Algebraic Representation of CSG Solids Built from Free-Form Primitives

Á.L. García, J. Ruiz de Miras, F.R. Feito
Departamento de Informática. Universidad de Jaén
Paraje Las Lagunillas s/n
23071, Jaén. Spain
{algarcia, demiras, ffeito}@ujaen.es

ABSTRACT
A mathematical model for free-form solid modelling was presented in previous published works. The key aspects of this model are the decomposition of the volume occupied by the solid into non-disjoint cells, and the representation of the solid as an algebraic sum of these cells. Here we apply this scheme to represent CSG solids built by combining free-form solids in boolean operations. As a proof of the validity of this scheme, we present an algorithm that allows us to visualize the non-evaluated result of the operations. We have worked with free-form solids whose surfaces are bounded by a set of low degree triangular parametric patches.

Keywords
Cell decomposition, boolean operations representation, free-form solid modelling.

1. INTRODUCTION
Solid modelling can be defined as a consistent set of principles for mathematical and computer modelling of three-dimensional solids [Sha02]. One open issue in solid modelling is the representation in an exact, useful and efficient way of solids bounded by free-form elements (curves in 2D, curved surfaces in 3D, etc.). These are usually named free-form solids.

Operations like visualization, boolean combination, surface determination, collision detection and so on, depend on the representation scheme used for solids; some schemes enhance the performance of some operations, while others are intended to be as versatile as possible.

The Extended Simplicial Chain (ESC) model was presented as a mathematical model to represent free-form solids in a simple and robust way by means of an algebraic decomposition [RF99b, GRF03]. Based on this scheme, algorithms have been developed to test the inclusion of a point [RF99a, GRF04], obtain a B-Rep or a voxel representation [RF02] of free-form solids.

Here we apply the ESC model to develop a method to obtain an algebraic decomposition of CSG solids built by applying boolean operations to a set of free-form primitives bounded by triangular Bézier patches [Far86]. Notice that the ESC model is not only a boundary representation, but also a volume representation scheme.

WSCG'2005 Short papers proceedings. ISBN 80-903100-9-5
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.

2. PREVIOUS WORK
Constructive Solid Geometry (CSG) is one of the most popular representation schemes for solid modelling [RV77]. CSG solids are built as boolean combinations of simpler solids (primitives). The classic data structure associated with this scheme is a binary tree, where the non-terminal nodes represent boolean operations, and the terminal nodes store either primitives or transformations.

There appear problems when applying the CSG approach to free-form solid modelling, mainly related to boundary evaluation and visualization. Using a reduced set of free-form primitives and exact computation [Key00] is one way to solve this; other solutions get approximate results with variable cost and precision [BKZ01].

The visualization of CSG solids is another important field of research. Many different approaches have been studied: using polyhedral approximations (lossy precision) [RS97], complex mathematical developments (numerical stability problems) [KGMM97], ray shooting techniques (computationally expensive) [HG96], or modifying the rendering pipeline (memory expensive and complexity) [SLJ02].

Feito et al. proposed the algebraic decomposition of polyhedral solids in previous works [FR98]; this was the Simplicial Chain (SC) model. Ruiz et al. extended the SC model to free-form solids, this is the Extended Simplicial Chain (ESC) model [RF99b]. The application of this scheme to free-form solids bounded by triangular Bézier patches has been recently developed [GRF03].

3. THE ESC MODEL
The Extended Simplicial Chain (ESC) model use a *divide and conquer* approach to model free-form solids. The solids are decomposed into a set of non-disjoint *extended cells*

of two types: simplices and free-form cells. Each cell has an associated sign; the volume of positive cells is added to the solid, and the volume of negative cells is subtracted from the solid. This decomposition scheme allows us to develop algorithms just by combining in a sum the results of computing the test locally in the individual cells.

The ESC model is intended for representing general free-form solids. Up to now, the model has been successfully applied to 2D solids bounded by conics and Bézier curves [RF97], and 3D solids bounded by triangular algebraic patches [RF99b, Rui01] and triangular Bézier patches [GRF03].

