

Boundary Detection Using Simulation of Particle Motion in a Vector Image Field

Nawapak Eua-Anant, *Student Member, IEEE*, and Lalita Udpa, *Senior Member, IEEE*

Abstract—This paper introduces a novel approach in image processing based on a vector image model. A major advantage of the model is that it allows vector operations to be performed on an image. An example of a vector operation is the computation of mechanical moments for detecting inhomogeneities in an object or equivalently edges in an image. A new edge operator derived from a vector image model yields an edge vector field analogous to the Hamiltonian gradient field of the image. The distinct feature of the edge vector field is that edge vectors form current loops encompassing the objects. This feature is exploited to develop a new boundary extraction algorithm based on particle motion in a force field. The edge vector field forces a particle to move along the edges while an orthogonal normalized Laplacian-gradient vector field guarantees that the particle will not drift away from the edges. The object boundary can be obtained from the convergent path of the particle trajectory. Using a fine stepping factor, the extracted boundary can achieve subpixel accuracy. The proposed algorithm has major advantages over the conventional edge-detection, edge-thinning, and edge-linking techniques in that it effectively utilizes both direction and magnitude of edges. The algorithm is simple, robust and performs very well even on high curvature objects.

Index Terms—Boundary extraction, edge operator, image moment, particle motion, vector model.

I. INTRODUCTION

IMAGE segmentation procedures can be broadly divided into three classes, namely, pixel-oriented, region-oriented, and edge-oriented algorithms. Pixel and region-oriented schemes segment the image based on similarity of pixel attributes within regions while edge-oriented segmentation, based on discontinuity between regions, focuses on extracting edges or boundaries rather than regions. The direct extraction of boundaries has a significant advantage over region extraction in that the number of parameters involved in the process is reduced. The disadvantage however is the possibility of getting incomplete or open boundaries. The most common approach in edge-oriented image segmentation is to first detect the object edges using an edge operator and then extract boundaries using the edge information.

In a monochrome image, edges are associated with abrupt changes in intensity distribution and edge detection techniques are employed to detect the existence of these transitions. Such changes can be detected in the form of extrema and zero

crossings of image derivatives. Derivative-based edge operators [1]–[6] which are convolutions of a scalar image with difference masks are simple and widely used for edge detection largely due to the fact that they do not require prior knowledge of the curvature of object boundaries. Each technique differs from the others only in the order and configuration of the masks. The first order derivative edge operator is specially designed for detecting a step edge because the extremum of the first order derivative occurs at an ideal step edge. Other attempts to develop the first order derivative edge operators capable of detecting multiscale edges and suppressing noise have been proposed using large, smooth and size-adjustable difference masks [4]–[10].

The next step following edge detection in edge-oriented image segmentation is boundary extraction. Many boundary models have been proposed such as Markovian [12], [13], rational Gaussian [14], and spline-based models [15]–[17]. In model-based techniques, the regularity of boundaries is controlled by parameters of the model. The model interacts with the image in a dynamical manner adjusting its parameters to maximize the joint likelihood of the image data and estimated boundary [12]–[14]. An alternate approach for finding the optimal boundary is based on minimization of an energy function composed of internal forces with constraints of rigidity and elasticity of the contour [15]–[17]. Beside edge models, an alternative edge-oriented segmentation technique based on a system of particles was presented by Dayanand *et al.* [19]. The particle driven by Newtonian attractive forces from data elements, i.e., edge pixels, follows edge contours creating a spiral contour around an object. A fusion algorithm is then needed to get a closed contour.

This paper describes a novel concept in image processing for image modeling and boundary extraction. Conventional edge operators are based on a scalar model of an image which is not totally consistent with image formation. An image is formed by an array of nearly parallel light beams propagating directly to our eyes. This in turn forms an image plane orthogonal to the line of sight. In accordance with image formation, an image is modeled in this paper as an array of intensity vectors perpendicular to an image plane. A vector image model provides a more complete description of an image in which both direction and magnitude of an image element are considered. From a vector image model, we develop a new derivative-based edge operator that yields the edge vector field analogous to the Hamiltonian gradient field of the image. In contrast to the gradient vector field obtained from conventional derivative-based edge operators, the edge vector field forms

Manuscript received April 10, 1998; revised February 11, 1999. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Robert J. Schalkoff.

The authors are with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 (e-mail: nawapak@iastate.edu; lalita@iastate.edu).

Publisher Item Identifier S 1057-7149(99)08760-6.

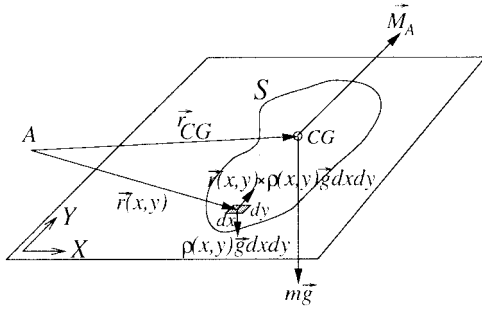


Fig. 1. Moment due to the weight of a planar object S with respect to point A .

current loops encompassing the object in the image. In the second step, a novel technique for boundary extraction is derived from the concept of particle motion in a force field. The edge vector field and Laplacian-gradient vector field are used as tangential and orthogonal force fields that drive a particle to describe a trajectory along the object boundary. The proposed algorithm effectively utilizes both direction and magnitude of the first and second order image derivatives to yield estimates of object boundaries with subpixel accuracy.

After a review of moment-based edge operator, a vector image model and the proposed edge operator are described in the first part of the paper. In the second part, the model of particle motion in a force field and its application to boundary extraction are discussed. Results of implementing the algorithm on objects with complex boundaries are presented.

II. PART I: A NEW EDGE OPERATOR BASED ON A VECTOR IMAGE MODEL

A. Moments for Detecting Inhomogeneity

In mechanics, the moment of a force on an object about a point is a measure of its tendency to turn or rotate the object about the point. In a right hand coordinate system, the moment \vec{M} of a force \vec{F} with respect to an observation point A can be expressed in terms of the cross product

$$\vec{M} = \vec{r} \times \vec{F} \quad (1)$$

where \vec{r} is a distance vector from point A to a point on the line of action of \vec{F} . The direction of \vec{M} is along the direction of the axis of rotation.

For a planar object S lying on the XY plane in a constant gravity field \vec{g} as shown in Fig. 1, the moment, due to its weight, with respect to the observation point A is calculated as

$$\vec{M}_A = \iint_S \vec{r}(x,y) \times \rho(x,y) \vec{g} dx dy \quad (2)$$

where $\vec{r}(x,y)$ is the distance vector from point A to point (x,y) and $\rho(x,y)$ is the object density at point (x,y) . Equivalently, (2) can be written as

$$\vec{M}_A = \vec{r}_{CG} \times \iint_S \rho(x,y) \vec{g} dx dy \quad (3)$$

$$= \vec{r}_{CG} \times m \vec{g} \quad (4)$$

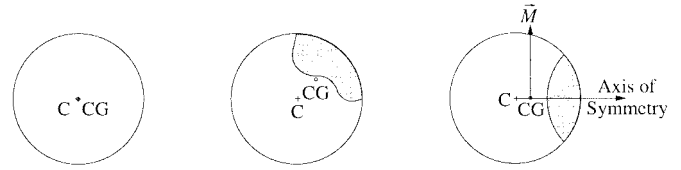


Fig. 2. Relation between the centroid (C) and the center of gravity (CG) of homogeneous and inhomogeneous objects.

where \vec{r}_{CG} is the distance vector from point A to the center of gravity CG of the object and m is the net object mass. From (2) through (4), the center of gravity of S can be computed directly by

$$CG = \frac{\iint_S \rho(x,y) \vec{r}(x,y) dx dy}{\iint_S \rho(x,y) dx dy}. \quad (5)$$

When the density ρ is constant, the geometrical center C (centroid) of the object S can be determined as

$$C = \frac{\iint_S \vec{r}(x,y) dx dy}{\iint_S dx dy}. \quad (6)$$

The centroidal moment \vec{M}_C is defined as the moment due to the object weight with respect to its centroid, i.e.

$$\vec{M}_C = \vec{r}_{CCG} \times \iint_S \rho(x,y) \vec{g} dx dy \quad (7)$$

where \vec{r}_{CCG} is the displacement vector from C to CG .

