Correction of Voxelization Artifacts by Revoxelization

Miloš Šrámek[1], Leonid I. Dimitrov[1], and J. Andreas Bærentzen[2]

[1] Austrian Academy of Sciences, VISKOM, Sonnenfelsgasse 19/2
A-1010 Vienna, Austria
{milos.sramek|leonid.dimitrov}@oeaw.ac.at

[2] Graphical Communication, Technical University of Denmark
DK2800 Lyngby, Denmark
jab@gk.dtu.dk

Abstract. Earlier proposed antialiasing techniques for voxelization of geometric objects in some cases do not result in completely alias-free data and image renditions. This is often the case for some implicit solids and CSG trees. In this paper we propose a set of operations, which can correct such corrupted data sets and subsequently lead to alias-free image renditions.

1 Introduction

The notion *voxelization* represents a set of techniques aiming to represent continuous geometric objects by means of 3-dimensional (3D) discrete rasters of samples with certain values (densities) assigned. Such 3D grids are the basic building component of techniques, commonly known as *Volume Graphics* [1]. Albeit still waiting for their widespread use, volume graphics techniques have already proven their superiority in numerous areas, where traditional surface-based graphics runs into troubles: simulation of fuzzy phenomena, volume deformation, weathering and subsurface scattering [2, 3].

The topic of this paper is antialiasing voxelization techniques, which in contrast to binary ones represent objects by means of samples with densities taken from an interval of real values in general. The great potential of antialiasing techniques is their ability to reconstruct original object surfaces and surface normals with high fidelity [4–7]. Thus, volume graphics not only extends the capabilities of surface graphics, but enables its replacement in its primary domain of application (surface rendering) due to advantages e.g. caused by improved rendering speeds due to the uniformity and simplicity of object representations [7].

Object voxelization and the subsequent rendering are, from the point of view of signal processing theory, a pair of sampling/resampling operations, and as such they must obey the rules specified by Shannon's sampling theorem. In other words, the input signal must be band-limited before sampling, in order to obtain alias free-data and/or renditions. However, a naive implementation of these rules [8, 9] is not sufficient for obtaining images with good visual quality (i.e., reconstructed surfaces with correct positions and surface normals).

As we have shown in our previous papers [6, 7, 10], a distance based voxelization should be preferred over plain filtering, since it enables the precise reconstruction the

surface position and normal for a wide range of surface curvatures. Two versions of the distance approach were proposed:

- distance field representation, and
- voxelization using V-model (voxelization model) representation of objects.

Objects in the former case are represented by an unbounded distance field [4, 5, 11–13], which means that a single object fills the whole volume grid. Positive (negative) distance values are assigned to voxels outside and negative (positive) ones to those inside, and the object surface is defined as the 0-level isosurface of the reconstructed volume. The distance field is linear and, therefore, trilinear interpolation and central differences are used in the reconstruction phase. The latter version stems from the concept of V-models, which we introduced earlier [7]. A V-model of an object is such a (continuous) volumetric representation (function) of the object, which can be reconstructed after sampling at least in the surface area with a minimal error (i.e. it is sufficiently smooth). A point of a V-model, deep enough in the object, is assigned some *inside* value (typically 255 in the case of an 8 bit representation) and, similarly, a point far outside has some *outside* value (typically 0). The in-between values follow some transition profile, which has to be carefully selected in accordance with the density and normal reconstruction filters, to be applied later in order to ensure good visual quality of the renditions. The concept of V-models was introduced in order to free the theoretical analysis and design of profile/filter pairs from object specific voxelization techniques, which can be spoiled by the presence of sharp edges. Two V-model profile/filter pairs were found to be suitable [7]:

1. Piecewise linear density profiles, with trilinear density reconstruction and central differences for normal estimation. The resulting representation is, in fact, the same as the distance field representation, however, objects are now kept localized by the finite thickness of the transition layer (Fig.1).
2. Error function (erfc) profiles with tricubic density reconstruction and Gabor filter for normal estimation.

