Title
A Unified Framework for Shape Segmentation, Representation, and Recognition

Authors
Malladi, R.
Sethian, J.A.

Publication Date
1994-08-01

LBL-36069
UC-405
Preprint

Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA

Physics Division

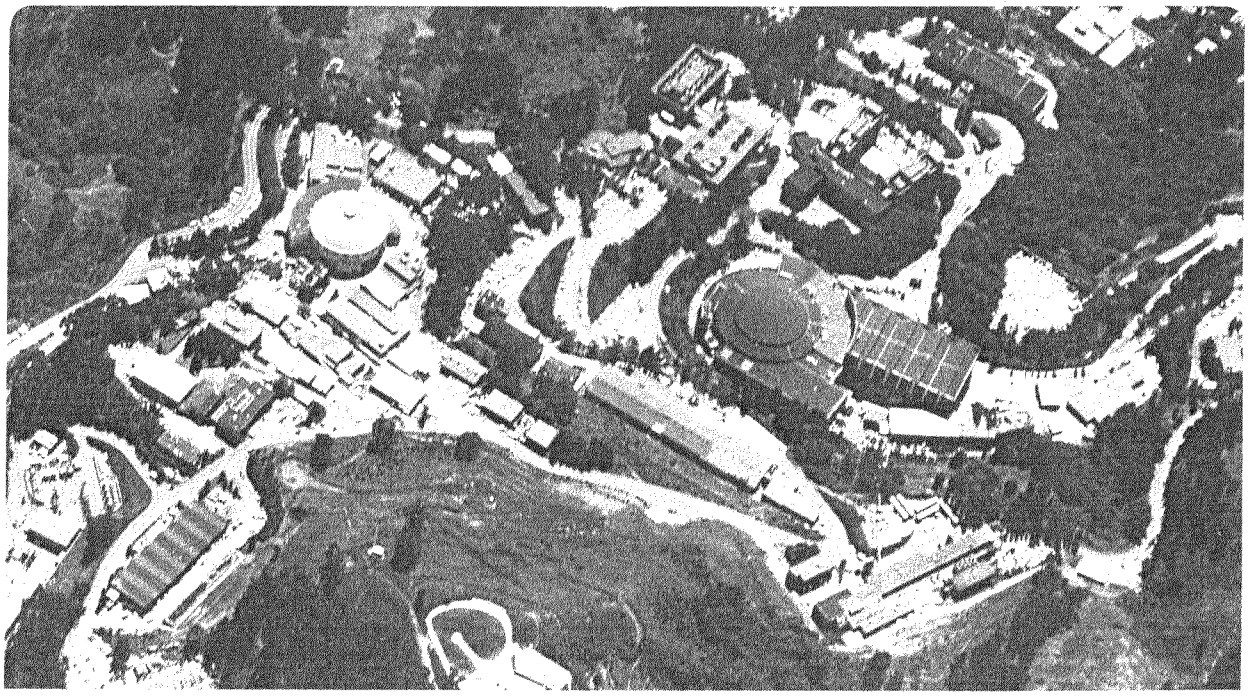
Mathematics Department

To be submitted for publication

A Unified Framework for Shape Segmentation, Representation, and Recognition

R. Malladi and J.A. Sethian

August 1994



Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098

LBL-36069
UC-405

A UNIFIED FRAMEWORK FOR SHAPE SEGMENTATION, REPRESENTATION, AND RECOGNITION[1]

R. Malladi[2] and J.A. Sethian

Lawrence Berkeley Laboratory
and
Department of Mathematics
University of California
Berkeley, CA 94720

August 1994

[1] This work was supported in part by the Applied Mathematical Sciences Subprogram of the Office of Energy Research, U.S. Department of Energy under Contract DE-AC03-76SF00098 and by the NSF ARPA under Grant DMS-8919074.

[2] NSF Postdoctoral Fellow in Computational Science and Engineering.

A Unified Framework for Shape Segmentation, Representation, and Recognition *

R. Malladi† and J. A. Sethian

Lawrence Berkeley Laboratory
and
Department of Mathematics
University of California, Berkeley, CA 94720

Abstract

In this paper, we present a unified approach to the problem of shape representation and recognition. The key idea is the level set representation of a shape, in which information about the boundary of shape is embedded in a level set function. This perspective on an object and shape boundary has several advantages. First, the level set function provides information which allows accurate comparison of a trial shape with a given collection of possible forms. Second, the level set function provides a convenient framework for perturbing and distorting the shape in order to locate a possible form near the trial shape. As an example, we use our level set approach to study the shapes of handwritten numerals, establishing the effectiveness of the level set function as a feature vector, and the value of flow perturbation to aid their recognition. We present results of applying our method to numeral shapes extracted from NIST databases 3 and 7 of segmented characters.

1 Introduction

Shape recognition requires several steps. The goal is to extract and identify a shape from a given image field. This may involve image enhancement to bring the shape of interest into greater relief, segmentation to isolate the boundary of the shape from the enhanced image, a representation scheme to bring the boundary into a convenient form for evaluating shape properties, and a recognition technique to compare the shape descriptor with other known shapes.

The methodology we are developing in this and related works is a unified approach to the above four steps. The key aspect is the level set perspective, in which a front (that is, shape) is represented

*Supported in part by the Applied Mathematical Sciences Subprogram of the Office of Energy Research, U.S. Dept. of Energy under Contract DE-AC03-76SD00098 and by the NSF ARPA under grant DMS-8919074.

†NSF Postdoctoral Fellow in Computational Science and Engineering.

as the zero level set of a higher dimensional function, see [32, 38]. This level set framework is convenient for advancing the front/shape boundary, and has proved to be of great use in a variety of interface motion calculations, see [38, 39, 7, 37, 35].

The first step, image enhancement, has met with great success in recent years, see ([2, 31]), using various techniques which include some of the level set flow ideas mentioned above. The second step, segmentation, has been tackled using a level set framework by Malladi, Sethian and Vemuri in [22, 23]. In that work, a general framework was developed to recover the level set implicit signed-distance function from image data, where the zero level set of the obtained level function represents the boundary of the shape.

In this work, we finish the sequence by constructing a level set approach to shape representation and shape recognition. A given shape is embedded in a family of level sets and thus contains considerable information expressed via neighboring level sets about the shape boundary. In addition, such quantities as curvature, boundary normals, and holes are easily captured in this perspective. Thus, qualities of a shape which may prove important in shape recognition are easily obtained. Finally, the level set framework allows one to perturb the shape of a boundary to discover the critical features inherent in its shape.

To demonstrate and quantify these ideas, we use our methodology to identify handwritten numerals. We begin by training a neural net classification system on the NIST segmented, handwritten numerical databases 3 and 7. These databases contain almost 100,000 handwritten characters obtained from a large number of subjects. Unlike other recognition schemes, which often rely on extracting shape features such as stroke elements, holes, and curvature extrema, we train on a two-dimensional grid of values which describes a discretized version of the level set function. The results of training on this feature vector gives the first stage of our recognition process. Following standard terminology, given a set of characters, we discard a certain number, and then choose to classify those that remain. On NIST database 3, we obtain classification rates of well over 99.00% using just the level set function as the feature vector.

The second part of this process demonstrates the additional power of the level set function. Given the set of discards, we then flow each under a variety of rules, corresponding to the curvature flow, expansion [36], anisotropy [39], and a new flow which extends protruded arms of objects. Such flows

are straightforward using the Osher-Sethian level set algorithm (see [32]). We study the effects of these perturbations under the neural classification system, and select shapes for identification as a particular numeral if continued perturbation carries it into a region of high confidence in the neural net classifier. In effect, we attempt to perturb characters by following gradients in neural net space, until they deform into shapes that move within the bounds of the acceptance criteria. By doing so, we can identify a large fraction of the discards with minimal decrease in the accuracy of our classifications.

2 Background and Related Work

Segmentation of shapes from image data and their subsequent recognition has been a focus of computer vision and pattern classification research efforts over the years. Typical application areas include automatic target recognition, industrial part recognition, and optical character recognition. In this section, we give some background work on edge detection, segmentation, shape description, and recognition.

To begin, edge finding [26, 25] and edge grouping [28] have been used for the purpose of determining boundaries of shapes in images. One aspect of such techniques is that edge detection is a local operation and thus edges do not necessarily correspond to boundaries of objects. In the context under discussion here, we are interested in mathematical descriptions of shapes appearing in images. In this setting, boundary finding as a means of segmenting shapes from images has been employed in many different arenas. For example, Widrow [42] used parameterized templates called rubber masks to model objects. Kass, Witkin and Terzopoulos [16] have investigated the use of energy-minimizing snakes to extract smooth shapes from images – different behavior is achieved by adjusting the weights of the smoothness and image force terms. Prior knowledge of natural objects found in biomedical images has been used to bias the shape boundary in the work on parametrically deformable models by Staib and Duncan [40].

