The Viterbi Optimal Runlength-Constrained Approximation Nonlinear Filter

Nicholas D. Sidiropoulos, *Member, IEEE*

***Abstract*— Simple nonlinear filters are often used to enforce "hard" syntactic constraints while remaining close to the observation data, e.g., in the binary case, it is common practice to employ iterations of a suitable median, or a one-pass recursive median, openclose, or closopen filter to impose a minimum symbol run-length constraint while remaining "faithful" to the observation. Unfortunately, these filters are—in general—suboptimal. Motivated by this observation, we pose the following optimization: Given a finite-alphabet sequence of finite extent $\boldsymbol{y} = \{y(n)\}_{n=0}^{N-1}$, find a sequence $\hat{\boldsymbol{x}} = \{\hat{x}(n)\}_{n=0}^{N-1}$ that minimizes $d(\boldsymbol{x}, \boldsymbol{y}) = \Sigma_{n=0}^{N-1} \; d_n(y(n), x(n))$ subject to the following: $\boldsymbol{x}$ is piecewise constant of plateau run-length $\geq M$. We show how a suitable reformulation of the problem naturally leads to a simple and efficient Viterbi-type optimal algorithmic solution. We call the resulting nonlinear input–output operator the *Viterbi optimal runlength-constrained approximation (VORCA)* filter. The method can be easily generalized to handle a variety of local syntactic constraints. The VORCA is optimal, computationally efficient, and possesses several desirable properties (e.g., idempotence); we therefore propose it as an attractive alternative to standard median, stack, and morphological filtering. We also discuss some applications.**

I. Introduction

The median filter[1] [1]–[7] is arguably one of the most frequently used tools in nonlinear signal processing. It has several desirable properties, and considerable effort has been spent in its analysis [7]. Due to its simplicity, it affords very efficient implementation. It also has two important disadvantages, namely, (as we will see) it is not optimal even under statistical scenarios which are well adapted to its purported strengths, and it is not idempotent, meaning that if and when it converges (it does in the case of ordered data of finite extent), it only does so after a number of passes [7].

The main textbook argument behind median filtering is that it preserves edges while effectively removing impulsive noise and outliers, i.e., it is a *robust* and *locally optimal* estimator of edge location. In addition, it is a *self-dual* operator [1] (more will be said about self-duality later). In this setting, the analysis is based on the concept of an *ideal edge,* which is really thought of as *a jump discontinuity which exhibits some degree of consistency*, i.e., a jump discontinuity in between two locally flat regions of sufficient breadth (i.e., plateaus of length greater than or equal to some constant). It is assumed that the ideal edge data is corrupted by independent identically distributed (i.i.d.) two-sided impulsive noise, and the purpose of applying the filter is to recover the true data by eliminating outliers (noise impulses, which are locally inconsistent with the data). Indeed, the median filter does a fairly good job in this setting, one that is remarkably better than that of a moving average. Unfortunately, local optimality of the median[2] does not suffice to guarantee global optimality of the solution (i.e., filtered data). Morphological filters [1] can also be applied, and they are idempotent by definition, but (as we will show) similar remarks hold regarding their optimality in this setting.

A. Constrained Optimization

Suppose we are given a set of ordered data (e.g., a function of compact support, or a sequence of finite extent), f, and we are interested in approximating, representing, or replacing f by a compact descriptor (i.e., reduced complexity set of data), g, which is optimal in some sense. Quite often, g is also required to satisfy certain criteria of local regularity (e.g., continuity, smoothness), and/or structural (syntactic) constraints.

This kind of problem often appears in a number of disciplines, including optimal filtering of time series, source coding and vector quantization, curve fitting, edge detection, and polygonal approximation of planar shape boundaries. There exists an immense body of literature which deals with these subjects. Some approaches are heuristic, while others are optimal. Optimal approaches typically start with a formal statement of the problem. This usually entails setting up a suitable optimization, which involves the specification of two fundamental components, namely, a *distortion measure* $d(f, g)$, which formalizes and quantifies the notion of similarity, i.e., provides a measure of how "close" g is to f, and a *complexity-conformity measure* $\lambda(g)$, which measures two things: the complexity of the resulting approximation and conformity to any prespecified regularity and/or structural constraints. In general, any prespecified constraints of the

Manuscript received April 10, 1995; revised September 1, 1995. This work was supported in part by core funds from the NSF ERC program, made available through the Communications and Signal Processing Group of the Institute for Systems Research of the University of Maryland; and industry, through Martin-Marietta Chair in Systems Engineering funds. The associate editor coordinating the review of this paper and approving it for publication was Dr. Thomas F. Quatieri, Jr.

The author is with the Institute for Systems Research, University of Maryland, College Park, MD 20742 USA.

Publisher Item Identifier S 1053-587X(96)02399-7.

[1] In [1], a *filter* is defined as an operator which is increasing and idempotent (these properties are explained later in this paper). In this sense, the median is not a filter, since it is increasing but not idempotent. However, it is standard engineering practice to call it a filter. We therefore adhere to this practice and reserve the term *morphological filter* for those operators which are increasing and idempotent, i.e., filters in the sense of [1].

[2] The median *locally* minimizes an l_1-type norm, i.e., mean absolute error [7]. It can also be viewed as minimizing a two-term composite cost function [8] in a local sense.

latter type can be incorporated in $\lambda(g)$ by setting $\lambda(g) = \infty$ whenever g is not compatible with the given constraints.

Within this general framework, there exist essentially two meaningful ways to pose the approximation of f as an optimization problem. These are

$$\textbf{minimize } d(f,g), \quad \textbf{subject to } \lambda(g) \leq t < \infty \tag{1}$$

or

$$\textbf{minimize } \lambda(g), \quad \textbf{subject to } d(f,g) \leq \epsilon. \tag{2}$$

Of course, there exists great freedom in choosing $d(\cdot,\cdot)$, and $\lambda(\cdot)$. Depending on the particular choice of these two measures, the optimization may, or may not have a solution, which may, or may not be unique, stable, meaningful, and/or computationally tractable. Typical choices for $d(\cdot,\cdot)$ include l_1, l_2, and l_∞ distance metrics. A typical constraint might be that g is piecewise linear and continuous, while complexity might be measured by the total number of line segments required to construct g.

It occasionally happens that a particular optimization admits an efficient recursive solution; in this case, the underlying synergy can often be attributed to the principle of optimality, a particularly pervasive "ground truth" of dynamic programming [9]–[11].

B. Organization

The rest of this paper is structured as follows. In Section I-C, we present a bare bones formal statement of the problem. Previous related work is reviewed in detail in Section II. Our solution and a simple example are presented in Section III. Several fundamental properties of the resulting optimal input–output operator are investigated in Section IV. The analysis adopts a nonlinear filtering viewpoint and focuses on general characterization principles. A discussion on implementation complexity is also included. A complete simulation experiment is presented in Section V, applications are discussed in Section VI, and conclusions are drawn in Section VII.

C. A Bare Bones Statement of the Problem

Suppose $y(n) \in \mathcal{A}, n = 0, 1, \cdots, N-1$, and $|\mathcal{A}| < \infty$. Let P_M^N denote the set of all sequences of N elements of $\mathcal{A}$ which are piecewise constant of plateau (run) length $\geq M$. Consider the following constrained optimization

$$\textbf{minimize} \quad \sum_{n=0}^{N-1} d_n(y(n), x(n)) \tag{3}$$

$$\textbf{subject to} \quad \boldsymbol{x} = \{x(n)\}_{n=0}^{N-1} \in P_M^N. \tag{4}$$

This particular optimization arose during the course of our investigations in nonlinear filtering.

II. Background and Related Work

There exist numerous references which are related—in various ways and degrees—to our present line of work. What follows is a (long) partial list. We highlight those contributions which are closest, in spirit, to our work. Additional references can be found in Section VI. We note that our particular formulation does not fit in any of the existing paradigms.

The piecewise-constant sequence approximation problem is a proper special case of the problem of piecewise-linear curve fitting. This latter problem (which in turn is a special case of the problem of piecewise polynomial functional approximation) has attracted a considerable amount of interest for more than three decades, triggered in part by a widely held belief in the importance of this line of work in shape recognition.

In 1961, Stone [12] considered piecewise-linear curve fitting as a formal optimization problem. The objective was to minimize the squared approximation error subject to a constraint on the number of linear segments. Bellman [13] soon followed with a solution based on his principle of optimality of dynamic programming [9]–[11]. Gluss [14], [15]–[17] expanded on the original idea of Bellman. Bellman *et al.* further extended these ideas in [18]. Cox [19] discussed a similar solution in his 1971 paper. The aforementioned authors consider a least-squares constrained complexity formulation (i.e., they fix the number of segments in the approximation and minimize squared error), and the common denominator is precisely the principle of optimality.

There exist two similarities, as well as two significant differences, between our formulation and Bellman's formulation. Both attempt to minimize distortion subject to a complexity-conformity constraint (i.e., they are type-(1) optimizations). Both can be solved by invoking the principle of optimality. However, our constraint is on the *minimum length* of segments, whereas Bellman's constraint is on the *maximum number* of segments. Observe that, for finite data, a constraint of the former type implies a constraint of the latter type, but the reverse is not true. The second noteworthy difference is that our distortion measure can be inhomogeneous, and in fact arbitrary, as long as it is the sum of individual per-letter costs.[3]

In 1986, Dunham [20] solved a related type-(2) optimization by applying the principle of optimality. His program seeks to minimize complexity (i.e., number of segments) subject to an l_∞ error bound. Kurozumi and Davis [21] considered a similar problem.

There exists a considerable amount of additional literature on the subject of piecewise-linear curve fitting. This includes the work of Montanari [22], who considered minimal length polygonal approximations, Ramer [23], and Duda and Hart [24], who considered successive refinement under an error-bound constraint, Slansky *et al.* [25], [26], Tomek [27], Rosenfeld and Weszka [28], Narayanan *et al.* [29], Pavlidis *et al.* [30]–[33], Vandewalle [34], Williams [35], [36], Badi'i and Peikari [37], Wu [38], who employed a statistical model, Bezdek and Anderson [39], Imai [40], Baruch [41], Teh and Chin [42], and Fahn *et al.* [43], among others. These references take on several variations of the problem, e.g., breakpoint continuity/discontinuity etc. Some approaches are *ad hoc*, while others attempt to compute a nearly-optimal solution.

