

Global Segmentation and Curvature Analysis of Volumetric Data Sets Using Trivariate B-Spline Functions

Octavian Soldea, Gershon Elber, *Member, IEEE*, and Ehud Rivlin, *Member, IEEE*

Abstract—This paper presents a method to *globally* segment volumetric images into regions that contain convex or concave (elliptic) iso-surfaces, planar or cylindrical (parabolic) iso-surfaces, and volumetric regions with saddle-like (hyperbolic) iso-surfaces, regardless of the value of the iso-surface level. The proposed scheme relies on a novel approach to globally compute, bound, and analyze the Gaussian and mean curvatures of an entire volumetric data set, using a trivariate B-spline volumetric representation. This scheme derives a new differential scalar field for a given volumetric scalar field, which could easily be adapted to other differential properties. Moreover, this scheme can set the basis for more precise and accurate segmentation of data sets targeting the identification of primitive parts. Since the proposed scheme employs piecewise continuous functions, it is precise and insensitive to aliasing.

Index Terms—Gaussian and mean curvature, symbolic computation, global analysis, segmentation.

1 INTRODUCTION

THE availability in recent years of a broad variety of range and volumetric images has presented new problems and challenges for the scientific community. In this context, segmentation is still a major conundrum and central research topic [8], [9], [32], [30], [46]. For example, in [8], the authors employed segmentation processes in order to reconstruct planar surfaces occluded by objects in range images. In [46], surface segmentation is employed in tasks of identification and searching of objects in a database of three-dimensional objects. In [32], range image segmentation is used in tasks of classification of objects, a process that is also known as generic recognition. In [9], the role of segmentation for reconstruction of volumetric parts from range data is discussed. In [30], a finite element segmentation technique is employed in the visualization and analysis of a heart captured in a sequence of volumetric images.

In [29], Marshall et al. presented a segmentation process that detects primitives such as planes, spheres, cylinders, cones, and tori from range data images. In this context, volumetric segmentation can be employed as well. A potential application of volumetric segmentation for use in identifying cylinders is illustrated in this work.

Most of the curvature computation literature deals with 3D meshes, which are typically obtained from 3D scanners. In contrast, volumetric images are commonly obtained from devices such as CTs, MRIs, and SEMs. Segmentation of volumetric images represents a new research field with immense and unexplored potential for computer vision applications. Our scheme represents a novel approach to

volumetric segmentation, employing a global analysis of the input data, and can be employed in classical segmentation of 3D images as well.

One way of handling the segmentation problem is by computing or estimating differential geometrical properties of the analyzed objects; see, for example, [3], [9], [18], [39], [43]. Among the differential properties that are widely used, Gaussian and mean curvatures take center stage, being fundamental prescriptions of an intrinsic surface property [26].

In this paper, we present a technique to globally and simultaneously compute the Gaussian and mean curvatures for (all iso-surfaces of) volumetric images. Given a volumetric data set $f(u, v, w)$, we are able to compute $K(u, v, w)$ and $H(u, v, w)$, the scalar fields that represent the Gaussian and mean curvatures of the iso-surface at (u, v, w) , as well as other differential characteristics. The introduced computational capability opens the way for a more precise and robust global curvature-based segmentation of volumetric data sets. To the best of our knowledge, all contemporary algorithms compute the curvature properties in discrete locations only. In the presented approach, we are able to globally, continuously, and simultaneously compute curvature property functions over the entire volume.

The presented scheme offers additional advantages in its improved accuracy in detecting boundaries of curvature-based segmented regions. Another of its benefits is its ability to perform global curvature analysis that is insensitive to aliasing as well as ignorant of a specific iso-level. This is in contrast to other methods that use discrete approaches.

This paper is organized as follows: In Section 2, we describe the current state-of-the-art in segmentation and curvature computation work. In Section 3, we provide some necessary mathematical background and, in Section 4, we describe the mechanism for evaluating the Gaussian and mean curvatures used in our presented approach. In Section 5, segmentation of the volume as well as curvature-based iso-surface extractions are considered using the

• The authors are with the Technion, Israel Institute of Technology, Department of Computer Science, Haifa 32000, Israel.
E-mail: {octavian, gershon, ehudr}@cs.technion.ac.il.

Manuscript received 23 Mar. 2005; accepted 29 June 2005; published online 13 Dec. 2005.

Recommended for acceptance by R. Klette.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0150-0305.

introduced scheme. A few advantages over traditional voxel-based schemes are also presented. In Section 6, several examples of our algorithm applied to volumetric and 3D scanned images are portrayed and, finally, in Section 7, we conclude.

2 RELATED WORK

Below, we provide an overview of the current state-of-the-art in curvature estimation over two-manifolds. The interest of the scientific community in segmenting 3D images has resulted in a plethora of techniques for surface segmentation. We describe several surface related works in Section 2.2. Several volumetric related studies that analyzed volumetric data sets are described in Section 2.3.

2.1 Evaluating Curvature

Being orientation independent of surface characteristics, curvature values are appropriate for registration and matching tasks. An example of curvature evaluation employed in tasks of registration and matching can be found in [45].

A significant part of contemporary research on curvature analysis focuses on polygonal meshes. In [34], a 3×3 symmetric matrix is defined by integrating the normal curvature around each vertex P . The neighborhood points of P are used in order to approximate several normal curvatures (in the directions of these neighbors). Thus, the integration becomes an angular discrete summation of normal curvature values. Two of the eigenvalues of the computed 3×3 symmetric matrix provide the principal curvatures.

The abundance of papers that deal with curvature estimation methods, employing polygonal meshes and also other representations, spurred interest in comparing the accuracies obtained by these methods [14]. In [14], the authors compared five methods of computing the curvature on surfaces. The first three methods are analytic and work in three steps: fit a surface to the values around the point of interest, evaluate the second derivatives, and compute the mean and Gaussian curvatures. The fitting of surfaces methods are different in each algorithm and are based on a least squares error with orthogonal polynomials, a linear regression with a biquadratic surface, and a B-spline approximation. Two other methods are based on measuring the change in the normal in the neighborhood of the analyzed point. These two methods employ an averaging over several normal curvature values in a number of directions, where the evaluation of the normal curvatures are discrete (numeric) computations. Flynn and Jain [14] concluded that the three analytic solutions are almost the same and are slightly better than the two discrete methods.

When the input data has a very high level of noise, techniques for computing the sign of the curvatures can be employed. In [33], the authors proposed a tensor voting scheme for inferring the sign and the direction of the principal curvatures from noisy 3D data. This method does not require local surface fitting, partial derivative computations, or normal vectors evaluation—tasks that are considered to be noise-sensitive when processing 3D data images. A first stage of voting provides orientation information represented by estimated tangents and normals, local structures being represented by ellipsoids. The ellipsoids are employed in a

second stage in order to detect the sign and directions of the principal curvatures, by voting, at each local structure.

2.2 Segmenting Range Images

Segmentation tasks often rely on edge-based techniques. In [41], the authors employed an edge growing method to segment range data images. The technique is based on employing noise adaptive masks, thus making this method suitable for noisy images.

A common problem strongly related to segmentation is the identification and fitting of surfaces of known types. In [29], the authors presented a segmentation process that detects primitives such as planes, spheres, cylinders, cones, and tori from range data images. The algorithm begins by selecting several arbitrary seeds points in the data and an iterative process that grows regions. At certain iterations, part of the detected regions are selected for further growing. Although during the iterative process the regions are allowed to overlap, at the end, the segmentation algorithm provides separate regions that are recovered as primitives.

Most segmentation research is based on curvature and differential characteristics evaluation. In [38], the authors proposed an evaluation scheme for a range image segmentation system. The segmentation system partitions the data into regions based on the signs of the Gaussian and mean curvatures. Morphology and threshold-based techniques are used to improve the quality of the results. The influence of several parameters is extensively studied and a framework for comparing implementations is suggested.

Unlike the segmentation computation literature, which is vast, studies that deal with comparison between segmentation methods are rare, mainly due to that fact that one is required to build a uniform platform for comparison and selection of methods from the literature. In [20], the authors presented a comparison of four algorithms for image segmentation of range images obtained from 3D laser scanners. One such algorithm, also published in [13], is based on curvature estimation. Gaussian and mean curvatures at each point in the resulting image are evaluated from a B-spline surface that locally approximates the geometry. The construction of the B-spline surface is achieved using a window of neighbors around the analyzed point. Interestingly enough, the conclusion of the authors in [20] is that, although this particular algorithm is the most time-expensive scheme, it also provides the best results.

2.3 Analyzing Volumetric Data Sets

Moving forward into the domain of volumetric data sets, we see that segmentation also comes into play in volumetric analysis. In [27], the authors presented an algorithm for the segmentation of volumetric images that works in four stages. In the first step, an intensity histogram of the voxels is computed. Based on the histogram, several thresholds are established. The thresholds define regions that are revealed in the second step. Once these are known, a seed point is selected and given as input to a region-growing algorithm. The boundaries obtained here are considered inaccurate. In the third step, rays are cast from each boundary voxel toward all the 26 neighboring voxels and a reconstruction and classification heuristic is applied. Finally, in the fourth step, a technique of contrast enhancement is applied toward the final volumetric rendering.

A comprehensive work on the theory of curvature evaluation on volumetric data sets, from a mathematical point of view, can be found in [31]. A systematic development of the theory of computing the curvature of volumetric images can be found in [16]. The author in [16] presented methods for computing the curvature of piecewise linear two-manifold surfaces (triangulated surfaces) and three-manifold graphs of trivariate functions. In the two-dimensional case, the neighbors of each node of the triangulated surface define an approximation of the surface that is modeled as a bivariate polynomial. The bivariate polynomial is characterized by several coefficients that are employed in the computation of Gaussian and mean curvatures. In the three-dimensional case, the computation of the curvatures of the three-dimensional graphs of trivariate functions consists of a generalization of the two-dimensional case. The bivariate polynomial represents a good local approximation of the surface at any point of the triangulation given in the input. Similarly to the bivariate polynomial case, the trivariate polynomial represents a good approximation of the space at any point of the three-dimensional graph given in the input. However, both computations are performed at discrete points only, as dictated by the triangulations or the graphs in the input. The author in [16] presented several examples of synthetic objects that were color-coded to follow curvature values. Moreover, several measurements of the errors of the approximation were presented using the root mean square between exact analytic and approximated values over each point of interest of several synthetic images.