The optimal halfwidth w of the transition region should bring together two contradictory demands: object localization (narrow transition region) and precise reconstruction and surface normal estimation. Experimentally, the values $w_{lin} = 1.8$ and $w_{erfc} = 1.0$ were found, together with the standard deviation of the Gabor filter in the second case being also around 1.0. These values are not hard thresholds and, especially in the second case, the technique is quite robust and the resulting image quality is insensitive to their change within some range. Since the transition area halfwidth influences the curvature of the smallest representable details, we get two slightly different techniques, representing a trade-off between precision and cost:

1. a less precise, but computationally cheaper one (linear profile/linear reconstruction filters), and
2. a more precise, but also more computationally expensive one (erfc/tricubic and Gabor filters).

As we have already stated, the primary purpose of the V-model introduction was to provide a theoretical environment for the search of suitable profiles and filters. As

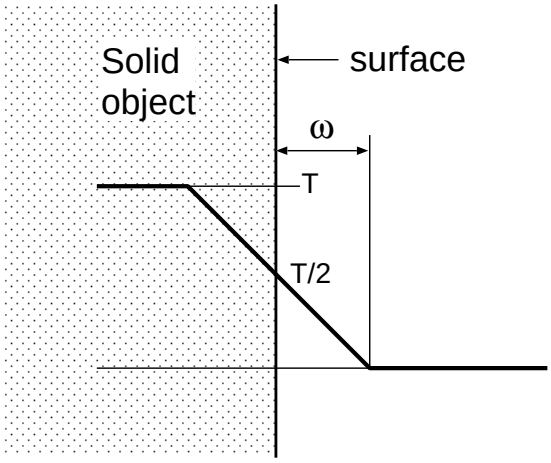


Fig. 1. *Surface density profile of a piecewise linear V-model of a solid object.*

such, it is an abstraction, which needs to be implemented differently for specific geometric models. The earlier proposed object voxelization techniques "quietly" assumed suitability of an object for voxelization (no sharp edges and small details) [14]. Since this is not always assured, artifacts may appear, especially at object edges and in areas, where the object thickness is comparable to the grid resolution.

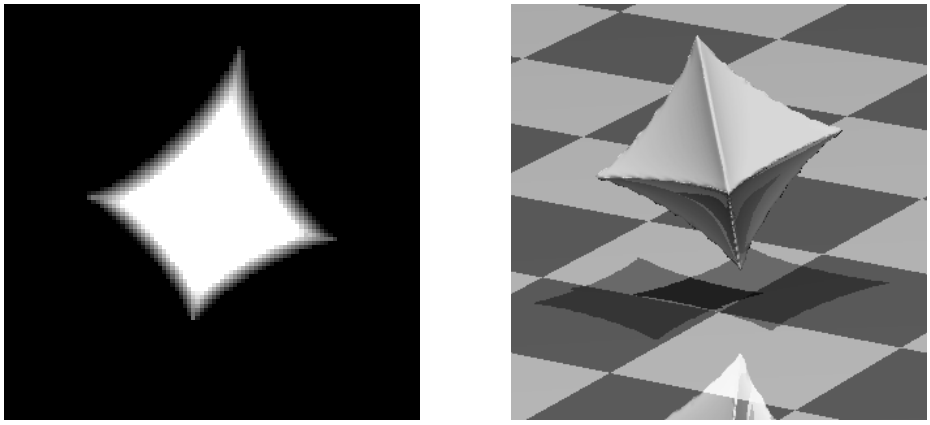
In this paper we propose a remedy for this problem by means of postprocessing the voxelized data. After an introductory summary of possible artifacts caused by small and sharp details (Section 2), we formulate a morphological criterion for voxelization suitability (Section 3). and propose a technique for building V-models for arbitrary solids. Further, we design new *extended binary opening* and *closing* operations, working on non-binary grid data and ensuring validity of the aforementioned suitability criterion by means of a postprocessing step—revoxelization. Finally in Section 4, we present experimental results, remaining problems and ideas for further work.