[3] Note that this "sum" could be interpreted in a more liberal sense, e.g., our method can also accommodate a minimax problem formulation, i.e., seeking to minimize the supremum of pairwise per-letter costs, subject to a hard syntactic constraint.

Additional related material can be found in the literature on regularization and edge detection (e.g., [44], [45]), and deformable contours, snakes, and related themes [46]–[48]. In a sense, the determination of optimal deformable contour dynamics is an "inverse" of our problem. The former *starts* with a "simple" user-supplied constrained approximation of a curve, then attempts to match this initial approximation to the data by deforming it under the influence of some appropriately chosen dynamics. The goal is to minimize a suitable energy functional. One particularly interesting reference in this area is the work of Amini *et al.* [49], in which the authors address dynamic programming solutions of some variational problems in early vision. The authors point out that when faced with so-called "hard" nondifferentiable constraints on the solution, Lagrangian-based methods, as well as regularization-based methods, typically fail to produce an answer. Lagrangian methods require additive-differentiable constraints. Both methods can "bias" the solution toward satisfying the constraints, but they cannot strictly enforce hard nondifferentiable constraints. On the other hand, dynamic programming can easily accommodate hard nondifferentiable constraints, and, in fact, *use* these constraints to reduce computational complexity. The drawback is that it does not provide a closed-form analytical solution, but this is something we can often live with. In the aforementioned reference the authors consider a particular problem which, when translated into our setting, reads as follows: minimize distortion, under the constraint that 1) the number of segments is fixed and equal to some predetermined constant (this is Bellman's constraint), and 2) the length of the plateaus is bounded below by some predetermined constant (which is our constraint). Thus they consider a significantly more constrained optimization. In contrast, we would like our method to *determine* the optimal number of segments *automatically, and on the fly*, by considering whether it pays to introduce additional segments as it parses the data.

Konstantinides and Natarajan [50] consider a type-(2) optimization, with complexity measured in terms of number of segments, present an $O(N)$ algorithm that solves it, and provide a real-time custom processor implementation. Papakonstantinou *et al.* [51] have recently pointed out that the solution of a particular type-(2) optimization (with complexity measured in terms of number of segments) is highly nonunique. They subsequently proposed further refinement of the solution by the method of least squares, i.e., among the set of all optimal solutions of (2), select the one which minimizes squared error. The overall optimization is a hybrid two-step process, combining elements of both type-(2) and type-(1) optimization. Their solution is based on a tree pruning approach.

Mumford and Shah have posed [52] and investigated [53] a general variational formulation of image segmentation. Their formalism is ambitious and powerful; it attempts to tackle the general problem of edge detection and low-level vision. Blake and Zisserman [54] have written a book on how to solve the Mumford–Shah optimization, based on the so-called *graduated nonconvexity* (GNC) algorithm. Morel and Solimini [55] have written a recent book on the mathematical analysis of the Mumford–Shah model, in which they also argue that the Mumford–Shah formalism unifies many seemingly disparate variational approaches to image segmentation. It is rather interesting to note that our particular optimization does not fit in the general Mumford–Shah formulation.

III. Solution

We show how a suitable reformulation of the problem naturally leads to a simple and efficient Viterbi-type optimal algorithmic solution.

Definition 1: Given any sequence $\boldsymbol{x} = \{x(n)\}_{n=0}^{N-1}, x(n) \in \mathcal{A}, n = 0, 1, \cdots, N-1$, define its associated *state sequence*, $\boldsymbol{s_x} = \{[x(n), l_{\boldsymbol{x}}(n)]^T\}_{n=-1}^{N-1}$, where $[x(-1), l_{\boldsymbol{x}}(-1)]^T = [\phi, M]^T, \phi \notin \mathcal{A}$, and, for $n = -1, \cdots, N-2$

$$l_{\boldsymbol{x}}(n+1) = \begin{cases} \min\{l_{\boldsymbol{x}}(n)+1, M\}, & x(n+1) = x(n) \\ 1, & \text{otherwise.} \end{cases}$$

$[x(n), l_{\boldsymbol{x}}(n)]^T$ is the state at time n, and, for $n = 0, 1, \cdots, N-1$, it assumes values in $\mathcal{A} \times \{1, \cdots, M\}$.

Clearly, we can equivalently pose the optimization (3), (4) in terms of the associated state sequence.

Definition 2: A subsequence of state variables $\{[x(n), l_{\boldsymbol{x}}(n)]^T\}_{n=-1}^{\nu}, \nu \leq N-1$, is *admissible* [with respect to constraint (4)] if and only if there exists a suffix string of state variables, $\{[x(n), l_{\boldsymbol{x}}(n)]^T\}_{n=\nu+1}^{N-1}$, such that $\{[x(n), l_{\boldsymbol{x}}(n)]^T\}_{n=-1}^{\nu}$ followed by $\{[x(n), l_{\boldsymbol{x}}(n)]^T\}_{n=\nu+1}^{N-1}$ is the associated state sequence of some sequence in P_M^N.

Let $\hat{\boldsymbol{x}} = \{\hat{x}(n)\}_{n=0}^{N-1}$ be *a* solution (one always exists, although it may not necessarily be unique) of (3), (4), and $\{[\hat{x}(n), l_{\hat{\boldsymbol{x}}}(n)]^T\}_{n=-1}^{N-1}$ be its associated state sequence. Clearly, $\{[\hat{x}(n), l_{\hat{\boldsymbol{x}}}(n)]^T\}_{n=-1}^{N-1}$ is admissible, and so is any subsequence $\{[\hat{x}(n), l_{\hat{\boldsymbol{x}}}(n)]^T\}_{n=-1}^{\nu}, \nu \leq N-1$. The following is a key observation.

Claim 1: Optimality of $\{[\hat{x}(n), l_{\hat{\boldsymbol{x}}}(n)]^T\}_{n=-1}^{N-1}$ implies optimality of $\{[\hat{x}(n), l_{\hat{\boldsymbol{x}}}(n)]^T\}_{n=-1}^{\nu}, \nu \leq N-1$, among all admissible subsequences of the same length which lead to the same state at time ν, i.e., all admissible $\{[\tilde{x}(n), l_{\tilde{\boldsymbol{x}}}(n)]^T\}_{n=-1}^{\nu}$ satisfying $[\tilde{x}(\nu), l_{\tilde{\boldsymbol{x}}}(\nu)]^T = [\hat{x}(\nu), l_{\hat{\boldsymbol{x}}}(\nu)]^T$

Proof: The argument goes as follows. Suppose that $\{[\tilde{x}(n), l_{\tilde{\boldsymbol{x}}}(n)]^T\}_{n=-1}^{\nu}$ is an admissible subsequence of states satisfying $[\tilde{x}(\nu), l_{\tilde{\boldsymbol{x}}}(\nu)]^T = [\hat{x}(\nu), l_{\hat{\boldsymbol{x}}}(\nu)]^T$. It is easy to see that $\{[\tilde{x}(n), l_{\tilde{\boldsymbol{x}}}(n)]^T\}_{n=-1}^{\nu}$ followed by $\{[\hat{x}(n), l_{\hat{\boldsymbol{x}}}(n)]^T\}_{n=\nu+1}^{N-1}$, is also admissible. The key point is that any suffix string of state variables which makes $\{[\hat{x}(n), l_{\hat{\boldsymbol{x}}}(n)]^T\}_{n=-1}^{\nu}$ admissible, will also make $\{[\tilde{x}(n), l_{\tilde{\boldsymbol{x}}}(n)]^T\}_{n=-1}^{\nu}$ admissible. If $\{[\tilde{x}(n), l_{\tilde{\boldsymbol{x}}}(n)]^T\}_{n=-1}^{\nu}$ has a smaller cost (distortion) than $\{[\hat{x}(n), l_{\hat{\boldsymbol{x}}}(n)]^T\}_{n=-1}^{\nu}$, then, by virtue of the fact that the cost is a sum of per-letter costs, $\{[\tilde{x}(n), l_{\tilde{\boldsymbol{x}}}(n)]^T\}_{n=-1}^{\nu}$ followed by $\{[\hat{x}(n), l_{\hat{\boldsymbol{x}}}(n)]^T\}_{n=\nu+1}^{N-1}$ will have a smaller cost than $\{[\hat{x}(n), l_{\hat{\boldsymbol{x}}}(n)]^T\}_{n=-1}^{N-1}$, and this violates the optimality of the latter.

This is a particular instance of the principle of optimality. The following is an important corollary.

Corollary 1: An optimal admissible path to any given state at time $n+1$ must be an admissible one-step continuation of an optimal admissible path to *some* state at time n.

This corollary leads to an efficient Viterbi-type algorithmic implementation of the optimal filter [56]–[58]. It remains to specify the costs associated with one-step state transitions in a way that forces one-step optimality and admissibility. This is easy. Let $c(\boldsymbol{s_x}(n) \rightarrow \boldsymbol{s_x}(n+1))$ denote the cost of a one-step state transition, and let $\vee$ and $\wedge$ denote logical OR and AND, respectively. Recall that, in so far as the hard constraint is concerned, every run of length $\geq M$ is acceptable, and, in order to save on the number of required states, every run length above M can be mapped back to M. Then

$$\textbf{\textit{if}}$$

$$[((l_{\boldsymbol{x}}(n) < M) \vee (n \geq N-M))\wedge$$

$$((x(n+1) \neq x(n)) \vee (l_{\boldsymbol{x}}(n+1) \neq \min\{l_{\boldsymbol{x}}(n)+1, M\}))]$$

/* current run incomplete, or not enough time to begin and subsequently complete a new run, and we try to do anything other than simply continue the current run */

$$\vee$$

$$[(l_{\boldsymbol{x}}(n) = M) \wedge (x(n+1) = x(n)) \wedge (l_{\boldsymbol{x}}(n+1) \neq M)]$$

/* current run is complete, and we decide to continue it, yet the length variable does not remain at M */

$$\vee$$

$$[(l_{\boldsymbol{x}}(n) = M) \wedge (x(n+1) \neq x(n)) \wedge (l_{\boldsymbol{x}}(n+1) \neq 1)]$$

/* current run is complete, and we decide to switch to another value, yet the length variable is not reset back to unity */

$$\begin{aligned} &\textbf{then } c([x(n), l_{\boldsymbol{x}}(n)]^T \rightarrow [x(n+1), l_{\boldsymbol{x}}(n+1)]^T) = \infty \\ &\textbf{else } c([x(n), l_{\boldsymbol{x}}(n)]^T \rightarrow [x(n+1), l_{\boldsymbol{x}}(n+1)]^T) \\ &\qquad = d_{n+1}(y(n+1), x(n+1)) \end{aligned} \tag{5}$$

will do it. A formal proof can be easily constructed and is hereby omitted. The possibility of having multiple solutions (minimizers) implies that the above specification of costs associated with one-step state transitions *does not* uniquely specify an input–output operator; a tie-breaking strategy is also required. Since this does not affect filter performance, we assume that one such strategy is given and call the resulting nonlinear input–output operator the *Viterbi optimal runlength-constrained approximation* (VORCA) filter.

