Alias-Free Voxelization of Geometric Objects

Milos Sramek and Arie E. Kaufman, Fellow, IEEE

Abstract—We introduce a new concept for alias-free voxelization of geometric objects based on a Voxelization model (V-model). The V-model of an object is its representation in three-dimensional continuous space by a trivariate density function. This function is sampled during the voxelization and the resulting values are stored in a volume buffer. This concept enables us to study general issues of sampling and rendering separately from object specific design issues. It provides us with a possibility to design such V-models, which are correct from the point of view of both the sampling and rendering, thus leading to both alias-free volumetric representation and alias-free rendered images. We performed numerous experiments with different combinations of V-models and reconstruction techniques. We have shown that the V-model with a Gaussian surface density profile combined with tricubic interpolation and Gabor derivative reconstruction outperforms the previously published technique with a linear density profile. This enables higher fidelity of images rendered from volume data due to increased sharpness of edges and thinner surface patches.

Index Terms—Volume graphics, volume rendering, filter-based voxelization, normal estimation, error estimation.

1 INTRODUCTION

Volume graphics delineates a set of techniques aimed at modeling, manipulation, and rendering of objects represented by means of a 3D raster of elementary volume primitives—voxels [1]. To be described by the voxel raster, an object has to be subjected to a process called voxelization or 3D scan conversion [2], [3]. The volume raster representation is similar to that of scanned real objects and therefore enables simultaneous handling and rendering of synthesized and real objects. Thus, representing a bridge between the areas of computer graphics and volume visualization, volume graphics has made possible the penetration of a wide plethora of volume visualization techniques [4], [5], [6], [7], [8] in the domain of computer graphics.

Volume graphics has several advantages in comparison with the traditional geometric approach. Voxelization decouples object specific issues from rendering and enables uniform representation of all objects by a single primitive. Thus, instead of dealing with a number of different objects with specific features, a renderer processes only one primitive object—the voxel—which simplifies its architecture and leads to a more effective implementation. Moreover, rendering is independent of the original number of objects and, hence, a complete independence on object and scene complexity is achieved. The voxelization itself is, for most object types, algorithmically easier than, for example, the ray-object intersection computation in ray tracing, and can be performed off-line. Another favorable characteristic of voxel representation is the simplicity, how Boolean, constructive solid geometry, and block operations are performed. Furthermore, additional view independent information as, for example, precomputed light source visibility or object color, can be stored as voxel attributes.

Two well-known drawbacks of volume graphics techniques are their high memory and processing time demands. However, due to progress in both memory technology, computers, and specialized volume rendering hardware [9], [10], these drawbacks are gradually losing their significance.

The goal of this paper is to introduce a new technique for alias-free voxelization of objects, based on its voxelization model. In Section 2, we review and classify known voxelization techniques and address important aspects of reconstruction from sampled data. In Section 3, we discuss the concept of V-models. Section 4 proposes several combinations of V-models and reconstruction techniques, which are then evaluated by means of numerous experiments in Section 5. Finally, in Section 6, we list specific voxelization techniques which implement the V-model concept for particular objects.

2 BACKGROUND

2.1 Voxelization Techniques

Voxelization is essentially a sampling process and, therefore, sampling theory rules should be taken into account. The first voxelization algorithms were binary, assigning, for example, 1 to occupied voxels and 0 to those unoccupied [11], [12]. This approach totally ignored sampling theory and, consequently, rendered pictures suffered from aliasing. This was predominantly related to the poor estimation of the surface normal vector. Techniques were proposed to improve the normal vector estimation by taking into account information from a larger neighborhood (e.g., contextual shading [13], context-sensitive normal estimation [14], center-of-gravity shading [15]). Although the improvement was significant, none of the techniques yielded a normal vector precise enough for the simulation of such effects as reflection and refraction of light on an object surface. Better results were obtained by discrete ray tracing [16]. In this technique, in addition to estimating the normal...
from the discrete data, the normal was confirmed from the analytic object description which is kept along with the voxel raster.

Höhne and Bernstein [17] pointed out that shading of scanned objects could be significantly improved if one takes advantage of “inaccuracy” of the 3D scanning device. Due to physical limitations, a point spread function of the scanner is not the ideal dimensionless pulse, but rather a Gaussian-like profile with a finite support. Therefore, it acts as a low-pass filter, suppressing high frequencies and blurring object edges. This is known as the partial volume effect (PVE). Using such data, realistic shading can be achieved when the normal is computed by means of a discrete gradient filter (e.g., by central differencing).

The first “smoothed” objects were synthesized for the sake of algorithm testing. A value, proportional to the distance from a center of the test sphere, was stored in voxels near its surface [18]. Other test objects were obtained by simulating the PVE by computation of relative occupancy of each voxel, shared both by the object and background [19]. Techniques for voxelization of smooth objects, primarily aimed at visualization, were proposed later. We classify them into two categories: filtration and distance field techniques.

Filtration techniques solve the problem of aliasing by low-pass filtering of the object. Different filters were used: coneshaped Bartlett filter [3], [20], Gaussian [21], and oriented box filter (a 1D box filter perpendicular to object surface) [22], [23]. The continuous filtered function is subsequently sampled at grid locations. Assuming that the filter support is smaller than the object, its interior is represented by some “inside” density and background is assigned some “outside” density. There is, of course, a transition area, which smoothly blends the inside and outside densities in a thin layer around the surface. Thus, this volumetric representation is similar to the data obtained by a 3D scanning device.

Distance field techniques assign to all voxels of a scene their distance to the nearest surface point of the object [24], [25], [26], [27]. Usually, a certain value, typically 0, is assigned to the point on the surface. The distances are, in general, unbounded and the distance field of an object embodies the whole scene, which means that the traditional notion of spatially localized objects is violated. Object interior and background are usually distinguished by a different sign of the distance.

We have shown in our introductory paper [23] that the design of the voxelization filter cannot be treated separately from the subsequent interpolation and gradient estimation technique. For example, if the aforementioned Bartlett filter is combined with linear reconstruction of density and gradient, an error is introduced which manifests itself as deviation of the estimated normal from the ideal normal by several degrees. Magnitude of the error depends solely on surface orientation and not on its curvature. This error was earlier misinterpreted as a quantization error [3] due to representation by 8-bit integers. However, the actual quantization error is more than an order of magnitude smaller [23]. We have further shown that, in spite of the 3D character of objects, the correct voxelization filter should not have a 3D support, but it rather should be a one-dimensional filter oriented perpendicular to surface normal. Using a filter with the 3D filter support, the reconstructed surface is shifted by an amount proportional to its curvature, which is due to averaging of the object within the extend of the support. We have also shown that a linear surface density profile, which was the result of the application of an oriented box filter, enabled alias-free rendering for a wide class of geometric objects, even with small surface details comparable in size with the width of the filter. We estimated the optimal width of the filter to be approximately 2*1.7 voxel units for surface reconstruction by trilinear interpolation and gradient estimation by central differences.

