Precision Isosurface Rendering of 3-D Image Data

Philippe Thévenaz and Michael Unser, *Fellow, IEEE*

***Abstract*—We address the task of rendering by ray tracing the isosurface of a high-quality continuous model of volumetric discrete and regular data. Based on first principles, we identify the quadratic B-spline as the best model for our purpose. The nonnegativity of this basis function allows us to confine the potential location of the isosurface within a binary shell. We then show how to use the space-embedding property of splines to further shrink this shell to essentially a single voxel width. Not all rays traced through a given shell voxel will intersect the isosurface; many may only graze it, especially when the ray-tracing vantage point is close or within the volume to render. We propose here an efficient heuristic to detect those cases. We present experiments to support our claims.**

***Index Terms*—Interpolation, order of approximation, ray tracing, visualization.**

I. Introduction

We consider the problem of displaying the isosurface of three-dimensional data f, which is defined by the frontier Ω of the continuously defined solid f that satisfies $\{\mathbf{x}|f(\mathbf{x}) < f_0\}$, where f_0 is some arbitrary threshold. The main difficulty of this problem is that our data are discrete; they are given by a three-dimensional (3-D) regular array of measurements $f(\mathbf{k})$, $\mathbf{k} \in \mathbb{Z}^3$. Discrete 3-D data are very common, especially in medical imaging where the output of most scanners consists of voxel arrays. Unfortunately, the set defined by the frontier of $\{\mathbf{k}|f(\mathbf{k}) < f_0\}$ is discrete for discrete data and does not lead to a well-defined continuous frontier Ω.

An early solution to this problem is called marching cubes [1]; it has attracted some interest by proposing a heuristic to perform the conversion of the discrete volumetric data $f(\mathbf{k})$ into a list of polygonal faces that can be used to build a continuously defined surface. Unfortunately, this conversion is ambiguous in some cases [2]; moreover, the number of faces may grow very large, which may become a hindrance to the rendering speed. In addition, the planarity of the faces stands in contradiction with the smooth, organic surfaces that one expects from biomedical volumes. In such a context, this precludes the use of marching cubes and related methods like marching tetrahedra [3] and skeleton climbing [4].

More modern methods [5]–[8] rely on fitting the discrete data $f(\mathbf{k})$ with a continuous model $f(\mathbf{x})$, which restores the continuous definition of Ω. Then, it becomes possible to render—by a technique such as ray tracing—the continuous surface defined by the frontier of $f(\mathbf{x}) < f_0$. The choice of the continuous model $f(\mathbf{x})$ is such that the continuous data $f(\mathbf{x})$ are related to the discrete data $f(\mathbf{k})$ by $f(\mathbf{x}) = \sum_{\mathbf{k}\in\mathbb{Z}^3} f(\mathbf{k})\varphi_{\mathrm{int}}(\mathbf{x}-\mathbf{k})$. This choice is appropriate because it ensures the linearity of the model with respect to the data.

In recent years, a lot of attention has been devoted to investigate which specific continuous function φ_{int} would result in the best performance for visualization [7]–[14]. This issue is relevant because the ideal basis function φ_{int} (sometimes called a filter) is known to be $\mathrm{sinc}(x) = (\sin \pi x)/(\pi x)$ and cannot be used in practice in reason of its slow, infinite decay. Since suboptimal basis functions must necessarily be used instead, it is important to be able to rank them in terms of quality and efficiency. This ranking is the study subject of the theory of approximation where a single number—the order of approximation L—characterizes the quality of a basis function [15]. One of several interpretations of L expresses that any polynomial of degree up to $(L-1)$ can be represented exactly as a weighted sum of shifted basis functions, or, differently put, $a_0 + \sum_{n=1}^{L-1} a_n x^n = \sum_{k\in\mathbb{Z}} c_k \varphi(x-k)$. To understand why this is important, it is enough to remember that a truncated Taylor series is a polynomial, and that its reminder gets smaller as more terms are considered. Since any basis function of order L can be expressed as the convolution of a generalized function (called a distribution) with a B-spline of degree $(L-1)$, and since those basis functions that are shortest are necessarily made of nothing else but a sum of B-splines and of their derivatives [16], then it should come as no surprise that splines play a major role in problems where the order of approximation is a relevant issue. This is also the case here.

The present paper is organized as follows: in Section II, we invoke first principles to determine which is the best basis function. To our knowledge, this is the first time that this approach is pursued in the context of rendering. The function that is singled out is a quadratic B-spline which is not interpolating; yet, we want to interpolate the data in an exact fashion. We show how to reconcile the two aspects. In Section III, we take advantage of the nonnegativity of B-splines to achieve a large reduction of the computational complexity. A second important property of B-splines is space embedding. We put it to good use in Section IV to reduce even further the number of isosurface candidates. Up to this point, all operations are view-independent and can be performed off-line. In Section V, this is no more the case; to gain run-time efficiency, we propose there a heuristic for the early rejection of potential candidates. We present an illustrated and detailed analysis of our rendering method in Section VI. We conclude in Section VII.

II. Interpolation Model

A. First Principles

1) Fidelity: Our first and foremost requirement is that the isosurface Ω be faithful to the original data from which we are

Manuscript received August 27, 2002; revised March 12, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Philippe Salembier.

The authors are with the Biomedical Imaging Group, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland (e-mail: philippe.thevenaz@epfl.ch).

Digital Object Identifier 10.1109/TIP.2003.814240

1057-7149/03$17.00 © 2003 IEEE

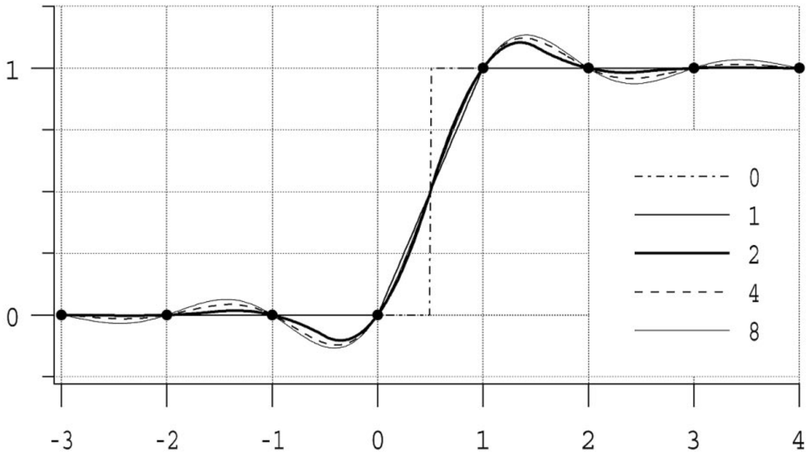


Fig. 1. Interpolation of a sampled step function by splines of several degrees. Specifically for B-splines β^n of degree n, the order of approximation $L = n + 1$ is one more than the spline degree.

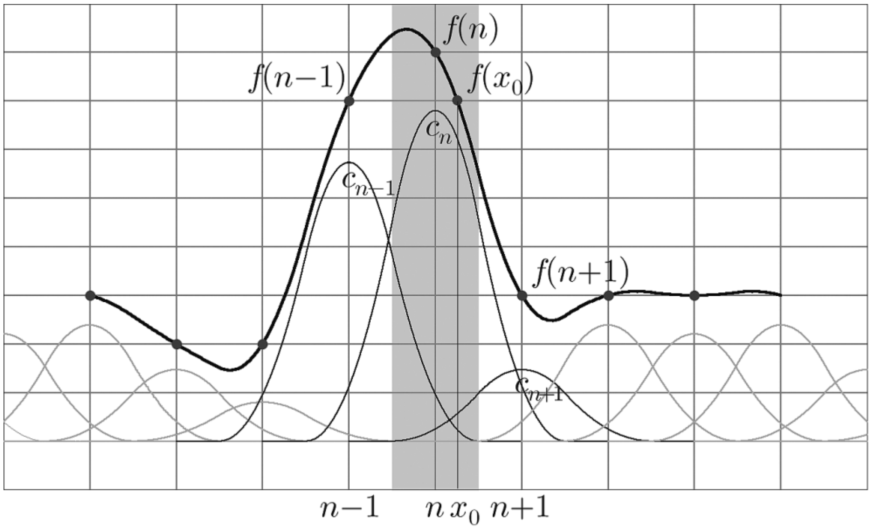


Fig. 2. Continuous model $f(x) = \sum c_k \beta^2(x - k)$. Here, the set of coefficients that influence the value $f(x_0)$ is $\mathbb{C} = \{c_{n-1}, c_n, c_{n+1}\}$. The gray area shows the domain of abscissa $\mathbb{X} = [n - (1/2), n + (1/2)]$ that share the same set of coefficients $\mathbb{C}$ for the computation of $f(x)$, $x \in \mathbb{X}$.

given the samples $f(\mathbf{k})$. For clinical applications, it is vital that the generated images represent reality [17]. This quest for high quality can be fulfilled only by selecting a basis function φ_{int} with a large order of approximation. Low-order basis functions, such as those that correspond to nearest-neighbor and (tri)linear interpolation, have a short support and do not oscillate. Nevertheless, similar to sinc, good candidates φ_{int} tend to have a larger support and do necessarily oscillate, as shown in Appendix A. A direct consequence is that the model may overshoot or undershoot the data; we illustrate both situations in Figs. 1 or 2.