Other types of local syntactic constraints can easily fit in this paradigm. Suppose we are interested in a piecewise linear solution of constraint length M (i.e., a piecewise linear optimal approximation of segment length $\geq M$). We may further augment the state to include the *discrete slope* of the "current" segment, i.e., set $\boldsymbol{s_x}(n) = [x(n), l_{\boldsymbol{x}}(n), t_{\boldsymbol{x}}(n)]^T$, where $t_{\boldsymbol{x}}(n)$ is the discrete slope state variable. The specification of corresponding one-step state transition costs in a way that enforces one-step optimality and admissibility is relatively straightforward, and it will not be further pursued here.

One may handle the most general type of *local* syntactic constraints, by augmenting the state to include $M-1$ "past" values. However, this corresponds to an exponential (in M) expansion of the Viterbi trellis state space, which quickly exhausts computational resources for moderate values of M. Most problems of practical interest do not require a full state expansion, thus being amenable to efficient Viterbi-type algorithmic solutions. We refer to an element of the resulting class of nonlinear input–output mappings as a *Viterbi optimal syntactic approximation* (VOSA) Filter.

A. A Simple Example

A simple example is presented in Fig. 1, which depicts the VORCA trellis for the case $d_n(y(n), x(n)) = |y(n) - x(n)|$, $\forall n \in \{0, 1, \cdots, N-1\}, N = 12, M = 4, \mathcal{A} = \{0, 1\}$, and input $\{y(n)\}_{n=0}^{11} = \{1,1,1,1,0,0,0,1,1,1,1,1\}$. The state space consists of eight possible states in $\{0,1\} \times \{1,2,3,4\}$. Solid lines represent transitions which involve unit cost, whereas dashed lines represent transitions which involve zero cost. Absence of a line indicates infinite transition cost. When two paths merge, the one with the higher cumulative cost can be safely eliminated.[4] When ambiguity exists, surviving paths are highlighted using an additional dotted line parallel to the path. The optimal path is clearly the one indicated by the dotted line which leads to state (1, 4) at time $n = 11$. We can read out the output (optimal approximation) by traversing this latter path backward, and registering the corresponding forward state transitions. The output then is $\{x(n)\}_{n=0}^{11} = \{1,1,1,1,0,0,0,0,1,1,1,1\}$.

IV. VORCA PROPERTIES

Definition 3: A filter, f, is *idempotent* if, and only if, $f(f(\boldsymbol{y})) = f(\boldsymbol{y}), \forall \boldsymbol{y}$.

We have the following proposition.

Proposition 1: If $d_n(\cdot,\cdot)$ is a distance metric between elements of $\mathcal{A}$ $\forall n \in \{0, 1, \cdots, N-1\}$, then the VORCA is idempotent, and the same is true, in general, for the VOSA.

Proof: We prove it for the VORCA. This way we avoid introducing unnecessary notation; the proof for the VOSA is exactly the same. The output of a single application of VORCA is obviously in P_M^N. Suppose $\boldsymbol{y} \in P_M^N$. Clearly, $\Sigma_{n=0}^{N-1} d_n(y(n), x(n)) \geq 0, \forall \boldsymbol{x} \in P_M^N$. By virtue of the fact that $d_n(\cdot,\cdot)$ is a distance metric $\forall n \in \{0, 1, \cdots, N-1\}$, the only element, $\boldsymbol{x}$, of P_M^N which makes $\sum_{n=0}^{N-1} d_n(y(n), x(n))$ zero is $\boldsymbol{y}$ itself. In fact, we can guarantee idempotence under the relaxed condition that $\forall n \in \{0, 1, \cdots, N-1\}$, $d_n(\cdot,\cdot)$ achieves its minimum value if and only if its arguments are equal. ■

In the following, let us assume, for the sake of simplicity, that $\mathcal{A}$ can be identified with $\{0, 1, \cdots, L\}$, and let us define the *complement*, y^c, of an element, $y \in \mathcal{A}$, as $y^c = L - y$, and the complement, $\boldsymbol{y}^c$, of a sequence, $\boldsymbol{y}$, in the obvious way, i.e., as the pointwise complement of its elements with respect to L.

[4] Ties do not appear in this simple example.

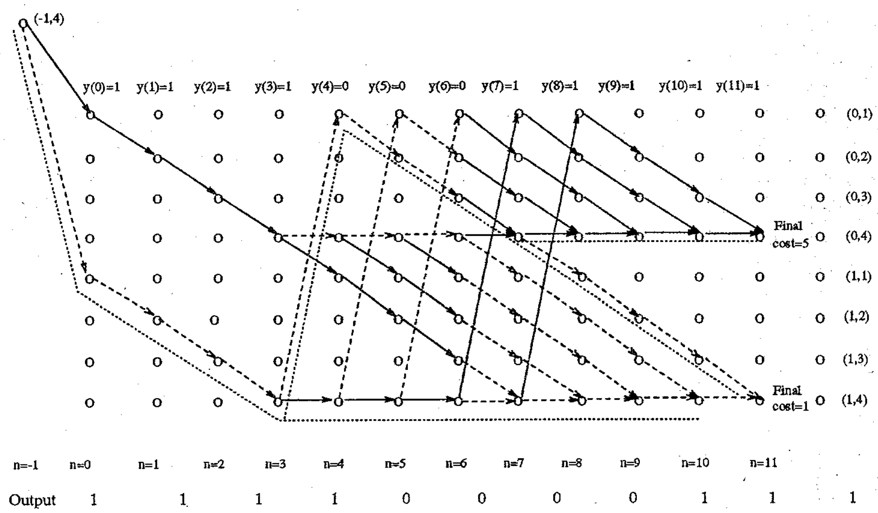


Fig. 1. VORCA trellis.

Definition 4: A filter, f, is *self-dual* if, and only if, $f(\boldsymbol{y}^c) = (f(\boldsymbol{y}))^c$.

We have the following proposition.

Proposition 2: If $d_n(y,x) = d_n(y^c, x^c), n = 0, 1, \cdots, N-1, \forall y, x \in \mathcal{A}$, and the VOSA constraint is self-dual (in the sense that $\boldsymbol{x}$ satisfies the constraint if and only if $\boldsymbol{x}^c$ does so), then, without loss of optimality, if so desired, the VOSA can be designed to be self-dual. Observe, in particular, that the VORCA can be designed to be self-dual, provided that the first condition holds.

Proof: We prove it for the VORCA. The proof for the VOSA is exactly the same. Let $\hat{\boldsymbol{x}}$ be *a* solution (minimizer) of (3), (4). Then

$$\sum_{n=0}^{N-1} d_n(y(n), \hat{x}(n)) \leq \sum_{n=0}^{N-1} d_n(y(n), x(n)), \quad \boldsymbol{x} \in P_M^N.$$

By the first condition in the statement of this proposition, it follows that

$$\sum_{n=0}^{N-1} d_n(y^c(n), \hat{x}^c(n)) \leq \sum_{n=0}^{N-1} d_n(y^c(n), x^c(n)), \quad \boldsymbol{x} \in P_M^N$$

and since $\boldsymbol{x} \in P_M^N$ if, and only if, $\boldsymbol{x}^c \in P_M^N$, this is the same as

$$\sum_{n=0}^{N-1} d_n(y^c(n), \hat{x}^c(n)) \leq \sum_{n=0}^{N-1} d_n(y^c(n), x^c(n)), \quad \boldsymbol{x}^c \in P_M^N$$

let $z = \boldsymbol{x}^c$, and

$$\sum_{n=0}^{N-1} d_n(y^c(n), \hat{x}^c(n)) \leq \sum_{n=0}^{N-1} d_n(y^c(n), z(n)), \quad \forall \boldsymbol{z} \in P_M^N$$

which implies that $\hat{\boldsymbol{x}}^c$ (which is in P_M^N) is *a* solution (minimizer) of (3) and (4) with $\boldsymbol{y}$ replaced by $\boldsymbol{y}^c$. So far, we have shown that if $\hat{\boldsymbol{x}}$ is optimal for $\boldsymbol{y}$, then $\hat{\boldsymbol{x}}^c$ is optimal for $\hat{\boldsymbol{y}}^c$. However, this does not immediately imply that any given implementation of VORCA will be self-dual. There is a subtle point here that arises due to the possibility of having multiple minimizers. Conventional tie-breaking strategies may violate self-duality. However, one can enforce self-duality without compromising optimality as follows. Given an input sequence, $\boldsymbol{y}$, one can decide whether to work with $\boldsymbol{y}$ or $\boldsymbol{y}^c$ in a consistent fashion, e.g., using a level test on $y(0)$, i.e., for L odd, test whether $y(0) < (\text{real})L/2$; if so, work as usual with $\boldsymbol{y}$; else, work with its complement, $\boldsymbol{y}^c$, *and complement the final result.* By virtue of the last inequality, this does not compromise optimality, for the solution obtained this way is as good as any. ■

In the binary case, self-duality means that the filter treats an "object" and its "background" in a balanced fashion. This is a desirable property.

The median is a self-dual filter, but it is not idempotent. This implies that, even though a single median filtering step (pass) is computationally less intensive than running the VORCA trellis, the overall computation required to iterate the median until convergence may surpass VORCA complexity, since the latter filter converges in one pass. Furthermore, the VORCA is optimal *by design*, while the median is not guaranteed to be optimal.

Definition 5: $\boldsymbol{y}_1 \leq \boldsymbol{y}_2$ if and only if $y_1(n) \leq y_2(n), \forall n \in \{0, 1, \cdots, N-1\}$.

Definition 6: A filter, f, is *increasing* if, and only if, $\boldsymbol{y}_1 \leq \boldsymbol{y}_2 \Rightarrow f(\boldsymbol{y}_1) \leq f(\boldsymbol{y}_2), \forall \boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathcal{A}^N$, where $\mathcal{A}^N$ stands for the set of all sequences of N elements of $\mathcal{A}$.

