TWO FRONTIERS IN MORPHOLOGICAL IMAGE ANALYSIS: DIFFERENTIAL EVOLUTION MODELS AND HYBRID MORPHOLOGICAL/LINEAR NEURAL NETWORKS *

PETROS MARAGOS[1&2], M. AKMAL BUTT[1], AND LÚCIO F. C. PESSOA[3]

[1] School of E.C.E., Georgia Institute of Technology, Atlanta, GA 30332, USA.
[maragos,akmal,lpessoa]@ee.gatech.edu
[2]Institute for Language and Speech Processing, Artemidos/Epidavrou Str., Marousi 15125, Greece.
[3] Motorola Semiconductor Products Sector, Austin, TX 78721, USA.

Abstract. In this paper we briefly describe advancements in two broad areas of morphological image analysis. **Part I** deals with differential morphology and curve evolution. The partial differential equations (PDEs) that model basic morphological operations are first presented. The resulting dilation PDE, numerically implemented by curve evolution algorithms, improves the accuracy of morphological multiscale analysis by Euclidean disks and (its anisotropic/heterogeneous version) is the basic ingredient of PDE models that solve image analysis problems such as gridless halftoning and watershed segmentation based on the eikonal PDE. **Part II** deals with morphology-related systems for pattern recognition. It presents a general class of multilayer feed-forward neural networks where the combination of inputs in every node is formed by hybrid linear and nonlinear (of the morphological/rank type) operations. For its design a methodology is formulated using ideas from the back-propagation algorithm and robust techniques are developed to circumvent the non-differentiability of rank functions. Experimental results in handwritten character recognition are described and illustrate some of the properties of this new type of neural nets.

Keywords: differential morphology, PDEs, curve evolution, neural nets, character recognition.

1 PART I: DIFFERENTIAL MORPHOLOGY AND CURVE EVOLUTION

Morphological image processing has been based traditionally on set and lattice theory. Thus, so far, the two classic approaches to analyze or design deterministic morphological operators have been: (i) *geometry* by viewing them as image set transformations in Euclidean spaces and (ii) *algebra* to analyze their properties using set or lattice theory. In parallel to these directions, there is a recently growing part of morphological image processing that uses tools from differential calculus and dynamical systems to model nonlinear multiscale analysis and distance propagation in images.

Recently, the multiscale morphological operators of dilation, erosion [1, 5, 24] and opening, closing [5] were modeled via nonlinear partial differential equations (PDEs) acting in scale-space. These advancements were inspired by previous work in computer vision where multiscale linear convolutions of an image were modeled via the heat PDE. For multiscale flat dilations and erosions of an image $f(x,y)$ by compact convex symmetric structuring sets $B \subseteq \mathbb{R}^2$ at a continuum of scales $s \geq 0$, their generating PDEs have the form

$$\partial\Psi/\partial s = \pm||\nabla\Psi||_B \quad , \quad \Psi(x,y,0) = f(x,y) \qquad (1)$$

where $\Psi(x,y,s)$ is the dilation $\oplus$ or erosion $\ominus$ of f by sB, $+/-$ corresponds to dilation/erosion respectively, $||(x,y)||_B \equiv \sup_{(a,b)\in B}(ax+by)$, $\nabla\Psi \equiv (\Psi_x, \Psi_y)$ is the spatial gradient, and $\Psi_x \equiv \partial\Psi/\partial x$ denote partial derivatives. For example, if B is the unit disk, $||\cdot||_B$ is the Euclidean norm $||\cdot||$ and

$$\partial\Psi/\partial s = \pm||\nabla\Psi|| = \pm\sqrt{(\Psi_x)^2 + (\Psi_y)^2} \qquad (2)$$

Even if the initial image f is smooth, at finite scales $s > 0$ the above dilation or erosion evolution may create discontinuities in the derivatives, like 'shocks'. Thus, the dilations $f \oplus sB$ or erosions $f \ominus sB$ are *weak solutions* of (1). Ways to deal with these shocks include replacing standard derivatives with morphological derivatives [5, 11] or replacing the PDEs with differential inclusions [13].

In parallel to the development of the above ideas, there have been some advances in the field of differential geometry for evolving curves or surfaces using level set methods [17, 23]. Consider at time $t = 0$ an initial simple, smooth, closed planar curve $\Gamma(0)$ which is propagated for $t > 0$ along its normal vector field with speed c. Let this evolving curve $\Gamma(t)$ be represented by its position vector $\vec{X}(p,t) = (x(p,t), y(p,t))$ and parameterized by $p \in J$ so that it has its interior on the left in the direction of increasing p. A general propagation law is

$$\frac{\partial\vec{X}(p,t)}{\partial t} = c\ \vec{N}(p,t) \quad , \quad \Gamma(0) = \{\vec{X}(p,0) : p \in J\}, \quad (3)$$

*This work was done mostly at Georgia Tech and was supported by the US NSF under grant MIP–94-21677 and by the ARO under grant DAAH04-96-1-0161. L. Pessoa was also supported by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), Brasília, Brazil, through a Doctoral Fellowship under grant 200.846/92-2. The paper was written while P. Maragos was at I.L.S.P.

Invited Paper. Proceedings of Int'l Symposium on Computer Graphics, Image Processing & Vision (SIBGRAPI-98), Brazil, Oct. 1998.

where $\vec{N}(p,t)$ is the instantaneous unit outward *normal* vector at points on the evolving curve, and $c = c(x,y,t)$ is the *speed* function which generally depends on local geometrical information such as the *curvature* $\kappa(p,t)$, global image properties, or other factors independent of the curve. If $c = 1$ or $c = -1$, then $\Gamma(t)$ is the dilation or erosion of the initial curve $\Gamma(0)$ by a disk of radius t. The speed model $c = 1 \pm \varepsilon\kappa$, $|\varepsilon| \leq 1$, has been extensively studied in [17, 23] for general evolution of boundaries and interfaces and in [8] for shape analysis in computer vision.