This effect spoils the determination of an isosurface: in its absence, it is trivial to show that no part of the isosurface can be located between adjacent samples $\{k, k+1\}$ such that $\text{sign}(f(k) - f_0) = \text{sign}(f(k+1) - f_0)$; but, when the basis function oscillates, the existence of a location $k < x < k+1$ such that $f(x) = f_0$ cannot be excluded, even when $\text{sign}(f(k) - f_0) = \text{sign}(f(k+1) - f_0)$. This counter-intuitive fact, which happens only for basis functions of sufficiently high quality, can be illustrated as follows: consider for a moment the trivial data $f(k)$, $k \in \mathbb{Z}$, that consist of a unique impulse $f(k) = \delta_k = \{\ldots, 0, 0, 1, 0, 0, \ldots\}$. A cursory inspection of this discrete data would seem to imply that the width of the domain defined by $f < f_0$ cannot be large for $0 < f_0 < 1$. Certainly—at least so says a mistaken intuition—no solution of $0 < f(x) = f_0 < 1$ should exist for x sufficiently far away from the origin. But using sinc as basis function, which is the *best possible* theoretical choice, we determine that the continuous model for the specific data of this example is $f(x) = \text{sinc}(x)$. Thus, when the arbitrary value f_0 happens to be close enough to zero, the support of the domain of $f < f_0$ can be made as large as one wishes—it tends to infinity for $f_0 \to 0$. It follows that a trivial inspection of the discrete data is not enough to predict the location of an isosurface, even approximately.

To resolve this issue, one can resort to brute-force approaches such as making use of a large oversampling of $f(x)$. For example, a ray-tracing sampling step of a tenth of a voxel is used in [13], a twentieth in [14]. This comes at a high computational cost, while the reward is better faithfulness to the data. Another approach is to select a basis function that doesn't oscillate, such as the quasiinterpolant B-spline used in [7], at the cost of an important loss of fidelity to the data.

2) Minimizing Ringing: Ringing is another name for the mechanism whereby oscillations of the basis function result in data overshoot and undershoot. An intriguing property, in the sinc case, is that its amplitude is not reduced by oversampling, as large as this oversampling may be (Gibb's effect). It is, however, the price to pay for better-quality interpolation. Nevertheless, its presence is disconcerting at first; therefore, our second requirement is that its contribution be minimal. We can control the amount of ringing by choosing the order of approximation L. We illustrate in Fig. 1 how this choice effects the overshoot of a step function that is interpolated by splines of several degrees.

3) Efficiency: Our third requirement will be to reduce computational costs as much as possible. A number of approaches exist to reduce the cost of finding an isosurface [18]–[21]. Unfortunately, none is convenient for oscillating high-quality basis functions. For a given oversampling factor, the computational cost ultimately depends on the support of the basis function itself. Therefore, we seek to design a basis function with minimal support.

4) Aesthetics: As fourth and last requirement, we ask that the basis function be well-suited for rendering by ray tracing. We observe that this technique requires the knowledge of a normal to the isosurface. If we model locally the continuous model $f(\mathbf{x})$ as the truncated Taylor series $f(\mathbf{x}+\Delta\mathbf{x}) = f(\mathbf{x}) + \langle\Delta\mathbf{x}, \boldsymbol{\nabla} f(\mathbf{x})\rangle$, then the isosurface condition $f(\mathbf{x}+\Delta\mathbf{x}) = f(\mathbf{x}) = f_0$ implies that $\langle\Delta\mathbf{x}, \boldsymbol{\nabla} f(\mathbf{x})\rangle = 0$, which expresses that the normal is indeed parallel to the gradient of the data since $\boldsymbol{\nabla} f(\mathbf{x})$ is perpendicular to every vector $\Delta\mathbf{x}$ that belongs to the isosurface. To obtain high-quality renderings, especially when zooming close to the isosurface, it is thus very desirable to have a sufficiently differentiable model to get well-defined gradients for the illumination model.

B. Outcome

In summary, our four requirements are

- the order of approximation L of φ_{int} must be large;
- ringing must be minimized;
- the support W of φ must be short;
- the function φ must be continuously differentiable.

Only one family of functions maximizes L while minimizing W; its members are called Maximal Order and Minimal Support basis functions, or MOMS [22], [23]. Within this family, the shortest-support function that is continuously differentiable is unique; it is the quadratic spline β^2. This function is different from the Dodgson's interpolator [24] used in [8], which has a lower order of approximation [25]. Contrary to Dodgson's interpolator, we observe that the quadratic spline does not satisfy the traditional interpolation condition; in other words, we have that $\beta^2(k) \neq \delta_k = \varphi_{\mathrm{int}}(k)$. We will see in Section II-C that this apparent difficulty can be bypassed by introducing the interpolating spline η^n of degree n, which has the same order $L = n+1$ as a B-spline of degree n, the same differentiability, but an infinite support. Yet, the infinite sum $f(\mathbf{x}) = \sum_{\mathbf{k}\in\mathbb{Z}^p} f(\mathbf{k})\eta^n(\mathbf{x} - \mathbf{k})$ is still realizable *exactly* in a finite (small) time thanks to some tricks [25]. Since η^n has the order of approximation $L = n+1$ and is interpolating, this basis function is a very good approximation of sinc; therefore, the resulting continuous model exhibits ringing, but it happens that the amplitude of the oscillations for $n = 2$ is least among all interpolating splines that are continuously differentiable, as illustrated in Fig. 1.

In this paper, we take advantage of numerous properties of splines to derive a scheme where we are able to tell apart those voxels that enclose the isosurface from those that do not, effectively bypassing the issue of ringing. In the context of ray tracing, this results in a large speed-up since it is necessary to evaluate the model $f(\mathbf{x})$ only in the vicinity of the isosurface, which avoids the need for the fruitless oversampling away from the isosurface that other high-quality methods such as [5]–[8] have to endure.

C. Data

Let β^n be the centered B-spline of degree n. Some of its many interesting properties are 1) Non-negativity $0 \leq \beta^n(x)$; 2) finite support of width $(n+1)$; 3) $(n+1)$-th order of approximation; 4) $(n-1)$-times continuous differentiability; and 5) multiresolution space embedding. Also of relevance to this paper is a computationally efficient recursion that allows one to compute simultaneously an array of samples of the B-spline and of its derivative. This recursion is exposed in Appendix B.

The interpolating spline η^n shares the properties 3–5 of β^n. It has an infinite support and closely approximates the sinc function, much better than any apodized sinc [26], [27]. It is defined by

$$\eta^n(x) = \sum_{k_0\in\mathbb{Z}} (b^n)^{-1}(k_0)\beta^n(x-k_0)$$

where the discrete sequence $(b^n)^{-1}(k)$ has the z-transform

$$(b^n)^{-1}(k) \stackrel{z}{\longleftrightarrow} \frac{1}{\beta^n(0) + \sum_{k=1}^{\lfloor\frac{n}{2}\rfloor} \beta^n(k)(z^k + z^{-k})}.$$

We can thus write

$$\begin{aligned} f(x) &= \sum_{k_1\in\mathbb{Z}} f(k_1)\eta^n(x-k_1) \\ &= \sum_{k_1\in\mathbb{Z}} f(k_1) \sum_{k_0\in\mathbb{Z}} (b^n)^{-1}(k_0)\beta^n(x-k_0-k_1) \\ &= \sum_{k\in\mathbb{Z}} \underbrace{\left(\sum_{k_1\in\mathbb{Z}} f(k_1)(b^n)^{-1}(k-k_1)\right)}_{c(k)} \beta^n(x-k) \\ &= \sum_{k\in\mathbb{Z}} c(k)\beta^n(x-k) \end{aligned}$$

where we have just converted the infinite support of the basis function η^n into the finite support of β^n, without approximation, but at the cost of the determination of coefficients c. As seen in the derivation, these coefficients can be obtained by digital filtering; an efficient procedure is described in [28].

D. Gradient

Differentiating the model above is done as follows:

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x}=\sum_{k\in\mathbb{Z}}c(k)\frac{\mathrm{d}\beta^n(x-k)}{\mathrm{d}x}.$$

Finally, to obtain the continuous model for volumetric instead of one-dimensional data, it is enough to consider the Cartesian tensor product of splines.

III. Ray Tracing

A. Geometry

Ray tracing is a rendering method that has evolved over the years into many variants. In its basic form, rays go through a virtual pinhole camera consisting of an eye $\mathbf{E}$ and of a projection plane. On one side of the eye, the rays intersect the image under construction (the projection plane plays the role of the retina); one pixel is painted per ray. On the other side of the eye, a given ray will propagate until it reaches an obstacle, say, the isosurface or the bounding box of the volumetric data to render. The illumination model will then determine which color has to be assigned to the pixel belonging to the ray. In addition, it is sometimes desirable to introduce a clipping plane that is parallel to the projection plane. This clipping plane determines two half spaces; any data that lies in the half space on the same side as the eye is ignored, which is particularly handful for close-up situations.