Being able to view only the outside boundaries of objects, in general, the human eye experiences difficulties in fully comprehending the geometry of a volumetric image. Ways to provide visual cues are used in nonphoto realistic rendering. For example, in [22], curvature evaluation is employed in computing strokes textures of volumetric images. Moreover, in [17] and [25], volumetric rendering processes employ transfer functions. The transfer functions that are used there associate color cues to voxels based on principal curvature values.

Weinkauff and Theisel [42] used a global curvature analysis approach and presented a technique for evaluating the curvature and torsion of 3D vector fields. Several differential characteristics of the 3D vector fields are used for computing iso-surfaces, which finally are employed toward the analysis and visualization of the input images. The importance of [42] could also be found in detecting and analyzing critical points of vector fields.

In [35] and [36], Thirion and Gourdon presented a technique for estimating curvature values of iso-intensity surfaces from volumetric data sets. The estimation of curvature properties requires the approximation of derivatives in the input images and, in [35] and [36], this is performed by applying a discrete local Gaussian filter [4] over the volumetric image. Moreover, in [35] and [36], the authors defined four types of possible registration curves, taking into consideration the local minimum or maximum of the largest or next-to-largest curvature along lines of curvature. The curves on the surface in which the largest principal curvature is at a local maximum are called crest lines. The authors claimed that the crest lines are the most stable ones of the four types of curves they considered. They foresee registration and pattern recognition as the main applications of crest lines.

3 BACKGROUND

Consider $f(u, v, w)$, a C^2 trivariate function. In this section, we briefly present the mathematical background necessary to compute the Gaussian and mean curvatures of an iso-surface $f(u, v, w) = f_0$. We express the main differential components as in [35] and [36].

Given a bivariate function $g(u, v)$, denote by g_u and g_v the two partial derivatives of $g(u, v)$ in the u and v directions, respectively. Similarly, for any trivariate function $h(u, v, w)$, let h_u , h_v , and h_w be the partial derivatives of $h(u, v, w)$ with respect to the u , v , and w directions.

From the implicit function theorem, there exists a scalar function $S(u, v)$ that dictates that the parametric surface $S(u, v) = (u, v, S(u, v))$, which is a local parametrization of an iso-surface level $f(u, v, S(u, v)) = f_0$, f_0 constant. Then, by differentiating with respect to u , we have

$$f_u(u, v, S(u, v)) + f_w(u, v, S(u, v))S_u = 0;$$

further, one can deduce that $S_u = -\frac{f_u}{f_w}$, and $S_u = (1, 0, -\frac{f_u}{f_w})$. Here, we assumed that $f_w \neq 0$. We show in Appendix A (see <http://computer.org/tpami/archives.htm> for more information) how we handle the case in which $f_w = 0$.

Let E, F, G and L, M, N be the coefficients of the first and second fundamental forms of iso-surface $f(S) = f_0$, respectively, and let $n = S_u \times S_v$ be the unnormalized normal to S at (u, v) . Then,

$$E = \langle S_u, S_u \rangle = 1 + S_u^2 = 1 + \frac{f_u^2}{f_w^2} = \frac{f_u^2 + f_w^2}{f_w^2} = \frac{\tilde{E}}{f_w^2}. \quad (1)$$

Similarly, other differential terms can be computed as well:

$$F = \langle S_u, S_v \rangle = \frac{f_u f_v}{f_w^2} = \frac{\tilde{F}}{f_w^2}, \quad (2)$$

$$G = \langle S_v, S_v \rangle = \frac{f_v^2 + f_w^2}{f_w^2} = \frac{\tilde{G}}{f_w^2}, \quad (3)$$

$$D = EG - F^2 = \frac{f_u^2 + f_v^2 + f_w^2}{f_w^2} = \frac{\tilde{D}}{f_w^2}, \quad (4)$$

$$\begin{aligned} L &= \langle S_{uu}, \bar{n} \rangle \\ &= \frac{2f_u f_w f_{uw} - f_u^2 f_{ww} - f_w^2 f_{uu}}{\tilde{D}^{\frac{1}{2}} f_w^2} = \frac{\tilde{L}}{\tilde{D}^{\frac{1}{2}} f_w^2}, \end{aligned} \quad (5)$$

$$\begin{aligned} M &= \langle S_{uv}, \bar{n} \rangle \\ &= \frac{f_u f_w f_{vw} + f_v f_w f_{uw} - f_u f_v f_{ww} - f_w^2 f_{uv}}{\tilde{D}^{\frac{1}{2}} f_w^2} = \frac{\tilde{M}}{\tilde{D}^{\frac{1}{2}} f_w^2}, \end{aligned} \quad (6)$$

and

$$\begin{aligned} N &= \langle S_{vv}, \bar{n} \rangle \\ &= \frac{2f_v f_w f_{vw} - f_v^2 f_{ww} - f_w^2 f_{vv}}{\tilde{D}^{\frac{1}{2}} f_w^2} = \frac{\tilde{N}}{\tilde{D}^{\frac{1}{2}} f_w^2}, \end{aligned} \quad (7)$$

where \bar{n} is the normalized surface normal. Thus, the Gaussian curvature of an iso-surface of $f(u, v, w)$ is the equivalent of

$$K = \frac{LN - M^2}{D} = \frac{\tilde{L}\tilde{N} - \tilde{M}^2}{D^2 f_w^6} \quad (8)$$

$$= \frac{\tilde{L}\tilde{N} - \tilde{M}^2}{\tilde{D}^2 f_w^2} = \frac{\tilde{K}}{\tilde{D}^2 f_w^2}.$$

The values in (1) to (8) are w -biased and sensitive to a vanishing f_w . Processing further, we have:

$$K = (2f_u f_v f_w f_{uv} f_{vw} + 2f_u f_w f_{uw} f_{vw} + 2f_v f_w f_{vw} f_{uw} - 2f_u f_w f_{uw} f_{vw} - 2f_v f_w f_{vw} f_{uw} - 2f_u f_v f_{uv} f_{vw} + f_w^2 f_{uu} f_{vv} + f_u^2 f_{vv} f_{ww} + f_v^2 f_{uu} f_{ww} - f_u^2 f_{vv}^2 - f_v^2 f_{uu}^2 - f_w^2 f_{uv}^2) / \tilde{D}^2 = \frac{\tilde{K}}{\tilde{D}^2}.$$

The details involved in the development of (9) can be found in Appendix A (see <http://computer.org/tpami/archives.htm> for more information). Further, K is now symmetric with respect to u , v , or w . $K(u, v, w)$ is defined for the entire parametric domain of $f(u, v, w)$. Denote by $\vec{\nabla} f = (\frac{\partial f}{\partial u}, \frac{\partial f}{\partial v}, \frac{\partial f}{\partial w})$ the gradient of f , and assume $\vec{\nabla} f$ is never zero. Then, given a (u_0, v_0, w_0) location, the iso-surface of $f(u, v, w)$ at (u_0, v_0, w_0) is well-defined. Having a well-defined iso-surface at (u_0, v_0, w_0) , $K(u_0, v_0, w_0)$ is also well-defined, as $\tilde{D}^2 = \langle \vec{\nabla} f, \vec{\nabla} f \rangle$ never vanishes. It is important to note that K is a rational expression of piecewise polynomials, provided f is a piecewise polynomial.

In a similar way to the computation of K in (8) and (9), a formula for H , the mean curvature, can be derived as

$$H = (2f_u f_v f_w f_{uv} + 2f_u f_w f_{uw} + 2f_v f_w f_{vw} - (f_u^2 + f_v^2) f_{ww} - (f_u^2 + f_w^2) f_{vv} - (f_u^2 + f_v^2) f_{ww}) / (2\tilde{D}^3) = \frac{\tilde{H}}{2\tilde{D}^3}.$$

Note that H^2 is a rational expression of piecewise polynomials as well.

4 CURVATURES OF ISO-SURFACES OF TRIVARIATE B-SPLINE FUNCTIONS

We propose to use trivariate B-spline functions for representing objects due to their properties of variation diminishing. These properties assure good approximations to shapes. Moreover, trivariate B-spline functions are piecewise polynomial. Thus, if $f(u, v, w)$ is the trivariate B-spline function in the input, the resulting $K(u, v, w)$ and $H(u, v, w)$ curvatures are rational expressions of piecewise polynomial functions. The fact that the input and the output are (rational) expressions of piecewise polynomial functions represents a type of closure. In other words, the whole range of operators that apply to B-spline functions can also be applied to the input as well as the output functions. Furthermore, trivariate B-spline functions have good continuity properties and an analysis of the continuity properties of the resulting Gaussian and mean curvatures is provided in this section.

Let $B_{i,k,\tau}(t)$ be the i th B-spline blending function of degree k defined over knot sequence τ [5]. Now, consider the trivariate B-spline function,

$$f(u, v, w) = \sum_{i=0}^{n_u} \sum_{j=0}^{n_v} \sum_{l=0}^{n_w} p_{i,j,l} B_{i,k_u,\tau_u}(u) B_{j,k_v,\tau_v}(v) B_{l,k_w,\tau_w}(w), \quad (11)$$

with $\chi = (n_u + 1)(n_v + 1)(n_w + 1)$ scalar coefficients $p_{i,j,l}$, B-spline basis functions $B_{i,k_u,\tau_u}(u)$, $B_{j,k_v,\tau_v}(v)$, and $B_{l,k_w,\tau_w}(w)$, degrees k_u , k_v , and k_w , and knot sequences τ_u , τ_v , and τ_w , respectively. Hereafter, we will employ $B_i(t)$ or $B_{i,k}(t)$ to denote $B_{i,k,\tau}(t)$ whenever the degree or the knot vector can be inferred from the context. Given a regular, piecewise constant volumetric data set, one can treat it as a piecewise constant B-spline trivariate. Moreover, a piecewise trilinear B-spline trivariate will also interpolate this volumetric data set by simply using the voxels' data values as the $p_{i,j,l}$ coefficients of the trivariate. For higher order trivariate functions, the result is only an approximation, when $p_{i,j,l}$ are the coefficients. Hence, in practice, two options are available. One option is to solve an interpolation problem, fitting $f(u, v, w)$ to the original piecewise constant data. Alternatively, a second option consists of providing a bound on the error of the approximation, when using the voxels' data values as the $p_{i,j,l}$ coefficients. Although interpolation is sometimes preferred, it could be less stable.