In order to perform recognition, it is important to represent a given shape in a distinctive fashion that makes is easy to place dissimilar shapes in different classes. Shape representation schemes can be broadly classified into boundary-based approaches vs. region-based approaches. In the first category, chain codes, introduced by Freeman [11], encode a shape by traversing the shape

boundary either assuming a 4 or an 8 connectivity. Polygonal approximations [33] describe a complex boundary by a sequence of straight lines. The accuracy of the approximation can be controlled by varying the number of lines. Shape features such as one-dimensional moments [12] and Fourier descriptors [34] are computed from the shape boundary. Not only can one encode and reconstruct a given shape boundary using Fourier features, but one can also render these features invariant to similarity [9] and affine transformations [3]. Such invariant features facilitate robust recognition.

Among the region-based shape representation schemes, one of the most typical is a "skeleton" [5] or the medial-axis transform [20] approach. A definition of skeleton based on the propagating fire front analogy is provided by Blum [5]; the skeleton is the locus of points at which a fire front started along the shape boundary and burning the enclosed region at unit speed meets. There are a large number of skeletonization algorithms (see [21, 27] and references therein). Although skeletons provide a compact shape description, they may contain a large number of redundant edges which may minimally contribute to shape information. Efficient edge pruning schemes are discussed in [30, 27, 18].

The above two classes of methods have met with varying degrees of success. We believe that one of the obstacles to greater success has been the separation of the segmentation step from the recognition step. In this paper, we design an approach in which the problems of shape segmentation/recovery and recognition are solved within the same framework.

For the segmentation step, Malladi, Sethian and Vemuri [22] have adopted the level set front propagation schemes developed by Osher and Sethian [32], and Sethian [38], to the problem of shape recovery in computer vision; a related approach has also been taken by Caselles *et al.* [6]. In these techniques, unknown shapes are represented by propagating fronts moving with curvature-dependent speed. For details and advantages of this approach over the existing shape finding methods, the reader is referred to [23].

The result of segmentation using this approach is the representation of a given shape by its signed-distance function defined on a square grid. Distances are positive at points lying outside the shape boundary and are negative inside. There are several advantages to this representation. First, boundary- and region-based representations are implicitly contained in the signed-distance function.

The boundary is simply its zero level set. Secondly, embedding the shape as a zero level set of a higher dimensional function enables us to adopt a global view instead of a local one. In this global view, shapes with disconnected boundaries can be represented by exploiting the property that the zero level set of a higher dimensional function need not be simply connected [38]. Also, shapes with and without holes can be modeled in a seamless fashion [22]. Lastly, by adopting the global approach, we can "flow" the shapes using the level set front propagation algorithms introduced by Osher and Sethian [32]. We wish to explore the notion of flowing (moving) a shape with different flow rules as a means of accentuating its shape features, thereby enhancing the chances of its correct classification.

To quantify these ideas, we focus on shape recognition schemes on the problem of handwritten numeral recognition. This area has been researched extensively and a gamut of shape features have been used for recognition purposes. A thorough review on historical development of OCR (optical character recognition) technology may be found in Mori *et al.* [29]. Such schemes include statistical classification methods with global feature analysis schemes like moments and mathematical transforms (Fourier, Walsh, Wavelet [19]), and syntactical methods used with structural features like loops, junctions, strokes, convexities, e.t.c. [41].

To execute shape recognition using our level set ideas, we begin by computing the signed-distance functions corresponding to numeral shapes and express them on a square grid. We then train a back propagation neural network [15] on the above feature vector. To obtain high reliability rates, we employ two or more neural networks trained on different global feature vectors, and classify a numeral instance only if they agree. Such stringent acceptance criteria generates a large set of rejected samples. We then show that these rejected samples can be brought into acceptance by modifying their shape via curvature-dependent level set flow [38]. We shall describe a set of flow rules that accentuate different shape features of some shapes. We present results of testing our scheme on the NIST databases 3 and 7 of segmented characters.

As presented, our shape representation scheme falls under the category of global analysis. As mentioned earlier, Caselles *et al.* [6] have also used the level set approach in boundary segmentation. Using much of the mathematical machinery introduced in [36] and level sets [32, 38], Kimia, Tannenbaum and Zucker [17] have tried to produce a theory of shape by suggesting a "reaction-

diffusion space of shapes"; which consists of computing deformed shapes under the general speed law examined extensively by Sethian in [36] via the Osher-Sethian algorithm. In this paper, we extend the general speed law to include new motions, and focus on a study of the complete process of segmentation, description, and recognition. We demonstrate this approach as applied to character recognition.

The outline of this paper is as follows. In Section 3, we review the application of level sets to shape segmentation. In Section 4, we discuss the value of the signed-distance function as a shape descriptor. In Section 5, we present details of our multi-neural network approach to handwritten numeral recognition and in Section 6, for certain test examples, we demonstrate how various rules for motion can be used to emphasize given aspects of the shape of an object. Finally, we apply our level set shape descriptor and flow rules to optical character recognition.

3 Shape Recovery in 2D

In the shape recovery scheme described in Malladi, Sethian and Vemuri [23], the goal is to extract mathematical descriptions of objects found in images without making too many restrictive assumptions about their shapes and topology. The boundary of the object is modeled as a front propagating along its gradient field with speed that is a function of its curvature. This front is expressed as the zero level set of a higher dimensional function defined in the image domain. In what has been called an Eulerian formulation [38], the equations of motion are written for the higher dimensional function in such as way that its zero level set will always yield the current position of the propagating front. Unknown shapes are recovered by making the front adhere to object boundaries. This is done by applying an artificial speed term on the front which acts like a halting criterion.

We begin by letting ψ be the level set function, K be the curvature, and $\hat{k}_I$ be the *extended* image-based speed term. The equation of motion is

$$\psi_t + \hat{k}_I(1 - \varepsilon K) \mid \nabla\psi \mid = 0, \tag{1}$$

with the value of $\psi(\mathbf{x}, t = 0)$ set to $\pm\mathbf{d}$, where $\mathbf{d}$ is the shortest distance from $\mathbf{x}$ to the initial front. Note that the word *extension* is meant in the sense of [39, 23]. The image-based speed term k_I,

which is defined as

$$k_I(x,y) = \frac{1}{1+\mid \nabla G_\sigma * I(x,y) \mid} \tag{2}$$

has values that are closer to zero in regions of high image gradient and values that are closer to unity in regions of relatively constant image intensity. The expression $G_\sigma * I$ denotes the image convolved with a Gaussian smoothing filter with characteristic width is σ. We solve the governing equation using entropy-satisfying upwind schemes developed in Osher and Sethian [32]. In [23], a variety of ways of computing the evolving front, including fast narrow band methods [1], reinitializations, extension of the method to handle 3D data sets, and different artificial stopping criteria are given. Given an image, our method requires the user to provide an initial contour, or a set of contours if there are more than one shapes of interest. The signed-distance function $\psi(\mathbf{x}, t = 0)$ is computed from the initial contour(s). Following this, fast narrow-band and *extension*-free schemes are used to move the front. The user may interact with the model by adjusting the smoothness control parameter ε until a desired degree of smoothness is achieved.

We illustrate this approach in the following examples. First, we recover the stomach shape from a 256×256 CT (Computed Tomography) image of an abdominal section shown in figure 1(a). The user places the initial contour inside the shape as shown in figure 1(a). The initial front is made to propagate in the normal direction with speed $F = \hat{k}_I(-1.0 - 0.025K)$. We employed the narrow-band update scheme to move the front. We show an intermediate stage in the front evolution in figure 1(b) and the final result in figure 1(c). Note that this method does not require the initial contour be placed close to the final shape. In addition, due to the εK component in the speed function, the front attains relatively smooth shapes like the thin-plate-membrane splines [16]. Calculations were performed on a 128×128 grid and the time step Δt was set to 0.0005.

Next, consider the image in figure 1(d). This image contains four character shapes two of which being shapes with holes. Our goal is to recover these shapes by performing just one calculation. For a complete description of shapes like "A" and "B", we need to recover their outer and inner boundaries. We initialize the front in such a way that it encloses all the character shapes (figure 1(d)). It is then made to propagate inward with a constant speed. As shown in figure 1(e), the front wraps itself tightly around the characters and subsequently splits into four separate parts (see

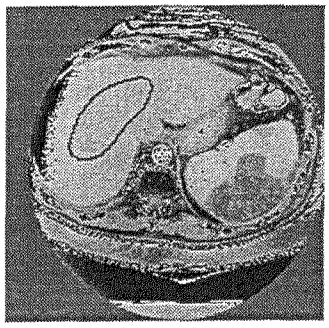
figure 1(f)). The first stage of this computation comes to an end when in figure 1(g), the level set $\{\psi = 0\}$ recovers the outer boundaries of four characters. In the second stage of our calculation, we momentarily relax the image-based speed component and flow the front into the character shapes (figure 1(h)), thereby recovering the holes present in the shapes of "A" and "B" (see figure 1(i)). The calculations for this experiment were done on a 128×128 grid and the time step Δt was set to 0.00025.

4 Shape Description with Signed-Distance

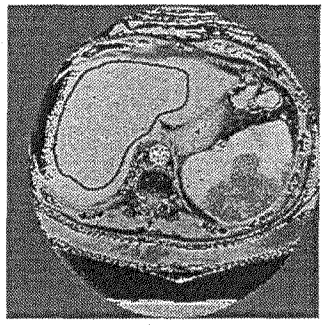
For a given shape, the above technique recovers the signed-distance function in the entire domain; the shape boundary is simply the zero level set of this function. In this section, we analyze the effectiveness of the signed-distance function for shape description.

