TECHNICAL RESEARCH REPORT

The Viterbi Optimal Runlength-Constrained Approximation Nonlinear Filter

by N.D. Sidiropoulos

T.R. 95-45

ISR
INSTITUTE FOR SYSTEMS RESEARCH

Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry

The Viterbi Optimal Runlength-Constrained Approximation Nonlinear Filter

N.D. Sidiropoulos

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, $\mathbf{y} = \{y(n)\}_{n=0}^{N-1}$, find a sequence, $\widehat{\mathbf{x}} = \{\widehat{x}(n)\}_{n=0}^{N-1}$, which minimizes $d(\mathbf{x},\mathbf{y}) = \sum_{n=0}^{N-1} d_n(y(n),x(n))$ subject to: $\mathbf{x}$ 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 and morphological filtering. We also discuss some potential applications.

Keywords

Nonlinear (Median, Morphological) Filtering, Principle of Optimality, Viterbi Algorithm

EDICS: Primary: SP 2.7.3, Secondary: SP 2.7.7, SP 6.1.7

N.D. Sidiropoulos is with the Institute for Systems Research, University of Maryland, College Park, MD 20742 U.S.A. He can be reached at (301) 405-6591, or via e-mail at nikos@Glue.umd.edu

I. INTRODUCTION

The *median filter*[1] [2], [3], [4], [5], [6], [7], [1] 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 i.i.d. two-sided impulsive noise, and the purpose is to recover the true data by eliminating outliers (noise impulses, which are locally inconsistent with the rest of the data). Indeed, the median 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 are idempotent by definition, but (as we will see) 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 extend), f, and we are interested in approximating, representing, or replacing f by a compact descriptor (i.e., reduced complexity data), g, which is optimal in some sense. Quite

---

[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.

often, g is also required to satisfy certain local regularity (e.g., continuity, smoothness), and/or structural (syntactic) constraints.

This kind of problem often appears in numerous 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 the subject. 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 quantifies faithfulness to the data; 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 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], [10], [11].

B. Organization

The rest of this paper is structured as follows. In the next subsection 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 note on implementation complexity, and a discussion on how to obtain an *a priori* minimax-type estimate of the approximation error are 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:

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

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

For obvious reasons, M will be referred to as the *constraint length.* 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. 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], [10], [11]. Gluss [14], [15], [16], [17] expanded on the original idea of Bellman. Bellman, Gluss, and Roth 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 **length** of segments, whereas Bellman's constraint is on the **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.

In 1986, Dunham [20] solved a related type-(2) optimization by applying the principle of optimality. His programme 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], [31], [32], [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.

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], [47], [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, Weymouth and Jain [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" 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 towards satisfying the constraints, but they cannot strictly enforce hard constraints. On the other hand, dynamic programming can easily accommodate hard 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 (a) the number of segments is fixed and equal to some predetermined constant (this is Bellman's constraint), **and** (b) 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 non-unique. 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 Non Convexity (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 $\mathbf{x} = \{x(n)\}_{n=0}^{N-1}$, $x(n) \in \mathcal{A}$, $n = 0, 1, \cdots, N-1$, define its associated *state sequence*, $\mathbf{s}_{\mathbf{x}} = \left\{[x(n), l_{\mathbf{x}}(n)]^T\right\}_{n=-1}^{N-1}$, where $[x(-1), l_{\mathbf{x}}(-1)]^T = [\phi, M]^T$, $\phi \notin \mathcal{A}$ and

$$l_{\mathbf{x}}(n+1) = \begin{cases} min\,\{l_{\mathbf{x}}(n)+1,\ M\} & ,\ x(n+1) = x(n) \\ 1 & ,otherwise \end{cases} \quad ,\ n = -1, \cdots, N-2$$

$[x(n), l_{\mathbf{x}}(n)]^T$ is the state at time n, and 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 $\left\{[x(n), l_{\mathbf{x}}(n)]^T\right\}_{n=-1}^{\nu}$, $\nu \leq N-1$, is *admissible* (with respect to constraint (4)) if and only if there exists a suffix string of state variables, $\left\{[s(n), l_{\mathbf{s}}(n)]^T\right\}_{n=\nu+1}^{N-1}$, such that $\left\{[x(n), l_{\mathbf{x}}(n)]^T\right\}_{n=-1}^{\nu}$ followed by $\left\{[s(n), l_{\mathbf{s}}(n)]^T\right\}_{n=\nu+1}^{N-1}$ is the associated state sequence of some sequence in P_M^N.
Let $\left\{[\hat{x}(n), l_{\hat{\mathbf{x}}}(n)]^T\right\}_{n=-1}^{N-1}$, be a solution of (3),(4). One always exists, although it may not necessarily be unique. Clearly, $\left\{[\hat{x}(n), l_{\hat{\mathbf{x}}}(n)]^T\right\}_{n=-1}^{N-1}$ is admissible, and so is any subsequence $\left\{[\hat{x}(n), l_{\hat{\mathbf{x}}}(n)]^T\right\}_{n=-1}^{\nu}$, $\nu \leq N-1$. The following is a key observation.

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

Proof: The argument goes as follows. Suppose that $\left\{[\tilde{x}(n), l_{\tilde{\mathbf{x}}}(n)]^T\right\}_{n=-1}^{\nu}$ is an admissible subsequence of states satisfying $[\tilde{x}(\nu), l_{\tilde{\mathbf{x}}}(\nu)]^T = [\hat{x}(\nu), l_{\hat{\mathbf{x}}}(\nu)]^T$. It is easy to see that $\left\{[\tilde{x}(n), l_{\tilde{\mathbf{x}}}(n)]^T\right\}_{n=-1}^{\nu}$ followed by $\left\{[\hat{x}(n), l_{\hat{\mathbf{x}}}(n)]^T\right\}_{n=\nu+1}^{N-1}$ is also admissible. The key point is that any suffix string of state variables which makes $\left\{[\hat{x}(n), l_{\hat{\mathbf{x}}}(n)]^T\right\}_{n=-1}^{\nu}$ admissible, will also make $\left\{[\tilde{x}(n), l_{\tilde{\mathbf{x}}}(n)]^T\right\}_{n=-1}^{\nu}$ admissible. If $\left\{[\tilde{x}(n), l_{\tilde{\mathbf{x}}}(n)]^T\right\}_{n=-1}^{\nu}$ has a smaller cost (distortion) than $\left\{[\hat{x}(n), l_{\hat{\mathbf{x}}}(n)]^T\right\}_{n=-1}^{\nu}$, then, by virtue of the fact that the cost is a sum of per-letter costs, $\left\{[\widetilde{x}(n), l_{\widetilde{\mathbf{x}}}(n)]^T\right\}_{n=-1}^{\nu}$ followed by $\left\{[\widehat{x}(n), l_{\widehat{\mathbf{x}}}(n)]^T\right\}_{n=\nu+1}^{N-1}$ will have a smaller cost than $\left\{[\widehat{x}(n), l_{\widehat{\mathbf{x}}}(n)]^T\right\}_{n=-1}^{N-1}$, 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: The 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], [57], [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 $c\left(\mathbf{s}_{\mathbf{x}}(n) \rightarrow \mathbf{s}_{\mathbf{x}}(n+1)\right)$ denote the cost of a one-step state transition. Then

