Volume Sculpting with 4D Spline Volumes

Article · January 2000

CITATIONS
8

READS
361

3 authors:

Benjamin Schmitt
Digital Media Profesionals
26 PUBLICATIONS **278** CITATIONS
SEE PROFILE

Alexander Pasko
Skolkovo Institute of Science and Technology
200 PUBLICATIONS **2,661** CITATIONS
SEE PROFILE

Vladimir Savchenko
Hosei University
117 PUBLICATIONS **1,682** CITATIONS
SEE PROFILE

CISST'2000, International Conference on Imaging @Science, Systems, and Technology (Las Vegas, USA, June 26-29, 2000), Ed. Hamid Arabnia, @CSREA Press, vol. II, ISBN 1-892512-54-8, pp. 475-483.

Volume Sculpting with 4D Spline Volumes

Benjamin Schmitt[*], Maxim Kazakov[δ], Alexander Pasko[#], Vladimir Savchenko[#]

([*]) LaBRI. Laboratoire Bordelais d'Informatique
Université de Bordeaux I. 33405 Talence, France
Phone : +33 660 687 078 / Fax : +33 556 846 910

([δ]) Moscow Institute of Physics and Technology,
MIPT-7 Dolgoprudny, Moscow region. 141700 Russia
Phone / Fax : +7 (095) 408 72 27

([#]) Laboratory of Shape Modeling. Department of Computer Software,
Department of Digital Media Science
Faculty of Computer and Information Sciences
Hosei University Tokyo,184-8584 Japan

Abstract. *We present here an approach for interactive volume sculpting. The general framework used for modeling is the extended space mapping. A 4D spline volume is employed to define a 3D solid and a scalar field. While the first three coordinate are used to represent the spatial component of the volume to be sculpted, the fourth coordinate is used as a scalar, which corresponds to a function value or a volume density. Thus, the shape can be manipulated while changing the space-function coordinates. An interactive sculpting system is described. It supports global deformation using spline volumes, and local deformations implemented as real time carving.*

Keywords: Volume Modeling, Implicit Surface, Extended Space Mapping, B-spline, Sculpting

1. Introduction

The research in volume modelling with voxel data includes Boolean operations and linear transformations, transformation from one voxel data structure to another by manual 3D painting and carving [1], volume sculpting [15], metamorphosis [6], and morphological operations [10]. However, we also need more global smooth transformations providing non-linear deformations and topological changes of volumes.

Let us consider a volume as a subset of 3D space with an additional scalar value given in each of its points. If this scalar value is interpreted as an additional point co-ordinate, the volume becomes a 4D "height field" or a hypersurface in 4D space. The defined scalar field can be thought of as a variable of the object's density, temperature, etc. Scalar values can be specified in the nodes of a regular space grid (voxel data) or by a continuous function of three variables (so-called *implicit surface* model or more generally the *function representation* [12]).

Sculpting can be modelled by a combination of global and local deformations. Parametric representations of curves, surfaces and volumes are the most common ones used in modelling free-form shapes. Another way to represent complex shapes is to use implicit surfaces. One of traditional techniques is using skeleton-based implicit surfaces. To model an irregular shape with skeleton-based implicits, one should usually use many skeleton primitives.

In this paper, we propose modelling a complex shape with a single volume primitive and its global deformations. We define primitives by mixing two types of representations: parametric and implicit. One of the main advantages of the parametric form is that it leads to a design paradigm based on a control-point array, and the implicit form allows for the use of multiple geometric operations [14, 12]. This combination of these representations introduces new possibilities for interactive volume modelling.

We choose trivariate B-spline volumes as parametric functions due to their attractive properties for modelling. We use them to define solids and their corresponding implicit surfaces in 3D space. Similar approaches have already been proposed to construct piecewise algebraic

surfaces [13] and thick free-form shells [7]. In contrast, we wish to use parametric volumes in functionally based constructive modelling. The positions of the control points in the co-ordinate-function space are used to manipulate function values. In this sense, the proposed technique falls into the category of a general extended space mapping [14].

