International Journal of Pure and Applied Mathematics
Volume 11 No. 1 2004, 1-19

AN INTRODUCTION TO AN ANALYTICAL WAY TO COMPUTE THE VOLUME OF BLOBS

Dominique Faudot[1 §], Gilles Gesquière[2], Lionel Garnier[3]
[1,2,3]LE2I, Informatique
Université de Bourgogne
BP 47870, 21078 Dijon Cedex, FRANCE
[1]e-mail: faudot@u-bourgogne.fr
[2]e-mail: gesquiere@u-bourgogne.fr
[3]e-mail: garnier@u-bourgogne.fr

Abstract: Using implicit surfaces is of interest in the field of interactive modelling. A first interest lies in automatic changes of topology. Volumic representation of objects is another interest. However, modelling non-compressible material with implicit surfaces remains difficult. The volume of implicit objects changes with the position of their centres. We can compute volume using numerical method, but these kinds of methods bring a quick but inaccurate result. In this paper, we propose an analytical method to obtain exact results. We present this method for two blobs. Afterward, we prove we can extend this method to several blobs.

AMS Subject Classification: 28A75, 51M25
Key Words: implicit surface, volume computation, analytical method

1. Introduction

These days, we are using computer graphics in several ways, e.g. mechanical tools building, physical phenomena simulations, or virtual objects animation.

Received: March 3, 2003
[§]Correspondence author

We often find two kinds of models to represent computer graphic objects. The first one is the B-Rep model, where the object is defined by its boundary [6], for instance a mesh. However, changes of topology during an animation are not easy to detect in general.

Another kind of representation is the volume representation model: a volume description defines the object, for instance an implicit or polyhydric surface. Implicit surfaces are interesting because it is not necessary to test changes of topology during an animation. Unfortunately, it remains of little use because using such kinds of surfaces leads to solving the problem of volume control during fusion or deformation of objects modelled with a non-compressible material [4]. In order to control the volume variation, we must compute volume when implicit objects parameters change. We may control volume variations through a proportional derived controller [2], [5]. These methods of control are based on a numerical volume computation, e.g. spatial enumeration or the territories method [4], [3].

Volume is just an approximation depending on the size of voxels or on the step of discretization. We would like to use an analytical solution to solve this seemingly complex and not yet solved problem. We are interested here in a particular kind of surfaces, which are equipotential[1] implicit surfaces called *blobs* [1].

A field function $f(x, y, z)$ defines an implicit surface and assigns a scalar value to each point in space. An implicit surface is then

$$\{(x, y, z)/f(x, y, z) = k\}\,,$$

where k is the threshold. For instance, Figure 1 is representing a blob. R, the radius of influence, defines the part of space where f influences points of space; r the effective radius defines the visible part of blob.

In our case, a blob B_i contains every point M of space, which satisfies:

$$B_i = \{M \in R^3/F_i(M) \geq T\}\,,$$

T, the threshold of the blob, regulates the size of the blob and F_i is an inverse exponential function [1]. An interesting property of this model is that it allows a simple formulation of fusion. The resulting equipotential function F is the sum of n functions

$$F(M) = \sum_{i=1}^{n} F_i(M)\,.$$

[1]An equipotential surface in 3D is a surface on which the value of a function $f(x, y, z)$ is a constant.

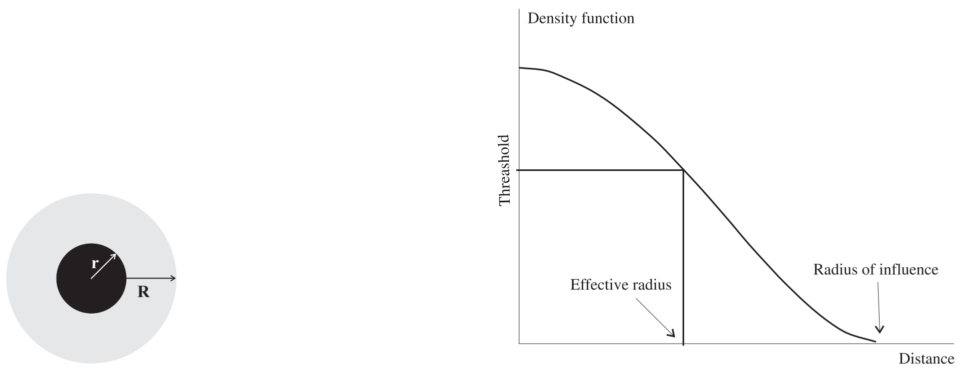


Figure 1: A blob

A function $F : R^3 \rightarrow R$ and the threshold $T > 0$ define an equipotential surface S satisfying:

$$S = \left\{ M \in R^3 / F(M) = T \right\} .$$

The surface S encloses the equipotential volume V that we now define by:

$$V = \left\{ M \in R^3 / F(M) \geq T \right\} ,$$

T regulates the size of the object composed of several blobs. With each blob B_i we associate a skeleton[2] Sk_i, a radius of influence R_i and an equipotential function $F : R^3 \rightarrow R$ monotonically decreasing with the distance to Sk_i. A skeleton may be a point, a segment, a curve or a surface. The equipotential function F_i is traditionally defined as the composition of two functions - the potential function $f_i : R^+ \rightarrow R$ and the distance function $d_i : R^3 \rightarrow R^+$ normalized by the radius of influence R_i

$$F_i(M) = f_i \left(\frac{d_i(M)}{R_i} \right) .$$

The distance function d_i creates the shape around the skeleton, representing the minimal distance between the skeleton and a point in space. We associate a distance $d_i(M)$ with any point M depending on Sk_i. The radius of influence R_i allows the control of the influence extent of the object.

