A General Framework for Low Level Vision

Nir Sochen, Ron Kimmel, and Ravikanth Malladi

***Abstract*—We introduce a new geometrical framework based on which natural flows for image scale space and enhancement are presented. We consider intensity images as surfaces in the $(\mathbf{x}, \mathbf{I})$ space. The image is, thereby, a two-dimensional (2-D) surface in three-dimensional (3-D) space for gray-level images, and 2-D surfaces in five dimensions for color images. The new formulation unifies many classical schemes and algorithms via a simple scaling of the intensity contrast, and results in new and efficient schemes. Extensions to multidimensional signals become natural and lead to powerful denoising and scale space algorithms.**

***Index Terms*— Color image processing, image enhancement, image smoothing, nonlinear image diffusion, scale-space.**

I. Introduction

THE IMPORTANCE of dynamics of image geometry in the perception and understanding of images is by now well established in computer vision. Geometry, symmetry, and dynamics are also the main issues in physics. Borrowing ideas from high-energy physics, we propose in this paper a geometrical framework for low-level vision. The two main ingredients of this framework are 1) defining images as embedding maps between two Riemannian manifolds, and 2) an action functional that provides a measure on the space of these maps. This action is the natural generalization of the L2 Euclidean norm to non-Euclidean manifolds and is known as the *Polyakov action* in physics. The justification for the use of this functional in computer vision is twofold: It unifies many seemingly unrelated scale space methods on one hand, and provides new and improved ways to smooth and denoise images on the other. It will lead us in this paper to the construction of image enhancement procedures for gray and color images. The framework also integrates many existing denoising and scale space procedures by a change of a single parameter that switches between the Euclidean L1 and L2 norms.

Motivated by [2] and [31], we consider low level vision as an input to output process. For example, the most common input is a gray-level image; namely, a map from a two dimensional (2-D) surface to a three-dimensional (3-D) space $(\mathbb{R}^3)$. We have at each point of the xy coordinate plane an intensity $I(x, y)$. The $\mathbb{R}^3$ *space-feature* has Cartesian coordinates (x, y, I) where x and y are the *spatial* coordinates and I is the feature coordinate[1]. The output of the low level process in most models consists of 1) a smoothed image from which reliable features can be extracted by local and, therefore, differential operators, and 2) a segmentation, that is, either a decomposition of the image domain into homogeneous regions with boundaries, or a set of boundary points—an "edge map."

The process assumes the existence of layers serving as operators such that the information is processed locally in the layers and forwarded to the next layer with no interaction between distant layers. This means that the output has the form $\mathbf{X}(\Sigma, t)$ which is the solution of $\partial_t \mathbf{X} = O\mathbf{X}$, where O is a local differential operator and the input image is given as initial condition. This process yields a one-parameter family of images on the basis of an input image. Normally, such a family is called a *scale-space* (see [35] and references therein).

The importance of edges that are obtained from the intensity gradient is acknowledged, and gradient-based edge detectors are a basic operation in many computer vision applications. Edge detectors appear by now in almost all image processing tools. The importance of edges in scale space construction is also obvious. Boundaries between objects should survive as long as possible along the scale space, while homogeneous regions should be simplified and flattened in a more rapid way. We propose here a new nonlinear diffusion algorithm which does exactly that.

Another important question, for which there is only partial answers, is how to treat multivalued images. A color image is a good example since we actually talk about three images (red, green, blue) that are composed into one. Should one treat such images as multivalued functions as proposed in [14]?

We attempt to answer the above question by viewing images as embedding maps, that flow toward minimal surfaces. We go two dimensions higher than most of the classical schemes, and instead of dealing with isophotes as planar curves we deal with the whole image as a surface. For example, a gray level image is no longer considered as a function but as a 2-D surface in 3-D space. This idea is quite old [20], [46] for gray-level images, yet, to the best of our knowledge, it was never carried on to higher dimensions. As another example, we will consider a color image as 2-D surfaces now in five dimensions. We thank the editors for communicating to us a related effort that is published in this issue (see [47]).

We have chosen to present our ideas in the following order. Section II introduces the basic concepts of a metric and the induced metric and presents a measure on maps

Manuscript received August 26, 1996; revised March 11, 1997. This work was supported in part by the Applied Mathematics Subprogram of the OER under DE-AC03-76SFOOO98, by the ONR under NOOO14-96-1-0381, and by the NSF under PHY-90-21139. All calculations were performed at the Lawrence Berkeley National Laboratory, University of California, Berkeley, CA.

N. Sochen was with the Physics Department, University of California, Berkeley, CA 94720 USA. He is now with the School of Physics and Astronomy, Tel Aviv University, Ramat-Aviv, Tel-Aviv 69978, Israel.

R. Kimmel and R. Malladi are with the Department of Mathematics and Lawrence Berkeley National Laboratory, University of California, Berkeley, CA 94720 USA (e-mail: ron@math.lbl.gov; malladi@math.lbl.gov).

Publisher Item Identifier S 1057-7149(98)01749-7.

[1] While in this paper, the feature coordinate is simply the zeroth jet space $j^0 I$, we use the term *feature space* to leave room for a more general cases like texture [24], etc.

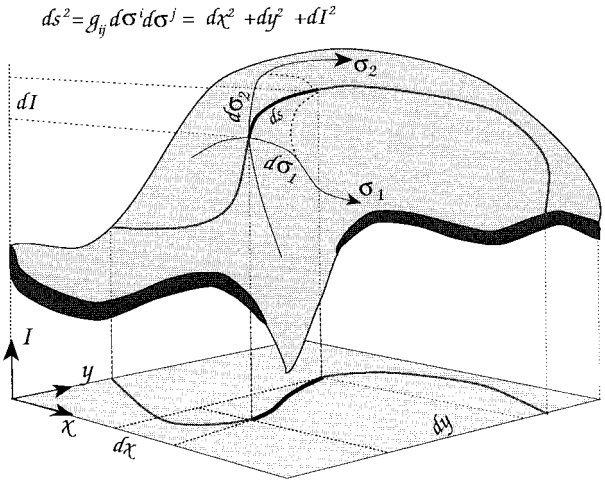

Fig. 1. Length element of a surface curve ds, may be defined either as a function of a local metric defined on the surface (σ_1, σ_2), or as a function of the coordinates of the space in which the surface is embedded (x, y, I).

between Riemannian manifolds that we borrowed from high-energy physics. This measure provides a general framework for nonlinear diffusion in computer vision, as shown in the following sections. In Section III, we introduce a new flow that we have chosen to name *Beltrami flow*, present a geometric interpretation in the simplest 3-D case, its relation to previous models, and two examples of the Beltrami flow for color images. Then, in Section IV, we refer to other models that are the result of the same action through different choices of the image metric and the minimization variables. We also study the geometrical properties of a generalized version of the mean curvature flow that is closely related to the proposed framework. We conclude in Section V with a summarizing discussion.

