G_3 and then this hairy cube was cut by the moving object P_1 which is a sphere G_4 rolled around the cube G_3. The subtraction was done as set-theoretic subtraction of the sweep G_5 created by moving sphere P_1 from the hairy cube G_3. The sweep G_5 is obtained as *R*-union of three real functions defining the solid generator in the initial $f_{p1}(x, y, z, t_1)$ and final $f_{p1}(x, y, z, t_2)$ positions and the envelope $f_{env}(x, y, z)$ swept by the generator. The surface of the envelope has the following description

$$\begin{cases} f(x, y, z, t) = 0 \\ \dfrac{df(x, y, z, t)}{dt} = 0 \end{cases}$$

Commonly, the second equation has to be solved for t in terms of (x,y,z) and then this t should be substituted in the first equation to reduce the dimension of variables and to yield the description of the envelope (Wang[24], Martin[25]). But this way is possible only if the function derivative is expressed analytically. In our case, when the defining function can be defined procedurally, this approach cannot be applied.

To find the algorithm of the envelope defining function evaluation, we reformulated the above description. Zero value of the derivative there indicates that for given (x,y,z) the function f_{p1} takes an extremal value on the interval $[t_1,t_2]$ on the envelope surface. Taking into account negative function values outside the object, it can be stated that

$$f_{env}(x, y, z) = f_{p1}(x, y, z, t_0)$$
$$\text{where} \quad t_0 = \arg\max(f_{p1}(x, y, z, t)), \quad t \in [t_1, t_2]$$

To calculate a maximum value of function we used the Newton's interpolation.

The trajectory of the generator may have selfintersections and the sweep created may be involved in other operations closed upon our model. The details of the function representation for sweeping by a moving solid can be found in Sourin[26].

All pictures presented have been obtained on the Silicon Graphics Crimson workstation. We used very simple ray-casting program. On average, wall clock time spent for pictures in Figures 2a-f with the size 400x400 was about 5 minutes. For picture 3 - about 1 hour because of higher precision of computations used for the swept solid.

4. Conclusion remarks

In this paper we tried to show the benefits of the function representation for advanced operations on geometric solids. In our opinion, this, not actually new representation, can get another life because of just recently achieved fruitful alliance with *R*-functions. These functions allow one to represent set-theoretic operations in the functional form and get any desired continuity for the final function. Analytical functions or procedurally defined functions can both equally contribute to creation of complex solids and obey the same rules.

We shown how to define complex geometric solids in the functional form and how easy the hairy solids can be created. Modelling hair we were able to define long, curly and wavy hair with a solid noise. Different hairstyles can be obtained as a result of varying defining functions and applying set-theoretic operations to the hair. Comparing with the existing methods of hair modelling we would point out the simplicity and compactness of our model and the relative ease with which the same hair can be applied to different solids defined functionally. We didn't pay much attention to the realism of the pictures and the development of the good rendering software is our next problem to be solved.

The other tasks for further research are solving collision detection and hair motion problems using real functions.

References

1. K.Perlin, E.M.Hoffert, Hypertexture. *Computer Graphics*, 23:2, 253-262 (1989).
2. W.T.Reeves, Particle systems - A technique for modeling a class of fuzzy objects. *Computer Graphics*, 17:3, 359-376 (1989).
3. W.T.Reeves, R.Blau, Approximate and probabilistic algorithms for shading and rendering particle systems, *Computer Graphics*, 21:4, 313-322 (1985).

4. J.T.Kajiya, T.L.Kay, Rendering fur with three dimensional textures. *Computer Graphics*, 23:3,271-280 (1989).
5. J.R.Woodwark, Blends in Geometric Modeling. In: Martin RR (ed) The mathematics of surfaces II. Clarendon Press, Oxford, 255-297 (1987).
6. C.M.Hoffmann, Implicit curves and surfaces in CAGD. *IEEE Computer Graphics and Applications*, 13:1, 79-88 (1993).
7. J.F.Blinn, A generalization of algebraic surface drawing. *ACM Transactionon Graphics*, 1:3, 235-256 (1982).
8. J.Bloomental, B.Wyvill, Interactive techniques for implicit modeling. *Computer Graphics*, 24:2, 109-116 (1988).
9. I.A.Essa, S.Sclaroff, A.Pentland, A unified approach for physical and solid modeling for graphics and animation. EUROGRAPHICS'92, *Computer Graphics Forum*, 11:3, 129-138 (1992).
10. V.L.Rvachev, Methods of logic algebra in mathematical physics. Naukova Dumka Publishers, Kiev, (1974).
11. V.Shapiro, Theory of R-finctions and applications: a primer. TR CPA88-3, Cornell University, (1988)
12. A.A.Pasko, Conceptual and instrumental tools for a direct method of multidimensional geometric problems solving with a computer. PhD Dissertation, Moscow Engineering Physics Institute, Moscow, (1988).
13. A.A.Pasko, V.V.Savchenko, V.D.Adzhiev, A.I.Sourin, Multidimensional geometric modeling and visualization based on the function representation of objects. University of Aizu, Technical Report 93-1-008, (1993)
14. V.Shapiro, Real functions for representation of rigid solids, *Computer Aided Geometric Design*, 11:2, 153-175 (1994).