In order to solve the Blinn function infinite support problem [1], Murakami [7], Nishimura [8] and Wyvill [13] proposed polynomial approximations. Other functions like those of Tsingos [12] allow better adjustments of the shape of

[2] A skeleton $Sk(X)$ of an object X is the set containing the centres of maximal balls.

the function f_i. The influence of the implicit objects on a point M of space is computed by summing the influence of each blob on M. Here the sum is used to merge blobs, but there are a lot of melting operations like those of Pasko [9] or Ricci [10].

In this paper we present an analytical volume computation for this particular kind of implicit surface defined by a density function. First, we explain the process for two blobs, and then demonstrate that the process may be extended to several blobs.

2. A First Observation for Two Blobs

So that the reader may understand our analytical computation for two blobs, four cases are identified, corresponding to changes of volume computation. We have illustrated each with a figure separated into two parts (see Figure 2). On the background, we describe the density function curves for the two blobs while on the foreground we represent the effective volume representation of these blobs (in black). The influence radius is drawn in light grey. A particular computation must be used in each following case:

Case 1. The two blobs are not in influence. Each blob is similar to a spherical volume object (see Figure 2(a)).

Case 2. There is an intersection between the influence radii. The influence of one blob implies the deformation of the other one (see Figure 2(b)).

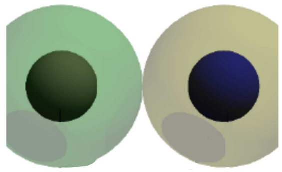
Case 3. The mutual influence grows. The two objects are melted together to give only one object (see Figure 2(c)).

Case 4. A blob totally influences the other one. We observe an ovoid object (see Figure 2(d)).

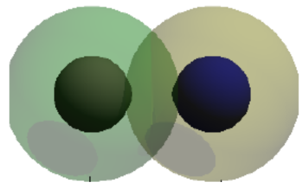
We will study these four different cases in this paper. We will use the items listed above to recognize the case we are in.

3. Volume of Two Blobs

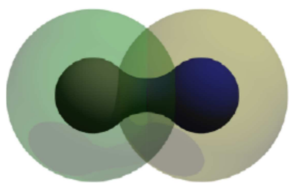
Simply let us assume, volume computation of blobs is just an integral computation whose bounds are unknown. Extracting bounds requires first studying the simple case, where two blobs are on one of the axes of the reference triad. If the two blobs are localized everywhere in space, we simply change the triad using rotation and translation to return to the reference triad.



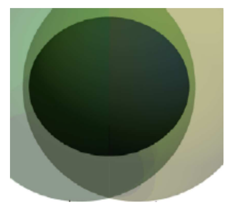
(a) The two blobs are merging. There is no influence



(b) Two separated parts with a deformation due to a mutual influence



(c) Only one object appears



(d) Only one object visually appears without any bumps

Figure 2: Fusion of two blobs

In the following, we will briefly study volume computation of a sphere, then volume of two blobs on an axis, and lastly we will extend the method to a generalized case.

3.1. Volume Computation of a Sphere

A sphere is considered as a blob. Let there be an axis passing through the centre of the blob (for example the z-axis). This axis intersects the blob at the two points -R and $+R$, where R is the effective radius. We consider the blob as a continuous set of disks perpendicular to this axis. It is easy to compute the area of each disk and then to integrate these areas according to the displacement. If the axis is the z-axis, radii of disks have to be expressed as a function of a z value. It appears that $r^2(z) = R^2 - z^2$, where $r(z)$ is the radius of a disk at a z value on the z-axis with $-R \leq r(z) \leq R$. As the centre of the disk is centre of symmetry, it follows that:

$$V = \int_{-R}^{R} \pi r^2(z) dz = 2 * \int_{0}^{R} \pi r^2(z) dz = 2 * \int_{0}^{R} \pi (R^2 - r^2) dz = \frac{4}{3} \pi R^3 \,.$$

After generalisation

$$V = 2 * \int_0^R \mathrm{Mes}\,(D(\Omega(z), r(z))dz\,,$$

where $\mathrm{Mes}(D(\Omega(z), r(z)))$ is the disk area whose centre is $\Omega(z)$ and radius (on the z-axis) is $r(z)$. Note that the volume of a sphere is easily computed with this method.

3.2. Two Blobs Centred on an Axis of the Reference Triad

Suppose the axis is the z-axis (the process is similar if we consider x- or y-axis). Let h and g be two functions: $h(z)$ gives radius variation as a function of z; $g(z)$ gives the intersection points of the implicit surface with the z-axis. Let there be two blobs: B_1 whose centre is $C_1(x_1, y_1, z_1)$, radius of influence R_1, and effective radius r_1; and B_2 whose centre is $C_2(x_2, y_2, z_2)$, radius of influence R_2, and effective radius r_2. It follows, therefore, four others functions.