II. Polyakov Action and Harmonic Maps

A. The Geometry of a Map

The basic concept of Riemannian differential geometry is distance. The natural question in this context is: How do we measure distances? We will first take the important example $\mathbf{X} : \Sigma \rightarrow \mathbb{R}^3$. Denote the local coordinates on the 2-D manifold Σ by (σ^1, σ^2), these are analogous to arc length for the one-dimensional (1-D) manifold, i.e., a curve (see Fig. 1). The map $\mathbf{X}$ is explicitly given by $[X^1(\sigma^1, \sigma^2), X^2(\sigma^1, \sigma^2), X^3(\sigma^1, \sigma^2)]$. Since the local coordinates σ^i are curvilinear, the squared distance is given by a positive definite symmetric bilinear form called the metric whose components we denote by $g_{\mu\nu}(\sigma^1, \sigma^2)$

$$\begin{aligned} ds^2 &= g_{\mu\nu} d\sigma^\mu d\sigma^\nu \\ &\equiv g_{11}(d\sigma^1)^2 + 2g_{12} d\sigma^1 d\sigma^2 + g_{22}(d\sigma^2)^2 \end{aligned}$$

where we used Einstein summation convention in the second equality; identical indices that appear one up and one down are summed over. We will denote the inverse of the metric by $g^{\mu\nu}$, so that $g^{\mu\nu} g_{\nu\gamma} = \delta^\mu_\gamma$, where δ^μ_γ is the Kronecker delta.

Let $\mathbf{X}$: $\Sigma \rightarrow M$ be an embedding of (Σ, g) in (M, h), where Σ and M are Riemannian manifolds and g and h are their metrics, respectively. We can use the knowledge of the metric on M and the map $\mathbf{X}$ to construct the metric on Σ. This procedure, which is denoted formally as $(g_{\mu\nu})_\Sigma = \mathbf{X}^*(h_{ij})_M$, is called the *pullback* for obvious reasons and is given explicitly as follows:

$$g_{\mu\nu}(\sigma^1, \sigma^2) = h_{ij}(\mathbf{X}) \partial_\mu X^i \partial_\nu X^j \tag{1}$$

where $i, j = 1, \cdots, \dim M$ are being summed over, and $\partial_\mu X^i \equiv \partial X^i(\sigma^1, \sigma^2)/\partial\sigma^\mu$.

Take, for example, a grey-level image which is, from our point of view, the embedding of a surface described as a graph in $\mathbb{R}^3$, as follows:

$$\mathbf{X}\text{:}\ (\sigma^1, \sigma^2) \rightarrow [x = \sigma^1, y = \sigma^2, z = I(\sigma^1, \sigma^2)] \tag{2}$$

where (x, y, z) are Cartesian coordinates. Using (1) we get

$$(g_{\mu\nu}) = \begin{pmatrix} 1 + I_x^2 & I_x I_y \\ I_x I_y & 1 + I_y^2 \end{pmatrix} \tag{3}$$

where we used the identification $x \equiv \sigma^1$ and $y \equiv \sigma^2$ in the map $\mathbf{X}$.

Actually, we can understand this result in an intuitive way: (1) means that the distance measured on the surface by the local coordinates is equal to the distance measured in the embedding coordinates (see Fig. 1). Under the above identification, we can write $ds^2 = dx^2 + dy^2 + dI^2 = dx^2 + dy^2 + (I_x\, dx + I_y\, dy)^2 = (1 + I_x^2)\, dx^2 + 2 I_x I_y\, dx\, dy + (1 + I_y^2)\, dy^2$.

Next we provide a measure on the space of these maps.

B. The Measure on Maps

In this subsection, we present a general framework for nonlinear diffusion in computer vision. We will show in the sequel that many known methods fall naturally into this framework and how to derive new ones. The equations will be derived by a minimization problem from an action functional. The functional in question depends on *both* the image manifold and the embedding space. Denote by (Σ, g) the image manifold and its metric and by (M, h) the space-feature manifold and its metric, then the map $\mathbf{X}$: $\Sigma \rightarrow M$ has the following weight [34]:

$$S[X^i, g_{\mu\nu}, h_{ij}] = \int d^m\sigma \sqrt{g} g^{\mu\nu} \partial_\mu X^i \partial_\nu X^j h_{ij}(\mathbf{X}) \tag{4}$$

where m is the dimension of Σ, g is the determinant of the image metric, $g^{\mu\nu}$ is the inverse of the image metric, the range of indices is $\mu, \nu = 1, \cdots, \dim \Sigma$, and $i, j = 1, \cdots, \dim M$. The metric of the embedding space is h_{ij}.

To gain some intuition about this functional, let us take the example of a surface embedded in $\mathbb{R}^3$ and treat both the metric $(g_{\mu\nu})$ and the spatial coordinates of the embedding space as free parameters, and let us fix them to

$$(g_{\mu\nu}) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad x = \sigma^1, \quad y = \sigma^2. \tag{5}$$

We also adopt in $\mathbb{R}^3$ the Cartesian coordinates (i.e., $h_{ij} = \delta_{ij}$). Then, we get the Euclidean L2 norm

$$S[I,\, g_{\mu\nu} = \delta_{\mu\nu},\, h_{ij} = \delta_{ij}] = \int d^2\sigma(|\nabla x|^2 + |\nabla y|^2 + |\nabla I|^2). \tag{6}$$

If we now minimize with respect to I, we will get the usual heat operator acting on I. We see that the Polyakov action is the generalization of the L2 norm to curved spaces. Here, $d^m\sigma\sqrt{g}$ is the volume element (area element for $d = 2$) of Σ—the image manifold and $g^{\mu\nu}\partial_\mu X^i \partial_\nu X^j h_{ij}(\mathbf{X})$ is the generalization of $|\nabla I|^2$ to maps between non-Euclidean manifolds. Note that the volume element as well as the rest of the expression is reparameterization invariant. This means that they are invariant under a smooth transformation $\sigma^\mu \to \sigma^\mu(\sigma^1, \sigma^2)$. The Polyakov action really depends on the geometrical objects and not on the way we describe them via our parameterization of the coordinates.

