First Order Augmentation to Tensor Voting for Boundary Inference and Multiscale Analysis in 3D

Wai-Shun Tong, *Student Member*, *IEEE Computer Society*,
Chi-Keung Tang, *Member*, *IEEE Computer Society*,
Philippos Mordohai, *Student Member*, *IEEE*, and Gérard Medioni, *Fellow*, *IEEE*

Abstract—Most computer vision applications require the reliable detection of boundaries. In the presence of outliers, missing data, orientation discontinuities, and occlusion, this problem is particularly challenging. We propose to address it by complementing the tensor voting framework, which was limited to second order properties, with first order representation and voting. First order voting fields and a mechanism to vote for 3D surface and volume boundaries and curve endpoints in 3D are defined. Boundary inference is also useful for a second difficult problem in grouping, namely, automatic scale selection. We propose an algorithm that automatically infers the smallest scale that can preserve the finest details. Our algorithm then proceeds with progressively larger scales to ensure continuity where it has not been achieved. Therefore, the proposed approach does not oversmooth features or delay the handling of boundaries and discontinuities until model misfit occurs. The interaction of smooth features, boundaries, and outliers is accommodated by the unified representation, making possible the perceptual organization of data in curves, surfaces, volumes, and their boundaries simultaneously. We present results on a variety of data sets to show the efficacy of the improved formalism.

Index Terms—Tensor voting, first order voting, boundary inference, discontinuities, multiscale analysis, 3D perceptual organization.

1 Introduction

We address two complex issues that often arise in perceptual organization problems; boundary inference and multiple-scale processing. The augmentation to the original tensor voting framework [1] proposed here enables us to simultaneously detect and extract curves, surfaces, and volumes along with their terminations in 3D, even under severe noise corruption. Since our method is model-free, the treatment of arbitrary curves, surfaces, and volumes is not harder or more computationally expensive than that of lines, planes, and rectangular regions. Furthermore, we present a scheme for multiscale analysis of the data that is founded on our novel boundary detection technique. This paper completes the compact description of the topic, which appears in [2], by providing full details for all cases of boundary inference and more illustrative examples, as well as a more detailed analysis of the multiple scale scheme.

The fundamental problem we address in this line of research is the development of a methodology for the perceptual organization of *tokens*. The tokens represent the position of elements such as points, curvels (curve elements), or surfels (surface elements) and can also convey other information, such as curve or surface orientation. Token generation is application specific and is not described in detail here. Tokens can be generated by processes that detect the presence of certain features, such as edges, corners, pixel correspondences, surface patches, intensity within certain thresholds, etc. Perceptual organization is achieved by enforcing constraints, as suggested by Gestalt psychology [3]. The inferred descriptions are in terms of junctions, curves, surfaces, volumes, and their boundaries. These structures are represented by grouped tokens that bear local estimates of the structure's orientation.

1.1 Boundary Inference

The first contribution of this paper deals with the fundamental smoothness versus discontinuities dilemma that occurs in most nontrivial perceptual organization scenarios. Many perceptual organization approaches operate either as grouping or as segmentation processes. We believe that both grouping ***and*** segmentation must be performed in order to tackle challenging problems. In both cases, boundaries play a critical part. Our strictly second order formalism [1] can be viewed as an excitatory process that facilitates grouping of the input data and is able to extrapolate and extract dense salient structures. The integration of boundary inference, via first order voting, provides a mechanism to inhibit the growth of the extracted structures. The addition of polarity vectors (first order tensors) to the representation complements the previously published second order representation that was insufficient for encoding first order properties, such as boundaries of perceptual structures. The new representation exploits the essential property of boundaries to have all their neighbors, at least locally, on the same side of a half-space. As described in the remainder of the paper, the voting scheme is identical to that of the second order case and the first order

- *W.-S. Tong and C.-K. Tang are with the Department of Computer Science, Hong Kong University of Science & Technology, Clear Water Bay, Hong Kong. E-mail: {cstws, cktang}@cs.ust.hk.*
- *P. Mordohai and G. Medioni are with the Institute for Robotics and Intelligent Systems, University of Southern California, Los Angeles, CA 90083-0273. E-mail: {mordohai, medioni}@iris.usc.edu.*

*Manuscript received 16 Dec. 2002; revised 24 July 2003; accepted 13 Oct. 2003.
Recommended for acceptance by E.R. Hancock.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117965.*

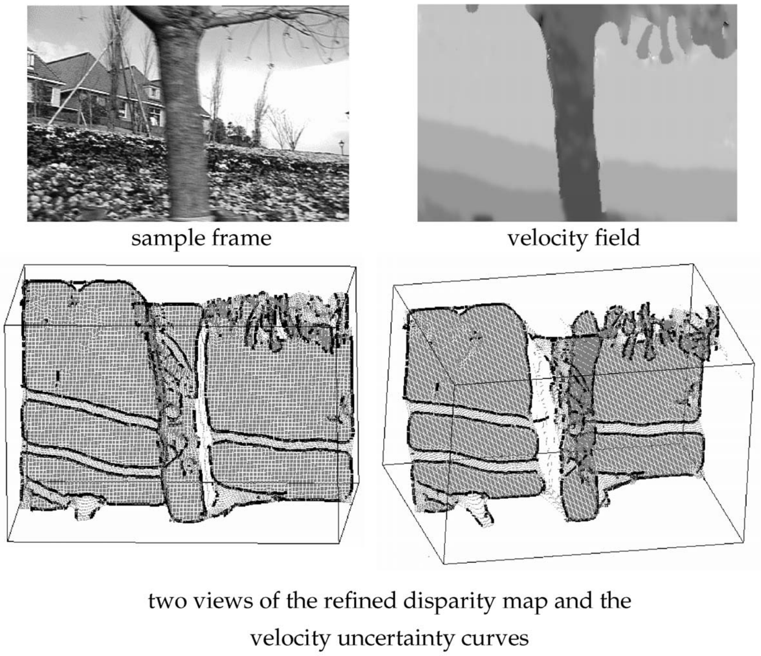

sample frame
velocity field
two views of the refined disparity map and the
velocity uncertainty curves

Fig. 1. Flower Garden. A sample frame and velocity field (x-component) are shown. (c) and (d) show the extracted discontinuity curves, together with the computed depth map.

vector voting fields can be easily derived from the second order fundamental tensor voting field.

Open surfaces in 3D often occur in stereo and other computer vision problems. For instance, in Fig. 1, the detected motion layers of the "flower garden" sequence (as computed by an algorithm such as [4]) are surfaces whose discontinuities convey important information. Our original framework fails to explicitly detect them, as illustrated in a simple 2D example in Fig. 2. Consider points A and A', which are smooth inliers of the contours. The second order tensors associated with these points are identical in terms of both saliency and orientation. The problem appears when comparing points B and C with B' and C'. The latter are inliers of the closed contour, while the former are the endpoints of the open contour. The second order tensor at B has identical orientation as the one at B'. Even though there is a difference in curve saliency since B receives less support from its neighborhood than B', the inferred description is very similar for two points which are qualitatively very different. This occurs because the second order representation is inadequate to capture the key property of endpoints: that all their neighbors in the contour are on the same side. The first order augmentation to the framework addresses this shortcoming by being sensitive to the direction from which votes are received.

The second order part of the representation encodes the preferred type of structure or structures of a token, the orientation of these structures and discontinuities in orientation. These can be viewed as second order discontinuities since the structure remains continuous but its orientation changes abruptly. For example, a contour is continuous at a corner, but its orientation is not. On the other hand, the first order part of the representation captures the directions from which votes are cast to a token or, in other words, the distribution of the token's neighbors in space. This information can be used to detect the terminations of structures, such as the boundaries of the surfaces of Figs. 1c and 1d.

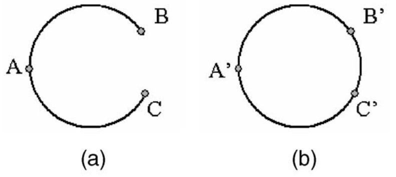


Fig. 2. (a) Open contour. (b) Closed contour. Open contours have endpoints where orientation still varies smoothly.

1.2 Multiscale Analysis

A second issue that often arises is the selection of the proper global scale for a data set, in case of single-scale methods, or the selection of the manner in which multiscale analysis should be performed. This usually occurs when dealing with data sets with varying density of data from region to region. The conflict is between the preservation of details, on one hand, and robustness to noise and completion of missing data, on the other. Consider Fig. 3a, where one can observe either a number of curve segments consisting of points in very close proximity or that these segments constitute a larger contour. Since human perception of this figure is not unique, but is a function of scale, an artificial perceptual grouping mechanism should also be able to derive the alternative groupings as the scale varies. At a small scale, the input can be grouped into the smaller segments and their endpoints can be detected as in Fig. 3b. Then, the gaps can be bridged using a larger scale, while leaving the already grouped points untouched (Fig. 3c). Alternatively, if the details and gaps of the contour are assumed to be due to noise, a large scale that guarantees smoothness and good continuation should be applied to the data set (Fig. 3d).

Motivated by these observations and scale-space theory [5], we present a multiscale implementation of our framework. We propose a data-driven, adaptive scheme where processing is performed at multiple scales according to local

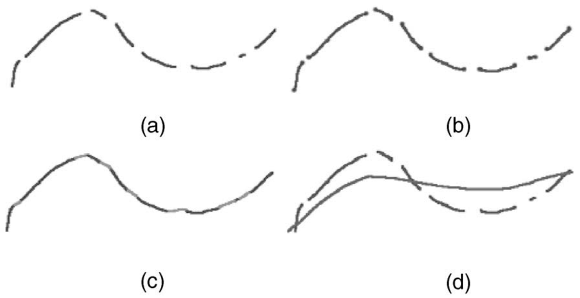


Fig. 3. Curve extraction at different scales. (a) Input data, (b) grouping at a small scale (endpoints in gray), (c) gaps are bridged, and (d) grouping at a large scale.

criteria. Our aim is to capture details where they exist and bridge gaps due to missing data without oversmoothing the rest of the input. We begin processing at a small scale and proceed with larger scales only in regions where discontinuities are detected. This hierarchical bottom-up scheme is consistent with preattentive human perception ([6], [7], [8], [9]) and offers many advantages over both symbolic and signal processing techniques ([10]). Once organization at a fine scale has been completed, we do not have to revisit these parts of the data set, thus avoiding unnecessary computations. In addition, since no convolutions with isotropic smoothing kernels are involved, features do not shift as scale increases. Accurate boundary detection is critical for the success of such a scheme since their presence is an indicator of potentially missing data and gaps that need to be bridged.

1.3 Organization of the Paper

This paper is organized as follows: In Section 2, we review related work. In Section 3, we present an overview of the second order tensor voting framework and introduce the first order representation, voting mechanism, and voting fields. In Section 4, the extraction of curve, surface, and region boundaries is described in detail and, in Section 5, the multiscale analysis of complex data sets is presented. Section 6 contains results on complex real three-dimensional data sets of various modalities. Finally, we discuss the contributions of this paper along with possible directions for future work in Section 7.

2 Related Work

Surface and curve inference from 3D data has been an active research area. Important issues include noise robustness, detection of orientation and depth discontinuities, and analysis at multiple scales. Since some of these issues have not been addressed in 3D, 2D research is also included here.

For surfaces represented in terms of energy functions, Terzopoulous and Metaxas [11] propose the *deformable surface model*. An initial shape is iteratively deformed until the surface parameters that maximize the fit are obtained. In [12], Sethian proposed a *level set* approach under which surfaces can be inferred as the zero-level iso-surface of a multivariate implicit function. The technique allows for topological changes, thus it can reconstruct surfaces of any genus as well as nonmanifolds. Osher et al. [13] and Osher and Fedkiw [14] proposed efficient ways of handling implicit surfaces as level sets of a function. A combination of points and elementary surfaces and curves can be provided as input to their technique which can handle local changes as well as global deformations and topological changes. The output however is limited to surfaces only. Lorigo et al. [44] extended the level set approaches in computer vision to codimension-two manifolds. Previously, level sets were limited to codimension-one manifolds, i.e., surfaces in 3D, while this generalization allows the inference of curves which are lower dimensional manifolds. All the implicit surface-based approaches are iterative and require careful selection of the implicit function and initialization. Furthermore, only one manifold type can be extracted at a time.