Functions h_1 and g_1 are associated with the blob B_1, and h_2 and g_2 are associated with the blob B_2. Functions $h(z)$ and $g(z)$ correspond to the object modelled with the two blobs during merging. Values of these functions will depend on the threshold T and the parameters of the density functions that we use. The functions h and g will be studied later.

The volume of the implicit object is computed in three steps:

- Evaluating radii variation as a function of z (with function $h(z)$),

- Evaluating values on the axis for which $g(z) = 0$,

- Integration of the function found in the first step over the range computed in the previous step.

3.3. Computation of Radius

Let T be a common threshold between B_1 and B_2. Functions of density for blobs B_1 and B_2 are $f\left(\frac{r_1(z)}{R_1}\right)$ and $f\left(\frac{r_2(z)}{R_2}\right)$ in which r_1 and r_2 are computed with Pythagoras Theorem (see Figure 3). That is, $r_1(z)^2 = r^2(z) + (z - z_1)^2$ and $r_2(z)^2 = r^2(z) + (z - z_2)^2$. The distance $r(z)$ from a point of the blob to its projection on the z-axis depends on the localisation of the blob centres. Thus the equation describing an implicit surface is:

$$f\left(\frac{r_1(z)}{R_1}\right)+f\left(\frac{r_2(z)}{R_2}\right)=f\left(\frac{\left(r^2(z)+(z-z_1)^2\right)^{1/2}}{R_1}\right)$$
$$+f\left(\frac{\left(r^2(z)+(z-z_2)^2\right)^{1/2}}{R_2}\right)=T\,.$$

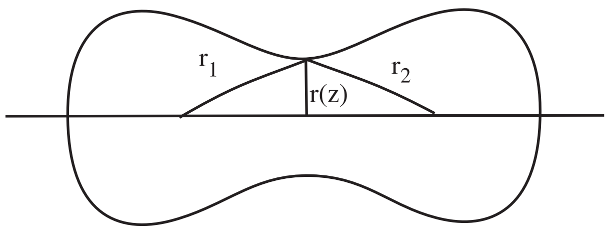


Figure 3: r_1 and r_2 are computed with Pythagoras Theorem

Let $h(z)=f\left(\frac{r_1(z)}{R_1}\right)+f\left(\frac{r_2(z)}{R_2}\right)-T$. We write the surface of the object as a function of radii and displacement along the z-axis. $h(z)=0$ represents the radius variation as a function of displacement. The intersection of the surface with the axis passing through the centres (here z-axis) is the locus of the point for which $r(z)=0$. If $r(z)$ is 0 in $h(z)$ then $g(z)=f\left(\frac{z-z_1}{R_1}\right)+f\left(\frac{z\ -z_2}{R_2}\right)-T$. Solving $g(z)=0$ is equivalent to finding intersections of the surface with the axis passing through the centres (here the z-axis). We can find in special parts of space (we will specify later) $g_1(z)=f\left(\frac{z-z_1}{R_1}\right)-T$ and $g_2(z)=f\left(\frac{z-z_2}{R_2}\right)-T$. Solving $g_1(z)=0$ (resp. $g_2(z)=0$) finds the intersection of the blob at left (resp. right) with the centres-axis if the two blobs are considered separately[3]. The radius is given by $h_1(z)=f\left(\frac{r_1(z)}{R_1}\right)$ and $h_2(z)=f\left(\frac{r_2(z)}{R_2}\right)$.

3.4. Case Studies

The functions $g(z)=0$ and $g_i(z)=0$ with $i=1,2$ give values to compute bounds of the integral and then to compute volume. However, roots number depends on the case during blending (see the four blending cases defined before). We have then to identify the case we are in.

[3]In this case we will see that the two blobs are not completely in mutual influence.

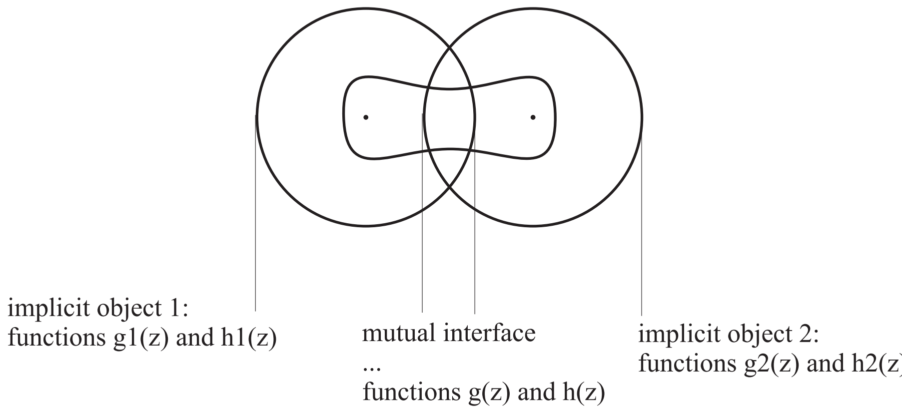


Figure 4: Definition of different functions

The radius depends on which case we are in. If influence between the two blobs is complete then only $h(z)$ is necessary. As soon as influence is no longer complete, the function that defines radius as a function of displacement is defined by pieces.