To overcome the topological problem of splitting and merging and numerical problems with the Lagrangian formulation (3), an Eulerian formulation was proposed in [17] where the original curve $\Gamma(0)$ is first embedded in the surface of an arbitrary 2D Lipschitz continuous function $\Phi_0(x,y)$ as its zero-level curve. (For example, we select $\Phi_0(x,y)$ to be equal to the signed ($\pm$) distance function from the boundary of $\Gamma(0)$ where $+$ is for points inside and $-$ is for points outside the curve.) Then, the evolving 2D curve is obtained as the zero-level curve $\Gamma(t) = \{(x,y) : \Phi(x,y,t) = 0\}$ of a 2D function $\Phi(x,y,t)$ that evolves according to the PDE

$$\partial\Phi/\partial t = c||\nabla\Phi|| \quad , \quad \Phi(x,y,0) = \Phi_0(x,y) \qquad (4)$$

This function evolution PDE makes all level sets expand at normal speed c. If $c = \pm 1$, it is identical to the flat circular dilation/erosion PDE (2) by equating scale s with time t.

1.1 Multiscale Analysis via Dilation PDE

Many applications of mathematical morphology [22, 12] such as nonlinear smoothing, geometrical feature extraction, skeletonization, size distributions, and segmentation, inherently require or can benefit from performing morphological image operations at multiple scales, which creates a morphological scale-space. For binary images, the **distance transform** is a compact way to represent their multiscale dilations and erosions by convex structuring elements whose shape depends upon the norm used to measure distances. Thresholding the distance transform at level $t > 0$ yields the morphological erosion of the image foreground (or the dilation of the background) by the norm-induced ball of radius (scale) t. To obtain isotropic distance propagation, the *Euclidean distance transform* is desirable because it gives multiscale morphology with the disk as the structuring element. However, it has a significant computational complexity. *Discrete approaches* use various techniques to obtain integer approximations to the Euclidean distance transform at a lower complexity. Notable such examples are the **chamfer metrics** [3], computed by running recursive min-sum difference equations over the image and thus propagating local distances within a neighborhood mask. Their associated unit ball is a polygon whose approximation of the disk improves by increasing the size of the mask and optimizing the local distances. In this paper, for optimal chamfer transforms we shall use the local distances found in [6].

The *continuous approach* uses the **dilation PDE** (2). This applies to both graylevel and binary images, because flat dilations commute with thresholding and hence, when a graylevel image is dilated, each one of its thresholded versions representing a binary image is simultaneously dilated by the same element and at the same scale. Thus, the dilation PDE plays a fundamental role for modeling morphological scale-space. However, its usefulness is greatly amplified by the existence of a stable and shock-capturing algorithm [17] for the **numerical implementation** of (4). Its main steps are :

- Let $\Phi^n_{i,j}$ be an estimate of $\Phi(i\Delta x, j\Delta y, n\Delta t)$ on a grid.
- $D^+_x = (\Phi^n_{i+1,j} - \Phi^n_{i,j})/\Delta x \ , \ D^-_x = (\Phi^n_{i,j} - \Phi^n_{i-1,j})/\Delta x$
- $D^+_y = (\Phi^n_{i,j+1} - \Phi^n_{i,j})/\Delta y \ , \ D^-_y = (\Phi^n_{i,j} - \Phi^n_{i,j-1})/\Delta y$
- $H^2 = \min^2(0, D^-_x) + \max^2(0, D^+_x) + \min^2(0, D^-_y) + \max^2(0, D^+_y)$
- $\Phi^{n+1}_{i,j} = \Phi^n_{i,j} + C^n_{i,j}|H|\Delta t \quad , \quad n = 1, 2, ..., (T_{max}/\Delta t)$

where T_{max} is the maximum time (scale) of interest, $\Delta x, \Delta y$ are the spatial grid spacings, Δt is the time (scale) step, and $C^n_{i,j} = c(i\Delta x, j\Delta y, n\Delta t)$. Thus, by choosing fine grids (and possibly higher order terms) an arbitrarily low error (between signal values on the continuous plane and the discrete grid) can be achieved in implementing morphological operations involving disks as structuring elements. This is a significant advantage of the PDE approach, as observed in [2, 20]. Thus, curve evolution provides a geometrically better implementation of multiscale morphological operations with the disk-shaped structuring element. See Fig. 1 for an example.

1.2 Eikonal PDE

Many tasks for extracting information from visible images have been related to eikonal optics and wave propagation via the **eikonal PDE** [4]

$$||\nabla U(x,y)|| = \eta(x,y) \qquad (5)$$

Its solution $U(x,y)$ can provide 3D shape, contour halftoning, or topographic segmentation of an image $f(x,y)$ by choosing the refractive index field $\eta(x,y)$ to be an appropriate function of the image [7, 25, 18, 16]. The eikonal PDE can be seen as a stationary formulation of the function evolution PDE (4) with positive speed $c(x,y) = c_0/\eta(x,y)$. Namely [17], if $T(x,y)$ is the time at which the zero level of $\Phi(x,y,t)$ crosses (x,y), then $||\nabla T|| = 1/c$. Setting $U = c_0 T$ leads to the eikonal.

The solution of the eikonal PDE can be viewed as a *weighted distance function* [4, 25, 19, 9, 11] between a point (x,y) and the sources along a path of minimal optical length. The optical length of any path is obtained by integrating the refractive index field $\eta(x,y)$ along this path and is proportional to the time required for light to travel this path. Thus, we can view the solution $U(x,y)$ to the eikonal as a **gray-weighted distance transform (GWDT)** whose values at each pixel give the minimum distance from the light sources weighted by the gray values of the refractive index field. Next we outline two ways of solving the eikonal PDE.

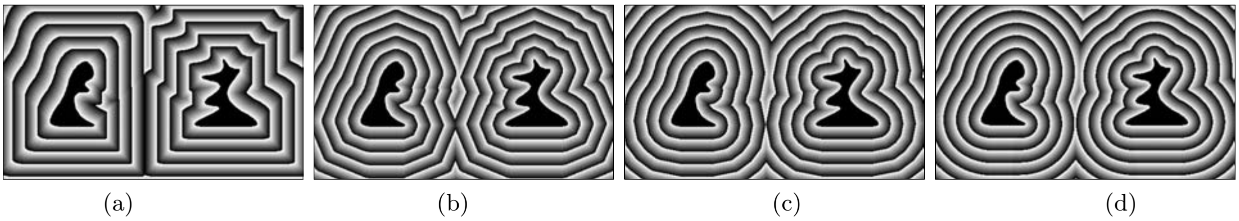

Figure 1. Distance transforms (modulo a constant) of a binary image, obtained via: (a) (1,1) chamfer metric; (b) optimal 3 × 3 chamfer metric; (c) optimal 5 × 5 chamfer metric; (d) curve evolution.