The usefulness of moments for detecting an inhomogeneity inside an object comes from the relation between the center of gravity and the centroid of an object. In particular, in the case of a homogeneous object of uniform density as illustrated in Fig. 2(a), the center of gravity coincides with the centroid of the object. As a result, the centroidal moment of the object vanishes. In contrast, in the case of an inhomogeneous object having no more than one axis of symmetry as shown in Fig. 2(b) and (c), the center of gravity is shifted from the centroid toward the high density area resulting in a nonvanishing centroidal moment. We can therefore conclude that the existence of the centroidal moment due to the object weight indicates presence of an inhomogeneity in the object. In addition, if an object has only one axis of symmetry as shown in Fig. 2(c), the center of gravity of the object will lie on an axis of symmetry and the centroidal moment will be parallel to the direction of an edge at the boundary between the two regions.

The concept of detecting the inhomogeneity in an object using the centroidal moment can be applied to an image where the underlying area in an image is considered to be a planar object with density distribution given by the pixel intensity distribution [7]–[9]. An area including edges is inhomogeneous and is thus characterized by a nonzero centroidal moment.

B. Vector Model for Images

In order to perform the vector cross product operation for calculating centroidal moments of areas in an image, a vector model for an image based on the image formation process is proposed [10] where each element of an image represents an

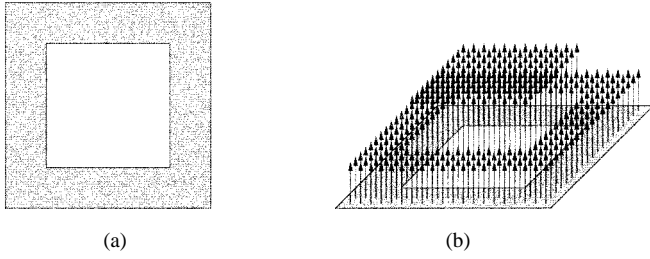


Fig. 3. (a) Scalar and (b) vector image models.

intensity vector normal to the image plane. The entire image is modeled as an array of intensity vectors, denoted as

$$\vec{I}(x, y) = I(x, y) \cdot \vec{n} \quad (8)$$

where \vec{n} is a normal to the image plane and $I(x, y)$ is the light intensity of pixel (x, y) . In contrast to conventional scalar models, a vector model of an image offers full description of an image element in terms of both magnitude and direction and is illustrated in Fig. 3. Since directions of all elements are constant, both scalar and vector models contain the same information. However, the main advantage of a vector model over a scalar model is the potential for performing both vector and scalar operations on the image data.

C. Image Moment

Making use of a vector image model, the moment of a neighborhood N of a pixel (i, j) yields the image moment vector (IMV) defined as

$$\vec{M}(i, j) = \sum_{(x, y) \in N} \vec{r}(x, y) \times \vec{I}(x, y) \quad (9)$$

where $\vec{r}(x, y)$ is a distance vector from point (i, j) to a point (x, y) in N . In order to make the image moment in (9) isotropic, a directional invariance property is incorporated by using a neighborhood N defined by a circle of radius R centered at point (i, j) . The array of image moment vectors at every pixel forms the image moment vector field.

The vector cross product term in (9) can be divided into x and y -components as

$$M_x(i, j) = \sum_{(x, y) \in N} (y - j) \cdot I(x, y) \quad (10)$$

$$= -G_y * I(x, y) \quad (11)$$

and

$$M_y(i, j) = \sum_{(x, y) \in N} -(x - i) \cdot I(x, y) \quad (12)$$

$$= G_x * I(x, y) \quad (13)$$

where

$$G_x(x, y) = \begin{cases} x, & \text{for } x^2 + y^2 \leq R^2 \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

and

$$G_y(x, y) = \begin{cases} y, & \text{for } x^2 + y^2 \leq R^2 \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

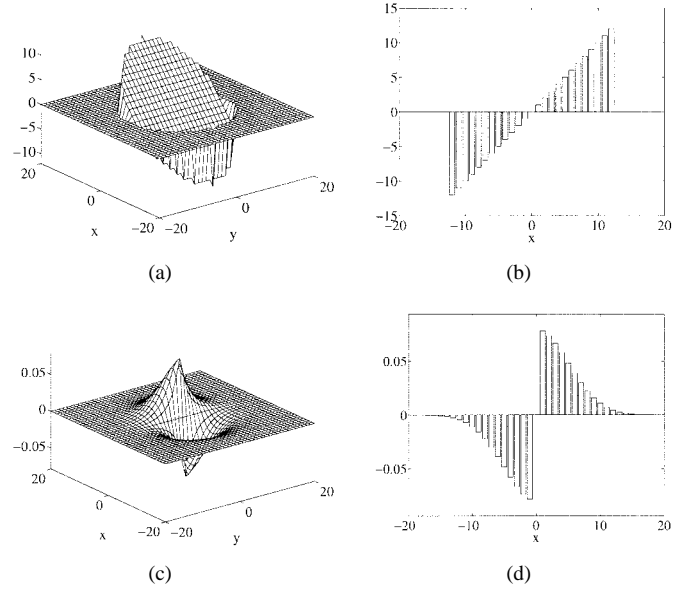


Fig. 4. (a) Difference mask profile of the IMV operator with $R = 12$ in x -direction. (b) Cross section at $y = 0$. (c) Directional difference mask profile of the GWIMV operator with $\sigma = 5$ in x -direction, and (d) its cross section at $y = 0$.

In other words, each component is a convolution between the image and the directional difference mask, G_x and G_y . Coefficients of each directional difference mask are distances in the mask direction from the center of the mask to neighboring points as illustrated in Fig. 4(a) and (b). Note that the sum of the coefficients in G_x and G_y are zero.

Except for the cross product in (9), the image moment vector operator is similar to the moment-based edge operator proposed by Machuca and Gilbert [7]. In most conventional moment-based edge operators, the importance of direction of a moment is not emphasized and, in fact, is discarded. Conventionally, moments are directed along the displacement between the centroid and the center of mass of a local area which is orthogonal to the direction of an image moment vector defined in this paper. The importance of the direction of IMV in (9) is seen in the boundary extraction algorithm [10], described in Part II of this paper. It is also important to mention that a region corrupted by uniformly distributed noise behaves like a homogeneous region. Consequently, from (11) through (13), the contribution of noise to the image moment is averaged out. This property makes the edge detection scheme using the IMV operator robust in the presence of noise. A major factor that affects noise suppression performance of the IMV edge operator is the size of the neighborhood and is discussed later.

An interesting feature of the masks shown in Fig. 4(a) is its contradiction to the behavior of the human visual system. Due to limited field of view, human visual system concentrates more on areas in the proximity of the point of interest than distant points. In other words, in the perceptive field, areas closer to the observation point are more significant than areas farther away. In contrast, in the image moment formula in (9), each data point is weighted in proportion to its distance from the center of the neighborhood. In other words, pixels farther

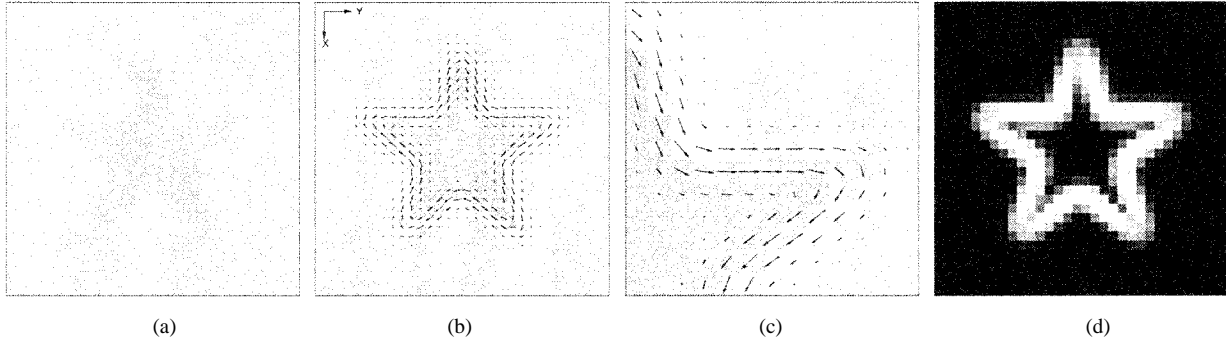


Fig. 5. (a) Original image. (b) Edge vector map with $\sigma = 1$. (c) Enlarged view. (d) Edge image.

away from the observation point are weighted higher than closer pixels. This characteristic yields poor edge localization performance of the IMV operator. To resolve this conflict and improve the edge localization performance, coefficients of the operator are rearranged in such a way that coefficients close to the center of the mask are large and then smoothly decrease with distance from the center. To satisfy this requirement, a weighting function $w(r)$ that decreases with r is used to get

$$\vec{M}(i, j) = \sum_{(x, y) \neq (i, j)} w(|\vec{r}(x, y)|) \cdot \frac{\vec{r}(x, y) \times \vec{I}(x, y)}{|\vec{r}(x, y)|} \quad (16)$$

where $|\vec{r}(x, y)|$ is the magnitude of $\vec{r}(x, y)$. The denominator in (16) is applied to neutralize the effect of the distance function on the image moment. Equation (16) can be considered as the generalized directional difference edge operator with its difference masks in x and y -directions given by

$$G_x = w(\sqrt{x^2 + y^2}) \cdot \frac{x}{\sqrt{x^2 + y^2}} \quad (17)$$

and

$$G_y = w(\sqrt{x^2 + y^2}) \cdot \frac{y}{\sqrt{x^2 + y^2}}. \quad (18)$$

Quantitative measures of the edge localization property of these edge detection operators can be calculated analytically using the method by Jain [11].