Consider $\tau_i \in \tau_u$, a single interior knot in the u direction such that $\tau_i < \tau_{i+1}$, and similarly for $\tau_j \in \tau_v$, $\tau_j < \tau_{j+1}$ and $\tau_l \in \tau_w$, $\tau_l < \tau_{l+1}$. Then, in each nontrivial polynomial subdomain $\mathcal{D}_{i,j,l} : [\tau_i, \tau_{i+1}] \times [\tau_j, \tau_{j+1}] \times [\tau_l, \tau_{l+1}]$ of the parametric space of f , the Gaussian curvature, K , is a rational function in $p_{i,j,l}$ and in (u, v, w) , where $u \in [\tau_i, \tau_{i+1}]$, $v \in [\tau_j, \tau_{j+1}]$, and $w \in [\tau_l, \tau_{l+1}]$. In this section, we will describe how to efficiently compute the Gaussian curvature $K(u, v, w)$, in two steps. First, in Section 4.1, we evaluate the numerator and the denominator of K as a trivariate Bézier representation in each subdomain, $\mathcal{D}_{i,j,l}$. In a second step that is briefly described in Section 4.2, we merge the rational form of K in all the polynomial subdomains into a simple B-spline trivariate function representation, over the entire domain of $f(u, v, w)$.

4.1 Evaluation of K and H as Bézier Forms

We seek to define K using the coefficients $p_{i,j,l}$ of $f(u, v, w)$ for a single subdomain $\mathcal{D}_{i,j,l}$ and compute the numerator and the denominator of the Gaussian curvature following (9). The expression for K as a function of $p_{i,j,l}$ was obtained with the aid of the Maple [6] symbolic manipulation program, also available at [2]. We use a symbolic interpolation process to convert the rational result to a Bézier form. In a similar way to K , H^2 can be evaluated as well.

Let

$$\theta_{i,n}(t) = \binom{n}{i} (1-t)^{n-i} t^i$$

be the Bernstein-Bézier basis function of degree n . Consider f for one subdomain $\mathcal{D}_{i,j,l}$. Let $\psi(u, v, w)$ be one of the differential components, defined over $\mathcal{D}_{i,j,l}$, as presented in (1)-(7), (9), and (10). f and ψ

$$\left\{ \psi \in \left\{ \tilde{D}, \tilde{E}, \tilde{F}, \tilde{G}, \tilde{L}, \tilde{M}, \tilde{N}, \tilde{K}, \tilde{K}, \tilde{H}, \tilde{H} \right\} \right\}$$

are both polynomials.

Let o_u , o_v , and o_w be the degrees of $\psi(u, v, w)$ in u , v , and w , in one $\mathcal{D}_{i,j,l}$ domain, respectively, following Table 1, and let

TABLE 1
Degrees of Differential Components

Component	Degree in u	Degree in v	Degree in w	Degree in $p_{i,j,l}$ coefficients
$f(u, v, w)$	k_u	k_v	k_w	1
$f_u(u, v, w)$	$k_u - 1$	k_v	k_w	1
$f_v(u, v, w)$	k_u	$k_v - 1$	k_w	1
$f_w(u, v, w)$	k_u	k_v	$k_w - 1$	1
$f_{uu}(u, v, w)$	$k_u - 2$	k_v	k_w	1
$f_{vv}(u, v, w)$	k_u	$k_v - 2$	k_w	1
$f_{ww}(u, v, w)$	k_u	k_v	$k_w - 2$	1
$f_{uv}(u, v, w)$	$k_u - 1$	$k_v - 1$	k_w	1
$f_{uw}(u, v, w)$	$k_u - 1$	k_v	$k_w - 1$	1
$f_{vw}(u, v, w)$	k_u	$k_v - 1$	$k_w - 1$	1
$\tilde{D} = f_u^2 + f_v^2 + f_w^2$	$2k_u$	$2k_v$	$2k_w$	2
$\tilde{D}^2 = (f_u^2 + f_v^2 + f_w^2)^2$	$4k_u$	$4k_v$	$4k_w$	4
$\tilde{E} = f_u^2 + f_v^2$	$2k_u$	$2k_v$	$2k_w$	2
$\tilde{F} = f_u f_v$	$2k_u - 1$	$2k_v - 1$	$2k_w$	2
$\tilde{G} = f_v^2 + f_w^2$	$2k_u$	$2k_v$	$2k_w$	2
$\tilde{L} = 2f_u f_v f_w$ $- f_u^2 f_{vw}$ $- f_v^2 f_{uw}$	$3k_u - 2$	$3k_v$	$3k_w - 2$	3
$\tilde{M} = f_u f_v f_{vw}$ $+ f_v f_w f_{uv}$ $- f_u f_v f_{uw}$ $- f_v^2 f_{uv}$	$3k_u - 1$	$3k_v - 1$	$3k_w - 2$	3
$\tilde{N} = 2f_v f_w f_{vw}$ $- f_v^2 f_{ww}$ $- f_w^2 f_{vv}$	$3k_u$	$3k_v - 2$	$3k_w - 2$	3
$\tilde{K} = \tilde{L}\tilde{N} - \tilde{M}^2$	$6k_u - 2$	$6k_v - 2$	$6k_w - 4$	6
$\tilde{K} = \tilde{K}/f_w^2$	$4k_u - 2$	$4k_v - 2$	$4k_w - 2$	4
$\tilde{H} = \tilde{E}\tilde{N} - 2\tilde{F}\tilde{M} + \tilde{G}\tilde{L}$	$5k_u$	$5k_v$	$5k_w - 2$	5
$\tilde{H} = \tilde{H}/2f_w^2$	$3k_u$	$3k_v$	$3k_w$	3
\tilde{H}	$6k_u$	$6k_v$	$6k_w$	6
$\tilde{D}^3 = (f_u^2 + f_v^2 + f_w^2)^3$	$6k_u$	$6k_v$	$6k_w$	6

$$u_i \neq u_j, \tag{13}$$

for any $i \neq j$ and $i, j \in \{0..o_u\}$; the same holds for v and w sequences.

Assume that $f(u, v, w)$ is a piecewise polynomial function of degrees k_u, k_v, k_w . Table 1 summarizes the degrees of the different differential terms leading to K and H^2 , following (1)-(10). With the aid of Maple [6], we represent the differential components in Table 1 as polynomial functions in u, v, w and $p_{i,j,l}$. For example, \tilde{D} , is a polynomial of degrees $2k_u, 2k_v, 2k_w$ in (u, v, w) , respectively. Further, because ∇f is a linear polynomial in $p_{i,j,l}$, $\tilde{D} = \langle \nabla f, \nabla f \rangle$ is a quadratic function in $p_{i,j,l}$. While, in general, the degrees of the terms grow larger as we progress, \tilde{K} is obtained from \tilde{K} by dividing by f_w^2 . Thus, the degrees of \tilde{K} are smaller than those of \tilde{K} (see (9)).

Consider a given trivariate f with known coefficients $p_{i,j,l}$, which could be substituted into $\psi(u, v, w)$. The problem of deriving $q_{i,j,l}$ in (12) could be mapped to a system of equations with \mathcal{O} unknowns and \mathcal{O} constraints:

$$\theta(u_i, v_j, w_l) = \psi(\tau_i + u_i, \tau_j + v_j, \tau_l + w_l), \tag{14}$$

where $i = 0, \dots, o_u, j = 0, \dots, o_v, l = 0, \dots, o_w$, and u_i, v_j , and $w_l \in [0..1]$. Note, $p_{i,j,l}$ are now specified. System (14) is modeled as follows: Let $\Theta \in M_{\mathcal{O} \times \mathcal{O}}, Q \in M_{\mathcal{O} \times 1}$, and $\Psi \in M_{\mathcal{O} \times 1}$, where $M_{i \times j}$ denotes a matrix of size i by j . Let $r = l(o_u + 1)(o_v + 1) + j(o_u + 1) + i$. Consequently, the s 'th element of the r 'th row, $\Theta_{r,s} \in \Theta$, equals

$$\Theta_{r,s} = \theta_{\alpha, o_u}(u_i) \theta_{\beta, o_v}(v_j) \theta_{\gamma, o_w}(w_l), \tag{15}$$

where $s = \gamma(o_u + 1)(o_v + 1) + \beta(o_u + 1) + \alpha$. Similarly, let the s 'th element of Q be $Q_s = q_{\alpha, \beta, \gamma}$ and the r 'th element of Ψ be $\Psi_r = \psi(\tau_i + u_i, \tau_j + v_j, \tau_l + w_l)$. Then, (14) is equivalent to

$$\Theta Q = \Psi. \tag{16}$$

Ψ_r is symbolically pre-evaluated once into \mathcal{O} polynomial equations in $p_{i,j,l}$, as in the right side of (14). Given a specific f , all the Ψ_r functions are evaluated by substituting in the $p_{i,j,l}$ coefficients of f .

We discuss below several options for choosing interpolation points. Moreover, we provide a justification of the fact that Θ is nonsingular in Appendix C (see <http://computer.org/tpami/archives.htm> for more information).

4.1.1 Greville Abscissas Interpolation Points

A first possible option for choosing interpolation points is $u_i = \frac{i}{o_u}, v_j = \frac{j}{o_v}$, and $w_l = \frac{l}{o_w}$, where $i \in \{0..o_u\}, j \in \{0..o_v\}$, and $l \in \{0..o_w\}$. In other words, we evaluate $\theta(x, y, z)$ at three-dimensional independent parametric points of the form,

$$\left(\frac{i}{o_u}, \frac{j}{o_v}, \frac{l}{o_w} \right), \tag{17}$$

points that are also known as node points or Greville abscissas [12]. The functions,

$$\theta_{i, o_u}(u), \theta_{j, o_v}(v), \text{ and } \theta_{l, o_w}(w),$$

form a basis for the polynomials of degree o_u, o_v , and o_w , in u, v , and w , respectively.

be a Bézier polynomial function of the same degree. We seek a Bézier representation for the polynomial function of ψ and find it by the uniqueness of the polynomial representation and symbolic interpolation constraints at $\mathcal{O} = (o_u + 1)(o_v + 1)(o_w + 1)$ unique parameter values.

In brief, pre-evaluate $K(u_i, v_j, w_l)$ for all $i \in \{0..o_u\}, j \in \{0..o_v\}$, and $l \in \{0..o_w\}$, so that we end up with \mathcal{O} equations in $p_{i,j,l}$ only. Here, u_i, v_j , and $w_l \in [0..1]$ for all $i \in \{0..o_u\}, j \in \{0..o_v\}$, and $l \in \{0..o_w\}$, representing interpolation argument values. Once a specific f is given, the $p_{i,j,l}$ coefficients of f are substituted in the equations. The values of K at these \mathcal{O} locations are used to formulate an interpolation problem, yielding the $q_{i,j,l}$ coefficients in (12). The details of this process follow.