1.2.1 GWDT based on Chamfer Metrics

Consider a sampled refractive index field $\eta[x,y]$ (viewed as a positive image) and a set S of sources. A discrete GWDT finds at each pixel $p=[x,y]$ the smallest sum of values of η over all possible paths connecting p to the sources S. This can also be viewed as a procedure of finding paths of minimal 'cost' among nodes of a weighted graph or as discrete dynamic programming. Such a discrete GWDT (which is an approximation [25, 11] to the solution of the eikonal PDE $||\nabla U|| = \eta$) can be obtained via the following 2D recursive erosion [25]

$$\begin{aligned} U_i[x,y] = \min\{ & U_i[x-1,y] + a\eta[x,y], \\ & U_i[x,y-1] + a\eta[x,y], U_i[x-1,y-1] + b\eta[x,y], \\ & U_i[x+1,y-1] + b\eta[x,y], U_{i-1}[x,y]\} \end{aligned}$$

where $U_0[x,y]$ is set equal to 0 if $[x,y]$ belongs to the sources S or $+\infty$ otherwise. The propagation of the local distance steps (a,b) in the 3×3 chamfer mask starts at the wave sources and moves at speeds proportional to $1/\eta[x,y]$. The above recursive equation is run over the whole image in forward and backward order, iteratively $(i=1,2,3,...)$ until stability. At convergence, U_∞ is the GWDT of S. The above is a sequential implementation of the GWDT. There are also other faster implementations using queues [25, 14]. To improve the GWDT approximation to the eikonal's solution, one can optimize (a,b). Using a larger (e.g., 5×5) mask can further reduce the approximation error but at the cost of an even slower implementation. However, if larger masks are used with GWDTs, they may give erroneous results since the large masks can bridge over a thin line that separates two segmentation regions.

1.2.2 GWDT based on Surface Evolution

In this approach, at time $t=0$ the boundary of each source is modeled as a curve $\Gamma(0)$ which is then propagated with normal speed $c(x,y) \propto 1/\eta(x,y)$. The propagating curve $\Gamma(t)$ is embedded as the zero-level curve of a function $\Phi(x,y,t)$, where $\Phi(x,y,0) = \Phi_0(x,y)$ is the signed (positive in the curve interior) distance from $\Gamma(0)$. The surface Φ evolves according to the PDE (4), which is solved via the numerical algorithm of [17]. The value of the resulting GWDT at any pixel (x,y) of the image is the time it takes for the evolving curve to reach this pixel, i.e. the smallest t such that $\Phi(x,y,t) \geq 0$. This continuous approach to GWDT can achieve sub-pixel accuracy, as investigated in [9]. To reduce the computational complexity of the above surface evolution algorithm, we have developed a queue-based implementation of the fast marching level set methods of [23, 10] adapted to computing GWDTs in case of multiple sources where triple points develop at the collision of several wavefronts.

1.2.3 Gridless Halftoning via Eikonal PDE

Inspired by the use in [21] of the eikonal function's contour lines for visually perceiving an intensity image $I(x,y)$, the work in [25] and especially in [18] attempts to solve the PDE $||\nabla U(x,y)|| = \text{const} - I(x,y)$ and create a binary *gridless* halftone version of $I(x,y)$ as the union of the level curves of the eikonal function $U(x,y)$. The larger the intensity value $I(x,y)$, the smaller the local density of these contour lines in the vicinity of (x,y). This eikonal PDE approach to gridless halftoning is indeed very promising and can simulate various artistic effects, as shown in Fig. 2. There we also see that the surface evolution GWDT gives a smoother halftoning of the image than the GWDTs based on chamfer metrics.

1.2.4 Watershed Segmentation via Eikonal

A powerful morphological approach to *image segmentation* is the *watershed* [15, 26] which transforms an image $f(x,y)$ to the crest lines separating adjacent catchment basins that surround regional minima or other 'marker' sets of feature points. In [14, 16] it has been established that (in the continuous domain and assuming that the image is smooth and has isolated critical points) the continuous watershed is equivalent to finding a skeleton by influence zones with respect to a weighted distance function that uses the (one-point) regional minima of the image as sources and $||\nabla f||$ as the field of indices. (If other markers different than the minima are to be used as sources, then the homotopy of the function must be modified via morphological recosntruction to impose these markers as the only minima.)

In our work we solve the above eikonal PDE model of watershed segmentation of an image-related function f by finding a GWDT via surface evolution (4) where the speed is $\propto 1/||\nabla f||$. Further we compare the results of this new segmentation to the digital watershed algorithm via flooding [26] and to the eikonal approach solved via a discrete GWDT based on chamfer metrics [25, 14]. In all three approaches, robust features are extracted first as markers of the regions, and the original image I is transformed to another function f by smoothing via alternat-

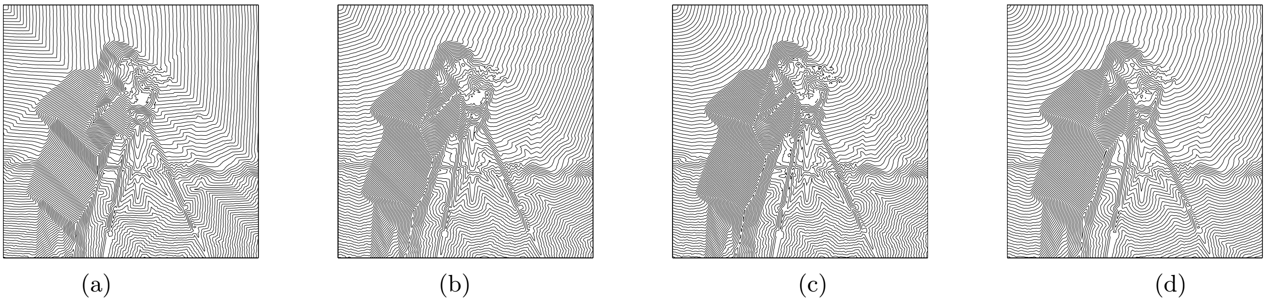

Figure 2. Gridless halftoning of the CAMERAMAN image from contour lines of GWDTs obtained via: (a) (1,1) chamfer metric; (b) optimal 3 × 3 chamfer metric; (c) optimal 5 × 5 chamfer metric; (d) curve evolution.

ing open/closing at multiple scales, taking the gradient magnitude of the filtered image, and changing (via morphological reconstruction) the homotopy of the gradient image so that its only minima occur at the markers. The segmentation is done on the final outcome f of the above processing.