$$c\left([x(n), l_{\mathbf{x}}(n)]^T \rightarrow [x(n+1), l_{\mathbf{x}}(n+1)]^T\right) =$$

$$\begin{cases} \infty & if\ [((l_{\mathbf{x}}(n) < M)\ OR\ (n \geq N-M))\ AND \\ & ((x(n+1) \neq x(n))\ OR\ (l_{\mathbf{x}}(n+1) \neq min\,\{l_{\mathbf{x}}(n)+1, M\}))]\ OR \\ & [(l_{\mathbf{x}}(n) = M)\ AND\ (x(n+1) = x(n))\ AND\ (l_{\mathbf{x}}(n+1) \neq M)]\ OR \\ & [(l_{\mathbf{x}}(n) = M)\ AND\ (x(n+1) \neq x(n))\ AND\ (l_{\mathbf{x}}(n+1) \neq 1)] \\ d_{n+1}(y(n+1), x(n+1)) & otherwise \end{cases} \tag{5}$$

will do it. A formal proof can be easily constructed, and is hereby omitted. We 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 $\mathbf{s}_{\mathbf{x}}(n) = [x(n), l_{\mathbf{x}}(n), t_{\mathbf{x}}(n)]^T$, where $t_{\mathbf{x}}(n)$ 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 straightforward, and is hereby omitted.

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 figure[3] 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 8 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 backwards, 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 $f(f(\mathbf{y})) = f(\mathbf{y})$, $\forall \mathbf{y}$.

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: The output of a single application of VORCA is obviously in P_M^N. Suppose $\mathbf{y} \in P_M^N$. Clearly, $\sum_{n=0}^{N-1} d_n(y(n), x(n)) \geq 0$, $\forall \mathbf{x} \in P_M^N$. 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, $\mathbf{x}$, of P_M^N which makes $\sum_{n=0}^{N-1} d_n(y(n), x(n))$ zero is $\mathbf{y}$ 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. ■

---

[3] Figures are placed at the end of this manuscript.

[4] Whenever a tie appears (i.e., two paths merge with equal cumulative costs), elimination should be performed in a consistent manner (e.g., always eliminate the upper path) in order to ensure uniqueness of the overall input-output nonlinear operator. Ties do not appear in this simple example.

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, $\mathbf{y}^c$, of a sequence, $\mathbf{y}$, in the obvious way, i.e., as the pointwise complement of its elements with respect to L.

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

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 $\mathbf{x}$ satisfies the constraint if and only if $\mathbf{x}^c$ does so), then the VOSA is a self-dual filter. Observe, in particular, that the VORCA is a self-dual filter, provided that the first condition holds.

Proof:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in P_M^N} \sum_{n=0}^{N-1} d_n(y(n), x(n)) = \arg \min_{\mathbf{x} \in P_M^N} \sum_{n=0}^{N-1} d_n(y^c(n), x^c(n))$$

$$= \arg \min_{\mathbf{x}^c \in P_M^N} \sum_{n=0}^{N-1} d_n(y^c(n), x^c(n)) \iff (\hat{\mathbf{x}})^c = \arg \min_{\mathbf{x} \in P_M^N} \sum_{n=0}^{N-1} d_n(y^c(n), x(n))$$

and the proof is complete. ■

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 well 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: $\mathbf{y}_1 \leq \mathbf{y}_2$ 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 $\mathbf{y}_1 \leq \mathbf{y}_2 \Longrightarrow f(\mathbf{y}_1) \leq f(\mathbf{y}_2)$, $\forall \mathbf{y}_1, \mathbf{y}_2 \in \mathcal{A}^N$, 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 figure 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 not a morphological filter.

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

A. Complexity

The VOSA has computational complexity which is *linear* in the number of observations, i.e., N. The proportionality constant is N_s^2, where N_s is the number of states in the trellis. This explicitly depends on $|\mathcal{A}|$, and the constraint length; e.g., the complexity of the VORCA is $O((|\mathcal{A}|M)^2 \times N)$. The worst-case storage requirements are $O(N_s \times N)$, and $O(|\mathcal{A}|M \times N)$ for the VORCA, but actual storage requirements are much more modest, due to path merging. Current Viterbi technology [59], [60], [61], [62], [63] can easily handle 2^{12} states. The availability of VLSI Viterbi decoder chips as well as several dedicated multiprocessor architectures for Viterbi-type decoding, makes the VOSA a realistic alternative to standard nonlinear filtering. One basically has to decide on a suitable off-the-shelf implementation according to his/her design constraints.

B. A Minimax Estimate of the Approximation Error

The choice of constraint length, M, depends on the idiosyncrasies of the particular application at hand. Observe that the VORCA not only outputs the optimal constrained approximation for any given M, it also outputs the actual *value* of the approximation error (distortion). However, one would in general be interested in an estimate of the error as a function of M. This would be helpful in deciding on an appropriate value of M beforehand. Such an estimate is made possible by virtue of *Rate-Distortion* Theory [64].

We can view the optimization in (3),(4), as run-length source coding. Since the length of each run is $\geq M$, there are at most $int(\frac{N}{M})$ runs in the approximation, where $int(\cdot)$ stands for integer part. Each run can be coded using at most $log(|\mathcal{A}|) + log(N - M + 1)$ bits, for a total of at most $int(\frac{N}{M}) \times [log(|\mathcal{A}|) + log(N - M + 1)]$ bits. Thus, the *rate* of the code is bounded above by

$$\bar{R} = \frac{int(\frac{N}{M}) \times [log(|\mathcal{A}|) + log(N - M + 1)]}{N \times log(|\mathcal{A}|)}$$

in bits per letter. For this worst-case rate, $\bar{R}$, the resulting per-letter distortion is bounded below by the value $D(\bar{R})$, where $D(R)$ denotes the distortion-rate function of the source. In terms of coding performance, the worst kind of source is a memoryless one, for which $D(R)$ can be computed. One may then obtain a minimax-type estimate on the per-letter distortion as a function of M by estimating the marginal source statistics, and calculating the corresponding rate-distortion-theoretic bound on the achievable distortion for a memoryless source, and for the worst-case rate $\bar{R}$. Alternatively, one may obtain a more conservative estimate by assuming a uniform distribution, which corresponds to the worst-case marginal statistics. In this case, assuming that $d_n(y(n), x(n)) = 1 - \delta(y(n) - x(n))$, $\forall n$, $\delta(\cdot)$ being the Kronecker delta, the rate-distortion function $R(D)$ (the "inverse" of $D(R)$) is simply given by [64]

$$R(D) = log|\mathcal{A}| - H_b(D) - Dlog(|\mathcal{A}| - 1)$$

where $H_b(\cdot)$ is Shannon's binary entropy function. Setting the left-hand side equal to $\bar{R}$, and solving for D, one obtains a minimax-type estimate on the per-letter distortion as a function of M.