Let $\mathbf{P}$ be the (3-D) coordinate of a pixel on the projection plane. Each coordinate $\mathbf{x}$ of a ray that goes through $\mathbf{P}$ satisfies $\mathbf{x}=\mathbf{E}+t(\mathbf{P}-\mathbf{E})$, where t is some rectilinear parameter. To determine the intersection of the ray with the isosurface, one must essentially solve for the value t_0 such that

$$f_0=f(\mathbf{x})=f(t_0)=f(\mathbf{E}+t_0\,(\mathbf{P}-\mathbf{E}))\,.$$

Without loss of generality, from now on we consider that $f_0=0$. Finding the whole isosurface is thus equivalent to finding the set of $\mathbf{x}$ such that

$$0=\sum_{\mathbf{k}\in\mathbb{Z}^3}f(\mathbf{k})\eta^n(\mathbf{x}-\mathbf{k})=\sum_{\mathbf{k}\in\mathbb{Z}^3}c(\mathbf{k})\beta^n(\mathbf{x}-\mathbf{k})$$

where the spline coefficients c are obtained from the data samples by recursive filtering.

B. Bystepping Ringing

Besides enjoying a basis function with finite support, a very interesting property of the rightmost term above is the nonnegativity of B-splines. To see why, consider some fixed $\mathbf{x}_0$: the finite support of the tensor-product B-spline $\beta(\mathbf{x})$ will determine the finite set $\mathbb{C}$ of those coefficients c that contribute to the sum for $\mathbf{x}_0$; meanwhile, the same set $\mathbb{C}$ has also to be used for every $\mathbf{x}$ in some continuous unit cube $\mathbb{X}$ around $\mathbf{x}_0$. For odd degrees, the center of this cube is $\mathbf{k}=\lfloor\mathbf{x}_0\rfloor+(1/2)\ \mathbf{1}$, while for even degrees it is $\mathbf{k}=[\mathbf{x}_0]$. Now, whenever all members of $\mathbb{C}$ happen to have a uniform sign, it can be ascertained that the unit cube contains no part of the isosurface because $0\leq\beta^n(\mathbf{x})$. We illustrate in Fig. 2 the fact that no more than 3 coefficients do contribute when evaluating the quadratic spline model of one-dimensional data. Volumes require $3^3=27$ coefficients.

This suggests the following preprocessing steps for image rendering: first, subtract the threshold f_0 from the data; then, determine the coefficients c out of these reduced data; produce an array of binary values B where each element $B(\mathbf{k})$ is set to true when $0<c(\mathbf{k})$, to false otherwise; finally, update this array by computing its morphological gradient with a structural element C that is a centered cuboid of odd size $(1+2\lfloor(n+1)/2\rfloor)$. The goal of this binary morphological operation is to flag every voxel $\mathbf{k}$ where a sufficiently close neighbor of the coefficient $c(\mathbf{k})$ differs by sign. It is obtained as $b=(B\oplus C)-(B\ominus C)$, where the operator $\oplus$ represents binary dilation, and where $\ominus$ represents binary erosion by the structuring element C. Since none of the preprocessing steps depends on the viewing orientation, the volumetric data can be stored directly as $\{b,c\}$. We note, however, that a modification of the threshold from f_0 to f_1 requires that b be recomputed from scratch, while c can be updated by simple addition of (f_0-f_1). We illustrate the sequence of operations in Fig. 3.

C. Walking the Ray

The rendering itself can now proceed. While exploring a ray for finding the isosurface, it is enough to hop[1] from voxel to voxel, based on the content of the binary array b. It is only when an element $b(\mathbf{k})$ indicates a potential zero-crossing that it is necessary to actually perform the costly operation of evaluating f for some noninteger $\mathbf{x}$. This allows for easy operations such as cutouts (realized by binary operations on b), and such as the representation of b by an octree [18] or by other efficient structures to represent binary data [29], which could potentially lead to a dramatic acceleration of the rendering by transforming hops into strides. Distance-based acceleration is another means of achieving the same goal [30].

IV. Voxel Pruning

The series of preprocessing steps proposed above already results in a significant reduction of the computational burden with respect to blind ray stepping. Nevertheless, yet another property of splines allows us to reduce it further: multiresolution space embedding. For odd degrees, it is possible to represent any spline $f(x)=\sum_{k\in\mathbb{Z}}c_1(k)\beta^{n_1}(x-k)$ with a basis function consisting of a B-spline of same degree but with a support scaled-down by a factor 2 (other factors are possible [31]). The same is true for even degrees, up to an additional shift. We have that

$$\begin{aligned}f(x)&=\sum_{k\in\mathbb{Z}}c_1(k)\beta^n(x-k)\\&=\sum_{k\in\mathbb{Z}}c_2(k)\beta^n\left(2x-k-\frac{(n+1)\mathrm{mod}2}{2}\right),\end{aligned}$$

where

$$c_2(k)=\sum_{l\in\mathbb{Z}}c_1(l)u_2^n(2l-k)$$

[1] We observe experimentally that the average jump length is about 0.653; we leave to the reader the task to find the exact value by an argument akin to Buffon's needle.

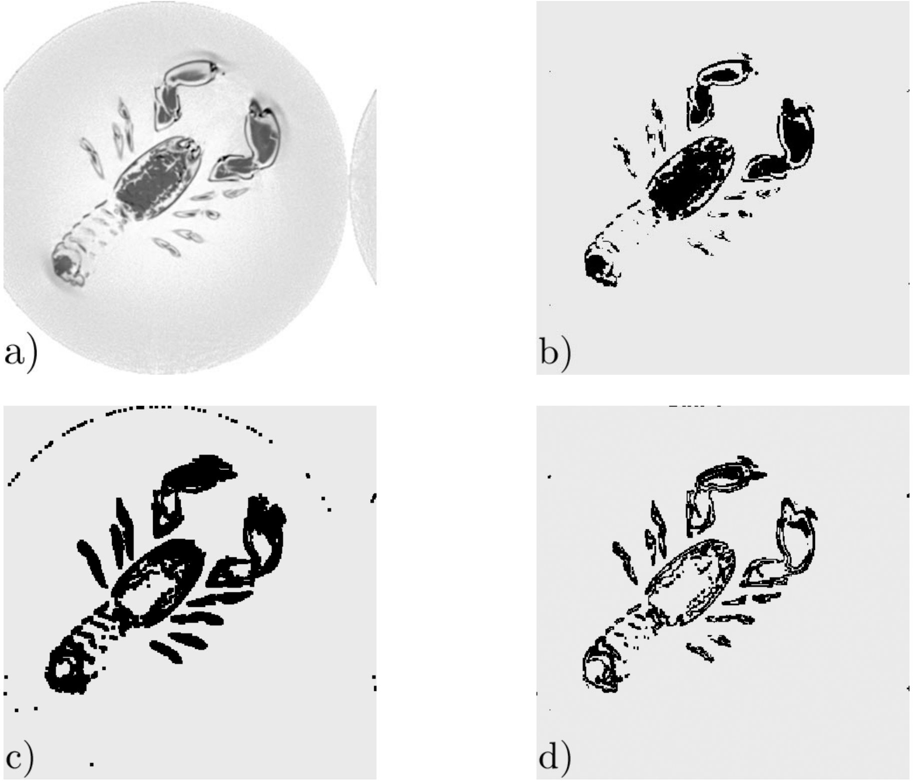


Fig. 3. (a) Slice of a volume. (b) Thresholded slice. Values that were above the threshold are shown in black. (c) Slice after application of a 3-D morphological gradient to the thresholded volume. The unflagged (gray) voxels indicate with certitude the absence of the isosurface. The flagged (black) voxels indicate the potential location of the isosurface. (d) Additional pruning is possible and is explained in Section IV. Here, the recursion depth is $d = 4$.

and

$$u_2^n(k) = \begin{cases} 2^{-n}\binom{n+1}{k+\lfloor\frac{n+1}{2}\rfloor} & k \in \left[-\lfloor\frac{n+1}{2}\rfloor, \lfloor\frac{n}{2}\rfloor + 1\right] \\ 0 & k \notin \left[-\lfloor\frac{n+1}{2}\rfloor, \lfloor\frac{n}{2}\rfloor + 1\right] \end{cases}.$$

The argument applies recursively to yield ever-finer coefficients $c_{1,2,4,8,\ldots}$. We take advantage of this alternate representation of $f(x)$ by remarking that the coefficients $c_{2^d}(k)$, $d \in \mathbb{N}$ are all multiplied by a nonnegative basis function. Thus, it is easy to detect those cases where the signs of $c_1 \in \mathbb{C}$ differ, but where in reality no part of the isosurface belongs to $\mathbb{X}$: when this is indeed the case, there will be a sufficient depth 2^d such that all coefficients $c_{2^d}(k)$ have the same sign over the set $\mathbb{X}$—note that $c_\infty(k)$ ultimately converges to some $f(x)$.