In the standard digital watershed algorithm [26, 15], the flooding at each level is achieved by a planar distance propagation that uses the chess-board metric. This kind of distance propagation is non-isotropic and could give wrong results, particularly for images with large plateaus, as we found experimentally. Eikonal segmentation using chamfer-based GWDTs improves this situation a little but not entirely. In contrast, for images with large plateaus/regions, segmentation via the eikonal PDE and surface evolution GWDT gives results close to ideal. As Fig. 3 shows, compared on a test image that is difficult (because expanding wavefronts meet watershed lines at many angles ranging from from being perpendicular to almost parallel), our continuous segmentation approach based on the eikonal PDE and surface evolution outperforms the discrete segmentation results (using either the digital watershed flooding algorithm or chamfer-based GWDTs). However, some real images, as in Fig. 4, may not contain many plateaus or only large regions, in which cases the digital watershed flooding algorithm may give comparable results (or slightly better for thin elongated regions) than the eikonal PDE approach. Of course, the fact that the eikonal PDE segmentation may not detect part or all of a thin elongated region could be an advantage in applications where such thin regions are noisy or unreliable and hence should not be detected by a robust segmentation scheme.

2 PART II: MRL NEURAL NETS

The perceptron, *i.e.*, a linear combiner followed by a nonlinearity of the logistic type, is the standard node structure used in neural networks (NNs). However, it has been observed that logic operations, which are not well modeled by perceptrons, can be generated by some internal interactions in a neuron [33]. To better represent such internal properties, we propose the MRL-NNs, a general class of NNs where the combination of inputs in every node is formed by hybrid linear and nonlinear (of the morphological/rank type) operations. The fundamental processing unit of this class of systems is the MRL-filter [29], which is a linear combination between a morphological/rank filter and a linear FIR filter. The MRL-NNs have the unifying property that the characteristics of both multilayer perceptrons (MLPs) and morphological/rank neural networks (MRNNs) [30] are observed in the same system. An important special case of MRNNs is the class of min-max classifiers [34]. Next, we formulate a simple and systematic training procedure using ideas from the back-propagation algorithm [31] and robust techniques to circumvent the non-differentiability of rank functions. (Note that, since adaptive filters and NNs are closely related [28], we have investigated the adaptation of MRL filters and the training of MRL NNs under the same framework.) Our approach to train the morphological/rank nodes is a theoretically and numerically improved version of the method proposed in [32] to design morphological/rank filters. Finally, we apply the proposed design methodology in a problem of handwritten character recognition, and provide some experimental evidences showing that not only the MRL-NNs can generate similar or better results when compared with the classical MLPs, but also they usually require smaller processing times for training.

2.1 The Structure of MRL-NNs

In general terms, a (multilayer feed-forward) NN is a layered system composed by similar nodes, with some of them non-observable (hidden), where the node inputs in a given layer depend only on the node outputs from the preceding layer. Every node performs a generic composite operation, where an input to the node is first processed by some function $h(\cdot,\cdot)$ of the input and internal weights, and then transformed by an activation function $f(\cdot)$. The node structure is defined by the function h. In the case of MLPs, h is a linear combination. The activation function f is usually employed for rescaling purposes. A general NN is formally defined by the following set of recursive equations.

$$\begin{gathered} \underline{y}^{(l)} \equiv F(\underline{z}^{(l)}) = (f(z_1^{(l)}), f(z_2^{(l)}), \cdots, f(z_{N_l}^{(l)})) \ , \\ l = 1, 2, \cdots, L \ , \\ z_n^{(l)} \equiv h(\underline{y}^{(l-1)}, \underline{w}_n^{(l)}) \ , \ n = 1, 2, \cdots, N_l \ , \end{gathered} \tag{6}$$

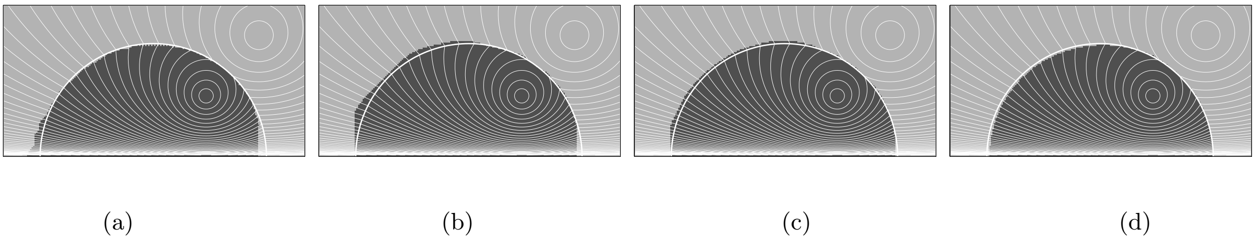

Figure 3. The Test image is the minimum of two potential functions. Its contour plot (thin bright curves) is superimposed on all segmentation results. Markers are the two source points of the potential functions. Segmentation results based on: (a) Digital watershed flooding algorithm. (b) (3×3) chamfer-based GWDT. (c) (5×5) chamfer-based GWDT. (d) Eikonal PDE and surface evolution GWDT. (The thick bright curve shows the correct segmentation.)