V. SIMULATION EXAMPLE

Let us now present a typical simulation experiment. Figure 3 depicts a typical input sequence. For this example, we take $d_n(y(n), x(n)) = |y(n) - x(n)|$, $\forall n \in \{0, 1, \cdots, N-1\}$, $\mathcal{A} = \{0, 1, \cdots, 99\}$, and $N = 512$. The resulting optimal approximation (VORCA output sequence) for $M = 5, 10, 15, 20, 25, 30, 40$ is depicted in figures 4, 5, 6, 7, 8, 9, 10, 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. Figure 11 presents a plot of the resulting average per-letter approximation error (i.e., $\frac{1}{N}\sum_{n=0}^{N-1} |y(n) - x(n)|$), as a function of M. Observe that this error figure (which, by virtue of optimality, is necessarily a non-decreasing function of M) stabilizes for values of M between 20 and 30, then grows approximately linearly with increasing M. The visually best compromise seems to be $M = 30$, which is consistent with the fact that the true uncorrupted edge data for this simulation experiment have a minimum plateau (run) length approximately equal to 30. This behavior is typical in several of our experiments. This suggests that one might be able to pick the "best" constraint length, M, by studying performance plots just like the one in figure 11. This possibility warrants further investigation.

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, $\mathbf{s}$, is said to be a *root*, or *fixed point* of an operator (filter) f, if and only if $f(\mathbf{s}) = \mathbf{s}$, in which case we also say that $\mathbf{s}$ is *invariant*, or *smooth* under f. The collection of all signals which are invariant under f is variably called the *root set, or, set of fixed points, or, domain of invariance of f*. We will adopt the latter convention, and denote this collection of signals by $Inv(f)$, yet we will sometimes refer to elements of $Inv(f)$ 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 $Inv(f)$ agrees as much as possible with (ideally, equal to) S, the set of all signals which 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 $Inv(f)$, and, therefore, one is always assured of obtaining a final estimate which 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.

Several authors have studied generalizations of the median (*rank-order* filters, and, in particular, *stack* filters) under a more appropriate blend of structural constraints and statistical hypotheses (e.g., [65], [66], [67]). Their models do not fit our objectives.

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 the Maximum Likelihood (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 [68], [69], [70], [1]. Let $med_D(\cdot)$ 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 [71], [72]. The median is not idempotent, so let $med_D^{\infty}(\cdot)$ denote the median root (one always exists in this case, since we have assumed finite-extent signals [7]). Obviously, $med_D^{\infty}(\cdot) \in Inv(med_D)$ (meaning that the output of the operator for *any* input will be in $Inv(med_D)$), which in this special case is known to be exactly P_M^N [7], [73], [74], [75]. Therefore, $med_D^{\infty}(\cdot) \in P_M^N$, 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?

$Inv(\gamma_W)$ is the collection of all binary sequences having plateaus (runs) of 1's of length $\geq M$ [70], [1]. Similarly, $Inv(\phi_W)$ is the collection of all binary sequences having plateaus of 0's of length $\geq M$. Clearly, $Inv(\gamma_W) \cap Inv(\phi_W) = P_M^N$, 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 $\gamma_W(\cdot)$ [73], [74], [75], and, therefore,

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

and

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

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) [73], [74], [75], [1]

$$\phi_W(\gamma_W(\{y(n)\}_{n=0}^{N-1})) \leq med_D^{\infty}(\{y(n)\}_{n=0}^{N-1}) \leq \gamma_W(\phi_W(\{y(n)\}_{n=0}^{N-1}))$$

Since opening and closing are known to be optimal in the case of one-sided noise [71], 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 figure 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. Of course, the same conclusion could have been reached by invoking Corollary 2.

In general, whenever we have a multi-valued finite-alphabet signal, $\mathbf{x}$, which is piecewise constant of plateau (run) length $\geq M$, and observation, $\mathbf{y}$, arising from additive two-sided finite-alphabet i.i.d. noise with marginal probability mass $p_{\mathcal{N}}(z)$, the ML principle leads to a constrained optimization of type (3),(4), and, therefore, the VORCA is an optimal (ML) estimator. This is a direct consequence of the ML principle:

$$\begin{aligned}
\widehat{\mathbf{x}}_{ML}(\mathbf{y}) &= \underset{\mathbf{x} \in P_M^N}{arg\ \ max} \quad logPr(\mathbf{y} \mid \mathbf{x}) \\
&= \underset{\mathbf{x} \in P_M^N}{arg\ \ max} \quad log\textstyle\prod_{n=0}^{N-1} p_{\mathcal{N}}(y(n) - x(n)) \\
&= \underset{\mathbf{x} \in P_M^N}{arg\ \ min} \quad \textstyle\sum_{n=0}^{N-1} -logp_{\mathcal{N}}(y(n) - x(n))
\end{aligned}$$

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

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. Optimal Complexity-Constrained Polygonalization of Shape Boundaries

We now turn our attention to a particular application of practical interest to us. The setup goes as follows. We are given the boundary curve of a planar shape. This may be manually outlined by a human operator, or the output of another algorithm. Let us assume, for the purposes of this discussion, that the given boundary is a simple, closed, 8-connected curve[5]. This boundary description will typically be "rough", due to sensor noise, tracking inaccuracies, preprocessing, as well as inherent shape complexity. We would therefore like to obtain a "smooth", reduced-complexity polygonal approximation of this boundary, which is "optimal" in a suitable sense. Such an approximation would be useful in, among other things, shape classification, identification, and indexing for the purposes of retrieval. In this setting, complexity is important because it significantly impacts the *speed* of retrievals, as well as *storage requirements.* Smoothness is important because it is intimately related to *robustness and invariance* of the overall scheme.

Complexity can obviously be measured in terms of number of edges in the resulting polygonal approximation. What about smoothness? Consider the polygon depicted in figure 12 using a solid line. We would expect any reasonable "smooth" approximation of this polygon to bridge the two lower vertices by replacing the broken lower line by the dashed edge. A constraint on the number of edges cannot explicitly model - or enforce - this intuitive requirement. A constraint on edge-length can. Recall that a constraint of the latter type implies a constraint of the former type, but the reverse is not true. However, is an edge-length constraint enough to guarantee

---

[5] In general, one may need to allow a wider neighborhood structure which permits a richer collection of turning angles. However, this does not affect the essence of the approach.

smoothness of the solution? Consider the polygon depicted in figure 13. This is clearly not a "smooth" polygon. It becomes evident that smoothness ("scale") of a polygon depends on two things: minimum edge-length, and minimum angle between successive edges.

Let us now sketch a solution. First, we map this two-dimensional problem into a one-dimensional problem using a standard tool, namely, the *turning function* (which is related to *chain coding*). We start from a conveniently chosen point on the curve (e.g., the lowest-rightmost point), and follow the curve (e.g., clockwise) while recording the transitions. Since we have assumed a simple, 8-connected curve, each point on the curve has exactly two 8-connected neighbors on the curve (cf. figure 14), and, therefore, there is no ambiguity in following the curve. The result of this process is a sequence of symbols, each assuming values in $\{0, 1, \cdots, 7\}$.

Thus each boundary curve is mapped to a sequence of symbols in $\{0, 1, \cdots, 7\}$ (the *turning sequence*)[6]. Observe that straight boundary segments correspond to 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.

Several minor issues need to be addressed before one can apply our optimization framework to this problem. First, any turning sequence which corresponds to a closed curve is constrained by the fact that it should conform to a total turning of 360 degrees. This can be taken care of in several ways, the easiest of which is to "close" the resulting polygonal approximation at the end of the process, even though this may marginally sacrifice optimality. The second issue is that one must further restrict allowable trellis state transitions to enforce a minimum allowable angle requirement, in accordance with our previous discussion on what determines whether a given polygon is smooth or not. E.g., in reference to figure 14, given that the transition which brought us to the current state was of type "0", we may turn off transitions of type "3", "4", or "5". This type of constraint can be easily coded in a Viterbi-type algorithm. The final issue is that Hausdorff-Euclidean distance of planar shape boundaries cannot be measured in terms of per-letter distortion in turning sequence space, because the former is *cumulative* with respect to the latter. We may ameliorate this problem by allowing our distortion measure to depend on the full *state* of both the input, and output sequences. These adaptations go beyond the scope of this paper; they will be presented elsewhere.