The left part of the left blob B_1 is defined thanks to $h_1(z) = f\left(\frac{r_1(z)}{R_1}\right) - T = 0$. At right of the right blob B_2, radius is given by $h_2(z) = f\left(\frac{r_2(z)}{R_2}\right) - T = 0$.

To identify the case we are in, let $h(z) = f\left(\frac{r_1(z)}{R_1}\right) + f\left(\frac{r_2(z)}{R_2}\right) - T$. This function gives variations of radii between the two blobs. With this, we can isolate the different cases enumerated before. Thanks to $g(z) = 0$ we find intersections of implicit object with centres-axis. Let α_1 and α_2 be the two real roots of $g(z) = 0$ if they exist. If $h\left(\frac{r_1(z)+r_2(z)}{2}\right) > 0$ then the object is ovoid and mutual influence is complete. In the other case, there are two separate, influenced, implicit components. If roots of $g(z) = 0$ are complex (not real) then $h(z)$ does not intersect centres-axis. Intersections α_1 and α_2 of implicit object with centres-axis are then found thanks to $g_1(z) = 0$ and $g_2(z) = 0$.

If $h\left(\frac{r_1(z)+r_2(z)}{2}\right) > 0$ then radius of object is given by $h_1(z) = 0$, $h(z) = 0$ and $h_2(z) = 0$. In the other case, there is no influence between blobs.

Case 1. The two blobs are not influenced. Roots are then computed independently for B_1 and B_2. We must solve $g_1(z) = 0$ and $g_2(z) = 0$. We have two spherical objects. Therefore, volume is just the sum of volumes of two blobs considered as spheres.

Case 2. (see Figure 5) The two blobs are in mutual influence. Solving

$g(z) = 0$ gives two roots and $g_1(z) = 0$ and $g_2(z) = 0$ give two other roots. The object is divided in four parts by the intersection of $h(z)$ with at first $h_1(z)$. Then $h_2(z)$ gives α_1 and α_2 as seen previously, and $g(z) = 0$ gives α_5 and α_6. These points correspond to the disappearance and appearance of the object. $g_1(z) = 0$ gives α_3, and $g_2(z) = 0$ gives α_4. It follows:

$$V = \int_{\alpha_3}^{\alpha_1} \pi \left(r_1^2(z)\right) dz + \int_{\alpha_1}^{\alpha_5} \pi \left(r^2(z)\right) dz + \int_{\alpha_6}^{\alpha_2} \pi \left(r^2(z)\right) dz + \int_{\alpha_2}^{\alpha_4} \pi \left(r_2^2(z)\right) dz \,.$$

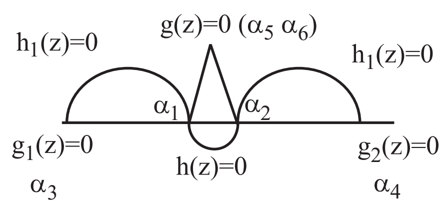


Figure 5: Case 2: influence is beginning

Case 3. The two blobs are in fusion. Just one object appears. However, neither blob totally influences the other. Two roots are computed from $g_1(z) = 0$ and $g_2(z) = 0$. The other roots are found with $g_1(z) = 0$, $h_1(z) = h(z)$ and $h_2(z) = h(z)$.

The radius function is in three pieces. It is necessary to know where each piece is changing. That means where to compute radius of the first blob, where to compute radius of the sum, and where to compute radius of the second blob. The intersections of $r(z)$ with the two blobs give us roots in the middle. We have written previously:

$$\begin{cases} g_1(z) = f\left(\frac{z-z_1}{R_1}\right) - T\,, \\ g_2(z) = f\left(\frac{z-z_2}{R_2}\right) - T\,, \\ h_1(z) = f\left(\frac{r_1(z)}{R_1}\right) - T\,, \\ h_2(z) = f\left(\frac{r_2(z)}{R_2}\right) - T\,, \\ h(z) = f\left(\frac{r_1(z)}{R_1}\right) + f\left(\frac{r_2(z)}{R_2}\right) - T\,. \end{cases}$$

We find $r(z), r_1(z)$ and $r_2(z)$ thanks to respectively $h(z) = 0, h_1(z) = 0$ and $h_2(z) = 0$. We have now to solve the system $\begin{cases} r_1(z) - r(z) = 0\,, \\ r_2(z) - r(z) = 0\,. \end{cases}$

We find two solutions of this system called α_1 and α_2. Opposite borders of the two blobs do not have influence on one another. Two roots are then computed due to the equations of the two blobs considered separately.