We now turn our attention to perceptual organization techniques that aim at grouping the primitives contained in the data set into perceptual structures. These primitives serve as tokens in a symbolic framework that aims at inferring meaningful groupings according to the Gestalt principles. More relevant to the work presented in this paper are methods that infer regions and their boundaries in 2D since, as pointed out in [15], there are not many perceptual organization methods that operate in 3D. Shashua and Ullman [16] first addressed the issue of structural saliency and how prominent curves are formed from tokens that are not salient in isolation. They define a locally connected network that assigns a saliency value to every image location according to the length and smoothness of curvature of curves going through that location. In [17], Parent and Zucker infer trace points and their curvature based on spatial integration of local information. An important aspect of this method is its robustness to noise. This work was extended to surface inference in three dimensions by Sander and Zucker [18]. Sarkar and Boyer [19] employ a voting scheme to detect a hierarchy of tokens. Unlike our voting scheme, voting in parameter space has to be performed separately for each type of feature, thus making the computational complexity prohibitive for generalization to 3D.

An important class of perceptual organization methods are inspired by human perception and research in psychophysiology and neuroscience. Grossberg and Mingolla and [20] and Grossberg and Todorovic [21] developed the *Boundary Contour System* and the *Feature Contour System* that can group fragmented and even illusory contours to form closed boundaries and regions by feature cooperation in a neural network. The cues they use are intensity edges and corners in images. Parallel and orthogonal grouping mechanisms allow the completion of regular and illusory contours. Heitger and von der Heydt [22], in a classic paper on neural contour processing, showed how elementary edges can be grouped into contours, including illusory ones, via convolution with a set of orientation selective kernels, whose responses decay with distance and difference in orientation. Both these approaches use terminations of image structures (edges and corners) as cues for contour formation, but do not explicitly detect the terminations of the contours being inferred. Williams and Jacobs [23] introduced the *stochastic completion fields* for contour grouping. Their theory is probabilistic and models the contour from a source to a sink as the motion of a particle performing a random walk. Particles decay after every step, thus minimizing the likelihood of completions that are not supported by the data or between distant points. Li [6] presented a contour integration model based on excitatory and inhibitory cells and a top-down feedback loop. What is more relevant to our research that focuses on the preattentive bottom-up process of perceptual grouping is that connection strength decreases with distance and that zero or low curvature alternatives are preferred to high curvature ones. The model for contour extraction of Yen and Finkel [24] is based on psychophysical and physiological evidence that has many similarities to ours. It employs a voting mechanism where votes, whose strength falls off as a Gaussian function of distance, are cast along a preferred orientation that is the tangent of the osculating circle. Review of perceptual grouping techniques based on cooperation and inhibition fields can be found in [25], [26]. It should be noted here that all these methods, unlike ours, require oriented inputs.

Based on scale-space representations for 1D signals [27], Mokhtarian [28] and Mokhtarian and Mackworth [29] proposed a scale-space representation for planar and three-dimensional curves. Special treatment for curvature singularities, even at straight lines, is necessary and the curves tend to shift toward the center of curvature as they are convolved with Gaussian kernels of increasing scale. Lowe [30] proposed a technique to compensate for this unwanted shrinking of the curves. Perona and Malik [31] used anisotropic diffusion instead of convolution to address this problem. Local differential properties are used to discourage interregion diffusion and facilitate intraregion diffusion. Lindeberg [5], [32] presented a framework for multiscale image analysis and automatic selection of scale. The appropriate scale at which each feature should be represented is found by selecting normalized measures of feature strength in the three-dimensional scale-space (x, y, σ). The problems of shrinkage and curvature singularities do not affect our approach that is symbolic, according to the computer vision paradigm proposed by Marr [33]. Along these lines, Saund [10], [34] has developed a framework that is symbolic, as opposed to the signal/image-based representation of previous multiscale methodologies. The major advantages gained by the symbolic representation are that features do not shift with scale, curvature discontinuities, and open curves do not pose additional difficulties and that features that exist in coarse scales do not affect adjacent features in fine scales and vice versa. Dolan and Riseman [35] also proposed a hierarchical fine-to-coarse scheme for the representation of curvilinear structure. Curve fragments are grouped by link filters according to proximity, angular compatibility, and continuation to form multilevel descriptions of the input.

Our method relates to the previous work in the following ways: The inputs can be oriented, unoriented, or a combination of both, while many of the techniques mentioned above require oriented inputs to proceed. Our representation is symbolic (in the sense defined in [10]) and, in addition to the advantages this brings, we are able to extract open and closed surfaces, curves, and junctions in 3D simultaneously. To our knowledge, the tensor voting framework is the only methodology that can represent and infer all possible types of structures in any dimension in the same space. Our voting function has many similarities with other voting-based methods, such as the decay with distance and curvature [22], [24], [6], and the use of constant curvature paths [17], [34], [19], [24] that result in an eight-shaped voting field (in 2D) [22], [24]. The major difference is that, in our case, the votes cast are tensors and not scalars, therefore, they are a lot richer in information. We apply multiple scale processing in a fine-to-coarse fashion, in accord with the majority of the multiscale techniques [33], [27], [36], [10], [35], [5]. Finally, as in the work of Lindeberg [5], [32], we detect each feature in as fine a scale as possible and then leave it unchanged as the scale becomes coarser.

3 The Augmented Tensor Voting Framework

In this section, we briefly review the original second order tensor voting framework [1] and show how it is augmented by first order representation and voting that enable us to detect discontinuities or structure terminations in the data. We begin by describing the representation, then illustrate the voting mechanism and introduce the concept of voting fields and how they are derived from the *2D second order fundamental stick field*. Finally, we briefly review the way dense structures such as surfaces and curves can be extracted from the sparse data. Pseudocode of the algorithms is available in [38] and the Appendix.

3.1 Representation by Polarity Vectors and Second Order Tensors

As mentioned in Section 1, we are interested in the perceptual organization of generic tokens. Each token represents the potential presence of a perceptual structure at its position. The representation of a token consists of a symmetric second order tensor that encodes *saliency* and a vector that encodes *polarity*. The representation is completed by the signs and directions of the principal curvatures that can be estimated as in [40]. Interested readers should refer to that paper for details since curvature estimation will not be described here. The tensor essentially indicates the saliency of each type of perceptual structure (surface, curve, or region in 3D) the token belongs to and its preferred normal and tangent orientations. The polarity vector, on the other hand, encodes the likelihood of the token being on the boundary of a perceptual structure.

In 3D, there are three possible types of inputs: unoriented, elementary curves, and elementary surfaces. Unoriented inputs are the most general case, have no preference of orientation, and the type of structure they may belong to can only be inferred based on the configuration of their neighbors. Their representation should be isotropic with respect to orientation. We choose to represent oriented tokens with their normals instead of their tangents. Therefore, an elementary surface patch is encoded with a tensor that is aligned with the patch's normal. A curve element, on the other hand, has a set of normals that span a 2D subspace in 3D. The curve's tangent is orthogonal to this subspace. Curvels are represented by a tensor that is orthogonal to its tangent, therefore belonging to the normal subspace. These tensors are presented in the following paragraphs. The first order representation for all cases is initialized to zero, since no a priori information is available.

A 3D, symmetric, nonnegative definite, second order tensor can be viewed as a 3×3 matrix or, equivalently, a 3D ellipsoid. Intuitively, its shape indicates the type of structure represented and its size the saliency of this information. The tensor can be decomposed as in the following equation:

$$\begin{aligned} T &= \lambda_1 \hat{e}_1 \hat{e}_1^T + \lambda_2 \hat{e}_2 \hat{e}_2^T + \lambda_3 \hat{e}_3 \hat{e}_3^T \\ &= (\lambda_1 - \lambda_2) \hat{e}_1 \hat{e}_1^T + (\lambda_2 - \lambda_3)(\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T) \\ &\quad + \lambda_3 (\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T + \hat{e}_3 \hat{e}_3^T), \end{aligned} \qquad (1)$$

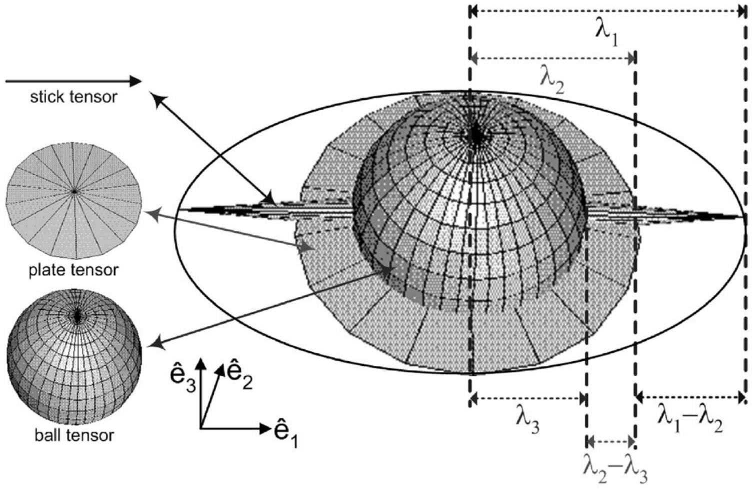


Fig. 4. Second order generic tensor and its decomposition into the *stick*, *plate*, and *ball* components in 3D.

where λ_i are the eigenvalues in decreasing order and $\hat{e}_i$ are the corresponding eigenvectors (see also Fig. 4). Note that the eigenvalues are nonnegative since the tensor is nonnegative definite. For instance, when $\lambda_1 = 1, \lambda_2 = \lambda_3 = 0$, only the first term in (1) remains, which corresponds to a degenerate elongated ellipsoid, termed hereafter the *stick tensor*, that represents an elementary surface token with $\hat{e}_1$ as its surface normal. When $\lambda_1 = \lambda_2 = 1, \lambda_3 = 0$, only the second term remains, which corresponds to a degenerate disk-shaped ellipsoid, termed hereafter the *plate tensor*, that represents a curve (or a surface intersection) with $\hat{e}_3$ as its tangent or, in other words, with $\hat{e}_1$ and $\hat{e}_2$ as the two normals that span the subspace of orientations that is orthogonal to the curve. Finally, when all eigenvalues are equal, only the third term remains, which corresponds to a sphere, termed the *ball tensor*, that corresponds to an unoriented token which can be a volume inlier or a junction. The size of the tensor indicates the certainty of the information represented by the tensor. A generic tensor can be decomposed as in (1) and each type of saliency can be evaluated. The size of the stick component $(\lambda_1 - \lambda_2)$ indicates surface saliency, the size of the plate component $(\lambda_2 - \lambda_3)$ indicates curve saliency, and that of the ball component (λ_3) junction or volume saliency. The interpretation of the inferred saliencies is described in Section 3.4.

The tensors can be initialized as balls with no preference of orientation or, if prior knowledge is available, with some preferred orientation. But, in general, after voting, a generic tensor comprising all three components will be the representation for each token. The benefit of having this representation is that the likelihood of the token belonging to each type of structure can be encoded simultaneously and carried throughout the processing stages without having to make premature hard decisions or maintain separate maps for every token type.