Given the above functional, we have to choose the minimization. We may choose for example to minimize with respect to the embedding alone. In this case the metric $g_{\mu\nu}$ is treated as a parameter of the theory and may be fixed by hand. Another choice is to vary only with respect to the feature coordinates of the embedding space, or we may choose to vary with respect to the image metric as well. We will see that these different choices yield different flows. Some flows are recognized as existing methods like the heat flow, a generalized Perona–Malik flow, or the mean-curvature flow. Other choices are new and will be described below in detail.

Another important point is the choice of the embedding space and its geometry. In general, we need information about the task at hand in order to fix the right geometry. Take for example the grey-level images. It is clear that the intensity I is not on equal footing as x and y. In fact the relative scale of I with respect to the spatial coordinates (x, y) is to be specified. This can be interpreted as taking the metric of the embedding space as follows:

$$(h_{ij}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \beta^2 \end{pmatrix}. \tag{7}$$

We will see below that different limits of this ratio β interpolate between the flows that originate from the Euclidean L1 and L2 norms.

Using standard methods in variational calculus, the Euler–Lagrange equations with respect to the embedding are (see [42] for derivation)

$$-\frac{1}{2\sqrt{g}}\, h^{il}\, \frac{\delta S}{\delta X^l} = \frac{1}{\sqrt{g}}\, \partial_\mu(\sqrt{g} g^{\mu\nu}\partial_\nu X^i) + \Gamma^i_{jk}\partial_\mu X^j \partial_\nu X^k g^{\mu\nu} \tag{8}$$

where Γ^i_{jk} are the Levi–Civita connection coefficients with respect to the metric h_{ij} that describes the geometry of the embedding space (see [42], [44] for a definition of the Levi–Civita connection).

Our proposal is to view scale-space as the gradient descent

$$X^i_t \equiv \frac{\partial X^i}{\partial t} = -\frac{1}{2\sqrt{g}}\, h^{il}\frac{\delta S}{\delta X^l}. \tag{9}$$

A few remarks are in order. First, notice that we used our freedom to multiply the Euler–Lagrange equations by a strictly positive function and a positive definite matrix. This factor is the simplest one that does not change the minimization solution while giving a reparameterization invariant expression. This choice guarantees that the flow is geometric and does not depend on the parameterization. We will see below that the Perona–Malik flow, for example, corresponds to another choice of the prefactor, namely one. The operator that is acting on X^i in the first term of (8) is the natural generalization of the Laplacian from flat spaces to manifolds and is called *the second order differential parameter of Beltrami* [27], or in short *Beltrami operator*, and we will denote it by Δ_g. When the embedding is in a Euclidean space with Cartesian coordinate system the connection elements are zero. If the embedding space is not Euclidean or if the coordinate system we use is not Cartesian, we have to include the Levi–Civita connection term since it is no longer equal to zero.

In general for any manifolds Σ and M, the map $\mathbf{X}$: $\Sigma \to M$ that minimizes the action S with respect to the embedding is called a *harmonic map*. The harmonic map is the natural generalization of the geodesic curve and the minimal surface to higher dimensional manifolds and for different embedding spaces. We have here a framework that can treat curves, surfaces, and higher dimensional image data embedded in gray, color and higher dimensional and geometrically nontrivial embedding spaces.

III. The Beltrami Flow

In this section, we present a new and natural flow. The image is regarded as an embedding map $\mathbf{X} : \Sigma \to \mathbb{R}^n$, where Σ is a 2-D manifold. We treat grey-level and color images as examples and then compare to related works. Explicitly, the maps for grey-level and color images are

$$\mathbf{X} = [x(\sigma^1, \sigma^2),\, y(\sigma^1, \sigma^2),\, I(\sigma^1, \sigma^2)]$$

and

$$\mathbf{X} = [x(\sigma^1, \sigma^2),\, y(\sigma^1, \sigma^2),\, \{I^i(\sigma^1, \sigma^2)\}_{i=1}^3] \tag{10}$$

respectively. In the above map, we have denoted (r, g, b) by $(1, 2, 3)$ for convenience, or in general notation by i. We minimize our action in (4) with respect to the metric and with respect to (I^r, I^g, I^b). The coordinates x and y are parameters from this view point and are identified as usual with σ^1 and σ^2, respectively. We note that there are obviously better selections to color space definition rather than the red–green–blue (RGB) flat space. Nevertheless, we get good results even from this oversimplified assumption.

Minimizing the metric for 2-D manifolds gives, as we have seen, the induced metric which is given for grey-level image in (3) and for color images by

$$(g_{\mu\nu}) = \begin{pmatrix} 1 + \sum_{i=1}^3 (I^i_x)^2 & \sum_{i=1}^3 I^i_x I^i_y \\ \sum_{i=1}^3 I^i_x I^i_y & 1 + \sum_{i=1}^3 (I^i_y)^2 \end{pmatrix} \tag{11}$$

and $g = \det(g_{ij}) = g_{11}g_{22} - g_{12}^2$. Note that this metric differs from the Di Zenzo matrix [14] (which is not a metric since it is not positive definite) by the addition of 1 to g_{11} and g_{22}. The source of the difference lies in the map used to describe the image; Di Zenzo used $\mathbf{X} : \Sigma \to \mathbb{R}^3$ while we use $\mathbf{X} : \Sigma \to \mathbb{R}^5$.

The action functional under our choice of the metric is the Nambu functional

$$S = \int d^2\sigma \sqrt{\det(\partial_\mu X^i \partial_\nu X_i)} = \int d^2\sigma \sqrt{1 + \sum_i |\nabla I^i|^2 + \tfrac{1}{2}\sum_{ij}(\nabla I^i, \nabla I^j)^2} \quad (12)$$

where $(\nabla I^i, \nabla I^j)$ stands for the magnitude of the vector product of the vectors ∇I^i and ∇I^j. For grey-level images the last term vanishes and we are left with $S = \int d^2\sigma\sqrt{1+|\nabla I|^2}$. The action in (12) is simply the area of the image surface.

Now, we compare our norm to that proposed by Shah in [41]: $\int \sqrt{\sum_{i=1} |\nabla I^i|^2}$. We notice that the proposed area norm in (12), includes an extra term that does not appear in Shah's norm and other previous norms in the literature. The term $\sum_{ij}(\nabla I^i, \nabla I^j)^2$ measures the directional difference of the gradient between different channels. The minimization of a norm that includes this term, directs different channels to align together as they become smoother and simpler in scale. One should recognize this cross correlation of orientation between the channels as a very important feature; overcoming the color fluctuations along edges as a result of a lossy Joint Photographic Expert Group (JPEG) standard compression is a good example.