Filtering of an object by the oriented box filter leads to a similar representation as the distance field technique. The resulting voxel density either directly reflects the distance, or at least linearly depends on it. Limitations of this representation reside in that the profile, which is a spatially first order polynomial for planar surfaces or surfaces with low curvature, becomes a higher order function in the vicinity of small surface details and sharp edges. In this case, the linear interpolation and gradient estimation becomes inappropriate and introduces artifacts. In this paper, we extend the technique of oriented box filtering to the V-model concept, which provides us with a framework for representation of solids, surfaces, and curves by means of a density function, nonlinearly depending on distance from the object surface (nonlinear distance map). Such representation, if combined with suitable reconstruction filters, leads to more precise reconstruction of surface details for the same volume buffer resolution as the linear profile.

### 2.2 Reconstruction from Sampled Data

Reconstruction of a continuous function and its derivative from sampled data is a complex problem, yet important for rendering [28], [29], [30], [31], [32]. The filter evaluation and design technique proposed by Möller et al. [31] stems from the Taylor series expansion of the interpolation convolution sum about the point $t$:

$$f^\omega_r(t) = \sum_{k=-\infty}^{\infty} f[k] \cdot w\left(t - \frac{k}{T}\right), \quad (1)$$

where $f^\omega_r(\cdot)$ is the reconstructed profile, $w(\cdot)$ is a general filter for reconstruction of the $n$th derivative of $f$, $T$ is the distance between samples, and $f[k]$ is a sample taken at point $kT$. The authors have shown that this sum can be expressed as

$$f^\omega_r(t) = \sum_{n=0}^{N} a_n^\omega_r(\tau) f^{(n)}(t) + r_N^\omega(\tau), \quad (2)$$

where $f^{(n)}(\cdot)$ is the $n$th derivative of the function being reconstructed, $a_n^\omega_r(\tau)$ is the Taylor coefficient for derivative $n$, $r_N^\omega(\tau)$ represents the remainder term, and, finally, $\tau$ is the relative position of coordinate $t$ between two consecutive sample points. It is first assumed that $N + 1$ derivatives of $f$ exist. Equation (2) allows us to define conditions for an ideal reconstruction filter and also provides a tool for
assessment of filter properties. Ideally, the \(n\)th derivative is reconstructed if all \(a_i = 0, i \neq n\) and \(a_n = 1\). Filters with coefficients \(a_i = 0\) for \(i < k\) and \(i \neq n\) are called \(k\)th degree error filters (\(k\)-EF) and are capable of reconstructing a polynomial function of up to \(k - 1\)st order without any error.

In this paper, we are interested only in reconstruction of the original signal and its first derivative. Therefore, henceforth we will use the notions “interpolation” or “derivative” filters, respectively.

Typically, a \(p\)th order polynomial filter is defined as nonzero in some interval \((-m, m)\), where \(m\) is an integer. In this case, the convolution sum takes only \(2m\) terms

\[
f_x(t) = \sum_{k=-m}^{m} f(t + k) \cdot w(t - k).
\]

In the case of a one-dimensional interpolation, we approximate the value of the function or its derivative at an arbitrary point \(t\) from known samples, situated in equidistant points along the axis. In the case of an interpolation at arbitrary 3D point \((x, y, z)\), we generally do not have any samples along lines running through the point and being parallel to coordinate axes. Consequently, in order to interpolate, for example, in the direction \(x\), we first have to reconstruct “artificial” samples at points \([x + k, y, z]\), \(k = -(m + 1), \ldots, m\):

\[
f_{xy}(x, y, z) = \sum_{k=-m}^{m} f_{x}(x + k, y, z) \cdot w(x + k).
\]

where subscript \(xyz\) (\(yz\)) emphasizes that the value was interpolated along axes \(z, y, x\) (\(y, z\)) in the given order. Similarly, \(f_{yz}\) is given as

\[
f_{yz}(x, y, z) = \sum_{k=-m}^{m} f_{y}(x, y + k, z) \cdot w(y + k).
\]

where boldface \(x\) indicates that the \(x\) coordinate is an integer. Finally, for interpolation in \(z\) direction:

\[
f_{z}(x, y, z) = \sum_{k=-m}^{m} f_{z}(x, y, z + k) \cdot w(z + k).
\]

Thus, if we interpolate the current value by a filter with width \(2m\), in 3D it is necessary to evaluate the interpolation formula \(1 + 2m + 2m \times 2m = 4m^2 + 2m + 1\) times, with \(2m + 1\) additions and multiplications for one evaluation. Together this gives \(C(n) = 2(2m + 1)(4m^2 + 2m + 1)\) operations. Hence, in order to keep the computational demands low, we prefer the narrowest filters, which also implies low degree of the filter.

The same technique can be also used for estimation of density gradient \(G\). In this case, we have to evaluate (4) three times, each with a different interpolation filter replaced by its derivative counterpart:

\[
G(x, y, z) = [f_{x,y,z}, f_{y,z,x}, f_{z,x,y}],
\]
by a volume buffer, enabling us to study properties leading to artifact-free images apart from association with particular objects. Its aim is to knit contradictory demands of both sides and define a compromise which would represent the object as precisely as possible. Knowing the necessary properties, one can design a suitable voxelization procedure for each object class.

Our main concern is the proper design of the V-model from the point of view of the sampling theory. Objects defined in the continuous space have, in general, an unbounded frequency spectrum due to the surface discontinuity between inside and outside. This means that there is no optimal sampling density; all finite densities are simply too low. The actual sampling rate is therefore defined by the limits imposed on the volume grid resolution by the resources of the computer and it is considered to be fixed. Since we have full control over the V-model properties, we design it with this sampling rate in mind to keep its frequency spectrum within the necessary limits. Without loss of generality, the sampling step defining the selected resolution of the volume buffer is treated as a basic unit length (1 voxel unit—VU) and all object dimensions and filter and V-model parameters are expressed in this scale.

To visualize the contents of the buffer, it is necessary to reconstruct the original continuous V-model by an appropriate interpolation technique. The reconstructed continuous function is then subjected to rendering, which samples it again and projects the samples onto an image plane. In order to achieve the highest possible quality of the rendered image, it is impossible to separate the V-model design from rendering and, particularly, from the reconstruction technique. In this paper, we optimize the V-model for a surface-based visualization technique, which reconstructs a surface by means of interpolation and thresholding at 50 percent of the maximum density [33], [34], [35], [36]. A surface gradient vector, which is necessary for surface shading and secondary ray spawning, may be reconstructed in a similar way directly from the volume data by means of a suitable gradient filter.