The advantages of using the above representation come at the cost of being insensitive to the direction from which the information is propagated to each token. As shown in Section 3.2, the second order votes are also symmetric, nonnegative definite, second order tensors. It leads to equal vote contributions at locations $\vec{u}$ and $-\vec{u}$ from a voter. The second order representation is adequate for representing tokens that belong to smooth structures or are located at orientation discontinuities; it fails, though, at structure discontinuities such as surface boundaries or curve endpoints since the surface boundaries, for instance, have a preferred normal that is a good continuation of the interior of the surface. What discriminates between the interior points of the surface and points on its boundaries is the fact that the former are surrounded by neighbors of the same surface, while the latter are not. The polarity vector is used to collect precisely this information since it is sensitive not only to the orientation of the vote, but also to the direction from which it is coming. Its magnitude is an indication of the likelihood of the token being on a boundary. A token on the boundary of a perceptual structure has a large polarity vector directed toward the majority of its neighboring tokens. On the other hand, a token in the interior of a surface or a curve has a locally nonmaximal polarity vector associated with it.

3.2 Tensor Voting

The core of our framework is the way information is propagated from token to token. The question we want to answer is: Assuming that a token at O with normal $\vec{N}$ and a token at P belong to the same smooth perceptual structure, what information should the token at O cast at P? We first answer the question for the 2D case of a voter with a pure *stick tensor* and show how all other cases can be derived from it. We claim that, in the absence of other information, the arc of the *osculating circle* at O that goes through P is the most likely smooth path since it maintains constant curvature. The osculating circle is the circle that shares the same normal as a curve at the given point. In case of straight continuation from O to P, the osculating circle degenerates to a straight line. It has enough degrees of freedom to connect two locations given only one orientation, that of the voter. Since voting is a pairwise operation, nothing suggests the use of smooth connections with varying curvature. Similar use of primitive circular arcs can also be found in [17], [34], [19].

As shown in Fig. 5, the second order vote is also a stick tensor and has a normal lying along the radius of the osculating circle at P, which has C as its center. The first order vote is a vector along the tangent of the same circle at P. What remains to be defined is the magnitude of these votes. Since nothing suggests the opposite, the first and second order votes should have the same magnitude and this should be a

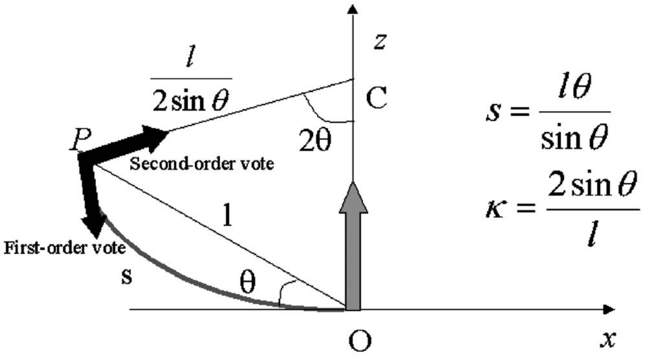


Fig. 5. Second and first order votes cast by a stick tensor located at the origin.

function of proximity and smooth continuation. The *saliency decay function* we have selected has the following form:

$$DF(s,\kappa,\sigma) = e^{-\left(\frac{s^2+c\kappa^2}{\sigma^2}\right)}, \quad (2)$$

where s is the arc length OP, κ is the curvature, c is a constant which controls the decay with high curvature[1] and σ is the scale of analysis, which determines the effective neighborhood size.[2] Note that σ is the only free parameter in the system. For a more detailed analysis on the selection of the smoothest paths and the saliency decay function, see [1].

In summary, the second order vote is a second order tensor that indicates the preferred orientation at the receiver according to the voter, while the first order vote is a first order tensor (a vector) that points toward the voter along the smooth path connecting the voter and receiver. Pseudocode for the voting functions is available in the Appendix. What should be noted is that both first and second order votes are generated by the second order part of the representation. This is because polarity vectors can only be initialized as zero vectors and because the second order tensor defines the type of structure and the way it should vote. The following equations define the second and first order votes cast by a unit stick tensor as functions of the parameters already defined.

$$\mathbf{S}_2(d,\theta) = DF(s,\kappa,\sigma)\begin{bmatrix} -sin(2\theta) \\ cos(2\theta) \end{bmatrix}[-sin(2\theta)cos(2\theta)]. \quad (3)$$

$$\mathbf{S}_1(d,\theta) = DF(s,\kappa,\sigma)\begin{bmatrix} -cos(2\theta) \\ -sin(2\theta) \end{bmatrix}. \quad (4)$$

A simple example to illustrate polarity is depicted in Fig. 6. The input consists of a set of coplanar unoriented tokens. They are encoded as ball tensors and cast first and second order votes to their neighbors. Figs. 6b and 6c show the maximum surface saliency and polarity at every (x,y) position. Cuts of these 2D maps can be seen in Figs. 6d and 6e. The surface saliency map indicates that there are salient surfaces formed by the tokens, but, since saliency is an excitatory process and its value drops gradually, detecting where the surface ends is not an easy task. However, when the surface saliency map is complemented by the polarity map, the surface boundaries can be detected as surface inliers that are also maxima of polarity along the direction of the polarity vectors.

3.3 Voting Fields

In this section, we will show how all the necessary votes can be cast in the same way as described in the previous section for the 2D stick tensor case and how all first and second order fields in any dimension can be derived. Finally, we will show how the votes cast by an arbitrary tensor can be computed given the voting fields.

The second order stick voting field $\mathbf{S}_2(P)$ is a second order tensor field. At every position, it contains a tensor that is the vote cast there by a unitary stick tensor located at the origin and aligned with the y axis. The shape of the field in 2D can be seen in Fig. 7a, which is generated by considering all P in the 2D space (Fig. 5). Depicted at every position is the eigenvector corresponding to the maximum eigenvalue of the second order tensor contained there. Its size is proportional to the magnitude of the vote. To compute a vote cast by an arbitrary stick tensor, we need to align the field with the orientation of the voter, and multiply the saliency of the vote that coincides with the receiver by the saliency of the arbitrary stick tensor, as in Fig. 7c. The same alignment holds for the first order case, where the voting fields are vector fields. Since the locations O and P and the unitary stick tensor define a plane in 3D, the generation of stick votes is identical in 2D, 3D, and ND. The stick voting fields in higher dimensions, therefore, can be derived by a simple rotation of the 2D stick field with respect to the axis aligned with the stick tensor. Specifically, note that a cut of the 3D stick voting field that contains the origin is identical to Fig. 5a since the voting stick tensor and the receiver define a plane in 3D where the voting takes place.

At the other end of the spectrum is the *ball voting field* $\mathbf{B}_2(P)$, a cut of which can be seen in Fig. 7b. The ball tensor has no preference of orientation, but still it can cast meaningful information to other locations. The presence of two proximate unoriented tokens, the voter and the receiver, indicates a potential perceptual structure. In the 3D case, this can either be a curve segment or a pencil of planes intersecting on that segment. Even though the voters are unoriented, surfaces can be inferred since the accumulation of votes from point to point with one degree of freedom in terms of surface orientation, from neighbors in the same surface, results in a high certainty for the correct surface normal and eliminates the degree of freedom. The case for curves is similar. The ball voting fields allow us to infer preferred orientations from unoriented tokens, thus minimizing initialization requirements.

To show the derivation of the ball voting fields from the stick voting fields, we can visualize the vote at P from a unitary ball tensor at the origin O as the integration of the votes of stick tensors that span the space of all possible orientations. In 3D, this can be simulated by a rotating stick tensor that spans the unit sphere. The 3D ball fields can be derived from the stick fields $\mathbf{S}_i(P)$, as follows: The first order ball field is derived from the first order stick field and the second order ball field from the second order stick field.

[1] In fact, c is a function of σ. Typical values of c range from 35.7 to 110.363 for σ ranging from 10 to 30. In our experiments, the results are not sensitive to the choice of c.

[2] Since we use a Gaussian decay function for $DF(\cdot)$, the effective neighborhood size is about 3σ. The derivation is as follows: let k be the effective neighborhood size. Then, we can set $e^{-\frac{k^2}{\sigma^2}} = \epsilon$, a small number, which is the lower bound of significant magnitude. We can then derive the effective neighborhood size k given σ.

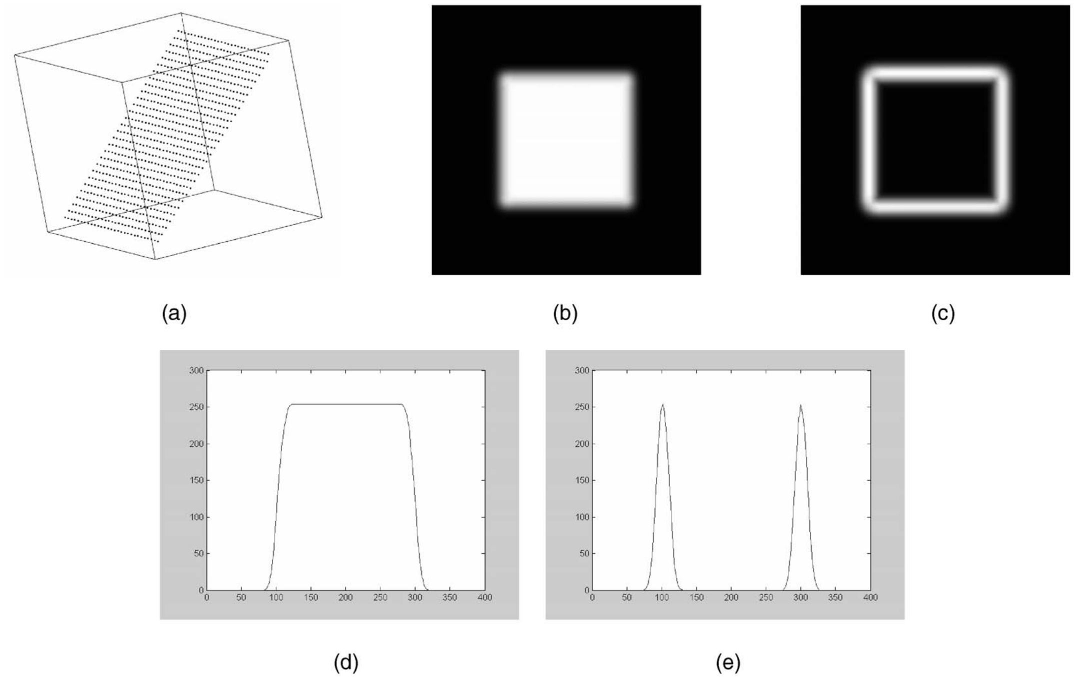


Fig. 6. (a) Input. (b) Surface saliency magnitude. (c) Polarity magnitude. (d) Cut of surface saliency. (e) Cut of polarity. Surface saliency and polarity for (a) a set of unoriented coplanar tokens. (b) Detection of the plane's boundaries using surface saliency alone is not clear. (c) Incorporating polarity information makes boundary detection straightforward. (d) and (e) are cuts of the surface saliency and polarity maps.

$$\mathbf{B_i}(P) = \int_0^\pi \int_0^\pi R^{-1}_{\theta\phi\gamma} \mathbf{S_i}(R_{\theta\phi\gamma}P) R^{-T}_{\theta\phi\gamma} d\phi d\gamma|_{\theta=0} \qquad i = 1, 2, \text{the order of the field,} \tag{5}$$

where $R_{\theta\phi\gamma}$ is the rotation matrix to align $\mathbf{S}_i$ with $\hat{e}_1$, the eigenvector corresponding to the maximum eigenvalue (the stick component) of the rotating tensor at P, and θ, ϕ, γ are rotation angles about the x, y, z axis, respectively.

In practice, the integration is approximated by a summation which, in the second order case, is performed as tensor addition, while, in the first order case, it is performed as plain vector addition. Normalization has to be performed in order to make the energy emitted by a unitary ball equal to that of a unitary stick. As a result of the integration, the second order ball field does not contain purely stick or purely ball tensors, but arbitrary second order symmetric tensors. The first order ball field holds a vector at each position as a result of vector addition. Both fields are radially symmetric, as expected, since the voter has no preferred orientation.