Solution of $g_1(z) = 0$ is α_3, and solution of $g_2(z) = 0$ is α_4. Therefore:

$$V = \int_{\alpha_3}^{\alpha_1} \pi \left(r_1^2(z)\right) dz + \int_{\alpha_1}^{\alpha_2} \pi \left(r^2(z)\right) dz \quad + \int_{\alpha_2}^{\alpha_4} \pi \left(r_2^2(z)\right) dz\,.$$

Case 4. There is only one ovoid object. Roots are found computing $g(z) = 0$.

We have seen previously that variation of radius is given by

$$h(z) = f\left(\tfrac{r_1(z)}{R_1}\right) + f\left(\tfrac{r_2(z)}{R_2}\right) - T.$$

Extracting roots is equivalent to considering $r(z) = 0$ in $h(z)$. Thus $g(z) = f\left(\frac{z-z_1}{R_1}\right) + f\left(\frac{z\ -z_2}{R_2}\right) - T$. At first, we must solve $g(z) = 0$ (for instance, for Murakami function we have to solve a 4-th degree polynomial). We obtain two real roots r_1 and r_2. Volume is then computed using displacement of a disk whose radius is $r(z)$ and integration over the interval $(\alpha_1,\ \alpha_2)$. The disk displacement $r(z)$ is found by solving $h(z) = 0$. Therefore $V = \int_{\alpha_1}^{\alpha_2} \pi r^2(z) dz$.

4. Generalisation to n Blobs

4.1. Minimizing a System of Equations

Before generalising for n blobs, we must study minimization of a system of equations.

Let f be a function of density defining blobs B_1 and B_2. Functions of density may be written

$$\begin{cases} f_1(x,y,z) = f\left(\dfrac{x-x_1}{R_1}, \dfrac{y-y_1}{R_1}, \dfrac{z-z_1}{R_1}\right)\,, \\ f_2(x,y,z) = f\left(\dfrac{x-x_2}{R_2}, \dfrac{y-y_2}{R_2}, \dfrac{z-z_2}{R_2}\right)\,. \end{cases}$$

Finding the isosurface is equivalent to finding the blob boundary hence

$$\{(x,y,z) \in B_1 \cup B_2 / f_1(x,y,z) + f_2(x,y,z) = T\}\,.$$

Let $f_2(x, y, z) = p$, $p \in]0, T[$. We are searching for each (x, y, z) verifying the last equality[4]. Finding the isosurface and solving the following parametric system are equivalent:

$$Sp_1 = \begin{cases} f_1(x, y, z) = T - p, \quad p \in]0, T[\ , \\ f_2(x, y, z) = p \, . \end{cases}$$

The problem is now a maxima-minima problem. Extremes may be obtained by squaring members of the system. We obtain

$$Sp_2 = \begin{cases} \min(f_1(x, y, z) - T + p)^2 \, , \\ \min(f_2(x, y, z) - p)^2 \, . \end{cases}$$

Systems Sp_1 and Sp_2 are not equivalent. Restrictions must be established to obtain solutions that are equivalent. If solutions of Sp_2 are such that the minimum is zero, these solutions are also solutions of Sp_1. Indeed, using Sp_2, we can find solutions of Sp_1 that are not on the border of one of the two blobs. A set of solutions vanishes when we simplify with $(f_2 - p)(f_1 - T + p)$ while solving the system. Let:

$$\begin{cases} F(x, y, z) = (f_1(x, y, z) - T + p)^2 \ , \\ G(x, y, z) = (f_2(x, y, z) - p)^2 \ . \end{cases}$$

Theorem 1 is used to minimize F and G simultaneously. The most interesting solutions are those for which the minimum is zero.

Theorem 1.[5] (see [11]) *Let f and g be two numerical functions of class C^1 over an open set U in a real normalized vector space E. Let $A = \{x \in U/g(x) = 0\}$. If the restriction f_A of f to A has an extremum at a point a such that $g'(a) \neq 0$, then there is a real number λ* such that *$f'(a) = \lambda g'(a)$ or using differential notations, $df_a = \lambda dg_a$.*

According to this theorem, there are solutions only if differentials of F and G are collinear

$$\begin{cases} \dfrac{\partial F}{\partial x} = \lambda \dfrac{\partial G}{\partial x} \, , \\ \dfrac{\partial F}{\partial y} = \lambda \dfrac{\partial G}{\partial y} \, , \\ \dfrac{\partial F}{\partial z} = \lambda \dfrac{\partial G}{\partial z} \, . \end{cases}$$

[4]Cases with $p = 0$ and $p = T$ are trivial.

[5]We are just working on the surface of implicit objects and not on the volume.