Given a 2D shape, one can represent it either by the boundary that separates it from its background or by the region enclosed by the boundary. In other words, a shape can be described either by a collection of points lying on its boundary or a collection of interior points. As noted by Kimia *et al.* [17] in their study of "multidimensional" nature of shape, although these two representations are equivalent, they highlight different aspects of the shape. The signed-distance function of a shape implicitly contains both these representations. This is very desirable because during recognition, what is missed by one may be captured by the other.

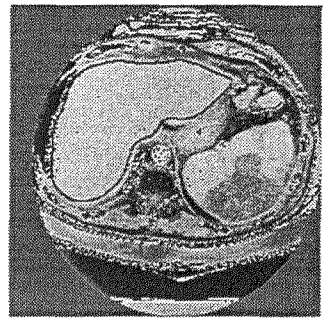
As an example, consider the shape of numeral two in figure 2(a). This shape can be represented by its boundary or by its skeleton as shown in figure 2(b). Instead, we embed the shape boundary in the signed-distance function ψ defined in a square region enclosing it. The value of $\psi(x, y)$ at a point (x, y) is defined to be the shortest distance from it to the shape boundary; distances are positive at points outside the shape boundary and negative inside it. From this definition it is clear that both the shape boundary and its skeleton are implicitly contained in the ψ function. To make the boundary information explicit, one need only collect the zero level set of the function ψ. If the skeleton is desired, one could use Blum's prairie fire analogy and propagate the boundary inward at constant speed. A convenient way to simulate this is to employ Osher-Sethian [32] schemes for front propagation. The reader is referred to Kimmel *et al.* [18] for a skeletonization algorithm based on level set ideas. In figure 2(c), we plot the level sets (equal distance contours) of the function ψ



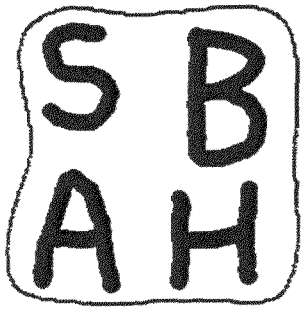
(a) $t = 0.0000$



(b) $t = 0.1500$

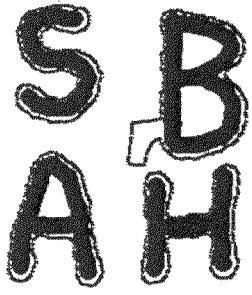


(c) $t = 0.2875$



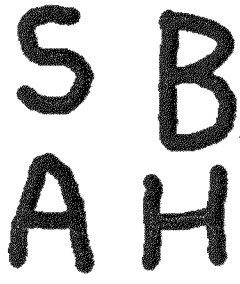
(d) $t = 0.0000$

(e) $t = 0.1362$



(f) $t = 0.1900$

(g) $t = 0.2100$

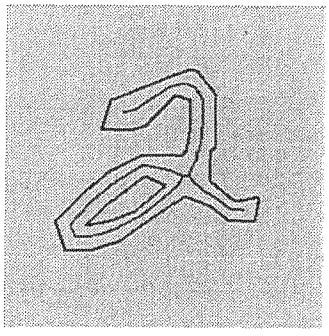


(h) $t = 0.2200$

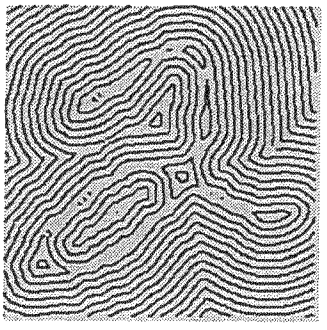
(i) $t = 0.2450$

Figure 1: Object shapes can be recovered from image data by using the level set curvature flow equation. (a)–(c): Recovery of the stomach shape from a CT image of an abdominal section. (d)–(i): A two-stage scheme for shape description of both simple shapes and shapes with holes.

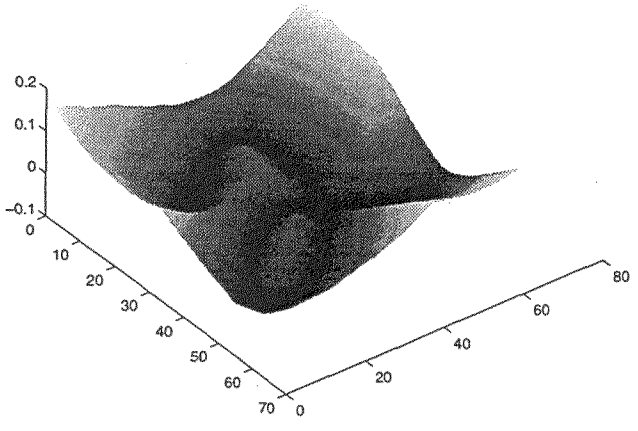
(a) Original image



(b) Boundary & Skeleton



(c) Level sets of ψ

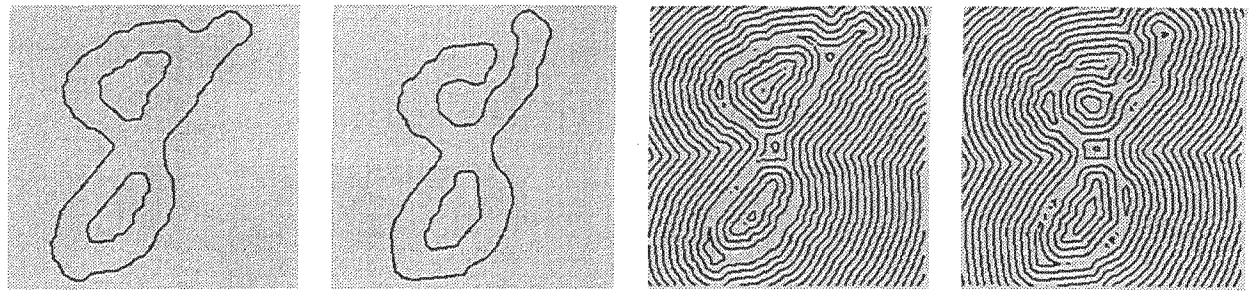


(d) Shaded surface $z = \psi(x, y)$

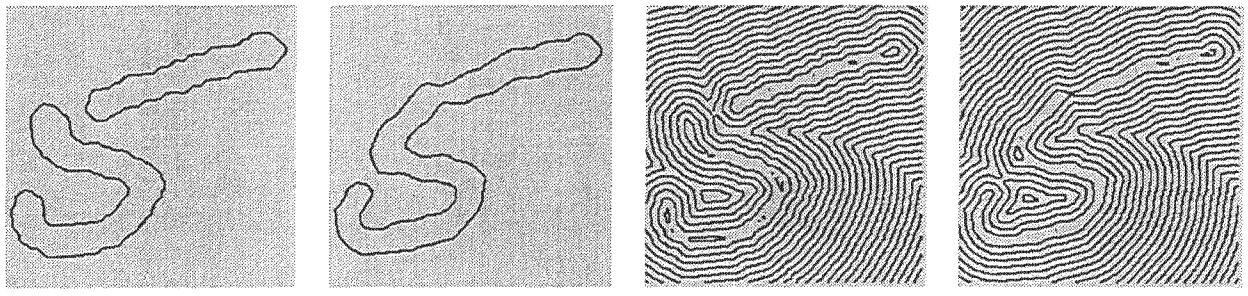
Figure 2: Boundary and region based shape representations are implicitly contained in the signed-distance function ψ.

and in figure 2(d), we render it as a surface $z = \psi(x, y)$. Thus, our shape representation contains both the boundary and region information and more, as depicted by the level sets lying outside the shape boundary (see figure 2(c)).