In this work, the starting point to construct the ESC associated to a 3D free-form solid is a triangle mesh that approximates the shape of the solid. The surface of the solid is built from these triangles [VPBM01, HB03].

Simplices

A d-dimensional simplex is defined as the convex hull of $d+1$ affinely independent points. We use 2D simplices for 2D solids, 3D simplices for 3D solids and so on.

In our model, each 3D simplex is built by joining one arbitrary point (e.g. the origin of coordinates) with the vertices of each triangle. These are named *original* simplices.

We compute the associated sign of a 3D simplex as the sign of its signed volume, which can be computed as:

$$\frac{1}{6}\cdot\begin{vmatrix} T_1.x & T_2.x & T_3.x & T_4.x \\ T_1.y & T_2.y & T_3.y & T_4.y \\ T_1.z & T_2.z & T_3.z & T_4.z \\ 1 & 1 & 1 & 1 \end{vmatrix}$$

T_1, T_2, T_3, T_4 being the vertices of the simplex.

Figure 1 shows two original simplices (tetrahedra).

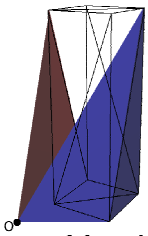

Figure 1. Wireframe model, positive (blue) and negative (red) original simplices

Free-form Cells

A general d-dimensional free-form cell (ffc) is defined as a set of points in $\mathbf{R}^d$, obtained as the intersection of the half-spaces defined by a free-form element of dimension d, and one or more planar elements of dimension $(d-1)$ that verifies a number of properties ensuring closure and conectivity (see [RF99b, GRF03] for details).

The ffc as defined in the bibliography needs to be specified in more detail, as the number and computation of the planar elements depend on the type of free-form elements chosen to bound the solids. This time, we are working with triangular Bézier patches [Far86], and the planar elements necessary to completely close the ffc are the plane containing the base triangle of the main patch of the cell and three additional planes (called *associated planes*); each plane contains one of the edges of the base triangle of the main patch, and is parallel to the weighted average vector of the normal vectors from the two triangles that share that edge. An ffc is bounded by the mentioned planar elements, the main patch, and all the patches that share a vertex with the main patch and intersect the planar elements [GRF04].

Figure 2 left shows the bounding elements of an ffc. The associated planes are drawn as quadrilaterals; the main triangular Bézier patch of the ffc is drawn in shaded mode, and one neighbour patch that also bounds the ffc is drawn in wireframe mode. Figure 2 right shows the ffc as a solid.

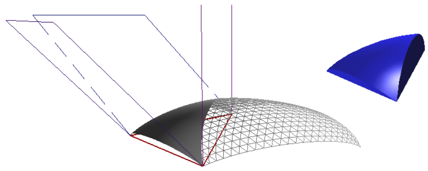

Figure 2. Left: bounding elements of an ffc. Right: solid view of the same ffc

Considering the triangle vertices disposed counterclockwise, an ffc is positive if it is in the positive half-space defined by the plane that contains the base triangle, and negative if it is in the negative one. If the patches pass through the plane that contains the base triangle, each connected component is considered a different ffc.

Extended Simplicial Chains

A (3-dimensional) Extended Simplicial Chain, δ, is defined as $\delta = \sum_i \alpha_i \cdot E_i$. E_i being extended cells, each of them multiplied by an integer α_i (its coefficient). There is an integer-valued function *associated* to each ESC, which is defined in the following way:

$$f_\delta : \mathbf{R}^3 \to \mathbf{Z}; \qquad f_\delta(\mathbf{Q}) = \sum_{\mathbf{Q}\in E_i} \alpha_i$$

For a point **Q** of the space, f_δ is the sum of the coefficients associated with the extended cells that contain **Q**.

The associated function of an ESC is different from zero only for those points that belong to the free-form solid represented by δ (noted FF_δ). There is a simple point in solid test based on this property [RF99a, GRF04].