2 Voxelization artifacts

As already mentioned in the Introduction, the existing anti-aliasing voxelization techniques follow the assumption of sufficient smoothness of the surfaces. In this section we critically review these voxelization techniques putting an emphasis on their artifact susceptibility when the smoothness condition is violated.

2.1 Implicit solids

All reviewed voxelization techniques first estimate the distance to a primitive for a voxel and then compute a voxel density, depending on the selected surface profile. In the case



(a) (b)

Fig. 2. *(a) One slice of a data set representing a voxelized supersphere. The sharp edges cause reconstruction artifacts visible in renditions (b).*

of an implicit solid $f(\mathbf{p}) \leq 0$, the distance is estimated by a linear approximation:

$$d(\mathbf{p}) = \frac{f(\mathbf{p})}{||\nabla f(\mathbf{p})||}, \tag{1}$$

where $\mathbf{p}$ is the point of interest in $\mathcal{R}^3$ and ∇ is the gradient operator. It may easily happen that volumetric objects thus defined are too thin or have sharp edges. Superquadrics present a typical example of such a misbehavior. A superellipsoid:

$$\left(\left(\frac{x}{a}\right)^{\frac{2}{e}} + \left(\frac{y}{b}\right)^{\frac{2}{e}}\right)^{\frac{e}{n}} + \left(\frac{z}{c}\right)^{\frac{2}{n}} - 1 \leq 0, \tag{2}$$

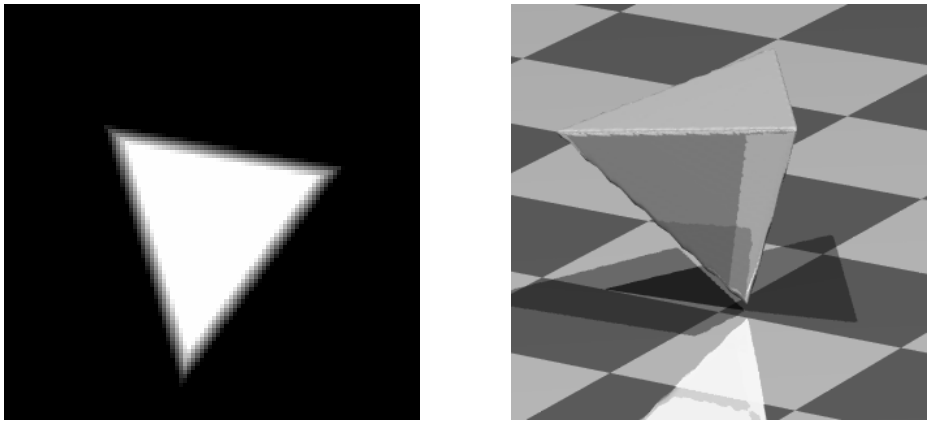
where e and n are east-west and north-south shape parameters respectively, has both thin areas and sharp edges for $e, n > 2$ (Fig. 2), when voxelized in the above way.

2.2 CSG trees

There are several possibilities for implementing voxel-by-voxel CSG operations (union, intersection and difference). Since it is necessary that the surface profile remains unchanged by the operation in areas influenced by one object only, the min/max version should be preferred (Tab.1). Its straightforward application though leads to sharp edges (intersection and difference) and sharp corners (union) (Fig. 3).

3 The morphological criterion

We have experimentally shown before [7] that both surface and normal reconstruction errors grow with the increasing surface curvature of the voxelized object. Later we



(a) (b)

Fig. 3. *(a) One slice of a data set representing a voxelized tetrahedron, defined as the intersection of 4 halfspaces. The sharp edges cause reconstruction artifacts visible in renditions (b).*

