

CSG Operations with Voxelized Solids

Pavol Novotný

Comenius University in Bratislava

Faculty of Mathematics, Physics and Informatics
Mlynská dolina, SK-842 48 Bratislava 4, Slovakia

novotny@pauli.fmph.uniba.sk

Leonid I. Dimitrov

Austrian Academy of Sciences

Commission for Scientific Visualization
Donau-City-Str. 1, A-1220 Vienna, Austria

leonid.dimitrov@oeaw.ac.at

Miloš Šrámek

Austrian Academy of Sciences

Commission for Scientific Visualization
Donau-City-Str. 1, A-1220 Vienna, Austria

milos.sramek@oeaw.ac.at

Abstract

We present a new technique for CSG operations with voxelized geometric objects, which are represented by truncated discrete distance fields supplemented by additional information about the surface normal. The technique removes artifacts of straightforward volumetric CSG operations by taking into account conditions for object representability, according to which sharp details are not correctly representable in discrete distance fields. The proposed technique solves this shortcoming by rounding edges and other sharp details. It works at the voxel level without the necessity for reconstruction of continuous object models.

1 Introduction

In this paper we present a new method for performing CSG operations with voxelized objects, which are represented by the technique called *truncated distance fields* (TDF) [5]. The TDF representation differs from the full signed distance field (DF) one in that the distances are stored only in a thin layer in the surface vicinity and constant values are stored otherwise. The main problem is that the result of CSG operation is often a solid with edges, which does not fulfill the morphologic criterion for suitability of an object for voxelization at a given resolution, introduced by Bærentzen *et al* [2]. According to it, a surface of an object can be successfully reconstructed from the corresponding distance field, if the object is both S_r -open and S_r -closed in respect to a spherical structuring element

with radius r . In other words, one can roll a sphere with radius r on both inner and outer side of the whole object's surface, in such a way as to have everywhere only a single common point with the object—the surface contact point. It is obvious, that suitable solids do not have any sharp details, and if they do, the details have to be smoothed out before voxelization. The radius r of the sphere is defined by the size of the reconstruction filter. If only the DF is stored, r is defined by the gradient reconstruction kernel $r = \sqrt{6}$. If the gradient of the field is stored along with the distances, r is determined by the size of the trilinear reconstruction kernel, $r = \sqrt{3}$.

Complex solids obtained by CSG operations (intersection, union and difference) on solid primitives are typical representatives of the category of solids with sharp details. The goal of this paper is to propose such an implementation of the CSG operations between *voxelized* primitives, which delivers correct results from the point of view of the aforementioned representability criterion and, moreover, which does not necessitate the reconstruction of continuous models of the involved primitives.

2 CSG Operations

2.1 Simple CSG Operation

CSG operations between binary voxelized models can be easily performed on the voxel level as operations between the values of the corresponding voxels [4]. For example, the *minmax* implementation in the case of DF models:

$$A \cap B : d_{A \cap B} = \min(d_A, d_B)$$

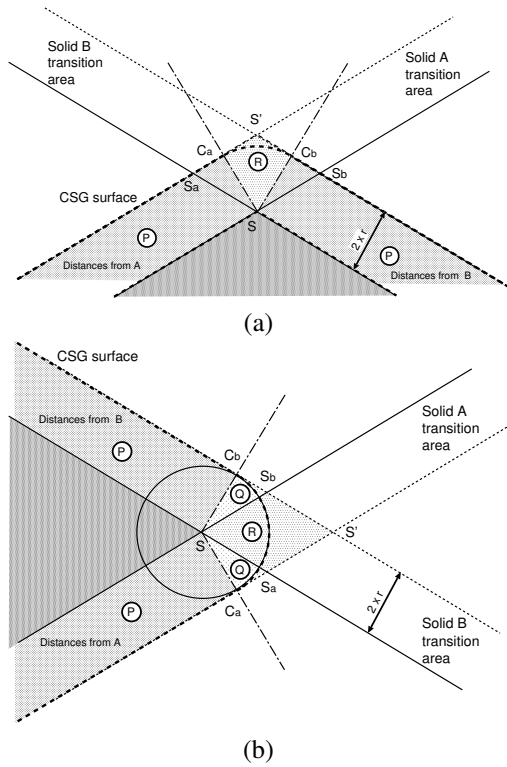


Figure 1. Intersection of two objects A, B . (a) edge with an obtuse angle, (b) edge with an acute angle

$$\begin{aligned} A \cup B &: d_{A \cup B} = \max(d_A, d_B) \\ A - B &: d_{A - B} = \max(0, d_A - d_B) \end{aligned} \quad (1)$$

gained a lot of interest [3, 1, 6], since it keeps the DF correct almost everywhere except in the edge areas which are influenced by both original solids. This insufficiency however is the source of severe problems in reconstruction sometimes, since it results in an unbounded DF error (Figure 5, upper row).