Minimizing (12) with respect to I^i gives the Beltrami flow

$$I_t^i = \frac{1}{\sqrt{g}}\,\partial_\mu(\sqrt{g}g^{\mu\nu}\partial_\nu I^i) \equiv \mathbf{H}^i. \quad (13)$$

It means that the velocity in the I^i direction is proportional to the component of the mean curvature vector in the I^i direction. Note the difference between (13) and the mean curvature flow in (20). Here, we only move the feature coordinates while keeping x and y fixed, where as in the mean curvature flow we move all coordinates. The projection of the mean curvature vector on the feature coordinates is an edge-preserving procedure. Intuitively it is obvious. Each point on the image surface moves with a velocity that depends on the mean curvature and the I^i components of the normal to the surface at that point. Since along the edges the normal to the surface lies almost entirely in the x–y plane, I^i hardly changes along the edges while the flow drives other regions of the image toward a minimal surface at a more rapid rate.

For a simple implementation of the Beltrami flow in color we first compute the following matrices: I_x^i, I_y^i, p^i, and q^i given by

$$p^i = g_{22}I_x^i - g_{12}I_y^i,$$
$$q^i = -g_{12}I_x^i + g_{11}I_y^i.$$

Then, the evolution is given by

$$I_t^i = \frac{1}{g}\left(p_x^i + q_y^i\right) - \frac{1}{2g^2}\left(g_x p^i + g_y q^i\right) \quad (14)$$

where $g_x = \partial_x g (g_y = \partial_y g)$.

For grey-scale case, we get the following expression after plugging the explicit form of $(g_{\mu\nu})$

$$I_t = \frac{(1+I_y^2)I_{xx} - 2I_xI_yI_{xy} + (1+I_x^2)I_{yy}}{(1+I_x^2+I_y^2)^2}. \quad (15)$$

Let us further explore the geometry of the flow and relate it to other known methods.

A. Geometric Flows Toward Minimal Surfaces

A minimal surface is the surface with least area that satisfies given boundary conditions. It has nice geometrical properties, and is often used as a natural model of various physical phenomena, e.g., soap bubbles "Plateau's problem," in computer-aided design, in structural design, and recently even for medical imaging [6]. It was realized by Lagrange in 1762 [28], that the mean curvature equal to zero is the Euler–Lagrange equation for area minimization. Hence, the mean curvature flow is the most efficient flow toward a minimal surface. Numerical schemes for the mean curvature flow, and the construction of minimal surfaces under constraints, where studied since the beginning of the modern age of numerical analysis [13], and is still the subject of ongoing numerical research [8], [11], [12].

For constructing the mean curvature flow of the image as a surface, we follow three steps.

1) Let the surface $\mathcal{S}$ evolve according to the geometric flow $(\partial\mathcal{S}/\partial t) = \vec{F}$, where $\vec{F}$ is an arbitrary smooth flow field. The geometric deformation of $\mathcal{S}$ may be equivalently written as $(\partial\mathcal{S}/\partial t) = \langle\vec{F}, \vec{\mathcal{N}}\rangle\vec{\mathcal{N}}$, where $\vec{\mathcal{N}}$ is the unit normal of the surface at each point, and $\langle\vec{F}, \vec{\mathcal{N}}\rangle$ is the inner product (the projection of $\vec{F}$ on $\vec{\mathcal{N}}$).
2) The mean curvature flow is given by $(\partial\mathcal{S}/\partial t) = H\vec{\mathcal{N}}$, where H is the mean curvature of $\mathcal{S}$ at every point.
3) Considering the image function $I(x, y)$, as a parameterized surface $\mathcal{S} = [x, y, I(x, y)]$, and using the relation in step 1, we may write the mean curvature flow as $(\partial\mathcal{S}/\partial t) = H/\langle\vec{\mathcal{N}}, \vec{z}\rangle\,\vec{z}$, for any smooth vector field $\vec{z}$ defined on the surface. Especially, we may choose $\vec{z}$ as the $\hat{I}$ direction, i.e., $\vec{z} = (0, 0, 1)$. In this case

$$\frac{1}{\langle\vec{\mathcal{N}}, \vec{z}\rangle} \cdot \vec{z} = \sqrt{1+I_x^2+I_y^2} \cdot (0, 0, 1) = \sqrt{g}(0, 0, 1). \quad (16)$$

Fixing the (x, y) parameterization along the flow, we have $\mathcal{S}_t = (\partial/\partial t)[x, y, I(x, y)] = [0, 0, I_t(x, y)]$. Thus, for tracking the evolving surface, it is enough to evolve I via $(\partial I/\partial t) = H\sqrt{1+I_x^2+I_y^2}$, where the mean curvature H is given as a function of the image I. (See Fig. 2 and (22); see [10] and [11] for the derivation of H, as Chopp summarizes the original derivation by Lagrange from 1762.)

Substituting the explicit equation for mean curvature, we end up with the equation

$$I_t = \frac{(1+I_y^2)I_{xx} - 2I_xI_yI_{xy} + (1+I_x^2)I_{yy}}{1+I_x^2+I_y^2} \quad (17)$$

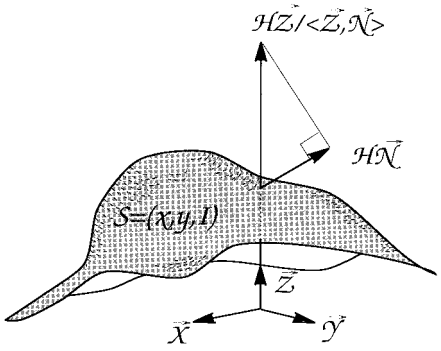

Fig. 2. Consider the surface mean curvature flow $\mathcal{S}_t = H\vec{\mathcal{N}}$, mean curvature H in the surface normal direction $\vec{\mathcal{N}}$. A geometrically equivalent flow is the flow $\partial(x, y, I)/\partial t = H(1 + |\nabla I|^2)^{1/2} \cdot (0, 0, 1)$ which yields the mean curvature flow when projected onto the normal.