We have the following proposition, which at first might seem counter-intuitive.

Proposition 3: The VORCA is not, in general, an increasing filter.

Proof: We prove it by what we think is a particularly illuminating counter-example. This is depicted in Fig. 2. For this example, we assume that $M = 5$, and $d_n(y(n), x(n)) = |y(n) - x(n)|, \forall n \in \{0, 1, \cdots, N-1\}$. The caption is self-explanatory. ■

We have the following important corollary.

Corollary 2: The VORCA is neither a morphological[5] nor a stack[6] filter.

[5] Morphological filters are increasing [1].

[6] *Stack filters* [59], [7] are a class of increasing nonlinear operators which obey the so-called *threshold decomposition* property. This class includes all rank-order filters, i.e., filters based on rank ordering, e.g., min/max/median filters and compositions thereof.

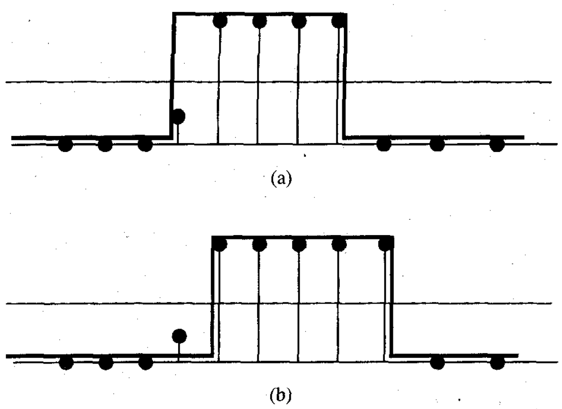


Fig. 2. Counterexample illustrating the fact that the VORCA is not, in general, increasing. Input data points are depicted using •. The optimal runlength-constrained approximation (VORCA output) is depicted using a thick continuous line, while the dashed line parallel to the horizontal axis is at half the level of the maximal value of the input data sequence. M is equal to five. The input in (a) lies below the input in (b), but the same is not true for the corresponding outputs.

As a direct consequence, there is no hope in trying to approximate the optimal filter (i.e., the VORCA) by using a morphological filter (e.g., by using the basis representation theory for morphological filters) or a stack filter.

A natural question that comes to mind is when can we expect to be faced with a constrained optimization of type (3) and (4) and, therefore, anticipate that the optimal filter is not increasing, and thus not in the usual toolbox of nonlinear filters? The following claim provides a reasonable scenario in which this happens.

Claim 2: Whenever we have a finite-alphabet signal, $\boldsymbol{x}$, which is piecewise constant of plateau (run) length $\geq M$, and observation, $\boldsymbol{y}$, arising from additive, two-sided, finite-alphabet, independent (yet not necessarily identically distributed) noise, with marginal probability mass $p_N^{(n)}(\cdot), n \in \{0, 1, \cdots, N-1\}$, the ML principle leads to a constrained optimization of type (3) and (4), and, therefore, the VORCA is an optimal (ML) estimator.

Proof: This is a direct consequence of the ML principle. Let arg max stand for "*an* argument that maximizes ..." Then

$$\hat{\boldsymbol{x}}_{ML}(\boldsymbol{y}) = \arg \max_{\boldsymbol{x} \in P_M^N} \log \Pr(\boldsymbol{y}|\boldsymbol{x})$$
$$= \arg \max_{\boldsymbol{x} \in P_M^N} \log \prod_{n=0}^{N-1} p_N^{(n)}(y(n) - x(n))$$
$$= \arg \min_{\boldsymbol{x} \in P_M^N} \sum_{n=0}^{N-1} -\log\, p_N^{(n)}(y(n) - x(n)).$$

If we let $d_n(y(n), x(n)) = -\log p_N^{(n)}(y(n) - x(n)), \forall n \in \{0, 1, \cdots, N-1\}$, then we end up with a constrained optimization of type (3) and (4). We point out that $d_n(\cdot, \cdot)$ need not be a distance metric; this is not required by our algorithm.

A. Complexity

The VORCA has computational complexity which is *linear* in the number of observations, i.e., N. The number of computations per symbol depends on the number of states, as well as state connectivity in the trellis. Each stage in the trellis has a total of $|\mathcal{A}|M$ states, out of which $|\mathcal{A}|$ are of the form $[v, 1]^T, v \in \mathcal{A}$, $|\mathcal{A}|$ are of the form $[v, M]^T, v \in \mathcal{A}$, and the remaining $|\mathcal{A}|M - 2|\mathcal{A}|$ are of the form $[v, l]^T, v \in \mathcal{A}, 1 < l < M$. States of the first kind at time $n+1$ can only be reached from $|\mathcal{A}| - 1$ candidate states at time n, namely those of type $[w, M]^T, w \in \mathcal{A}, w \neq v$. States of the second kind at time $n+1$ can only be reached from two states at time n, namely $[v, M-1]^T$, and $[v, M]^T$. States of the third kind at time $n+1$ can only be reached from one state at time n, namely, $[v, l-1]^T$. Therefore, one needs $|\mathcal{A}| - 1$ distance (branch metric) calculations and integer additions, and $|\mathcal{A}| - 2$ integer comparisons per state of the first kind, times $|\mathcal{A}|$ states of the first kind, for a subtotal of $|\mathcal{A}|^2 - |\mathcal{A}|$ distance calculations and integer additions, and $|\mathcal{A}|^2 - 2|\mathcal{A}|$ integer comparisons for all states of the first kind, per symbol. Similarly, one needs a subtotal of $2|\mathcal{A}|$ distance calculations and integer additions, and $|\mathcal{A}|$ integer comparisons for all states of the second kind, per symbol. Finally, one needs a subtotal of $|\mathcal{A}|M - 2|\mathcal{A}|$ distance calculations and integer additions, and zero integer comparisons for all states of the third kind, per symbol. The grand totals are $|\mathcal{A}|^2 + |\mathcal{A}|(M-1)$ distance calculations and integer additions, and $|\mathcal{A}|^2 - |\mathcal{A}|$ integer comparisons per symbol (i.e., trellis stage). This translates to an overall computational complexity of $O(|\mathcal{A}|^2 + |\mathcal{A}|(M-1))$ integer operations per symbol, and $O((|\mathcal{A}|^2 + |\mathcal{A}|(M-1)) \times N)$ integer operations for the entire optimization. The required number of distance calculations and integer additions per symbol is tabulated in Table I, for typical values of $|\mathcal{A}|, M$ (the number of required integer comparisons is always less). Clearly, $|\mathcal{A}|$ (i.e., the size of the alphabet) is the dominating factor.

TABLE I
NUMBER OF DISTANCE CALCULATIONS AND INTEGER ADDITIONS PER SYMBOL (I.E., PER VORCA TRELLIS STAGE). THE NUMBER OF INTEGER COMPARISONS IS ALWAYS LESS THAN THIS NUMBER, AND THE COMPUTATIONAL COMPLEXITY PER VORCA TRELLIS STAGE IS ALWAYS LESS THAN TWICE THIS NUMBER

	$M=5$	$M=10$	$M=15$	$M=20$	$M=25$	$M=30$
$\lvert\mathcal{A}\rvert = 2$	12	22	32	42	52	62
$\lvert\mathcal{A}\rvert = 16$	320	400	480	560	640	720
$\lvert\mathcal{A}\rvert = 32$	1152	1312	1472	1632	1792	1952
$\lvert\mathcal{A}\rvert = 64$	4352	4672	4992	5312	5632	5952
$\lvert\mathcal{A}\rvert = 128$	16896	17536	18176	18816	19456	20096
$\lvert\mathcal{A}\rvert = 256$	66560	67840	69120	70400	71680	72960

The worst-case storage requirements of VORCA are $O(|\mathcal{A}|M \times N)$, but actual storage requirements are much more modest, due to path merging.

The availability of VLSI Viterbi decoding chips, as well as several dedicated multiprocessor architectures for Viterbi-type decoding, makes the VORCA a realistic alternative to standard nonlinear filtering, at least for moderate values of $|\mathcal{A}|, M$. In the binary case, current Viterbi technology [60]–[64] can handle 2^{12} states. Hardware capability is continuously improving, and at a rather healthy pace. Viterbi-type techniques, like the VORCA, will certainly benefit from these developments.

V. SIMULATION EXAMPLE

Let us now present a complete simulation experiment. Fig. 4 depicts a typical input sequence. This particular input has been generated by adding i.i.d. noise on some artificial

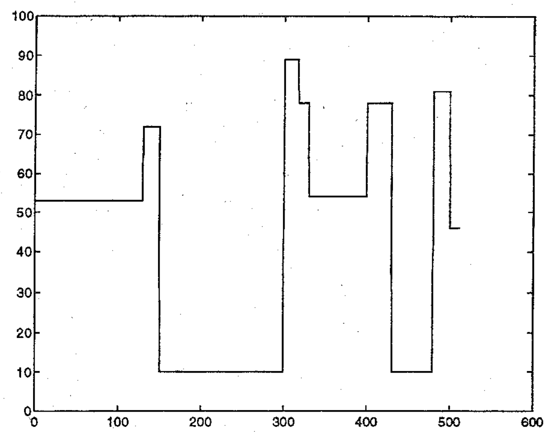


Fig. 3. "True" noise-free test data.

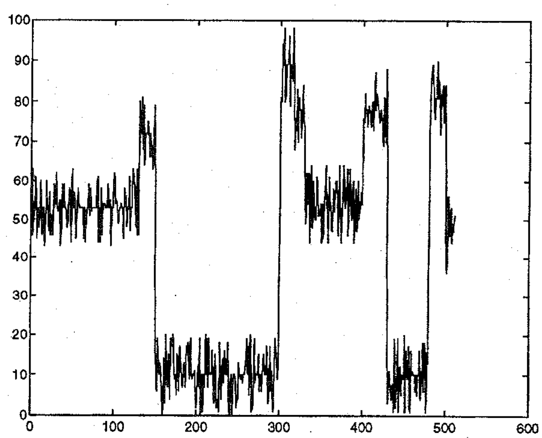


Fig. 4. Input sequence $\{y(n)\}_{n=0}^{511}$.