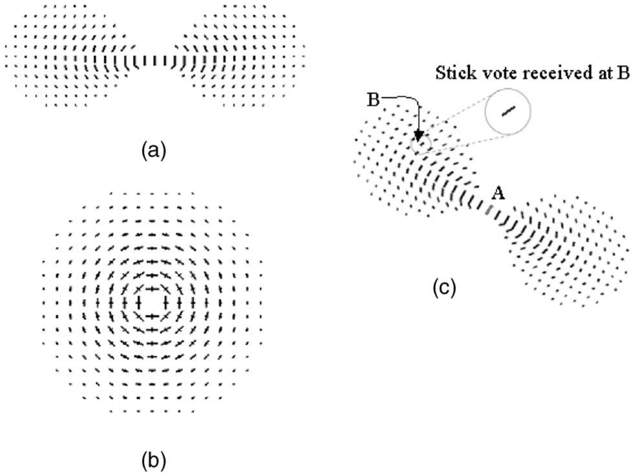


Fig. 7. (a) Two-dimensional stick voting field. (b) Two-dimensional ball voting field. (c) A casts a stick vote to B, using the 2D stick voting field. Due to symmetry, the orthographic projection of a slice of the 3D ball voting field viewed along the direction parallel to the $\hat{e}_3$ component of the tensor at the center of the field looks the same as (b). The fields are defined everywhere where the empty areas denote very small and negligible tensor votes.

To complete the description of the voting fields for the 3D case, we need to describe the *plate voting fields* $\mathbf{P}_i(P)$. Since the plate tensor encodes uncertainty of orientation around one axis, it can be derived by integrating the votes of a rotating stick tensor that spans the unit circle, in other words, the plate tensor. The formal derivation is analogous to that of the ball voting fields and can be written as follows:

$$\mathbf{P_i}(P) = \int_0^\pi R^{-1}_{\theta\phi\gamma} \mathbf{S_i}(R_{\theta\phi\gamma}P) R^{-T}_{\theta\phi\gamma} d\gamma|_{\theta=\phi=0} \qquad i = 1, 2, \text{the order of the field,} \tag{6}$$

where θ, ϕ, γ, and $R_{\theta\phi\gamma}$ have the same meaning as in the previous equation.

We have derived six voting fields for the 3D case, namely, the first and second order stick, plate, and ball fields. They are functions of the position of the receiver relative to the voter and a single parameter, the scale of the saliency decay function. After these fields have been precomputed at the desired resolution, computing the votes cast by any 3D second order tensor is reduced to a few look-up operations and linear interpolation. These fields are adequate for vote generation by any second-order nonnegative definite tensor. As described in Section 3.1, any 3D tensor can be decomposed into the stick, plate, and ball components, according to its eigensystem. Then, the corresponding fields can be aligned with each component. Votes are retrieved by simple look-up operations

TABLE 1
Summary of First and Second Order Tensor Structure for Each Feature Type in 3D

3-D Feature	Saliency	Second order tensor orientation	Polarity	Polarity vector
surface interior	high $\lambda_1 - \lambda_2$	normal: $\hat{e}_1$	low	-
surface end-curve	high $\lambda_1 - \lambda_2$	normal: $\hat{e}_1$	high	orthogonal to $\hat{e}_1$ and end-curve
curve interior	high $\lambda_2 - \lambda_3$	tangent: $\hat{e}_3$	low	-
curve endpoint	high $\lambda_2 - \lambda_3$	tangent: $\hat{e}_3$	high	parallel to $\hat{e}_3$
volume interior	high λ_3	-	low	-
volume boundary	high λ_3	-	high	normal to bounding surface
junction	locally max λ_3	-	low	-
outlier	low	-	low	-

and their magnitude is multiplied by the corresponding saliency. Recall from (1) that the saliency of the stick component is $\lambda_1 - \lambda_2$, of the plate component $\lambda_2 - \lambda_3$, and of the ball component λ_3.

3.4 Vote Collection and Interpretation

Votes are cast from token to token, as described in the previous section, and they are accumulated by tensor addition in the case of the second order votes, which are, in general, arbitrary second order tensors, and by vector addition in the case of the first order votes, which are vectors. Voting takes place in a finite neighborhood within which the magnitude of the votes cast remains significant.

Analysis of the second order votes can be performed once the eigensystem of the accumulated second order 3×3 tensor has been computed. Then, the tensor can be decomposed into the stick, plate, and ball components:

$$\begin{aligned} T =& (\lambda_1 - \lambda_2)\hat{e}_1\hat{e}_1^T + (\lambda_2 - \lambda_3)(\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T) \\ &+ \lambda_3(\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T + \hat{e}_3\hat{e}_3^T), \end{aligned} \tag{7}$$

where $\hat{e}_1\hat{e}_1^T$ is a *stick tensor*, $\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T$ is a *plate tensor*, $\hat{e}_1\hat{e}_1^T + \hat{e}_2\hat{e}_2^T + \hat{e}_3\hat{e}_3^T$ is a *ball tensor*. The following cases have to be considered: If $\lambda_1 \gg \lambda_2, \lambda_3$, this indicates a dominant stick component, thus a preference for a normal orientation and the token most likely belongs on a surface. In case of a token that belongs on a curve, or surface intersection, the uncertainty in normal orientation spans a plane perpendicular to the tangent. Hence, the inferred tensor is plate-like, that is, $\lambda_1 \approx \lambda_2 \gg \lambda_3$. If the token has no preference of orientation, $\lambda_1 \approx \lambda_2 \approx \lambda_3$ and the dominant component is the ball. Tokens that belong to volumes have high λ_3 values. Junctions can be discriminated from volume inliers since they are distinct local maxima of λ_3. That is, their λ_3 values are considerably larger than those of their neighbors. An outlier receives only inconsistent votes, so all eigenvalues are small.

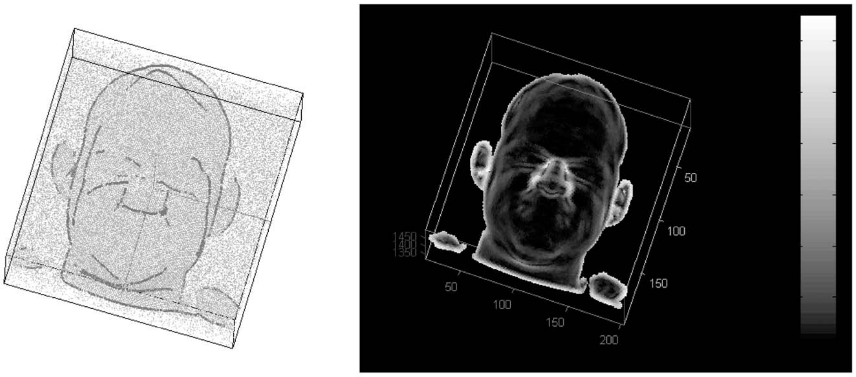
Vote collection for the first order case is performed by vector addition. The accumulated result is a vector whose direction points to a weighted center of mass from which votes are cast and whose magnitude encodes polarity. Since the first order votes are weighted by the saliency of the voters and attenuate with distance and curvature, their vector sum points to the direction from which the most salient contributions were received. A relatively low polarity indicates a token that is in the interior of a curve, surface, or region, therefore surrounded by neighbors whose votes cancel out each other. On the other hand, a high polarity indicates a token that is on or close to a boundary, thus receiving votes from only one side with respect to the boundary, at least locally. The interpretation of polarity vectors for boundary inference is done in conjunction with second order tensors and is described in Table 1. A detailed analysis for each case can be found in Section 4.

3.5 Dense Structure Extraction

Now that the most likely type of feature at each token has been estimated, we want to compute the dense structures (connected curves and surfaces in 3D) that can be inferred from the tokens. This can be achieved by casting votes to *all* locations, whether they contain a token or not, using the same voting fields and voting mechanism. Then, each site contains a 2-tuple $(s, \hat{v})$, indicating feature saliency and direction. Given this dense information, a modified marching algorithm [38] is used to extract surfaces and curves which correspond to the loci of zero crossings in s along $\hat{v}$s. Junctions are isolated and, therefore, are extracted as maxima of junction saliency. Extraction stops at boundaries, thus overcoming the inherent limitation of the Marching Cubes algorithm [39] that only extracts closed surfaces. Interested readers are referred to [1] or [38] for more details.

4 Boundary Inference

In this section, we describe how the theory developed in the previous section can be used to infer boundaries of 3D perceptual structures. We begin with surface boundary detection (surface end-curves) and then turn our attention to curve endpoint detection and region boundary inference (volume boundary). The optimality of these extracted structures is explained by our modified marching algorithms



(a) (b)

Fig. 8. (a) Noisy input (only one out of three points is correct) and the extracted curves. (b) Three-dimensional curve saliency map of surface boundaries, where white indicates high saliency and black indicates low saliency.

[38] for maximal surface and curve extraction, which extracts maxima along the detected polarity direction.

4.1 Surface Boundary Inference

We are interested in extracting surface end-curves that, in some applications, may indicate depth discontinuities or occlusion boundaries. In the case of surfaces, both interior points and points on boundaries are characterized by a dominant stick component. Our objective is to associate each input point Q with a 2-tuple $(s, \hat{v})_S$, where s denotes *surface boundary saliency* and $\hat{v}$ is a unit vector which indicates the normal direction to the underlying boundary curve. The curves depicted in Fig. 1 are points with large s values. When all $(s, \hat{v})_S$ are available, we extract the curves corresponding to maxima in s along the polarity vector direction, using a modified curve marching process [38].

Assume we are given an open smooth surface patch in 3D, encoded as a sparse set of tokens, possibly contained within a larger data set. The tokens are initially encoded as ball tensors since their preference of orientation is unknown. After a pass of second order voting, the tokens that lie on the surface, both in the interior and on the boundaries, have accumulated second order tensors with dominant stick components consistent with the normal of the surface at each location.

With oriented normal vectors, tokens propagate first order votes to their neighbors. As seen in Section 3.2, these votes will be along the tangent of the circular arc connecting the voter and the receiver. Therefore, the resulting polarity vector at the receiver after vote accumulation lies on a plane perpendicular to the estimated local surface normal. In case of a token in the interior of the region, the first order votes come from all directions and cancel each other out. On the other hand, close to the surface boundaries, a large vector sum is accumulated, pointing toward the average direction (weighted by vote saliencies) from which the votes came. This polarity direction is locally orthogonal to the boundary by definition. If the detected polarity vector is not exactly orthogonal to the estimated surface normal that is an indication of interference by noise, we use its projection on the plane orthogonal to the normal. The tensor is robust against this kind of interference since it contributes to the ball component and does not affect the estimated normal.

Fig. 8 shows the input and output and Fig. 9 illustrates a complete rundown of the major steps in surface end-curves extraction. The range scan of a human face is courtesy of the Signal Analysis and Machine Perception Laboratory, Ohio State University. To make the input more difficult, we add random noise to it, making the signal-to-noise ratio 1:2. After applying the procedure described, we extract the end-curves of the surfaces.

4.2 Curve Endpoint Inference

Besides the usefulness curve endpoints exhibit by themselves as the boundaries of smooth curves, they are also very useful in a hierarchical grouping scheme, under which elementary curve segments are grouped into extended curves and gaps are bridged as scale increases [35], [41], [34], [8]. In the context of single-scale tensor voting, the detection of curve endpoints is important since they indicate where the dense curve extraction process should be terminated. As in the case of surface boundaries, the second order representation alone cannot convey whether a point is in the interior of a curve or a curve endpoint since they are both characterized by a dominant plate component with curve saliency $\lambda_2 - \lambda_3$ and preferred tangent parallel to $\hat{e}_3$, the eigenvector corresponding to the minimum eigenvalue.