Thus, we propose an additional preprocessing step that consists in visiting each relevant voxel in b, and that removes all those cases where it can be determined by the procedure above that no isosurface is contained in the voxel. For a quadratic model $n = 2$, this procedure can as much as halve the number of candidates in b; the reduction is even more pronounced for higher degrees. Ideally, the resulting binary volume b should be a 6-connected, 1 voxel-width shell. In practice, at any depth d it is necessary to produce and examine the sign of many coefficients. As the support of a B-spline of degree n is $(n+1)$, and as we deal with three dimensions, the number of coefficients grows like $\mathcal{O}(n^3)$. At the same time, each finer level results in a doubling of the number of coefficients along each dimension. Finally, at any depth d it is necessary to produce and examine the sign of $\mathcal{O}(2^{3d}n^3)$ coefficients, which grows very quickly out of manageable size, even considering that this preprocessing can be performed off-line. Nevertheless, we will see in Section VI that a shallow recursion depth is enough to bring significant benefits. We have applied four recursions to get the bottom-right part of Fig. 3.

V. Root Finding

The set $\mathbb{X}$ of coordinates $\mathbf{x}$ that share a set $\mathbb{C}$ of coefficients $c(\mathbf{k})$ is centered on a voxel for even degrees, and on the corner of a voxel for odd degrees. To simplify the discussion, we will consider only even degrees from now on.

To find a potential solution to the equation $0 = f(\mathbf{E}+t(\mathbf{P}-\mathbf{E}))$ within a voxel indicated by $b(\mathbf{k})$ as potentially containing a part of the isosurface, we first identify the two rectilinear parameters t_1 and t_2 that correspond to the locations where the ray enters, respectively exits, the voxel $\mathbf{k}$. We then compute f_1 and f_2, which should bracket the isosurface, a condition revealed by $f_1 f_2 \leq 0$. If such is the case, we proceed with a standard procedure for root finding such as Brent's [32] to get t_0 such that $f(t_0) = 0$. But more than often, this bracketing fails, either because 1) there is really no isosurface within the voxel (rare after the pruning steps of Section IV), or because 2) the ray grazes the isosurface without piercing it (grazing miss), or because 3) the ray pops in and out of the isosurface within the voxel at least twice.

A. Grazing Hits

The third case is of interest; we call it a grazing hit. To identify it, we propose the following heuristic: we compute the rectilinear derivatives $g_1 = (\mathrm{d}f(t_1))/\mathrm{d}t$ and $g_2 = (\mathrm{d}f(t_2))/\mathrm{d}t$, which requires the knowledge of the spatial gradients f_x, f_y, and f_z. Those come at essentially no computational cost thanks to the procedure given in Appendix B. Then, out of all the possible combinations of signs for sign(f_1) = sign(f_2), sign(g_1), and sign(g_2), we retain only two cases for which we investigate further the possibility of an intersection of the ray with the isosurface. These two cases are ($f_1 < 0$, $0 < g_1$, $g_2 < 0$) and ($0 < f_1$, $g_1 < 0$, $0 < g_2$). If one of these two conditions is satisfied, we proceed by stepping the ray between t_1 and t_2, eventually finding (or not) the isosurface; else, we skip the voxel and resume with the exploration of b.

The cases where the ray barely grazes the isosurface are few and far apart. It then may seem at first that their importance is minor. In practice, however, we observe that these cases play a preponderant role, for they determine the outline of the rendered isosurface. This outline is a key perceptual element of the quality of the rendering. Some illustrative examples will be presented in Section VI.

VI. Experiments

A. Synthetic Volumes

We have synthesized a tiny $(5 \times 5 \times 5)$ volume $s(\mathbf{k})$ by sampling at integer values $\{k\} \in [0 \ldots 4]$ the continuous Gaussian $s(\mathbf{x}) = \exp[-((x-2)^2 + (y-2)^2 + (z-2)^2)]$. For $f_0 = 1/2$, the true isosurface of s is a centered sphere of radius 0.83 voxel; the volume enclosed by the ideal sphere is therefore only 2.42 cubic voxel. Meanwhile, in the sampled volume, there is a single voxel above the threshold. Typically, rendering techniques that rely on polygon faces would represent this single voxel as an octahedron. For example, Fig. 4 has been built by applying the method [33], which is available from IDL [34]. By contrast, the method proposed in this paper results in a much smoother sphere around this single voxel.

The morphological gradient operation expands this voxel to a $(3 \times 3 \times 3)$ cube of 27 elements. The optional pruning step removes the corners of the cube after five iterations, which leaves 19 potential candidates in b. We show the result of the rendering in the left part of Fig. 5, where it can be seen that the isosurface determined from the sampled volume $s(\mathbf{k})$ gives a good approximation of the true isosurface of the continuous volume $s(\mathbf{x})$, despite the very small amount of available data. This good result can be obtained only if we activate the mechanism of Section V-A that recovers the grazing hits. In the present case, ignoring them leads to the right part of Fig. 5, where 20 760 erroneously unpainted pixels can be found along the silhouette of the sphere. Those unpainted pixels are located between the outline of the left part of the figure (shown on the right as a black thin line), and the remaining gray values.

To check that the sphere above is not just a blob in disguise, we need to make sure that we can build different shapes within the same tiny volume. We have thus synthesized another volume by sampling the continuous function $s(\mathbf{x}) = r^2 - \left(R - \sqrt{(x-x_0)^2 + (y-y_0)^2}\right)^2 - (z-z_0)^2$, again on a $(5 \times 5 \times 5)$ array, and with $x_0 = y_0 = z_0 = 2$. In this case, the true isosurface of the continuous s for $f_0 = 0$ is a torus with minor radius $r = 1/3$ and major radius $R = \sqrt{2}$. The sampled volume has 8 values above the threshold that form a ring in the central slice; the 75 candidates resulting from the morphological gradient fill the three (5 × 5) central slices. Five pruning steps result in the following number of candidates: $75 \rightarrow 75 \rightarrow 72 \rightarrow 72 \rightarrow 64 \rightarrow 56$. We show in Fig. 6 how the location of the torus is refined by the pruning process, which allows us to insist on the small size of the volume we are considering. This makes the problem of reproducing the continuous $s(\mathbf{x})$ out of the discrete $s(\mathbf{k})$ much more difficult and interesting. High-quality isosurface extraction is also a prerequisite to obtain satisfactory renderings of such a small volume. We give in Fig. 7 the result of our approach when applied to the torus in a $(5 \times 5 \times 5)$ array.

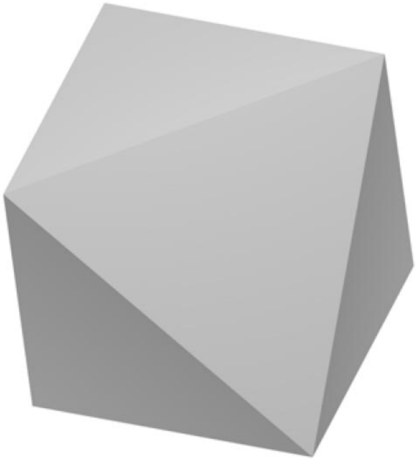


Fig. 4. Tiny sphere, as rendered by polygonal faces.

We indicate in Fig. 8 where the integer lattice intersects the torus. The vertices of most polygon-based methods necessarily belong to this intersection. For example, applying the method [33], which is similar to [1], we obtain Fig. 9. We observe that the polygons are very coarse, which is due to the small size of the volume considered. The method proposed in this paper is not sensitive to this aspect. In addition, a careful analysis of the polygons reveals that they do not satisfy expected relations of symmetry, which is due to ambiguities of the conversion between volumetric and polygonal representation [2]. The method proposed in this paper does not suffer from this limitation.

Finally, we have produced Fig. 10 by slightly modifying the parameters of the torus, with $r = 1/2$ and $R = 1$. To achieve disparity without changing the location of the eye $\mathbf{E}$, we have imposed a slight relative shift to the torus between the left and right image; on the left, $x_0 = 1.75$, and $x_0 = 2.25$ on the right. Within a $(5 \times 5 \times 5)$ volume, it would be extremely difficult to obtain so fine a control with traditional techniques that rely on the initial conversion of the volume $s(\mathbf{k})$ into a list of polygon faces.

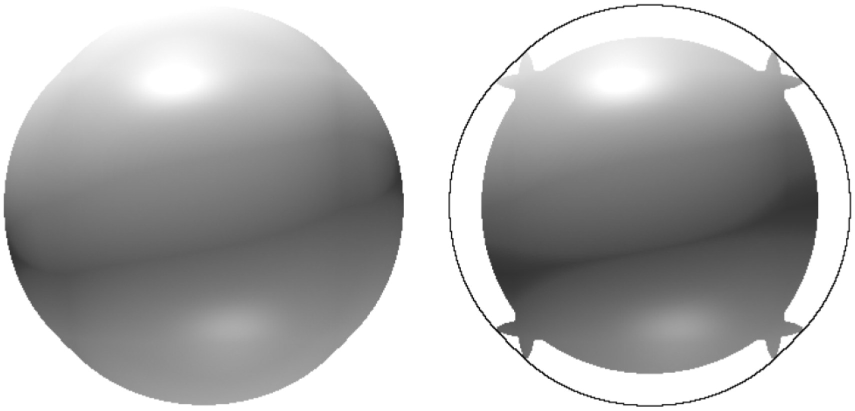


Fig. 5. Left: Phong rendering of a sphere inside a $(5 \times 5 \times 5)$ volume, with two positional light sources. Right: The grazing hits cannot be disregarded.