D. Gaussian Weighted Image Moment Vector Operator

Based on the study of the human visual system and the raw primal sketch [3], a suitable choice for the weighting function in (16) is the Gaussian function

$$w(r) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (19)$$

where σ is the spread constant. The Gaussian weighted image moment vector operator (GWIMV) [10] obtained from (16) using the Gaussian function is expressed in terms of x and y -components as

$$M_x(i, j) = -G_y * I(x, y) \approx \frac{\partial I(x, y)}{\partial y} \quad (20)$$

and

$$M_y(i, j) = G_x * I(x, y) \approx \frac{\partial I(x, y)}{\partial x} \quad (21)$$

where difference masks G_x and G_y are

$$G_x(x, y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \cdot \frac{x}{\sqrt{x^2 + y^2}} \quad (22)$$

and

$$G_y(x, y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \cdot \frac{y}{\sqrt{x^2 + y^2}}. \quad (23)$$

The difference mask of the GWIMV operator with $\sigma = 5$ in x -direction (G_x) and its cross section at $y = 0$ are illustrated in Fig. 4(c) and (d), respectively. In the one-dimensional (1-D) case, the difference mask of the GWIMV operator is similar to that of the Argyle operator [5].

The normalized field using a normalization constant k

$$\vec{e}(i, j) = \frac{1}{k} (M_x(i, j)\vec{i} + M_y(i, j)\vec{j}) \quad (24)$$

is called the edge vector field and the corresponding magnitude yields the edge image. Regardless of the scale σ , the edge vector field approximates the Hamiltonian gradient field of an image. Examples of the original image, edge vector field and edge image obtained using the GWIMV operator are displayed in Fig. 5. While the gradient field obtained from the conventional gradient-based edge operators displays steepness of an intensity surface, the edge vector field given by the GWIMV operator, as shown in Fig. 5(b), represents a map of “edge currents” flowing around an object. With the edge vector as a driving force, a particle immersed in edge vector streams will therefore be pushed along the edges. This idea is exploited for boundary extraction [10].

III. PART II: BOUNDARY EXTRACTION ALGORITHM

A. Particle in a Force Field

Consider a system of dimensionless particles in a force field, governed by the law of motion. In general, the trajectory of a particle can be determined from the 4-tuple $(\vec{P}, m, \vec{v}, \vec{F})$ representing its position, mass, velocity, and the driving force, respectively. A simulation model of a particle in a force field can be used to calculate and update the position of a particle at each incremental time step. Using a suitable interval, the trajectory and motion of a particle can be estimated. In this paper, a simplified model of a particle in a force field is proposed where the velocity and mass of the particle are

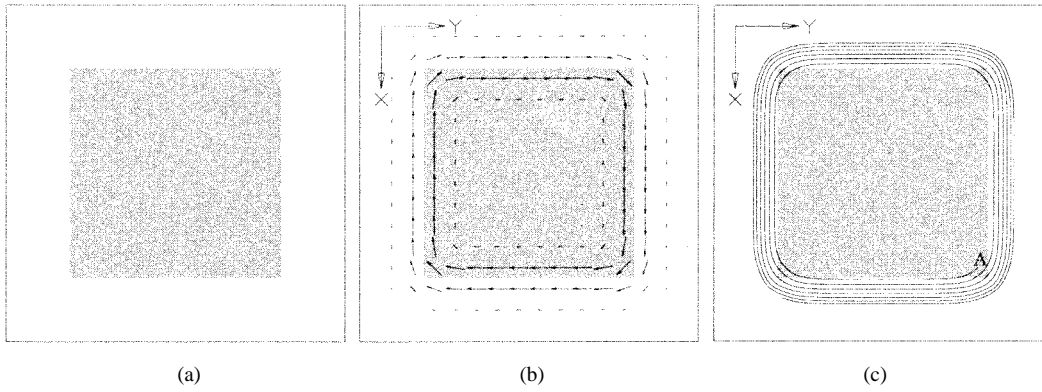


Fig. 6. (a) Image of a square. (b) Edge vector field with $\sigma = 1$. (c) Trajectory of the particle in the edge vector field with $\alpha = 0.1$ after 4000 time steps.

ignored and the position of a particle at a time instant t is assumed to depend only on the position and force at time $t - \Delta t$. The position vector \vec{P}_k of a particle at the time step k is therefore computed using the recursive equation:

$$\vec{P}_k = \vec{P}_{k-1} + \alpha \vec{F}_{k-1} \quad (25)$$

where \vec{F}_{k-1} is the force at the previous position \vec{P}_{k-1} , and α , ($\alpha \neq 0$), is the factor that determines the convergence rate of the model. In order to obtain a continuous trajectory, both \vec{P} and \vec{F} are required to be continuous in space. However, when dealing with a digital image where the field $\vec{F}(i, j)$ is observable only at discrete spatial points (i, j) , a bilinear interpolation scheme is used to obtain a continuous field value $\vec{F}(x, y)$ as follows:

$$\begin{aligned} \vec{F}(x, y) = & \vec{F}(i, j)(i+1-x)(j+1-y) \\ & + \vec{F}(i, j+1)(i+1-x)(y-j) \\ & + \vec{F}(i+1, j)(x-i)(j+1-y) \\ & + \vec{F}(i+1, j+1)(x-i)(y-j) \end{aligned} \quad (26)$$

where i and j are the integer parts of x and y , respectively.

B. Simulation of Particle Motion in an Edge Vector Field

In order to simulate the behavior of a particle in an edge vector field, vector \vec{F} in (25) is replaced by the edge vector \vec{e} derived in (24). In the simulation of particle motion, the particle is initially placed at the maximum gradient point in the image. The successive positions of the particle are then calculated using (25). Fig. 6 shows (a) the image of size 16×16 pixels with a square of size 10×10 pixels, (b) edge vector field and (c) the trajectory of the particle in the edge vector field with $\alpha = 0.1$ after 4000 time steps. As shown in Fig. 6(c), starting from an initial point A , the trajectory of the particle grows parallel to edges of the square with gradually increasing orbits. This phenomenon is directly influenced by the current loop feature in the edge vector field. Along the sides, the particle moves in the constant orbit. However, as it approaches the corners of the square, the particle deviates to the larger orbit resulting in a spiral trajectory pattern encompassing the object. Since the magnitude of the edge vector field reduces with the distance from object edges, the particle moves slower in the outer orbits. As the time

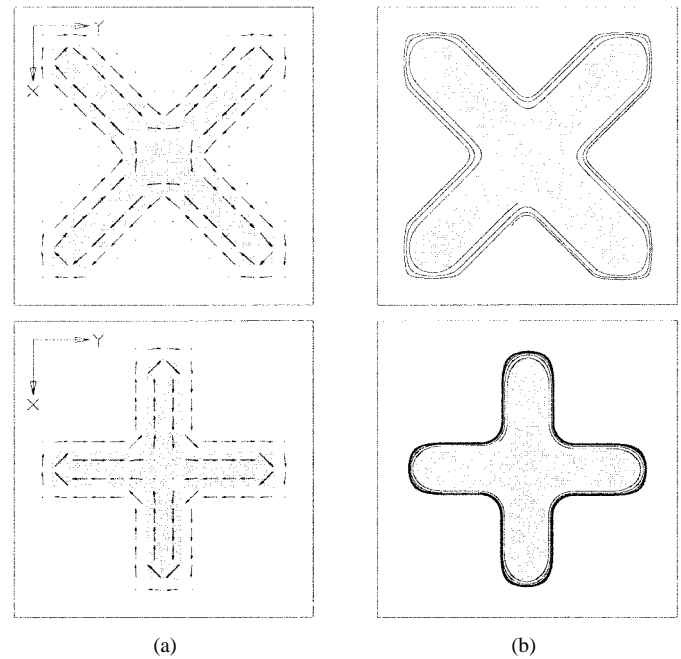


Fig. 7. (a) Edge vector fields with $\sigma = 0.5$. (b) trajectories of the particle in the edge vector fields with $\alpha = 0.1$ after 4000 time steps.