After second order voting, the eigenvector corresponding to the smallest eigenvalue ($\hat{e}_3$), of the tensor inferred at each location gives the tangent orientation. Signs of curvature are also detected. Then, we extract curve endpoints by casting first order votes and inferring a 2-tuple $(s, \hat{v})_C$ at each point: s encodes *curve endpoint saliency* and $\hat{v}$ is parallel to the tangent. At tokens that lie in the interior of the curve, first order votes come from both directions and cancel each other out. At the endpoints, all first order votes are cast from the same direction, thus combining into a large vector sum pointing toward the interior of the curve. Since curve endpoints are isolated in space, no marching process is needed for their extraction. The exterior tokens with respect to the curve tangent that have accumulated high polarity are selected as the endpoints. Fig. 10 shows the input and the salient curve endpoints extracted for the end-curves of the surfaces inferred from the noisy data.

An example of simultaneous inference of surfaces, curves, surface intersections, junctions, surface boundaries, and curve endpoints is presented in Fig. 11.

4.3 Volume Boundary Inference

Given a noisy set of points that belong to a 3D volume, we infer its boundaries by associating a 2-tuple $(s, \hat{v})_V$ to each point after noise removal: s denotes *region boundary saliency* and $\hat{v}$ is the normal pointing to the "inside" of the point

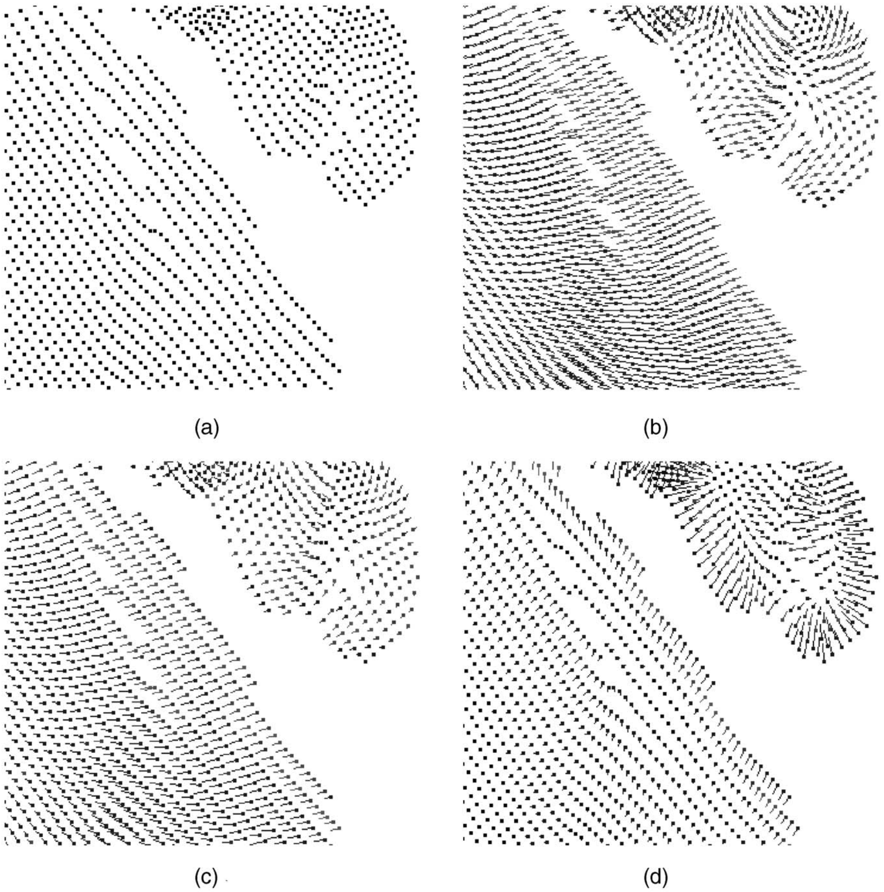


Fig. 9. Complete rundown of the major steps in end-curve detection. (a) A magnified portion of the point data in Fig. 8 (after noise rejection by tensor voting). (b) The result after the first pass. Surface saliency is indicated by the length of each normal. (c) Sign of curvature is used to orient normal vector. (d) Polarities obtained by first order voting.

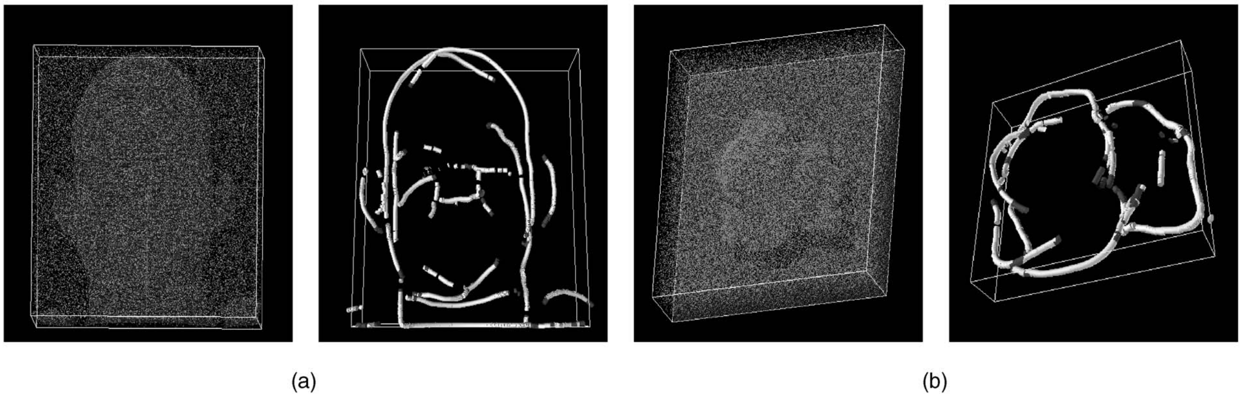


Fig. 10. Two results: (a) Face and (b) frog on curve endpoint inference. The noisy data set is obtained by adding two random points for every true data point. The result of surface boundary inference of the objects of interest are marked in white and points with high curve endpoint saliency are coded in in dark gray. (The face scan data is courtesy of the Signal Analysis and Machine Perception Laboratory, Ohio State University.)

cluster. Note that, in this case, the normal refers to the vector inferred by first order voting since the characteristic second order tensor of a region is a ball that has no orientation preference.

In terms of second order tensors, regions are characterized by a dominant ball component since they collect second order votes from all directions in 3D. The same holds for tokens close to the region boundaries since second order votes are a function of orientation but not direction. Once second order information is available at each token, first order votes are cast. The bounding surface of a 3D region can be extracted by the modified surface marching algorithm [38] as the maximal isosurface of s along $\hat{v}$.

Fig. 12 shows input and results on surface boundary extraction. The input is a set of approximately 600,000 unoriented tokens uniformly distributed inside a peanut-like shape. A number of random points, ranging from 600,000 to 2.4 million, drawn from a uniform distribution is added to the data set. Fig. 12a shows the noise free input, while Fig. 12b shows the input with 600,000 random points added. Inputs

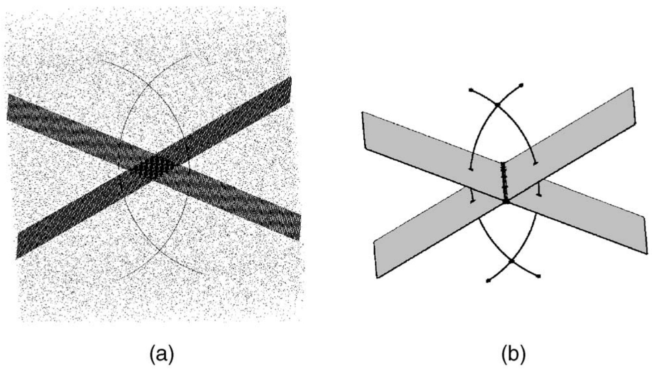


Fig. 11. Results on simultaneous inference of multiple types of structures. (a) Unoriented data set that consists of two intersecting planes, two intersecting curves, and random outliers. (b) Output after voting. Outliers have been rejected due to very low saliency. Surface inliers are marked in gray, curves and boundaries in black. Curve endpoints and junctions have been enlarged.

with more random points are too noisy to be displayed. Second and first order voting is performed using the same scale in all cases and the bounding surfaces of the peanut are extracted based on their high polarity. Table 2 reports the true positive rates for the extracted boundaries. Tokens labeled volume boundaries are considered correct if they are within 1 percent of the peanut's size from the actual boundary as defined by the equations used to generate the input. The processing time for the data set with 3 million total points was 43 minutes on a Pentium 4 PC at 2.8GHz.

5 Multiscale Analysis

In this section, we present an algorithm that uses first and second order voting to adjust the scale of analysis, a natural application of the first order augmented framework. Our approach operates in a fine-to-coarse fashion, much like most other multiple scale methods in computer vision [33], [27], [36], [10], [35], [5]. This allows the preservation of fine details where they exist, while it delays the enforcement of global smoothness as long as possible. Tokens that have already been processed at a fine scale are left untouched at subsequent larger scales that are only applied to regions with discontinuities. Accurate boundary detection, therefore, is a critical module in such a scheme.

A technique for automatic scale selection based on the existence of discontinuities at the previous scale will be illustrated in the running example of this section that comes from the field of medical imaging. In medical image analysis, we usually assume the absence of junctions. Even though the surfaces of organs and tissues can sometimes be very convoluted, they should still be smooth. Taking advantage of this domain-specific knowledge, voting is first performed with a small scale and we can determine whether it is sufficient for each region based on the presence or absence of junctions.

We first use a 2D synthetic image to illustrate a complete rundown of our multiscale algorithm. By thresholding image intensity, we produce an initial set of tokens shown in Fig. 13a. The desired feature is a ribbon-like structure. However, intensity thresholding is inaccurate and misses some data in the middle. We begin by detecting region boundaries in 2D. We obtain estimates of polarity directions that indicate region boundary saliency by second and first order tensor voting (Fig. 13b). Then, using the detected region boundaries as inputs, these estimates are refined into more accurate normal directions of the bounding curves. Tokens with high junction saliency indicate highly convoluted features or errors resulting from missing data (both cases are characterized by a set of inconsistent normals violating the continuity constraint). A fixed size neighborhood (e.g., $3 \times 3 \times 3$, independent of the scale σ) with high junction saliency is removed around each detected junction (Fig. 13c). Then, we perform 2D endpoint detection on the remaining region boundaries, (Fig. 13d). These will serve as interfaces where gaps will be bridged as the scale is increased. Scale is further increased until the detected endpoints are connected smoothly or we reach the largest permissible scale (Fig. 13e). Finally, the same marching curve process [38] is applied to produce a dense continuous curve (Fig. 13f).

5.1 Multiscale Algorithm

The algorithm for 3D presented below is a generalization of the 2D case. Scales σ_1 and σ_n are the smallest and largest permissible scales, respectively. The sensitivity of this selection is very low and any reasonable range of scales should produce very similar results. b denotes a token and $neighborhood(b)$ is a fixed size neighborhood around it.

```
SURFACEEXTRACT($\sigma_1, \sigma_n$)
1. Threshold intensity (Fig. 13a)
2. Vote for the bounding surfaces of 3D regions
         using $\sigma_1$ (Fig. 13b)
3. For each $neighborhood(b)$ do
     $\sigma \leftarrow \sigma_1$
     repeat
       Vote for bounding surface normals
           using $\sigma$ (Fig. 13c)
       Vote for signs of curvatures using $\sigma$ [40]
       if ($\max(neighborhood(b).JunctionSaliency)) > \mu_j$)
           (Fig. 13c)
         Remove tokens around $b$'s neighborhood
       Vote for surface end-curves using $\sigma$ (Fig. 13d)
       if ($\max(neighborhood(b).SurfaceBoundarySaliency))$
           $> \mu_e$)
         $\sigma \leftarrow \sigma + k$
       Vote for bounding surfaces of regions
           (Figs. 13e and 13f)
     until ($neighborhood(b).SurfaceBoundarySaliency < \mu_e$)
           or ($\sigma > \sigma_n$)
```

The other parameters are: μ_j is the maximum acceptable junction saliency, below which a token is not classified as a junction; μ_e is the minimum volume boundary saliency, above which a token is classified as a boundary; k is the step by which scale is incremented at each iteration.