We would like to emphasize another feature of the above approach. Boundaries of simple shapes, disconnected shapes, and shapes with holes can be embedded alike in the ψ function by exploiting the property that its zero level set need not be simply connected. This is a major advantage considering the vast variation that exists in the way characters are written. Shape boundary can be disconnected even as a result of noise in the image. As an illustration of these variations, consider the



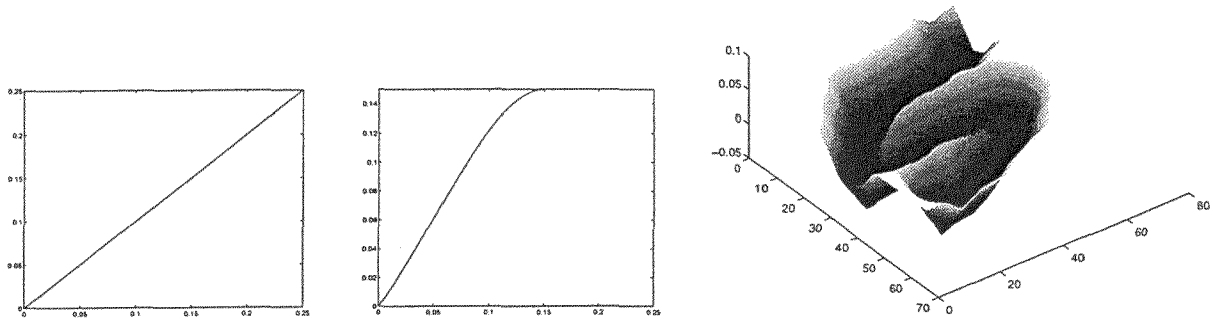
(a) Eight-1 (b) Eight-2 (c) $\psi^{Eight-1}$ (d) $\psi^{Eight-2}$



(e) Five-1 (f) Five-2 (g) ψ^{Five-1} (h) ψ^{Five-2}

Figure 3: Our level set shape representation scheme handles simple shapes, shapes with disconnected boundaries, and shapes with holes in a seamless fashion.

shapes shown in figure (3). The numeral eight in figure 3(a) can be described by one outer boundary and two inner boundaries corresponding to the two holes present in its shape. On the other hand, the eight shape in figure 3(b), although very similar to figure 3(a), has only one hole. Therefore, there is a great difference in their boundary representations. The ψ functions computed from their boundaries, however, are very "close" in the feature space as evidenced by the configuration of their level sets in figures 3(c) and 3(d). This is because by embedding our shape boundary in a higher dimensional function, we are focussing on its global structure instead of adhering to a local one. Next, shapes with disconnected boundaries can also be handled naturally. Figures 3(e) and 3(f) the shapes of numeral five with disconnected and single boundary respectively. Again, their corresponding signed-distance functions, shown in figures 3(g) and 3(h), are very similar. Lastly, the differential properties of the shape like normal and curvature are embedded in the function ψ and can be extracted upon demand.



(a) Original distance (b) Truncated distance (c) $S_{0.07}(\psi)$

Figure 4: Desired amount of shape information can be retained by truncating the distance function.

5 Handwritten Numeral Recognition

In this section, we shall analyze the effectiveness of the signed-distance function in shape recognition. We apply our scheme to the problem of recognizing handwritten numerals from their images. A back-propagation neural network classifier has been used in our recognition system. We train the network on a random set of N sample shapes, store the weights of the network, and test it on N testing samples. Our feature vector is the signed-distance function expressed on a 16×16 grid. Therefore, the neural network is a fully-connected architecture consisting of 256 input nodes, 10 internal nodes, and 10 output nodes corresponding to 10 numeral classes. The neural network we use has been implemented by B. Kalman and is based on the ideas developed in [15]. This version uses the quadratically convergent conjugate gradient descent method for minimizing the error during the training process. In addition, the network minimizes a different error function which according to [15] is superior to the traditional sum of squares error function.

Our numeral samples have been extracted from the NIST special databases 3 and 7 of segmented characters. Prior to training, we perform the following preprocessing steps. First, we perform a noise removal step and trace the boundaries of each numeral using a chain code procedure. Following this, size normalization is done by finding the bounding rectangle of every numeral shape and mapping it into a square region of side 0.5 and centered at the origin. During this processing

step, care must be taken not to artificially skew the shape either horizontally or vertically. Next, the signed-distance function is computed in the square region by considering all the contours found by the chain code procedure.

For large distances, all characters have essentially the same values for the signed-distance function; thus little distinguishing information is contained in these values. Conversely, very close to the boundary, the difference in values for the signed-distance function may be large even between two versions of the same character. Thus, we would like to determine the appropriate amount of information to carry in our classification vector; that is, how much information should we use away from the boundary?

We leave this issue up to empirical tests by capturing the detail in a given region around the zero set by means of a filter function. If we plot the Euclidean distance measured from a given point, we obtain the graph shown in figure 4(a). Define a function S_α as follows:

$$S_\alpha(t) = \begin{cases} t & \text{if } t < 0 \\ \alpha & \text{if } t > \alpha \\ -\frac{t^3}{\alpha^2} + \frac{t^2}{\alpha} + t & \text{otherwise} \end{cases}$$

The plot of $S_\alpha(t)$ with α set to 0.07 is shown in figure 4(b). We thus represent shape by the function $S_\alpha \circ \psi = S_\alpha(\psi(x, y))$. This definition effectively blocks all positive distances greater than α. When $\alpha = 0$, the function $S_\alpha(\psi)$ simply represents the boundary and its interior. As the value of α increases, we gradually approach the full signed-distance function. Figure 4(c) shows the surface of a truncated signed-distance function.

5.1 Step 1: Neural Network Classification

Given the level set signed-distance function description of character shapes, we use a multiple neural net approach to achieve high reliability. First, we train a network on a random set of 3189 numeral samples and upon convergence test it on another set of 3158 samples. Given a testing numeral sample, the trained network classifies it with a confidence value lying in the interval [0..1]. We can improve the reliability by accepting only classifications made with high confidence and discarding the rest. In figure 5(a), we show a plot of reliability vs. percentage discards on a uniform grid. Note that the percentage of correctly classified samples is 95.47 with 0.0% discards and with 12.0% discards, we get 98.27% of them right. We have also trained the neural net on over 10,000 samples

and the classification rates are measurably better. Therefore, we believe that everything that follows also holds for much larger sample sizes.

To understand how the parameter α in the function $S_\alpha(\psi)$ affects the correct classification rate, we train several nets on the truncated distance function each computed with a different value of α. The plot of α vs the zero-discard classification rate is shown in figure 5(b). This curve was generated by finding exact values of percentage correct (zero-discard) at $\alpha = 0.0, 0.025, 0.05, 0.1, 0.15, 0.2, 0.35$, and smoothly interpolating at other points. From this plot it is clear that as we include more distance information by increasing α, the neural network finds it easier to place character samples into their correct classes. We have also noticed that the training time increases with decreasing α. Obviously, this trend does not continue for all values of α. As we mentioned before, the difference between distance values of two shapes becomes fuzzier at points very far from the zero set. We have quantized this notion in the above experiment by showing that the distance information at points farther than 0.25 from the shape boundary does not add to the degree of generalization that the neural net is capable of performing. We place the shape in a square box of size 0.5 and centered at the origin.

Given the optimum box size in which to express the signed-distance function, the next issue is grid size and grid point placement. We discretize the ψ function on a 16×16 grid. Our goal now is to improve the quality of the feature vector that the neural network is training on. One way to do this is to provide uniformly more information by increasing the grid size to say 20×20. But it is known [4] that networks with large feature vectors have poorer generalization capability. Alternately, we could keep the grid size constant and work with different point placements. That way we hope to capture different aspects of the same shape. We examine the following four options: uniform grid, horizontal-stagger grid, vertical-stagger grid, and a grid that is staggered in both directions (see figures 5(c)-(f)). We express the signed-distance functions on these grids and then train a neural network on each one of them. Now, note that although the distance values at a grid point for two different shapes can be close, the configuration of level curves passing through it may be very different. We capture this by computing the angle of the gradient vector on a uniform grid, i.e. the value of $\tan^{-1}(\frac{\partial\psi/\partial y}{\partial\psi/\partial x})$, and train another net on the resulting feature vector. We have added to this list of feature vectors by computing the wavelet (using the Daubechies wavelets [10]) and Fourier

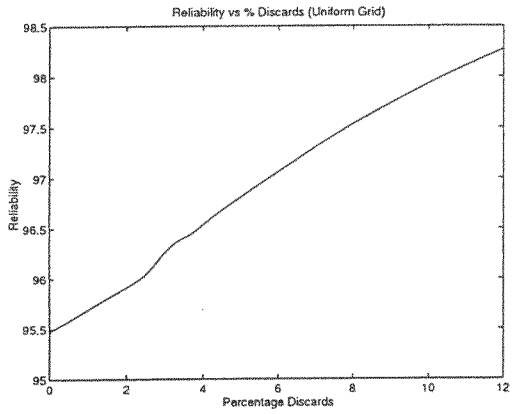
Network	Confidence	% Discard	% Correct	Confidence	% Discard	% Correct
(U)niform	0.5	0.0	95.27	1.0	12.00	98.27
(H)orizontal	0.5	0.0	95.59	1.0	10.13	97.67
(V)ertical	0.5	0.0	95.63	1.0	8.58	97.95
(B)oth directions	0.5	0.0	95.18	1.0	9.97	97.74
(A)ngle	0.5	0.0	89.01	1.0	27.16	94.26
(W)avelet	0.5	0.0	95.75	1.0	13.29	98.28

Table 1: Single-net classification rates with confidence values set to 0.5 and 1.0.

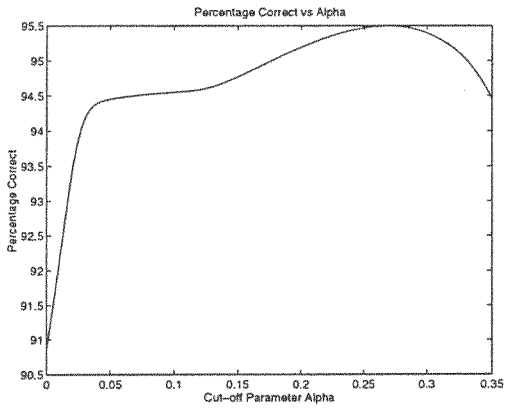
transforms of the function $\psi(x, y)$, but neither helped significantly in terms of our multi-network approach described below.