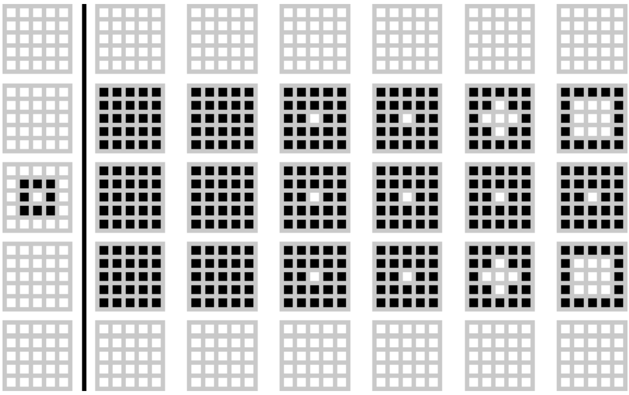


Fig. 6. Evolution of b through the preprocessing steps. Each column depicts the whole volume as a series of slices. The first column corresponds to the thresholding of $s(\mathbf{k})$, the second to the morphological gradient. The remaining columns give the result of the pruning steps that confine more and more the potential location of the isosurface.

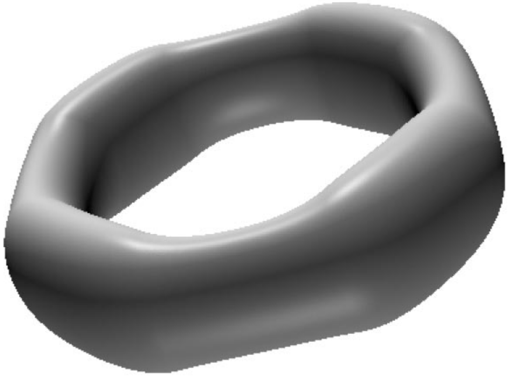


Fig. 7. Phong rendering of a torus inside a $(5 \times 5 \times 5)$ volume, with two positional light sources.

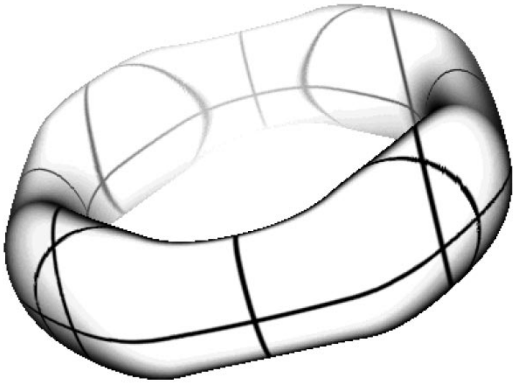


Fig. 8. Location of integer-level (x,y,z)-curves on a torus inside a $(5 \times 5 \times 5)$ volume.

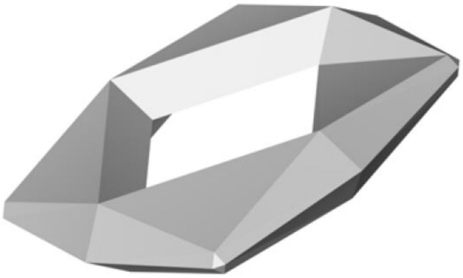


Fig. 9. Methods that rely on polygons have difficulties to obtain a smooth rendering of the torus of Fig. 7.

B. Real Volumes

We show in Fig. 11 our Phong rendering of the close-up of a $320 \times 320 \times 34$ lobster volume.[2] The size of this image is 640×360, without antialiasing. The isosurface threshold is $f_0 = 50$. In Fig. 12, we propose similar close-up renderings for a $128 \times 128 \times 150$ human spine.[3] In this case, the threshold has been set to $f_0 = 100$. The aspect ratio of a voxel is cubic neither for the lobster nor for the spine, which has to be taken into account when computing the normal to the isosurface and when performing rotations of the volume. Finally, we show in Fig. 13 the rendering of a 3-D reconstruction of a $255 \times 255 \times 256$ papilloma virus,[4] where a cutout has been realized on b to reveal its interior.

[2] Lobster dataset courtesy of Advanced Visual Systems, via Mark Kessler, University of Michigan Medical School.

[3] Orthogonal axial spiral CT dataset of part of a human spine. Dataset courtesy of Ramani Pichumani, Stanford University School of Medicine.

[4] Papilloma virus dataset courtesy of Benes L. Trus, National Institutes of Health [35].

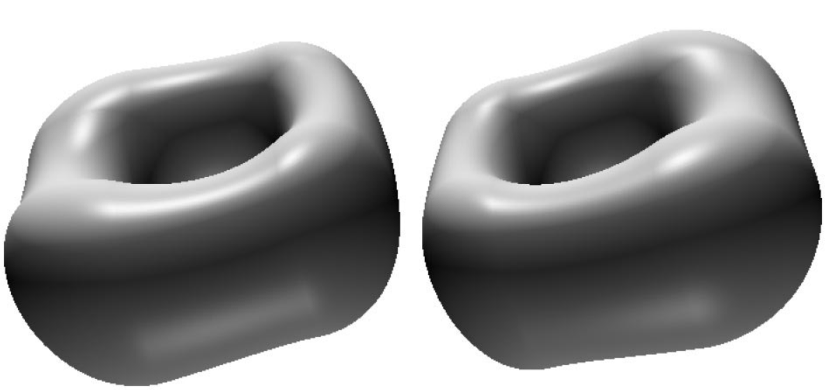


Fig. 10. Torus inside a $(5 \times 5 \times 5)$ volume with a larger minor radius r and a smaller major radius R, and a disparity between the left and right image obtained by displacing the torus. The other rendering parameters are the same as in Fig. 7.

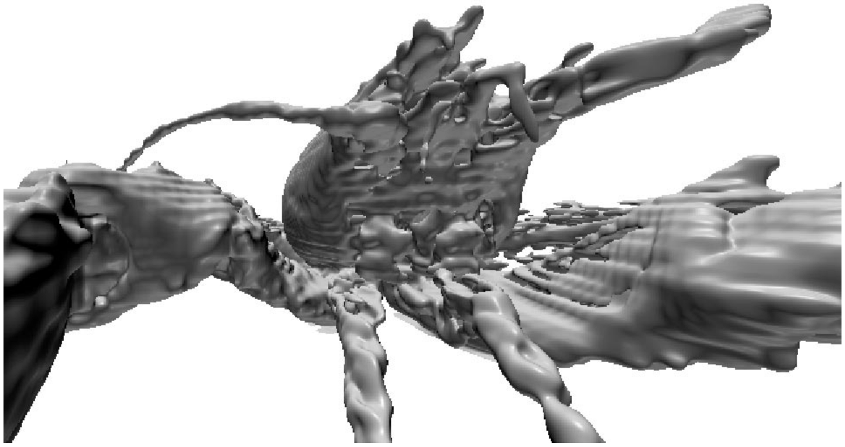


Fig. 11. Phong rendering of a lobster with two positional light sources.

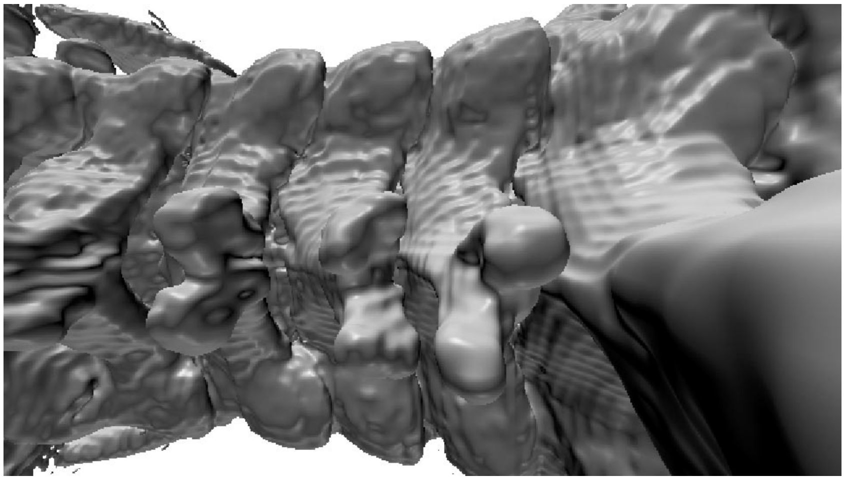


Fig. 12. Phong rendering of a human spine with two positional light sources.