---

[6] Note that this is an invertible mapping.

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 complexity $O(N)$, and it can be efficiently implemented using off-the-shelf hardware, by virtue of the wide availability of Viterbi decoder chips and chip sets. 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 self-dual. We have also demonstrated that it is not increasing, by means of a counter-example. 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 the median, the recursive median, and its (increasing) variants. This has been discussed in some detail in the binary case.

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

VIII. ACKNOWLEDGMENTS

Thanks are due to Prof. J.S. Baras for providing suggestions, support, and several references [76], Prof. C.A. Berenstein for several technical discussions, and Prof. A. Makowski, for a fruitful technical discussion. This research has been 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.

REFERENCES

[1] H.J.A.M. Heijmans, *Morphological Image Operators*, Academic Press, Boston, 1994.

[2] N.C. Gallagher Jr. and G.W. Wise, "A theoretical analysis of the properties of median filters", *IEEE Trans. ASSP*, 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., pp. 161–196. Springer-Verlag, Berlin, 1981.

[4] T. A. Nodes and N. C. Gallagher, "Median filters: some modifications and their properties", *IEEE Trans. ASSP*, 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. PAMI*, 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*, Kluwer, Boston, 1990.

[8] G. Qiu, "Functional optimization properties of median filtering", *IEEE Signal Processing Letters*, vol. 1, pp. 64–65, 1994.

[9] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, N.J., 1957.

[10] R. Bellman and S. Dreyfus, *Applied Dynamic Programming*, Princeton University Press, Princeton, N.J., 1962.

[11] S. Dreyfus and A. Law, *The Art and Theory of Dynamic Programming*, Academic, New York, NY, 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", *Information and Control*, vol. 5, pp. 261–267, 1962.

[15] B. Gluss, "Further remarks on line segment curve-fitting using dynamic programming", *Commun. ACM*, vol. 5, pp. 441–443, 1962.

[16] B. Gluss, "Least squares fitting of planes to surfaces using dynamic programming", *Commun. ACM*, vol. 6, pp. 172–175, 1963.

[17] B. Gluss, "An alternative method for continuous line segment curve-fitting", *Information and Control*, 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. Applic.*, vol. 8, pp. 36–52, 1971.

[20] J.G. Dunham, "Optimum uniform piecewise linear approximation of planar curves", *IEEE Trans. PAMI*, vol. PAMI-8, pp. 67–75, Jan. 1986.

[21] Y. Kurozumi and W.A. Davis, "Polygonal approximation by the minimax method", *Computer 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", *Computer Graphics Image Processing*, vol. 1, pp. 244–256, 1972.

[24] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[25] J. Slansky, R.L. Chazin, and B.J. Hansen, "Minimum perimeter polygons of digitized silhouettes", *IEEE Trans. on Computers*, vol. C-21, pp. 260–268, Mar. 1972.

[26] J. Slansky and V. Gonzalez, "Fast polygonal approximation of digitized curves", *Pattern Recognition*, vol. 12, pp. 327–331, 1980.

[27] I. Tomek, "Two algorithms for piecewise linear continuous approximations of functions of one variable", *IEEE Trans. on Computers*, 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. on Computers*, 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. Systems, Man, and Cybernetics*, vol. SMC-12, no. 1, pp. 91–96, 1982.

[30] T. Pavlidis and S.L. Horowitz, "Segmentation of plane curves", *IEEE Trans. on Computers*, vol. C-23, pp. 860–870, Aug. 1974.

[31] T. Pavlidis, "A review of algorithms for shape analysis", *Computer Graphics Image Processing*, vol. 7, pp. 242–258, 1978.

[32] T. Pavlidis, "Algorithms for shape analysis of contours and waveforms", *IEEE T. PAMI*, vol. PAMI-2, pp. 301–312, 1980.

[33] D. Lee and T. Pavlidis, "One-dimensional regularization with discontinuities", *IEEE Trans. PAMI*, vol. PAMI-10, pp. 822–829, Nov. 1988.

[34] J. Vandewalle, "On the calculation of the piecewise linear approximation to a discrete function", *IEEE Trans. on Computers*, vol. C-24, pp. 843–846, 1975.

[35] C.M. Williams, "An efficient algorithm for the piecewise linear approximation of planar curves", *Computer Graphics Image Processing*, vol. 8, pp. 286–293, 1978.

[36] C.M. Williams, "Bounded straight-line approximation of digitized straight curves and lines", *Computer 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. PAMI*, 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, Cybernetics*, vol. SMC-15, pp. 637–641, Sept. 1985.

[40] H. Imai, "Computational-geometric methods for polygonal approximations of a curve", *Computer 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. and Image Processing II*, 1987, pp. 159–166, SPIE Vol. 845.

[42] C.H. Teh and R.T. Chin, "On the detection of dominant points on digital curves", *IEEE Trans. PAMI*, 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. PAMI*, vol. PAMI-11, pp. 967–973, Sept. 1989.

[44] D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities", *IEEE Trans. PAMI*, vol. PAMI-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. PAMI*, vol. PAMI-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. PAMI*, vol. PAMI-17, no. 3, pp. 294–302, Mar. 1995.

[48] D. Terzopoulos, "Regularization of inverse visual problems involving discontinuities", *IEEE Trans. PAMI*, 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. PAMI*, vol. PAMI-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, Sep. 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. Computer Vision and Pattern Recognition*, San Francisco, 1985.

[53] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems", *Communications on Pure and Applied Math.*, vol. 42, pp. 577–685, 1989.

[54] A. Blake and A. Zisserman, *Visual Reconstruction*, MIT Press, Cambridge, Mass., 1987.

[55] Jean-Michel Morel and Sergio Solimini, *Variational Methods in Image Segmentation*, Birkhauser, Boston-Basel-Berlin, 1994.

[56] A.J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", *IEEE Trans. Information Theory*, vol. IT-13, pp. 260–269, Apr. 1967.

[57] J.K. Omura, "On the Viterbi decoding algorithm", *IEEE Trans. Information Theory*, vol. IT-15, pp. 177–179, Jan. 1969.

[58] B. Sklar, *Digital Communications*, Prentice Hall, Englewood Cliffs, NJ, 1988.

[59] 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, Sep. 1993.

[60] P.G. Gulak and E. Shwedyk, "VLSI structures for Viterbi receivers: Part I - general theory and applications", *IEEE J. Selected Areas in Communications*, vol. 4, pp. 142–154, Jan. 1986.

[61] S. Kubota, S. Kato, and T. Ishitani, "Novel Viterbi decoder VLSI implementation and its performance", *IEEE Trans. Communications*, vol. 41, no. 8, pp. 1170–1178, Aug. 1993.

[62] K.K. Parhi, "High-speed VLSI architectures for Huffman and Viterbi decoders", *IEEE Trans. Circuits and Systems II*, vol. 39, no. 6, pp. 385–391, June 1992.