2.2 Advanced CSG Operation

Here we describe a new technique which works in a single run through all voxels and it identifies voxels where the simple minmax criterion (1) yields correct values. For the remaining voxels the resulting densities are computed by checking a local voxel neighborhood.

When object A is represented by a TDF, each voxel of its grid belongs either to the inside $\mathcal{I}_A$, outside $\mathcal{O}_A$ or transient area $\mathcal{T}_A$. We denote the boundaries of the transition area as *inner surface* I_A and *outer surface* O_A —the real object surface passes between them in the center of the transition area. Since the thickness of the transient area is $2r$, the distance from a surface point to I_A is r and therefore the inside area

is identical to the S_r -erosion of the object. In order to compute the intersection of objects A and B , it is necessary to reconstruct the inner surface $I_{A \cap B}$, find points in the transient area of $A \cap B$ and correctly compute the distances of these points to $I_{A \cap B}$. This approach is illustrated in Figure 1, which shows orthogonal cuts through edges with obtuse and acute angles. The minmax criterion (1) is applied in such areas, where the corresponding voxels of both solids are non-transitional. The remaining space can be divided in areas P, Q and R according to the geometry of the newly created edge. We apply the minmax criterion also in the P area, while in the voxels of Q and R areas, we ideally have to compute the distance to line S instead. Line S is defined as the intersection of the inner surfaces of A and B .

The core of the *advanced* CSG operation resides in a local approximation of the inner surface of both objects by a plane. Here, we have to identify regions P, Q and R . We cannot distinguish between regions P and Q solely by point operations. Furthermore, in the region Q we need information about both surfaces, but we have it just about one surface. How to overcome this problem, we are going to discuss later.

In a voxel V of the region R or Q we estimate density and gradient so that it corresponds to the direction and distance to the nearest point of line S , which is an estimate of the intersection of the inner surfaces (Figure 1). The outer surface of the result then forms an arc with center S , radius $2r$ and endpoints C_a, C_b , where r is the radius of the transitional area. Likewise, the real surface is formed by an arc with the same center, but with radius r . Thus we exactly fulfill the S_r -openness and S_r -closeness criterion for object representability.

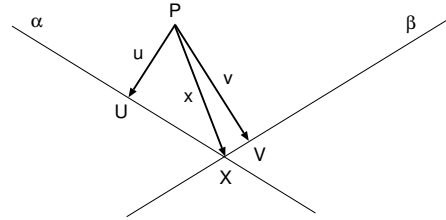


Figure 2. Intersection of two planes. Planes α, β are given by point P and vectors $\vec{u}, \vec{v}$. X is to P the nearest point of the intersection of these planes. Our goal is to determine vector $\vec{x}$.

First, we solve the problem depicted in Figure 2. Given is point P and two linearly independent vectors $\vec{u}, \vec{v}$ which determine planes α, β — $\vec{u}, \vec{v}$ are respectively normal vectors of planes α, β . The distances between the planes and P are $\|\vec{u}\|, \|\vec{v}\|$. Let X be the nearest point of intersection of planes α, β to P . Our goal is to find vector $\vec{x} = \overrightarrow{PX}$. We

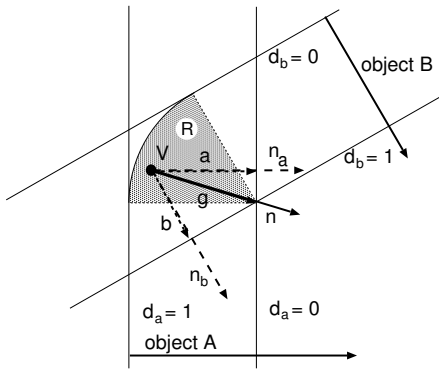


Figure 3. Intersection of two objects—obtuse angle. Density and gradient of voxel $V \in \mathcal{T}_{A \cap B}$ are estimated through vector $\vec{g}$.

use three facts:

1. $(\vec{x} - \vec{u}) \perp \vec{u}$
2. $(\vec{x} - \vec{v}) \perp \vec{v}$
3. $\vec{x}$ is a linear combination of vectors $\vec{u}, \vec{v}$.