In Table 1, we show the single-net classification rates resulting from the aforementioned feature vectors trained and tested on 3189 and 3158 samples respectively. To improve the reliability further, we apply the following rule to make a selection: for a pair of nets, say PQ, select a sample for classification if both nets classify it with confidence 1.0 and the classes agree. This multi-network approach is likely to produce higher reliability since our feature vectors capture different aspects of the same shape. We could add to the number of nets that must agree to make a classification, but that will only add to the number of discards. In Table 2, we list the results of passing 3158 samples to all possible pairwise nets.

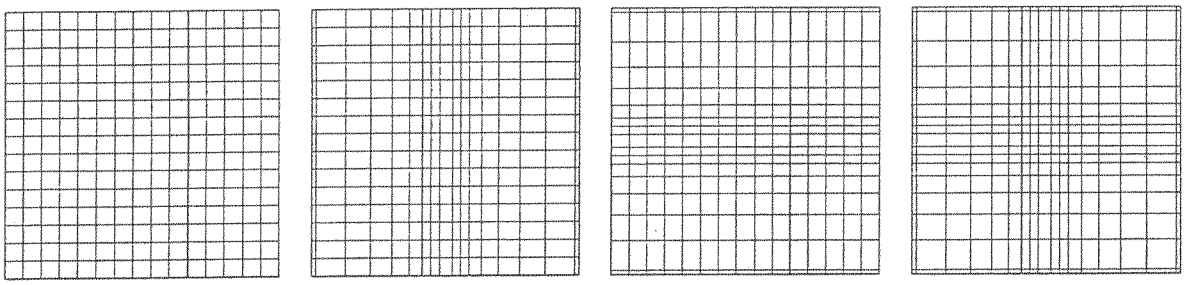
Although the pairwise nets are discarding a large number of samples, it is unlikely that they are discarding the same set of samples. Our next step is to decrease the number of these discarded samples. To this end, we pass the discards generated by the H-A pair to all other pairs. We choose H-A pair because it gives us the highest reliability, i.e., 99.70%. The results of this second pass are given in Table 3. Following this, we put these two stages together to arrive at our final analysis as shown in Table 4. A reliability factor of 99.56% results from the network combination H-A/V-W after discarding 12.88% of the samples. The numeral samples in the above argument have been selected from the NIST database 3 of segmented characters. The unconstrained character shapes in NIST database 7 (see [43]) are known to be much harder to recognize. We have manually classified around 6000 numerals from this database to conduct our experiment. By following a similar multi-neural net approach on 2997 training and 2998 testing samples, we arrive at the results shown in



(a)



(b)



(c) Uniform (d) Horizontal (e) Vertical (f) Both directions

Figure 5: (a): Reliability vs percentage discards on a uniform grid (b): Percentage correct vs the cut-off parameter α (in $S_\alpha(\psi)$) (c): Uniform grid (d): Horizontal-stagger grid (e): Vertical-stagger grid (f): Staggered in both directions

Net-Pair	Chosen	% Discarded	(Correct) % Correct	Wrong
U-A	2009	36.38	(2003) 99.70	6
U-W	2510	20.51	(2496) 99.44	14
U-H	2586	18.11	(2564) 99.14	22
U-V	2614	17.22	(2595) 99.27	19
U-B	2568	18.68	(2552) 99.37	16
H-A	2039	35.43	(2033) 99.70	6
H-W	2550	19.25	(2538) 99.52	12
H-V	2649	16.11	(2626) 99.13	23
H-B	2603	17.57	(2587) 99.38	16
V-A	2066	34.57	(2056) 99.51	10
V-W	2582	18.32	(2570) 99.53	12
V-B	2642	16.33	(2622) 99.24	20
B-A	2043	35.30	(2033) 99.51	10
B-W	2553	19.29	(2535) 99.29	18
A-W	1978	37.36	(1972) 99.69	6

Table 2: Initial pass of 3158 numeral samples to all possible pairwise networks.

Table 5.

6 Perturbing/Deforming Characters

In the previous section, we have shown that by using signed-distance function as a feature vector, numeral samples from NIST databases can be recognized with reliabilities well over 99.00%. However, the above step generates a significant number of discarded samples (more so with database 7). One reason this discard set is large is that we have trained on remarkably few characters; this will be discussed in detail in the conclusions. However, for our purposes now, a highly accurate recognition scheme with sizeable discards is preferable to a less accurate scheme. The reason is that we now wish to examine this discard set. The idea is to deform a character in the discard set in such a way that the deformed shape moves into the acceptance criteria of one of the classification categories.

There are two stages to this process. First, we must devise a set of perturbation/deformations rules which are used to accentuate different aspects of a character's shape. Second, we must develop

Net-Pair	Chosen	% Discarded	(Correct) % Correct	Wrong
U-A	72	93.56	(69) 95.83	3
U-W	671	40.03	(659) 98.21	12
U-H	649	42.00	(630) 97.07	19
U-V	717	35.92	(701) 97.76	16
U-B	703	37.17	(688) 97.86	15
H-A	0	100.00	-	-
H-W	642	42.62	(633) 98.59	9
H-V	676	39.58	(657) 97.18	19
H-B	655	41.46	(642) 98.01	13
V-A	93	91.68	(87) 93.54	6
V-W	712	36.37	(706) 99.15	6
V-B	750	32.97	(732) 97.60	18
B-A	95	91.51	(88) 92.63	7
B-W	705	36.99	(690) 97.87	15
A-W	70	95.71	(67) 95.71	3

Table 3: Results of passing 3158*0.3543 = 1119 discarded samples of the pair H-A to all other network pairs.

acceptance criteria with which to decide if a character is moving towards acceptance.

6.1 Flow Rules

We consider the following basic flows rules. All flows are numerically executed using the Osher-Sethian level set algorithm [32].

- *Modified Curvature Flow:* The interface (that is, the boundary between the inside and outside of a character) is moved inwards in its normal direction according to its curvature. In the curvature flow analyzed by Gage [13] and Grayson [14], convex parts of the curve move inwards while convex sections move outwards. Extensive numerical studies of such flows are given in [38]. This flow serves to smooth out oscillations in the boundary, thereby reducing noise in the shape boundary. We amend this flow as follows: we let the speed F in the normal direction be given by $F = \max(K, 0.0)$. Thus convex parts move inwards, but concave parts stay fixed. As shown below, such a flow preserves some important features of characters.

Net-Combination	Chosen	% Discarded	(Correct) % Correct	Wrong
H-A/U-A	2111	33.15	(2102) 99.57	9
H-A/U-W	2710	14.18	(2692) 99.33	18
H-A/U-H	2688	14.88	(2663) 99.06	25
H-A/U-V	2756	12.72	(2734) 99.20	22
H-A/U-B	2742	13.17	(2721) 99.23	21
H-A/H-A	-	-	-	-
H-A/H-W	2681	15.10	(2666) 99.44	15
H-A/H-V	2715	14.02	(2690) 99.07	25
H-A/H-B	2694	14.69	(2675) 99.29	19
H-A/V-A	2132	32.48	(2120) 99.43	12
H-A/V-W	2751	**12.88**	(2739) **99.56**	12
H-A/V-B	2789	11.68	(2765) 99.13	24
H-A/B-A	2134	32.42	(2121) 99.39	13
H-A/B-W	2744	13.10	(2723) 99.23	21
H-A/A-W	2109	33.21	(2100) 99.57	9

Table 4: Results of applying our multi-neural network classification approach to 3158 numeral samples from NIST database #3.

Net-Combination	Chosen	% Discarded	(Correct) % Correct	Wrong
H-V/U-A	1672	44.22	(1661) 99.34	11
H-V/U-H	1735	42.12	(1721) 99.19	14
H-V/U-V	1561	47.93	(1549) 99.23	12
H-V/U-B	1839	**38.65**	(1824) **99.18**	15
H-V/H-A	1683	43.86	(1670) 99.22	13
H-V/H-V	-	-	-	-
H-V/H-B	1835	38.79	(1817) 99.01	18
H-V/V-A	1536	48.76	(1526) 99.34	10
H-V/V-B	1610	46.29	(1596) 99.13	14
H-V/A-B	1779	40.66	(1764) 99.15	15

Table 5: Results of applying our multi-neural network classification approach to 2998 numeral samples from NIST database #7.