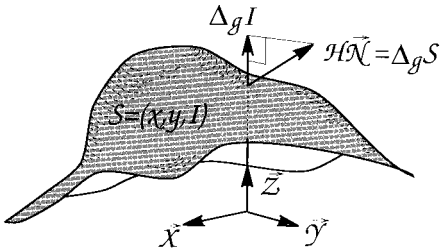

Fig. 3. Consider the mean curvature H in the surface normal direction $\vec{\mathcal{N}}$. It can also be expressed as $H\vec{\mathcal{N}} = \Delta_g\mathcal{S}$. Beltrami operator that operates on I: $\Delta_g I$, is the third component of this vector: Projection onto the I ($\vec{z}$) direction.

with the initial condition $I(x, y, t = 0) = I_0(x, y)$. Using the notation of Beltrami second order operator Δ_g and the metric g, (17) may be read as $I_t = g\Delta_g I$. This equation was studied in depth in [16] and [32], where the existence and uniqueness of weak solutions was proved under some mild conditions on the behavior of the curvature on the boundary and the smoothness of the initial condition. The Beltrami flow itself (selective mean curvature flow) $I_t = \Delta_g I$ is given explicitly for the simple 2-D case in (15). The difference between the two flows is the factor g. This factor has an important significance in keeping the flow geometrical, that is, it depends on geometrical objects and not on coordinates used in describing them. It also serves as an edge detector by behaving like an edge-preserving flow; see Fig. 3.

B. Related Works

In [17], the authors propose the following similar flow for grey-level images: $I_t = H = \sqrt{g}\Delta_g$. Geometrically, they rotate the curvature normal vector so that it coincides with the $\vec{z}$ axis. This equation was studied extensively by mathematicians [15], [19], [29], where the existence and uniqueness of weak solutions was discussed. It is located somewhere between the mean curvature flow for the image as a surface $I_t = g\Delta_g I = H\sqrt{g}$ that was used in [30] to denoise images, and our Beltrami flow, which in 2-D case simplifies to $I_t = \Delta_g I = H/\sqrt{g}$. All of the above flows lead toward a minimal surface, yet our proposed framework better preserves the edges, naturally extends to any number of dimensions, and is reparameterization invariant.

Let us show next the direct relation to TV methods [36] and especially for the regularization introduced by Vogel and Oman [45], and efficiently implemented for changing the regularization ratio (from large to small) in [8]. We will show that by modifying the aspect ratio between the intensity and the xy coordinates, we are able to switch between norms. It is possible to obtain the TV norm, travel through minimal surfaces, and end up with potential surfaces at the other limit.

The regularized TV is defined by $\min \int \sqrt{\beta^2 + |\nabla I|^2}$, where β is a real number, subject to constraints that are used to monitor the drifting of the evolving image away from the initial one. Contrast scaling of $I \to \beta I$, we have $\nabla I \to \beta \nabla I$ and the TV norm becomes $\int \sqrt{1 + |\nabla I|^2}$. This is exactly an area minimization toward a minimal surface that could be realized through mean curvature flow with constraints imposed by the noise variance and scale. In other words, the regularized TV is in fact a flow toward a minimal surface with respect to the scaled surface $(x, y, \beta I)$. The ratio between the image size (resolution) and the gray level is taken in an arbitrary way for creating an artificial Euclidean metric, therefore, setting this ratio to β brings us to the minimal surface computation. It is important to note that the β ratio should be determined for every image processing algorithm. The β ratio may be introduced via Polyakov action by defining the embedding metric h_{ij} to be as in (7). The only way to avoid the β ratio dependence is to construct planar curve evolution for the gray-level sets, such that embedding is preserved [1], [22], [39]. This was called "contrast invariance" in [2]. Yet, these schemes are pure smoothing schemes that do not preserve edges.

We note that it is possible to impose constraints on the functional that modify the flow like the variance constraints of the Rudin–Osher–Fatemi total variation (TV) method [36]. We have just shown that large β ratio leads to potential surfaces, while at small ratio we have the TV norm. We have thereby linked together many classical schemes via a selection of one parameter, that is, the image gray-level scale with respect to its xy coordinates. This scale is determined arbitrarily anyhow in most of the current schemes.

Note that for color images we have a different situation. First we can have three different regularization ratios one for each channel. Second, even when we take a common ratio for all channels, in the limit $\beta \to \infty$ we get an action that does not agree with the color TV [3], [37], [38], [41]. This can be seen easily by observing (12) where only the third term survives in this limit. This is the term that contributes the most for coupling between the color channels. The other limit $\beta \to 0$ gives a channel by channel linear diffusion.

Because of space limitations, we refer the reader to our papers [23], [25], [42], and [43] for comparison with other methods suggested recently for nonlinear color image processing, like [3], [7], [37], [38], and [41].

C. Beltrami Flow in Color Space: Results

We now present some results of denoising color images using our model. Spatial derivatives are approximated using central differences and an explicit Euler step is employed to reach the solution. We represent the image in the RGB

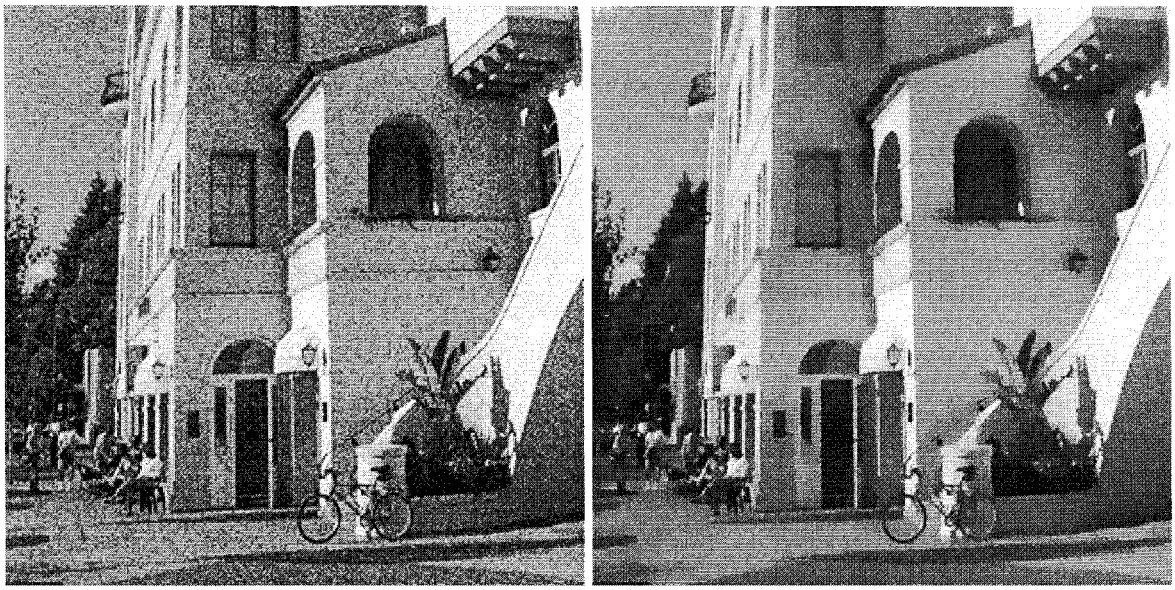

Fig. 4. Reconstruction of a color images corrupted with Gaussian noise (for the color version, refer to www.lbl.gov/~ron/belt-html.html).