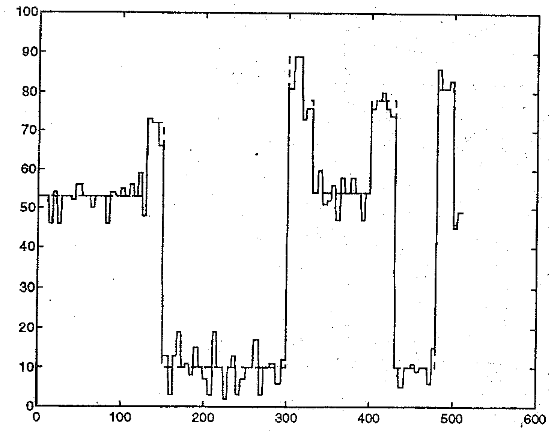


Fig. 5. VORCA output $M = 5$.

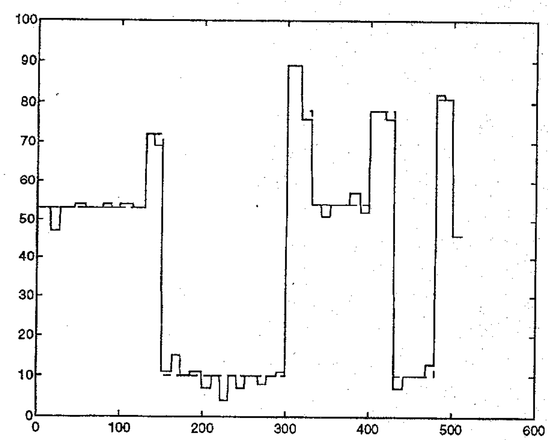


Fig. 6. VORCA output $M = 10$.

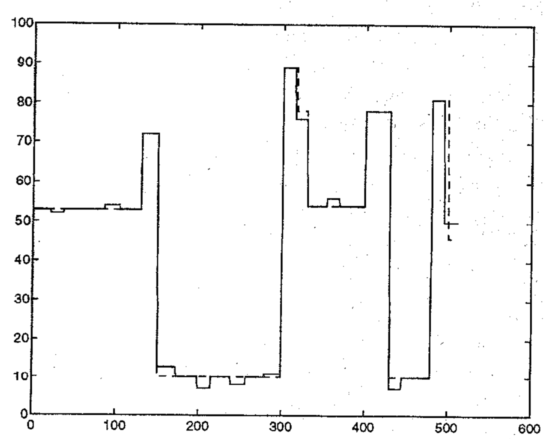


Fig. 7. VORCA output $M = 15$.

"true" noise-free test data, depicted in Fig. 3. The noise has been generated according to a uniform distribution, and most of the data points are contaminated. It should be stressed that this is a "distribution-free" experiment, in that we do not use our prior knowledge of the noise model to match VORCA to the noise characteristics, which is certainly a possibility (cf. Claim 2 in Section IV); the VORCA can handle both short-tailed, and long-tailed, even inhomogeneous noise with equal ease. The noise-free test data of Fig. 3 is also overlaid on subsequent plots. This is meant to help the reader judge filtering "quality," yet, again, "true" data should be taken with a grain of salt, for *in practice we obviously do not have access to the true data*, and, therefore, comparisons relative to the "true" data may be a bit misleading. Visual perception is arguably the ultimate "gold standard," and the reader is encouraged to attempt to trace edges in the observation data depicted in Fig. 4. Chances are that his/hers sketch will occasionally differ from the "true" data.

For this example, we take $d_n(y(n), x(n)) = |y(n) - x(n)|, \forall n \in \{0, 1, \cdots, N-1\}, \mathcal{A} = \{0, \cdots, 99\}$, and $N = 512$. The resulting optimal approximation (VORCA output sequence) for $M = 5, 10, 15, 20, 25, 30$, and 40 is depicted in Figs. 5–11, respectively. The results are rather remarkable. Observe that strong edges in the data remain uniformly localized for a wide range of values of M. This is a desirable property. Fig. 12 presents a plot of the resulting average per-letter approximation error (i.e., $1/N\ \Sigma_{n=0}^{N-1}\ |y(n) - x(n)|$), as a function of M. Observe that this error figure (which, by virtue of optimality, is necessarily a nondecreasing function of M) stabilizes for values of M between 20 and 30, then

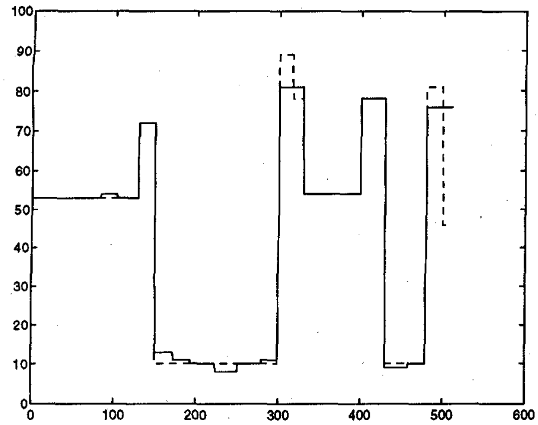


Fig. 8. VORCA output $M = 20$.

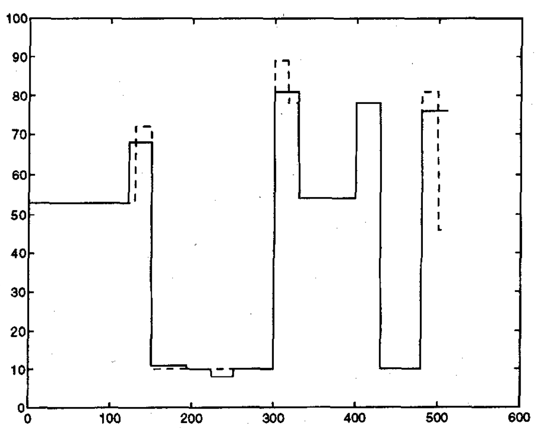


Fig. 9. VORCA output $M = 25$.

TABLE II

COMPARISON OF VORCA VERSUS SOME WELL-KNOWN NONLINEAR FILTERS ON THE BASIS OF THE RESULTING AVERAGE PER-LETTER DISTORTION, FOR A TYPICAL INPUT. FILTER LENGTH PARAMETERS HAVE BEEN INDIVIDUALLY ADJUSTED TO PRESERVE SIGNALS PRESERVED BY VORCA OPERATING WITH $M = 30$ WHILE MAXIMALLY SUPPRESSING THE NOISE

	VORCA, $M = 30$	Median, $\|D\| = 59$	Median root, $\|D\| = 59$	Openclose, $\|W\| = 30$
APLD	4.962	7.115	7.142	12.433

TABLE III

SYNTACTIC COMPARISON OF VORCA VERSUS SOME WELL-KNOWN NONLINEAR FILTERS FOR A TYPICAL INPUT. FILTER LENGTH PARAMETERS HAVE BEEN ADJUSTED TO APPROXIMATELY EQUALIZE THE RESULTING AVERAGE PER-LETTER DISTORTION

	VORCA, $M = 30$	Median, $\|D\| = 39$	Median root, $\|D\| = 39$	Openclose, $\|W\| = 7$
APLD	4.962	4.923	4.992	5.107

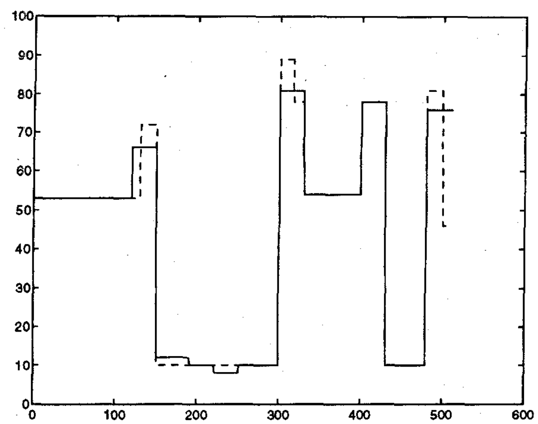


Fig. 10. VORCA output $M = 30$.

grows approximately linearly with increasing M. The visually best compromise seems to be around $M = 25$ or 30, which is consistent with the fact that the uncorrupted edge data for this simulation experiment has *significant* plateaus commensurate with this choice. This behavior is typical in several of our experiments. This suggests that one might be able to pick the "best" M, by studying performance plots just like the one in Fig. 12. This possibility warrants further investigation.

Two comparative simulation experiments are presented in Figs. 10, 13, and 14 and Table II, and Figs. 10, 15, and 16 and Table III, respectively. In these experiments, the output of VORCA for $M = 30$ is compared with a standard median with respect to a convex symmetric window D of length $|D|$, the median root with respect to D, and a morphological openclose filter with respect to a convex symmetric structural element W of length $|W|$. For each filter, the length parameter is *individually* adjusted to provide a common basis for a meaningful comparison. The conclusions for the closopen filter are very similar to the ones for the openclose filter; results for the former are thereby omitted. In the same spirit, and for the two experiments below, plots for the median filter are very close (and, in fact, slightly inferior) to those for the median root, and, therefore, the former are omitted.

The first experiment is a comparison on the basis of the resulting average per-letter distortion, for a typical input (namely, that of Fig. 4). For this comparison, the length parameter of each filter has been adjusted to preserve signals which are piecewise-constant of plateau (run) length ≥ 30 while maximally suppressing the noise. Results are presented in Figs. 10, 13, and 14 and Table II. Observe that the VORCA is not only reliably picking up the signal edges while at the same time essentially eliminating the noise, it also beats the other filters in terms of distortion, and by a significant margin. This also results in visually superior performance.

The second experiment is a syntactic comparison for the same input. Filter length parameters have been adjusted to approximately equalize average per-letter distortion. Results are presented in Figs. 10, 15, and 16 and Table II. Observe that the VORCA exhibits better signal edge localization while at the same time suppressing more noise than its competitors. This results in visually superior performance. Observe, in particular, the poor noise suppression capabilities of openclose, for this particular choice of $|W|$.

The drawback of VORCA relative to these filters is that, in general, it has higher complexity. However, this complexity is not prohibitive, and, given enough resources (as is more and more often the case in these days of exponential increases in hardware capability), we should opt for the best possible filter.