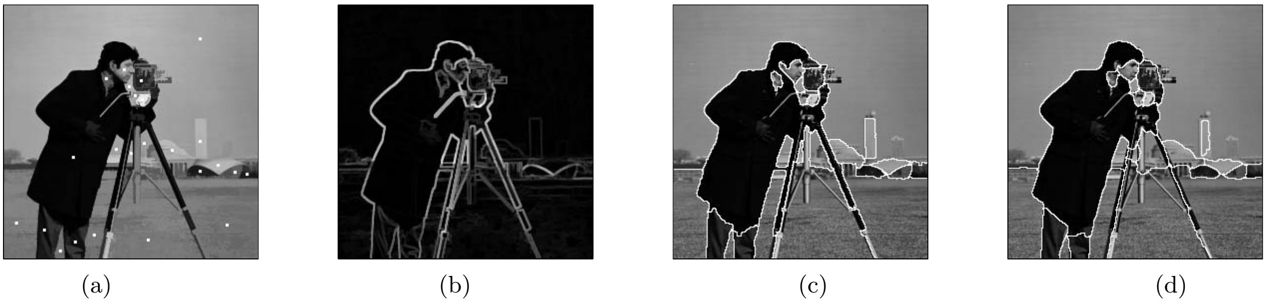

Figure 4. (a) Original Cameraman image with markers placed within desired regions. (b) Gradient magnitude of filtered image. (c) Segmentation result (superimposed on original) from digital watershed flooding algorithm. (d) Segmentation from eikonal PDE and surface evolution GWDT.

where l is the layer number, and N_l is the number of nodes in layer l. The weight vectors $\underline{w}_n^{(l)}$ represent the tuning parameters in the system. Besides this, the input and output of the system are

$$\begin{aligned} \underline{y}^{(0)} &= \underline{x} = (x_1, x_2, \cdots, x_{N_0}) \qquad \text{(input)} \\ \underline{y}^{(L)} &= \underline{y} = (y_1, y_2, \cdots, y_{N_L}) \qquad \text{(output)} \end{aligned} \tag{7}$$

The MRL-NN is the system defined by (6) and (7) such that

$$\begin{aligned} z_n^{(l)} &\equiv \lambda_n^{(l)} \alpha_n^{(l)} + (1 - \lambda_n^{(l)}) \beta_n^{(l)} \\ \alpha_n^{(l)} &= \mathcal{R}_{r_n^{(l)}} (\underline{y}^{(l-1)} + \underline{a}_n^{(l)}) \\ \beta_n^{(l)} &= \underline{y}^{(l-1)} \cdot (\underline{b}_n^{(l)})' + \tau_n^{(l)} \end{aligned} \tag{8}$$

where $\lambda_n^{(l)}$, $\tau_n^{(l)} \in \mathsf{IR}$; $\underline{a}_n^{(l)}$, $\underline{b}_n^{(l)} \in \mathsf{IR}^{N_{l-1}}$; and '′' denotes transposition.

$\mathcal{R}_r(\underline{t})$ is the r-th rank function of the vector $\underline{t} \in \mathsf{IR}^n$. It is evaluated by sorting the components of $\underline{t} = (t_1, t_2, \cdots, t_n)$ in decreasing order, $t_{(1)} \geq t_{(2)} \geq \cdots \geq t_{(n)}$, and picking the r-th element of the sorted list, *i.e.*, $\mathcal{R}_r(\underline{t}) = t_{(r)}$, $r = 1, 2, \cdots, n$.

Observe from (6) and (8) that the underlying function h is an MRL-filter [29] shifted by a threshold $(1 - \lambda_n^{(l)})\tau_n^{(l)}$. The variables $\tau_n^{(l)}$ are important when $\lambda_n^{(l)} = 0$. For every MRL-filter, the vector $\underline{b}_n^{(l)}$ corresponds to the coefficients of a linear FIR filter, and the vector $\underline{a}_n^{(l)}$ represents the coefficients of a morphological/rank filter. The variables $r_n^{(l)}$ and $\lambda_n^{(l)}$ are the rank and the mixing parameters, respectively. The resulting weight vector for every node is then defined by

$$\underline{w}_n^{(l)} \equiv (\underline{a}_n^{(l)},\ \rho_n^{(l)},\ \underline{b}_n^{(l)},\ \tau_n^{(l)},\ \lambda_n^{(l)}) \ , \tag{9}$$

where we use a real variable $\rho_n^{(l)}$ instead of an integer rank variable $r_n^{(l)}$ because we will need to evaluate derivatives during the design of MRL-NNs. The relation between $\rho_n^{(l)}$ and $z_n^{(l)}$ will be defined later via a differential equation, and $r_n^{(l)}$ is obtained from $\rho_n^{(l)}$ using [1]

$$r_n^{(l)} \equiv \left\lfloor N_{l-1} - \frac{N_{l-1} - 1}{1 + \exp(-\rho_n^{(l)})} + 0.5 \right\rfloor \ . \tag{10}$$

Two important special cases of MRL-NNs are obtained when f is the identity, called MRL-NN of type I (*e.g.*, MRNN: $\lambda_n^{(l)} = 1\ \forall n, l$), and when f is a nonlinearity of the logistic type, called MRL-NN of type II (*e.g.*, MLP: $\lambda_n^{(l)} = 0\ \forall n, l$).

2.2 Adaptive Design

Based on the LMS criterion and using ideas from the back-propagation algorithm, we propose a steepest descent method to optimally design general NNs, and then apply it to MRL-NNs. The design goal is to achieve a set of parameters $\underline{w}_n^{(l)}$, $n = 1, 2, \cdots, N_l$, $l = 1, 2, \cdots, L$, such that some cost function is minimized using a supervised procedure. Consider the training set

$$\{(\underline{x}(k), \underline{d}(k))\ ,\ \ k = 0, 1, \cdots, K - 1\}\ , \tag{11}$$

[1] $\lfloor \cdot \rfloor$ denotes the usual truncation operation, so that $\lfloor \cdot + 0.5 \rfloor$ is the usual rounding operation.