Fig. 5. Reconstruction of an image that has been corrupted by JPEG compression algorithm (for the color version, refer to www.lbl.gov/~ron/belt-html.html).

space; however, other representations and different numerical schemes (as in [8]) are possible.

In the first example, we corrupt a given image with Gaussian noise and denoise it using our method. The left image in Fig. 4 shows an image corrupted with noise and the image on the right depicts its reconstruction. In the second example, we consider noise artifacts introduced by lossy compression algorithms such as JPEG. In Fig. 5, the left image shows a JPEG compressed image and the right image is its "corrected" version using our Beltrami flow.

IV. Choices that Lead to Known Methods

We will survey in this section different choices for the dynamic and parametric degrees of freedom in the action functional.

A. Linear Scale-Space

Recently, Florac *et al.* [18] invoked reparameterization invariance in vision. The basic motivation in their work is to give a formulation of the linear scale-space, which is based on the linear heat flow, that lends itself to treatment in different coordinate systems. They also noted on the possibility to use a nonflat metric, and raised the idea of using an image induced metric.

In order to have reparameterization invariance, one has to write an invariant differential operator. The simplest second-order invariant differential operator is the Beltrami operator. The major difference then between our approach and the one given in [18] is the class of metrics allowed. Since a change in parameterization cannot change the geometry of the problem, and since they are interested in a linear scale-space, they only

allow metrics for which the Riemann tensor vanishes, that is metrics of a flat space.

Our point of view is that an image is a surface embedded in $\mathbb{R}^n$ (or a more general Riemannian manifold). From this perspective the natural metric to choose is the induced metric of the surface. This metric is never flat for a significant image.

B. Generalized Perona–Malik Flows

We fix, as in the linear case, the xy coordinates and vary the action with respect to I while the metric is arbitrary for the time being. Using the Euler–Lagrange equation without any pre-factor, we get the following flow

$$I_t = \partial_\mu \sqrt{g} g^{\mu\nu} \partial_\nu I.$$

We assume now that the image is a d-dimensional manifold embedded in $\mathbb{R}^{d+1}$. The task at hand is to find the right choice of the metric to reproduce the Perona–Malik flow. We select $(g_{\mu\nu}) = \tilde{f}\mathcal{I}_d$, where $\mathcal{I}_d$ is the identity matrix. The determinant is $g = (\tilde{f})^d$, and consequently the flow becomes

$$I_t = \sum_{\mu=1}^{n} \partial_\mu \tilde{f}^{(d/2)-1} \partial_\mu I.$$

For any dimension different from two, we can choose $\tilde{f}^{(d/2)-1} = C(I)$ to get

$$I_t = \mathrm{div}[C(I)\nabla I]$$

which is the basic idea of Perona and Malik [33]. If we further specify $\tilde{f}^{(d/2)-1} = C(I) = [f(I_0)/|\nabla I|]$, where I_0 is the original image, we arrive at

$$I_t = \mathrm{div}\left[f(I_0) \frac{\nabla I}{|\nabla I|} \right]$$

which is the core (up to the $|\nabla I|$ normalization factor) of what is known in the literature as the geodesic active contours [4]–[6], [21], [40]. Note that this works only for dimensions different from two. Examples of higher dimensional manifolds in vision and image processing are 3-D images and movies as 3-D manifolds [26], 3-D movies as four-dimensional (4-D) manifolds, and texture as a 4-D manifold embedded in six-dimensional (6-D) space [24].

A simple way to get the 2-D Perona–Malik flow is to go one dimension higher: Imagine a map that is the embedding of a 3-D hyperplane as follows: $(x, y, z, I(x, y))$. Note that I depends only on x and y. Now choose a metric that is zero except the diagonal elements $(f^{-1}(x, y), f^{-1}(x, y), f^2(x, y))$, so that the determinant is 1 and the diagonal of the inverse metric matrix reads (f, f, f^2). Since both the metric and the intensity do not depend on z, then the derivative with respect to z vanishes and we get the 2-D Perona–Malik flow: $I_t = \partial_x(fI_x) + \partial_y(fI_y)$. In fact, f can depend on z since $\partial_z(f(x, y, z)I_z = 0$ if $I_z = 0$, so that we can indentify z with the parameter in the Perona–Malik diffusion function, e.g., $f = exp(-z(I_x + I_y)^2)$. Our approach gives the z and f a special form that has a well-defined geometrical meaning and it is derived from a minimization of an action functional.

C. The Mean Curvature Flow

In this subsection, we choose to minimize with respect to all the embedding variables in the action. We also choose the induced metric as the image metric.

Going back to the action in (4) and minimizing with respect to each one of the embedding coordinates X^i, we get the Euler–Lagrange equations (see [42] for derivation)

$$\frac{1}{\sqrt{g}} \partial_\mu (\sqrt{g} g^{\mu\nu} \partial_\nu X^i) + \Gamma^i_{jk} \partial_\mu X^j \partial_\nu X^k g^{\mu\nu} = 0. \tag{18}$$

We take the image metric to be $g_{\mu\nu} = \partial_\mu X^i \partial_\nu X^j h_{ij}$ which is by definition the induced metric. For the case of grey-level image (i.e., $\mathbf{X}$: $\Sigma \to R^3$), it is given explicitly in (3).

Substituting the induced metric in (9), we get the generalized mean curvature flow, namely

$$X^i_t = \frac{1}{\sqrt{g}} \partial_\mu (\sqrt{g} g^{\mu\nu} \partial_\nu X^i) + \Gamma^i_{jk} \partial_\mu X^j \partial_\nu X^k g^{\mu\nu} \equiv \mathbf{H}^i \tag{19}$$

where $\mathbf{H}$ is the mean curvature vector by definition [9], [44].

For embedding of a manifold in $\mathbb{R}^n$ with Cartesian coordinate system, the affine connection is identically zero and we are left with the Laplace–Beltrami operator

$$X^i_t = \frac{1}{\sqrt{g}} \partial_\mu (\sqrt{g} g^{\mu\nu} \partial_\nu X^i). \tag{20}$$

Plugging the explicit expression of the induced metric (3) for the case $\mathbf{X}$: $\Sigma \to R^3$, in the above equation, we obtain

$$\mathbf{X}_t = \mathbf{H} = H\vec{\mathcal{N}} \tag{21}$$

where $\mathbf{H}$ is the mean curvature vector[2] that can be written for surfaces as the mean curvature H times the unit normal to the surface $\vec{\mathcal{N}}$

$$H = \frac{(1 + I_x^2) I_{yy} - 2 I_x I_y I_{xy} + (1 + I_y^2) I_{xx}}{g^{\frac{3}{2}}}$$
$$\vec{\mathcal{N}} = \frac{1}{\sqrt{g}} (-I_x, -I_y, 1)^T \tag{22}$$

where $g = 1 + I_x^2 + I_y^2$.