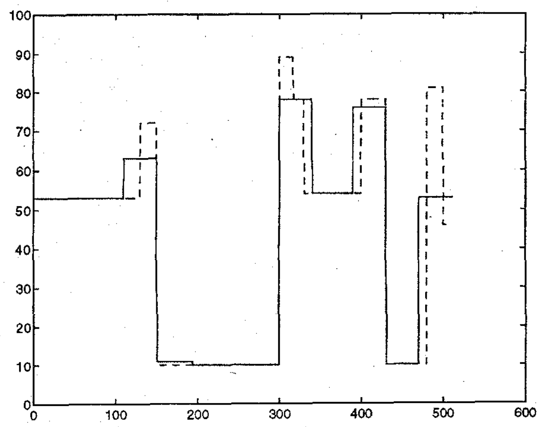


Fig. 11. VORCA output $M = 40$.

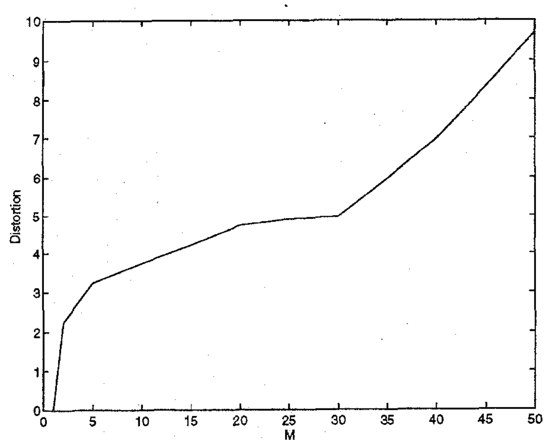


Fig. 12. Plot of average per-letter approximation error as a function of M.

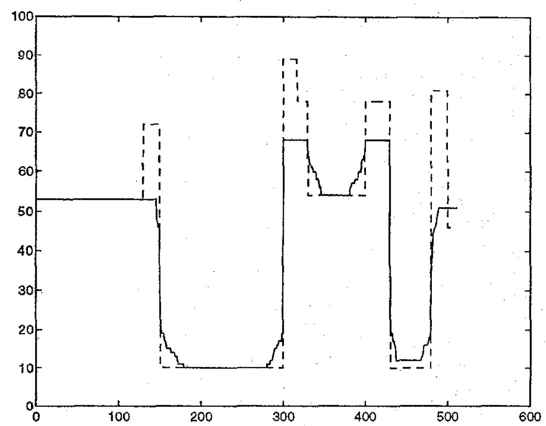


Fig. 13. Median root $|D| = 59$.

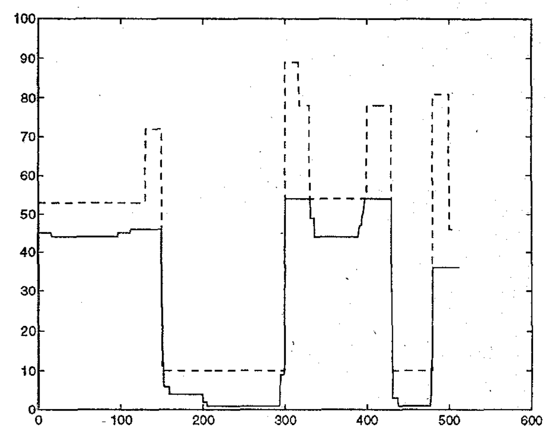


Fig. 14. Output of openclose $|W| = 30$.

VI. Applications

A. Optimal Filtering

Nonlinear filter analysis and synthesis typically draws heavily on two important tools, namely, root signal structure and output distribution for i.i.d. input statistics. A signal s is said to be a *root* or *fixed point* of an operator (filter) f if and only if $f(s) = s$, in which case, we also say that s is *invariant* or *smooth* under f. The collection of all signals which are invariant under f is variably called the *root set, set of fixed points, or domain of invariance of* f. We will adopt the latter convention and denote this collection of signals by $\text{Inv}(f)$, yet we will sometimes refer to elements of $\text{Inv}(f)$ as *roots of* f.

For ideal linear filters, the domain of invariance is given by the set of all signals in the filter's passband. Unfortunately, the analogy stops here, for nonlinear filters do not obey the superposition principle. Nevertheless, root signal analysis is still useful, since it allows one to specify *structural* (i.e., syntactic) constraints on filter behavior. This kind of analysis is purely deterministic. Idempotent filters converge to a signal in their domain of invariance in just one step, for all input signals. Several useful filters are not idempotent. A prime example is the median filter. For finite-duration signals, the median, although not idempotent, always converges to some signal in its domain of invariance, and in a finite number of steps (passes). Similar results exist for other nonlinear filter classes of practical interest. The idea, then, becomes clear: Given the syntactic properties of some desirable signal, which is embedded in noise, design a nonlinear filter f to extract this signal from a noisy observation by specifying the domain of invariance of f in such a way that $\text{Inv}(f)$ agrees as much as possible with (ideally, equal to) S, which is the set of all signals that comply with the given set of syntactic properties of the desirable signal. If repeated applications of f converge, they must converge to a signal in $\text{Inv}(f)$, and therefore, one is always assured of obtaining a final estimate that complies with the given set of desirable syntactic properties. Nevertheless, this estimate may be very far off from the true input signal; the hope is that if the noise level is low, and/or the noise is highly unstructured, then the resulting estimate will be reasonably close to the true signal. This approach obviously ignores signal and/or noise statistics; instead, it focuses solely on syntactic properties.

The output distribution for i.i.d. input statistics is often used as a "rule of thumb" for judging the noise attenuation capabilities of a particular nonlinear filter structure. This kind of (elementary) analysis is clearly inadequate in most cases of interest.

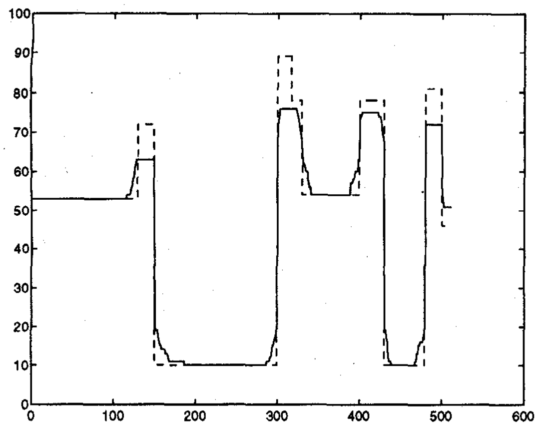


Fig. 15. Median root $|D| = 39$.

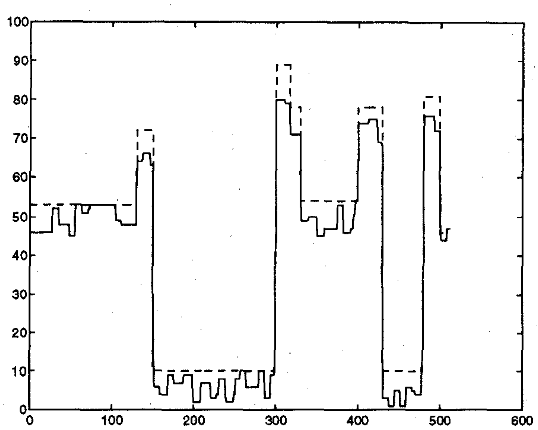


Fig. 16. Output of openclose $|W| = 7$.

Several authors have studied generalizations of the median (*rank-order* and *stack* filters) under a more appropriate blend of structural constraints and statistical hypotheses (e.g., [65]–[67]). In a very recent article [68], Yin considers a related design problem under a hard symbol runlength constraint, and a statistical model which assumes that the input is a constant plus additive i.i.d. noise[7] and optimizes over the class of stack filters of a given length. However, we have shown (cf. Claim 2) that, given a rather general formulation of the problem of optimal runlength-constrained approximation (which includes, in particular, per-letter distance-based optimal filtering criteria, and the ML criterion), the optimal filter is not, in general, an increasing operator, and, therefore, the class of stack filters is suboptimal, for stack filters are *increasing by definition* [59].

In a recent paper [69], Niedzwiecki and Sethares propose a novel nonlinear filtering approach based on the idea of using a *set* of competing forward and backward linear predictors and (possibly nonlinear) smoothers, along with a nonlinear combiner or decision rule. Their means are different from ours, yet their aim is close in spirit to ours; interested readers should consult [69]. The work of Restrepo and Bovik [70] is another interesting reference. In their formulation, the set of all locally monotonic (*lomo*) signals of length N and lomo-degree α plays the role of P_M^N in our formulation. Theirs is a wider class of admissible signals, which may or may not be proper for a particular application. They provide an elegant mathematical framework in which they consider existence and uniqueness of solutions. However, the complexity of their algorithms is combinatorial in N. In contrast, the complexity of our algorithm is linear in N.

Let us now shift gears and present a concrete example. Let $\{x(n)\}_{n=0}^{N-1}$ be a finite-duration sequence of binary variables. This is our signal. Suppose that it is piecewise-constant of plateau (run) length $\geq M$. Assume that $\{x(n)\}_{n=0}^{N-1}$ is transmitted over a memoryless Binary Symmetric Channel (BSC), of symbol inversion probability p. We may assume, without loss of generality, that $p \leq 0.5$ (otherwise, we simply invert the channel outputs). The output (observable) sequence is $\{y(n)\}_{n=0}^{N-1}$. We wish to recover (i.e., form an estimate of) $\{x(n)\}_{n=0}^{N-1}$ on the basis of $\{y(n)\}_{n=0}^{N-1}$. It is easy to see that, in accordance with Claim 2, the ML principle leads in this case to the optimization given by (3), (4). The optimal (ML) solution is given by the VORCA.

"Standard" approaches of smoothing the output data in this case, while hopefully remaining "close" to the true signal (i.e., preserving plateaus), include using a median, recursive median, morphological openclose, or closopen filter [1], [71]–[73]. Let $\text{med}_D(\cdot)$ denote the median with respect to a convex symmetric window, D, of size $2(M-1)+1$, and $\gamma_W(\cdot), \phi_W(\cdot)$ denote morphological opening, and closing, respectively, with respect to a convex structural element, W, of size M. Opening and closing are idempotent filters [1]. They have been shown to be optimal with respect to the given criteria under one-sided noise [74], [75]. The median is not idempotent, so let $\text{med}_D^{\infty}(\cdot)$ denote the median root (one always exists in this case, since we have assumed finite-extent signals [7]). Obviously, $\text{med}_D^{\infty}(\cdot) \in \text{Inv}(\text{med}_D)$ (meaning that the output of the operator for *any* input will be in $\text{Inv}(\text{med}_D)$), which in this special case is known to be exactly P_M^N [7], [76]–[78]. Therefore, $\text{med}_D^{\infty}(\cdot) \in P_M^N$, i.e., the set of all piecewise-constant sequences of plateau length $\geq M$. Thus, filtering $\{y(n)\}_{n=0}^{N-1}$ using iterations of the median will result in a sequence satisfying constraint (4). But how close is this final result to the true data?