The idea of object representation by means of the V-model evolves from the aforementioned techniques of oriented box filtering and representation by the distance field (Fig. 2, Section 2.1). In both cases, the density of the voxelized object linearly depends on the distance from the surface. In the first case, this density is bounded by some maximum and minimum values and, in the second case, it is unbounded. We prefer the first configuration since such objects are localized, while, in the latter case, they encompass the whole grid. This case leads to more effective implementations since fewer voxels of the volume grid are affected by the object.

Our introductory study [23] and the theoretical analysis [26] pointed out limitations imposed by the linear distance field in the vicinity of sharp surface details (edges, thin objects). Therefore, we analyze nonlinear surface density profiles (nonlinear distance fields) and we search for such combinations of profiles and reconstruction filters which lead to more precise representation of details for the same volume buffer resolution.

The signed density field cannot be used to represent zero thickness surface patches or curves since they do not have an inside or outside. Therefore, the V-model of a surface patch (curve) is defined by a density, decaying in directions perpendicular to the surface (curve). Thus, after reconstruction and thresholding, a thin solid object is in fact reconstructed (Fig. 2b).

4 V-MODELS AND RECONSTRUCTION FILTERS
Let the V-model of a solid object be described by a nonincreasing density function $V_3(d, w)$, where $d$ is the signed Euclidean distance from the object boundary ($d = 0$ on the surface) and $w$ characterizes the width of the surface transient area (Fig. 2a). Subscript 3 (2 for surface patches and 1 for curves) emphasizes the three- (two-, one-) dimensional character of the primitive. Let $V_3$ be deep enough in the object equal to some maximum density $V_{in}$ and far enough from the object be equal to $V_{out}$. From the point of view of object representation, the only constraint laid upon $V_3(d, w)$ is that a surface, defined by thresholding at $V_s = (V_{in} + V_{out})/2$, follows the genuine object surface as precisely as possible.

The following density profiles are investigated (Fig. 3):

1. Piecewise linear profile:
2. Quadratic profile:

\[
V^Q_3(t, w) = \begin{cases} 
1 & \text{if } t < -2w \\
1 - b(-t, w) & \text{if } t < 0 \\
b(t, w) & \text{if } t < 2w \\
0 & \text{otherwise}
\end{cases}
\]

where \(b(x, w) = \frac{x^2}{8w^2} - \frac{x}{2w} + \frac{1}{2}\). Profile \(V^Q_3(\cdot)\) fulfills \(C^1\) connectivity conditions:

\[
\frac{dV^Q_3}{dt}(-2w, w) = \frac{dV^Q_3}{dt}(2w, w) = 0
\]

3. Gaussian profile:

\[
V^G_3(t, w) = \int_{-\infty}^{t} G\left(\frac{x}{\sigma}\right) \, dx = \frac{1}{2} \text{erfc}\left(\frac{t}{\sigma}\right)
\]

where \(G(\xi)\) is a Gaussian with standard deviation \(\sigma = \frac{2w}{\sqrt{\pi}}\).

The Gaussian profile \(V^G_3(t, w)\) can be obtained by filtering of the object by a Gaussian filter. It has been selected since its spectral properties are the most favorable for representation of the V-model density profile. It is known from spectral analysis of signals that Gaussian has the smallest bandwidth-width product, defined as a product of variances of energy distribution along the frequency and spatial axes [37]. In other words, for the given width in spatial domain, the Gaussian has the narrowest width in frequency domain among all possible functions and, therefore, also the lowest proportion of high frequencies. This is exactly what we are looking for since we need a narrow transition between the inside and outside (favorable to precision of the object representation) and narrow bandwidth (advantageous for reconstruction). This feature is illustrated in Fig. 3c. We can see that, for the same spatial width, the Gaussian is the first filter which decays to zero in the frequency domain.

All three profiles are normalized to yield the same slope at the surface: \(\frac{dV}{dw}(0, 0.5) = -\frac{1}{2}\). Therefore, they have a different extent of \(d\), where \(V_3(d, w)\) is neither equal to \(V_{in}\) nor to \(V_{out}\):

\[
\begin{align*}
V_{in} &\geq V^L_3(d, 1) \geq V_{out} & \text{for } -w \leq d \leq w \\
V_{in} &\geq V^Q_3(d, 1) \geq V_{out} & \text{for } -2w \leq d \leq 2w \\
V_{in} &\geq V^G_3(d, 1) \geq V_{out} & \text{for } -4\sigma \leq d \leq 4\sigma
\end{align*}
\]

All profiles are antisymmetric with respect to point \((0, 0.5)\). \(V^L_3(d, w)\) is \(C^0\) continuous, with a first derivative discontinuity at \(d = -w\) and \(d = w\). \(V^Q_3(t, w)\) is the simplest \(C^1\) continuous profile being composed of two quadratic segments with a discontinuous second derivative at \(d = 0\). The Gaussian profile \(V^G_3(t, w)\) is \(C^\infty\) continuous in \((-\infty, \infty)\).

The aforementioned density functions are used also for modeling of surface patches and curves. The necessary thickness \(2\tau\) is added to the V-model by shifting the corresponding solid object profile \(V_3\) from the surface by a value \(\tau\) (Fig. 2b):

\[
V_2(t, w) = V_3(t - \tau, w)
\]
A list of the tested piecewise polynomial interpolation and derivative filters is given in Table 1. For the sake of simplicity and clarity, only coefficient matrices are displayed in the table. The corresponding segment formulae of the piecewise polynomial functions should be computed, for example, by

\[
\begin{bmatrix}
    w_{-2}(t) \\
    w_{-1}(t) \\
    w_0(t) \\
    w_2(t)
\end{bmatrix} =
\begin{bmatrix}
    a_{03} & a_{02} & a_{01} & a_{00} \\
    a_{13} & a_{12} & a_{11} & a_{10} \\
    a_{23} & a_{22} & a_{21} & a_{20} \\
    a_{33} & a_{32} & a_{31} & a_{30}
\end{bmatrix}
\begin{bmatrix}
    t^3 \\
    t^2 \\
    t \\
    1
\end{bmatrix}.
\] (13)

The number of matrix columns defines the order of the polynomial function, while the number of its rows defines its width \(2m\).