Consider three sequences of interpolation argument values u_i, v_j , and $w_l \in [0..1]$ for all $i \in \{0..o_u\}, j \in \{0..o_v\}$, and $l \in \{0..o_w\}$. Moreover, we assume that

The diagonal elements of matrix Θ are larger than any other element in the same column or row due to the fact that $\theta_{j,k}(\frac{2}{k}) \geq \theta_{j,k}(x)$ for any $x \in [0, 1]$ [5]. Hence, this interpolation scheme provides more stability when the linear system of equations is solved, as

$$Q = \Theta^{-1}\Psi. \quad (18)$$

4.1.2 Chebyshev Abscissas Interpolation Points

A better option for choosing interpolation points is

$$u_i = \frac{1 + \cos\left(\frac{(2i+1)\pi}{o_u+1}\right)}{2}, \quad v_j = \frac{1 + \cos\left(\frac{(2j+1)\pi}{o_v+1}\right)}{2}, \text{ and}$$

$$w_l = \frac{1 + \cos\left(\frac{(2l+1)\pi}{o_w+1}\right)}{2},$$

where $i \in \{0..o_u\}$, $j \in \{0..o_v\}$, and $l \in \{0..o_w\}$. In other words, we evaluate $\theta(x, y, z)$ at three-dimensional independent parametric points of the form,

$$\left(\frac{1 + \cos\left(\frac{(2i+1)\pi}{o_u+1}\right)}{2}, \frac{1 + \cos\left(\frac{(2j+1)\pi}{o_v+1}\right)}{2}, \frac{1 + \cos\left(\frac{(2l+1)\pi}{o_w+1}\right)}{2} \right), \quad (19)$$

points that are also known as Chebyshev abscissas [7]. The condition numbers of the matrices $\theta(x, y, z)$ evaluated at Chebyshev abscissas are from twice up to 60 times lower than their counterparts evaluated at Greville abscissas, for the degrees we use: 6, 8, 9, 10, 12, and 18.

When interpolating at equidistant points, the approximation graph can have a disturbing course between the interpolation points (see Runge phenomenon [7]). In contrast, interpolation via Chebyshev points is the best choice for minimizing numerical errors and coping with Runge phenomenon.

Θ is also independent of the input and, hence, one can precisely precompute Θ and Θ^{-1} once. Nevertheless, the direct inversion of Θ is not trivial. In the following, we analyze the reasons for which the inversion of Θ is not trivial. Moreover, we propose several possible solutions to the interpolation problem based on Θ and Θ^{-1} .

Let y_j be a sequence of numerical values, $j \in \{0, 1, \dots, r\}$. The Vandermonde matrix is a matrix of the form

$$\mathbf{Van}(y_0, y_1, \dots, y_r) = \begin{pmatrix} y_0^0 & y_0^1 & \dots & y_0^r \\ y_1^0 & y_1^1 & \dots & y_1^r \\ \vdots & \vdots & \ddots & \vdots \\ y_r^0 & y_r^1 & \dots & y_r^r \end{pmatrix}.$$

There is a strong relationship between Θ and the Vandermonde matrices. We show this relationship in Appendix C (see <http://computer.org/tpami/archives.htm> for more information).

It is known that Vandermonde matrices have large condition numbers and this fact implies that the computation of their inverse is numerically unstable. Moreover, multiplying matrices with increasing condition numbers would typically yield results with increased numerical errors. A good presentation of the numerical problems that appear when using matrices with large condition numbers can be found in [21]. If not enough, herein the size of matrices Θ and

Θ^{-1} is very large; in the thousands. For example, in the case when we compute the denominator of K and $k_i = k_j = k_l = 3$, we have $o_i = o_j = o_l = 12$ and $\Theta, \Theta^{-1} \in M_{2,197 \times 2,197}$. In Appendix C, we discuss in detail ways to compute Θ^{-1} in a more stable manner. In Appendix D (see <http://computer.org/tpami/archives.htm>) for more information, we present a computation scheme that has lower memory complexities than (16), a computation scheme that improves the time complexity as well.

The direct computation of Θ^{-1} is time consuming and presents high numerical errors. In Appendix D (see <http://computer.org/tpami/archives.htm> for more information), we present a method of computing Θ^{-1} employing the inverses of three matrices that are significantly smaller compared to Θ^{-1} . The inverted matrices are further used in a system decomposition described in Appendix D.

In order to alleviate the difficulty of direct computation of Θ^{-1} , we decompose (16) into smaller subsystems based on matrices θ^u , θ^v , and θ^w . In Appendix D (see <http://computer.org/tpami/archives.htm> for more information), we present a computation scheme that is more efficient as well as having less memory requirements than the ones used in (16).

The most important factor in the accuracy of the computations relies on choosing the interpolation points. In the case of Gaussian curvature computation for $k = 3$ and double precision, K is computed with an accuracy of five significant decimal digits in the mantissa using the solution presented in Appendix D (see <http://computer.org/tpami/archives.htm> for more information) with an interpolation on Chebyshev points. Under the same conditions, one can only achieve an accuracy of three significant decimal digits when working on Greville's interpolation abscissas. We tested the accuracy on a machine that follows the IEEE numeric standards for floating point representation with 64 bits.

4.2 Merging into a B-Spline Form

Recall in (11) that the input of our scheme consists of $\chi p_{i,j,l}$ coefficients and that we want to compute the Gaussian and the mean curvatures as trivariate functions. Our aim now is to merge the χ Bézier trivariates in (11), each defined for a different domain $\mathcal{D}_{i,j,l}$, into one large B-spline trivariate function that is defined over the entire domain of $f(u, v, w)$. Herein, we consider the univariate case and describe how to merge several Bézier curve segments into one B-spline curve with C^0 continuity. The motivation for the C^0 examples stems from considering the curvature continuity of cubic splines. The extension to the trivariate case is a simple generalization that takes place in each of its three axes, independently.

Let $\{\tau_j\}$ be a sequence of knots with the support domain $[\tau_k \dots \tau_{n+1}]$. Assume $\theta(x) = \sum_{i=0}^k q_i \theta_{i,k}(x)$ is a Bézier curve of degree k that defines the B-spline polynomial segment between τ_j and τ_{j+1} . Construct a new knot sequence $t = \{t_j\}$ with a multiplicity of k knots at each interior knot $\{\tau_j\}$. Then, the coefficients of the merged B-spline curve defined over $\{t_j\}$ for $x \in [\tau_j, \tau_{j+1})$, such that $\tau_{j+1} > \tau_j$ equated with q_i , are

$$\theta(x) = \sum_{i=0}^k q_i \theta_{i,k}(x) = \sum_{l=jk}^{(j+1)k} p_l B_{l,k,t}(x), \quad (20)$$

where $p_l = q_i$ for any $i \in \{0..k\}$ where $l = jk + i$.

While (20) offers one simple way of merging several Bézier segments into a single C^0 continuous curve, one can clearly

repeat the process for arbitrarily prescribed continuity. The prescribed continuity is directly governed by the multiplicity of the interior knots. For cubic B-spline trivariate functions, which are C^2 , in general, the curvature function is continuous; hence, the assumed C^0 continuity in this example.

5 SEGMENTATION OF VOLUMETRIC DATA SETS USING K AND H

This section presents several segmentation mechanisms for volumetric data sets using the scalar Gaussian curvature function we have just computed. The segmentation using the mean curvature is similar to the Gaussian one. Furthermore, several issues of augmenting and speeding up the evaluation process are also discussed.

Consider the Gaussian curvature of an iso-surface represented by a trivariate B-spline function. Given a scalar B-spline trivariate function $f(u, v, w)$ (11), we are able to *symbolically* compute (9), and represent the trivariate function $K(u, v, w)$ as a scalar B-spline trivariate function that globally represents the Gaussian curvature of any iso-surface of $f(u, v, w)$, for all possible locations and, hence, iso-levels. In other words, $K(u, v, w)$ is a rational form, provided f is. If f can be represented as a B-spline volumetric function, so can K . For example, if f is a tricubic polynomial, the numerator of K is a trivariate function of degrees 6 or 10, respectively, whereas its denominator has degrees 8 or 12, respectively, in each direction (see Table 1). With this approach, we are able to globally and simultaneously analyze all the regions in the entire volume for which the iso-surfaces assume certain Gaussian curvature values. $K(u, v, w)$ could be fixed as either a Bézier or a B-spline trivariate function, two forms that can yield bounds on the values that K can assume at a certain arbitrary subdomain $\mathcal{D}_{i,j,l}$ by simply examining the coefficients of the function at that subdomain. Further, with the subdivision capability of these representations, one can easily construct a divide-and-conquer algorithm to robustly converge at locations with specific values of K . These properties allow one to segment volumes in regions of interest (characterized by certain Gaussian curvature values) directly and without the need for an exhaustive sampling search. Moreover, this search, being symbolic and global, is immune to aliasing, is precise to within machine precision, and is independent of specific iso-values.

One of the most difficult problems in volumetric image processing is handling the size of the data. As stated earlier, if f is a tricubic, the numerator of K is a trivariate function of degree 10 in u , v , and w , whereas its denominator has degree 12. As a consequence, the two trivariates that represent the numerator and the denominator of $K(u, v, w)$ increase the needed data size by a factor of $(\frac{11+13}{4})^3$ for a tricubic, in each axis. For contemporary volumetric data sets, such an increase, of more than two orders of magnitudes, could be devastating. A remedy might be found in breaking the input volume into pieces and examining $K(u, v, w)$ incrementally in each polynomial subdomain instead of the entire domain of $f(u, v, w)$. In other words, we evaluate the Gaussian curvature for each polynomial subdomain $\mathcal{D}_{i,j,l}$ as a $K_{i,j,l}(u, v, w)$ function described in Section 4.1, process it, and immediately purge this $K_{i,j,l}$ for $\mathcal{D}_{i,j,l}$. No merging stage, as described in Section 4.2, is actually conducted for this segmentation application. At every point of time, only one $K_{i,j,l}(u, v, w)$ for one domain $\mathcal{D}_{i,j,l}$ is allocated.

We now start with a simple segmentation example, considering the solution for $K(u, v, w) = K_0$. This problem could be solved simply by applying the traditional Marching Cubes [28] algorithm to K . For example, if $K_0 = 0$, one is simultaneously extracting all the parabolic manifolds in the volume, regardless of their iso-values.