Table 1. *Voxel based CSG operations between objects. (d – voxel density)*

Operation	Symbol	Density
Intersection	$A \cap B$	$d_{A \cap B} = min(d_A, d_B)$
Union	$A \cup B$	$d_{A \cup B} = max(d_A, d_B)$
Difference	$A - B$	$d_{A-B} = max(0, d_A - d_B)$

analyzed this dependency theoretically [10] and concluded that a solid is suitable for voxelization at a given resolution if it is simultaneously open (S_r-open) and closed (S_r-closed) with respect to a spherical structuring element S_r with radius r. This radius r can be derived from the shape of the reconstruction filter: for the trilinear interpolation of densities the filter needs 8 samples in voxel vertices, Therefore, the radius r is equals $\sqrt{3}$. However, for the reconstruction of the surface gradients we need 32 samples, which results in $r = 2\sqrt{3}$. These values were verified experimentally [7]. According to [10], the S_r-open/S_r-close criterion is equivalent to the condition that the medial surface of the solid is nowhere closer to the outer surface of the solid than the chosen r. In this case and only then all samples of the reconstruction filter kernel lie on one side of this medial surface.

We now propose a way to construct a V-model of a solid following from the S_r-open/S_r-closed condition, and finally, a technique for correction of the voxelization artifacts mentioned in the Introduction.

3.1 S_r-opening and S_r-closing

Formally, the S_r-open and S_r-closed solid S^{OC} for an arbitrary solid S with respect to a structuring element (sphere) S_r can be defined as

$$S^{OC}(S_r) \equiv \quad S^{OC}(S_r)(\mathbf{x}) = C(O(S, S_r), S_r)(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{R}^3. \tag{3}$$

The first operation, $O(S, S_r)$ is illustrated in Fig. 4(a). Since closing is dual to opening, Eq. 3 can be rewritten as

$$S^{OC}(S_r) \equiv \quad S^{OC}(S_r)(\mathbf{x}) = \neg O(\neg O(S, S_r), S_r)(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{R}^3, \tag{4}$$

where $\neg$ stands for object complement. Further, S_r-opening can be defined by means of the union operator as

$$O(S, S_r) \equiv \quad O(S, S_r)(\mathbf{x}) = \bigcup_{\forall \mathbf{p}: S_r \subset S} S_r(\mathbf{p})(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{R}^3, \tag{5}$$

where $S_r(\mathbf{p})$ is an instance of the S_r kernel centered in $\mathbf{p}$. We call the $S_r \subset S$ operation *fitting* and the S_r-kernel a *fit kernel*. For each $\mathbf{p}$, where $S_r(\mathbf{p})$ fits we add the $S_r(\mathbf{p})$ kernel by means of the union operation. Since this reminds us of the popular splatting technique, in this context we call the $S_r(\mathbf{p})$ kernel *splat or splatting kernel*. Note that we use the same sphere S_r in the role of a fit and a splat kernel for both S_r-opening and S_r-closing.

3.2 V-opening and V-closing

A V-model of a solid S can be formally defined in a similar way as S^{OC}, but with different fit and splat kernels.

Let the new kernel $V_r(p)$ be a V-model of a sphere with radius r and center $\mathbf{p}$, i.e. a trivariate function

$$V_r(\mathbf{p}) \equiv \quad V_R(\mathbf{p})(\mathbf{x}) = g(|\mathbf{x} - \mathbf{p}|), \tag{6}$$

where g is a function defining the selected surface profile of the V-model. For example, for linear V-models the function g is defined as

$$g(e) = \begin{cases} 1 - \frac{e}{2r} \text{ if } \frac{e}{2r} \leq 1 \\ 0 \qquad \text{otherwise} \end{cases} \tag{7}$$

Then a V-model V of a solid object S is defined by means of the V-open and V-close functions O^V and C^V respectively:

$$\begin{aligned} V(S, F_O, S_O, F_C, S_C) \equiv V(S, F_O, S_O, F_C, S_C)(\mathbf{x}) = \\ = C^V\left(O^V(S, F_O, S_O), F_C, S_C\right)(\mathbf{x}), \end{aligned} \tag{8}$$

where F_O, F_C are fit kernels and S_O, S_C are splat kernels for the operations respectively. In the case of the V-open function O^V, $F_O = S_r$ and $S_O = V_r$ (Fig.4b):

$$O^V(S, F_O, S_O) \equiv O^V(S, S_r, V_r)(\mathbf{x}) = \max_{\forall \mathbf{p}: S_r(\mathbf{p}) \subset S} V_r(\mathbf{p})(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{R}^3. \tag{9}$$