$\text{Inv}(\gamma_W)$ is the collection of all binary sequences having plateaus (runs) of 1's of length $\geq M$ [73], [1]. Similarly, $\text{Inv}(\phi_W)$ is the collection of all binary sequences having plateaus of 0's of length $\geq M$. Clearly, $\text{Inv}(\gamma_W) \cap \text{Inv}(\phi_W) = P_M^N$, i.e., the collection of all binary sequences which are piecewise constant of plateau length $\geq M$. The composite filters $\phi_W(\gamma_W(\cdot)), \gamma_W(\phi_W(\cdot))$ are known as openclose, and closopen, respectively [1]. In this special case, they are both invariant under further application of $\phi_W(\cdot)$ or $\gamma_W(\cdot)$ [76]–[78], and, therefore

$$\phi_W(\gamma_W(\cdot)) \in \text{Inv}(\gamma_W) \cap \text{Inv}(\phi_W) = P_M^N$$

and

$$\gamma_W(\phi_W(\cdot)) \in \text{Inv}(\gamma_W) \cap \text{Inv}(\phi_W) = P_M^N.$$

[7]This simplified model amounts to optimizing filter behavior in regions where the signal is approximately constant, and relying on the structural constraints to control behavior at or close to discontinuities. This compromise is motivated by the need to circumvent analytical difficulties.

Thus, filtering $\{y(n)\}_{n=0}^{N-1}$ using either $\phi_W(\gamma_W(\cdot))$, or $\gamma_W(\phi_W(\cdot))$ will also result in a sequence satisfying constraint (4). Furthermore, the final output of iterations of the median will obey the following pointwise order relation (note that this is not true in general) [1], [76]–[78]

$$\phi_W(\gamma_W(\{y(n)\}_{n=0}^{N-1})) \leq \mathrm{med}_D^{\infty}(\{y(n)\}_{n=0}^{N-1}) \leq \gamma_W(\phi_W(\{y(n)\}_{n=0}^{N-1})).$$

Since opening and closing are known to be optimal in the case of one-sided noise [74], is it possible that any one of the three filters above (openclose, iterations of the median, and closopen) is optimal in the more general setting of two-sided noise? The answer is a resounding **no**. Consider the simple example of Fig. 1. For this choice of $\{y(n)\}_{n=0}^{N-1}$, all three filters above result in a sequence of all 1's, which is clearly suboptimal. The same is true for the recursive median.

B. Edge Detection

As mentioned earlier, there exists an almost endless list of variational as well as *ad hoc* approaches to edge detection and image segmentation, and we certainly do not make any strong claims here. The reader is referred to Morel and Solimini [55] for an up-to-date exposition. However, we do want to point out that, in the context of edge detection, a minimum plateau (run) length constraint is more natural, robust, and effective than a constraint on the number of edges. The latter requires one to come up with a good *a priori* estimate of the true number of edges in the data (i.e., the *complexity* of the data) before one can apply a dynamic programming algorithm with sufficiently good results. If one overestimates the true number of edges in the data, one is likely to end up with many spurious and locally inconsistent noisy edges. On the other hand, the former method merely requires one to *define* what he or she considers to be the minimum acceptable plateau (run) length in the true data, i.e., where to "call it;" any structure below this prespecified threshold will be classified as noise and eliminated.

C. Polygonalization of Shape Boundaries

As previously mentioned in Section II, polygonalization of planar shape boundaries is a classic problem in the literature on shape representation and recognition and for a good reason: it offers an intuitive and practical means of obtaining compact shape descriptions which can capture "essential" shape structure, while eliminating unimportant details and/or noise artifacts. After all, there is ample evidence (e.g., the popular family game *Pictionary*® is one example that comes immediately to mind) that humans can effectively communicate visual information by means of sketches or line drawings.

One can map the boundary of a shape in 2-D discrete space into a 1-D equivalent description, using a standard tool, namely the so-called *turning sequence* (which is related to *chain coding*). Roughly speaking, one starts from a conveniently chosen point on the curve (e.g., the lowest-rightmost point) and follows the curve (e.g., clockwise) while recording the slope of point-to-point transitions. Given the turning sequence, it is possible to reconstruct the original boundary, modulo a rotation and/or translation. Observe that straight line pieces of the boundary manifest themselves as plateaus in the turning sequence. We may therefore pose the problem of polygonal boundary approximation as a piecewise-constant sequence approximation problem in terms of the associated turning sequence. In effect, this is a *landmark*-based approach, in that the final polygonal approximation is formed by connecting points on the original shape boundary, selected according to an *edge consistency* criterion, i.e., the selected points are "vertices" in-between two boundary pieces of sufficient ($\geq M$) length, which can be well approximated by linear segments. This approach provides an alternative polygonalization method, based on a *formal* definition of *edge saliency*, as opposed to previous landmark-based approaches, which were largely heuristic. This work is currently in progress, and results will be reported elsewhere.

VII. Conclusions and Further Research

Motivated in part by an observation related to some open problems in modern nonlinear filtering, we have posed and investigated a new formal optimization problem, namely, that of optimally approximating a sequence by a runlength-constrained sequence. We have demonstrated that a simple recasting of this latter problem leads to an efficient Viterbi-type optimal algorithmic solution.

We call the resulting input–output operator the Viterbi optimal runlength-constrained approximation filter. This filter is optimal *by design*, has reasonable complexity, and can be efficiently implemented in dedicated Viterbi hardware, as well as in general-purpose workstations. Its fundamental properties have been studied by adopting a nonlinear filtering viewpoint. In particular, we have shown that, under mild conditions, the VORCA is idempotent and, without loss of optimality, can be designed to be self-dual. We have also demonstrated that it is not increasing, by means of a counterexample. This implies that the VORCA is not a morphological filter and, therefore, any morphological filter provides a suboptimal solution to our optimization problem. This result is rather surprising, given our earlier results on the optimality of certain elementary morphological filters for some special cases of the problem at hand. The same suboptimality remark holds for all increasing nonlinear filters, including the median, the recursive median, rank-order, and stack filters.

A complete simulation experiment that corroborates our theoretical findings has also been presented. The results are quite impressive. We have also highlighted some potential applications, including edge detection, and polygonalization of planar shape boundaries. The latter is of interest to us, and it warrants further investigation. Finally, we have hinted at possible extensions (e.g., piecewise-linear runlength-constrained approximation), and these are also of interest.

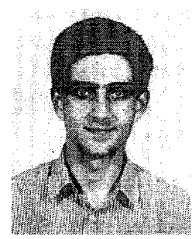
Acknowledgment

The author wishes to thank the anonymous reviewers, whose comments helped improve this manuscript. Thanks are also due to Prof. J. S. Baras for providing suggestions, support, and several references [79], Prof. C. A. Berenstein for several technical discussions, and Prof. A. Makowski for a fruitful technical discussion.