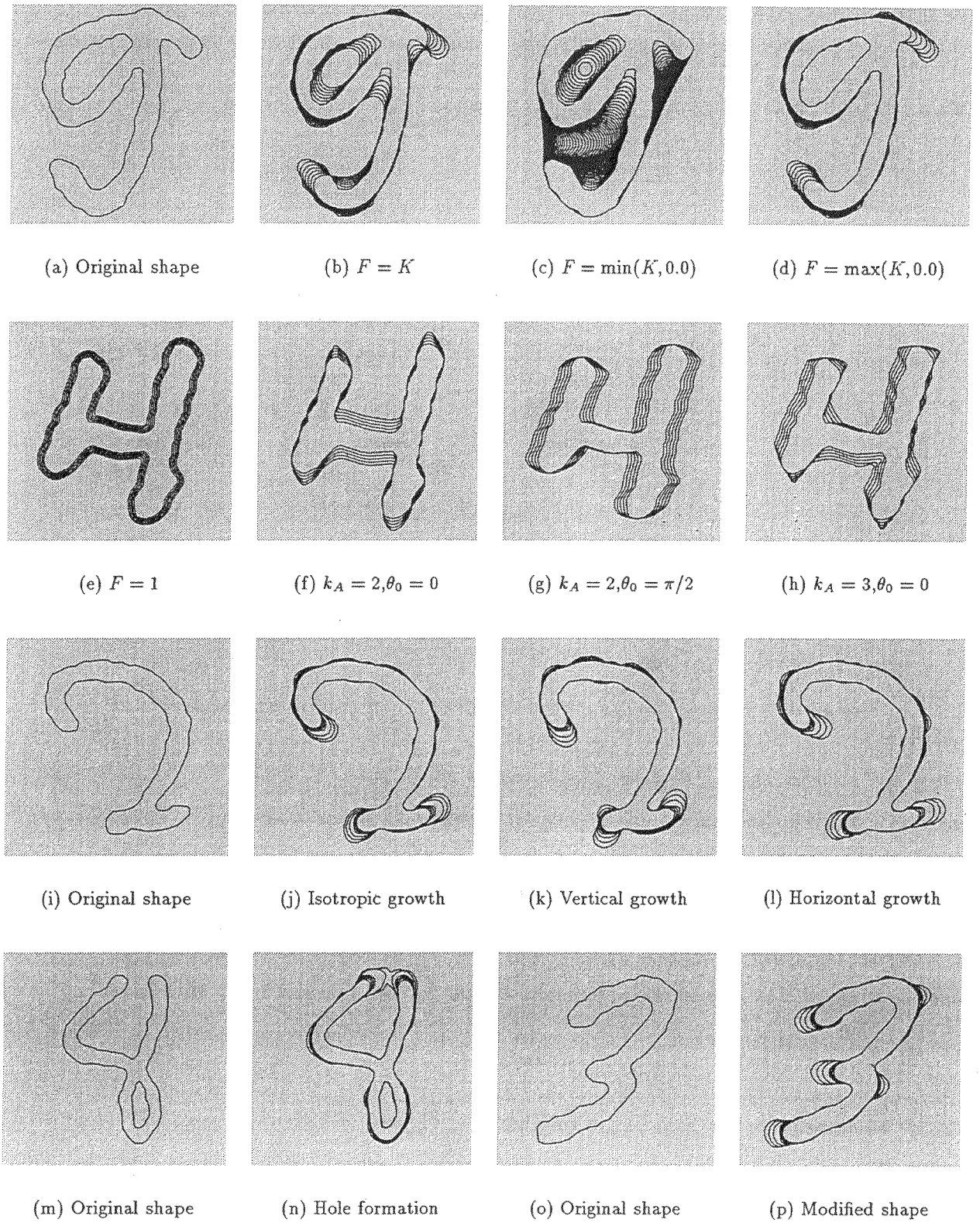
- *Outwards Constant Flow:* The interface propagates outwards with constant speed, see [36, 32, 38].

- *Horizontal Anisotropic Flow:* As discussed in [39], anisotropic flow is an integral part of crystal growth and dendritic solidification. The anisotropic symmetry gives preferred directions of growth. Here we consider two-fold symmetry in the horizontal direction, which means that shapes move under speed laws which prefer to move them in horizontal directions.

- *Vertical Anisotropic Flow:* Same as the above, except flow in the vertical direction is preferred.

- *Growing Limbs:* One would like a flow that performs the opposite of the curvature flow, in that limbs (regions of high positive curvature) grow outwards. However, as stated in [36], this is the backwards heat equation which is ill-posed and numerically unstable. We instead invent a variation on this idea which grows limbs for a short time. Begin by flowing under curvature flow for a given number of time steps. Then, using this final shape and the initial shape, interpolate backwards in time to a previous state, which will have larger "limbs" at places where the initial curvature is positive. Repeat the process of curvature flow to smooth oscillations in the front, plus backwards time interpolation to further extend limbs. This process is unstable, in that refining the time step produces uncontrollable oscillations. However, done on a coarse enough scale, it produces shapes which exaggerate and accentuate limbs and protrusions.

In the next set of figures, we demonstrate the motion of various characters in the discard set under these flow rules. First, consider moving a shape boundary with speed $F(K)$. Consider the shape of numeral nine in figure 6(a). This shape has protruding arms and therefore could not be accepted with high confidence. We move its boundary with speed $F = K$ as shown in figure 6(b). Convex regions move inward and concave regions outward. The protrusions are smoothed out, but we also lose the essential shape features in the bargain. Next, we move it with speed $F = \min(K, 0.0)$ as shown in figure 6(c). This flow is not very useful because concave parts move out, the hole disappears, and the shape becomes the convex hull of the enclosed points in the limit $t \to \infty$. On the other hand, the flow with speed $F = \max(K, 0.0)$, shown in figure 6(d), retains holes present in a shape, and will smooth out the protrusions in the boundary.



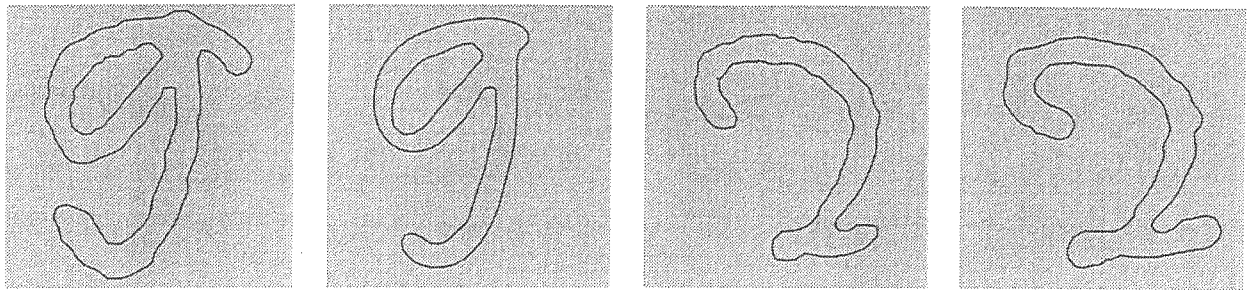
(a) Original shape (b) $F = K$ (c) $F = \min(K, 0.0)$ (d) $F = \max(K, 0.0)$

(e) $F = 1$ (f) $k_A = 2, \theta_0 = 0$ (g) $k_A = 2, \theta_0 = \pi/2$ (h) $k_A = 3, \theta_0 = 0$

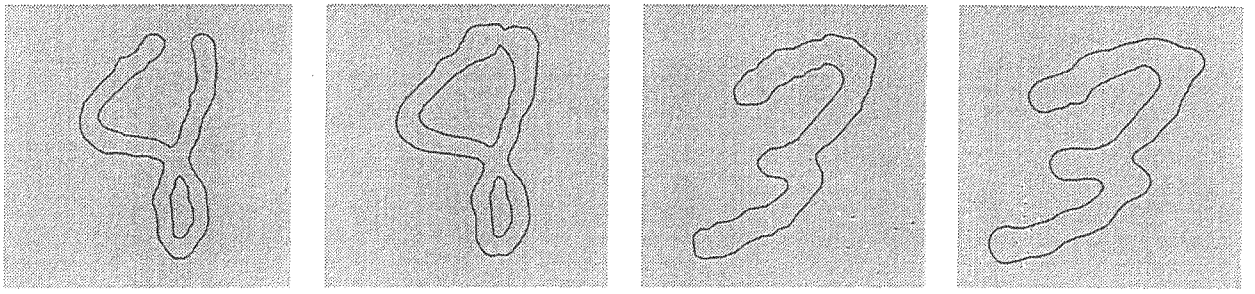
(i) Original shape (j) Isotropic growth (k) Vertical growth (l) Horizontal growth

(m) Original shape (n) Hole formation (o) Original shape (p) Modified shape

Figure 6: Results of moving character shapes under various flow rules. (b)–(d): curvature flow, (e): constant expansion, (f)–(h): anisotropic expansion, (j): limb growing, (k), (l), (n), (p): anisotropic growth.



(a) Original shape (b) Perturbed shape (c) Original shape (d) Perturbed shape



(e) Original shape (f) Perturbed shape (g) Original shape (h) Perturbed shape

Figure 7: Comparison between original and perturbed character shapes.

Now, we consider moving a given shape boundary at constant speed, $F = 1.0$, and under anisotropy which is modeled by multiplying the speed with the coefficient $(1 - A\cos(k_A\theta + \theta_0))$. Here, θ is the angle between normal $\vec{n}$ and the x-axis. If $A > 0$, the boundary moves with k_A-fold symmetry with the symmetric axis making an angle θ_0 with the x-axis (see [39]). In figure 6(e), we show the boundary of numeral four moving with unit speed. Expansion can be carried out in preferred directions by setting the parameters k_A and θ_0; two-fold symmetry with $\theta_0 = 0$ in figure 6(f), tow-fold symmetry with $\theta_0 = \pi/2$ in figure 6(g), and three-fold symmetry with $\theta_0 = 0$ in figure 6(h).