V-open is a trivariate function, which already has some of the desired properties of V-models: it has the suitable profile in all planar and convex surface areas. However, the profile is not the correct one in the vicinity of sharp concavities. V-open takes as an argument binary objects and therefore its fit kernel (F_O) has to be also binary (S_r). However, the input argument of the V-close function is a non-binary function O^V and therefore the fit kernel too should be a non-binary one, namely the splatting kernel. Therefore we define

$$C^V(O^V(S), F_C, O_C) \equiv C^V(O^V(S), V_r, V_r)(\mathbf{x}) = \\ = \neg \max_{\forall \mathbf{p}: V_r(\mathbf{p}) \sqsubset \neg O(S)} V_r(\mathbf{p})(\mathbf{x}), \forall \mathbf{x} \in \mathcal{R}^3, \qquad (10)$$

where $\neg$ stands for object complement and the modified inclusion operation is defined by

$$a \sqsubset b = \begin{cases} true & \text{if } \forall x : a(x) \leq b(x) \\ false & \text{otherwise} \end{cases} . \qquad (11)$$

Both the V-open and V-close functions are defined over a continuous domain (i.e., the fit and splat operations need to be performed at infinite number of points). Therefore they are more suitable for the explanation of our approach than for an implementation. Fortunately, we do not need to know the V-model function value for each spatial point; it is sufficient to evaluate it only at grid point locations, since we work in the discrete domain. Therefore, in the practical implementation of the technique, we replace the continuous object model by its (partially incorrect) preliminary voxelization G, described in the Introduction and proposed earlier [6]. The technique produces correct surface profiles nearly everywhere, except in the vicinity of sharp edges, corners and small details. The erroneous regions are then corrected in a post-processing step (which we call revoxelization), comprising a subsequent application of the discrete versions of the V-open and V-close operators. As opposed to its continuous version, the V-open operator (Eq. 9) is applied, similarly as the V-close one (Eq. 10), to multi-valued grid data. Therefore we use V_r both as the fit and splat kernel and Eq. 9 changes to:

$$O^E(G, V_r) \equiv \quad O^E(G, V_r)(\mathbf{x}) = \max_{\forall \mathbf{p}: V_r(\mathbf{p}) \sqsubset G} V_r(\mathbf{p})(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{Z}^3 \qquad (12)$$

Note that $\mathbf{x} \in \mathcal{Z}^3$ now. However, in order to achieve subvoxel precision of the technique, which is necessary for a high quality and artifact-free voxelization, we let $\mathbf{p} \in \mathcal{R}^3$. We show in the following section, how to cope with this obstacle. Since both kernels are the same, the V-close operator (Eq. 10) simplifies to

$$C^E(G, V_r) = \neg O^E(\neg G, V_r) \qquad (13)$$

The discrete versions of V-open and V-close, applied to multi-valued grids with special features (profile), are counterparts to the traditional opening and closing operations for binary grids. Since they implement the same operation, but with subvoxel precision, we call them *extended opening* and *closing* (therefore the superscript E in the last formulas).