In Step 3 above, we want to find the smallest $\sigma \in [\sigma_1, \sigma_n]$ such that continuity is satisfied at the smallest possible scale and, at the same time, the finest details are preserved. At each iteration, smoothness and continuity are examined by the two conditional statements in the algorithm. Smoothness is ensured if no junctions can be detected and continuity is ensured if no structural boundaries exist. We have exactly four cases (recall the saliency definitions in Table 1):

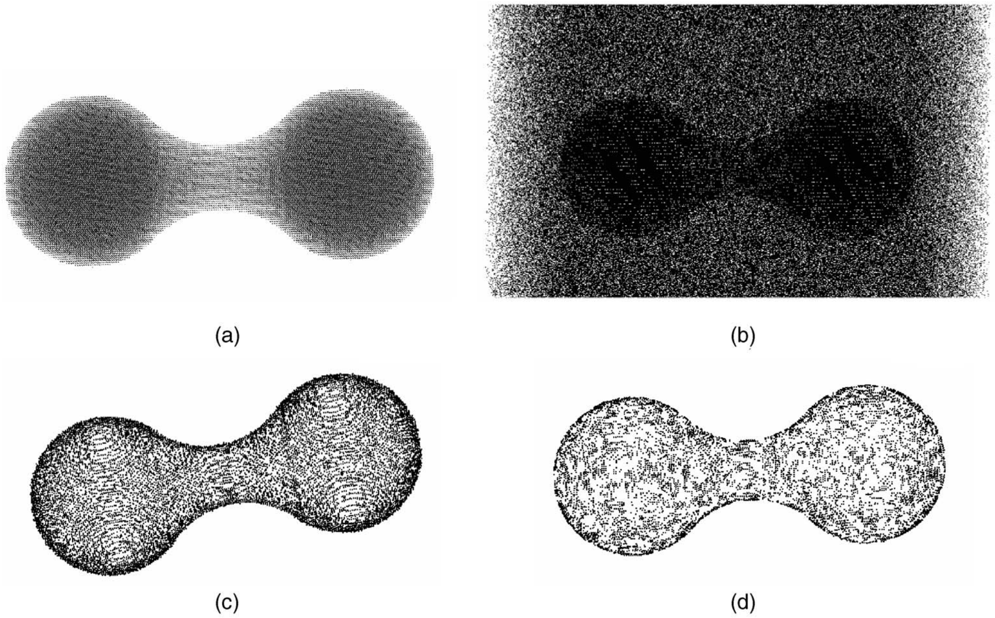


Fig. 12. (a) Results on volume boundary inference. The unoriented inliers form a peanut. Noise-free input. Large numbers of random points are added to the data set. (b) Input with 600,000 random. (c) Boundaries with 1.2 million outliers. (d) Boundaries with 2.4 million outliers.

TABLE 2
Correct Detection Rates for the Boundaries of the "Peanut" of Fig. 12

Number of random points	0	0.6M	1.2M	1.8M	2.4M
Boundary detection true positive rate	100%	99.2%	98.8%	98.4%	97.7%

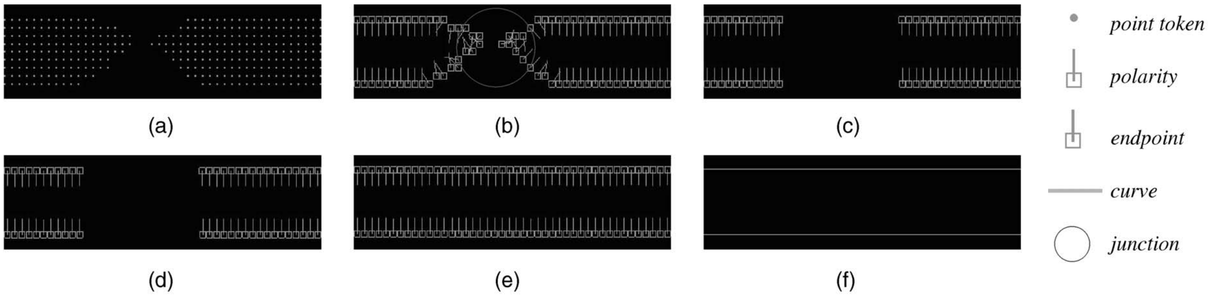


Fig. 13. Illustration of our multiscale technique (a) initial set of tokens, (b) polarities; a junction is labeled, (c) removal of tokens around the detected junction, (d) endpoint detection, (e) missing tokens are inferred during the multiscale analysis, and (f) final curve extraction result.

1. *High junction saliency and high end-curve saliency*. Since no junction exists and a token with high end-curve saliency also has high surface saliency, this scenario indicates the presence of highly convoluted surface, where many surface normals exist in the neighborhood. This scenario is very possible for the brain data or complicated tissue we test in the next section. As continuity and smoothness constraints are violated at the current scale σ, we increase σ by k to enforce more smoothness in a larger scale in the next scale iteration.
2. *High junction saliency and low end-curve saliency.* This scenario indicates a salient junction, where end-curve (and, thus, surface) saliency is low. A salient point junction should be absent from real medical data. Therefore, the reason why the smoothness constraint is violated at the current scale σ is very possibly due to missing or noisy data. Therefore, we remove a small, fixed size neighborhood at this point, which should automatically result in high end-curve saliency at the current scale due to the breaking of continuity. The resulting gap can be bridged as the scale is increased in the next scale iteration.
3. *Low junction saliency and high end-curve saliency*. Without any salient junction, a salient end-curve indicates the presence of erroneous data because, supposedly, only smooth and continuous organ/tissue structure exists. In this case, we increase the scale so that the fragmented data can be connected smoothly in the next scale iterations.
4. *Low junction saliency and low end-curve saliency*. The current scale of analysis is acceptable for maintaining both smoothness and continuity. Therefore, the repeat loop stops if this condition is true.

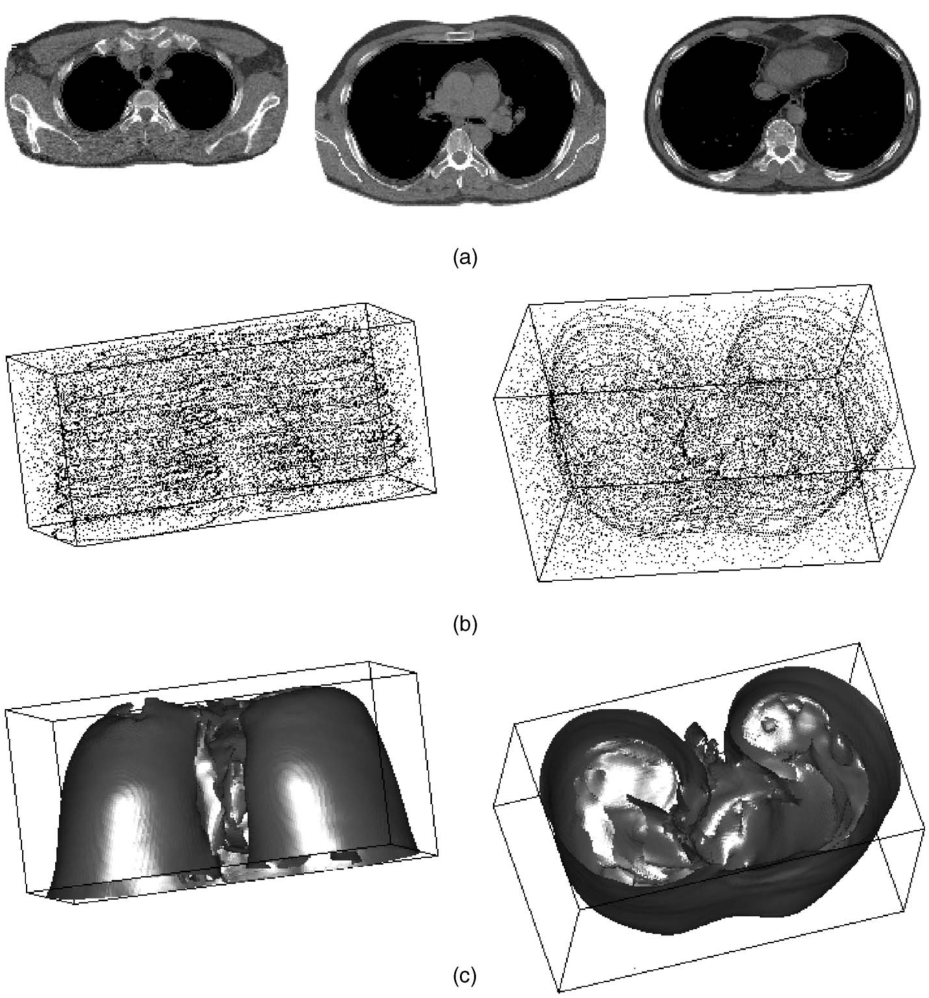


Fig. 14. Results on the 12 slices of the CT scan of the Thorax data set. (a) Three slices of the CT scan showing the inferred boundaries of the thorax (the bright curves), (b) 3D boundary points before surface extraction, and (c) result of the surface extraction using our method.

A typical value of k is 1 in a $150 \times 150 \times 150$ quantized grid for medical data. Experiments in [1] show that the step size k is not critical since tensor voting has low sensitivity over a reasonable range in $[\sigma_1, \sigma_n]$. If k is too large, oversmoothing may occur. The step size has a marginal effect on the performance of the system.

6 Experimental Results

We apply the augmented framework on medical images, as the data objects exhibit smooth structure and would benefit the most from our method that enforces/facilitates smoothness. Since CT and MRI scans are 2D images, we prepare the 3D data by applying intensity segmentation on the images by thresholding and stacking the thresholded points to produce a 3D point set. First order region inference as well as the augmented surface extraction method are then applied on the 3D data. As we shall see, with minimal knowledge, our method can refine and extract accurate results. Running times are measured on a Pentium III at 600 MHz, with 384MB RAM.

Fig. 14 shows the reconstruction result of the THORAX data set. This is a set of real CT scans (courtesy of the University of Washington Health Science Center). Only 12 interleaved slices, whose z-coordinates range unevenly from 110 to 180, are used. To make the example more challenging, we add a total of 12,000 noisy points to the volume (signal: noise = 1:2). After intensity thresholding, we initialize the tokens as ball tensors and perform 2D first order voting on the 12 slices to infer the region boundaries.[3] The segmentation results on three sample slices are shown in Fig. 14a. Next, we stack together the resulting thorax boundaries into a 3D volume of dimensions $170 \times 114 \times 70$ (Fig. 14b). The complete surface is then extracted by the augmented version of 3D tensor voting, which consists of 3D second order voting to refine the normal of the sparse boundary points, inference of sign of curvature, followed by surface extraction. Two views of the segmented surface are shown in Fig. 14c. We do not use any assumption other than the smoothness and continuity requirements mentioned in the second paragraph of Section 5 in our reconstruction. Processing times are shown in Table 3.

The MCGILL BRAIN, from the Brainweb [42], is used to evaluate our algorithm. The dimensions of the MRI data set are $181 \times 217 \times 181$, with resolution set at $1mm^3$. We are interested in three types of tissues: the cerebrospinal fluid or CSF, the gray matter or GM, and the white matter or WM. We perform initial intensity segmentation on the MRI images and

[3] In most situations, region boundary inferences are performed in the 3D volume except here when the input consists of widely separated and interleaved slices, region inferences will be more stable in 2D. Surface extraction is, however, always performed in 3D.