Local deformations are implemented using real-time carving on the approximate model (3D regular grid with real function values in the nodes). We use analytically defined convolution surfaces as a model for removing material in carving. The final model is represented as a volume with subtracted convolution surfaces. We describe an interactive sculpting system with which we are able to design complex 3D shapes.

2. Extended Space Mapping with Curves and Surfaces

A space mapping establishes a one-to-one correspondence between points of a given space and, if applied to some point set in the space, it changes this set to a different one. A mapping can be defined by the functional dependence between the new and old co-ordinates of points.

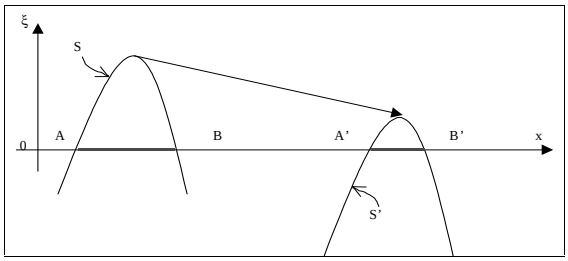

Figure 1: Transforming a 1D segment AB into A'B' by applying extended space mapping to a curve S defined as $\xi = f(x)$

Here, we discuss the example illustrated in Fig.1. The intention is to model a 1D geometric object (a line segment AB) on the x-axis, and then to transform it to A'B'. Let us introduce a function of one variable $\xi = f(x)$. It defines the segment AB as $f(x) \geq 0$. At the same time, it defines a curve S in the (x, ξ) plane. We call this plane an *extended space*, because it has a geometric co-ordinate and an additional function co-ordinate. The segment AB can be also considered as a projection of the part of the curve S, which is above the *x*-axis. Now, let us translate the curve S to S' as it is shown in Fig. 1. By doing this, we map the extended space (x, ξ) onto itself and thus define an *extended space mapping*. Now, the segment A'B' can be also considered as the projection of the part of the curve S', which is above the *x*-axis. Therefore, we transformed AB to A'B' with the single extended space mapping. Note that, to obtain this transformation with mappings of the *x* co-ordinate only, one has to think of some composition of 1D translation and scaling. The general definition of extended space mappings and details can be found in [14].

Let us now consider the case where the intention is to model a 2D solid as it is shown in Fig. 2. The surface S is a B-spline (parametric) function. It is defined in the (x, y, ξ) space, where $\xi = f(x, y)$. The 2D solid belongs to the (x, y) plane and is defined as the zero-contour line of the surface and its inside part, i.e. the projection of the positive part of the 3D surface onto the plane. The surface is defined parametrically, then, by moving its control points vertically (*function mapping*), along the ξ axis, one can define different extended space mappings, and transform the 2D solid. In the following sections we apply this approach to define and to deform 3D solids using parametric volumes in 4D extended space.

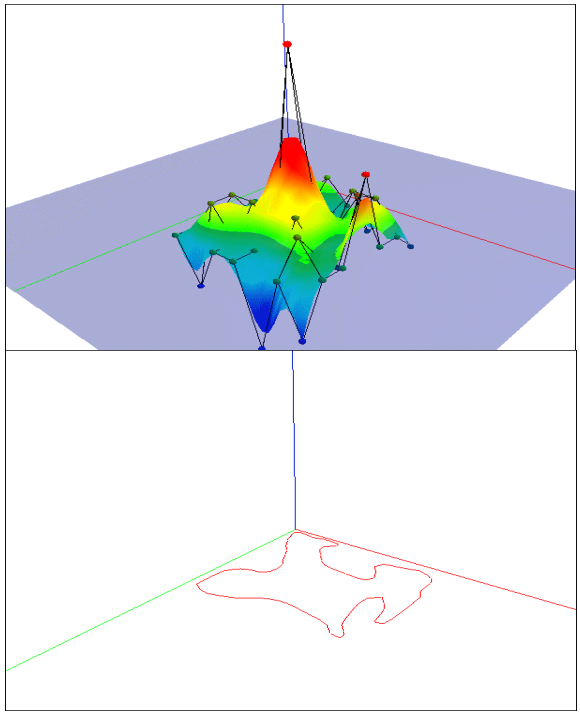

Figure 2: A parametric function (B-splines) in the extended 3D space and its corresponding 2D solid in the 2D space.