On the basis of this observation we arrange a system of equations, from which we get the following relation:

$$\begin{aligned} \vec{x} &= \frac{(\|\vec{u}\|^2 - \vec{u}\vec{v}) \cdot \|\vec{v}\|^2}{\|\vec{u}\|^2\|\vec{v}\|^2 - (\vec{u}\vec{v})^2} \cdot \vec{u} + \\ &+ \frac{(\|\vec{v}\|^2 - \vec{u}\vec{v}) \cdot \|\vec{u}\|^2}{\|\vec{u}\|^2\|\vec{v}\|^2 - (\vec{u}\vec{v})^2} \cdot \vec{v} \end{aligned} \quad (2)$$

Figure 3 presents a scheme that we use to evaluate density and gradient of voxels, which lie around an edge of intersection of two objects. In voxel V we know the densities d_a, d_b and the normalized gradients $\vec{n}_a, \vec{n}_b$ of the input volumes A, B and we want to obtain the values d and $\vec{n}$ of the resultant intersection. We use the equations $\vec{a} = d_a \vec{n}_a$, $\vec{b} = d_b \vec{n}_b$ and exploiting (2) we get:

$$\begin{aligned} \vec{g} &= K \vec{n}_a + L \vec{n}_b \\ K &= \frac{d_a - d_b \vec{n}_a \vec{n}_b}{1 - (\vec{n}_a \vec{n}_b)^2} \\ L &= \frac{d_b - d_a \vec{n}_a \vec{n}_b}{1 - (\vec{n}_a \vec{n}_b)^2} \end{aligned} \quad (3)$$

For $K > 0, L > 0$ voxel V lies in region **R** or **Q**, and we use vector $\vec{g}$ for the evaluation of its density and gradient:

$$\begin{aligned} d &= \|\vec{g}\| \\ \vec{n} &= \frac{\vec{g}}{\|\vec{g}\|} \end{aligned} \quad (4)$$

Otherwise, we use the minmax criterion (1).

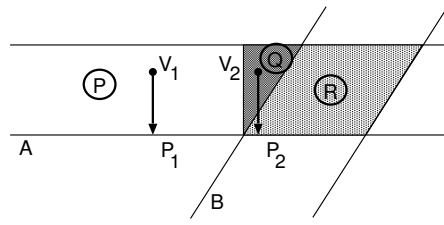


Figure 4. Classification of the region $\mathcal{T}_A$ in subregions **P**, **Q** and **R**.

Now we explain, how to solve the above mentioned problem of region **Q** in Figure 1b, where we have information just about one surface, but where we need information about both surfaces in order to smooth the edge correctly. For this purpose we have to add to our algorithm the following steps:

1. For voxels lying in the vicinity of just one surface we need to distinguish if they belong to region **P** or **Q**.
2. For voxels in the region **Q** we have to estimate the missing information about the second surface before performing the CSG operation.

A test for the classification of voxels in regions **P** and **Q** is illustrated in Figure 4. Using the density (distance) and gradient of voxel $V : V \in \mathcal{T}_A \wedge V \notin \mathcal{T}_B$ we construct point P as the foot of a perpendicular from V to I_A . If $P \in \mathcal{T}_B$, then voxel $V \in \mathbf{Q}$, otherwise $V \in \mathbf{P}$.

It is not quite trivial to test if point $P \in \mathcal{T}_B$. P as a point of the continuous space is located in one cell of the grid with 8 voxels in its corners—some of them can be inside and some of them outside the transition area of **B**. We use the following rules to decide about P :

- If all voxels of the cell lie outside (inside) $\mathcal{T}_B$, then P is outside (inside), too.
- Otherwise we check all 26 neighboring cells and from them we choose those, which have all vertices in $\mathcal{T}_B$. Using trilinear interpolation for voxels of every chosen cell we estimate the density at point P . Each selected cell is further assigned a weight on the basis of its distance from the point P . Finally, we compute a weighted average from the estimated values of the density at P , which determines if the voxel V lies in the region **Q**.

In the second step we add the missing information about the surface **B** to voxel V . We distinguish two cases:

- If there are at least in one direction two nearest voxels to V in $\mathcal{T}_B$, we use information from these voxels. For