[63] T.K. Truong, M.T. Shih, I.S. Reed, and E.H. Satorius, "A VLSI design for a trace-back Viterbi decoder", *IEEE Trans. Communications*, vol. 40, no. 3, pp. 616–624, Mar. 1992.

[64] R.E. Blahut, *Principles and Practice of Information Theory*, Addison-Wesley, Reading, Mass., 1987.

[65] E.J. Coyle and J.H. Lin, "Optimal stacking filters and mean absolute error nonlinear filtering", *IEEE Trans. ASSP*, vol. ASSP-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. ASSP*, vol. ASSP-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. ASSP*, vol. ASSP-38, no. 6, pp. 955–968, Jun. 1990.

[68] J. Serra, *Image Analysis and Mathematical Morphology*, Academic, New York, 1982.

[69] J. Serra Ed., *Image Analysis and Mathematical Morphology, vol. 2, Theoretical Advances*, Academic, San Diego, 1988.

[70] G. Matheron, *Random Sets and Integral Geometry*, Wiley, New York, 1975.

[71] 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.

[72] N.D. Sidiropoulos, J.S. Baras, and C.A. Berenstein, "An Algebraic Analysis of the Generating Functional for Discrete Random Sets, and Statistical Inference for Intensity in the Discrete Boolean Random Set Model", *Journal of Mathematical Imaging and Vision*, vol. 4, pp. 273–290, 1994.

[73] P. Maragos, *Unified Theory of Translation-Invariant Systems with Applications to Morphological Analysis and Coding of Images*, PhD thesis, School of Electrical Engineering, Georgia Institute of Technology, Atlanta, 1985.

[74] P. Maragos and R. W. Schafer, "Morphological filters - part I : Their set-theoretic analysis and relations to linear shift-invariant filters", *IEEE Trans. on Acoustics, Speech and Signal Proc.*, vol. ASSP-35, no. 8, pp. 1153–1169, Aug. 1987.

[75] P. Maragos and R. W. Schafer, "Morphological filters - part II : Their relations to median, order-statistic, and stack filters", *IEEE Trans. on Acoustics, Speech and Signal Proc.*, vol. ASSP-35, no. 8, pp. 1170–1184, Aug. 1987.

[76] D.C. MacEnany and J.S. Baras, "Scale-space polygonalization of target silhouettes and applications to model-based ATR", in *Proc. Second ATR Systems and Technology Conf., Center for Night Vision and Electro-Optics, Ft. Belvoir, VA*, Mar. 1992, pp. 223–247, Vol. II.

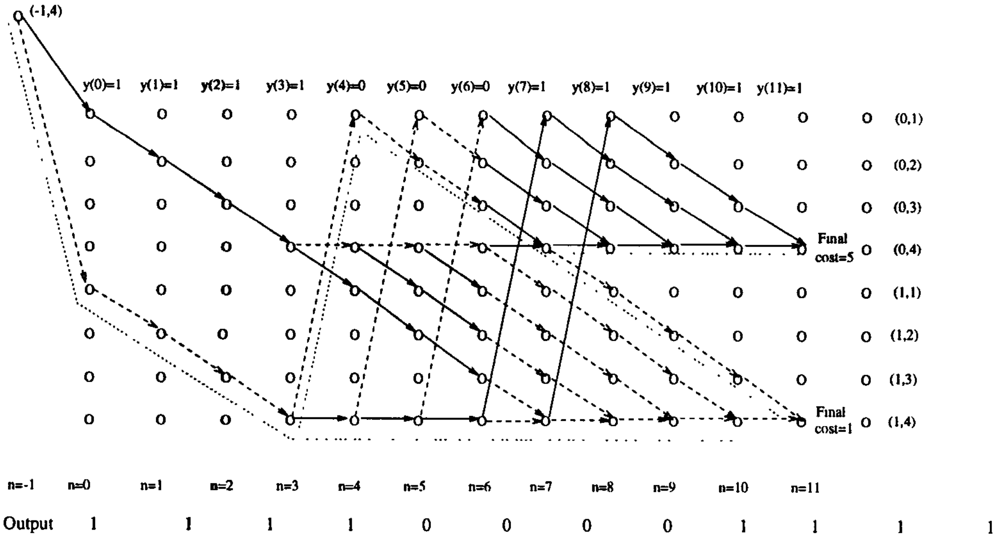
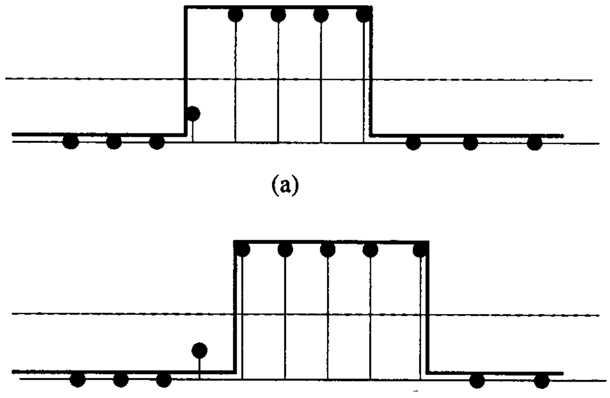


Fig. 1. VORCA trellis


(a)

(b)

Fig. 2. Counter-example 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. The constraint length is $M = 5$. The input in (a) lies below the input in (b), but the same is not true for the corresponding outputs.