The same solid can be represented by several ESC's. We define a *normal extended simplicial chain* as the one that verifies $f_\delta(\mathbf{Q}) = 1 \quad \forall \mathbf{Q} \in FF_\delta$.

From now on, we will consider only normal ESC's.

3.3.1 Operating with ESC's

Let $\delta = \sum_{i=1}^{n} \alpha_i \cdot E_i$ and $\delta' = \sum_{j=1}^{m} \alpha_j \cdot E_j$ be two ESC's, and λ a scalar value. The sum of two ESC's, and the product of an ESC by a scalar value are defined as:

$$\delta + \delta' = \sum_{i=1}^{n} \alpha_i \cdot E_i + \sum_{j=1}^{m} \alpha_j \cdot E_j;$$

$$\lambda \cdot \delta = \sum_{i=1}^{n} (\lambda \cdot \alpha_i) \cdot E_i;$$

The regularization of these operations is described in [RF99b, Rui01, GRF03]. We consider always regularized operations.

4. REPRESENTATION OF BOOLEAN OPERATIONS

Using the ESC model, boolean operations can be represented applying the *divide and conquer* approach to reduce them to a combination of operations with extended cells, which are much simpler than the original solids.

Let us note the intersection of extended cells E and E' as *ExCell(E∩E')*. Computing *ExCell(E∩E')* involves computing the intersection of the bounding elements of E and E'. It would be also necessary to define a new type of extended cell to represent the sets of points bounded by fragments of the bounding elments of E and E'. At the moment, we are working on this topic. An alternative option for this process consists of decomposing the intersection in an approximative way, using smaller simplices and ffc's to represent the resulting sets.

Using ESC's to represent solids and boolean operations allows us to delay the computation of *ExCell(E∩E')* depending on how the result will be applied. As the volume of positive and negative cells often compensates, the evaluation of *ExCell(E∩E')* can be avoided for cells whose intersection do not add volume to the final solid.

Now we present the expressions for the representation of boolean operations. Theorems establishing these formulas and their demonstrations can be found in [Rui01, GRF03].

Let FF_1 and FF_2 be two free-form solids, and let $\delta_1 = \sum_{i=1}^{n} \alpha_i \cdot E_i$, and $\delta_2 = \sum_{j=1}^{m} \alpha_j \cdot E_j$ be their associated normal ESC's. The associated normal ESC for the solid obtained as the intersection $FF_\delta = FF_1 \cap FF_2$ is:

$$\delta = \sum_{i=1}^{n} \sum_{j=1}^{m} (\alpha_i \cdot \alpha_j) \cdot ExCell(E_i \cap E_j);$$

In practice, if there is no intersection between one E_i and one E_j, $ExCell(E_i \cap E_j)$ is an empty set, and therefore it can be deleted from the sum.

The union and difference operations can be expressed using the intersection:

Let $\delta_{FF1 \cap FF2}$ be the ESC of the intersection of FF_1 and FF_2. The normal ESC for the solid obtained as the union $FF_\delta = FF_1 \cup FF_2$, is $\delta = \delta_1 + \delta_2 - \delta_{FF_1 \cap FF_2}$, the normal ESC of the difference of FF_1 and FF_2, noted $FF_\delta = FF_1 - FF_2$ is $\delta = \delta_1 - \delta_{FF_1 \cap FF_2}$, and finally, the extended simplicial chain for the solid obtained by applying the complement operation on a free-form solid, noted FF_1^C is $\delta_C = \delta_R - \delta_1$, where δ_R verify $f_{\delta_R} = 1 \ \forall \mathbf{Q} \in \mathbf{R}^3$. In practice, δ_R can be implemented as a cube that contains the whole scene.

5. REPRESENTATION OF CSG SOLIDS

The expressions from section 4 allow us to build an ESC for the result of a boolean operation between two free-form solids represented by ESC's. Since the result of this operation is another ESC, it can be used again as an operand in a new boolean operation. If we repeat this process, we can obtain a chain for the final result of a whole tree of boolean operations to design a complex CSG solid.

Figure 3 shows an example of a CSG tree where the leaf nodes are free-form solids represented by ESC's.