As part of the volumetric segmentation process, one can employ a geometric constraint solver for multivariate rational B-spline functions [11]. There are many methods that attempt to solve multivariate algebraic constraints. These methods include homotopy methods, symbolic Groebner basis, interval arithmetic methods, and subdivision methods (which) was used in our (work). An overview of various constraint solvers can be found in [19]. The specific solver is beyond the scope of this work, however, any such solver could be employed here once the curvature field is formulated as an algebraic constrain.

Specifically, the solver can seek the simultaneous solution of

$$\begin{cases} f(u, v, w) = f_0, & f_0 \text{ constant,} \\ K(u, v, w) = K_0, & K_0 \text{ constant.} \end{cases}$$

Equality as well as inequality constraints can be given to the solver. Hence, one can also solve for

$$\begin{cases} f(u, v, w) = f_0 & f_0 \text{ constant,} \\ |K(u, v, w)| \leq K_0, & K_0 \text{ positive constant.} \end{cases}$$

The solver has two phases. In the first phase, it recursively subdivides the parametric domain of the trivariate(s) in u , v , and w into voxels of certain set dimensions and tries to find an approximated solution point in each voxel. For each approximated solution that is found in the first stage, a second phase is applied. In this second phase, the solver improves the solution point set using a multivariate Newton-Raphson [7] iterative method.

Assume we are interested in processing and segmenting several iso-levels, f_0, f_1, \dots, f_n . For each polynomial subdomain $\mathcal{D}_{i,j,l}$, we compute the gradient of $f(u, v, w)$ as a trivariate Bézier, $\vec{\nabla} f_{i,j,l}(u, v, w) = (\frac{\partial f}{\partial u}, \frac{\partial f}{\partial v}, \frac{\partial f}{\partial w})$. We further process subdomain $\mathcal{D}_{i,j,l}$ only if $\|\vec{\nabla} f_{i,j,l}(u, v, w)\|$ presents magnitudes greater than a certain threshold in $\mathcal{D}_{i,j,l}$. This test is conducted by examining the magnitude of the control points of $\vec{\nabla} f_{i,j,l}$ and allows us to process only subdomains that contain information above a certain noise level. Again, note that the gradient does not depend on a certain iso-level value. For subdomains that are found to contain a sufficiently large gradient, we simultaneously solve, for example, for $|K| \leq \epsilon$ and $f = f_i$, $i = 0, 1, \dots, n$ using the abovementioned multivariate solver [11]. While solving for $K = 0$ is potentially simpler, as only the numerator of K needs to be processed, the approaching $K = 0$ was found to be unstable and too noise-sensitive when real noisy data was provided. In order to solve for $|K| \leq \epsilon$, we have to intersect the solutions of $\tilde{K} - \epsilon \tilde{D}^2 \leq 0$ with the ones of $\tilde{K} + \epsilon \tilde{D}^2 \geq 0$ (recall (9)), where $\epsilon > 0$ is some low positive constant. This scheme is demonstrated in Algorithm 1 for the example that seeks the parabolic regions in an iso-surface.

Algorithm 1:

Input:

- $f(u, v, w)$: a trivariate volumetric data set;
- f_0 : desired iso-level;
- ϵ : level of Gaussian curvature below which, we assume it is a parabolic domain;
- n_0 : gradient's noise level;

Output:

- iso-surface f_0 , with a prescribed curvature property;

Algorithm:

SegmentVolume(f, f_0, ϵ, n_0);

Begin

- if** ($\min(f) \leq f_0$ and $\max(f) \geq f_0$)
- if** (f is not a single polynomial)
- $f_a, f_b \leftarrow$ subdivided f in an interior knot along u, v , or w ;

SegmentVolume(f_a, f_0, ϵ, n_0);

SegmentVolume(f_b, f_0, ϵ, n_0);

else

$\nabla f \leftarrow$ gradient of f ;

if ($\exists u, v, w$ such that $\|\nabla f\| > n_0$)

$K \leftarrow$ Gaussian scalar field of f ;

 solve for $\begin{cases} f = f_0, \\ |K| \leq \epsilon, \end{cases}$

fi
 purge ∇f ;

fi

End

The scalar field of K spans the entire volume of f . Hence, one can sample K at any location (u_0, v_0, w_0) in the domain. Specifically, given an iso-surface $f_0 \in f$, one can sample K along f_0 and color iso-surface f_0 with color-coded curvature values.

Consider a fixed set S of close iso-surfaces. Two or more close iso-surfaces pass through almost the same trivariate cells. Thus, the time that is required for computing the trivariate functions in the cells that are passed by any subset of close iso-surfaces of S is almost the same. While other schemes require for each iso-surface at each point a new execution of a certain algorithm of curvature evaluation or approximation, our scheme implies only an evaluation of a trivariate B-spline function representing the curvature property.

An efficient scheme for generating iso-surfaces can be found in [23]. This algorithm is designated for cases when one wishes to generate many iso-surfaces in a huge volume. Although this algorithm could be reformulated for working with objects represented by trivariate B-spline functions, the problem of surface generation is beyond the goal of this work.

6 EXPERIMENTAL RESULTS

In all examples presented in this section, we compute the Gaussian and mean curvatures over each subdomain $\mathcal{D}_{i,j,l}$, evaluate or compute solutions for curvature constraints with



Fig. 1. Color coding for K or H only based segmentation. Blue means negative values, green means zero values, and red means positive values.

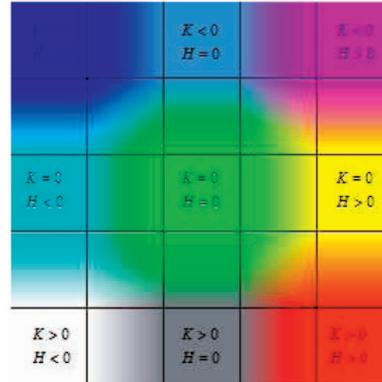


Fig. 2. Color coding for K and H based segmentation.

the multivariate solver, and then immediately purge the trivariate representing K and H over $\mathcal{D}_{i,j,l}$, as described in Algorithm 1. As presented in Section 4.2, it is possible to compute the whole trivariate B-spline function K for all the domain of f . The memory requirements for K for a 40^3 tricubic volumetric function are around $40^3 \times 8 \times (11^3 + 13^3)$ bytes ≈ 1.7 gigabytes, where 40^3 is the number of $\mathcal{D}_{i,j,l}$ subdomains, 8 bytes are assumed for each double precision number, and in each subdomain there are $11^3 + 13^3$ coefficients in the rational Bézier representation of K . Hence, the explicit representation of K and H for the entire domain is expected to be rarely computed using contemporary hardware.

We present a few examples of segmenting volumetric and 3D range data sets using the proposed curvature computation scheme. The examples presented were processed with the aid of the IRT Solid Modeling system [10]. In the figures in which either K or H are represented alone, the colors are coded as in Fig. 1, where red corresponds to a positive value, green to a zero value, and blue to a negative value. In the case of Gaussian curvature the colors red, green, and blue indicate convex or concave (elliptic) iso-surfaces, planar or cylindrical (parabolic) iso-surfaces, and volumetric regions with saddle-like (hyperbolic) iso-surfaces, respectively. In images in which only H is represented, the colors represent the square of the mean curvature where the sign of the mean curvature differentiates between red and blue. In the figures in which K and H are presented simultaneously, the color coding is provided, as seen in Fig. 2. Here, we employ the same convention that the colors represent the square of the mean curvature and the sign differentiates between the red-violet column and the white-blue one. Table 2 shows several interpretations for different combinations of K and H .

TABLE 2
Interpretations for Different Values of Combinations of Values of K and H

	$H < 0$	$H = 0$	$H > 0$
$K < 0$	hyperbolic saddle-like	hyperbolic - minimal surface	hyperbolic saddle-like
$K = 0$	concave parabolic	planar - minimal surface	convex parabolic
$K > 0$	concave elliptic	impossible	convex elliptic

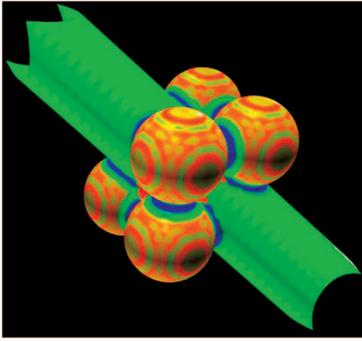


Fig. 3. A synthetic volumetric image of a cylinder and six spheres. Red, green, and blue regions represent volumetric regions with elliptic, parabolic, and hyperbolic iso-surfaces, respectively, having positive, zero, and negative Gaussian curvature values. The colors follow the coding described in Fig. 1.

The images are presented in two sections of this paper. The first, Section 6.1, includes synthetically generated images and, in the second, Section 6.2, we include scanned volumetric and 3D range data.

In Figs. 8, 9, and 12, the colors in parts a and b represent the values of the Gaussian and mean curvatures, respectively,

and follow the coding described in Fig. 1. The colors in part c follow the coding described in Fig. 2 and Table 2.

6.1 Synthetic Images

Fig. 3 is a synthetic volumetric image of a cylinder and six spheres where the Gaussian curvature of a certain pre-selected constant iso-level was computed and colored using this new curvature analysis scheme. The trivariate volumetric image has $40 \times 40 \times 40$ coefficients. It took 25 minutes and 3.5 minutes to compute the numerator and denominator of K , respectively.

In [17], the authors present a data set that contains several objects that have similar sizes and shapes to the objects in Fig. 3. The time required for computation of one iso-level is similar in [17] to our scheme. It is around 30 minutes.

Fig. 4 shows an image of two cylinders with a sphere. The segmentation scheme produces two cylinders. This example shows the applicability of segmentation to detect objects that have cylindrical form. The curvatures in this image were computed by interpolating over Chebyshev (see Section 4.1.2) points. The trivariate volumetric image has $40 \times 40 \times 40$ coefficients.