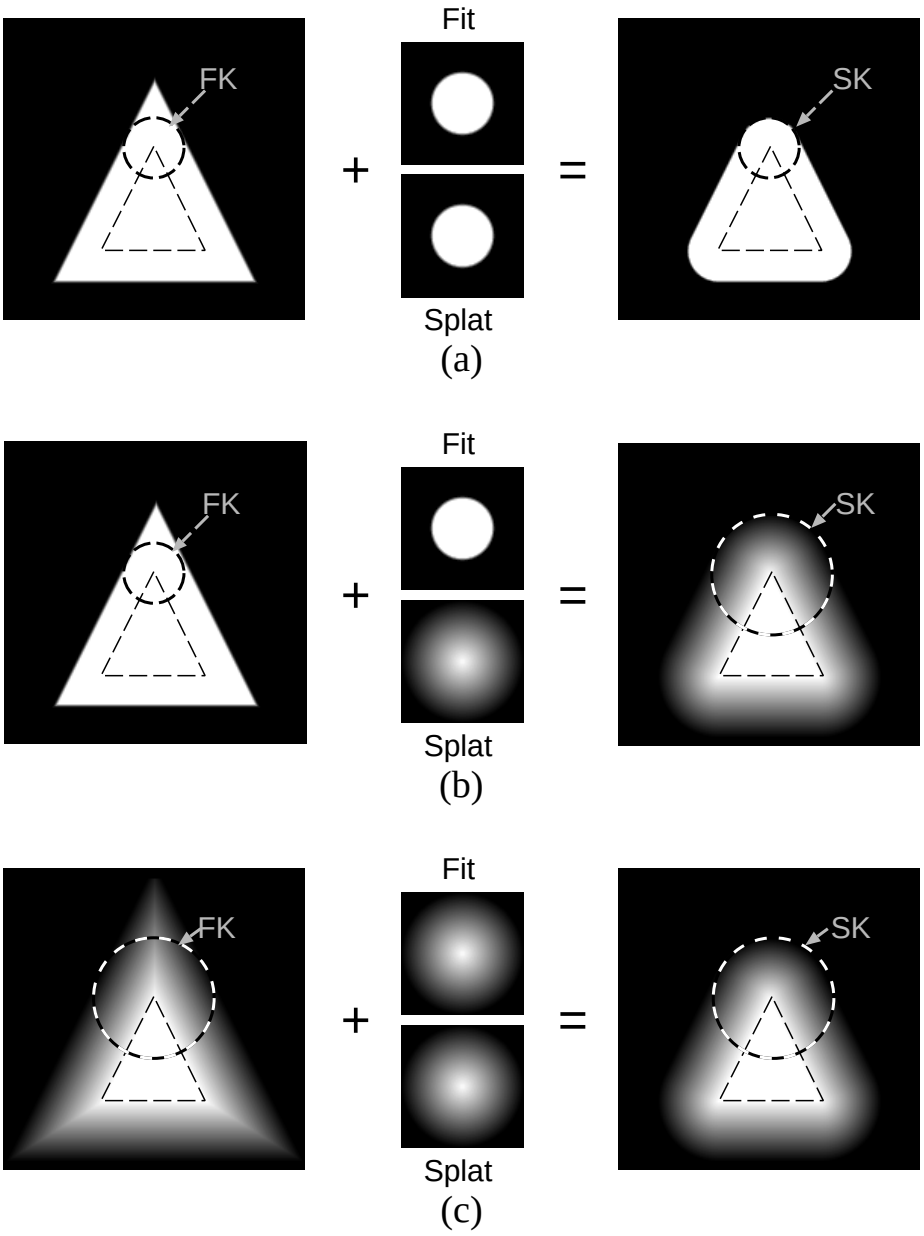


Fig. 4. *(a) S_r-opening of a geometrically defined solid, (b) construction of the V-open function O^V for a geometrically defined solid, (c) revoxelization of a discrete data set with sharp edges. In each pair, one instance of the fit kernel (FK) is depicted (right), together with a corresponding splat kernel (SK, left).*

4 Implementation

The V-model based voxelized objects lie halfway between continuous and discrete binary representations: being discrete, they maintain the precision of the continuous model (of course, within the limits defined by the sampling theorem). Therefore the extended opening must also satisfy the same demands: discreteness and the possibility for arbitrary precision. The algorithm implementing this operation has several steps:

Input: *A partially incorrect voxelization of a solid.*
For the sake of simplicity we assume that the densities $d \in \langle 0, 1\rangle$. *The revoxelization operation should be applied only to transition area voxels, i.e. those with density* $d \in (0, 1)$.
Step 1: *Apply the V-open operation (Eq. 12).*
Step 2: *Compute a complementary volume with densities* $d' = 1 - d$
Step 3: *Apply the V-open operation (Eq. 12).*
Step 4: *Compute a complementary volume with densities* $d' = 1 - d$
Output: *Voxelized V-model of the solid object.*