Let $F_i(x,\ y,\ z)$ and $G_i(x,\ y,\ z)$ be partial derivatives of F and G with respect to $i \in x, y, z$. We obtain:

$$\begin{cases} Fx\ *Gy &= Fy\ *Gx, \\ Fx\ *Gz &= Fz\ *Gx, \\ Fy\ *Gz &= Fz\ *Gy. \end{cases}$$

According to the above-mentioned theorem, we write equations (eq_1), (eq_2), and (eq_3) of Sp_2 hereafter. Equations (eq_4) and (eq_5) come from the inequalities[6]

$$\left(\frac{x-x_1}{R_1}\right)^2 + \left(\frac{y-y_1}{R_1}\right)^2 + \left(\frac{z-z_1}{R_1}\right)^2 < 1\,, \quad \text{and}$$
$$\left(\frac{x-x_2}{R_2}\right)^2 + \left(\frac{y-y_2}{R_2}\right)^2 + \left(\frac{z-z_2}{R_2}\right)^2 < 1\,,$$

$$Sp_2 \Leftrightarrow \begin{cases} Fx\ *Gy\ -\ Fy\ *Gx = 0\,, & (\mathrm{eq}_1) \\ Fx\ *Gz\ -\ Fz\ *Gx = 0\,, & (\mathrm{eq}_2) \\ Fy\ *Gz\ -\ Fz\ *Gy = 0\,, & (\mathrm{eq}_3) \\ \left(\frac{x-x_1}{R_1}\right)^2 + \left(\frac{y-y_1}{R_1}\right)^2 + \left(\frac{z-z_1}{R_1}\right)^2 - m = 0\,, & \\ \qquad m \in [0,1]\,, & (\mathrm{eq}_4) \\ \left(\frac{x-x_2}{R_2}\right)^2 + \left(\frac{y-y_2}{R_2}\right)^2 + \left(\frac{z-z_2}{R_2}\right)^2 - m = 0\,, & \\ \qquad m \in [0,1]\,. & (\mathrm{eq}_5) \end{cases}$$

Simplifying equations (eq_1), (eq_2), and (eq_3) with $f_1(x,y,z) - T + p$ and $f_2(x,y,z) - p$ gives (eq_{1a}), (eq_{2a}), and (eq_{3a}).

This implies $f_1(x,y,z) \neq 0$ and $f_2(x,y,z) \neq 0$. Roots on border of old blobs are not extracted and must be studied separately. We have to solve a system of 5 simultaneous equations $eq_{1a} = 0$, $eq_{2a} = 0$, $eq_{3a} = 0$, $eq_4 = 0$, and $eq_5 = 0$. Roots are functions of z:

$$\text{Solutions} = \begin{cases} x_{sol} = x(z)\,, \\ y_{sol} = y(z)\,, \\ z_{sol} = z(z)\,. \end{cases}$$

[6] Points of blob are in ball, which origin is centre of blob and which radius is radius of influence. Dividing by Ri, $i = 1, 2$ gives inequalities above.

We have to compute solutions of $f_1(x_{root},\ y_{root}, z_{root})+ f_2(x_{root},\ y_{root}, z_{root}) - T = 0$. Roots of this system are not on the border of the object, however they are known because the distance is greater than the sum of effective radius and the influence radius. Studying the density function gives number of solutions and computes roots on the border of the implicit object (as seen previously). These roots are studied independently. Either two or four real roots are extracted. In fact real roots are bounds of the integral used to compute volume of the implicit object. Therefore, if there are three real roots, two of them are not distinct. We must find the intermediate root and integrate over the extremes[7].

4.2. Extracting Missing Roots

Previous equation may have no real solution. Blobs are then very distant. Intersections of threshold with the curve of the sum of the two implicit functions and intersections of threshold with curves of blobs are the same. Roots may be extracted easily thanks to $f^{-1}\ (T) = r$. If the object shape is modified the intermediary roots are computed as previously.

Once roots are extracted and verified, volume is computed. Previous roots give intervals of the integral of the $h(z)$ function, which represents variation of the border as a function of z. If two roots α_1 and α_2 were extracted then:

$$v = \int_{\alpha_1}^{\alpha_2} h(z)dz\,.$$

If mutual influence between blobs is not complete, the object has to be considered in three parts: the non-influenced part of the first blob, the non-influenced part of second one (these two parts may be considered as spheres parts), and the part between the two others.

4.3. Application to n Blobs

Theorem 2. (Generalisation of Theorem 1) *Let $f,\ g_1,\ g_2, \cdots, g_m$ be real functions continuously derivable over an open U in a Euclidean space $\mathbb{R}^n$. Let a be a point in U verifying equations $g_1(a) = g_2(a) = ... = g_m(a) = 0$. If the linear derivative forms $g_1' = g_2' = ... = g_m' \in R$ are independent, a necessary condition to have a as either a maximum or a minimum relative to f over the subset A of Ω defined with equations $g_1(x) = g_2(x) = ... = g_m(x) = 0$ is that there are m*

[7] A double root is also solution of the derivative of studied polynomial.

real constants $\lambda_1, \lambda_2, ..., \lambda_m$ *such that* $f'(a) = \lambda_1 g'_1(a) + \lambda_2 g'_2(a) + ... + \lambda_m g'_m(a)$. λ_i *are Lagrange multipliers relative to the extreme* a.

This theorem says that generalisation to n blobs is allowed. Indeed, according to this theorem g_i, $i = 1 \cdots m$ have to be free. If workspace is $\mathbb{R}^n$, then m+1 vectors are dependent. Here we are in $\mathbb{R}^3$. We have at most four free constraints. Generalising a solution to n blobs is solving the system in a vectorial normalized space whose dimension is at least $n - 1$. This gap to an upper dimension is given in the next section.