$t \rightarrow \infty$, an edge vector at the particle position approaches zero magnitude and the particle comes to rest. Fig. 7 presents other examples of particle trajectories in the edge vector fields. These results clearly show that the trajectory of a particle in the edge vector field does not coincide with object edges. Instead, it exhibits a spiral pattern surrounding the object which is undesirable for boundary extraction. This is due to the fact that the edge vector field which is tangential to the edges is not sufficient to describe the object boundary. It is required to introduce an external orthogonal force to control the motion of the particle as described in the next section.

C. Simulation of Particle Motion in an Orthogonal Vector Field

In the R^n space, it is well known that points on a curve can be uniquely and completely described by a linear combination of the two orthogonal vectors, directed tangential and normal to the curve as illustrated in Fig. 8. We can represent points

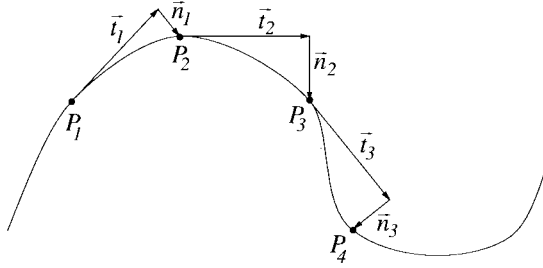


Fig. 8. Curve representation using linear combination of two orthogonal sets of vectors, directed tangential and normal to the curve.

on the curve by the a set of equations

$$\vec{P}_{k+1} = \vec{P}_k + \alpha_k \vec{t}_k + \beta_k \vec{n}_k, \quad \text{for } k = \dots, -2, -1, 0, 1, 2, \dots \quad (27)$$

where \vec{P}_k is the position of the k th point, \vec{t}_k and \vec{n}_k are the tangential and normal vectors of the curve at the k th point and α_k and β_k are constant numbers. Consequently, an additional set of normal vectors is required for obtaining a unique and complete description of the boundary edges. In this paper we use the normalized gradient of an edge image and the normalized Laplacian-gradient of the original image as candidate choices to be used as the orthogonal force field.

1) *Normalized Gradient of an Edge Image* The normalized gradient of an edge image $e(i, j)$ is defined by

$$\nabla|\vec{e}(i, j)| = \frac{1}{c} \left(\frac{\partial|\vec{e}(i, j)|}{\partial x} \vec{i} + \frac{\partial|\vec{e}(i, j)|}{\partial y} \vec{j} \right) \quad (28)$$

$$\approx \frac{1}{c} (-G_x * |\vec{e}(i, j)| \vec{i} - G_y * |\vec{e}(i, j)| \vec{j}) \quad (29)$$

where G_x and G_y are given by (22) and (23), $|\vec{e}(i, j)|$ is the magnitude of $\vec{e}(i, j)$, and c is the normalization factor. Fig. 9(a) and (b) show examples of the normalized gradient vector fields of edge images. Because the gradient of a scalar field points in the direction of steepest ascent of the field, for step edges, the gradient of an edge image points in the direction orthogonal to the maximum gradient paths and can therefore be used to force a particle to stay confined to the maximum gradient path. Using an edge vector field of the original image as the tangential force field and the normalized gradient vector field of the edge image as the additional orthogonal field, the recursive equation for the position of a particle in the combined force field at the k th time step is given by

$$\vec{P}_k = \vec{P}_{k-1} + \alpha \vec{e}_{k-1} + \beta \nabla|\vec{e}|_{k-1} \quad (30)$$

where α , $(-1 \leq \alpha \leq 1, \alpha \neq 0)$, and β , $(0 < \beta \leq 1)$, are the tangential and normal stepping factors that determine the rate of convergence.

Based on (30), using the orthogonal force fields \vec{e} and $\nabla|\vec{e}|$ derived from the images in the previous section, the resultant particle trajectories are given in Fig. 9(c) and (d). As seen, the particle trajectories obtained using (30) rapidly converge to unique closed paths very close to the actual object boundaries. It should be noted that both images in Fig. 9(c) and (d) are of size 16×16 pixels and hence the deviation observed in the trajectories from the true boundaries is in actuality less

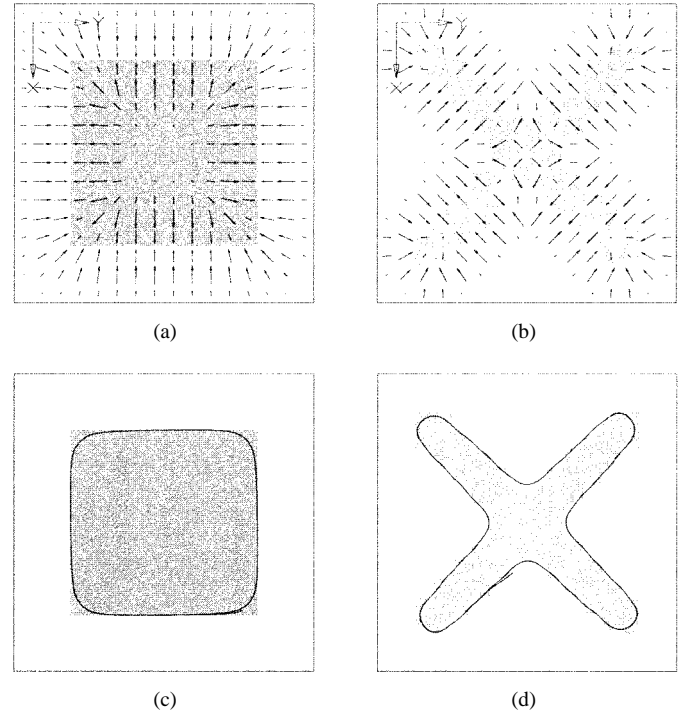


Fig. 9. Normalized gradient fields of the edge image with (a) $\sigma = 1$ and (b) $\sigma = 0.5$; (c) and (d) trajectories of the particle in the combined vector field based on (30) with $\alpha = 0.1$ and $\beta = 0.05$ after 4000 time steps.

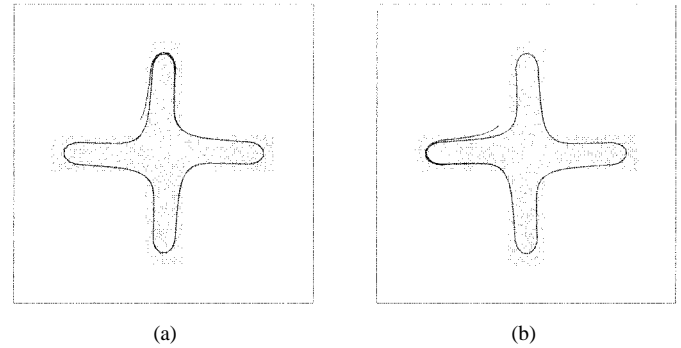


Fig. 10. Trajectories of the particle in the combined vector field based on (30) with $\sigma = 0.5$, $\beta = 0.05$ and (a) $\alpha = 0.1$ and (b) $\alpha = -0.1$ after 4000 time steps.

than one pixel. The results of using the combined vector field are significantly improved compared to those using only the edge vector field. One feature observed in this model is that, at the corners, the particle does not pass through tips of the corners but cuts through the inside. This is due to the fact that the maximum gradient path obtained using a finite difference operator always passes inside the corner of an object [18] resulting in a particle trajectory that tends to be smoother than the actual edges at the corner.

An interesting property of this model is the anti-symmetry property demonstrated in Fig. 10. The trajectory of the particle using a positive α shown in Fig. 10(a) is a mirror reflection of that obtained using a negative α shown in Fig. 10(b). Since a negative α is equivalent to an edge vector field in an opposite direction, the particle is forced to move backward causing the reverse trajectory.