The 2EF interpolation filter \(h_{2EF}(t)\) (Fig. 4a) is a piecewise linear \(C^0\) continuous function that can reconstruct linear functions without errors. The 3EF interpolation filter \(h_{3EF}(t)\) is the well-known Catmul-Rom spline filter (\(C^1\) continuous), capable of error-free reconstruction of quadratic profiles. Both \(h_{2EF}\) and \(h_{3EF}\) filters are fixed-width filters, with width defined by resolution of the sample grid. The third filter, Gaussian

\[
h^G(t) = \frac{1}{\sqrt{\pi} \sigma} \exp\left(-\frac{t^2}{2\sigma^2}\right)
\] (14)

has adjustable width. A Gaussian is theoretically nonzero in \((-\infty, \infty)\); however, its value beyond \(t = 4\sigma\) can be safely neglected.

Reconstruction of a continuous function from sampled points implemented by the convolution (1) is a linear operation. Therefore, we can reconstruct a derivative of the sampled function by means of a filter \(d(t)\), which is defined as a derivative of some interpolation filter. Thus, for example, we get a 1EF filter \(d_{1EF}(t) = \frac{dh_{2EF}(t)}{dt}\), capable of reconstructing a profile with a constant derivative (Fig. 4b). Similarly, the Gabor derivative filter is derived from the Gaussian:

\[
d^G(t) = \frac{dh^G(t)}{dt}
\] (15)

and also features the adjustable width.

Trivariate versions of interpolation and derivative filters can be obtained by subsequent application of one-dimensional filters for each axis. Since all the filters are linear, order of application is not important. We use the same types of 1D filters for interpolation: the trilinear filter \(h_{3EF}\) and the tricubic filter \(h_{3EF}\). In the case of derivative filters, 1D interpolation and derivative filters with similar properties are combined:

1. \(d_{3EF}^1\): combination of \(h_{2EF}\) and \(d_{1EF}\),
2. \(d_{3EF}^2\): combination of \(h_{2EF}\) and \(d_{2EF}\),
3. \(d_{3EF}^3\): combination of \(h_{3EF}\) and \(d_{1EF}\), and
4. \(d_{3EF}^4\): combination of \(h^G\) and \(d^G\).

In the latter case, the same variance is used for all components. For illustration, a 2D version of the Gaussian filter and the two corresponding Gabor filters are depicted in Fig. 5.

We have just proposed the use of three different density profiles, three interpolation filters, and four derivative filters, some with variable width. This leads to an enormous amount of combinations. Therefore, in order to describe a combination of the V-model profile interpolation and derivative filter in a consistent and transparent way, we introduce notation in the form

\[
[P(w), n : I, D],
\] (16)

with the following meaning:
5 EXPERIMENTAL EVALUATION

In order to be able to reconstruct an underlying function profile from sampled data without errors, both the function and reconstruction filter must fulfill several conditions. The first and foremost one is given by Shannon’s sampling theorem, which states conditions for error free reconstruction by the ideal reconstruction filter \( \text{sinc}(t) = \frac{\sin(t)}{t} \). The minimal sampling frequency \( f \), which would allow an
error-free reconstruction of a signal by the sinc filter, should be at least equal to double its maximum frequency $f_m$.

The first step of the V-model design is, therefore, the assessment of their frequency properties which enables us to estimate their optimal width. Unfortunately, frequency spectra of all profiles are unbounded. Thus, we define

$$p_x(w) = \frac{\int_{-\infty}^{\infty} |F(\omega, w)|^2 d\omega}{\int_{-\infty}^{\infty} |F(\omega, w)|^2 d\omega} \times 100\%$$  \hspace{1cm} (17)$$

as a fraction of total energy of the signal below the Nyquist rate $\pi$ (assuming the fixed sampling frequency $1VU$), to help us in a profile comparison and an estimation of the optimal widths. Fig. 6 confirms the expected favorable properties of the Gaussian since its $p_x(w)$ nearly approaches 100 percent at width below 1VU. The quadratic profile is worse, approaching the 100 percent value only at widths above 1.5VU. Unfortunately, we do not know how significant is the deviation of $p_x(w)$ from the ideal 100 percent value. Therefore, the estimated widths can serve only as guides in a precise experimental evaluation.

The graph shows significantly worse results for the linear profile. However, if we assume a reconstruction filter, which is narrower than the width of the profile, the situation is changed. Such a filter does not “see” the discontinuities at $x = -w$ and $x = w$ and, therefore, the profile appears to be infinitely wide, which implies infinitely narrow spectrum. Therefore, even the central differences filter with poor frequency properties is able to reconstruct the normal correctly if the profile is wide enough (being estimated to 1.7VU [23]).

The ideal interpolation sinc filter, as well as its derivative cosc, have both an unbounded spatial domain and cannot be used in practice. Practically usable approximations of these ideal filters are analyzed and compared both in the frequency domain [38], [29], [30] and in the spatial domain [31]. These analyses usually compare filters among themselves, either quantitatively or by means of certain error measures, assuming a general band limited signal. In our case, in order to ensure sufficient surface smoothness and normal precision, we need to find the edge of applicability of the proposed filters and V-models and, consequently, we cannot rely on general results, which are not specific enough for our purpose. Therefore, the only possibility for evaluating properties of the V-model/filter pairs is to set up an experiment and compute the error measures of interest directly in the environment where they are supposed to be used.

In the case of the V-model, we have, on one hand, several density profiles, each having a variable width. On the other hand, we have reconstruction filters, again some of them with a variable width. We optimize the profile/filter combinations from the point of view of the following tasks:

- Reconstruction of a planar (low curvature) surface of a solid, or of a surface patch. The aim is to find the profile/reconstruction combination which leads to as narrow a surface transition width as possible, while keeping reconstruction errors in acceptable range.
- Assessment of applicability limits of a thus selected combination for high curvature object details.

Fig. 5. (a) 2D Gaussian filter, (b) 2D Gabor filter in $x$ direction, and (c) 2D Gabor filter in $y$ direction.
The signed Euclidean distance is positive if the detected surface point lies outside of the object and negative if it is inside. Therefore, a detected surface with an equal proportion of positive and negative deviations results in a small \( E_p \) value, but with high variance

\[
E_p = \frac{1}{N} \sum_{i=1}^{N} d_i,
\]

where \( d \) is the signed Euclidean distance. The condition of normal precision can be formulated using the mean angle between the correct and estimated normal at the detected surface point. Since the angle can have only positive values, its mean is used as the normal error measure:

\[
E_n = \frac{1}{N} \sum_{i=1}^{N} \left( \hat{n}_i \cdot \hat{n}_e \right),
\]

where \( \hat{n}_i \) is true surface normal, \( \hat{n}_e \) is estimated surface normal by the tested reconstruction technique, and \( N \) is the number of test values.

The detected object surface must have both a precise position and be as smooth as the surface of the geometric object. Error measure connected with the first demand can be as smooth as the surface of the geometric object. Error measure connected with the first demand can be formulated using the number of test values.