Fig. 5 shows an image of several cylinders. Figs. 5a and 5b show different cylinders detected as different iso-levels

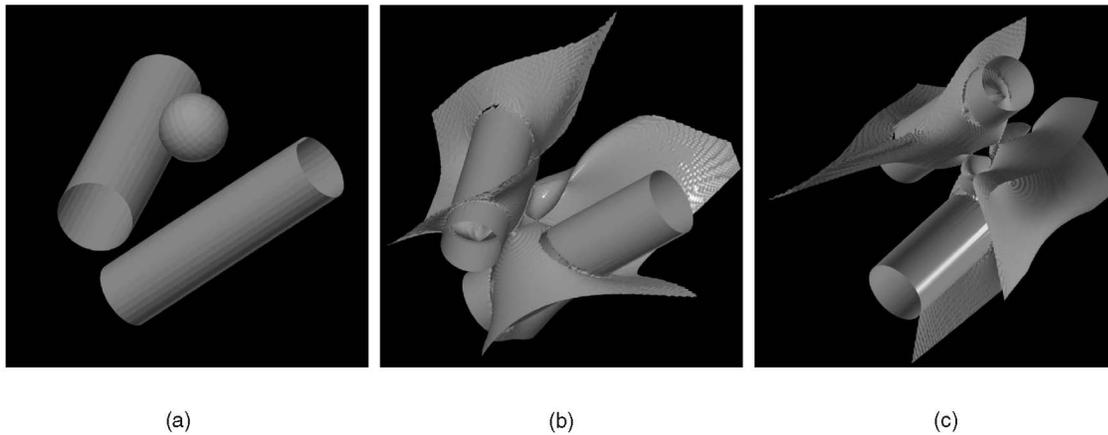


Fig. 4. A volumetric image of a synthetically generated image of two cylinders and a sphere. In (a), an iso-level constant of f is shown. In (b) and (c), the iso-level of $K = 0$ is shown. (b) and (c) represent the same image viewed from different points of view. (b) and (c) show the selection of two cylinders and underline the applicability of volumetric segmentation in primitives detection applications.

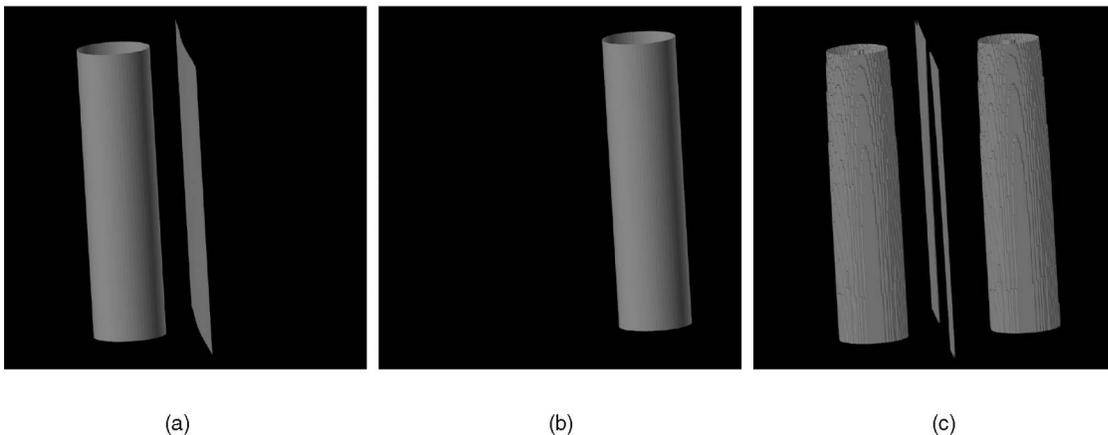


Fig. 5. A volumetric image of a synthetically generated image of two cylinders. In (a) and (b), different iso-levels constants of f reveal different cylinders. In (c), the iso-level of $K = 0$ is shown. (c) shows the detection of both cylinders in the volumetric image regardless of their iso-levels in f and underlines the applicability of volumetric segmentation in primitives detection applications.

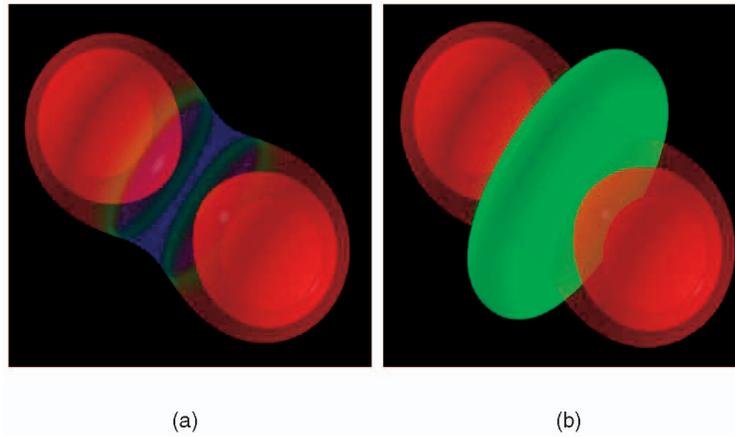


Fig. 6. A synthetic volumetric image of two centers of potential field. Red, green, and blue regions represent volumetric regions with elliptic, parabolic, and hyperbolic iso-surfaces, respectively, having positive, zero, and negative Gaussian curvature values. The colors follow the coding described in Fig. 1. (b) consists of the volumetric image shown in (a) where the constant iso-surface of $K = 0$ was superimposed. The iso-surface of $K = 0$ is colored in green. This iso-surface globally prescribes the parabolic regions of all iso-surfaces, simultaneously.

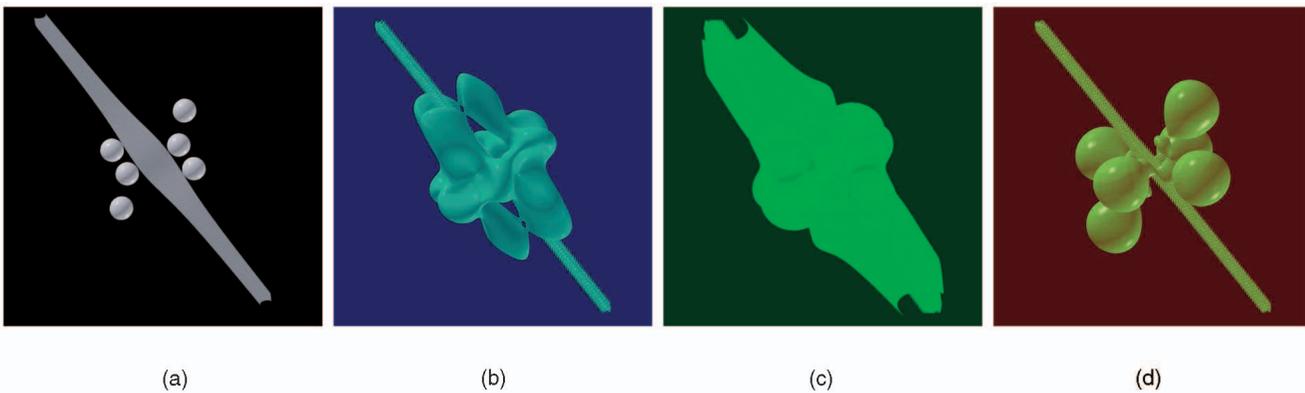


Fig. 7. A synthetic volumetric model of a physical potential field generated by an axis and six centers. (a) is an iso-constant level of f while (b), (c), and (d) are constant K surfaces generated by the potential field distribution of which one iso-constant level is shown in (a). Red, green, and blue regions represent volumetric regions with elliptic, parabolic, and hyperbolic iso-surfaces, respectively, having positive, zero, and negative Gaussian curvature values. The colors follow the coding described in Fig. 1.

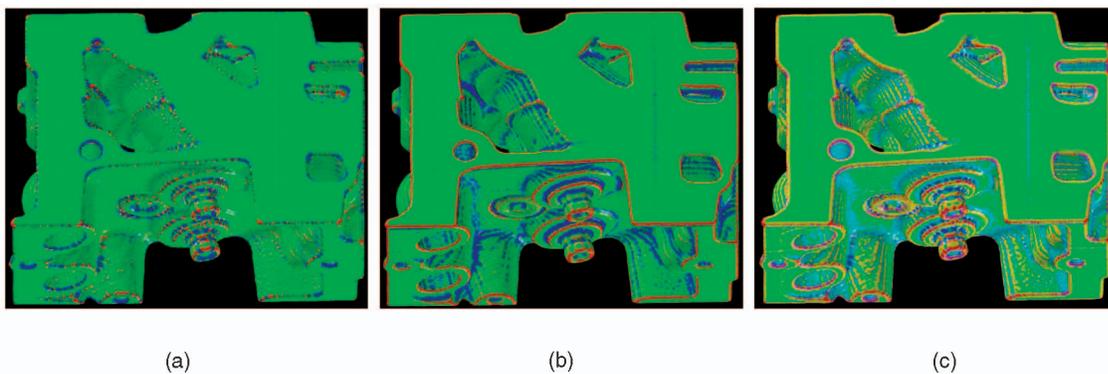


Fig. 8. A volumetric image of an engine block. In (a), red, green, and blue regions represent volumetric regions with elliptic, parabolic, and hyperbolic iso-surfaces, respectively, having positive, zero, and negative Gaussian curvature values. In (b), red, green, and blue regions represent regions with preponderant positive, zero, and negative mean curvature values. The colors represent the square of the mean curvature where the sign of the mean curvature differentiates between red and blue. In (c), the image is segmented with eight colors; each one represents a region classified after the values of K and H ; for example, red means both $K > 0$ and $H > 0$, and green means $K = H = 0$. The colors in (a) and (b) represent the values of the Gaussian and mean curvatures, respectively, and follow the coding described in Fig. 1. The colors in (c) follow the coding described in Fig. 2 and Table 2.

constants of f . Fig. 5c shows the iso-level of $K = 0$. It also shows the detection of both cylinders in the volumetric image regardless of their iso-levels in f and underlines the applicability of volumetric segmentation in primitives detection applications. This example, similarly to Fig. 4, shows the applicability of segmentation to detect primitives that have

cylindrical form. The curvatures in this image were computed interpolating over Chebyshev (see Section 4.1.2 points). The trivariate volumetric image has $80 \times 80 \times 80$ coefficients.