3. F-rep with Spline Volumes

A spline volume is defined with 3-dimensional control points [11], i.e. the *x*, *y* and *z* co-ordinates. We use the definition of F-reps, where a solid is defined by its defining function *f* and the inequality $f(x, y, z) \geq 0$, (and its boundary is defined by the equality). We follow this definition and apply it to parametric volumes ([8]). In the parameter space defined by $\{(x,y,z) : 0 \leq (x,y,z) \leq 1\}$, we assume that the *x*,*y* and *z* co-ordinates of the surface $S(u,v,w)$ can be expressed as a regular grid, i.e. $P^x_{ijk} = \frac{i}{l}, P^y_{ijk} = \frac{j}{m}$ *and* $P^z_{ijk} = \frac{k}{n}$, where P is a control point of the spline, and *l*,*m*,*n* are the number of control points on each axis.

We add one more dimension to those points, the ξ co-ordinate, corresponding to the function value. Then, we define a 3D solid as $f(x,y,z) = S^{\xi}(u,v,w)$. Hence, we can modify the 3D solid by changing the ξ co-ordinate of control points' volume. This is a direct application of the function mapping defined with the extended space mapping. In addition, one can move control points in (x, y, z) space and thus deform the 3D solid in a different way. This requires implementation of the inverse space mapping described elsewhere [8].

3.1. Functional Clipping

One undesirable property of the B-spline volume, which is due to the used parametric function, is that there is no control over the behaviour of the volume outside the unit cube domain. In order to overcome this, we introduce functional clipping. Fig. 3, left part, shows a solid, based on a B-spline parametric function, which is bounded by the unit cube. The aim was to create a solid inside this domain. As it can be observed, the behaviour of the parametric function outside the domain contradicts the requirement that there be only negative values of a defining function for points outside the defined object. This results in undesirable ("ghost") solids.

Let $S(u,v,w)$ be a function defining a parametric volume where $S(u,v,w) = S^{\xi}(u,v,w)$. Consider the intersection between the unit cube and the 3D solid defined by this function:

$$S_{clip}(u,v,w) = S(u,v,w) \,\&\, F_s(u,v,w)$$

where $F_s(u,v,w) = F_b(u) \,\&\, F_b(v) \,\&\, F_b(w)$ is a defining function of the unit-cube with F_b defining the unit strip by each variable:

$$F_b(t) = (1-t)t\,,$$

and & is the symbol of the intersection operation defined by the R-function [12] as:

$$G = g_1 \,\&\, g_2 = g_1 + g_2 - \sqrt{g_1^2 + g_2^2}\,.$$

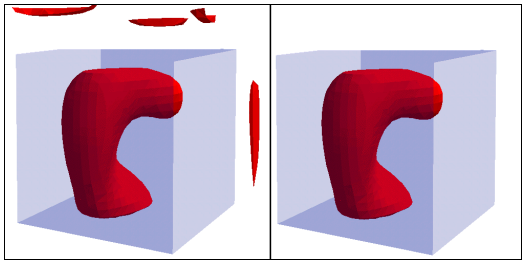

Figure 3: (Left) The unit cube, the desired solid inside it, and "ghost" solids outside. (Right) The same solid after functional clipping

The result of this procedure is illustrated in Fig. 3, right part. The defining function is negative outside the domain (no "ghosts") and the desirable 3D solid is unchanged.

In our implementation, the procedures defining parametric volumes encapsulate the introduced functional clipping. Once a solid is modelled, this allows us to apply any operation provided by a F-rep modeller to it.

At this stage, we are able to model a complex shape with a single primitive.