Error measure connected with the second demand can be similarly formulated as in the previous case by means of the signed Euclidean distance \( d \) between the true and estimated surface points:

\[
d = |p_i - p_e|.
\]

Then, the position error is given by

\[
E_p = \frac{1}{N} \sum_{i=1}^{N} d_i,
\]

where \( d_i \) is the signed Euclidean distance. The signed Euclidean distance is positive if the detected surface point lies outside of the object and negative if it is inside. Therefore, a detected surface with an equal proportion of positive and negative deviations results in a small \( E_p \) value, but with high variance

\[
E_p = \frac{1}{N} \sum_{i=1}^{N} (d_i - E_p)^2.
\]

5.1 Error Measures

The condition of normal precision can be formulated using the mean angle between the correct and estimated normal at the detected surface point. Since the angle can have only positive values, its mean is used as the normal error measure. Error measure connected with the first demand can be as smooth as the surface of the geometric object. Error measure connected with the first demand can be formulated using the number of test values.

The detected object surface must have both a precise position and be as smooth as the surface of the geometric object. Error measure connected with the first demand can be similarly formulated as in the previous case by means of the signed Euclidean distance between the true and estimated surface points:

\[
d = |p_i - p_e|.
\]

Then, the position error is given by

\[
E_p = \frac{1}{N} \sum_{i=1}^{N} d_i.
\]

The signed Euclidean distance is positive if the detected surface point lies outside of the object and negative if it is inside. Therefore, a detected surface with an equal proportion of positive and negative deviations results in a small \( E_p \) value, but with high variance

\[
E_p = \frac{1}{N} \sum_{i=1}^{N} (d_i - E_p)^2.
\]

5.2 Surface and Normal Error for Low Surface Curvatures

In the first experiment, we tested dependency of the surface error measures (mean position error \( E_p \) and its variance \( E_{p^2} \)) and the normal error measure \( E_n \) on the width of the V-model surface profile. We tested all combinations of the proposed profiles and filters, both for solid \( V_1 \) and surface patch \( V_2 \) density profiles. As a test object, a sphere of diameter 45VU was used. The radius was selected to be large enough in comparison to the filter radius support. In order to suppress artifacts introduced by the surface curvature, a planar object would be better from this point of view. However, a sphere enables us to study the behavior of the filters with respect to different surface normal orientations.

Both for solid and surface patch objects (except of the Gaussian surface patches), the measured mean was lower than 0.005VU for widths \( w > 0.8 \) (Fig. 7a, b). For narrower profiles, in the case of solids, the value was still very low. However, surface objects showed higher error for \( w < 0.8 \) and holes in the surface were often detected. Deeper insight into results is given by the dependency of distance variance (Fig. 7b) on the transition area thickness \( w \). The plotted curves were only for solid objects since the variance was approximately equal for both \( V_1 \) and \( V_2 \) cases for the given profile/filter combination and for width above 0.7VU.
However, for narrower profiles, error connected with $V^2$ rapidly increased, indicating loss of surface smoothness and eventual surface holes.

How can we explain the fact that $E_v$ is falling to negligible values, indicating ideally smooth surface, for the linear profile $V^L$ and quadratic profile $V^Q$ are depicted, together with their interpolation, for different displacement with respect to the sampling grid. As far as two adjacent samples of $V^L$ lie on the same line of the piecewise linear profile, no reconstruction error is introduced by the linear filter. This always holds for all points between the pair of samples situated left and right with respect to the genuine surface position if the profile width is at least $2VU$ wide (i.e., for $w > 1$). Therefore, $E_v$ falls to negligible values for widths $w$ above 1 (Fig. 7b). If $w$ decreases below 1, some detected surface points fall between samples, which are taken from different lines of the profile. Therefore, their reconstruction error increases.

Fig. 8b shows that the reconstruction error is zero only for those quadratic profiles with no displacement with respect to the sampling grid. Moreover, even in this case, the reconstruction error is zero only for the density level $(V_{in} + V_{out})/2$. This behavior is confirmed in Fig. 9, where actual distance between the true surface and surface estimated by ray tracing is depicted as a function of standard parameterization of a sphere. The concentric circles of high distance error with thickness inversely proportional to the radius of the circle witness the presence of the error just described.

In comparison to other surface profiles, mean surface error $E_p$ of the Gaussian surface patch $V^G$ shows significantly different behavior (Fig. 7c). We see a systematic increase of the error for increasing thickness of the transient area. This feature is due to nonsymmetry of the Gaussian. This nonsymmetric profile is interpolated by a symmetric interpolation kernel, which results in blurring in directions perpendicular to the surface and, thus, to a shift of the detected surface in the direction from the true surface. The systematic surface shift is proportional to the transient area width $w$.

Results of the normal error experiment are depicted in Fig. 10. There we analyze the measured dependency of the normal error $E_n$ on surface thickness independently for each surface profile.

We have shown earlier in this section that the combination of a linear profile with the linear reconstruction filters leads to exact reconstruction of the surface point, as soon as the width of the transition region is larger than 2 ($w > 1$). The same also holds for the derivative filter $d_{1EF}^3$ since it reconstructs the derivative only from samples situated at vertices of the smallest grid cell with side $1VU$ surrounding the point. This is illustrated in Fig. 10a by the curve $[L(w), 3 : [2EF]^-]$. We can see that it actually decays to minimum at $w = 1$. However, this minimal error is kept for all radii $w > 1$. This behavior is caused by nonlinearity introduced by a nonzero object surface curvature and decreases to zero for a planar surface. Specifically, the error is introduced since the 1EF gradient filter does not correctly reconstruct nonlinear profiles. It results in a typical strip error pattern (Fig. 11a). Frequency of the strips is inversely proportional to the resolution of the volume buffer, which
confirms that the origin of the error is in erroneous interpolation between samples.

Minimal error of a higher order filter $d^{2\ EF}_3$ completely decreases to zero for radii $w > 1.7$ (Fig. 10a, curve [L(w), 3 : −[2EF]). We reported the same result earlier for the trilinearly interpolated normal estimated by the central differences filter ($d^{CD}_3$, [23]). Moreover, all curves for the $d^{2\ EF}_3$ and $d^{CD}_3$ filters were exactly the same, which indicates that they are only different implementations of the same filter. Negligible normal error for $w > 1.7$ can be explained by the fact that the whole (linear) filter kernel fits into the linear transient area [23].