The fact that this choice gives us the mean curvature flow should not be a surprise, since if we check how the choice of metric $g_{\mu\nu}$ effects the action functional, we notice that

$$S = \int d^2\sigma \sqrt{g} = \int d^2\sigma \sqrt{\det(\partial_\mu X^i \partial_\nu X^j h_{ij})} \tag{23}$$

which is the Euler functional that describes the area of the surface (also known in high-energy physics as the *Nambu action*). The geometrical meaning of this flow is evident. Each point of the surface moves in the direction of the normal with velocity proportional to the mean curvature. If the embedding space is not Euclidean or if we use a non-Cartesian coordinate system, we have to use the more general flow, (19). In this way, we generalize the mean curvature flow to any dimension, co-dimension, and geometry.

[2]Note that some definitions of the mean curvature include a factor of two that we omit in our definition.

V. Concluding Remarks

Inventing a perceptually good smoothing process which is compatible with a segmentation process, and formulating a meaningful scale space for images is not an easy task, and is actually what low level vision research is about. Here we tried to address these questions and to come up with a new framework that both introduces new procedures and unifies many previous results. There are still many open questions to be asked, like what is the right aspect ratio between the intensity and the image plane? Or in a more general sense, a deeper question that both the fields of string theory and computer vision try to answer, is what is the "right" embedding space h_{ij}?

The question of what is the "right norm" when dealing with images is indeed not trivial, and the right answer probably depends on the application. For example, the answer for the "right" color metric h_{ij} is the consequence of empirical results, experimental data, and the application. Here we covered some of the gaps between the two classical L_p norms in a geometrical way and proposed a new approach to deal with multidimensional images. We used recent results from high-energy physics that yield promising algorithms for enhancement, segmentation, and scale space.

Acknowledgment

The authors thank D. Adalsteinsson and K. Bardakçi for interesting discussions, and D. Marimont for supplying the color images.

References

[1] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel, "Axioms and fundamental equations of image processing," *Arch. Rational Mechan.,* vol. 123, pp. 199–257, 1993.

[2] L. Alvarez and J. M. Morel, "Morphological approach to multiscale analysis: From principles to equations," in *Geometric-Driven Diffusion in Computer Vision,* B. M. ter Haar Romeny, Ed. Boston, MA: Kluwer, 1994.

[3] P. Blomgren and T. F. Chan, "Color TV: Total variation methods for restoration of vector valued images," *Cam TR,* Univ. Calif., Los Angeles, CA, 1996.

[4] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," in *Proc. ICCV'95,* Boston, MA, June 1995, pp. 694–699.

[5] ———, "Geodesic active contours," *Int. J. Comput. Vis.,* vol. 22, pp. 61–79, 1997.

[6] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert, "Minimal surfaces: A geometric three dimensional segmentation approach," *Numer. Math.,* vol. 77, pp. 423–451, 1997.

[7] A. Chambolle, "Partial differential equations and image processing," in *Proc. IEEE ICIP,* Austin, TX, Nov. 1994.

[8] T. F. Chan, G. H. Golub, and P. Mulet, "A nonlinear primal-dual method for total variation-based image restoration," in *AMS/SIAM Workshop Linear and Nonlinear CG Methods,* July 1995.

[9] I. Chavel, *Riemanian Geometry: A Modern Introduction*. Cambridge, U.K.: Cambridge Univ. Press, 1993.

[10] D. L. Chopp, "Computing minimal surfaces via level set curvature flow," Ph.D. dissertation, Lawrence Berkeley Lab. and Dept. Math. LBL-30685, Univ. Calif., Berkeley, CA, May 1991.

[11] ———, "Computing minimal surfaces via level set curvature flow," *J. Computat. Phys.,* vol. 106, pp. 77–91, May 1993.

[12] D. L. Chopp and J. A. Sethian, "Flow under curvature: Singularity formation, minimal surfaces, and geodesics," *J. Exper. Math.,* vol. 2, pp. 235–255, 1993.

[13] P. Concus, "Numerical solution of the minimal surface equation," *Math. Computat.,* vol. 21, pp. 340–350, 1967.

[14] S. Di Zenzo, "A note on the gradient of a multi image," *Comput. Vis., Graph., Image Processing,* vol. 33, pp. 116–125, 1986.

[15] K. Ecker, "Estimates for evolutionary surfaces of prescribed mean curvature," *Math. Zeitschr.,* vol. 180, pp. 179–192, 1982.

[16] K. Ecker and G. Huisken, "Mean curvature motion of entire graphs," *Ann. Math.,* vol. 130, pp. 453–471, 1989.

[17] A. I. El-Fallah, G. E. Ford, V. R. Algazi, and R. R. Estes, "The invariance of edges and corners under mean curvature diffusions of images," in *Proc. Processing III SPIE,* 1994, vol. 2421, pp. 2–14.

[18] L. M. J. Florack *et al.*, "Nonlinear scale-space," in *Geometric-Driven Diffusion in Computer Vision,* B. M. ter Haar Romeny, Ed. Boston, MA: Kluwer, 1994.

[19] C. Gerhardt, "Evolutionary surfaces of prescribed mean curvature," *J. Differen. Geom.,* vol. 36, pp. 139–172, 1980.

[20] W. E. L. Grimson, *From Images to Surfaces*. Cambridge, MA: MIT Press, 1981.

[21] S. Kichenassamy *et al.*, "Gradient flows and geometric active contour models," in *Proc. ICCV'95,* Boston, MA.

[22] R. Kimmel, "Intrinsic scale space for images on surfaces: The geodesic curvature flow," in *Lecture Notes in Computer Science: First Int. Conf. Scale-Space Theory in Computer Vision,* Springer-Verlag, 1997, vol. 1252, pp. 212–223.

[23] ———, "A natural norm for color processing," in *Proc. 3rd Asian Conf. Computer Vision*, Hong Kong, Jan. 8–11, 1998.

[24] R. Kimmel, N. Sochen, and R. Malladi, "On the geometry of texture," UC-405, Lawrence Berkeley Labs. Rep. LBNL-39640, Univ. Calif., Berkeley, CA, Nov. 1996.