C. Performance Analysis

We now examine the practical benefits of the approach presented in this paper and analyze the performance of our algorithms for the lobster image. We show in Table I that the lobster volume contains 270 160 voxels indexed by $\mathbf{k}_1$ that are close enough to the isosurface that there exists at least one voxel indexed by $\mathbf{k}_2$ within the volume of influence $\mathbb{C}(\mathbf{k}_1)$ such that $\text{sign}(c(\mathbf{k}_1)) \neq \text{sign}(c(\mathbf{k}_2))$. This amounts to only 8% of the 3 481 600 voxels that constitute the lobster data; without the nonnegativity of B-splines, ringing would force us to perform ray stepping inside each and every voxel to ascertain the absence of an isosurface, whereas our scheme allows us to reject 92% right away. Moreover, after four of the pruning recursions described in Section IV, we were able to reject 43.8% of the remaining candidates, a nearly optimal score—in the case of a quadratic spline, we could hope for 50% at best. These savings are orientation-independent. In the case of the specific orienta-

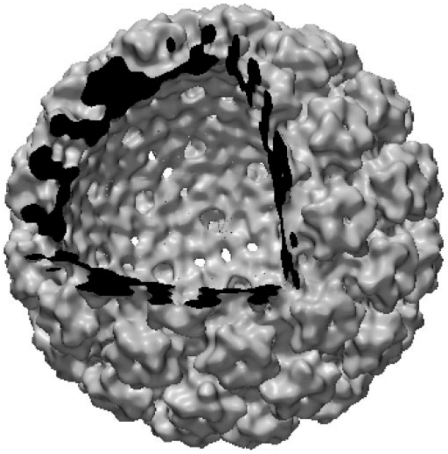


Fig. 13. Rendering of a papilloma virus with a cutout realized as a binary operation on b.

TABLE I
BENEFIT OF PRUNING. A) PRUNING TIME IN [S]. B) RENDERING TIME IN [S]. C) RELATIVE TIME SAVINGS

	card(b)		a)	b)	c)
Initial	270,160	100%	0	35.7	0%
1 level	212,249	77%	25.0	25.2	29%
2 levels	178,178	66%	42.0	19.7	45%
3 levels	160,651	59%	100.2	17.4	51%
4 levels	151,660	56%	384.9	16.3	54%

TABLE II
TALLY OF THE WORKLOAD

1) Rays	230,400	
2) Clipped ray segments	23,736,114	
3) Explored ray segments	20,754,910	100%
4) Ray segments rejected by b	20,217,938	97.4%
5) Candidates from b	536,972	100%
6) Direct hit ($f_1 f_2 < 0$)	103,240	19.2%
7) Rejected by gradients (g_1, g_2)	371,290	69.1%
8) Grazing misses	59,409	11.1%
9) Grazing hits	3,033	0.6%

tion that corresponds to Fig. 11, we show in the last column of Table I how these savings in card (b) translate into savings of the rendering time, as performed on a 450 MHz PowerPC G4 Macintosh. We note that the pruning time can be disregarded because this preprocessing step can be performed off-line and is view-independent.

The vantage point we have selected is in front of the head of the lobster, midway between its pincers. Since this is inside the volume, it is necessary to explicitly ignore any section of ray that would extend behind the eye $\mathbf{E}$, while keeping those that lay in front. A clipping plane is therefore desirable. We show in the second line of Table II to how many ray segments this condition has been applied, where we count as one ray segment that part of the ray that extends inside a single voxel. We give in the third line the number of ray segments that have been explored on the interesting side on the clipping plane. We see in the fourth line of Table II that a large fraction of those segments did not lead to an isosurface computation because they could be rejected by considerations on b alone. This fraction is larger than $(1 - (151\,660/3\,481\,600)) = 95.6\%$ (the score predicted from Table I), because the latter is view-independent while the result of Table II is view-dependent.

The number of nontrivial potential intersections between a ray segment and the isosurface is given in the fifth line of Table II. Of those, 19.2% led to the straightforward determination of the real existence of an intersection since the ray crosses the isosurface in an unambiguous fashion. As can be seen in the seventh line of Table II, the cases rejected by the heuristic proposed in Section V-A are numerous since they account for more than two thirds of the workload, bringing substantial additional savings in computation time. Unfortunately, while the savings of Section IV are secure, those of Section V-A are less conservative because the function $f(t)$, $t_1 \leq t \leq t_2$ is a polynomial of degree 6 for a quadratic model in 3-D, while our heuristic can only determine with certitude which polynomials of degree at most 2 have no root within $[t_1, t_2]$. Thus, it is interesting to further investigate whether the rejection cases are legitimate. Since there exists no closed form for expressing by radicals the root of a sixth degree polynomial, we have resolved the issue experimentally by stepping the ray for every voxel. We conclude that only 12 cases out of 371 290 were wrongly rejected, which we consider negligible. Meanwhile, our heuristic accounts for 86.2% of all rejection cases.

The eighth line of Table II contains the number of voxels that were considered worth of exploration, but that nonetheless led to no discovery of an intersection with the isosurface. This exploration is conducted by recursively dividing the interval $[t_1, t_2]$ in smaller steps until two samples of $f(t)$ exhibit a differing sign, taking care never to perform two estimations that would have the same rectilinear argument t. We abandon this bracketing search when the step becomes smaller than $(1/32)(t_2 - t_1)$, which, on average, represents about 50 samples per voxel. Whenever the root can be bracketed, we proceed with the standard procedure for root finding.

Finally, the last line of Table II shows how many times we were able to recover a near miss, or, equivalently, a grazing hit, which happens whenever a ray crosses the isosurface within a single voxel at least twice. This represents 2.85% of the 106 273 painted pixels, which one could perhaps conclude is insignificant. To convince the reader of the contrary, we produce Fig. 14, on the left part of which we have forgone ray stepping. That we do not notice the plunge into the isosurface doesn't necessary mean that there is not another intersection farther down the ray; this explains why we still paint 939 pixels out of the 3,033 that one would expect to see vanish. Nevertheless, it is very apparent that the few missing pixels play a significant perceptual role: for example, one of the frontal left appendages of the lobster is nearly dissolved when the grazing hits are ignored. Also, its right supraorbital spine becomes more angular than necessary.

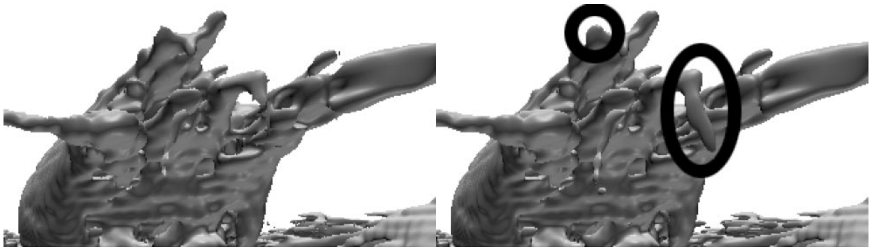


Fig. 14. Effect of the grazing hits. Left: ignoring them; right: taking them into account.

VII. Conclusion

We have singled out the quadratic B-spline as the one and only basis function that satisfies the requirement of smallest support for a continuously differentiable function of maximal order of approximation. The high order of approximation involves high-quality renderings, as predicted by classical results in the theory of approximation. The drawback is data overshoot and undershoot, which complicates the determination of an isosurface.

To solve this difficulty, we have proposed three complementary ways to accelerate the rendering of an isosurface by ray tracing in the context of high-quality spline interpolation. At first, by expressing the spline model as a sum of nonnegative basis functions, we are able to bypass the difficulties associated with their oscillations, while maintaining high fidelity to the data. This results in a binary volume that indicates every potential location of the isosurface. As second step, we have used the space embedding property of splines to prune candidates in this binary volume. The gain can be as large as the spline degree, a factor two in our case. As third and last step, we have proposed a heuristic based on the computation of exact gradients that tends to avoid the unnecessary exploration of many voxels. While the two first acceleration methods guarantee that no part of the isosurface is left aside, the proposed heuristic is less conservative. Nevertheless, experiments show that the true rejection rate is very beneficial (86.2% savings), while the error rate is insignificant (0.0032% false rejection, no false accept).

Appendix

VIII. Oscillating Basis Functions

All interpolating basis functions of sufficient order $L > 2$ do oscillate, where we understand the interpolation condition as being $\varphi_{\mathrm{int}}(k) = \delta_k$. We explain here why this is so. A well-known result of the theory of approximation is the equivalence between, on one hand, the condition on the reproduction of polynomials up to degree $(L-1)$ mentioned in Section I and, on the other hand, the fact that the discrete moments computed as $\mu_n = \sum_{k\in\mathbb{Z}}(x-k)^n\varphi_{\mathrm{int}}(x-k)$, $\forall n \in [0\ldots L-1]$, depend on integer n only. In particular, we can choose the integer $n=2$ and the integer $x=k_0$ to compute $\mu_2 = \sum_{k\in\mathbb{Z}}(k_0-k)^2\delta_{k_0-k} = 0$. But μ_2 cannot vanish for a general x because it is the sum of the product of two terms, both of which are positive, unless the interpolating basis function φ_{int} oscillates. Therefore, the order of approximation of nonoscillating basis functions that satisfy the interpolation condition cannot exceed $L=2$.