Earlier in this section, by means of Fig. 8, we have explained why interpolation by the cubic filter introduces error in surface position detection in the case of a quadratic profile $V^Q$. The same also holds for normal estimation. Moreover, since we also interpolate at off-surface points, where the error is larger than near the surface, we get an even worse result. This is illustrated in Fig. 10a by curve [Q(w), 3 : −[3EF]. We can see in Fig. 11b that the corresponding error pattern is in this case different in comparison to the $d^{1\ EF}_3$ filter (Fig. 11a). Since the tricubic interpolation $d^{3\ EF}_3$ was involved, we observe similar concentric circles as in Fig. 9b. However, there is an

Fig. 8. Interpolation of (a) the linear profile $V^L$ by the linear filter $h^{2\ EF}$, and (b) the quadratic profile $V^Q$ by the cubic filter $h^{3\ EF}$ for different shifts of the profile with respect to the sampling grid. Notice error introduced for the shifted $V^Q$ profiles.
additional low frequency pattern superimposed, indicating dependency of the error magnitude on surface orientation. The profile is very similar to that observed earlier [23], which was caused by an improper combination of the surface profile and derivative filter (attempt to reconstruct a Gaussian profile by a linear filter). The situation is also similar in this case, though the combination profile/filter is different.

Interesting results were obtained by the combination of Gaussian profile/Gabor derivative filter. The first observation is robustness of the Gabor filter with respect to the precision of surface point estimation. Curves in Fig. 10b were exactly the same for different surface interpolation techniques. The second observation is that Gabor filter, due to its adjustable support, can reconstruct the normal with higher precision than the aforementioned fixed width filters. The best results were obtained for Gabor $\sigma > 1.0$. For these filters, the mean normal error was below 0.1 degree for solids with transient profile $w > 0.6$. Behavior of normal error for surfaces was very similar to that of solids for profiles with width $w > 1.1$. For narrower profiles, the error increased up to 0.3 degree at $w = 0.6$. For even narrower profiles with $w < 0.6$ holes in the surface were detected.

5.3 Surface and Normal Error for High Surface Curvatures

In Section 5.1, we introduced a methodology for performance comparison of various V-model/reconstruction combinations and the corresponding error measures. Earlier in this section, we performed the first part of the tests, aimed at surface thickness analysis. Thus, for each V-model/reconstruction combination, we have some numbers and/or graphs which allow us to order them from the point of view of quality. In order to proceed with our test, we have to decide which are the values which set a frontier between images of acceptable and poor quality.

As the threshold between good and bad normal estimation, we use the value 0.01 radians (0.5 degree), which was reported as a result of empirical tests by Deering [39]. We are not aware of any published estimates of what concerns the surface position error. However, 1/5 of a voxel seems to be a reasonable value.

These threshold values enable us to fully discard quadratic profile from further consideration. In the following, we study behavior of three combinations of V-model/reconstruction pairs with respect to variable surface curvature:
Gaussian profile with width \( w \in (0.6, 1) \), cubic interpolation filter, and Gabor derivative reconstruction filter (\( \{ G(w), 3 : 3EF, G(1.0) \} \)).

- Linear profile with width \( w > 1 \), linear interpolation, and constant normal estimation (\( \{ L(w), 3 : 2EF, 1EF \} \)), and

- Linear profile with width \( w > 1.5 \), linear or cubic interpolation, and linear normal estimation (\( \{ L(w), 3 : 2EF, 2EF \} \) and \( \{ L(w), 3 : 3EF, 2EF \} \)).

We will not study the Gaussian interpolation filter \( dG \) in spite of the fact that it results in the smoothest surfaces (lowest variance). However, it is the widest filter. For example, for \( \sigma = 1 \), at least width \( 2m = 6 \) is necessary. Evaluation of such a filter is about 15 times more expensive than filter with \( m = 1 \), and 5 times more expensive in comparison to \( m = 2 \). This leads to too low a performance of a ray tracer since the interpolation filter should be evaluated many times for each projection ray.

Fig. 12 and Table 2 summarize results obtained for solid test spheres with radii starting from 1VU. For lower radii, some rays totally skipped the object and, thus, the resulting values were not reliable.

The linear interpolation technique \( d_3^{EF} \) shows significantly worse performance than \( d_1^{EF} \) and \( d_2^{EF} \) in the area of smallest diameters (Fig. 12a). This means that \( d_3^{EF} \) introduces distortion at sharp object details. Negative value of the error indicates that details with positive curvature are eroded, while that with negative curvature (holes) are filled.

Normal estimation tests (Fig. 12b) show poorest performance of the \( d_1^{EF} \) filter. We know that this filter is theoretically able to reconstruct only a constant gradient and, therefore, the result is not surprising. This constant gradient condition is fulfilled, at least approximately, for objects with a low surface curvature. However, for small objects, this condition is severely violated, which results in large normal error.

The Gaussian filter gives low error even for the smallest spheres. Error of the linear \( d_3^{EF} \) filter grows moderately below sphere radius 8 and reaches its maximum 1 degree for the smallest tested spheres. We also tested the combination of the central differences filter \( d_{CD} \) with cubic interpolation \( h_3^{MF} \) of thus obtained gradient components. However, results were much worse than for the linear \( d_3^{EF} \) filter.

6 OBJECT SPECIFIC V-MODEL DESIGN

In previous sections of the paper, we introduced the concept of V-models, which enabled us to separately investigate general rendering issues of voxelization from particular object properties. We proposed and verified a V-model, based on a density profile which decays in the object surface area proportionally to the distance from the surface.
We obtained best results with the linear profile with width 2*1.7VU and the Gaussian profile with parameter sigma slightly below 1VU.

For each sample point, the proposed technique requires knowledge of its distance to the surface of the voxelized primitive. In this section, we briefly summarize several object specific procedures which allow us to determine this distance. Often, there are two or even more such techniques for a given object. However, they are usually not equivalent from the point of view of efficiency.

6.1 Direct Distance Computation

Distance of each grid voxel to the nearest object point can be computed for any kind of objects [27]. However, in practice, it might lead to minimization of systems of nonlinear functions, which is usually costly and may lead to incorrect results (trapping in local minima). For example, this approach was used for voxelization of superellipsoids to distance volumes [27].

In some special cases, the analytic equation of an object directly defines the desired distance (plane, sphere, or torus [26]). This is a special case of more general distance computation by means of an implicit function \( g(p) = 0 \) and its gradient magnitude:

\[
d(p) = \frac{g(p)}{\|\nabla g(p)\|}. \quad (22)
\]

This formula assumes nonzero gradient magnitude and a possibility to linearize the implicit function in the surface vicinity within a small error, which is not always fulfilled. Violation of the second condition indicates that the selected sampling frequency is too low to represent the object correctly.