15. A.Pasko, V.Adzhiev, A.Sourin, V.Savchenko, Function Representation in Geometric Modeling: Concepts, Implementation and Applications, to appear in *The Visual Computer*.

16. Ricci A., A constructive geometry for computer graphics, *The Computer Journal*, 16:2,157-160 (1973).

17. P.Poulin,A.Fournier, A model for anisotropic reflection, *Computer Graphics*, 24:4,273-282 (1990)

18. H.Chen, En-Hua Wu, Radiosity for furry surfaces, Eurographics'91, Elsevier Science Publishers B.V. (North-Holland), 447-457 (1991).

19. Y.Watanabe, Y.Suenaga, A trigonal prism-based method for hair image generation, *IEEE Computer Graphics and Applications*, 12:1, 47-53 (1992).

20. Ken-ichi Anjyo, Y.Usami, T.Kurihara, A simple method for extracting the natural beauty of hair, *Computer Graphics*, 26:2, 111-120 (1992).

21. A.Daldegan, N.Magnenat-Thalmann, T.Kurihara, D.Thalmann, An integrated system for modeling, animating and rendering hair, Eurographics'93, Blackwell Publishers, Oxford UK, 211-221 (1993).

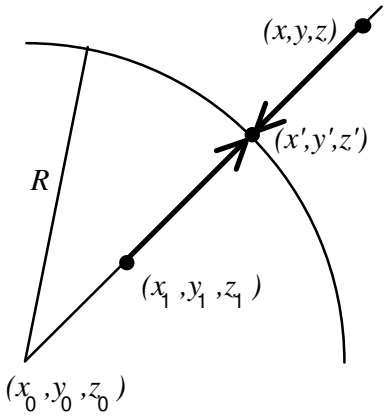
22. G.Gardner, Simulation of natural scenes using textured quadric surfaces, *Computer Graphics* 18:3,11-20 (1984)

23. J.P.Lewis, Algorithm for solid noise synthesis, *Computer Graphics*, 23:3,263-170, (1989).

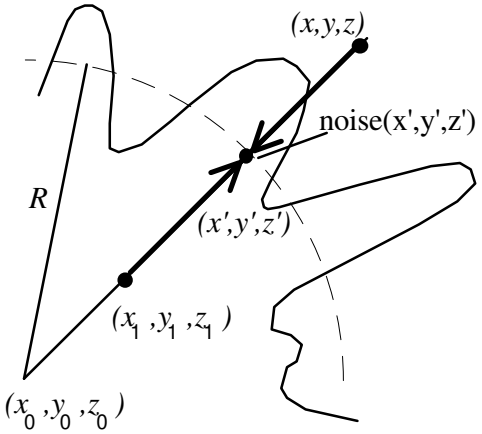
24. W.P.Wang, K.K.Wang, Geometric modeling for swept volume of moving solids, *IEEE Computer Graphics and Applications*,6,12,8-17 (1986).

25. R.R.Martin,P.C.Stephenson, Sweeping of three-dimensional objects, *Computer-Aided Design*, 22,4,223-234 (1990).

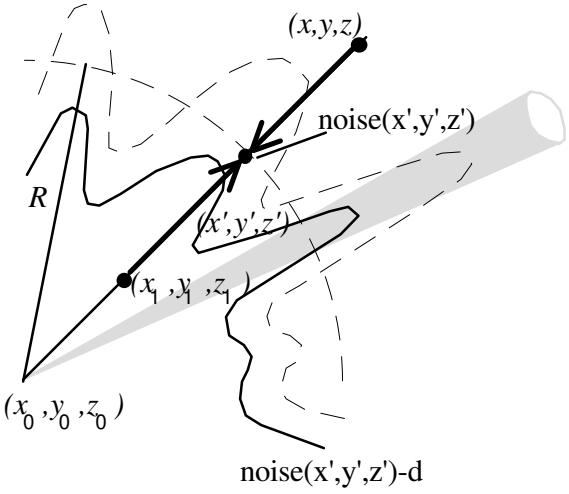
26. A.Sourin, A.Pasko, Time-dependent set-theoretic operations for functionally represented solid objects. IEEE Region 10's Ninth Annual Int.Conf TENCON'94, Singapore, 222-229, (1994)



a)



b)



c)

d)