Let F be a density function defining a blob. Let T be the threshold of the isosurface.

$$F : R^3 \to R \,,$$
$$(x, y, z) \to F(x, y, z), B = F^{-1} \left(\{\mathrm{T}\}\right).$$

B is a 2-surface object of $\mathbb{R}^3$ in $\mathbb{R}$, F is a submersion of $\mathbb{R}^3$ in $\mathbb{R}$. Now, we try to generalise to p blobs. For p blobs, we have $\tilde{F} : \mathbb{R}^{2+k} \to \mathbb{R}^k$.

Let $\tilde{B}$ be the implicit surface with

$$\tilde{B} = \widetilde{F^{-1}} \left(\{T, 0, 0, ..., 0\}\right).$$

We search $\tilde{F}$ with Rank$(D\tilde{F}) = k$. We choose

$$\tilde{F} : \mathbb{R}^3 \times \mathbb{R}^{k-1} \to \mathbb{R}^k \,,$$
$$(x, y, z, x_4, ..., x_{k+2}) \to \left(F(x, y, z), x_4, ..., x_{k+2}\right) .$$

The application defines the implicit. Let I_{k-1} be the identity application of $\mathbb{R}^{k-1}$. We have $\begin{array}{ll} I : \mathbb{R}^{k-1} \to \mathbb{R}^{k-1} \\ x \quad \to x \end{array}$.

The Jacobean of $\tilde{F}$ is $D\tilde{F} = \begin{pmatrix} DF & 0 \\ 0 & I_{k-1} \end{pmatrix}$.

We must check that the rank of the differential of $\tilde{F}$ is k. Since it is not a squared matrix, we cannot compute the determinant. We must extract a non-null determinant whose dimension is k. With this kind of matrix, we can extract a non-null determinant with dimensions $k - 1$ (the dimension of the identity matrix). If $\frac{\partial F_3}{\partial x} = 0$, a permutation of one of the three first lines or one of the three first column of $D\tilde{F}$, we can find a non-null term because rank$(D\tilde{F})$=1. So, rank$(D\tilde{F}) = k$. Let:

$$\Pi : \mathbb{R}^3 \times \mathbb{R}^{k-1} \to \mathbb{R}^3 \,,$$
$$x, y, z, x_4, ..., x_{k+2} \to (x, y, z) \,.$$

We have $\Pi(\tilde{B}) = B$. We have done a projection of $\mathbb{R}^{k+2}$ in $\mathbb{R}^3$. With the projection, we can find the isosurface of $\mathbb{R}^3$. We conclude that if we go in the $\mathbb{R}^{k+2}$ space, we can compute the isosurface given by $k+3$ blobs of $\mathbb{R}^3$. We could not compute the isosurface for a number of blobs greater than 4 with the linked maxima and minima method in $\mathbb{R}^3$. If we immerge the isosurface in a $\mathbb{R}^{k+2}$ space, we can compute the isosurface of $k+3$ blobs. Immerging the blobs in a space of higher size enables us to find roots (points on the border of the isosurface according to a given axis). When we study the case of two blobs, it is necessary to seek the roots of g, g_1, and g_2. In the case of n blobs, it is necessary to calculate the roots for the n joined blobs. We must then seek the roots for all the sets made up from $n-1$, $n-2$, $\ldots$, 1 blobs. The preceding paragraph demonstrates that the search of these roots is possible.

4.4. Volume Computation

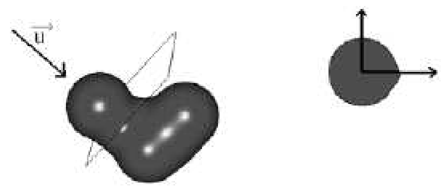
Knowing the roots of the blobs, we seek the border of the implicit surface in order to calculate its volume. Previously for two blobs, calculating h, h_1 and h_2 was enough. The values of these functions are used to move between the two centres along the axis as it were the radius of the disk. The continuous displacement of this disk between the roots computed previously enables volume computation. When the number of the blobs increases, the problem becomes more difficult to solve.

As an example in the case of three blobs (see Figure 6(a)), volume can no longer be calculated by cutting the figure with successive disks along an axis connecting two centres. The successive sections are not circular any more. Moreover the radius variations according to z are insufficient. For the same z, it is necessary to rewrite h according to a curved displacement in the plan perpendicular to displacement (see Figure 6(b)).

A second problem comes because the volume of a part of space should not be counted several times. Thus we choose to partition the surface[8] (see Figure 7(a)). According to encountered cases, this partition of space also enables establishing a placement strategy of the motion axis between the centres (see Figure 7(b)).

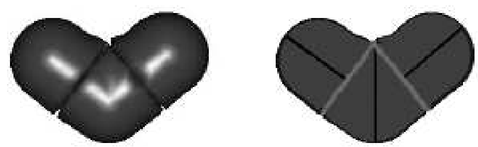
In conclusion, the complexity of the various cases to be managed leads us to stop in this way. For example, solving the partition of space problem would lead us to the implementation of Voronoi 3D. It would then be necessary to index all the roots and to know the profiles of the curves to calculate the radius for each case.

[8]For example, we can break up the implicit object into Voronoi partitions.