[25] ———, "From high energy physics to low level vision," in *Lecture Notes in Computer Science: First Int. Conf. Scale-Space Theory in Computer Vision,* Springer-Verlag 1997, vol. 1252, pp. 236–247.

[26] ———, "Images as embedding maps and minimal surfaces: Movies, color, and volumetric medical images," in *Proc. IEEE CVPR'97,* June 1997, pp. 350–355.

[27] E. Kreyszing, *Differential Geometry*. New York: Dover, 1991.

[28] J. L. Lagrange, *Essai d'une nouvelle méthode pour déterminer les maxima et les minima des formules intégrales indéfinies,* vol. I, D. J. Struick, Trans. Paris, France: Gauthier-Villars.

[29] A. Lichnewski and R. Temam, "Pseudo solutions of the time dependent minimal surface problem," *J. Differ. Equ.,* vol. 30, pp. 340–364, 1978.

[30] R. Malladi and J. A. Sethian, "Image processing: Flows under min/max curvature and mean curvature," *Graph. Models Image Processing,* vol. 58, pp. 127–141, Mar. 1996.

[31] D. Marr, *Vision*. San Francisco, CA: Freeman, 1982.

[32] V. I. Oliker and N. N. Urlatseva, "Evolution of nonparametric surfaces with speed depending on curvature ii. The mean curvature case," *Commun. Pure Appl. Math.,* vol. 46, pp. 97–135, 1993.

[33] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 12, pp. 629–639, 1990.

[34] A. M. Polyakov, "Quantum geometry of bosonic strings." *Phys. Lett.,* vol. 103B, pp. 207–210, 1981.

[35] B. M. ter Haar Romeny, Ed., *Geometric-Driven Diffusion in Computer Vision*. Boston, MA: Kluwer, 1994.

[36] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D,* vol. 60, pp. 259–268, 1992.

[37] G. Sapiro, "Vector-valued active contours," in *Proc. IEEE CVPR'96,* pp. 680–685.

[38] G. Sapiro and D. L. Ringach, "Anisotropic diffusion of multivalued images with applications to color filtering," *IEEE Trans. Image Processing,* vol. 5, pp. 1582–1586, 1996.

[39] G. Sapiro and A. Tannenbaum, "Affine invariant scale-space," *Int. J. Comput. Vis.,* vol. 11, pp. 25–44, 1993.

[40] J. Shah, "A common framework for curve evolution, segmentation and anisotropic diffusion," in *Proc. IEEE CVPR'96,* pp. 136–142.

[41] ———, "Curve evolution and segmentation functionals: Application to color images," in *Proc. IEEE ICIP'96,* pp. 461–464.

[42] N. Sochen, R. Kimmel, and R. Malladi, "From high energy physics to low level vision," Lawrence Berkeley Labs. Rep. 39243, Univ. Calif., Berkeley, Aug. 1996.

[43] N. Sochen and Y. Y. Zeevi, "Representation of images by surfaces and higher dimensional manifolds in non-Euclidean space," in *Proc. Mathematical Methods for Curves and Surfaces II,* M. Daehlen, T. Lyche, and L. L. Schumaker, Eds., in press.

[44] M. Spivak, *A Comprehensive Introduction to Differential Geometry*. Berkeley, CA: Publish or Perish, 1979.

[45] R. Vogel and M. E. Oman, "Iterative methods for total variation denoising," *SIAM. J. Sci. Comput.,* 1996, vol. 17, pp. 227–238, 1996.

[46] S. D. Yanowitz and A. M. Bruckstein, "A new method for image segmentation," *Comput. Vis., Graph., Image Processing,* vol. 46, pp. 82–95, 1989.

[47] A. Yezzi, "Modified curvature motion for image smoothing and enhancement," this issue, pp. 345–352.

Nir Sochen was born 1962 in Israel. He received the B.Sc degree in physics in 1986 and the M.Sc. degree in theoretical physics in 1988, both from the University of Tel-Aviv, Israel, and the Ph.D. degree in theoretical physics in 1992 from the Université de Paris-Sud, Paris, France. During his studies, he conducted research at the Service de Physique Théorique at the Centre d'Etude Nucleaire, Saclay, France.

He spent one year conducting research at the Ecole Normale Superieure, Paris, on the Haute Etude Scientifique Fellowship, and three years with the Physics Department, University of California, Berkeley, on an NSF fellowship. He spent 1997 as a Research Fellow in the Physics Department, University of Tel-Aviv. He is currently a Senior Research Fellow in the Faculty of Electrical Engineering, Technion—Israel Institute of Technology, Haifa. His prior research interests were in quantum field theories and integrable models related to high-energy physics and string theory; his current interests are in computer vision and image processing.

Ron Kimmel was born in Haifa, Israel, in 1963. He received the B.Sc. (cum laude) in computer engineering (computer science and electrical engineering), the M.Sc. degree, and the D.Sc. degree, all from the Technion—Israel Institute of Technology, Haifa, in 1986, 1993, and 1995, respectively.

During 1986 to 1991 he served as an R&D officer in the Israeli Air Force. He is currently a post-doctoral fellow at Lawrence Berkeley National Laboratory and the Mathematics Department, University of California, Berkeley. His research interests are in computational methods and their applications, including topics in differential geometry, numerical analysis, nonlinear image processing, geometric methods in computer vision, and numerical geometry methods in computer-aided design, robotic navigation, and computer graphics.

Dr. Kimmel was awarded the HTI Post-doctoral Fellowship, and the Wolf, Gutwirth, Ollendorff, and Jury fellowships during his graduate studies.

Ravikanth Malladi received the B.Eng. degree (honors) in electrical engineering and the M.Sc. degree in physics from Birla Institute of Technology and Science, India, in 1988, and the M.S. degree in 1991 and the Ph.D. degree in 1993, both in computer vision, from the University of Florida, Gainesville.

In the summer of 1985, he worked at the National Geophysical Research Institute, India. During the spring of 1988, he held a Research Assistant position with the semiconductor physics group in the Central Electrical and Electronics Research Institute, India. At the University of Florida, he was a Research Assistant in the Center for Computer Vision and Visualization. From spring 1994 to summer 1995, he was a post-doctoral fellow at Department of Mathematics and Lawrence Berkeley National Laboratory, University of California, Berkeley. He is currently a Staff Computer Scientist at the Lawrence Berkeley National Laboratory. His research interests are in image processing, computational vision, shape modeling and recognition, computer graphics, and medical image interpretation.

Dr. Malladi was awarded the NSF Post-doctoral Fellowship in Computational Science and Engineering in June 1994, and his work received best peer reviews at the Third European Conference on Computer Vision, Stockholm, Sweden, May 1994.