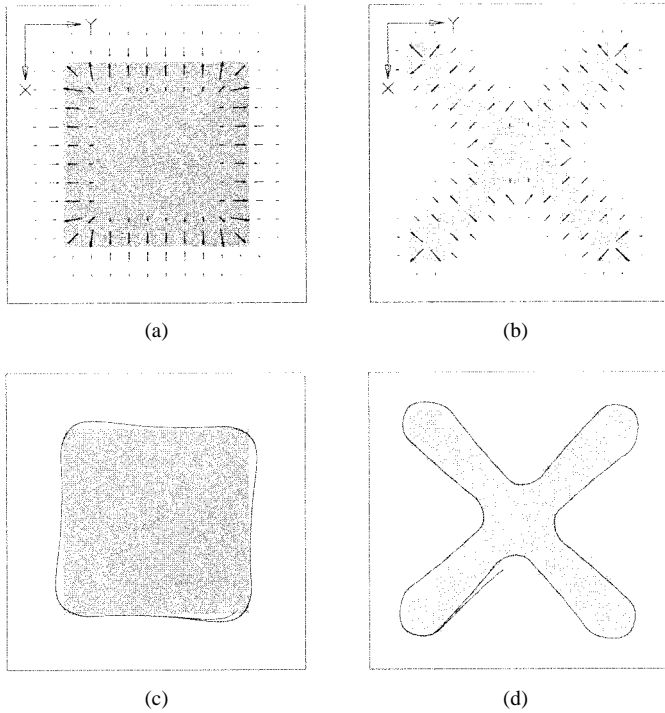


Fig. 11. Laplacian-gradient vector fields with (a) $\sigma = 1$ and (b) $\sigma = 0.5$; (c) and (d) trajectories of the particle based on (34) with $\alpha = 0.1$ and $\beta = 0.05$ after 4000 time steps.

2) Normalized Laplacian-Gradient of an Original Image

An alternate choice for a vector field orthogonal to edges is the normalized Laplacian-gradient vector field of an original image defined as

$$\vec{n}(i, j) = \frac{\nabla^2 I(i, j)}{k} \nabla I(i, j) \quad (31)$$

where

$$\nabla^2 I(i, j) = \frac{\partial^2 I(i, j)}{\partial x^2} + \frac{\partial^2 I(i, j)}{\partial y^2} \quad (32)$$

$$\approx G_x * (G_x * I(i, j)) + G_y * (G_y * I(i, j)) \quad (33)$$

and k is the normalization factor. The normalized Laplacian-gradient vector field satisfies two requirements: the field is orthogonal to the edge vector field and secondly, at each point, the normalized Laplacian-gradient vector points toward the nearest edge as demonstrated in Fig. 11(a) and (b). Similarly, using an edge vector field of an original image as a tangential force field and the normalized Laplacian-gradient vector field as an orthogonal force field, the recursive equation for the position of a particle at the k th time step is calculated as

$$\vec{P}_k = \vec{P}_{k-1} + \alpha \vec{e}_{k-1} + \beta \vec{n}_{k-1}. \quad (34)$$

The trajectories of the particle obtained using (34) are illustrated in Fig. 11(c) and (d). It is seen that the trajectory rapidly converges to unique closed paths conforming with true object edges. In this technique, it is seen that, at the corners, the particle drifts away from edges before returning back to edges causing a small distension of the trajectory. As demonstrated in Fig. 12, the distension of the particle trajectory at the corner is directly proportional to parameter β . Nevertheless, as in

the case of Fig. 9(c) and (d), due to the large pixel size, the deviation observed in the particle trajectory from the true boundary at the corners is less than one pixel and hence has little effect on the total object boundary. A comparison between the models described by (30) and (34) is illustrated in Fig. 13. The trajectory of a particle obtained using (34) shown in Fig. 13(b) is much closer to the actual boundary than that obtained using (30) shown in Fig. 13(a).

D. Application of Particle Motion for Boundary Extraction

We have so far presented two models of particle motion in the combined force field that result in closed trajectories. The models can be used for boundary extraction where the convergent path of the particle trajectory in the combined force field of the image directly represents an estimate of the object boundary. The accuracy and resolution of the extracted boundary depend mainly on parameters σ , α and β . Two orthogonal vector fields, including tangential and normal force fields, are required for implementing the algorithm. The edge vector field is used as the tangential force field while the orthogonal force field can be either the gradient vector field of an edge image or the Laplacian-gradient vector field of the original image. The streamline feature of the edge vectors and the fact that vectors in both the gradient vector field of the edge image and the Laplacian-gradient vector field of the original image point toward proximate edges guarantee that the extracted boundary will be connected and follow the true edges. The model using the Laplacian-gradient vector field performs better than the model using the gradient vector field of an edge image in that it preserves corners of the object. Implementation of the complete boundary extraction algorithm and experimental results are presented in the next section.

IV. EXPERIMENTAL RESULTS

This section presents a two stage procedure for estimating object boundaries. In the first stage, the edge vector field and the normalized Laplacian-gradient vector field are calculated. In the second stage, the combined vector field is used as a force field to drive a particle that describes a path close to the actual object boundary. The resulting edge-oriented image segmentation algorithm can be summarized in the following four steps.

- 1) Select the optimum scaling parameter σ of the GWIMV operator for detecting edges at a desired scale.
- 2) Generate the edge vector field and the normalized Laplacian-gradient vector field of the image.
- 3) Select the starting point for the particle trajectory.
- 4) Simulate particle motion and extract object boundary by following the particle trajectory.

The edge vector field and the Laplacian-gradient vector field are generated by convolving the image with the difference masks in (20), (21) and (31). After obtaining both vector fields, the initial point for the model is selected. The initial point of the particle trajectory is very crucial for the outcome of the process. If the initial point is far away from edges, the particle will accelerate slowly resulting in a longer time to complete the trajectory. Therefore it is necessary to ensure that the particle

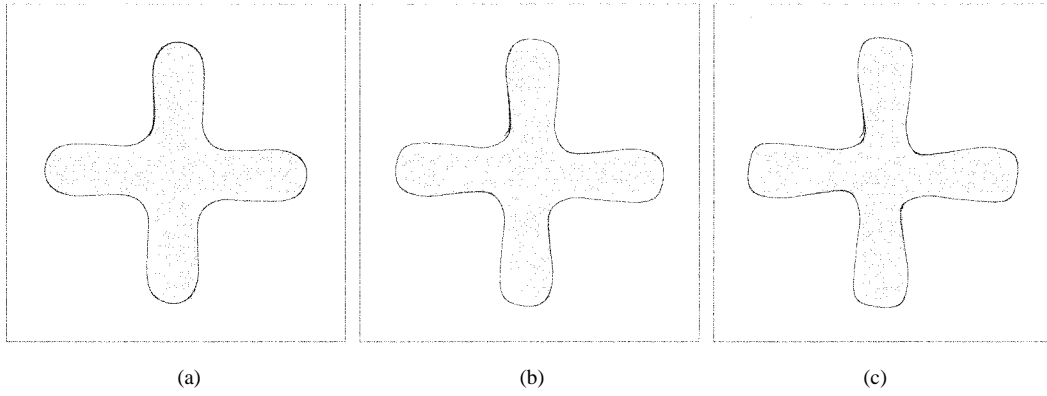


Fig. 12. Trajectories of the particle based on (34) with $\sigma = 0.5$, $\alpha = 0.1$ and (a) $\beta = 0.02$, (b) $\beta = 0.05$, and (c) $\beta = 0.1$ after 4000 time steps.

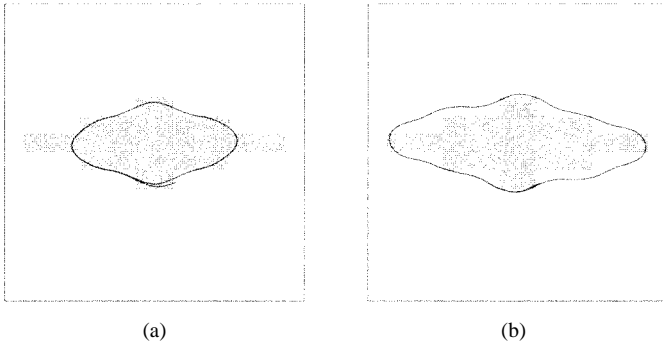


Fig. 13. Trajectories of the particle based on (a) (30) and (b) (34) with $\sigma = 1$, $\alpha = 0.1$ and $\beta = 0.05$ after 4000 time steps.