6.2 Voxelization of Surface Patches and Curves

A technique for voxelization of parametric surfaces, termed 3D splatting was introduced earlier [20]. First, the object was point sampled into its binary representation and, second, the contribution of each nonzero discrete point was accumulated to grid points within the reach of its support. However, sampling the object with unit steps has proven to be insufficient. Therefore, a superbuffer technique was introduced, increasing resolution of the binary volume \( n \) times. Due to extreme memory demands of the algorithm (\( n^3 \) increase) only \( n = 4 \) times higher resolution for models with reasonable resolution was possible.

As we have shown, in order to get correct images, the transient region density profile should be dependent on the distance from the surface. To fulfill this condition, we proposed a different approach [23], based on sampling of the surface patch and, instead of summing of the splat contributions, we registered their maximum value. To avoid the superbuffer, the algorithm sampled the continuous representation of the object directly. This technique ensured that density in the surface vicinity had exactly the same profile as the splat, which was initialized according to the selected profile type and its radius.

The question of optimal sampling of the parametric domain of the patch, which was represented by a two dimensional interval \( I \), was solved by its binary hierarchical subdivision (Fig. 13). In each step, the actual patch was divided into two smaller patches, until the patch size \( s_p \) reached some predefined value. Then, for each leaf of the tree, a single sample was taken in its center. The optimal sample density was estimated to be about two samples/VU.

<table>
<thead>
<tr>
<th>Object</th>
<th>( w )</th>
<th>( E_p )</th>
<th>( E_v )</th>
<th>( E_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear profile</td>
<td>( L(w), (2, 3) : E13/E12 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol, 51.2</td>
<td>1.0</td>
<td>-0.001</td>
<td>0.003</td>
<td>1.86</td>
</tr>
<tr>
<td>Sol, 51.2</td>
<td>1.7</td>
<td>-0.000</td>
<td>0.000</td>
<td>0.04</td>
</tr>
<tr>
<td>Sol, 1.0</td>
<td>1.0</td>
<td>-0.036</td>
<td>0.031</td>
<td>2.88</td>
</tr>
<tr>
<td>Sol, 1.0</td>
<td>1.7</td>
<td>-0.033</td>
<td>0.018</td>
<td>0.92</td>
</tr>
<tr>
<td>Surf, 51.2</td>
<td>1.0</td>
<td>-0.01</td>
<td>0.009</td>
<td>3.20</td>
</tr>
<tr>
<td>Surf, 51.2</td>
<td>1.7</td>
<td>-0.001</td>
<td>0.001</td>
<td>0.06</td>
</tr>
<tr>
<td>Gaussian profile</td>
<td>( G(w), (2, 3) : G(1) )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sol, 51.2</td>
<td>0.5</td>
<td>-0.064</td>
<td>0.056</td>
<td>0.28</td>
</tr>
<tr>
<td>Sol, 51.2</td>
<td>0.75</td>
<td>-0.033</td>
<td>0.030</td>
<td>0.06</td>
</tr>
<tr>
<td>Sol, 51.2</td>
<td>1.0</td>
<td>-0.002</td>
<td>0.017</td>
<td>0.03</td>
</tr>
<tr>
<td>Sol, 1.0</td>
<td>0.5</td>
<td>-0.007</td>
<td>0.04</td>
<td>0.18</td>
</tr>
<tr>
<td>Sol, 1.0</td>
<td>0.75</td>
<td>-0.025</td>
<td>0.04</td>
<td>0.06</td>
</tr>
<tr>
<td>Sol, 1.0</td>
<td>1.0</td>
<td>-0.02</td>
<td>0.03</td>
<td>0.02</td>
</tr>
<tr>
<td>Surf, 51.2</td>
<td>0.5</td>
<td>-0.027</td>
<td>0.09</td>
<td>1.13</td>
</tr>
<tr>
<td>Surf, 51.2</td>
<td>0.75</td>
<td>-0.023</td>
<td>0.036</td>
<td>0.23</td>
</tr>
<tr>
<td>Surf, 51.2</td>
<td>1.0</td>
<td>-0.008</td>
<td>0.0015</td>
<td>0.16</td>
</tr>
</tbody>
</table>

The first column defines type and radius of the test sphere and the \( w \) column defines the surface profile width.

Fig. 13. Sampling of a spherical surface with binary subdivision. Samples are positioned at the centers of the quadrilaterals. (For the sake of clarity, the sampling density for this picture was set to a significantly lower value than the estimated optimum).
6.3 Voxelization of Planar Primitives
An incremental technique for voxelization of triangles, which can be easily extended also for polygons and even solid polyhedra, can be used. The technique scans the bounding box of the triangle pixel by pixel, updating distances 1) from the plane defined by the triangle, 2) from the triangle edge lines, and 3) from the triangle vertices. In addition, it also computes the line parameter of an intersection point of each edge line with a plane, perpendicular to it and passing through the actual grid point. These parameters are maintained incrementally and enable correct and effective computation of the distance from the triangle, which is used to compute the density according to the selected V-model.

7 IMPLEMENTATION AND SUMMARY
The proposed techniques were implemented as a C++ class library [40], which provides for voxelization of analytic objects based on different kinds of V-models. Fig. 14 presents a scene, composed of three test objects:

1. A reflective sphere, defined by its analytic equation;
2. A higher complexity polygonal toy car model (9,572 triangles);
3. A scene consisting of a sphere and triangular objects.