TABLE 3
Processing Times for the THORAX Data Set

	intensity|σ	times (secs)	Output
Thresholding	0-1	10	thresholded data
First order voting	3	45	2-D region boundary
Second order voting	15	495	3-D normal, sign of Gaussian curvature, surface extraction
total		550	surface model

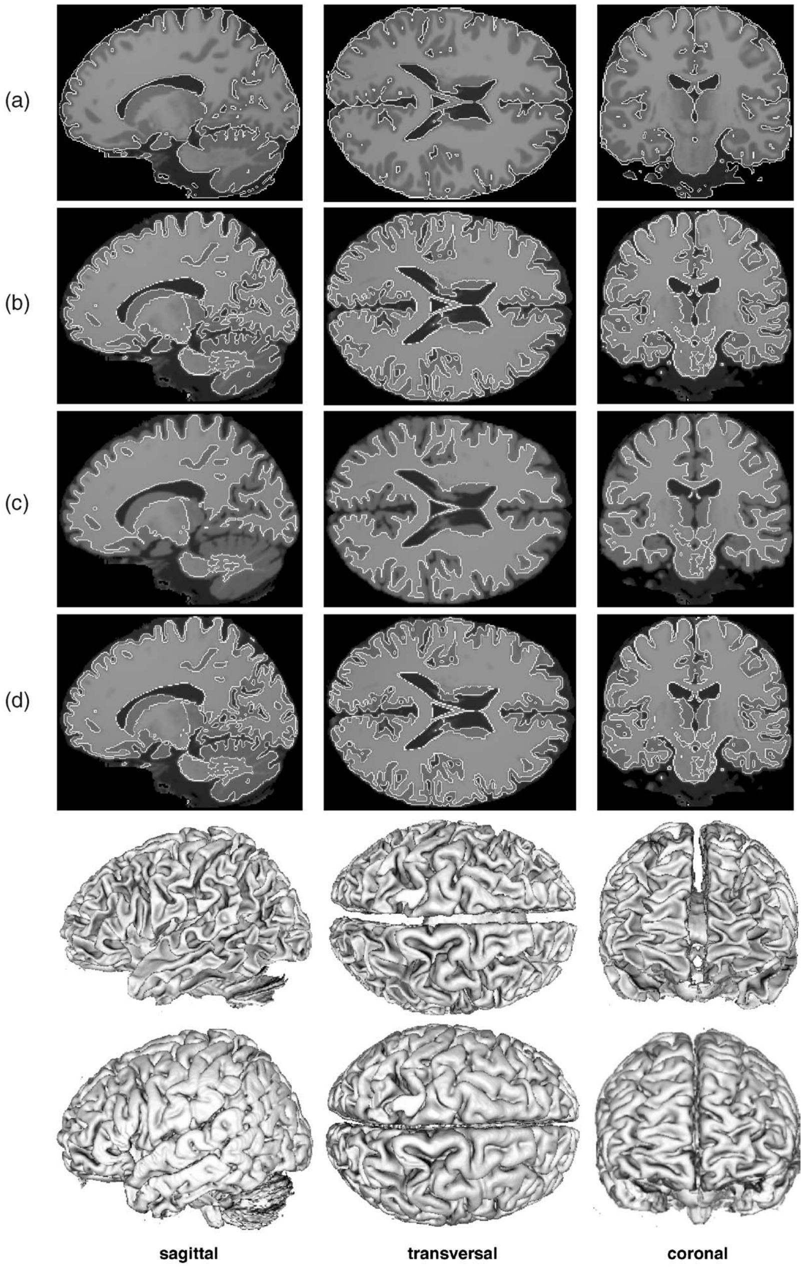


Fig. 15. Results on the MCGILL BRAIN data set. The sagittal, transversal, coronal views of the extracted (a) CSF, (b) gray matter, and (c) white matter boundaries are shown. The inner cortical surface and the outer cortical surface are shown together in (d). In (e) and (f), three views of the inner and outer cortical surface mesh are shown.

stack them up into three 3D tissue point sets. This is a very challenging data set since all surfaces are very convoluted. Without any a priori assumption or initialization, we can still robustly separate two closely spaced surfaces that define the boundaries of the CSF ribbon (or CSF), the gray matter ribbon (or GM), and white matter (or WM) (Figs. 15a, 15b,

TABLE 4
Processing Times for the McGill Brain Data Set

	intensity|σ			Processing times (secs)		
	CSF	GM	WM	CSF	GM	WM
Thresholding	1-89	121-255	90-120	35	41	43
First order	0.5	0.5	0.5	468	615	723
Second order	2	2	2	≈1500	≈1500	≈1500

and 15c) by using 3D first order voting for region boundary detection for the respective tissue volume. To further delineate the two bounding surfaces of the gray matter, we perform two additional operations, based on the knowledge that CSF encloses GM which, in turn, encloses WM. Thus, $CSF \cap GM$ gives the outer cortical surface, whereas $GM \cap WM$ produces the inner cortical surface. Fig. 15d shows the resulting intersections for both the inner and the outer surface. Finally, Figs. 15e and 15f show the result of the inner and outer cortical surface using the augmented tensor voting, which consists of 3D second order voting to refine the normal, inference of sign of curvature, followed by surface extraction. While only the continuity and smoothness constraints are used when we apply our first order augmented tensor voting framework, there exist other specialized methods in the medical image computing literature that make use of prior knowledge. The use of prior models to extract brain surfaces is however out of the scope of this paper. Table 4 summarizes the processing times for the MCGILL BRAIN data set.

Algorithm 1 GENTENSORVOTE (voter,votee)

It uses GENNORMALVOTE to compute the most likely normal direction at the votee. Then, plate and ball tensors are computed, by integrating the resulting normal votes cast by voter.

```
for all $0 \leq i, j < 3$, outTensor[i][j] $\leftarrow 0$
for all $0 \leq i < 2$,
    voterSaliency[i] $\leftarrow$ voter[$\lambda_i$] $-$ voter[$\lambda_{i+1}$]
voterSaliency[2] $\leftarrow$ voter[$\lambda_2$]
if (voterSaliency[0] $> 0$) then
  vecVote $\leftarrow$ GENNORMALVOTE (voter,votee)
  {Compute stick component }
  COMBINE (outTensor,vecVote)
end if
transformVoter $\leftarrow$ voter
for $i = 1$ to 2   // 1: plate, 2: ball do
  if (voterSaliency[i] $> 0$) then
    // count[i] is a sufficient number of samples uniformly distributed on a unit $(i+1)$-D
    sphere.
    while (count[i] $\neq 0$) do
      transformVoter[direction] $\leftarrow$ sample[direction] $\leftarrow$ GENUNIFORMDISTRIBUTEPT()
      if ($i \neq 2$) then
        /* Compute the alignment matrix, except for the isotropic ball tensor */
        transformVoter[direction] $\leftarrow$ voter[eigenvectorMatrix] $\times$ sample[direction]
      end if
      vecVote $\leftarrow$ GENNORMALVOTE (transformVoter,votee)
      COMBINE (outTensor, vecVote, voterSaliency[i])
      count[i] $\leftarrow$ count[i] $- 1$
    end while
  end if
end for
return outTensor
```

Fig. 16. Algorithm 1.

```
**Algorithm 2** GENNORMALVOTE (voter, votee)
A vote (vector) on the most likely normal direction is returned.
  $v \leftarrow$ votee[position] $-$ voter[position]
  /* voter and votee are connected by high curvature? */
  if (angle(voter[direction],$v$) $< \pi/4$) then
    return NULL {smoothness constraint violated}
  end if
  stickvote[position] $\leftarrow$ votee[position]
  /* voter and votee on a straight line, or voter and votee are the same point */
  if (angle(voter[direction],$v$) $= \pi/2$) or (voter $=$ votee) then
    stickvote[direction] $\leftarrow$ voter[direction]
    stickvote[length] $\leftarrow e^{-\frac{s^2}{\sigma^2}}$ {equation (2)}
    return stickvote
  end if
  Compute center and radius of the osculating hemisphere
  /* assign stick vote */
  stickvote[direction] $\leftarrow$ center $-$ votee[position]
  stickvote[length] $\leftarrow e^{-(\frac{s^2+c\kappa^2}{\sigma^2})}$ {equation (2)}
  return stickvote
```

Fig. 17. Algorithm 2.

We perform a simple error analysis on the resulting gray matter surfaces obtained above (Fig. 15), for which ground truth is available. Qualitatively, the results are very satisfactory. Our quantitative comparison is as follows: The true bounding surfaces of gray matter are first derived from the set of pixels known to be gray matter (ground truth) by manual expert tracing. We count the number of pixels classified as cortex surfaces, and check against the true boundary pixels. Let $TP\%$ be the true positive percentage that indicates our correct classification. Let $FP\%$ and $FN\%$ be the false positive and negative percentage, respectively. We obtained the following results: $TP = 89.76\%$, $FP = 4.36\%$, $FN = 5.10\%$. We have approximately 90 percent correctness, which is comparable to the results reported in [43] for the same example. Their focus is the correctness of tissue volume labeling and a $TP\%$ of 92 is reported. On the other hand, our interest is the correctness of tissue surface labeling, which could be a more sensitive metric and, hence, we have a slightly lower $TP\%$. We indicate the cortical bounding surface tokens in our figures.

7 Discussion and Conclusion

In this paper, we have presented a first order augmentation to the tensor voting framework that fits naturally within the theory described in [1]. It was illustrated in 3D, but it provides a boundary detection mechanism for manifolds of dimension 1 to N in any N-dimensional space. We have shown how the first order vector fields can be derived from the *fundamental 2D first order stick field*, in the same way as the second order tensor fields. Our new representation maintains all the critical properties of the strictly second order one, including reasonable computational complexity, and, at the same time, provides a richer description for perceptual structures in any dimension. In the same way that the second order tensor simultaneously encodes the saliency of the token as an elementary surface patch, an elementary curve segment, a junction, or noise, thus allowing the decisions to be made at a later stage, the polarity vector encodes the saliency of the token as a boundary of a perceptual structure without labeling the token as a boundary before the local neighborhood is considered. The voting fields allow tokens with different tensor structures to interact with each other and provide the means for junctions, curves, surfaces, volumes, and their boundaries to emerge from the same space. We claim that this is an important contribution and that results should be superior with unified processing as opposed to treating each type of structure separately.

Putting the capability to detect discontinuities to use, we have also opened the door to multiscale data analysis using tensor voting. The results we have obtained are encouraging and they justify both the fine-to-coarse manner in which we process the data, as well as the automatic scale selection technique we use. These allow us to locally resolve the conflict between detail preservation, on one hand, and smoothness and continuity, on the other. Scale adaptation based on local criteria bypasses the need for computing the entire scale-space of large three-dimensional data sets and the subsequent need for searching for the proper output in scale-space. A major axis of our future work will be along further experiments on multiscale data analysis. There is a plethora of issues that are worthy of investigation, such as the selection of the range of scales as a function of the level of noise, the choice of the local criteria used for scale adaptation, and the development of multiresolution representations for the data.

```
Algorithm 3 COMBINE (tensorvote, stickvote, weight)
It performs tensor addition, given a stick vote.
  for all $i, j$ such that $0 \leq i, j < 3$ do
    tensorvote[i][j] ← tensorvote[i][j] + weight × stickvote[direction][i] × stickvote[direction][j]
  end for
```

Fig. 18. Algorithm 3.

```
Algorithm 4 ADDTENSOR (outTensor, inTensor, weight)
Add two second order symmetric tensors – simply matrix addition
  for all $i, j$ such that $0 \leq i, j < 3$ do
    outTensor[i][j] ← outTensor[i][j] + weight × inTensor[i][j]
  end for
```

Fig. 19. Algorithm 4.