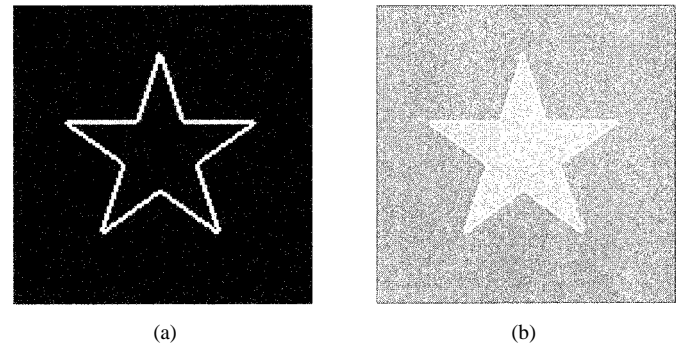


Fig. 14. (a) Edge image of the star image with $\sigma = 0.5$ and (b) extracted edge sequence with $\alpha = 0.1$, and $\beta = 0.07$.

starts close to the edges, for example, at a point of maximum gradient. The boundary is generated by computing the particle path according to (34) and the process is terminated when the trajectory completes a loop or reaches other previously extracted paths. Since, the particle may not reach a true edge in the first few iterations, the particle must be allowed to move a few steps before starting the boundary extraction process. Results of implementing the procedure on noise-free and noisy images are presented in the following section. Objects with sharply varying boundaries are synthetically generated to evaluate the performance of the algorithm. Results of boundary extraction in real images are also presented.

A. Boundary Extraction of a Single Object in Noise-Free Images

In the case of a noise-free image with a single object, the edge vector field clearly exhibits the current loop flowing around the object in one direction ensuring closed edge sequences. Two examples of simulated images are presented in this section. In the first example, the edge image with $\sigma = 0.1$ and the extracted edge sequence based on (34) with $\alpha = 0.1$ and $\beta = 0.07$ are shown in Fig. 14(a) and (b), respectively. The results show that the algorithm performs with subpixel accuracy. The second example demonstrates the object composition at several scales, captured by the algorithm. The snowflake is a multiple scale object consisting of three different scales. Fig. 15 shows edge images and extracted edge sequences at three values of σ . The parameters α and β used in

this study were 0.1 and 0.03 respectively. At the smallest scale $\sigma = 0.1$, the algorithm captures all details of the snowflake as seen in Fig. 15(a). Choosing increasing values of σ , the resultant edge images and object boundaries capture larger scale objects as seen in Figs. 15(b) and (c). These results demonstrate the multiscale edge detection capability of the algorithm.

B. Boundary Extraction in Noisy Images

The algorithm was then implemented on simulated noisy images. The images used in the previous experiment were corrupted by additive Gaussian noise. Fig. 16(a) shows the degraded image of a star with SNR = 2. To get good noise suppression, σ was chosen to be 3.41. Other parameters were $\alpha = 0.1$ and $\beta = 0.04$. The extracted edge sequence is shown in Fig. 16(b). In contrast, the result obtained using a conventional thresholding process with threshold value = 140 is shown in Fig. 16(c). Similar results for the degraded snowflake image with SNR = 2 are shown in Fig. 17. The boundary obtained with $\sigma = 2.12$ and the edge image after thresholding with the threshold value = 100 are presented in Fig. 17(b) and (c). As seen from Figs. 16(b) and 17(b), the extracted edge sequences obtained using the proposed algorithm are closed, continuous and smooth. A small distortion in the boundary can be seen at corners of the objects. In contrast, in edge images obtained using thresholding, displayed in Figs. 16(c) and 17(c), boundaries are jagged, several pixels wide and discontinuous at object corners.

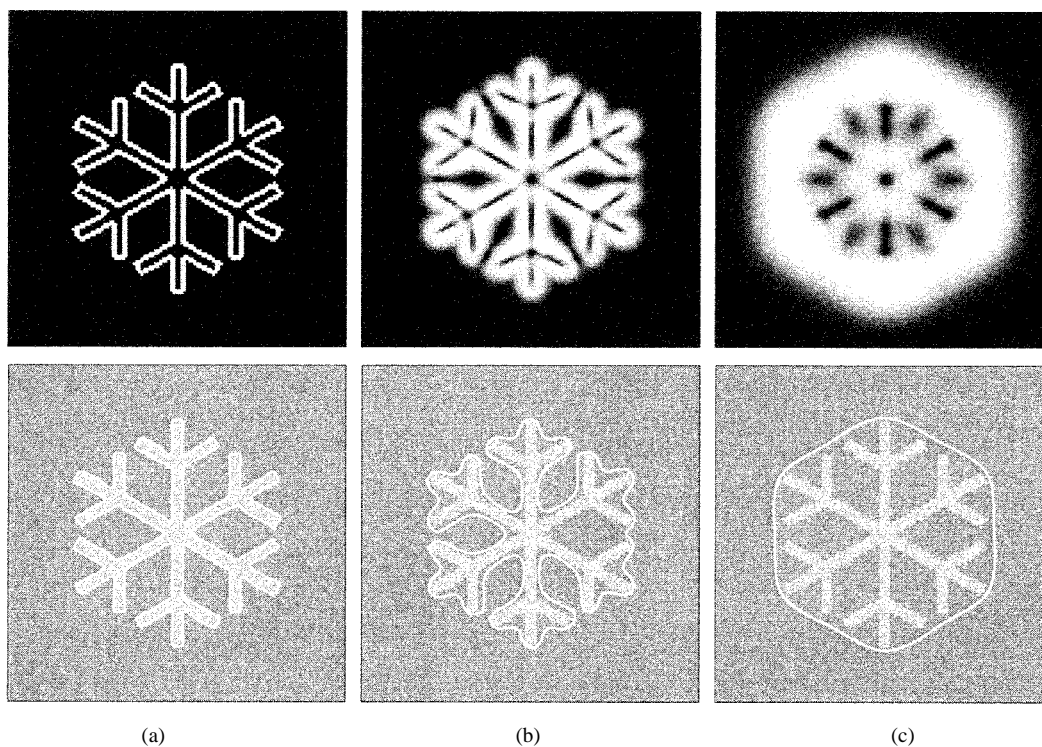


Fig. 15. Edge images and extracted boundaries with $\alpha = 0.1$, $\beta = 0.03$ and (a) $\sigma = 0.1$, (b) $\sigma = 4$, and (c) $\sigma = 11.46$.

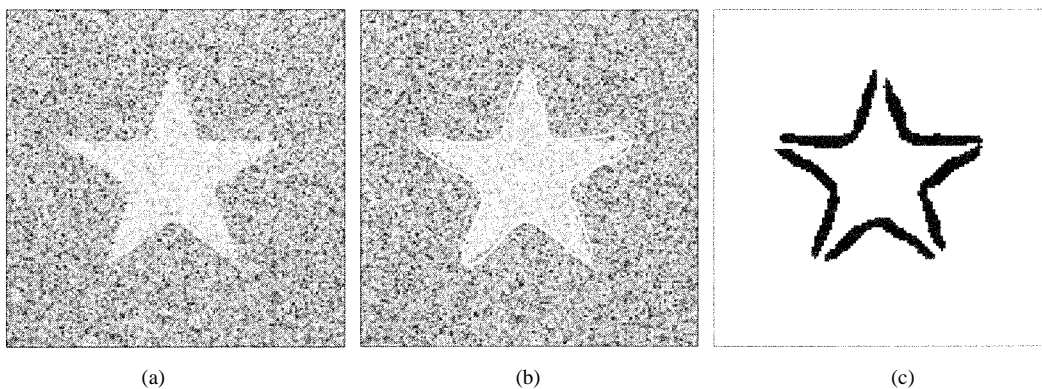


Fig. 16. (a) Degraded image with $\text{SNR} = 2$. (b) Extracted edge sequences with $\sigma = 3.41$, $\alpha = 0.1$ and $\beta = 0.04$. (c) Edge image after thresholding with threshold value = 140.