IX. B-Spline Computation

We propose here an efficient recursive scheme to compute simultaneously an array of B-spline values and of their derivatives for $(n+1)$ arguments spaced one unit apart:

$$\begin{cases} \beta^n(x) = & \frac{1}{n}\left(\left(x+\frac{n+1}{2}\right)\frac{\mathrm{d}\beta^n(x)}{\mathrm{d}x}+(n+1)\beta^{n-1}\left(x-\frac{1}{2}\right)\right) \\ \frac{\mathrm{d}\beta^n(x)}{\mathrm{d}x} = & \beta^{n-1}\left(x+\frac{1}{2}\right)-\beta^{n-1}\left(x-\frac{1}{2}\right) \end{cases}.$$

For better efficiency, we complement the scheme above by making use of the property known as the partition of unity, which is expressed as $1 = \mu_0 = \sum_{k\in\mathbb{Z}}\varphi(x-k)$, $\forall x \in \mathbb{R}$. This results in extra savings. For example, letting $n=2$ results in only 10 operations to get $\{\beta^2(x-1), \beta^2(x), \beta^2(x+1), (\mathrm{d}\beta^2(x-1)/\mathrm{d}x), (\mathrm{d}\beta^2(x)/\mathrm{d}x), (\mathrm{d}\beta^2(x+1)/\mathrm{d}x)\}$ under the hypothesis that $-1/2 < x < 1/2$, as follows:

$$\begin{cases} \beta^1\left(x-\frac{1}{2}\right) = x+\frac{1}{2} \\ \beta^1\left(x+\frac{1}{2}\right) = 1-\beta^1\left(x-\frac{1}{2}\right) \end{cases}$$

$$\begin{cases} \dot\beta^2(x-1) = \beta^1\left(x-\frac{1}{2}\right) \\ \dot\beta^2(x) = \beta^1\left(x+\frac{1}{2}\right)-\beta^1\left(x-\frac{1}{2}\right) \\ \dot\beta^2(x+1) = -\beta^1\left(x+\frac{1}{2}\right) \end{cases}$$

$$\begin{cases} \beta^2(x-1) = \frac{1}{2}\left(\left(x+\frac{1}{2}\right)\dot\beta^2(x-1)+3\beta^1\left(x-\frac{3}{2}\right)\right) \\ \qquad = \frac{1}{2}\beta^1\left(x-\frac{1}{2}\right)\dot\beta^2(x-1) \\ \beta^2(x) = \frac{1}{2}\left(\left(x+\frac{3}{2}\right)\dot\beta^2(x)+3\beta^1\left(x-\frac{1}{2}\right)\right) \\ \qquad = \frac{1}{2}+\beta^1\left(x+\frac{1}{2}\right)\beta^1\left(x-\frac{1}{2}\right) \\ \beta^2(x+1) = 1-\beta^2(x)-\beta^2(x-1) \end{cases}.$$

A. Proof

We now give a proof of the validity of the recursion of Appendix B. To save space, we proceed in Fourier; we leave the pointwise proof as an exercise to the reader. Our convention and notation for a Fourier transform is

$$\mathcal{F}\left\{f(x)\right\}_x(\omega) = \int_{-\infty}^{\infty} f(x)e^{-j\omega x}\mathrm{d}x = \hat f(\omega).$$

By definition, the Fourier transform of a B-spline of degree n is

$$\mathcal{F}\left\{\beta^n(x)\right\}_x(\omega) = \left(\frac{\sin\frac{\omega}{2}}{\frac{\omega}{2}}\right)^{n+1} = \mathrm{sinc}^{n+1}\frac{\omega}{2\pi}. \tag{1}$$

Here are three properties that are useful for our purpose:

$$\mathcal{F}\left\{\frac{\mathrm{d}f(x)}{\mathrm{d}x}\right\}_x(\omega) = j\omega\hat{f}(w); \tag{2}$$

$$\mathcal{F}\left\{f(x-t)\right\}_x(\omega) = e^{-j\omega t}\hat{f}(w); \tag{3}$$

$$\mathcal{F}\left\{x\frac{\mathrm{d}f(x)}{\mathrm{d}x}\right\}_x(\omega) = -\hat{f}(w) - \omega\frac{\mathrm{d}\hat{f}(\omega)}{\mathrm{d}\omega}. \tag{4}$$

The first derivative of the $(n+1)$-th power of a sinc function will also come in handy

$$\frac{\mathrm{d}}{\mathrm{d}\omega}\mathrm{sinc}^{n+1}\frac{\omega}{2\pi} = \frac{n+1}{\omega}\left(\cos\frac{\omega}{2\pi} - \mathrm{sinc}\frac{\omega}{2\pi}\right)\mathrm{sinc}^n\frac{\omega}{2\pi}. \tag{5}$$

We can finally proceed with the proof. Using (1), (2), and (3), along with Eulers' formula, we write

$$\begin{aligned}\frac{\mathrm{d}\beta^n(x)}{\mathrm{d}x} &= \mathcal{F}^{-1}\left\{\mathcal{F}\left\{\frac{\mathrm{d}\beta^n(t)}{\mathrm{d}t}\right\}_t(\omega)\right\}_\omega(x)\\ &= \mathcal{F}^{-1}\left\{j\omega\mathrm{sinc}^{n+1}\frac{\omega}{2\pi}\right\}_\omega(x)\\ &= \mathcal{F}^{-1}\left\{\left(e^{j\frac{\omega}{2}} - e^{-j\frac{\omega}{2}}\right)\mathrm{sinc}^n\frac{\omega}{2\pi}\right\}_\omega(x)\\ &= \beta^{n-1}\left(x+\frac{1}{2}\right) - \beta^{n-1}\left(x-\frac{1}{2}\right),\end{aligned}$$

which proves the second line of the recursion. After substitution into its first line, we can rewrite it as

$$\beta^n(x) = \frac{n+1}{2n}\left(\beta^{n-1}\left(x+\frac{1}{2}\right) + \beta^{n-1}\left(x-\frac{1}{2}\right)\right) + \frac{1}{n}x\frac{\mathrm{d}\beta^n(x)}{\mathrm{d}x}.$$

Using (4), (5), and some trivial algebra, its Fourier transform becomes

$$\begin{aligned}\mathcal{F}\left\{\beta^n(x)\right\}_x(\omega) &= \frac{n+1}{2n}\left(e^{j\frac{\omega}{2}}\mathrm{sinc}^n\frac{\omega}{2\pi} + e^{-j\frac{\omega}{2}}\mathrm{sinc}^n\frac{\omega}{2\pi}\right)\\ &\quad - \frac{1}{n}\mathrm{sinc}^n\frac{\omega}{2\pi}\left(\mathrm{sinc}\frac{\omega}{2\pi} + \omega\frac{n+1}{\omega}\right.\\ &\quad \left.\times\left(\cos\frac{\omega}{2\pi} - \mathrm{sinc}\frac{\omega}{2\pi}\right)\right)\\ &= \mathrm{sinc}^{n+1}\frac{\omega}{2\pi}.\end{aligned}$$

The proof follows by identification with (1).

B. Partition of Unity

Let us proof that the B-spline β^n satisfies the partition of unity condition. Setting $p = 1$ in the Poisson sum formula $\sum_{k\in\mathbb{Z}} f(x-kp) = (1/p)\sum_{m\in\mathbb{Z}}\hat{f}(2\pi(m/p))\exp[j2\pi(m/p)x]$, we get

$$\mu_0 = \sum_{k\in\mathbb{Z}}\beta^n(x-k) = \sum_{m\in\mathbb{Z}}\mathrm{sinc}^{n+1}(m)e^{j2\pi mx}.$$

Because $\mathrm{sinc}(m) = \delta_m$, the only contributing term of the right-hand-side sum has index $m = 0$; this yields $\mu_0 = 1, \forall x \in \mathbb{R}$.

References

[1] W. E. Lorensen and H. E. Cline, "Marching cubes: a high-resolution 3D surface construction algorithm," *Comput. Graph.*, vol. 21, no. 4, pp. 163–169, July 1987.

[2] J. Wilhelms and A. Van Gelder, "Topological considerations in isosurface generation—extended abstract," *Comput. Graph.*, vol. 24, no. 5, pp. 79–86, Nov. 1990.

[3] P. Shirley and A. Tuchman, "A polygonal approximation to direct scalar volume rendering," *Comput. Graph.*, vol. 24, no. 5, pp. 63–70, Nov. 1990.

[4] T. Poston, H. T. Nguyen, A.-H. Pheng, and T.-W. Tien, ""Skeleton climbing": Fast isosurfaces with fewer triangles," in *Proc. 5th Pacific Conf. Computer Graphics and Applications*, Seoul, Korea, Oct. 13–16, 1997, pp. 117–126.

[5] M. Levoy, "Volume rendering," *IEEE Comput. Graph. Applicat.*, vol. 8, pp. 29–37, May 1988.

[6] Z. Mihajlović, A. Goluban, and D. Kovačić, "The B-spline interpolation in visualization of the three-dimensional objects," in *Proc. 20th Int. Conf. Information Technology and Interfaces (ITI'98)*, Pula, Croatia, June 16–19, 1998, pp. 425–431.