4. Interactive Sculpting

4.1. Visual Representation of Control Points

Each control point P_{ijk} is visually represented by a sphere. Its position is centred in $\left(P^x_{ijk}, P^y_{ijk}, P^z_{ijk}\right)$ using the usual 3D space with its three axis. In order to represent the function value of the control point P^{ξ}_{ijk}, we first define a colour scale function to the value, and apply it to the control points. A complete grid is shown in Fig. 4. In the case of a NURBS volume, the radius of the sphere allows us to represent the homogeneous co-ordinate P^w_{ijk} of the control point (kind of 5D). When its absolute value increases, the radius of the sphere increases. And either the character "+" or "-" is then mapped on the sphere, according to the sign of the co-ordinate.

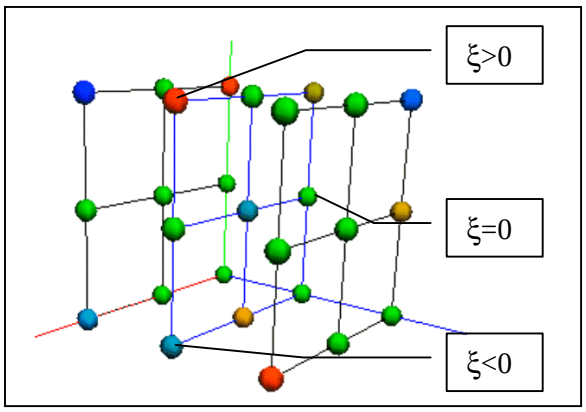

Figure 4: 3D Grid of colored control points of a B-spline volume.

4.2. Modeling Interface

We have developed an interactive modeller using Tcl/Tk for the interface and MAM/VRS for graphics. In this section we illustrate main features of the modeller. The main idea in modelling a volume is to deal with slices of it. A volume is defined by a grid of $l \times m \times n$ control points. To change their function values, the user first selects a slice of the volume in a grid similar to Fig. 4, i.e. a patch defined by a $l \times m$ grid. This patch is then represented in a separate window (left part Fig. 5). To change the function value of a control point the user drags the corresponding sphere along the axis perpendicular to the patch. At the same time, the corresponding zero-contour is drawn (right part Fig. 5). Finally, the volume can be rendered as a polygonal model (Fig 5.).

4.3. Interactivity

When one models a solid, he/she defines parameters that allow for a global speed up of the calculations. For example, the grid density for the solid is known, and will not be changed while dragging control points. According to the general formula of B-spline, all the coefficients can be evaluated before this operation. So, we may say that this part of the modeller is completely interactive. Three different ways exist to render the solid. The first one is direct polygonisation included in the modeller, which is the fastest one because is based on the previous described calculation. The two other ways is to use a script, generated by the modeller when one saves its current model, and call HyperFun [3], which is a high-level F-rep modelling language. A default polygonisation or a raytraced image can then be rendered.

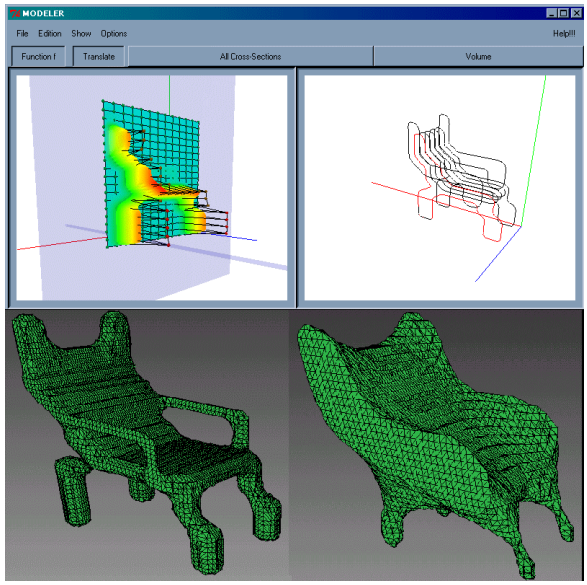

Figure 5: A selected slice of a B-spline volume. By moving a control point along the cylinder (upper-left), the corresponding zero-contour (red) is updated (upper-right). The lower-left part shows the corresponding polygonised surface. The lower-right part shows the result with a Bézier volume using the same set of control points.