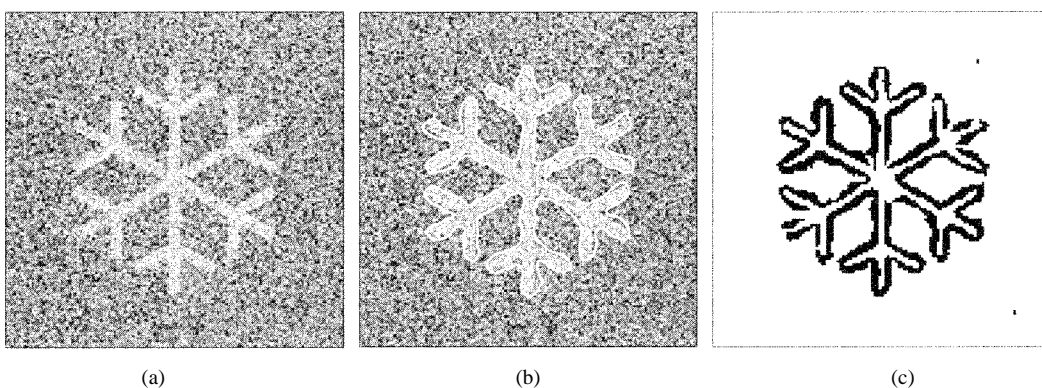


Fig. 17. (a) Degraded image with $\text{SNR} = 2$. (b) Extracted edge sequences with $\sigma = 2.12$, $\alpha = 0.1$ and $\beta = 0.04$. (c) Edge image after thresholding with threshold value = 100.

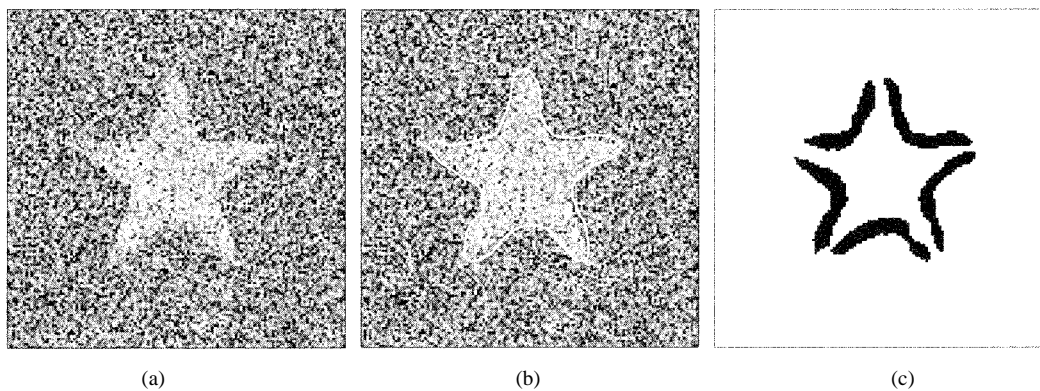


Fig. 18. (a) Degraded image with $\text{SNR} = 1$. (b) Extracted edge sequences with $\sigma = 4.44$, $\alpha = 0.1$ and $\beta = 0.04$. (c) Edge image after thresholding with threshold value = 140.

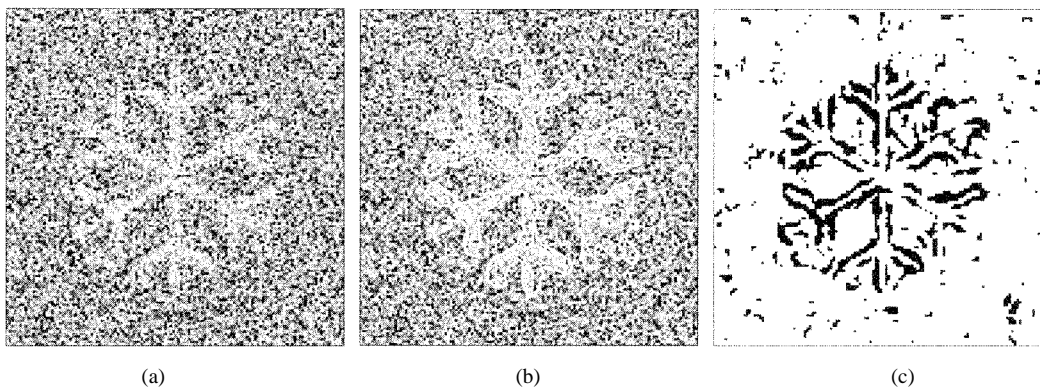


Fig. 19. (a) Degraded image with $\text{SNR} = 1$. (b) Extracted edge sequences with $\sigma = 2.12$, $\alpha = 0.1$ and $\beta = 0.04$. (c) Edge image after thresholding with threshold value = 100.

Figs. 18 and 19 show noisy images with $\text{SNR} = 1$ and corresponding results of the proposed algorithm along with those obtained using thresholding. As seen above, the results obtained using thresholding are very noisy, discontinuous at object corners and also several pixels wide. The discontinuity of edges at object corners occurs because the strengths of edges at object corners are relatively small. This makes the thresholding method ineffective for detecting boundaries of high curvature objects. Noise in the image also makes results of thresholding uneven and unstable. On the other hand, object boundaries based on the trajectory of a particle in the edge vector field are continuous, closed and complete with subpixel accuracy and no further processing is required. The algorithm can also be easily extended to images with multiple objects.

C. Boundary Extraction of Multiple Objects

When there are several objects of different mean intensities in the image, the boundary edge sequence is not a single loop as in the case of a single object. An intermediate step is needed for scanning over the entire image for finding the local maxima of varying edge strengths in the edge image. A set of local maxima is used as a set of initial points for the boundary extraction process. The initial points are sorted in descending order and the point with the highest edge strength is selected first. For each initial point, the boundary extraction process is performed until the termination criterion is satisfied, i.e., the

particle completes the loop or reaches a previously extracted edge. The process then automatically restarts at the next initial point and repeats the boundary extraction algorithm until all initial points are used.

An example of a multiple object image is shown in Fig. 20(a). The corresponding edge image with $\sigma = 0.5$ and resultant boundary with $\alpha = 0.1$ and $\beta = 0.05$ are shown in Fig. 20(b) and (c). Due to the inhomogeneous nature of the edge vector field of a multiple object image, the boundary extraction algorithm is required to track the particle trajectory along two opposite directions. The result obtained using the two-way boundary extraction algorithm is shown in Fig. 20(d). In practice, the object topology is of major concern in model-based edge detection algorithms. However, in the proposed algorithm, the particle sequentially searches for all possible edges. Hence, the object topology is irrelevant.

D. Experimental Results of Natural Images

Finally, the results of applying the boundary extraction algorithm to natural images are described. In natural images, object edges are not ideal step edges which is the fundamental assumption of all first order difference edge operators. Moreover, regions, scales and edges in natural images are fuzzy and corrupted by noise. A low contrast image of an airplane is shown in Fig. 21(a). The edge image with $\sigma = 0.1$, extracted edge sequence with $\alpha = 0.2$ and $\beta = 0.3$ and edge image after

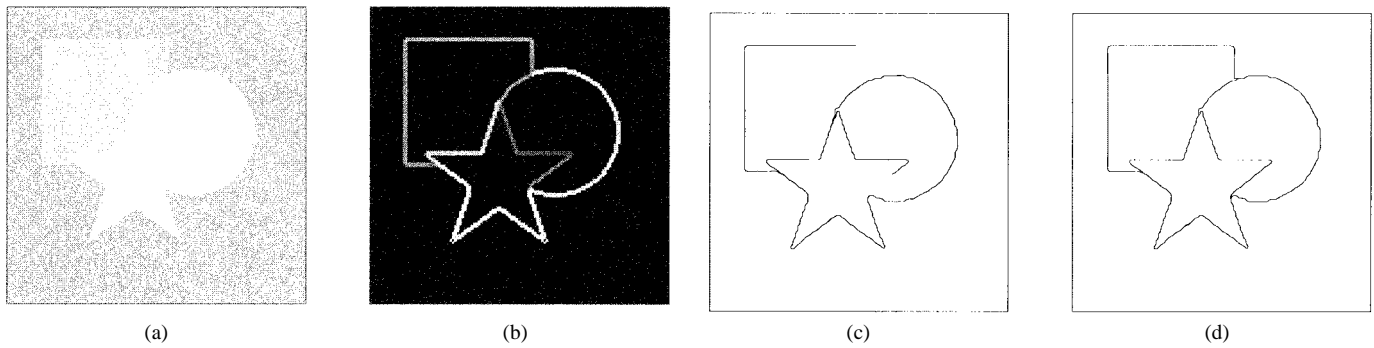


Fig. 20. (a) Original image of multiple objects. (b) Edge image with $\sigma = 0.5$. (c) One-way extracted edge sequences with $\alpha = 0.1$ and $\beta = 0.05$. (d) Two-way extracted edge sequences with $\alpha = 0.1$ and $\beta = 0.05$.