Figure 1

$f_{hairy_head}(x, y, z)$

{

$f_{hairy_head} = f_{head} \vee f_{hair}$

$if\,(f_{head} \geq 0)\ then\ color = color_1$

$if\,(f_{hair} \setminus f_{head} \geq 0)\ then\ color = color_2$

}

$f_{head}(x, y, z)$

{

$color_1 = '\,yellow'$

$f_{head} = 1 - \left(\frac{x}{4}\right)^2 - \left(\frac{y}{6}\right)^2 - \left(\frac{z}{4}\right)^2$

$f_{hair_cut} = (f_{head} + hair_cut_offset)$

$f_{nouse} = 1 - \left(\frac{x}{1.27}\right)^2 - \left(\frac{y+2.5}{1.5}\right)^2 - \left(\frac{z-3}{1.27}\right)^2$

$f_{head} = f_{head} \vee f_{nouse}$

$f_{2eyes} = (1-(x+1.2)^2-(y+1)^2-(z-4)^2) \vee (1-(x-1.2)^2-(y+1)^2-(z-4)^2)$

$f_{2irises} = (0.3^2-(x+1.2)^2-(y+1)^2-(z-3)^2) \vee (0.3^2-(x-1.2)^2-(y+1)^2-(z-3)^2)$

$if\,(f_{2irises} \geq 0)\ then\ color_1 = '\,cyan'$

$f_{head} = f_{head} \setminus f_{2eyes}$

$f_{head} = f_{head} \vee f_{2irises}$

$f_{mouth} = \left(1 - \left(\frac{x}{2}\right)^2 - \left(\frac{y+4}{1.2}\right)^2 - \left(\frac{z-2.5}{1.2}\right)^2\right) \wedge (-y-4)$

$if\,(f_{mouth} \geq -0.1)\ then\ color_1 = '\,red'$

$f_{head} = f_{head} \setminus f_{mouth}$

}

$f_{hair}(x, y, z)$

{

$(x', y', z') = displacement(x, y, z)$

$(x'', y'', z'') = projection(x', y', z')$

$f_{hair} = noise(x'', y'', z'')$

$f_{hair} = f_{hair} - offset$

$f_{hair} = f_{hair} \wedge f_{hair_cut} \wedge (y'-2) \wedge (x'+1) \wedge (1-x')$

$color_2 = '\,violet'$

}

Figure 3

a)

$$G_1: f_1(x, y, z) = f_{hair}(x, y, z)$$
$$G_2: f_2(x, y, z) = f_{cube}(x, y, z) = (x - x_1) \wedge (x_2 - x) \wedge (y - y_1) \wedge (y_2 - y) \wedge (z - z_1) \wedge (z_2 - z)$$
$$G_3 = G_1 \cup G_2: f_3(x, y, z) = f_1 \vee f_2$$
$$G_4: f_4(x, y, z) = r^2 - x^2 - (y_2 - y)^2 - (z_2 - z)^2$$
$$P_4: f_{p4}(x, y, z, t) = f_4(x, y, z) \wedge (t - t_1) \wedge (t_2 - t)$$
$$P_1 = T_{trajectory}(P_4): f_{p1}(x, y, z, t) = f_{p4}(f^{-1}_{trajectory}(x, y, z, t))$$
$$G_5 = Sweep(P_1, t_1, t_2): f_{sweep}(x, y, z) = f_{p1}(x, y, z, t_1) \vee f_{p1}(x, y, z, t_2) \vee f_{env}(x, y, z)$$
$$G_6 = G_3 \setminus G_5: f_6(x, y, z) = f_3 \setminus f_{sweep}$$

b)

Figure 4

Captions:

Figure 1
Growing of hair strands

Figure 2
Examples of haircuts

Figure 3
Example of the function definition

Figure 4
A hairy cube cut by the sphere rolled around the cube