<table>
<thead>
<tr>
<th>Rendered object</th>
<th>V-model/Geometry</th>
<th>Volume Resolution</th>
<th>Voxelization time [s]</th>
<th>Rendering time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Man</td>
<td>Linear</td>
<td>118 × 70 × 166</td>
<td>15.1</td>
<td>10.4</td>
</tr>
<tr>
<td></td>
<td>Linear</td>
<td>226 × 130 × 324</td>
<td>33.8</td>
<td>24.5</td>
</tr>
<tr>
<td></td>
<td>Gaussian</td>
<td>118 × 68 × 166</td>
<td>13.3</td>
<td>12.8</td>
</tr>
<tr>
<td></td>
<td>Gaussian</td>
<td>226 × 128 × 322</td>
<td>30.4</td>
<td>26.7</td>
</tr>
<tr>
<td></td>
<td>Geometry</td>
<td>2656 triangles</td>
<td>-</td>
<td>25.0</td>
</tr>
<tr>
<td>Car</td>
<td>Linear</td>
<td>106 × 165 × 90</td>
<td>37.0</td>
<td>25.7</td>
</tr>
<tr>
<td></td>
<td>Linear</td>
<td>204 × 319 × 170</td>
<td>72.1</td>
<td>40.0</td>
</tr>
<tr>
<td></td>
<td>Gaussian</td>
<td>106 × 165 × 90</td>
<td>32.8</td>
<td>33.7</td>
</tr>
<tr>
<td></td>
<td>Gaussian</td>
<td>204 × 318 × 170</td>
<td>64.5</td>
<td>49.8</td>
</tr>
<tr>
<td></td>
<td>Geometry</td>
<td>9572 triangles</td>
<td>-</td>
<td>235.0</td>
</tr>
<tr>
<td>Sphere</td>
<td>Linear</td>
<td>86 × 86 × 86</td>
<td>3.6</td>
<td>27.6</td>
</tr>
<tr>
<td></td>
<td>Linear</td>
<td>166 × 166 × 166</td>
<td>17.6</td>
<td>35.2</td>
</tr>
<tr>
<td></td>
<td>Gaussian</td>
<td>88 × 88 × 88</td>
<td>6.3</td>
<td>34.5</td>
</tr>
<tr>
<td></td>
<td>Gaussian</td>
<td>168 × 168 × 168</td>
<td>27.5</td>
<td>43.3</td>
</tr>
<tr>
<td></td>
<td>Geometry</td>
<td>1 sphere</td>
<td>-</td>
<td>5.0</td>
</tr>
<tr>
<td>Scene</td>
<td>Linear</td>
<td>-</td>
<td>55.7</td>
<td>44.6</td>
</tr>
<tr>
<td></td>
<td>Linear</td>
<td>-</td>
<td>123.5</td>
<td>60.8</td>
</tr>
<tr>
<td></td>
<td>Gaussian</td>
<td>-</td>
<td>52.4</td>
<td>59.2</td>
</tr>
<tr>
<td></td>
<td>Gaussian</td>
<td>-</td>
<td>122.4</td>
<td>71.3</td>
</tr>
<tr>
<td></td>
<td>Geometry</td>
<td>12229 objects</td>
<td>-</td>
<td>488.0</td>
</tr>
</tbody>
</table>

Results in the “Man,” “Car,” and “Sphere” rows reflect rendering of individual objects in the same position as depicted in the figure. In the case of the “Scene” row, they were rendered simultaneously.
3. A lower complexity polygonal toy figure model (2,656 triangles).

For the purpose of testing, we rendered the scene both directly, using the geometric models, and by means of object voxelization. The background, a plane with checkerboard texture mapping, was rendered using only its geometric representation. Detailed results of the tests are summarized in Table 3, for an SGI Challenge, equipped with a 196 MHz R10000 processor and 3,072 MBytes of main memory. The OORT ray-tracer [41] was used for rendering, modified by adding classes for processing of volumetric objects and volumetric texture maps. The surface of the volumetric object was defined by interpolation corresponding to the selected V-model, and by subsequent thresholding at 50 percent of the maximum object intensity. The position of the intersection between the ray and the surface was first roughly approximated by a discrete ray traversal technique accelerated by means of cubic macro regions [35], and then refined by bisection.

Each object was voxelized using two V-models: linear, with transition area thickness 1.8VU and Gaussian with \( \sigma = 1.0 \) (Section 4), and two volume buffer resolutions. Trilinear interpolation and normal estimation by central differences were used in the linear case, while tricubic interpolation and the Gabor gradient filter were used in the case of the Gaussian V-model. In correspondence with the results presented earlier in this paper, the high resolution Gaussian V-model yielded superior results, with high fidelity of model surface details and sharp edges. Rendering of the Gaussian V-models was slower by 20-35 percent than that of the linear models, due to the higher order reconstruction technique involved. Voxelization for both cases was equally fast in the case of triangular models since lookup tables were involved and no profile function evaluation was done for individual voxels. This was not the case for the solid sphere, which resulted in longer voxelization times for the Gaussian model (60-100 percent).

Doubled resolution of the volume buffer resulted in approximately doubling the voxelization times for the triangular models. However, voxelization of the sphere took 4.6-6 times longer. On the other hand, due to the accelerated ray traversal technique involved, rendering time of twice higher resolution volume buffers was only 20-35 percent slower for the whole scene.

Interesting results were obtained through comparison of rendering times for the voxelized model (the slowest, high resolution Gaussian case, with voxelization and rendering times added) and the original geometric model. In this comparison, rendering via the voxelization was approximately 2.5 times faster for the whole scene. However, if we take only rendering into account—which is meaningful if several projections are rendered—this ratio increases to nearly 7. We, of course, admit that better acceleration techniques for ray-tracing of geometric objects, than those implemented in OORT, can change this ratio. The acceleration decreases with the decreasing complexity of objects. For simple objects as, for example, the sphere, the voxelized model even slows down the rendering (5-9 times, according to the V-model).

We further performed exhaustive testing of a large number of combinations of surface profiles, as well as interpolation and derivative filters. The results were summarized and analyzed earlier in Section 5. The best results were obtained for V-model with linear density profile \( V^L(t) \) and Gaussian density profile \( V^G(t) \), where \( t \) is distance measured from the surface of the primitive. Optimal width of the linear V-model profile was estimated to be 1.7VU. This value, obtained for the linear derivative filter \( \delta^L_{EF} \), is the same as the earlier reported result for the central differences filter [23]. Trilinear density interpolation, advocated in [23], gives sufficiently good results for this profile for surface curvatures, which can be approximated by a sphere with a radius not lower than 10VU. However, for sharper details, down to radius 1VU, the tricubic Catmull-Rom filter \( h^L_{EF} \) gave much better results, with maximal surface position error about 0.05VU even for the smallest tested sphere.

The very best results were obtained for the combination of the Gaussian density profile \( D^G(w), w \in (0.8, 1.1) \), combined with tricubic density interpolation \( \delta^L_{EF} \) and Gabor gradient filter \( \delta^G_{EF} \). In this case, even for the smallest sphere with radius 1.0, negligible normal error was measured and, this case, the maximal surface position error was about 0.05VU.

Width of the V-model density profile approximately defines maximal sharpness of surface details (edges, vertices) of the voxelized solid. For the Gaussian profile, this value was estimated to be approximately one half of the optimal width of the linear profile. This means that sharper edges and vertices can be obtained if the Gaussian V-model is involved in the voxelization procedure. It, however, results in longer rendering times since the computationally more expensive Gabor filter should be used for gradient reconstruction. Lower width of the Gaussian profile has, of course, positive consequences also for voxelization of surface patches and curves since, for the same volume buffer resolution, it enables voxelization of them with half thickness, in comparison to the case of the linear profile.

ACKNOWLEDGMENTS

This work has been supported by U.S. Office of Naval Research grant N000149710402, NRL grant N00014961G015, and U.S. National Science Foundation grant MIP9527694. The polygonal toy models were provided by 3DCAFE.

REFERENCES