Next, we investigate the idea of growing limbs in regions of high curvature. For some badly written characters, this flow rule will serve to accentuate their prominent shape features. As an example, consider the shape of numeral two in figure 6(i). The idea is to flow under the speed $F = \max(K, 0.0)$ for a small number of time steps and then interpolate backwards in time to a previous state. This cycle can be repeated for a couple of times for further growth. By appending the anisotropic term to the speed, i.e., $F = (1 - A\cos(k_A\theta + \theta_0))\{\max(K, 0.0)\}$, we can also achieve

growth in preferred directions. In figure 6(j) we show three steps of isotropic growth, i.e., with $A = 0$, and in figures 6(k) and 6(l), we show anisotropic growth with two-fold symmetry in vertical and horizontal directions respectively. Anisotropic limb growing also closes off a hole in the numeral eight (see figures 6(m) and 6(n)). Finally, in figures 6(o) and 6(p), we depict the initial and modified shapes of numeral three which has been perturbed using horizontal growth rule. For comparison, in figure (7), we place the original and the perturbed shapes of these numerals beside one another.

6.2 Step 2: Processing the Discarded Samples

In the second step of our recognition scheme, we take the discarded samples from the previous step and flow them under the influence of various flow rules discussed above. A numeral sample is a member of the discard set either if one of the nets does not classify it with high confidence or if the classifications of a net pair do not agree. Our idea is to steer the character sample into acceptance by perturbing its shape. However, not all character shapes move into correct category upon deformation. Therefore, we have to device an acceptance criterion.

To begin, consider the flow rule $F = \max(K, 0.0)$. Given a character, we create three feature vectors each corresponding to three time instants in its evolution. Let v_0 be the initial feature vector and v_1, v_2, and v_3 be the feature vectors collected after flowing the initial shape for 50, 100, and 150 time steps respectively. We perform the flow calculations using the Osher-Sethian algorithm on a 64×64 grid with a time step $\Delta t = 0.000025$. Although the flow computations are done on a finer grid, we finally express the signed-distance of the resulting shapes on a 16×16 uniform grid. Following this we present the four feature vectors to the uniform grid neural network. Let s_0, s_1, s_2, and s_3 be the confidence values (in the range [0..1]) with which the network classifies the shapes as c_0, c_1, c_2, and c_3 respectively. We accept this classification if flowing has not changed the class, i.e., all c_is are the same, and the network classifies them with increasing confidence; in other words, $s_3 \geq s_2 \geq s_1 \geq s_0$. If the above conditions are not met, we discard that character.

Consider the $(0.3865 * 2998 =)$ 1159 discarded characters from NIST database 7. When we flow them under $F = \max(K, 0.0)$ and apply the acceptance criterion described above, we classify 93.96% of them correctly after discarding 54.27% of the shapes. On NIST database 3, on applying the same rule to $(0.1288 * 3158 =)$ 407 discarded shapes, we classify 92.78% of them correctly after

Flow Rule	NIST Database#	% Discarded	% Correct
$F = \max(K, 0.0)$	7	54.27	93.97
$F = \max(K, 0.0)$	3	48.89	92.78
Horizontal Growth	7	62.81	86.77
Horizontal Growth	3	54.32	81.47
Vertical Growth	7	45.85	80.06
Vertical Growth	3	57.45	75.70

Table 6: Flow results on the discarded numeral shapes from NIST databases 3 and 7. The percentages were calculated on 1159 samples for database 7 and 407 samples for database 3.

discarding 48.89% of the shapes. These results are shown in Table 6.

Next, we grow limbs with anisotropy. We only consider two-fold symmetry with $\theta_0 = 0$ (vertical growth) and $\theta = \pi/2$ (horizontal growth). In both cases, for each character, we create three feature vectors g_0, g_1, and g_2; g_0 corresponds to the initial shape and g_1, g_2 are collected after each successive growing step. Following the same procedure as above, we present them to the uniform net to obtain three classes with corresponding confidence values. Now, we accept a classification if the classes agree and the network classifies them with increasing confidence. The results of this flow procedure are also shown in Table 6.

Finally, for database 3, when we put the two steps together, multi-neural net and flow under $F = \max(K, 0.0)$, we correctly recognize 99.08% of the numerals after discarding 6.30% of them. The figures for database 7 are 98.05% correct with a discard set of 20.98%. As further experiments, we could continue this two-step recognition procedure a step further by presenting the discard set of one flow to the other. For example, if we grow horizontal limbs on the (2998*.2098=) 629 discarded samples from database 7 and put results from three stages together, we get 96.20% correct after discarding 13.00% of the numerals. Therefore, while the classification rates will remain high as one chains these procedures together, there is a limit to how much the discard set can be reduced; since at some point the discard set will consist almost entirely of characters "too far" from the training set of the neural net scheme. Thus, for the final set of discards one might choose to switch from a neural net approach to an extensive search like the K nearest neighbor approach [8]. Figure 8 shows a sample of numeral shapes that have been recognized correctly by our algorithm.

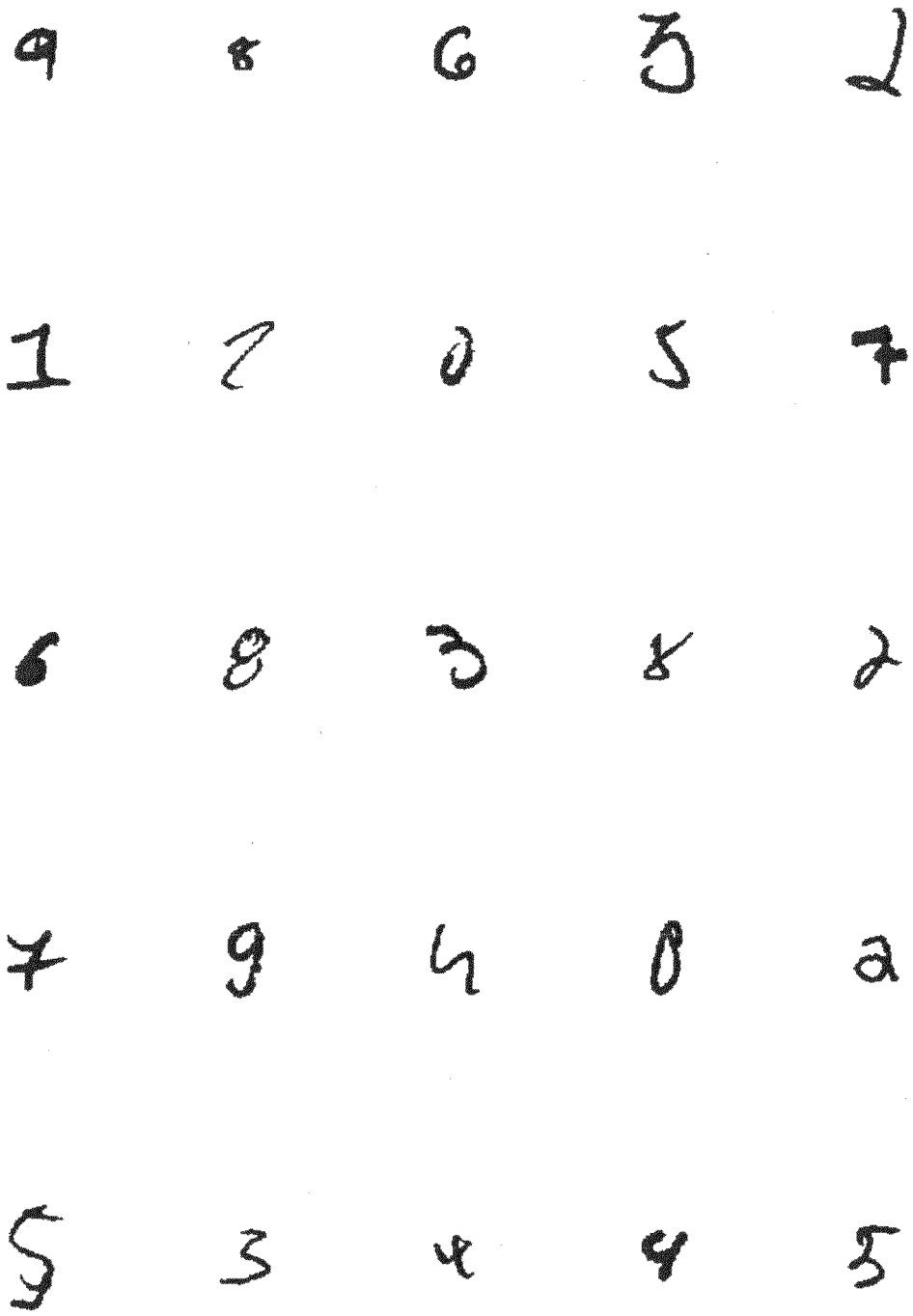


Figure 8: A sample set of numeral shapes that have been correctly classified.