Fig. 6a is a synthetic volumetric image of two centers of a potential field of a nonphysical model, where the Gaussian curvatures of three different preselected constant iso-levels

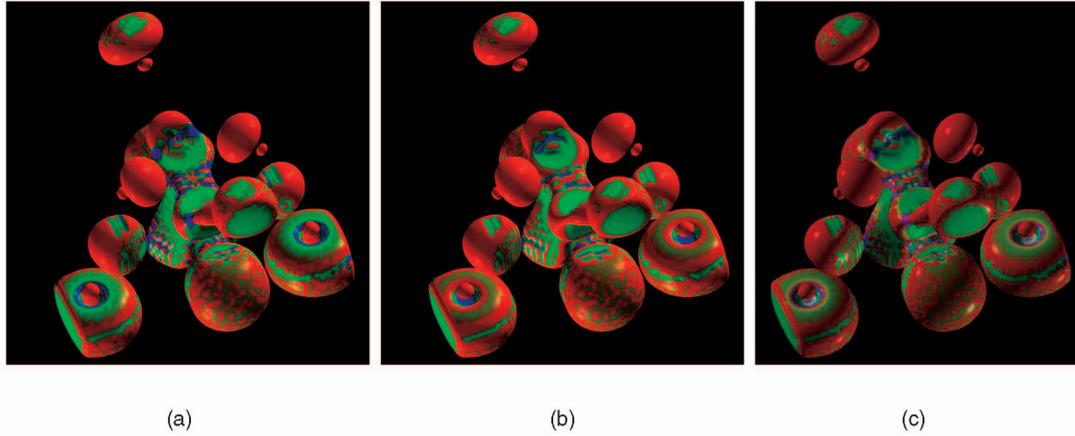


Fig. 9. A volumetric image of an iron protein molecule. The colors in (a) and (b) represent the values of the Gaussian and mean curvatures, respectively, and follow the coding described in Fig. 1. The colors in (c) follow the coding described in Fig. 2 and Table 2.

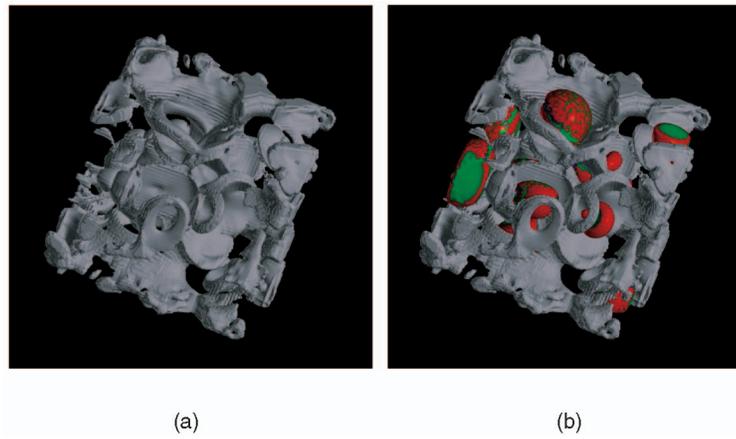


Fig. 10. (a) represents an iso-surface at level zero of K of the iron protein in Fig. 9a. This iso-surface globally fixes the parabolic regions of all iso-surfaces, simultaneously. (b) consists of the volumetric image shown in (a), with the constant iso-level of f from Fig. 9a superimposed on it.

were computed and colored using this new curvature analysis scheme. Let the centers of this potential field be C_1 and C_2 . Then, $f(P) = \|P - C_1\| \cdot \|P - C_2\|$ for any $P = (u, v, w)$, which is a point in the volumetric domain. The trivariate volumetric image has $40 \times 40 \times 40$ coefficients. Fig. 6b consists of the volumetric image shown in Fig. 6a with the constant iso-surface of $K = 0$ superimposed on it. The iso-surface of $K = 0$ is colored in green. This ability to robustly derive the zero set of K , or any other differential form, is a direct consequence of our global rational representation of $K(u, v, w)$ for the given function $f(u, v, w)$, either in a one B-spline form or in piecewise Bézier form. It took 18 and two hours to compute the numerator and denominator of K , respectively. While the peak memory consumption was around 100 megabytes of memory at the initialization stages, less than 20 megabytes of memory were required during the effective computation.

Fig. 7a represents an iso-constant level of a volumetric model of a physical potential field generated by an axis and six centers. Let $C_i, i \in \{1..6\}$ be the six centers of a gravitational potential field. The axis generates the gravitational potential field $f_0(P) = \frac{100}{\text{distance from } P \text{ to axis}}$, and each center contributes the gravitational potential field $f_i(P) = \frac{100}{\|P - C_i\|}$, where $i \in \{1..6\}$ and for any $P = (u, v, w)$, which is a point in the volumetric domain. Here, $f(u, v, w) = f_0(u, v, w) + \sum_{i=1}^6$

$f_i(u, v, w)$. Figs. 7b, 7c, and 7d, are volumetric images of the constant iso-surfaces of $K = -0.00125, 0.0$, and 0.00125 , respectively, computed for $f(u, v, w)$. The trivariate volumetric image has $120 \times 120 \times 120$ coefficients. It took 166 and 19 hours to compute the numerator and denominator of K , respectively. While the peak memory consumption is around 100 megabytes of memory at the initialization stages, less than 20 megabytes of memory were required during the effective computation.

6.2 Real Volumetric and 3D Range Images

Fig. 8 portrays three volumetric images of an engine block (see [40] for details) where the Gaussian and square of the mean curvatures of a certain preselected constant iso-level were computed using this new scheme. In Fig. 8a, the colors are computed using the Gaussian curvature. In Fig. 8b, the colors are computed using the mean curvature. Fig. 8c represents a combination of the first two segmentations where eight kinds of regions are detected, following the coding described in Fig. 2 and Table 2. The trivariate volumetric images of this engine block have $256 \times 256 \times 110$ coefficients. It took 11 and a half, and seven and a half hours to compute the numerator and denominator of K , respectively. Around 100 megabytes



Fig. 11. A digital camera photo of several objects on a table.

of memory were required to compute and analyze the volume using the Gaussian curvature.

Fig. 9 portrays three volumetric images of an iron protein molecule (see [37] for details) where the Gaussian and square of the mean curvatures of a certain pre-selected constant iso-level were computed using this new scheme. In Fig. 9a, the colors are computed using the Gaussian curvature. In Fig. 9b, the colors are computed using the mean curvature. Fig. 9c represents a combination of the first two segmentations where eight kinds of regions are detected, following the coding described in Fig. 2 and Table 2. The trivariate volumetric image has $68 \times 68 \times 68$ coefficients. The computation of the numerator of K required one and a half hours while its denominator computation took 16 minutes. The computation of the numerator of H necessitated 41 minutes. The computation of the numerator of H^2 required 4 hours and 45 minutes while its denominator's computation took 48 minutes. Around 120 and 150 megabytes of peak memory were detected when computing and analyzing the volume using the Gaussian and mean curvatures, respectively.

Fig. 10a shows the zero level set of $K(u, v, w) = 0$, of the iron protein model shown in Fig. 9. This ability to robustly derive the zero set of K , or any other differential form, is a direct consequence of our global rational representation of $K(u, v, w)$ for the given function $f(u, v, w)$, either in a one B-spline form or in piecewise Bézier form. Fig. 10b consists of the volumetric image shown in (a) but with the constant iso-level of f from Fig. 9a superimposed on it. The computation of the numerator of K required 32 hours while its denominator's computation took 4 hours.

Fig. 11 is a digital photo of several objects on a table. Fig. 12 shows the scanned images of the objects that appear in Fig. 11. The objects were scanned using a 3D Cyberware scanner [1] and the Gaussian and square of the mean curvatures of a certain preselected constant iso-level were computed using our new scheme. In Fig. 12a, the colors are computed using the Gaussian curvature. In Fig. 12b, the colors are computed using the mean curvature. Fig. 12c represents a combination of the first two segmentations where eight kinds of regions are detected, following the coding described in Fig. 2 and Table 2. Figs. 12a, 12b, and 12c were obtained from a range image on which we ran a Gaussian filter before computing curvatures.

7 CONCLUSIONS

In this work, we have presented a scheme to globally segment and derive curvature properties of volumetric data sets. The scheme is global, immune to aliasing, and capable of detecting curvature properties regardless of iso-level values. We map a given scalar field $f(u, v, w)$ to other differential scalar fields such as $K(u, v, w)$ and $H(u, v, w)$. The scalar fields $K(u, v, w)$ and $H^2(u, v, w)$ are used to separate convex or concave (elliptic) iso-surfaces, planar or cylindrical (parabolic) iso-surfaces, and volumetric regions with saddle-like (hyperbolic) iso-surfaces, regardless of the value of the iso-surface level. The proposed scheme enables the identification of eight kinds of regions (the case $K > 0$ and $H = 0$ is impossible), and this allows a more reliable identification of primitive parts.

We performed a large variety of experiments on 3D as well as volumetric data. The accuracy of the segmentation can be evaluated in the images presented in Section 4.

In Section 6, we also provided insight about the time and memory consumption required by our system. The main time consuming steps are the evaluations required at the interpolation stages, which represent more than 90 percent of the overall time computation. Speeding up the algorithm execution is a future objective. While the peak memory consumption is around 100 megabytes of memory at the initialization stages, less than 20 megabytes of memory are required during the effective computation of each example. The memory consumption is reasonable on today's computers.

Although the problem has a continuous nature, as in many other volumetric processing problems, the proposed

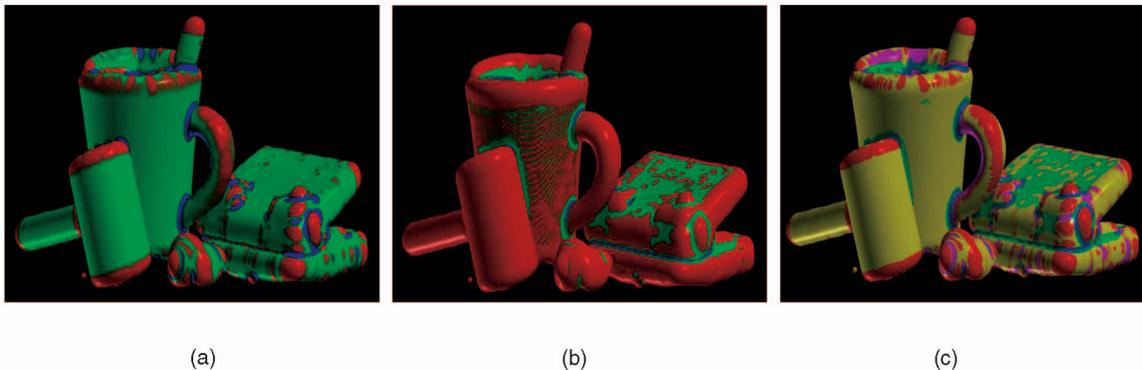


Fig. 12. Three curvature colored images of several objects on a table that appear in Fig. 11. The colors in (a) and (b) represent the values of the Gaussian and mean curvatures, respectively, and follow the coding described in Fig. 1. The colors in (c) follow the coding described in Fig. 2 and Table 2.

solution is clearly parallelizable. We believe that employing concurrent or parallel variants of the algorithms, and/or implementing the curvature evaluation schemes on dedicated hardware could greatly speed up this process.