4.4. Comparison between solids based on Bézier, B-spline and NURBS functions

A NURBS volume is the most general case of the parametric functions we have used. Three different parameters can be used to modify it: the order on each axis, the homogeneous co-ordinate of the control points and, of course, the function value. Because of this it allows the largest degree of freedom.

A Bézier volume can be compared with a B-Spline volume, where the order on each axis is maximum, i.e., is equal to the number of control points on each axis. When modifying a volume, the deformations are global. With the case of the B-spline, the possibility of defining the order of the volume allows to do more local deformations (i.e., less global, but still influencing the entire volume). A comparison is shown in Fig. 5 (lower part).

Finally the NURBS functions add one more way of modification, that of changing the weight of each control point. But the effect has some restrictions. First, if a control point is inside the solid, with a positive function value, the effect of changing its weight can be considered as null. On the other hand, if the control point is close to the boundary of the solid, i.e. the last control point with a positive value, the effect is locally important. If the

control point has negative value, the effect is locally important too, depending of course on the order of the volume. From the interactive modelling point of view, there is no big difference between NURBS and B-spline functions, because for each of the results obtained with NURBS, we were able to get an equivalent one result with B-splines.

4.5. About intuitive modelling

The aim of this paper is to present a modeller which should be able to model complex shapes. But one can wonder if the way of modelling is intuitive. Dealing with slices of volume is very simple. Dragging control points along the ξ axis and watching the resulting zero-contour at the same time illustrate clearly what deformations are occurring to the solid. Moreover, the primitive based on spline is defined in a 4D space and by the way, the user does not have to care about the topology of his model.

The main restriction of the presented modeller comes from the geometric properties of the chosen splines. In fact, if one choose a low degree for the spline, an undesirable effect may appear, comparable as small swellings. In this case, the expected result is not intuitive, and may be disappointing. This effect is shown in Fig 6. (left part). In this figure, swellings have been amplified in order to have a better visualisation of the problem.

To avoid this, one may increase this degree. But the counterpart is that the modification of a control point becomes more global (Fig 6. right part, and previous sub-section). Another way is to change one by one all the control points. But this task is impossible to realise when the grid contains a large number of control points.

An interesting way to overcome this problem may be the use of multiresolution, based on wavelets. At a low level of details, i.e. a grid with a few control points, to model the global shape becomes easy and the previous problem disappears. It should be then possible to add details to the shape when one increases the resolution.

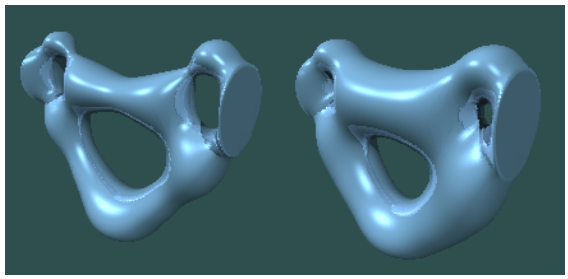

Figure 6: The upper part is a ray traced surface of a solid defined with a B-spline of degree 3, and the lower part, the same solid with a degree 5. The swellings disappeared.

5. Real Time Carving: Local Deformation

We present here our methods for sculpting using local deformations of solids with the primitive previously defined. Carving stands for modifying the solid locally in its contact area with an instrument (cutter).

The main problem with this approach is the speed of the rendering. Methods of near real time implicit surface polygonisation are usually modifications of the base Marching Cubes algorithms [5], and derivatives (see, for example, [2, 16]). But the huge number of polygons is still too important for an interactive visualisation.

Our approach uses hierarchical implicit surface reconstruction, based on an octree data structure [17], as a way to achieve a good interactive visualisation times [16]. The data-set bounding the volume is recursively subdivided into eight subvolumes (cells) until the unit cell size is reached. The detalisation criterion determines the cell's hierarchy level at which the surface should be constructed. Because of possible cracks between neighbour cells from different detalisation levels [2, 17], a special stitching procedure is applied to the cells' surface patches. This idea is shared with the one described in [17], but a substantial increase in the frame rate can be reached as a result of its combination with the minimisation of processing efforts during the implicit surface extraction and polygonisation. We minimise the volume of the data-set to process and introduce maximal caching of a constructed surface mesh after the detalisation criterion's modifications. Such an approach results in a substantial difference from [17]. A complete description of our method and its implementation would require a separate paper,

however here we give a brief account of our optimisations as follows:

- only the visible part of the data-set (i.e., one that lies within the viewing frustum) is processed for surface extraction;
- instead of dropping the whole polygonised surface for each movement of the observer (practically every frame under some circumstances) we interactively update the visible surface part processing data-set regions that come into the viewing frustum;
- localised modifications are made to the reconstructed surface according to updates in the detalisation criterion value (i.e., only those parts of a surface that belong to the data subset where required level of detail has changed are updated);
- localised modifications in the surface are made according to updates in the data-set (e.g., as a result of carving).

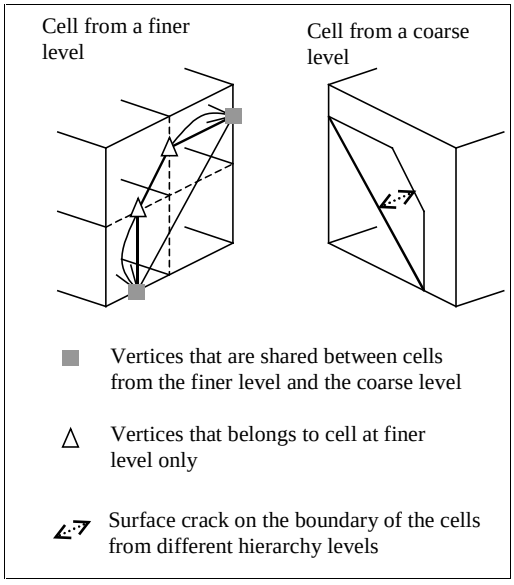

Figure 7: Boundary vertices from finer levels are moved to corresponding vertices at a coarse level thus stitching surface patches in cells of different size

A simple scheme for stitching the neighbouring surface patches generated at different levels of detail was used. The idea at its core is illustrated in Fig. 7. The implementation uses lookup tables for fast performance

To facilitate navigation inside the resulting polygonal model, we deploy a mechanical model for camera movement as described in [4]. This model is similar to underwater submarine navigation in which a repulsive force from the isosurface is experienced. Our implementation differs from [4] in the way the repulsive force is calculated, regarding to the gradient of the scalar field associated with the scalar data-set near the camera position.

A fish-like object was prepared using the global deformation tool and exported to the interactive local deformation tool (Fig 8.). A ray-traced image of the implicit surface for this data-set is shown at Fig. 9. Then eyes, mouth, and some features on its body were carved.

We selected analytically defined convolution surfaces [9] as the model for removing material in carving.

Thus, the cutter's trajectory information was exported into an HyperFun script [3]. Finally, the model is represented as a volume with subtracted convolution surfaces.

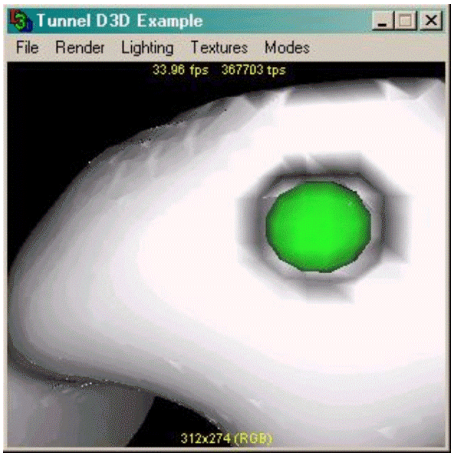

Figure 8. Surface view during carving a fish's eye.

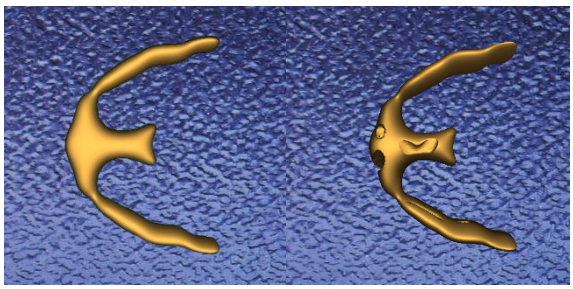

Figure 9. Ray-traced Fish data-set (left) with eyes and mouth carved (right).