Implementation of the O^E operation:

For each transition voxel $\mathbf{x}$*:*

Step 1: *Find the nearest* $\mathbf{p}$ *such that* $V_r(\mathbf{p}) \sqsubset G$
Step 2: *Assign to* $\mathbf{x}$ *density according to* $|\mathbf{x}-\mathbf{p}|$ *and the selected V-model type (linear or* erfc*).*

We used a steepest descend technique for finding the nearest fitting instance of V_r, which minimizes a cost function reflecting both the distance $e = |\mathbf{x}-\mathbf{p}|$ and the distance D of $V_r(\mathbf{p})$ from the V-model surface. D was defined as

1. 0 if the $V_r(\mathbf{p})$ (Eq. 11), and
2. the sum of differences between V_r and G for those samples, where the kernel value V_r exceeded the density of G for the non-fitting instances of $V_r(\mathbf{p})$.

Figure 5 depicts both the dependency of D and the minimized $D + e$ on the distance from the solid's surface.

Minimization of the cost function is costly, since it has to be evaluated many times during the minimization in the n^3 neighborhood of each point $\mathbf{p}$ (n=9 for the linear V-model profile). This procedure can be accelerated by precomputing look-up tables of the fit/splat kernels. Actually, we used the same look-up tables as those used for voxelization of parametric surfaces by splatting [6].

The other speed-up technique stems from the observation that not all transition voxels but only those in edge vicinity have to be revoxelized. Such voxels were detected by means of the Laplacian operator. Voxels above certain threshold were identified and the resulting areas were dilated by a 5^3 structuring element. This operation enabled us to speed-up the revoxelization operation by nearly two orders of magnitude, which resulted in processing times comparable to the (first) voxelization step.

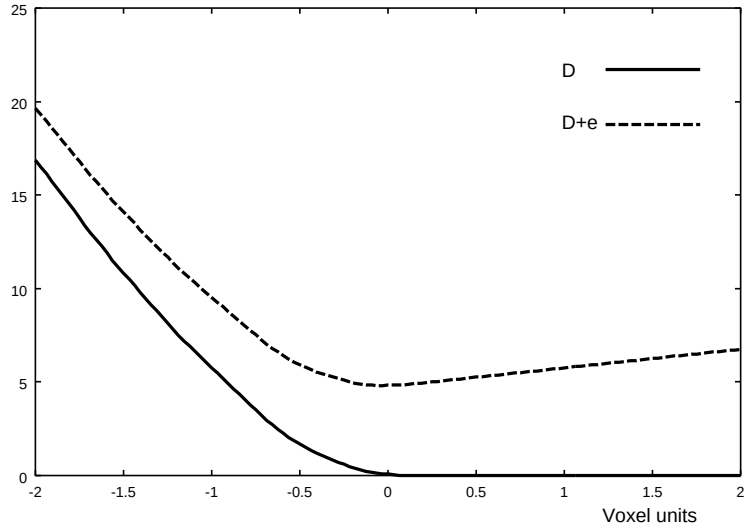


Fig. 5. *Dependency of D and minimized cost function $D + e$ on distance from a surface.*

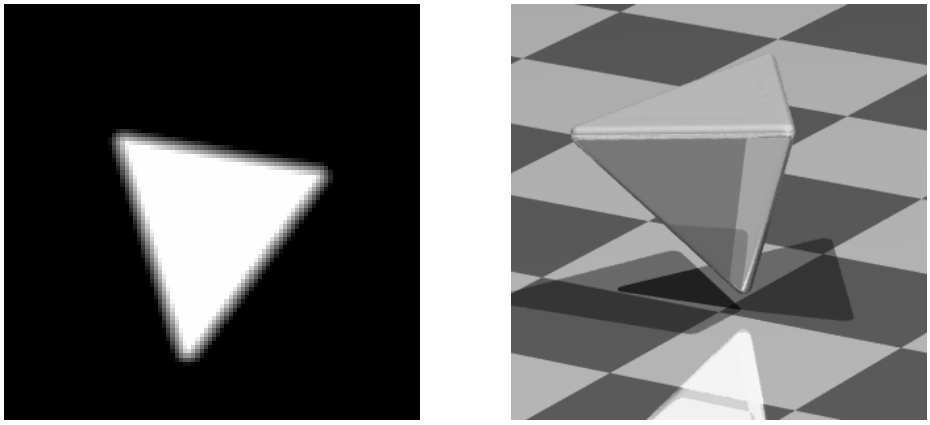
Figure 6a shows a slice of a revoxelized data set. The corners are not sharp anymore, which results in an image with correctly rendered object edges (Figure 6b).