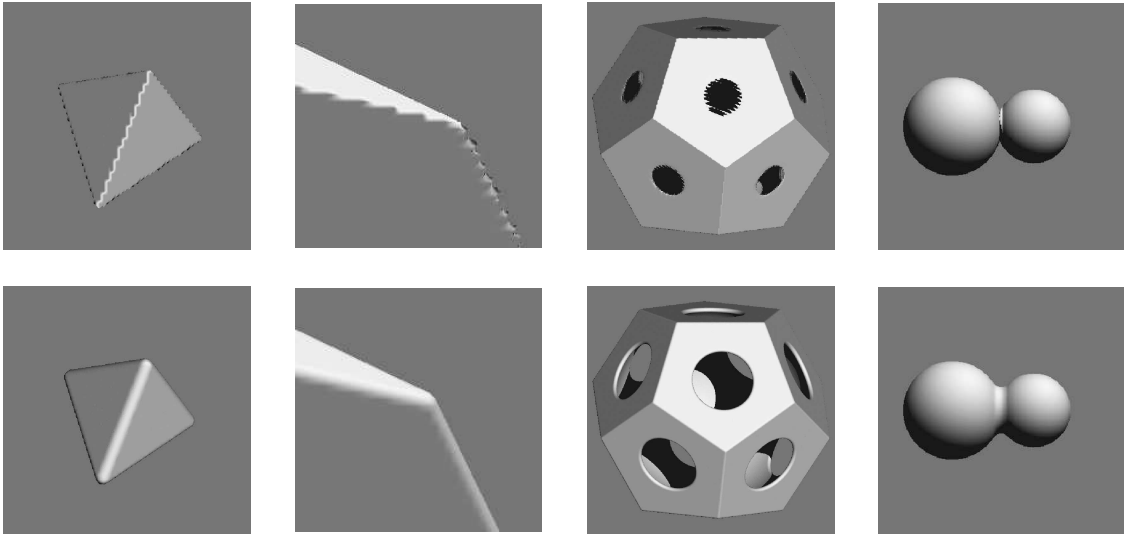


Figure 5. Edge artifacts for different CSG implementations of voxelized models. Operations: top row - just minmax criterion, bottom row - the new method. Object from left: tetrahedron, wedge, dodecahedron minus sphere, union of two touching spheres.

instance, if we want to evaluate density and gradient in voxel $V_{i,j,k}$ and we know density and gradient in voxels $V_{i+1,j,k}$, $V_{i+2,j,k}$, we set:

$$\begin{aligned} d_{i,j,k} &= 2d_{i+1,j,k} - d_{i+2,j,k} \\ \vec{n}_{i,j,k} &= 2\vec{n}_{i+1,j,k} - \vec{n}_{i+2,j,k} \end{aligned} \quad (5)$$

If we can do this estimation for pairs of voxels in more than one direction, the final values will be calculated as an average of these estimations.

- If there is no such pair of voxels in the neighborhood of V , we use the information determined in the first step for point P . The gradient of the point P is calculated in the same way as its density. We evaluate the difference x of the densities in points V , P from the known vectors $\vec{n}$, $\vec{u} = \overrightarrow{PV}$. Then voxel V has values:

$$\begin{aligned} d_V &= d_P + x \\ \vec{n}_V &= \vec{n}_P. \end{aligned} \quad (6)$$

3 Conclusion

The advanced CSG operation successfully solves voxelization of representable solids. As we can see in Figure 5, second row, sharp edges are now without artifacts. They are visibly smoothed, but it is the only way to represent them in a grid with the given resolution if the representability criterion is to be fulfilled. Similarly, in the union of two almost touching spheres, the thin gap between them is not representable, and therefore is filled.

In our future work we want to focus on other classes of solids, which are similarly not representable without additional modifications (e.g., superquadrics).

4 Acknowledgments

This work has been partly supported by a VEGA grant Virtual Environments for WWW No. 1/0174/03.

References

- [1] A. Bærentzen. Octree based volume sculpting. In C. Wittenbrink and A. Varshney, editors, *LBHT Proceedings of IEEE Visualization 98*, Oct. 1998.
- [2] J. A. Bærentzen, M. Šrámek, and N. J. Christensen. A morphological approach to voxelization of solids. In *The 8-th International Conference in Central Europe on Computer Graphics, Visualization and Digital Interactive Media 2000*, pages 44–51, Pilsen, Czech republic, 2000.
- [3] D. Breen, S. Mauch, and R. Whitaker. 3D scan conversion of CSG models into distance volume. In *IEEE Symposium on Volume Visualization*, pages 7–14, 1998.
- [4] A. Kaufman, D. Cohen, and R. Yagel. Volume graphics. *IEEE Computer*, 26(7):51–64, July 1993.
- [5] M. Šrámek and A. Kaufman. Alias-free voxelization of geometric objects. *IEEE Transactions on Visualization and Computer Graphics*, 5(3):251–266, 1999.
- [6] M. Šrámek and A. Kaufman. `vxt`: a c++ class library for object voxelization. In M. Chen, A. E. Kaufman, and R. Yagel, editors, *Volume Graphics*, pages 119–134. Springer Verlag, London, 2000.