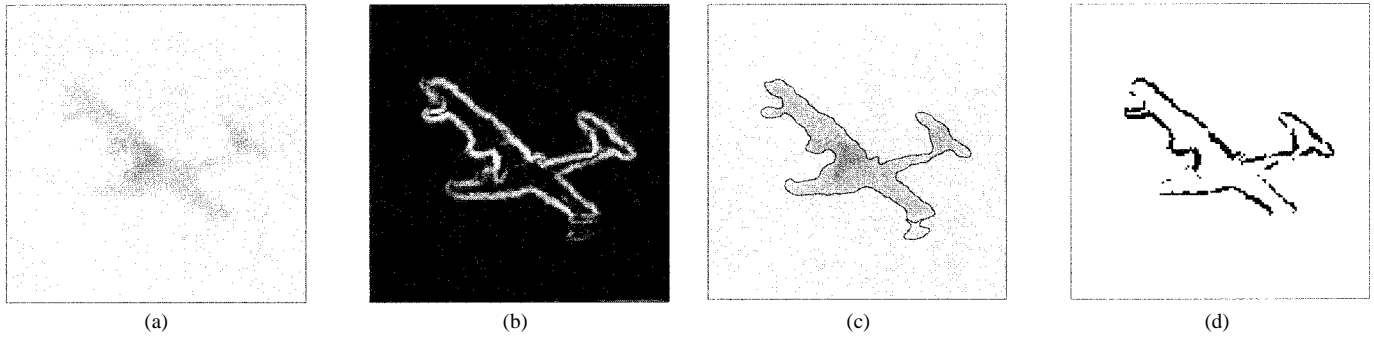


Fig. 21. (a) Image of an airplane. (b) Edge image with $\sigma = 0.1$. (c) Extracted boundary with $\alpha = 0.2$ and $\beta = 0.3$. (d) Edge image after thresholding with the threshold value = 100.

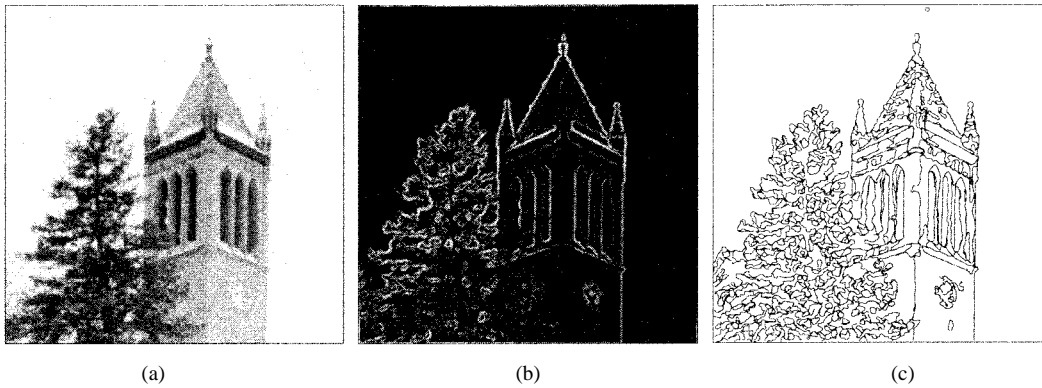


Fig. 22. (a) Original image. (b) Edge image with $\sigma = 0.1$. (c) Extracted boundaries.

thresholding are displayed in Fig. 21(b) to (d) respectively. A second natural image shown in Fig. 22(a) consists of complex structures with fine details of pine trees and varying object intensities. The edge image with $\sigma = 0.1$ and extracted boundaries are shown in Fig. 22(b) and (c) respectively. Most of details in the images are captured by the algorithm. These results clearly show the robustness and power of the algorithm to handle nonideal edges in real images.

V. CONCLUSION

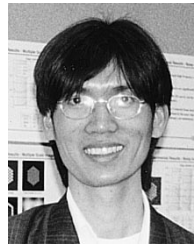
A novel approach in image processing based on a vector image model and the use of this model to derive a new edge operator are presented. A vector operation in the image yields an edge vector field analogous to the Hamiltonian gradient

field. By varying a scaling parameter σ , the operator is capable of handling multiscale edge detection problems. Second, a new boundary extraction algorithm based on particle motion in a force field is proposed. The edge vector field is used as a tangential driving force field while either the gradient vector field of an edge image or the Laplacian-gradient vector field are used as an additional orthogonal compressing force field. The models utilize both first and second order derivatives of an image to get the complete boundary. There are several major advantages of this algorithm over the conventional edge-oriented segmentation techniques. Since only initial points are required, initialization of the model is much simpler. By searching for all possible edges, the algorithm can handle the object of unknown topology. The boundary extraction based on a simple particle motion equation is computationally simple.

Using a fine stepping factor, the result can achieve subpixel accuracy. The sequential nature of the algorithm guarantees continuities of extracted boundaries. The algorithm is also capable of handling multiscale edges by varying the scaling parameter of the edge operator. Additional work is underway to evaluate the performance of the algorithm on nonideal edges of different types.

REFERENCES

- [1] J. M. S. Prewitt, "Object enhancement and extraction," *Picture Processing and Psychopictorics*, B. S. Lipkin and A. Rosenfield, Eds. New York: Academic, 1970, pp. 75–149.
- [2] G. S. Robinson, "Edge detection by compass gradient masks," *Comput. Graph. Image Process.*, vol. 6, pp. 492–501, 1977.
- [3] D. C. Marr, *Vision*. San Francisco, CA: W. H. Freeman, 1982, pp. 54–74.
- [4] A. Rosenfield and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Trans. Comput.*, vol. C-20, pp. 562–569, 1971.
- [5] E. Argyle, "Techniques for edge detection," in *Proc. IEEE*, pp. 285–287, 1970.
- [6] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 679–698, 1986.
- [7] R. Machuca and A. L. Gilbert, "Finding edges in noisy scenes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, pp. 103–111, 1981.
- [8] E. P. Lyvers, O. R. Mitchell, M. L. Akey, and A. P. Reeves, "Subpixel measurements using a moment-based edge operator," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 1293–1309, 1989.
- [9] A. J. Tabatabai and O. R. Mitchell, "Edge localization to subpixel values in digital imagery," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 188–201, 1984.
- [10] N. Eua-Anant, "A novel boundary extraction algorithm based on a vector image model," M.S. thesis, Iowa State Univ., Ames, IA, 1996.
- [11] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989, pp. 355–356.
- [12] D. B. Cooper *et al.*, "Stochastic boundary estimation and object recognition," *Comput. Graph. Image Process.*, vol. 12, pp. 326–356, 1980.
- [13] H. Elliott and L. Srinivasan, "An application of dynamic programming to sequential boundary estimation," *Comput. Graph. Image Process.*, vol. 17, pp. 291–314, 1981.
- [14] A. Goshtasby and H. Shyu, "Edge detection by curve fitting," *Image Vis. Comput.*, vol. 13, pp. 169–177, Apr. 1995.
- [15] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, pp. 312–331, 1988.
- [16] L. D. Cohen, "Note: On active contour models and balloons," *Comput. Vis., Graph., Image Process.: Image Understand.*, vol. 53, pp. 211–218, Mar. 1991.
- [17] R. Ronfard, "Region-based strategies for active contour models," *Int. J. Comput. Vis.*, vol. 13, pp. 229–251, 1994.
- [18] M. M. Fleck, "Some defect in finite-difference edge finders," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 337–345, 1992.
- [19] S. Dayanand, W. R. Uttal, T. Shepherd, and C. Lunsis Jr., "A particle system model for combining edge information from multiple segmentation modules," *Comput. Vis., Graph., Image Process.: Image Understand.: Graph. Models Image Process.*, vol. 56, pp. 219–230, May 1994.



and neural networks.



of NDE signals.

Nawapak Eua-Anant (S'98) received his B.E. degree in electrical engineering from Khon Kaen University, Thailand, in 1991, and the M.S. degree in electrical engineering from Iowa State University, Ames, in 1996. He is currently pursuing the Ph.D. degree in electrical and computer engineering at Iowa State University.

From 1991 to 1993, he was an instructor at Department of Electrical Engineering, Khon Kaen University. His research interests are image processing, boundary extraction, multiscale image analysis,

Lalita Udpa (SM'92) received the Ph.D. degree in electrical engineering from Colorado State University, Fort Collins, in 1986.

She is currently an Associate Professor of electrical engineering at Iowa State University, Ames. She works primarily in the area of nondestructive evaluation (NDE). Her research interests include development of solutions to forward and inverse problems in NDE. She also works extensively on the application of signal and image processing and pattern recognition algorithms for characterization

Dr. Udpa is a member of ASNT, Sigma Xi and Eta Kappa Nu.