Conclusion

We have developed a new technique for interactive volume sculpting using parametrically defined free-form volumes. Even though such volume models are widely used, there are difficulties with interactively modifying them. In order to overcome these difficulties, we use parametrically defined volumes with functional clipping to define global deformations. For local deformations, we use real time carving.

We implemented interactive sculpting software based on the proposed solutions. We extended the F-rep library of the HyperFun language [3] with primitives based on the Bézier, B-spline and NURBS parametric functions. We also have implemented software for real time carving. The resulting object is a volume with subtracted convolution surfaces. Our interactive tool generates the corresponding model in the HyperFun language. In this way, the results of sculpting can be used in F-rep modelling along with other primitives and operations.

References

[1] T.A. Galyean, J.F. Hughes, Sculpting: An interactive volumetric modeling technique, *SIGGRAPH'91, Computer Graphics Proceedings*, 25(4), 267-274 (1991).

[2] T. He, L. Hong, A. Varshney, S. Wang, Controlled topology simplification, *IEEE* Trans. on Visualization & Computer Graphics, vol. 2, No. 2, 1996, pp. 171-184.

[3] HyperFun, http://www.hyperfun.org.

[4] L. Hong, S.Muraki, A. Kaufman, D. Bartz, and T.He, Virtual Voyage: interactive navigation in the human colon, *Proc. Ann. Conf. Series SIGGRAPH'97*, 1997, pp. 27-34.

[5] W. Lorensen, H. Cline, Marching Cubes: A high resolution 3D surface construction algorithm, *Computer Graphics*, vol. 21, No. 4, 1987, pp. 163-169.

[6] A. Lerios, C.D. Garfinkle, M. Levoy, Feature-based volume metamorphosis, *SIGGRAPH'95, Computer Graphics Proceedings*, 449-456 (1995).

[7] J.P. Menon. Constructive shell representations for free-form surfaces and solids, *IEEE Computer Graphics and Applications*, vol. 14, No. 2, 1994, pp. 24-27.

[8] K.T. Miura, A.A. Pasko, V.V. Savchenko, Parametric patches and volumes in the functional representation of geometric solids, *CSG 96 Conference Proceedings*, Information Geometers, Winchester, UK, 1996, pp. 217-231.

[9] J. McCormack, A. Sherstyuk, Creating and rendering convolution surfaces, Computer Graphics Forum, vol. 17, No. 2, 1998, pp. 113-120.

[10] N. Ozawa, I. Fujishiro, A morphological approach to volume synthesis of weathered stones, *Volume Graphics workshop '99*, Swansea, UK, vol. II, 207-220 (1999).

[11] L.A. Piegl, Rational B-spline curves and surfaces for CAD and graphics, *State of Art in Computer Graphics*, Springer-Verlad, 1991, pp. 225-269.

[12] A.Pasko, V. Adzhiev, A. Sourin and V. Savchenko, Function representation in geometric modeling: concepts, implementation and applications, *The Visual Computer*, vol.11, No.8, 1995, pp. 429-446.

[13] T.W. Sederberg. Piecewise algebraic surface patches, *Computer Aided Geometric Design* vol. 2, 1985, pp. 53-59.

[14] V. Savchenko, A. Pasko, Transformation of functionally defined shapes by extended space mappings. *The Visual Computer*, vol. 14, No. 5/6, 1998, pp. 257-270.

[15] S. Wang and A. Kaufman, Volume sculpting, *Symposium on Interactive 3D Graphics*, ACM Press, 151-156 (1995).

[16] R. Westermann, L. Kobbelt, T. Ertl, Real-time exploration of regular volume data by adaptive reconstruction of iso-surfaces, *The Visual Computer*, vol. 15, 1999, pp. 100-111.

[17] J. Wilhelms and A. Van Gelder, Octree for faster isosurface generation, *ACM Computer Graphics*, vol. 24, No. 5, 1990, pp. 57-62.