APPENDIX

ALGORITHMS FOR 3D TENSOR VOTING

We detail the general second order tensor voting algorithm [37] in this section. The voter uses GENTENSORVOTE to cast a tensor vote to the vote receiver (votee). Votes generated by GENNORMALVOTE are accumulated using COMBINE. A 3×3 $outTensor$ is the output. The votee thus receives a set of $outTensor$ from voters within its neighborhood. The resulting tensor matrices can be summed up by ADDTENSOR, which performs ordinary 3×3 matrix addition. The final matrix after accumulation describes an ellipsoid in 3D. For first order voting, ADDVECTOR is used for vote accumulation, which is plain vector addition. See Figs. 16, 17, 18, and 19.

ACKNOWLEDGMENTS

This research is supported by the Hong Kong Research Grant Council (Project No. HKUST 6193/02E), the National Science Foundation (Project No. 9811883), and the Integrated Media Systems Center, and National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152.

REFERENCES

[1] G. Medioni, M.S. Lee, and C.K. Tang, *A Computational Framework for Segmentation and Grouping.* Elsevier, 2000.

[2] W.S. Tong, C.K. Tang, and G. Medioni, "First Order Tensor Voting and Application to 3-D Scale Analysis," *Proc. Computer Vision and Pattern Recognition,* vol. I, pp. 175-182, 2001.

[3] M. Wertheimer, "Laws of Organization in Perceptual Forms," *Psycologische Forschung,* translation by W. Ellis, A source book of Gestalt psychology (1938), vol. 4, pp. 301-350, 1923.

[4] M. Nicolescu and G. Medioni, "Layered 4D Representation and Voting for Grouping from Motion," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 4, pp. 492-501, Apr. 2003.

[5] T. Lindeberg, "Scale-Space: A Framework for Handling Image Structures at Multiple Scales," *CERN School of Computing,* pp. 695-702, 1996.

[6] Z. Li, "A Neural Model of Contour Integration in the Primary Visual Cortex," *Neural Computation,* vol. 10, pp. 903-940, 1998.

[7] C.M. Gray, "The Temporal Correlation Hypothesis of Visual Feature Integration: Still Alive and Well," *Neuron,* vol. 24, pp. 31-47, 1999.

[8] E. Mingolla, W.D. Ross, and S. Grossberg, "A Neural Network for Enhancing Boundaries and Surfaces in Synthetic Aperture Radar Images," *Neural Networks,* vol. 12, pp. 499-511, 1999.

[9] R. Raizada and S. Grossberg, "Context-Sensitive Bindings by the Laminar Circuits of v1 and v2: A Unified Model of Perceptual Grouping, Attention, and Orientation Contrast," *Visual Cognition,* vol. 8, nos. 3-5, pp. 431-466, June 2001.

[10] E. Saund, "Symbolic Construction of a 2D Scale-Space Image," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 8, pp. 817-830, Aug. 1990.

[11] D. Terzopoulos and D. Metaxas, "Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 7, pp. 703-714, July 1991.

[12] J.A. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Science.* Cambridge Univ. Press, 1996.

[13] S. Osher, H.K. Zhao, and R. Fedkiw, "Fast Surface Reconstruction Using the Level Set Method," *UCLA Computational and Applied Math. Reports,* pp. 32-40, 2001.

[14] S. Osher and R.P. Fedkiw, *The Level Set Method and Dynamic Implicit Surfaces.* Springer Verlag, 2002.

[15] K.L. Boyer and S. Sarkar, "Perceptual Organization in Computer Vision: Status, Challenges, and Potential," *Computer Vision and Image Understanding,* vol. 76, no. 1, pp. 1-6, Oct. 1999.

[16] A. Shashua and S. Ullman, "Structural Saliency: The Detection of Globally Salient Structures Using a Locally Connected Network," *Proc. Int'l Conf. Computer Vision,* pp. 321-327, 1988.

[17] P. Parent and S.W. Zucker, "Trace Inference, Curvature Consistency, and Curve Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 8, pp. 823-839, Aug. 1989.

[18] P.T. Sander and S.W. Zucker, "Inferring Surface Trace and Differential Structure from 3D Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 9, pp. 833-854, Sept. 1990.

[19] S. Sarkar and K.L. Boyer, "A Computational Structure for Preattentive Perceptual Organization: Graphical Enumeration and Voting Methods," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 24, no. 2, pp. 246-267, Feb. 1994.

[20] S. Grossberg and E. Mingolla, "Neural Dynamics of Form Perception: Boundary Completion," *Psychological Rev.,* pp. 173-211, 1985.

[21] S. Grossberg and D. Todorovic, "Neural Dynamics of 1-D and 2-D Brightness Perception: A Unified Model of Classical and Recent Phenomena," *Perception and Psychophysics,* vol. 43, pp. 723-742, 1988.

[22] F. Heitger and R. von der Heydt, "A Computational Model of Neural Contour Processing: Figure-Ground Segregation and Illusory Contours," *Proc. Int'l Conf. Computer Vision,* pp. 32-40, 1993.

[23] L.R. Williams and D.W. Jacobs, "Stochastic Completion Fields: A Neural Model of Illusory Contour Shape and Salience," *Neural Computation,* vol. 9, no. 4, pp. 837-858, 1997.

[24] S.C. Yen and L.H. Finkel, "Extraction of Perceptually Salient Contours by Striate Cortical Networks," *Vision Research,* vol. 38, no. 5, pp. 719-741, 1998.

[25] L.R. Williams and K.K. Thornber, "A Comparison of Measures for Detecting Natural Shapes in Cluttered Backgrounds," *Int'l J. Computer Vision,* vol. 34, no. 2/3, pp. 1-16, Aug. 1999.

[26] H. Neumann and E. Mingolla, "Computational Neural Models of Spatial Integration in Perceptual Grouping," *From Fragments to Objects: Grouping and Segmentation in Vision,* T.F. Shipley and P.J. Kellman, eds., pp. 353-400, 2001.

[27] A.P. Witkin, "Scale-Space Filtering," *Proc. Fourth Int'l Joint Conf. Artificial Intelligence,* pp. 1019-1022, 1983.

[28] F. Mokhtarian, "Multi-Scale Description of Space Curves and Three-Dimensional Objects," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 298-303, 1988.

[29] F. Mokhtarian and A.K. Mackworth, "A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, no. 8, pp. 789-805, Aug. 1992.

[30] D.G. Lowe, "Organization of Smooth Image Curves at Multiple Scales," *Int'l J. Computer Vision,* vol. 3, no. 2, pp. 119-130, June 1989.

[31] P. Perona and J. Malik, "Scale Space and Edge Detection Using Anisotropic Diffusion," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 7, pp. 629-639, July 1990.

[32] T. Lindeberg, "Principles for Automatic Scale Selection," *Handbook on Computer Vision and Applications,* vol. 2, pp. 239-274, 1999.

[33] D. Marr, *Vision.* Freeman Press 1982.

[34] E. Saund, "Labeling of Curvilinear Structure Across Scales by Token Grouping," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 257-263, 1992.

[35] J. Dolan and E.M. Riseman, "Computing Curvilinear Structure by Token-Based Grouping," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 264-270, 1992.

[36] F. Mokhtarian and A.K. Mackworth, "Scale Based Description and Recognition of Planar Curves and Two-Dimensional Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, no. 1, pp. 34-43, Jan. 1986.

[37] C.K. Tang, G. Medioni, and M.S. Lee, "N-Dimensional Tensor Voting and Application to Epipolar Geometry Estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 8, pp. 829-844, Aug. 2001.

[38] C.K. Tang and G. Medioni, "Inference of Integrated Surface, Curve, and Junction Descriptions from Sparse 3D Data," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 11, pp. 1206-1223, Nov. 1998.

[39] W.E. Lorensen and H.E. Cline, "Marching Cubes: A High Resolution 3D Surface Reconstruction Algorithm," *Computer Graphics,* vol. 21, no. 4, pp. 163-169, 1987.

[40] 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, pp. 858-864, June 2002.

[41] R. Mohan and R. Nevatia, "Perceptual Organization for Scene Segmentation and Description," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, no. 6, pp. 616-635, June 1992.

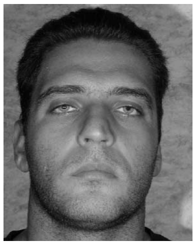
[42] "Brainweb," McConnel Brain Imaging Center at the Montreal Neurological Inst. http://www.bic.mni.mcgill.ca/, 2004.

[43] X. Zeng, L.H. Staib, R.T. Schultz, and J.S. Duncan, "Segmentation and Measurement of the Cortex from 3-D MR Images," *Proc. Int'l Conf. Medical Image Computing and Computer-Assisted Intervention,* pp. 519-530, 1998.

[44] L.M. Lorigo, O.D. Faugeras, W.E.L. Grimson, R. Keriven, R. Kikinis, A. Nabavi, and C.F. Westin, "Codimension-Two Geodesic Active Contours for the Segmentation of Tubular Structures," pp. 444-451, *Proc. Computer Vision and Pattern Recognition,* 2000.

Wai-Shun Tong received the BEng and MPhil degrees from the Hong Kong University of Science of Technology in computer science. He was a visiting student at the University of Southern California (USC), Los Angeles, in 2001. In 2002, he was awarded the Microsoft Fellowship Asia. He is currently a visiting student at the Visual Computing Group of Microsoft Research Asia. His research interests include low to mid-level vision such as segmentation, correspondence, shape analysis, and vision and graphics topics. He is a student member of the IEEE Computer Society.

Chi-Keung Tang received the MS and PhD degrees in computer science from the University of Southern California (USC), Los Angeles, in 1999 and 2000, respectively. He has been with the Computer Science Department at the Hong Kong University of Science and Technology since 2000, where he is currently an assistant professor. He is also an adjunct researcher at the Visual Computing Group of Microsoft Research, Asia, working on various exciting research topics in computer vision and graphics. His research interests include low- to mid-level vision such as segmentation, correspondence, shape analysis, and vision and graphics topics such as image-based rendering and medical image analysis. He is a member of the IEEE Computer Society.



Philippos Mordohai received the Diploma in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 1998 and the MS degree in electrical engineering from the University of Southern California (USC), Los Angeles, in 2000. He is pursuing the PhD degree in electrical engineering at USC. He is a graduate research assistant at the Computer Vision Laboratory of the Institute of Robotics and Intelligent Systems and the Integrated Media Systems Center at USC. His research interests in computer vision include perceptual organization and 3D reconstruction from two or multiple images. He is a student member of the IEEE and the IEEE Computer Society.

Gérard Medioni received the Diplome d'Ingenieur Civil from the Ecole Nationale Superieure des Telecommunications, Paris, France, in 1977 and the MS and PhD degrees in computer science from the University of Southern California, Los Angeles, in 1980 and 1983, respectively. He has been with USC since 1983, where he is currently a professor of computer science and electrical engineering, codirector of the Computer Vision Laboratory, and chairman of the Computer Science Department. He was a visiting scientist at INRIA Sophia Antipolis in 1993 and chief technical officer of Geometrix, Inc. during his sabbatical leave in 2000. His research interests cover a broad spectrum of the computer vision field and he has studied techniques for edge detection, perceptual grouping, shape description, stereo analysis, range image understanding, image to map correspondence, object recognition, and image sequence analysis. He has published more than 100 papers in conference proceedings and journals. Dr. Medioni is a fellow of the IEEE and a fellow of the IAPR. He has served on the program committees of many major vision conferences and was program chairman of the 1991 IEEE Computer Vision and Pattern Recognition Conference in Maui, program cochairman of the 1995 IEEE Symposium on Computer Vision held in Coral Gables, Florida, general cochair of the 1997 IEEE Computer Vision and Pattern Recognition Conference in Puerto Rico, program cochair of the 1998 International Conference on Pattern Recognition held in Brisbane, Australia, and general cochairman of the 2001 IEEE Computer Vision and Pattern Recognition Conference in Kauai. Professor Medioni is an associate editor of *Pattern Recognition and Image Analysis* and one of the North American editors for *Image and Vision Computing*.