where $\underline{d}(k)$ is the desired system output to the training sample $\underline{x}(k)$. From (11) we generate the training sequence [2]

$$(\underline{x}([k]_{\bmod K}), \underline{d}([k]_{\bmod K})) \ , \ k \in \mathbb{Z} \ , \tag{12}$$

by making a periodic extension of (11). Every period of (12) is usually called an *epoch*. A general supervised training algorithm is of the form

$$\begin{gathered} \underline{w}_n^{(l)}(i+1) = \underline{w}_n^{(l)}(i) + \mu\, \underline{v}_n^{(l)}(i) \ , \ \mu > 0 \ , \\ n = 1, 2, \cdots, N_l \ ; \ l = 1, 2, \cdots, L \ , \end{gathered} \tag{13}$$

where the positive constant μ controls the tradeoff between stability and speed of convergence, $\underline{v}_n^{(l)} = -\nabla J$, and J is some cost function to be minimized. Let us define the error signal

$$\begin{gathered} \underline{e}(k) = (e_1(k), e_2(k), \cdots, e_{N_L}(k)) \equiv \\ \underline{d}([k]_{\bmod K}) - \underline{y}(k) \ , \end{gathered} \tag{14}$$

and the cost function

$$J(i) \equiv \frac{1}{M} \sum_{k=i-M+1}^{i} \xi(k) \ , \ 1 \leq M \leq K \ , \tag{15}$$

where

$$\xi(k) \equiv \sum_{n=1}^{N_L} e_n^2(k) \ . \tag{16}$$

Based on the steepest descent algorithm, it follows from (13) and (15) that

$$\underline{v}_n^{(l)}(i) = \frac{1}{M} \sum_{k=i-M+1}^{i} \underline{u}_n^{(l)}(k) \ , \tag{17}$$

where

$$\underline{u}_n^{(l)}(k) = -\ \frac{\partial \xi(k)}{\partial \underline{w}_n^{(l)}} \ . \tag{18}$$

If we define the matrices $W^{(l)}$, $V^{(l)}$ and $U^{(l)}$ by

$$\begin{gathered} W^{(l)} \equiv \begin{pmatrix} \underline{w}_1^{(l)} \\ \underline{w}_2^{(l)} \\ \vdots \\ \underline{w}_{N_l}^{(l)} \end{pmatrix} \ , \\ V^{(l)} \equiv \begin{pmatrix} \underline{v}_1^{(l)} \\ \underline{v}_2^{(l)} \\ \vdots \\ \underline{v}_{N_l}^{(l)} \end{pmatrix} \ , \ U^{(l)} \equiv \begin{pmatrix} \underline{u}_1^{(l)} \\ \underline{u}_2^{(l)} \\ \vdots \\ \underline{u}_{N_l}^{(l)} \end{pmatrix} \ , \end{gathered} \tag{19}$$

then the algorithm (13) can be written as

$$\begin{gathered} W^{(l)}(i+1) = W^{(l)}(i) + \mu\, V^{(l)}(i) \ , \\ V^{(l)}(i) = \frac{1}{M} \sum_{k=i-M+1}^{i} U^{(l)}(k) \ , \\ l = 1, 2, \cdots, L \end{gathered} \tag{20}$$

In this way, using the chain rule to evaluate $U^{(l)}(k)$, it can be shown that

$$U^{(l)}(k) = 2 \ \mathrm{diag}(\underline{\delta}^{(l)}(k)) \cdot \Gamma^{(l)}(k) \ , \tag{21}$$

where [3]

$$\underline{\delta}^{(l)}(k) = \underline{e}^{(l)}(k) \odot \dot{F}(\underline{z}^{(l)}(k)) \ , \tag{22}$$

$$\underline{e}^{(l)}(k) = \begin{cases} \underline{e}(k) & , \ l = L \\ \underline{\delta}^{(l+1)}(k) \cdot \Theta^{(l+1)}(k) & , \ \text{otherwise} \end{cases} \tag{23}$$

The matrices $\Gamma^{(l)}$ and $\Theta^{(l)}$ are defined by

$$\Gamma^{(l)} = \begin{pmatrix} \underline{\gamma}_1^{(l)} \\ \underline{\gamma}_2^{(l)} \\ \vdots \\ \underline{\gamma}_{N_l}^{(l)} \end{pmatrix} \ , \ \Theta^{(l)} = \begin{pmatrix} \underline{\theta}_1^{(l)} \\ \underline{\theta}_2^{(l)} \\ \vdots \\ \underline{\theta}_{N_l}^{(l)} \end{pmatrix} \ , \tag{24}$$

where

$$\underline{\theta}_n^{(l)} = \frac{\partial z_n^{(l)}}{\partial \underline{y}^{(l-1)}} \ , \ \underline{\gamma}_n^{(l)} = \frac{\partial z_n^{(l)}}{\partial \underline{w}_n^{(l)}} \ .$$

Based on this framework, the design of MRL-NNs can easily be derived. The difficulty is due to the non-differentiability of rank functions, but we can circumvent this problem by using pulse functions as follows [29].

$$\begin{gathered} \frac{\partial \alpha_n^{(l)}}{\partial \underline{y}^{(l-1)}} = \frac{\partial \alpha_n^{(l)}}{\partial \underline{a}_n^{(l)}} = \underline{c}_n^{(l)} \equiv \\ \frac{Q(\alpha_n^{(l)}\underline{1} - \underline{y}^{(l-1)} - \underline{a}_n^{(l)})}{Q(\alpha_n^{(l)}\underline{1} - \underline{y}^{(l-1)} - \underline{a}_n^{(l)}) \cdot \underline{1}'} \end{gathered} \tag{25}$$

$$\begin{gathered} \frac{\partial \alpha_n^{(l)}}{\partial \underline{\rho}_n^{(l)}} = s_n^{(l)} \equiv \\ 1 - \frac{1}{N_{l-1}} Q(\alpha_n^{(l)}\underline{1} - \underline{y}^{(l-1)} - \underline{a}_n^{(l)}) \cdot \underline{1}' \ . \end{gathered} \tag{26}$$

In (25) and (26), $Q(\underline{v}) \equiv (q(v_1), q(v_2), \cdots, q(v_n))$, where

$$q(v) \equiv \begin{cases} 1 & , \ \text{if } v = 0 \\ 0 & , \ \text{if } v \in \mathbb{R} \setminus \{0\} \end{cases} \tag{27}$$

and $\underline{1} = (1, 1, \cdots, 1)$. To avoid abrupt changes and achieve numerical robustness, we frequently replace the function $q(v)$ by smoothed impulses $q_\sigma(v)$, $\sigma \geq 0$, such as $\exp[-\frac{1}{2}(v/\sigma)^2]$ or $\mathrm{sech}^2(v/\sigma)$.