7 Conclusions and Discussion

In this paper, we have tried to show that the level set formulation provides a very good shape descriptor for object recognition. Furthermore, deformation in the shape may be easily performed to search for nearby shapes which are easily classified. For the NIST database 3, we achieve recognition rates of 99.08% with 6% discard; for NIST database 7, we achieve 98.05% discard with around 20% discard.

We believe that these results are excellent. Most commercial packages achieve accuracy values similar to ours, but on a very hard database such as the NIST database 7, and their discard rates are about half of ours; see [43] for extensive testing of various optical character recognition techniques on the same databases. However, we view this as a very important point. Those schemes typically trained on over 100,000 characters; this is a sample training set 30 times larger than ours. Roughly speaking, they have over 100,000 known points where the shape of a character is linked to its classification category.

In contrast, we have trained on only about 3000 characters. There are several reasons for this. First, we wanted to show that the level set function is such a powerful shape descriptor that few known examples between trial shapes and classification categories are enough to obtain very high classification rates. Second, we wanted a significant enough discard set so that the idea of identifying shapes through deformation could be explored. Third, we lacked computing resources to perform a training run on a very large sample.

We believe that a neural net trained on 100,000 using the above level set shape descriptor and secondary classification scheme under reformation motions will yield significant improvement in our final discard rates; first, because more elements will be identified on the first pass, and second, because the deformation mechanism will be able to locate a familiar shape in the much denser space of known test examples. We are in the process of performing this classification now using a public domain neural net code; results of this work will be reported elsewhere [24].

Acknowledgements

We thank Sergio Schuler for helping to procure the neural net code and Adam Greenberg, Niranjan Mayya, Steve Smith and Lew Tucker for many fruitful discussions. All computations were done at the Lawrence Berkeley Laboratory and the University of California, Berkeley.

References

[1] D. Adalsteinsson and J. A. Sethian, "A fast level set method for propagating interfaces," submitted for publication, *Journal of Computational Physics*, 1994.

[2] L. Alvarez, P. L. Lions, and J. M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion. II," *SIAM Journal on Numerical Analysis*, Vol. 29(3), pp. 845–866, 1992.

[3] K. Arbter, W. E. Snyder, H. Burkhardt, and G. Hirzinger, "Application of affine-invariant Fourier descriptors to recognition of 3-D objects," *IEEE Trans. on Patt. Anal. and Mach. Intell.*, Vol. 12, No. 7, pp. 640–647, July 1990.

[4] E. B. Baum and D. Haussler, "What size net gives valid generalizations?," *Neural Computations*, Vol. 1, pp. 151–160, 1989.

[5] H. Blum, "A transformation for extracting new descriptors of shape," *Models for the Perception of Speech and Visual Form* (W. Wathen-Dunn, ed.), Cambridge MA: MIT Press, 1967.

[6] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," Internal report no. 9210, CEREMADE, Université de Paris-Dauphine, France.

[7] D. L. Chopp, "Computing minimal surfaces via level set curvature flow," *Journal of Computational Physics*, Vol. 106, pp. 77–91, 1993.

[8] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inform, Theory*, Vol. 13, pp. 21–27, 1967.

[9] T. Crimmins, "A complete set of Fourier descriptors for two-dimensional shapes," *IEEE Trans. on Syst. Man, and Cyber.*, Vol. SMC-12, No. 6, 1982.

[10] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. on Pure and Applied Math.*, Vol. 41, pp. 909–996, 1988.

[11] H. Freeman, "On the encoding of arbitrary geometric configurations," *IEEE Trans. on Electronic Computers,* Vol. EC-10, pp. 260–268, 1961.

[12] R. C. Gonzalez and P. Wintz, *Digital Image Processing (2nd Ed.),* Addison-Wesley, 1987.

[13] M. Gage, "Curve shortening makes convex curves circular," *Inventiones Mathematica*, Vol. 76, pp. 357, 1984.

[14] M. Grayson, "The heat equation shrinks embedded plane curves to round points," *J. Diff. Geom.*, Vol. 26, pp. 285–314, 1987.

[15] B. L. Kalman, "Super linear learning in back propagation neural nets", Report WUCS-90-21, Dept. of Computer Science, Washington University, St. Louis, June 1990.

[16] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, pp. 321–331, 1988.

[17] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, "Toward a computational theory of shape: An overview," in *Proceedings of ECCV*, Antibes, France, 1990.

[18] R. Kimmel, D. Shaked, N. Kiryati, and A. M. Bruckstein, "Skeletonization via distance maps and level sets," EE PUB # 933, Dept. of Electrical Engg., TECHNION – Israel Inst. of Tech., Haifa Israel, Aug. 1994.

[19] A. F. Laine, S. Schuler, and V. Girish, "Wavelet representation for recognizing complex annotations," *Machine Vision and Applications*, Vol. 6, pp. 110–123, 1993.

[20] D. T. Lee, "Medial axis transformation of a planar shape," *IEEE Trans. Patt. Anal. Machine Intell.* PAMI-4, No. 4, pp. 363–369, 1982.

[21] F. Leymarie and M. D. Levine, "Simulating the grassfire transform using an active contour model," *IEEE Trans. Patt. Anal. Machine Intell.*, Vol. 14, No. 1, pp. 56–75, 1992.

[22] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Evolutionary fronts for topology-independent shape modeling and recovery," in *Proceedings of Third European Conference on Computer Vision*, LNCS Vol. 800, pp. 3–13, Stockholm, Sweden, May 1994.

[23] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," Report LBL-35733, Lawrence Berkeley Laboratory, University of California, Berkeley, June 1994. To appear in *IEEE Trans. on Pattern Analysis and Machine Intelligence.*

[24] R. Malladi and J. A. Sethian, "Level set shape descriptor for OCR," under preparation.

[25] D. Marr and H. K. Nishihara, "Visual information processing: Artificial Intelligence and the sensorium of sight," *Technology Review*, Vol. 81(1), Oct. 1978.

[26] D. Marr and E. Hildreth, "A theory of edge detection," *Proc. of Royal Soc. (London)*, Vol. B207, pp. 187–217, 1980.

[27] N. Mayya and V. T. Rajan, "Voronoi diagrams of polygons: A framework for shape representation," IBM Research Report, RC 19282, Nov. 1993 (to appear in the *Journal of Mathematical Imaging and Vision*).

[28] R. Mohan and R. Nevatia, "Segmentation and description based on perceptual organization," in *Proc. IEEE Conf. Comp. Vision and Pattern Recog.*, pp. 333–334, June 1989.

[29] S. Mori, C. Y. Suen, and K. Yamamoto, "Historical review of OCR research and development," *Proc. of the IEEE*, Vol. 80, No. 7, pp. 1029–1057, 1992.

[30] R. Ogniewicz and M. Ilg, "Voronoi skeletons: Theory and application," *Proc. of ICPR*, pp. 63–69, 1992.

[31] S. Osher and L. Rudin, "Feature-oriented image enhancement using shock filters," *SIAM J. Numerical Anal.*, Vol. 27, pp. 919–940, 1990.

[32] S. Osher and J. A. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation," *Journal of Computational Physics*, Vol. 79, pp. 12-49, 1988.

[33] T. Pavlidis, "Polygonal approximations by Newton's method," *IEEE Trans. on Computers*, Vol. C-26, No. 8, pp. 800–807, 1977.

[34] E. Pearson and K. S. Fu, "Shape discrimination using Fourier descriptors," *IEEE Trans. System, Man, and Cyber.*, Vol. SMC-7, No. 3, pp. 170–179, 1977.

[35] C. Rhee, L. Talbot, and J. A. Sethian, "Dynamical study of a premixed V flame," to appear, *J. Fluid Mechanics.*

[36] J. A. Sethian, "Curvature and the evolution of fronts," *Commun. in Math. Physics*, Vol. 101, pp. 487–499, 1985.

[37] J. A. Sethian, "Curvature flow and entropy conditions applied to grid generation," to appear *Journal Comp. Phys.*, Dec. 1994.

[38] J. A. Sethian, "Numerical algorithms for propagating interfaces: Hamilton-Jacobi equations and conservation laws," *Journal of Differential Geometry*, Vol. 31, pp. 131–161, 1990.

[39] J. A. Sethian and J. Strain, "Crystal growth and dendritic solidification," *Journal of Computational Physics*, Vol. 98, pp. 231–253, 1992.

[40] L. H. Staib and J. S. Duncan, "Boundary finding with parametrically deformable models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 11, pp. 1061–1075, 1992.

[41] C. Y. Suen, C. Nadal, R. Legault, T. A. Mai, and L. Lam, "Computer recognition of unconstrained handwritten numerals," *Proc. of the IEEE*, Vol. 80, No. 7, pp. 1162–1180, 1992.

[42] B. Widrow, "The rubber mask technique-I and II," *Pattern Recognition*, Vol. 5, pp. 175–211, 1973.

[43] R. A. Wilkinson *et al.*, "The first census OCR system conference", U.S. Dept. of Commerce, NIST, #NISTIR 4912, Gaithersburg, MD, August 1992.