By applying the ESC model, we can obtain an homogeneous representation for both free-form primitives and CSG solids, and algorithms designed for free-form solids with ESC's can be easily applied to CSG solids [RF02].

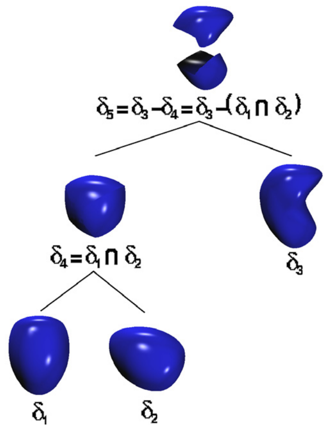

Figure 3. Example of CSG tree

As it was mentioned in section 4, the intersection of extended cells is a very important issue to be considered when creating the ESC that represents the result of a boolean operation. This can be handled in two ways:

- Compute the intersection of the cells and construct a new ESC that represents the resulting set of points. This can be done by defining a new type of extended cell to represent the sets of points, or by decomposing the result in simplices and ffc's. The second choice can result in a huge number of cells that can make the resulting ESC useless, while the first one seems more efficient.
- Do not evaluate the intersection, but keep a pointer to each extended cell instead. In this case, when applying an operation on an intersection of cells it is necessary to operate on both cells, and then combine the results.

Proceeding as explained in the first option will produce a completely evaluated CSG solid. The second option corresponds to a classic non-evaluated CSG solid.

6. VISUALIZATION OF NON-EVALUATED CSG SOLIDS

Based on the point in solid test mentioned in section 3.3, we have developed a visualization algorithm for solids based on ray tracing techniques [Gla93]. The algorithm consists basically of the following steps:

- Cast a ray through the centre of each pixel.
- Compute all the intersections of the ray with each cell.
- Use the point in solid test to find the first intersection point that belongs to the solid.
- Use the colour properties of that intersection point to draw the corresponding pixel as usual.

To render images of a non-evaluated CSG solid, we just have to consider the second option from section 5 to represent the intersection of extended cells. To compute the intersection of a ray with *ExCell(E∩E')*, simply compute the intersections of the ray with *E* and *E'*, and then study the position of the intersection points of *E* with regard to *E'* and vice versa. A point will be considered as an intersection with *ExCell(E∩E')* only if it belongs to *E* and *E'*.

Figure 4 shows a pair of free-form solids and the visualization of the boolean operations applied to them.

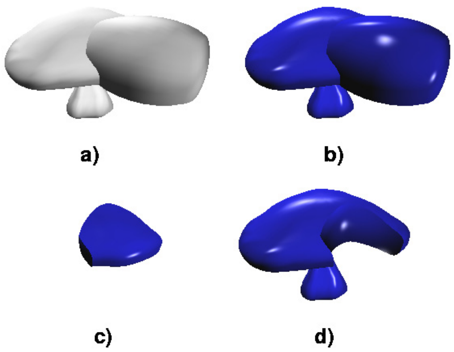

Figure 4. a) Two free-form solids. b) Union. c) Intersection. d) Difference

The visualization algorithm also makes it possible to get partial images of a given solid. This is possible by passing portions of the ESC as arguments for the visualization algorithm. Doing this allows us to observe how the positive and negative cells compensate to obtain the final result.

7. CONCLUSIONS

The application of a mathematical model to the representation of CSG solids built from free-form primitives has been presented. This model makes it possible to decompose the solids in an algebraic expression, which allows a simple and uniform handling of the primitives and the final solids just by working with the individual cells and then combining the results. This decomposition considers the solid not only as its boundary, but also as a volume, and the cells are allowed to intersect with other cells. This feature makes the decomposition process very simple.

The utility of the model to develop robust and simple algorithms has been proved in previous published works [RF99a, RF02, GRF04], and also in this paper, where an algorithm to render images of non-evaluated CSG solids based on ray shooting techniques has been presented.