The work is not finished yet. Unfortunately, the above mentioned steepest descend minimization procedure sometimes fails by getting trapped in local minima. This results in data with a certain level of noise which, albeit not visible directly in the volume data, is accentuated by the gradient operation used for shading. The trapping is probably caused by the discrete character of the search space. In future, we want to concentrate on alternative cost function designs and/or optimization methods not sensitive to the presence of local minima.

Acknowledgments This work has been partially supported by Slovak Grant Agency grants VEGA 2/6017/99 and VEGA 1/7666/20.

References

1. A. Kaufman, D. Cohen, and R. Yagel. Volume graphics. *IEEE Computer*, 26(7):51–64, July 1993.
2. Julie Dorsey, Alan Edelman, Justin Legakis, Henrik Wann Jensen, and Hans Køhling Pedersen. Modeling and rendering of weathered stone. In Alyn Rockwood, editor, *Proceedings of the Conference on Computer Graphics (Siggraph99)*, pages 225–234, N.Y., August8–13 1999. ACM Press.
3. Nao Ozawa and Issei Fujishiro. A morphological approach to volume synthesis of weathered stones. In M. Chen, A.E. Kaufman, and R Yagel, editors, *Volume Graphics*, pages 367–378. Springer-Verlag, 2000.
4. S.F.F. Gibson. Using distance maps for accurate surface reconstruction in sampled volumes. In *IEEE Symposium on Volume Visualization*, pages 23–30, 1998.



(a) (b)

Fig. 6. *(a) One slice of a revoxelized data set with a tetrahedron and its rendition (b). Note the unsharp edges (vertices in 2D) in (a) and the artifact-free edges in (b).*

5. S. F. Frisken, Ronald N. Perry, Alyn P. Rockwood, and Thouis R. Jones. Adaptively sampled distance fields: A general representation of shape for computer graphics. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, Annual Conference Series, pages 249–254. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.
6. M. Šrámek and A. Kaufman. Object voxelization by filtering. In *IEEE Symposium on Volume Visualization*, pages 111–118, 1998.
7. M. Šrámek and A. Kaufman. Alias-free voxelization of geometric objects. *IEEE Transactions on Visualization and Computer Graphics*, 5(3):251–266, 1999.
8. S.W. Wang and A. Kaufman. Volume sampled voxelization of geometric primitives. In *Visualization '93*, pages 78–84, San Jose, CA, October 1993.
9. S.W. Wang and A. Kaufman. Volume-sampled 3D Modelling. *IEEE Computer Graphics and Applications*, 14(5):26–32, September 1994.
10. J. Andreas Bærentzen, M. Šrámek, and Niels Jorgen 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.
11. B.A. Payne and A.W. Toga. Distance field manipulation of surface models. *IEEE Computer Graphics and Applications*, 12(1):65–71, January 1992.
12. M.W. Jones. The production of volume data from triangular meshes using voxelisation. *Computer Graphics Forum*, 15(5):311–318, December 1996.
13. D.E. Breen, S. Mauch, and R.T. Whitaker. 3D scan conversion of CSG models into distance volume. In *IEEE Symposium on Volume Visualization*, pages 7–14, 1998.
14. M. Šrámek and A. Kaufman. `vxt`: a c++ class library for object voxelization. In Min Chen, Arie E. Kaufman, and Roni Yagel, editors, *Volume Graphics*, pages 119–134. Springer Verlag, London, 2000.