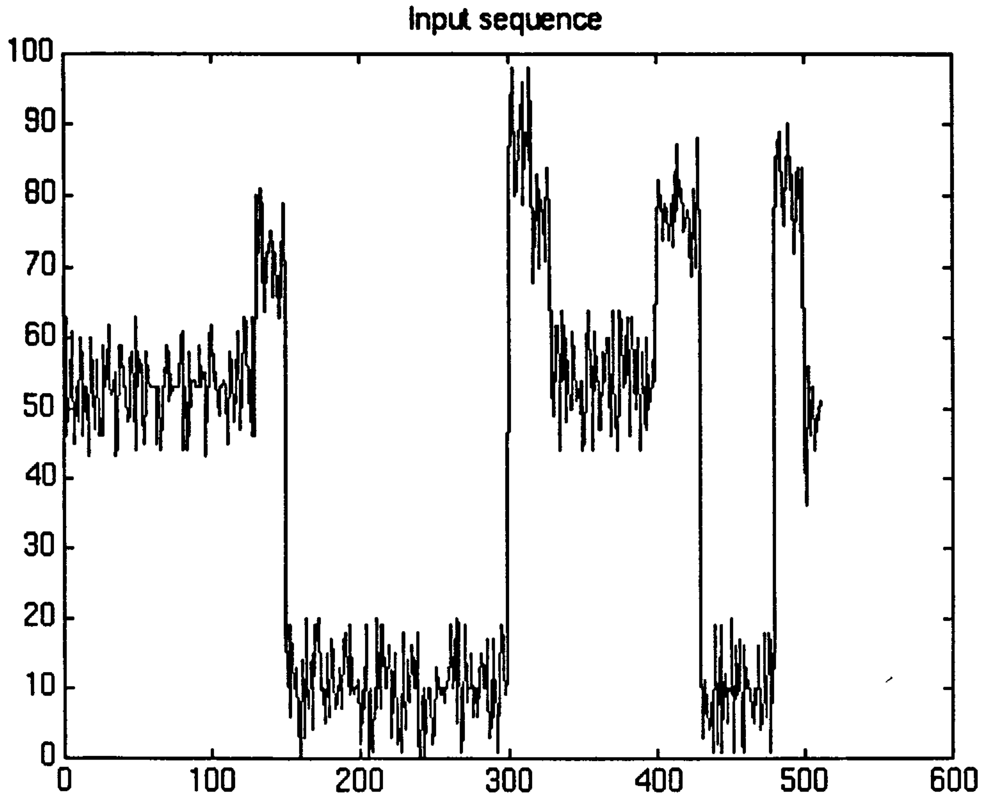


Fig. 3. Input sequence, $\{y(n)\}_{n=0}^{511}$

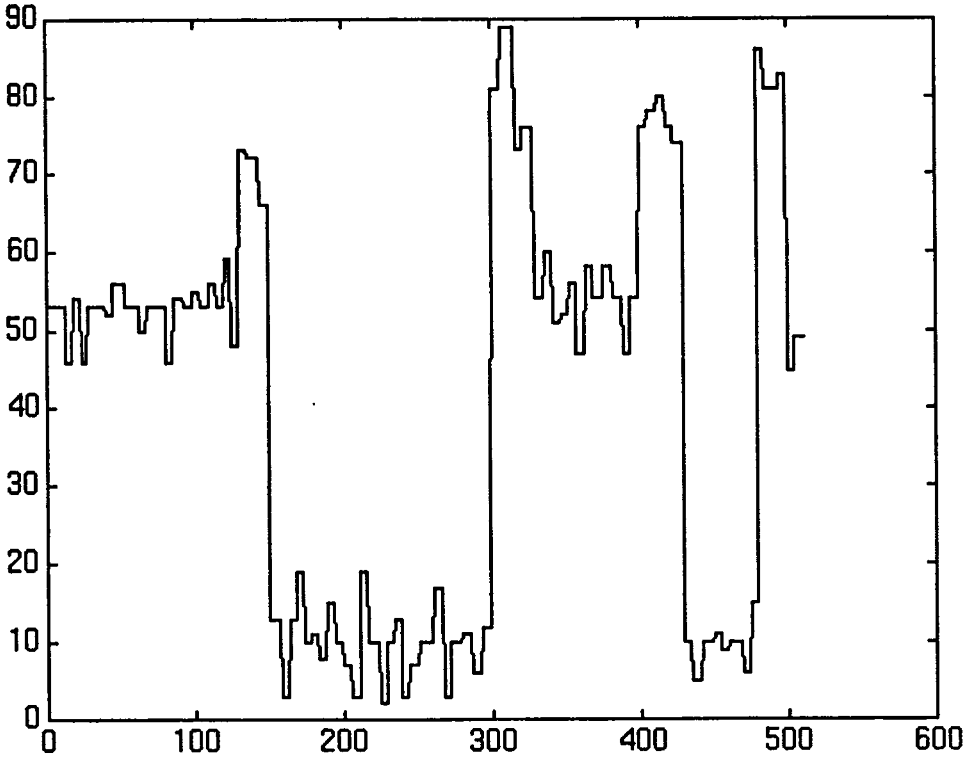


Fig. 4. VORCA output, $M = 5$.

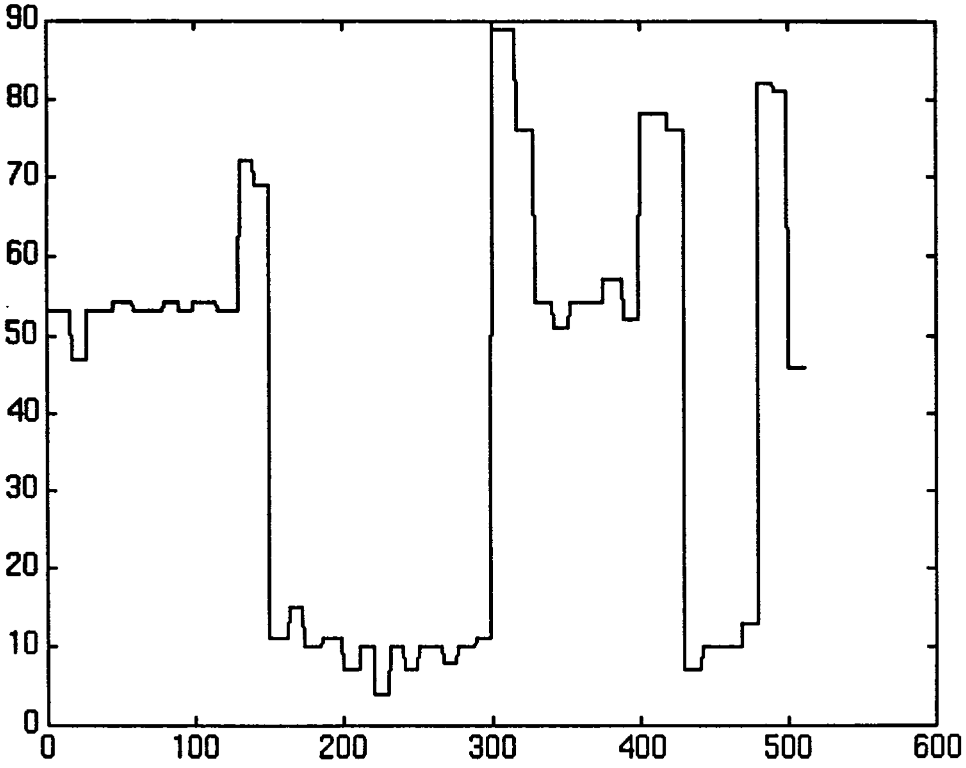


Fig. 5. VORCA output, $M = 10$.

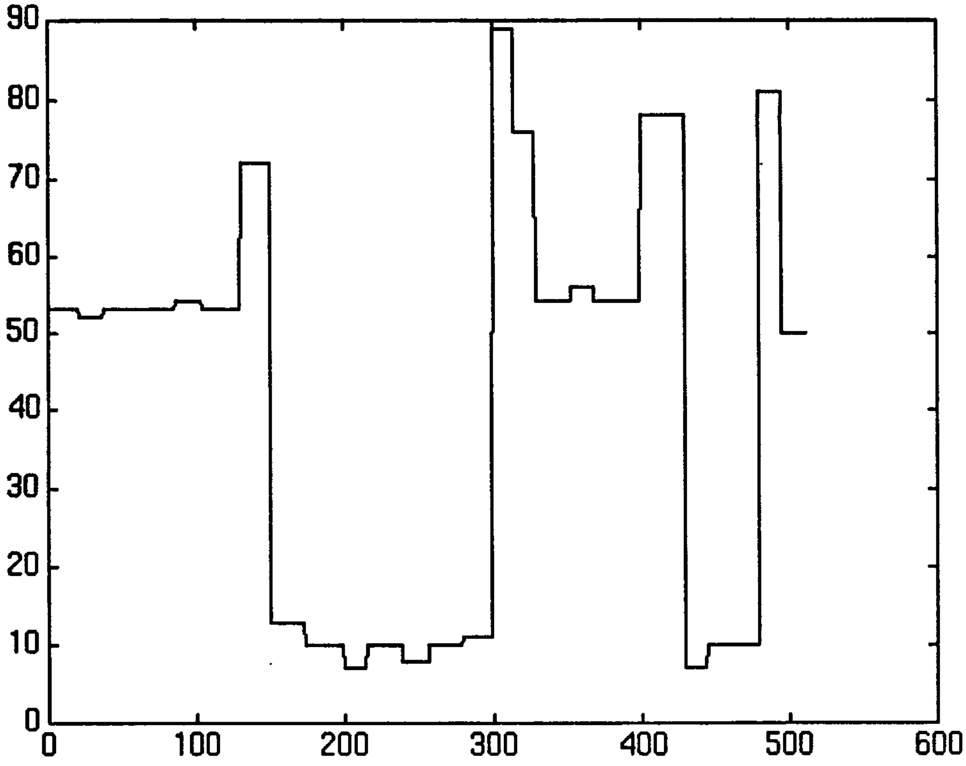


Fig. 6. VORCA output, $M = 15$.