The remaining unknown is $\dot{F}(\cdot)$, that depends on the type of the MRL-NN in use. For the MRL-NN of type I, $F(\underline{z}^{(l)}) = \underline{z}^{(l)}$, so that $\dot{F}(\underline{z}^{(l)}) = \underline{1}$. For the MRL-NN of type II, we will use $f(z) = [1 + \exp(-\eta\, z)]^{-1}$, $\eta \geq 1$, whose derivative is $\dot{f}(z) = \eta\, f(z)[1 - f(z)]$, so that $\dot{F}(\underline{z}^{(l)}) = \eta \underline{y}^{(l)} \odot [\underline{1} - \underline{y}^{(l)}]$.

2.3 Application in OCR

Using the design framework discussed in the previous section, we now describe some experimental results in a problem of optical character recognition (OCR). Our

[2] $[k]_{\bmod K} \equiv k - K\, \lfloor k/K \rfloor$ denotes the index k modulo K.

[3] We denote $\dot{F}(\underline{z}) \equiv (\dot{f}(z_1), \dot{f}(z_2), \cdots, \dot{f}(z_n))$. The symbol '$\odot$' denotes an array (element-by-element) multiplication.

	FM $/\|E(\theta)\|/\|R(\theta)\|$					
	MRL5	MLP5	MRL10	MLP10	MRL20	MLP20
Training	11.8/13.2/10.5	9.9/8.8/11.1	**7.4**/6.9/7.8	7.4/7.0/7.7	8.4/7.8/9.0	**7.5**/6.8/8.2
Testing	18.7/22.4/15.0	18.4/19.9/16.9	**11.0**/13.1/8.9	11.1/10.9/11.4	17.4/24.6/10.2	**11.8**/12.8/10.9
Epoch	3	10	**62**	96	9	**88**

Table 1: Figure of merit / mean error rate / mean rejection rate corresponding to the optimal set of weights of best MRL-NNs vs. best MLPs for $\mu = 0.05$.

approach is to perform a comparative analysis of MRL-NNs versus MLPs, illustrating some of the characteristics of both systems. We show that the MRL-NNs are a good alternative to MLPs, usually providing equal or better performance with smaller training times.

To do so, we used a large database of handwritten characters provided by the National Institute of Standards and Technology (NIST) [27]. We selected a total of $K = 61,094$ samples of handwritten digits to form our data set. In our simulations, we normalized the feature vectors (64 dimensional Karhunen-Loève transforms) so that each $\underline{x}(k) \in [0,1]$. The data set was split such that 45,000 digits were used for training and the remaining 16,094 digits were used for testing. The first 15,000 elements of the training set were used as a validation set during the training process. The training sequence was ordered such that one instance of every digit is presented to the system in each iteration.

After making several tests, we have set a group 12 experiments with 3 different network topologies: 64-N-10 MRL-NNs and MLPs, $N = 5, 10, 20$. This notation indicates a system with 64 inputs, N hidden nodes, and 10 outputs. Two different step sizes were tested: $\mu = 0.005, 0.05$. Every experiment was repeated 5 times with different random initial conditions, and the best result is reported here. Among many possible ways to initialize the systems, and after performing various tests, we initialized the weights randomly in the ranges: $\underline{a}_n^{(l)}$: $[-0.1, 0.1]$, $r_n^{(l)}$: $[1, N_{l-1}]$, $b_n^{(l)}$: $[-1/\sqrt{N_{l-1}}, 1/\sqrt{N_{l-1}}]$, $\tau_n^{(l)}$: $[-0.1, 0.1]$, $\lambda_n^{(l)}$: $[0.4, 0.6]$. Further, in order to estimate gradients, we smoothed impulses with $q_\sigma(v) = \exp[-\frac{1}{2}(v/\sigma)^2]$, $\sigma = 0.05$. Due to the size of the training set, we have used the proposed training algorithm with $M = 1$ only. We have tested the case $M > 1$ with a small subset of the training set, but no signanificant improvements were observed. Both MRL-NNs and MLPs were defined using a sigmoid activation function with $\eta = 1$ (MRL-NNs of type II). As usual, the desired system output $\underline{d} = (d_0, d_1, \cdots, d_9)$ was defined by

$$d_n = \begin{cases} 1 & , \underline{x} \leftrightarrow \text{digit n} \\ 0 & , \text{otherwise} \end{cases} \tag{28}$$

In the attempt to compare different systems, a figure of merit (FM) was defined as follows

$$FM(t) = \frac{1}{2}\left(\|E(\theta)\| + \|R(\theta)\|\right) \tag{29}$$

where $\theta \in [0,1]$ is the confidence threshold; t is the epoch; E is the error rate (%), computed, for a given θ, as the ratio of the number of misclassified digits over the number of digits that were not rejected during the classification (in a percentage basis); R is the rejection rate (%), computed, for a given θ, as the ratio of the number of rejected digits over the total number of elements in the set under consideration (also in a percentage basis); and

$$\|E(\theta)\| = \frac{1}{10}\sum_{i=0}^{9} E(\frac{i}{10}) \;\; , \;\; \|R(\theta)\| = \frac{1}{10}\sum_{i=0}^{9} R(\frac{i}{10})$$

The training process tends to decrease the figure of merit, and good performance corresponds to small values of FM. A given classification is rejected if the desired n-th output is $d_n = 1$ but the actual n-th output has the property $\max\{\underline{y}\} = y_n < \theta$, where $\underline{y}$ is the system output. An error (misclassification) is obtained when $d_n = 1$ but $\max\{\underline{y}\} \neq y_n$. The error rate is computed excluding the rejected digits.

Using our proposed training algorithm with all the above considerations, we observed that, either for a step size $\mu = 0.005$ or $\mu = 0.05$, the MRL-NNs required a smaller number of iterations than MLPs, and provided similar performances (FMs). Computing the figures of merit of MLPs with equal number of iterations of MRL-NNs, we usually observed better performances of MRL-NNs. Table 1 summarizes some of the results. The best training performance was obtained with a 64-10-10 MRL-NN (MRL10, FM=7.4%). Similar results were obtained with a 64-10-10 MLP (MLP10, FM=7.4%) and a 64-20-10 MLP (MLP20, FM=7.5%), but with a larger number of iterations. These best results were all obtained with $\mu = 0.05$.

References

References for Part I

[1] L. Alvarez, F. Guichard, P.L. Lions, and J.M. Morel, "Axioms and Fundamental Equations of Image Processing", *Archiv. Rat. Mech.*, vol. 123 (3), pp. 199–257, 1993. Also, *C. R. Acad. Sci. Paris*, pp. 265-268, t.315, Serie I, 1992.