REFERENCES

[1] H. J. A. M. Heijmans, *Morphological Image Operators*. Boston: Academic, 1994.
[2] N. C. Gallagher Jr. and G. W. Wise, "A theoretical analysis of the properties of median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 1136–1141, Dec. 1981.
[3] B. I. Justusson, "Median filtering: Statistical properties," in *Two-Dimensional Digital Signal Processing II: Transforms and Median Filters*, T. S. Huang, Ed. Berlin: Springer-Verlag, 1981, pp. 161–196.
[4] T. A. Nodes and N. C. Gallagher, "Block median filters: Some modifications and their properties," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 739–746, 1982.
[5] A. C. Bovik, T. S. Huang, and D. C. Munson Jr., "The effect of median filtering on edge estimation and detection," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, pp. 181–194, Mar. 1987.
[6] N. C. Gallagher Jr., "Median filters: A tutorial," in *Proc. IEEE Int. Symp. Circ., Syst., ISCAS-88*, 1988, pp. 1737–1744.
[7] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications*. Boston: Kluwer, 1990.
[8] G. Qiu, "Functional optimization properties of median filtering," *IEEE Signal Processing Lett.*, vol. 1, pp. 64–65, 1994.
[9] R. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton University Press, 1957.
[10] R. Bellman and S. Dreyfus, *Applied Dynamic Programming*. Princeton, NJ: Princeton University Press, 1962.
[11] S. Dreyfus and A. Law, *The Art and Theory of Dynamic Programming*. New York: Academic, 1977.
[12] H. Stone, "Approximation of curves by line segments," *Math. Comput.*, vol. 15, pp. 40–47, 1961.
[13] R. Bellman, "On the approximation of curves by line segments using dynamic programming," *Commun. ACM*, vol. 4, pp. 284, 1961.
[14] B. Gluss, "A line segment curve-fitting algorithm related to optimal encoding of information," *Inform. Contr.*, vol. 5, pp. 261–267, 1962.
[15] ——, "Further remarks on line segment curve-fitting using dynamic programming," *Commun. ACM*, vol. 5, pp. 441–443, 1962.
[16] ——, "Least squares fitting of planes to surfaces using dynamic programming," *Commun. ACM*, vol. 6, pp. 172–175, 1963.
[17] ——, "An alternative method for continuous line segment curve-fitting," *Inform. Contr.*, vol. 7, pp. 200–206, 1964.
[18] R. Bellman, B. Gluss, and R. Roth, "On the identification of systems and the unscrambling of data: Some problems suggested by neurophysiology," *P. NAS US*, vol. 52, pp. 1239–1240, 1964.
[19] M. G. Cox, "Curve fitting with piecewise polynomials," *J. Inst. Math. Applications*, vol. 8, pp. 36–52, 1971.
[20] J. G. Dunham, "Optimum uniform piecewise linear approximation of planar curves," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, pp. 67–75, Jan. 1986.
[21] Y. Kurozumi and W. A. Davis, "Polygonal approximation by the minimax method," *Comput. Graphics Image Processing*, vol. 19, pp. 248–264, 1982.
[22] U. Montanari, "A note on minimal length polygonal approximation to a digitized contour," *Commun. ACM*, vol. 13, pp. 41–47, Jan. 1970.
[23] U. E. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Comput. Graphics Image Processing*, vol. 1, pp. 244–256, 1972.
[24] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[25] J. Slansky, R. L. Chazin, and B. J. Hansen, "Minimum perimeter polygons of digitized silhouettes," *IEEE Trans. Comput.*, vol. C-21, pp. 260–268, Mar. 1972.
[26] J. Slansky and V. Gonzalez, "Fast polygonal approximation of digitized curves," *Patt. Recog.*, vol. 12, pp. 327–331, 1980.
[27] I. Tomek, "Two algorithms for piecewise linear continuous approximations of functions of one variable," *IEEE Trans. Comput.*, vol. C-23, pp. 445–448, Apr. 1974.
[28] A. Rosenfeld and J. S. Weszka, "An improved method of angle detection on digital curves," *IEEE Trans. Comput.*, vol. C-24, pp. 940–941, Sept. 1975.
[29] K. A. Narayanan, D. P. O'Leary, and A. Rosenfeld, "Image smoothing and segmentation by cost minimization," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-12, no. 1, pp. 91–96, 1982.
[30] T. Pavlidis and S. L. Horowitz, "Segmentation of plane curves," *IEEE Trans. Comput.*, vol. C-23, pp. 860–870, Aug. 1974.
[31] T. Pavlidis, "A review of algorithms for shape analysis," *Comput. Graphics Image Processing*, vol. 7, pp. 242–258, 1978.
[32] ——, "Algorithms for shape analysis of contours and waveforms," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-2, pp. 301–312, 1980.
[33] D. Lee and T. Pavlidis, "One-dimensional regularization with discontinuities," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 10, pp. 822–829, Nov. 1988.
[34] J. Vandewalle, "On the calculation of the piecewise linear approximation to a discrete function," *IEEE Trans. Comput.*, vol. C-24, pp. 843–846, 1975.
[35] C. M. Williams, "An efficient algorithm for the piecewise linear approximation of planar curves," *Comput. Graphics Image Processing*, vol. 8, pp. 286–293, 1978.
[36] ——, "Bounded straight-line approximation of digitized straight curves and lines," *Comput. Graphics Image Processing*, vol. 16, pp. 370–381, 1981.
[37] F. Badi'i and B. Peikari, "Functional approximation of planar curves via adaptive segmentation," *Int. J. Syst. Sci.*, vol. 13, no. 6, pp. 667–674, 1982.
[38] L. D. Wu, "A piecewise linear approximation based on a statistical model," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-6, pp. 41–45, Jan. 1984.
[39] J. C. Bezdek and M. Anderson, "An application of the c-varieties clustering algorithm to polygonal curve fitting," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, pp. 637–641, Sept. 1985.
[40] H. Imai, "Computational-geometric methods for polygonal approximations of a curve," *Comput. Graphics Image Processing*, vol. 36, pp. 31–34, 1986.
[41] O. Baruch, "Segmentation of 2-D boundaries using the chain code," in *Proc. SPIE Visual Commun. Image Processing II*, vol. 845, 1987, pp. 159–166.
[42] C. H. Teh and R. T. Chin, "On the detection of dominant points on digital curves," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, no. 8, pp. 859–872, Aug. 1989.
[43] C. S. Fahn, J. F. Wang, and J. Y. Lee, "An adaptive reduction procedure for piecewise linear approximation," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, pp. 967–973, Sept. 1989.
[44] D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 14, no. 3, pp. 367–384, Mar. 1992.
[45] D. Geman, S. Geman, C. Graffigne, and P. Dong, "Boundary detection by constrained optimization," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 12, no. 7, pp. 609–627, 1990.
[46] U. Montanari, "On the optimal detection of curves in noisy pictures," *Commun. ACM*, vol. 14, no. 5, pp. 335–345, 1971.
[47] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos, "Dynamic programming for detecting, tracking, and matching deformable contours," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 17, no. 3, pp. 294–302, Mar. 1995.
[48] D. Terzopoulos, "Regularization of inverse visual problems involving discontinuities," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, no. 4, pp. 413–424, Jul. 1986.
[49] A. A. Amini, T. E. Weymouth, and R. C. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 12, no. 9, pp. 855–867, Sept. 1990.
[50] K. Konstantinides and B. K. Natarajan, "An architecture for lossy compression of waveforms using piecewise linear approximation," *IEEE Trans. Signal Processing*, vol. 42, no. 9, pp. 2449–2454, Sept. 1994.
[51] G. Papakonstantinou, P. Tsanakas, and G. Manis, "Parallel approaches to piecewise linear approximation," *Signal Processing*, vol. 37, pp. 415–423, 1994.
[52] D. Mumford and J. Shah, "Boundary detection by minimizing functionals," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, San Francisco, 1985.
[53] ——, "Optimal approximations by piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math.*, vol. 42, pp. 577–685, 1989.
[54] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, MA: MIT Press, 1987.
[55] J.-M. Morel and S. Solimini, *Variational Methods in Image Segmentation*. Berlin: Birkhauser, 1994.
[56] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 260–269, Apr. 1967.
[57] J. K. Omura, "On the Viterbi decoding algorithm," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 177–179, Jan. 1969.
[58] B. Sklar, *Digital Communications* Englewood Cliffs, NJ: Prentice-Hall, 1988.

[59] P. D. Wendt, E. J. Coyle, and N. C. Callagher, "Stack filters," *IEEE Trans. Acoust,, Speech, Signal Processing*, vol. ASSP-34, pp. 898–911, 1986.
[60] G. Feygin, P. G. Gulak, and P. Chow, "A multiprocessor architecture for Viterbi decoders with linear speedup," *IEEE Trans. Signal Processing*, vol. 41, no. 9, pp. 2907–2917, Sept. 1993.
[61] P. G. Gulak and E. Shwedyk, "VLSI structures for Viterbi receivers: Part I—General theory and applications," *IEEE J. Selected Areas Commun.*, vol. JSAC-4, pp. 142–154, Jan. 1986.
[62] S. Kubota, S. Kato, and T. Ishitani, "Novel Viterbi decoder VLSI implementation and its performance," *IEEE Trans. Commun.*, vol. 41, no. 8, pp. 1170–1178, Aug. 1993.
[63] K. K. Parhi, "High-speed VLSI architectures for Huffman and Viterbi decoders," *IEEE Trans. Circ. Syst. II*, vol. 39, no. 6, pp. 385–391, June 1992.
[64] T. K. Truong, M. T. Shih, I. S. Reed, and E. H. Satorius, "A VLSI design for a trace-back Viterbi decoder," *IEEE Trans. Commun.*, vol. 40, no. 3, pp. 616–624, Mar. 1992.
[65] E. J. Coyle and J. H. Lin, "Optimal stacking filters and mean absolute error nonlinear filtering," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. 36, no. 8, pp. 1244–1254, Aug. 1988.
[66] E. J. Coyle, J. H. Lin, and M. Gabbouj, "Optimal stack filtering and the estimation and structural approaches to image processing," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. 37, no. 12, pp. 2037–2066, Dec. 1989.
[67] M. Gabbouj and E. J. Coyle, "Minimum mean absolute error stack filtering with structural constraints and goals," *IEEE Trans. Acoust, Speech, Signal Processing*, vol. 38, no. 6, pp. 955–968, June 1990.
[68] L. Yin, "Stack filter design: A structural approach," *IEEE Trans. Signal Processing*, vol. 43, no. 4, pp. 831–840, Apr. 1995.
[69] M. Niedzwiecki and W. A. Sethares, "Smoothing of Discontinuous Signals: The Competitive Approach," *IEEE Trans. Signal Processing*, vol. 43, no. 1, pp. 1–13, Jan. 1995.
[70] A. Restrepo and A. C. Bovik, "Locally monotonic regression," *IEEE Trans. Signal Processing*, vol. 41, no. 9, pp. 2796–2810, Sept. 1993.
[71] J. Serra, Ed., *Image Analysis and Mathematical Morphology*. New York: Academic, 1982.
[72] ——, *Image Analysis and Mathematical Morphology, vol. 2, Theoretical Advances*. San Diego: Academic, 1988.
[73] G. Matheron, *Random Sets and Integral Geometry*. New York: Wiley, 1975.
[74] N. D. Sidiropoulos, J. S. Baras, and C. A. Berenstein, "Optimal filtering of digital binary images corrupted by union/intersection noise," *IEEE Trans. Image Processing*, vol. 3, no. 4, pp. 382–403, 1994.
[75] ——, "An algebraic analysis of the generating functional for discrete random sets, and statistical inference for intensity in the discrete boolean random set model," *J. Math. Imag., Vision*, vol. 4, pp. 273–290, 1994.
[76] P. Maragos, "Unified theory of translation-invariant systems with applications to morphological analysis and coding of images," Ph.D. dissertation, School of Elect. Eng., Georgia Inst. of Technol., Atlanta, 1985.
[77] P. Maragos and R. W. Schafer, "Morphological filters—part I: Their set-theoretic analysis and relations to linear shift-invariant filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, no. 8, pp. 1153–1169, Aug. 1987.
[78] ——, "Morphological filters—part II: Their relations to median, order-statistic, and stack filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, no. 8, pp. 1170–1184, Aug. 1987.
[79] D. C. MacEnany and J. S. Baras, "Scale-space polygonalization of target silhouettes and applications to model-based ATR," in *Proc. 2nd ATR Syst. Technol. Conf.*, Center for Night Vision and Electro-Optics, Ft. Belvoir, VA, vol. II, Mar. 1992, pp. 223–247.



Nicholas D. Sidiropoulos (S'90–M'92) received the Diploma in electrical engineering from the Aristotelian University of Thessaloniki, Greece, in 1988 and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland at College Park, in 1990 and 1992, respectively.

From 1988 to 1992, he was a Fulbright Fellow and a Research Assistant at the Institute for Systems Research of the University of Maryland. From September 1992 to June 1994, he served in the Hellenic Air Force. From October 1992 to October 1993, he served as a Lecturer at the Hellenic Air Force Academy. From October 1993 to June 1994, he was also a Member of the Technical Staff, Systems Integration Division, G-Systems Ltd., Athens, Greece. Since August 1994, he has been a Post-Doctoral Research Associate with the Institute for Systems Research of the University of Maryland at College Park. Since January 1995, he has also been an Adjunct Professor with the Department of Electrical Engineering of the University of Maryland at College Park. His research interests are in signal and image processing, nonlinear filtering, estimation and detection, random set theory, morphology, and medical imaging.

Dr. Sidiropoulos is a member of the Technical Chamber of Greece and is a registered Professional Engineer in Greece.