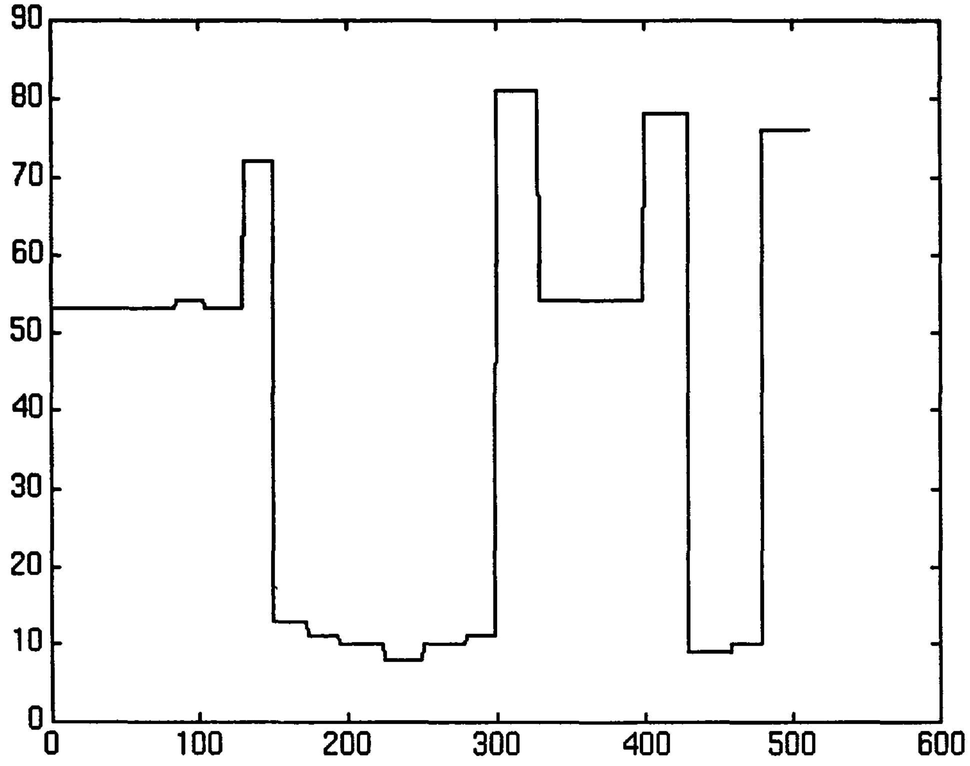


Fig. 7. VORCA output, $M = 20$.

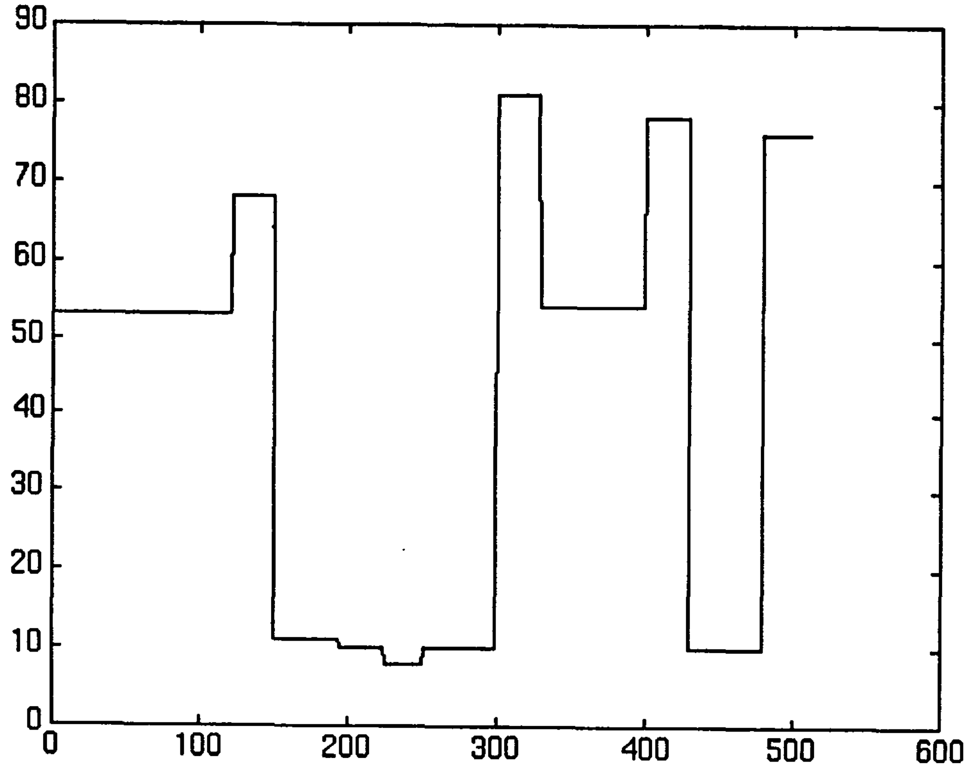


Fig. 8. VORCA output, $M = 25$.

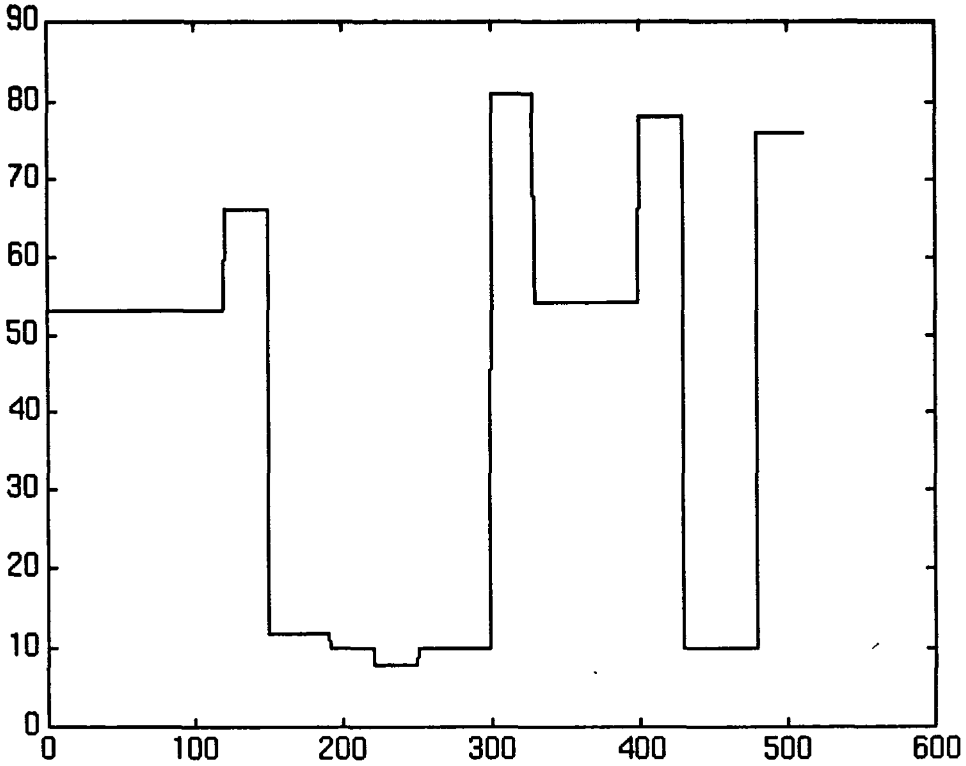


Fig. 9. VORCA output, $M = 30$.