[7] J. C. Carr, A. H. Gee, R. W. Prager, and K. J. Dalton, "Quantitative visualization of surfaces from volumetric data," in *Proc. 6th Int. Conf. Central Europe on Computer Graphics and Visualization (WSCG'98)*. Plzen-Bory, Czech Republic: Univ. West Bohemia Press, Feb. 9–13, 1998, vol. 1, pp. 57–64.

[8] M. K. Bosma, J. Smit, and S. Lobregt, "Iso-surface volume rendering," *Proc. SPIE*, vol. 3335, pp. 10–19, Feb. 22–24, 1998.

[9] I. Carlbom, "Optimal filter design for volume reconstruction and visualization," in *Proc. 1993 IEEE Int. Symp. Visualization (VIS'93)*, San Jose, CA, Oct. 1993, pp. 54–61.

[10] M. E. Goss, "An adjustable gradient filter for volume visualization image enhancement," in *Proc. 1994 Canadian Information Processing Society International Conference on Graphics Interface*, Banff, AB, Canada, May 1994, pp. 67–74.

[11] S. R. Marschner and R. J. Lobb, "An evaluation of reconstruction filters for volume rendering," in *Proc. 1994 IEEE Int. Symp. Visualization (VIS'94)*, R. D. Bergeron and A. E. Kaufman, Eds., Washington, DC, Oct. 1994, pp. 100–107.

[12] M. J. Bentum, B. B. A. Lichtenbelt, and T. Malzbender, "Frequency analysis of gradient estimators in volume rendering," *IEEE Trans. Visual. Comput. Graph.*, vol. 2, pp. 242–254, Sept. 1996.

[13] T. Möller, R. Machiraju, K. Mueller, and R. Yagel, "Evaluation and design of filters using a Taylor series expansion," *IEEE Trans. Visual. Comput. Graph.*, vol. 3, no. 2, pp. 184–199, Apr.–June 1987.

[14] T. Möller, K. Mueller, Y. Kurzion, R. Machiraju, and R. Yagel, "Design of accurate and smooth filters for function and derivative reconstruction," in *Proc. 1998 IEEE Int. Symp. Visualization (VIS'98)*, Research Triangle Park, NC, Oct. 18–23, 1998, pp. 143–151.

[15] G. Strang and G. Fix, "A Fourier analysis of the finite element variational method," in *Constructive Aspect of Functional Analysis*. Rome, Italy: Edizioni Cremonese, 1971, pp. 796–830.

[16] P. Thévenaz, T. Blu, and M. Unser, "Complete parametrization of piecewise-polynomial interpolators according to degree, support, regularity, and order," in *Proc. 2000 IEEE Int. Conf. Image Processing (IGIP'00)*, vol. II, Vancouver, BC, Canada, Sept. 10–13, 2000, pp. 335–338.

[17] U. Tiede, K. H. Hoehne, M. Bomans, A. Pommert, M. Riemer, and G. Wiebecke, "Investigation of medical 3-D-rendering algorithms," *IEEE Comput. Graph. Applicat.*, vol. 10, pp. 41–53, Mar. 1990.

[18] J. Wilhelms and A. Van Gelder, "Octrees for faster isosurface generation—extended abstract," *IEEE Trans. Med. Imag.*, vol. 19, pp. 739–758, July 2000.

[19] Y. Livnat, H.-W. Shen, and C. R. Johnson, "A near optimal isosurface extraction algorithm using the span space," *IEEE Trans. Visual. Comput. Graph.*, vol. 2, pp. 73–84, Mar. 1996.

[20] ——, "Correction—a near optimal lsosurface extraction algorithm using the span space," *IEEE Trans. Visual. Compt. Graph.*, vol. 2, p. 184, June 1996.

[21] Y. Livnat and C. Hansen, "View dependent isosurface extraction," in *Proc. 1998 IEEE Int. Symp. Visualization (VIS'98)*, Research Triangle Park, NC, Oct. 18–23, 1998, pp. 175–180.

[22] T. Blu, P. Thévenaz, and M. Unser, "MOMS: maximal-order interpolation of minimal support," *IEEE Trans. Image Processing*, vol. 10, pp. 1069–1080, July 2001.

[23] ——, "Minimum support interpolators with optimum approximation properties," in *Proc. 1998 IEEE Int. Confe. Image Processing (ICIP'98)*, vol. III, Chicago, IL, Oct. 4–7, 1998, pp. 242–245.

[24] N. A. Dodgson, “Quadratic interpolation for image resampling,” *IEEE Trans. Image Processing*, vol. 6, pp. 1322–1326, Sept. 1997.

[25] P. Thévenaz, T. Blu, and M. Unser, “Interpolation revisited,” *IEEE Trans. Med. Imag.*, vol. 24, pp. 57–62, Nov. 1990.

[26] T. M. Lehmann, C. Gönner, and K. Spitzer, “Survey: Interpolation methods in medical image processing,” *IEEE Trans. Med. Imag.*, vol. 18, pp. 1049–1075, Nov. 1999.

[27] ——, “Addendum: B-spline interpolation in medical image processing,” *IEEE Trans. Med. Imag.*, vol. 20, pp. 660–665, July 2001.

[28] M. Unser, A. Aldroubi, and M. Eden, “B-spline signal processing: Part II—efficient design and applications,” *IEEE Trans. Signal Processing*, vol. 41, pp. 834–848, Feb. 1993.

[29] B. H. Kim, J. Seo, and Y. G. Shin, “Binary volume rendering using slice-based binary shell,” *Vis. Comput.*, vol. 17, no. 4, pp. 243–257, 2001.

[30] M. Sramek and A. Kaufman, “Fast ray-tracing of rectilinear volume data using distance transforms,” *IEEE Trans. Visual. Comput. Graph.*, vol. 6, pp. 236–252, July–Sept. 2000.

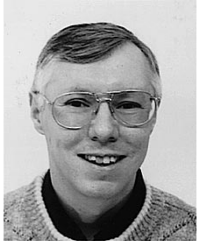
[31] M. Unser, A. Aldroubi, and M. Eden, “B-spline signal processing: part I—theory,” *IEEE Trans. Signal Processing*, vol. 41, pp. 821–833, Feb. 1993.

[32] R. P. Brent, Ed., *Algorithms for Minimization Without Derivatives*. Englewood Cliffs, NJ: Prentice-Hall, 1973.

[33] J. B. Klemp, M. J. McIrvin, and W. S. Boyd, “PolyPaint—a three-dimensional rendering package,” in *Proc. 6th Int. Conf. Interactive Information and Processing Systems*. Anaheim, CA: Amer. Meteorol. Soc., Feb. 5–9, 1990, pp. 286–293.

[34] Research Systems, A Kodak Company, Interactive Data Language, [Online] Available: http://www.rsinc.com/idl/.

[35] B. L. Trus, R. B. S. Roden, H. L. Greenstone, M. J. Vrhel, J. T. Schiller, and F. P. Booy, “Novel structural features on bovine papillomavirus capsid revealed by a three-dimensional reconstruction to 9 AA resolution,” *Nature Structural Biology*, vol. 4, no. 5, pp. 413–420, May 1997.



Philippe Thévenaz was born in 1962 in Lausanne, Switzerland. He graduated from the Lausanne Swiss Federal Institute of Technology (EPFL) in January 1986, with a diploma in microengineering. He received the Ph.D. degree in June 1993, with a dissertation on the use of the linear prediction residue for text-independent speaker recognition.

He then joined the Institute of Microtechnology (IMT) of the University of Neuchâtel, Switzerland, where he worked as Research Assistant and Scientist, first in the domain of image processing (optical flow), then in the domain of speech processing (speech coding and speaker recognition). He then worked as a Visiting Fellow with the Biomedical Engineering and Instrumentation Program, National Institutes of Health (NIH), Bethesda MD, where he developed research interests that include splines and multiresolution signal representations, geometric image transformations, and biomedical image registration. Since November 1998, he has been with the Swiss Federal Institute of Technology Lausanne as First Assistant, where he extends his activities to research in interpolation and in volume rendering, and to mentoring of Ph.D. students.

Michael Unser (M’89–SM’94–F’99) received the M.S. (summa cum laude) and Ph.D. degrees in electrical engineering in 1981 and 1984, respectively, from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.

From 1985 to 1997, he was with the Biomedical Engineering and Instrumentation Program, National Institutes of Health, Bethesda, MD. He is now Professor and Head of the Biomedical Imaging Group at EPFL. His main research area is biomedical image processing. He has a strong interest in sampling theories, multiresolution algorithms, wavelets, and the use of splines for image processing. He is the author of 90 published journal papers in these areas. He was on the editorial board of *Signal Processing*.

Dr. Unser is an Associate Editor for the IEEE TRANSACTIONS ON MEDICAL IMAGING. He has been on the editorial boards the IEEE TRANSACTIONS ON IMAGE PROCESSING from 1992 to 1995 and the IEEE SIGNAL PROCESSING LETTERS from 1994 to 1998. He serves as regular chair for the SPIE Conference on Wavelets, which has been held annually since 1993. He received the 1995 Best Paper Award and the 2000 Magazine Award from the IEEE Signal Processing Society.