[2] A. Arehart, L. Vincent and B. Kimia, "Mathematical Morphology: The Hamilton-Jacobi Connection", in *Proc. ICCV-93*, pp. 215–219, 1993.

[3] G. Borgefors, "Distance Transformations in Digital Images", *Comp. Vision, Graphics, Image Process.*, 34, pp. 344–371, 1986.

[4] M. Born and E. Wolf, *Principles of Optics*, Pergamon Press, Oxford, England, 1959 (1987 edition).

[5] R. Brockett and P. Maragos, "Evolution Equations for Continuous-Scale Morphological Filtering", *IEEE Trans. Signal Processing*, vol. 42, pp. 3377-3386, Dec. 1994. Also, in *Proc. ICASSP-92*, San Francisco, 1992.

[6] M. A. Butt and P. Maragos. "Optimal Design of Chamfer Distance Transforms", *IEEE Trans. Image Processing*, in press.

[7] B.K.P. Horn, *Robot Vision*, MIT Press, Cambridge, MA, 1986.

[8] B. Kimia, A. Tannenbaum, and S. Zucker, "Toward a Computational Theory of Shape: An Overview", *Proc. ECCV-90*, France, April 1990.

[9] R. Kimmel, N. Kiryati, and A. M. Bruckstein, "Sub-Pixel Distance Maps and Weighted Distance Transforms", *J. Math. Imaging and Vision*, 6, pp. 223-233, 1996.

[10] R. Malladi, J. A. Sethian, and B. C. Vemuri, "A Fast Level Set Based Algorithm for Topology-Independent Shape Modeling", *J. Math. Imaging and Vision*, 6, pp. 269-289, 1996.

[11] P. Maragos, "Differential Morphology and Image Processing" *IEEE Trans. Image Processing*, vol. 78, pp. 922–937, June 1996.

[12] P. Maragos and R. W. Schafer, "Morphological Systems for Multidimensional Signal Processing", *Proc. IEEE*, vol. 78, pp. 690-710, Apr. 1990.

[13] J. Mattioli, "Differential Relations of Morphological Operators", *Proc. 1st Int'l Workshop on Math. Morphology and its Application to Signal Processing*, Barcelona, Spain, May 1993.

[14] F. Meyer, "Topographic Distance and Watershed Lines", *Signal Processing*, vol. 38, pp. 113-125, July 1994.

[15] F. Meyer and S. Beucher, "Morphological Segmentation", *J. Visual Commun. Image Representation*, 1(1):21–45, 1990.

[16] L. Najman and M. Schmitt, "Watershed of a Continuous Function", *Signal Processing*, vol. 38, pp. 99-112, July 1994.

[17] S. Osher and J. Sethian, "Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations", *J. Comput. Physics*, 79, pp. 12–49, 1988.

[18] Y. Pnueli and A. M. Bruckstein, "Digi$_D$ürer - a digital engraving system", *The Visual Computer*, 10, pp. 277–292, 1994.

[19] E. Rouy and A. Tourin, "A Viscocity Solutions Approach to Shape from Shading", *SIAM J. Numer. Anal.*, vol. 29 (3), pp. 867-884, June 1992.

[20] G. Sapiro, R. Kimmel, D. Shaked, B. Kimia, and A. Bruckstein, "Implementing Continuous-scale Morphology via Curve Evolution", *Pattern Recognition*, 26(9), pp. 1363–1372, 1993.

[21] M. Schröder, "The Eikonal Equation", *Math. Intelligencer*, 1, pp. 36–37, 1983.

[22] J. Serra, *Image Analysis and Mathematical Morphology*, Acad. Press, NY, 1982.

[23] J. A. Sethian, *Level Set Methods*, Cambridge Univ. Press, 1996.

[24] R. van der Boomgaard and A. Smeulders, "The Morphological Structure of Images: The Differential Equations of Morphological Scale-Space", *IEEE Trans. Pattern Anal. Mach. Intellig.*, vol. 16, pp.1101-1113, Nov. 1994. Also, Ph.D. Thesis, Univ. of Amsterdam, 1992.

[25] P. Verbeek and B. Verwer, "Shading from shape, the eikonal equation solved by grey-weighted distance transform", *Pattern Recogn. Lett.*, 11:618–690, 1990.

[26] L. Vincent and P. Soille, "Watershed In Digital Spaces: An Efficient Algorithm Based On Immersion Simulations", *IEEE Trans. PAMI*, vol. 13, pp. 583–598, June 1991.

References for Part II

[27] M.D. Garris, J.L. Blue, G.T. Candela, D.L. Dimmick, J. Geist, P.J. Grother, S.A. Janet, and C.L. Wilson, "NIST form-based handprint recognition system", *Tech. Rep. NISTIR 5469*, Nat'l Inst. Standards & Technology, July 1994.

[28] S. Marcos, O. Macchi, C. Vignat, G. Dreyfus, L. Personnaz, and P. Roussel-Ragot, "A unified framework for gradient algorithms used for filter adaptation and neural network training", *Int'l J. Circuit Theory & Applications*, 20:159–200, 1992.

[29] L. F.C. Pessoa and P. Maragos, "MRL-Filters: A general class of nonlinear systems and their optimal design for image processing," *IEEE Trans. Image Processing*, July 1998.

[30] L. F.C. Pessoa and P. Maragos, "Morphological/rank neural networks and their adaptive optimal design for image processing", *Proc. ICASSP-96*, vol.6, pp.3399–3402, May 1996.

[31] D.E. Rumelhart, G.E. Hinton, and R.J. Willians, " Learning Internal Representations by Error Propagation," *Parallel Distributed Processing, Vol.1*, D.E. Rumelhart and J.L. McClelland, eds., pp.318–362, MIT Press, Cambridge, MA, 1986.

[32] P. Salembier, "Adaptive rank order based filters," *Signal Processing*, 27:1–25, Apr. 1992.

[33] G.M. Shepherd and R.K. Brayton, "Logic operations are properties of computer-simulated interactions between excitable dendritic spines," *Neuroscience*, 21(1):151–165, 1987.

[34] P.-F. Yang and P. Maragos, "Min-max classifiers: Learnability, design and application," *Pattern Recognition*, 28(6):879–899, 1995.