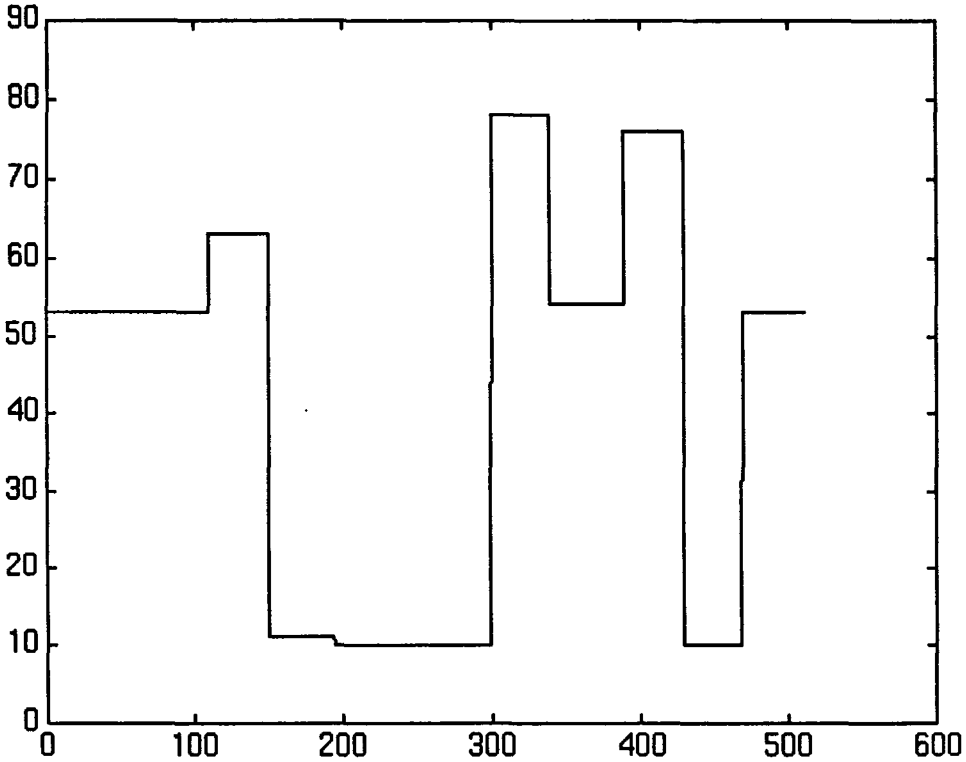


Fig. 10. VORCA output, $M = 40$.

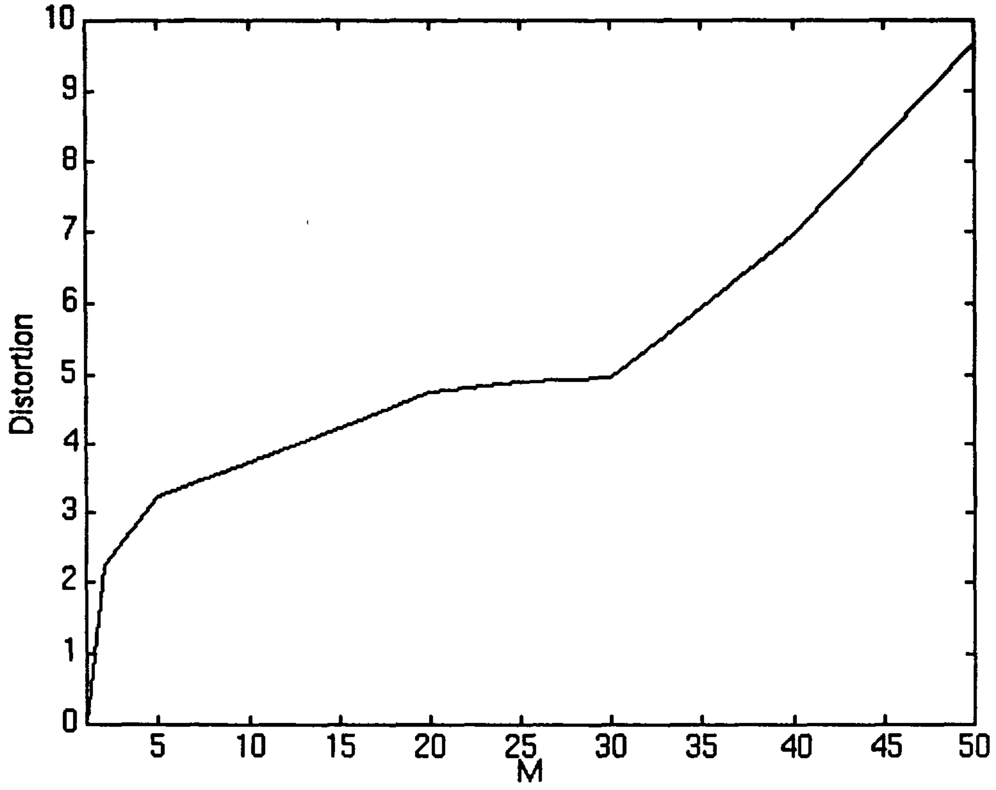


Fig. 11. Plot of average per-letter approximation error as a function of M.

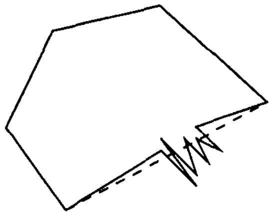


Fig. 12. Optimal polygonalization of shape boundaries: an edge-length constraint is more natural and robust than a constraint on the number of edges

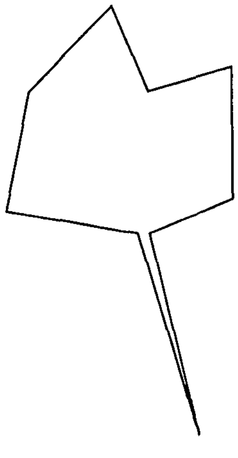


Fig. 13. Is this a "smooth" polygon?

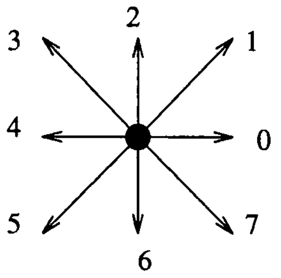


Fig. 14. Possible moves along a simple 8-connected curve