8. ACKNOWLEDGEMENTS

This work has been partially supported by the Ministry of Science and Technology of Spain and the European Union by means of the ERDF funds, under the research projects TIC2001-2099-C03-03 and TIN2004-06326-C03-03.

The work of Á.L. García was supported by a Junta de Andalucía (Spain) fellowship.

9. REFERENCES

[BKZ01] Biermann, H. Kristjansson, D. Zorin, D. Approximate boolean operations on free-form solids. In ACM Siggraph conf. proc., Los Angeles, ACM Press, pp. 185-194. 2001

[Far86] Farin, G. Triangular Bernstein-Bézier patches. *Computer Aided Geometric Design*, Vol. 3, 83-127. 1986.

[FR98] Feito, F.R. Rivero, M. Geometric modelling based on simplicial chains. *Computers & Graphics*, Vol. 22 (5), 611-619. 1998.

[Gla93] Glassner, A. (Ed.). An Introduction to Ray Tracing. Academic Press, 1993.

[GRF03] García, Á.L. Ruiz de Miras, J. Feito, F.R. Free-form solid modelling based on extended simplicial chains using triangular Bézier patches. *Computers & Graphics*, Vol. 27 (1), 27-39. 2003.

[GRF04] García, Á.L. Ruiz de Miras, J. Feito, F.R. Point in solid test for free-form solids defined with triangular Bézier patches. *The Visual Computer*, Vol. 20 (5), 298-313. 2004.

[HB03] Hahmann, S. Bonneau, G.-P. Polynomial surfaces interpolating arbitrary triangulations. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 9 (1), 99-109. 2003.

[HG96] Hasenfratz, J.-M. Ghazanfarpour, D. Rendering CSG scenes with general antialiasing. In CSG 96 Set-theoretic Solid Modelling: Techniques and Applications. pp. 275-289. Informations Geometers Ltd. 1996.

[Key00] Keyser, J. Exact Boundary Evaluation for Curved Solids. PhD Thesis, University of North Carolina at Chapel Hill. 2000.

[KGMM97] Krishnan, S. Gopi, M. Manocha, D. Mine, M. Interactive boundary computation of boolean combinations of sculptured solids. *Computer Graphics Forum*, Vol. 16 (3), 67-78. 1997.

[RF97] Ruiz de Miras, J. Feito, F.R. Inclusion test for curved edge polygons. *Computers & Graphics*, Vol. 21 (6), 815-824. 1997.

[RF99a] Ruiz de Miras, J. Feito, F.R. Inclusion test for free-form solids. *Computers & Graphics*, Vol. 23 (2), 255-268. 1999.

[RF99b] Ruiz de Miras, J. Feito, F.R. Mathematical free-form solid modeling based on extended simplicial chains. *Journal of WSCG*, Vol. 7, 241-248. 1999.

[RF02] Ruiz de Miras, J. Feito, F.R. Direct and robust voxelization and poligonalization of free-form CSG solids. 3DVPT 2002: 1st International Symposium on 3D Data Processing, Visualization and Transmission. 2002.

[RS97] Rappoport, A. Spitz, S. Interactive boolean operations for conceptual design of 3-d solids. ACM 24th Conference on Computer Graphics & Interactive Techniques. 269-278. 1997.

[Rui01] Ruiz de Miras, J. Free-Form Solid Modeling. PhD Thesis, Universidad de Granada. 2001.

[RV77] Requicha, A. Voelcker, H. Constructive Solid Geometry. Tech. Memo 25, Production Automation Project, University of Rochester. 1977.

[Sha02] Shapiro, V. Solid modeling. In Farin, G. Hoschek, J. Kim, M.-S. (eds.) Handbook of Computer Aided Geometric Design. North-Holland, 2002.

[SLJ02] Stewart, N. Leach, G. John, S. Linear-time CSG rendering of intersected convex objects. *Journal of WSCG*, Vol. 10, 437-444. 2002.

[VPBM01] Vlachos, A. Peters, J. Boyd, C. Mitchell, J. Curved PN triangles. ACM Symposium on Interactive 3D Graphics, 159-166. 2001.