ACKNOWLEDGMENTS

The engine block's volumetric data set is part of the VolVis [40] public domain library and the iron protein model is part of the public data set of VTK [37]. The first author thanks Emil Dan Kohn and Yekutiel Pekar for their technical advice related to various programming tasks in this research and Professor Avraham Sidi for several observations focused on the Kronecker product. This work was supported in part by the Fund for Promotion of Research at the Technion, Haifa, Israel, in part by the Bar-Nir Bergreen Software Technology Center of Excellence, in part by US National Science Foundation grant IIS0218809, and in part through a donation from IBM's Shared University Research (SUR—<http://www-3.ibm.com/software/info/university/sur/>) program to the Technion in 2003. All opinions, findings, conclusions, or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

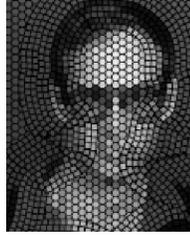
REFERENCES

- [1] <http://www.cyberware.com>, 2005.
- [2] *Maple and C Code of Gaussian and Mean Curvature Computation of Volumetric Data Sets*. http://www.cs.technion.ac.il/octavian/gaussian_and_mean_curvature_vol, 2006.
- [3] P.J. Besl and R.C. Jain, "Segmentation Through Variable-Order Surface Fitting," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 2, pp. 167-192, Mar. 1988.
- [4] R.N. Bracewell, *The Fourier Transform and Its Applications*, second ed. McGraw-Hill, 1986.
- [5] E. Cohen, R.F. Riesenfeld, and G. Elber, *Geometric Modeling with Splines: An Introduction*, A.K. Peters, ed. July 2001.
- [6] R.M. Corless, *Essential Maple 7. An Introduction for Scientific Programmers*, second ed. Springer, 2002, www.maplesoft.com.
- [7] G. Dahlquist and Å Björck, *Numerical Methods*. Prentice-Hall, 1974.
- [8] F. Dell'Acqua and R. Fisher, "Reconstruction of Planar Surfaces Behind Occlusions in Range Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, Apr. 2002.
- [9] S.J. Dickinson, D. Metaxas, and A. Pentland, "The Role of Model-Based Segmentation in the Recovery of Volumetric Parts from Range Data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 259-267, Mar. 1997.
- [10] G. Elber, *Irit Version 9.0*. <http://www.cs.technion.ac.il/irit>, 2006.
- [11] G. Elber and M.-S. Kim, "Geometric Constraint Solver Using Multivariate Spline Functions," *Proc. Sixth ACM Symp. Solid Modeling and Computer Graphics*, pp. 171-175, Oct. 2001.
- [12] G. Far, *Curves and Surfaces for Computer Aided Geometric Design, A Practical Guide*, fourth ed. Academic Press, 1996.
- [13] A.W. Fitzgibbon, D.W. Eggert, and R.B. Fisher, "High-Level CAD Model Acquisition from Range Images," *Computer Aided Design*, vol. 29, no. 4, Apr. 1997.
- [14] P.J. Flynn and A.K. Jain, "On Reliable Curvature Estimation," *IEEE Proc. Conf. Computer Vision and Pattern Recognition*, 1989.
- [15] G.H. Golub and C.F. Van Loan, *Matrix Computations*, third ed., The Johns Hopkins Univ. Press, pp. 180-181, 1996.
- [16] B. Hamann, "Visualization and Modeling Contours of Trivariate Functions," PhD thesis, Arizona State Univ., 1991.
- [17] J. Hladuvka, A. König, and E. Gröller, "Curvature-Based Transfer Functions for Direct Volume Rendering," *Proc. Spring Conf. Computer Graphics and Its Applications (SCCG 2000)*, pp. 58-65, 2000.
- [18] R. Hoffman and A.K. Jain, "Segmentation and Classification of Range Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, Sept. 1987.
- [19] C.M. Hoffmann and R. Joan-Arinyo, "A Brief on Constraint Solving," *Computer-Aided Design and Applications*, vol. 2, no. 5, June 2005.
- [20] A. Hoover, G. Jean-Baptiste, X. Jiang, P.J. Flynn, H. Bumke, D.B. Goldof, K. Bowyer, D.W. Eggert, A. Fitzgibbon, and R.B. Fisher, "An Experimental Comparison of Range Image Segmentation Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 673-689, July 1996.
- [21] R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge Univ. Press. p. 336, pp. 338-339, 1985.
- [22] V. Interrante, "Illustrating Surface Shape in Volume Data via Principal Direction-Driven 3D Line Integral Convolution," *Computer Graphics (Proc. SIGGRAPH 97)*, pp. 109-116, 1997.
- [23] T. Itoh and K. Koyamada, "Automatic Isosurface Propagation Using an Extrema Graph and Sorted Boundary Cell Lists," *IEEE Trans. Visualization and Computer Graphics*, vol. 1, no. 4, pp. 319-327, Dec. 1995.
- [24] I. Kaufman, "The Inversion of the Vandermonde Matrix and the Transformation to the Jordan Canonical Form," *IEEE Trans. Automatic Control*, vol. 14, Dec. 1999.
- [25] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Möller, "Curvature-Based Transfer Functions for Direct Volume Rendering: Methods and Applications," *IEEE Visualization*, Oct. 2003.
- [26] R. Klette and A. Rosenfeld, *Digital Geometry—Geometric Methods for Digital Picture Analysis*. Morgan Kaufmann, 2004.
- [27] S. Lakare, M. Wan, M. Sato, and A. Kaufman, "3D Digital Cleansing Using Segmentation Rays," *Proc. IEEE Visualization Ann. Conf.*, pp. 37-44, 2000.
- [28] W.E. Lorensen and H.E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics (Proc. SIGGRAPH 87)*, vol. 21, no. 4, pp. 163-169, July 1987.
- [29] D. Marshall, G. Lukacs, and R. Martin, "Robust Segmentation of Primitives from Range Data in the Presence of Geometric Degeneracy," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, Mar. 2001.
- [30] T. McInerney and D. Terzopoulos, "A Dynamic Finite Element Surface Model for Segmentation and Tracking in Multidimensional Medical Images with Application to Cardiac 4D Image Analysis," *Computerized Medical Imaging and Graphics*, vol. 19, no. 1, Jan. 1995.
- [31] M. Spivak, *A Comprehensive Introduction to Differential Geometry*. Publish or Perish, vol. I-V, 1970-1975.
- [32] M. Sutton, L. Stark, and K. Bowyer, "Function from Visual Analysis and Physical Interaction: A Methodology for Recognition of Generic Classes of Objects," *Image and Vision Computing*, vol. 16, no. 11, pp. 745-763, 1998.
- [33] C.-K. Tang and G. Medioni, "Curvature-Augmented Tensor Voting for Shape Inference from Noisy 3D Data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, June 2002.
- [34] G. Taubin, "Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation," *Proc. Fifth Int'l Conf. Computer Vision*, pp. 902-907, June 1995.
- [35] J.-P. Thirion and A. Gourdon, "The 3D Marching Lines Algorithm and its Application to Crest Lines Extraction," *Rapports de Recherche*, No. 1672, Programme 4, Robotique, Image et Vision, Unité de Recherche Inria-Rocquencourt, Inria, May 1992.
- [36] J.-P. Thirion and A. Gourdon, "Intrinsic Surface Properties from Surface Triangulation," *Differential Characteristics of Isointensity Surfaces, Computer Vision and Image Understanding*, vol. 61, no. 2, pp. 190-202, Mar. 1995.
- [37] Visualization Toolkit, <http://www.vtk.org>, 2005.
- [38] E. Trucco and R.B. Fisher, "Experiments in Curvature-Based Segmentation of Range Data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, Feb. 1995.
- [39] B.C. Vemuri, A. Mitiche, and J.K. Aggarwal, "Curvature-Based Representation of Objects from Range Data," *Image and Vision Computing*, vol. 4, no. 2, pp. 107-114, May 1986.
- [40] Volume Visualization, <http://www.volvis.org>, 2005.
- [41] M.A. Wani and B.G. Batchelor, "Edge-Region-Based Segmentation of Range Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 3, pp. 314-319, Mar. 1994.
- [42] T. Weinkauff and H. Theisel, "Curvature Measures of 3D Vector Fields and Their Applications," *J. Int'l Conf. Central Europe on Computer Graphics, Visualization, and Computer Vision*, no. 10, pp. 507-514, Feb. 2002.

- [43] D. Weinshall, "Direct Computation of Qualitative 3D Shape and Motion Invariants," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 12, pp. 1236-1240, Dec. 1991.
- [44] H.J. Wertz, "On the Numerical Inversion of the Recurrent Problem: the Vandermonde Matrix," *IEEE Trans. Automatic Control*, vol. 10, Oct. 1965.
- [45] S.M. Yamany and A.A. Farag, "Surface Signatures: An Orientation Independent Free-Form Surface Representation Scheme for the Purpose of Objects Registration and Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, Aug. 2002.
- [46] E. Zuckerberger, A. Tal, and S. Shlafman, "Polyhedral Surface Decomposition with Applications," *Computer & Graphics*, vol. 26, no. 5, pp. 733-743, Oct. 2002.



Octavian Soldea received the BSc degree (1993) from the Alexandru Ioan Cuza University of Iasi, Romania, and the MSc degree (1999) from the Technion, Israel Institute of Technology, both in computer science. He is a PhD student in the Computer Science Department at the Technion, Israel Institute of Technology. His research interests include computer vision, computer-aided geometric design, and continued fractions.



Gershon Elber received the BS degree in computer engineering and the MS degree in computer science from the Technion, Israel, in 1986 and 1987, respectively, and the PhD degree in computer science from the University of Utah, in 1992. He is a professor in the Computer Science Department, Technion, Israel. His research interests span computer-aided geometric designs and computer graphics. He is a member of the ACM and IEEE. He serves on the editorial board of *Computer Aided Design*, *Computer Graphics Forum*, and the *International Journal of Computational Geometry and Applications* and has served on many conference program committees including solid modeling, Pacific Graphics, Computer Graphics International, and Siggraph. He was one of the paper chairs of solid modeling 2003 and solid modeling 2004. He is a member of the IEEE.



Ehud Rivlin received the BSc and MSc degrees in computer science and the MBA degree from the Hebrew University in Jerusalem, and the PhD degree from the University of Maryland. Currently, he is an associate professor in the Computer Science Department at the Technion, Israel Institute of Technology. His current research interests are in machine vision and robot navigation. He is a member of the IEEE.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.