(a) (b)

Figure 6: Volume may not be computed with disk sections: (a) Implicit object composed of three blobs; (b) Cut of the implicit object (in the plan described in fig a) in the direction of u



(a) (b)

Figure 7: The three blobs must be separated in partitions: (a) The object is partitioned; (b) The partitions (separated with grey lines) make it possible to place the axis

4.5. Conclusion

It is possible to compute the volume of n, $n \geq 4$, blobs in an analytical way. The implicit surface is plunged in an upper dimension space. However, the resolution of this kind of system is difficult. The computation depends on the number of blobs, and it is necessary to solve the system when this number changes.

5. Numerical Results

We compared analytical results with those obtained using voxels[9]. Figure 8 shows that curves representing volume variations are similar. Maximum error is due to an undetected change of shape *(error* = 0.003%). To obtain such a result, voxels size is very small and then computation time with voxels very high. But computation time is not of interest. What we are looking for is just to demonstrate that such an analytical method is possible. We have to note that the two methods give identical numerical results.

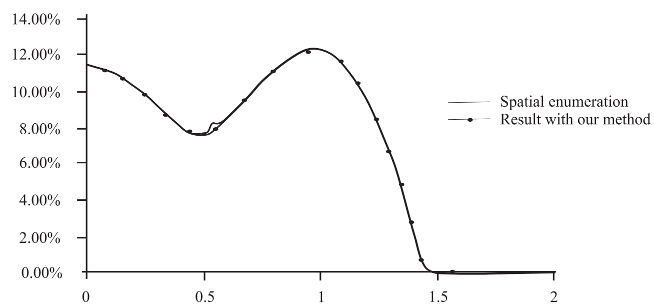


Figure 8: Comparison of results obtained with spatial enumeration and analytical method

Remark. Using all density functions is easy under the condition that the derivative is continuous. Since many of these polynomials are in pieces, we have then to compute other primitives. For instance, the function defined in Tsingos [10] is in two pieces.

6. Summary and Future Work

In this paper, we have presented a new method to compute volume of implicit objects in an analytical way. First, we introduced this method for two blobs localized on an axis of the reference triad. We said that we can generalize easily for two blobs localized anywhere in space. With this method we can compute the volume of a few number of blobs (if this number is less than or equal to 4). Using a linked maxima-minima method in R^3, we demonstrated that this method may be generalized to n blobs, $n \geq 4$. The number of constraints we can solve gives the number 4. If we choose to immerse the implicit surface

[9] Computation time is not of interest.

in an upper space dimension, this number of constraints can increase. We demonstrated it is possible to obtain volume of n blobs with $n-1$ constraints defined with functions of density. Dimension of working space must be at least $n-1$.

We can compute volume of blobs defined with different radii and different functions of density. The function must be anyone but the first derivative must be continuous. However, the use of function in pieces may be more complicated.

In future work, we would like to implement this extension, as it should be a good way to compute volume in an analytical way. We would like to use this method to compute implicit volume for physical measures or to control volume of blobs during fusion, too.

References

[1] J.F. Blinn, A generalization of algebraic surface drawing, *ACM Transactions on Graphics*, **1**, No. 3 (1982), 235–256.

[2] M.P. Cani-Gascuel, M. Desbrun, Animation of deformable models using implicit surfaces, *IEEE Transactions on Visualization and Computer Graphics*, **3**, No. 1 (1997), 39-50.

[3] M. Desbrun, N. Tsingos, M.P. Cani-Gascuel, Adaptive sampling of implicit surfaces for interactive modeling and animation. *Computer Graphics Forum*, **15**, No. 5 (1996), 319-325.

[4] G. Gesquière, Calcul et contrôle du volume d'objets implicites, In: *UFR des Sciences et Techniques*, Bourgogne, Dijon (2000), 147.

[5] G. Gesquière, D. Faudot, D. Rigaudière, Volume control of equipotential implicit surfaces, In: *Implicit Surfaces'99*, Bordeaux, France (1999).

[6] M.E. Mortenson, *Geometric Modeling*, John Wiley and Sons (1985).

[7] S. Muraki, Volumetric shape description of range data using blobby model, In: *Siggraph'91* (1991).

[8] T. Nishita, E. Nakamae, A method for displaying metaballs by Bézier clipping, *Computer Graphics Forum*, **13**, No. 3 (1994), C/271-C/280.

[9] A.A. Pasko, V.V. Savchenko, Blending operations for the functionally based constructive geometry, In: *CSG'94, Set Theoretic Solid Modeling: Techniques and Applications* (1994), 151–161.

[10] A. Ricci, A Constructive geometry for computer graphics, *The Computer Journal*, **16**, No. 2 (1973), 157–160.

[11] L. Schwarz, *Cours d'Analyse*, Tome 1, Hermann (1981).

[12] N. Tsingos, E. Bittar, M.P. Gascuel, Implicit surfaces for semi-automatic medical organs reconstruction, In: *Computer Graphics International'95* (1995).

[13] G. Wyvill, C. McPheeters, B. Wyvill, Data structure for soft objects, *The Visual Computer*, **2**, No. 4 (1986), 227-334.