Fast Digital Locally Monotonic Regression

Nicholas D. Sidiropoulos, *Member, IEEE*

***Abstract*—Locally monotonic regression is the optimal counterpart of iterated median filtering. In a previous paper, Restrepo and Bovik developed an elegant mathematical framework in which they studied locally monotonic regressions in $\mathbf{R}^N$. The drawback is that the complexity of their algorithms is exponential in N. In this paper, we consider *digital* locally monotonic regressions, in which the output symbols are drawn from a finite alphabet and, by making a connection to Viterbi decoding, provide a fast $O(|\mathcal{A}|^2\alpha N)$ algorithm that computes any such regression, where $|\mathcal{A}|$ is the size of the digital output alphabet, α stands for lomo degree, and N is sample size. This is *linear* in N, and it renders the technique applicable in practice.**

I. Introduction

LOCAL MONOTONICITY is a property that appears in the study of the set of root signals of the median filter [2]–[8]; it constrains the roughness of a signal by limiting the rate at which the signal undergoes changes of trend (increasing to decreasing or vice versa). In effect, it *limits the frequency of oscillations without limiting the magnitude of jump level changes that the signal exhibits* [1].

A classic problem in the true spirit of nonlinear filtering is the recovery of a piecewise smooth signal embedded in impulsive noise. In this paradigm, it is natural to model the signal as locally monotonic and ask for optimal smoothing under an approximation or estimation criterion. This often amounts to picking a signal from a given class of locally monotonic signals, which minimizes a distortion measure between itself and the observation, and it is referred to as *locally monotonic regression*. In [1], Restrepo and Bovik developed an elegant mathematical framework in which they studied locally monotonic regressions in $\mathbf{R}^N$ (throughout this work, $\mathbf{R}$ denotes the set of real numbers, and $|\cdot|$ stands for set cardinality). Unfortunately, the complexity of their algorithms is exponential in N. The authors admit that their algorithms are computationally very expensive, even for signals of relatively short duration; this hampers potential applications of the method.

Recently, a related nonlinear filtering technique has been proposed [9] that attempts to overcome the complexity of earlier algorithms by considering instead a "soft" constraint formulation in which non-locally monotonic solutions are penalized but not disqualified. This alternative approach is an interesting one, but it addresses a different problem.

Locally monotonic regression provides a median root that is optimal in a suitable sense, e.g., closest to the observable data in some metric or semimetric. It is meant as an "optimal median," while iterating the median may be thought of as a suboptimal "regression" that trades optimality for simplicity. In practice, one usually deals with digital (finite-alphabet) data. If the input (observable data) is finite alphabet, then the output of any number of iterations of the median is also finite-alphabet and, in fact, of the same alphabet as the input; it is therefore natural to consider digital locally monotonic regression in which the output symbols are drawn from a finite alphabet as the optimal counterpart of median filtering of digital signals. Even if the observable data is real valued, one would probably still be interested in digital locally monotonic regression because, on one hand, by proper choice of quantization, it may provide an answer that is sufficiently close to the underlying regression in $\mathbf{R}^N$, and that may well be all that one cares about; on the other hand, it provides a way to perform simultaneous smoothing, quantization, and compression of noisy discontinuous signals. In this paper, we consider digital locally monotonic regression and, by making a connection to Viterbi decoding,[1] provide a fast $O(|\mathcal{A}|^2\alpha N)$ algorithm that computes any such regression, where

- $|\mathcal{A}|$ size of the digital output alphabet
- α lomo degree (usually, the assumed *lomotonicity* of the signal, i.e., the highest degree of local monotonicity that the signal possesses)
- N size of the sample.

This is *linear* (as opposed to *exponential* in the work of Restrepo and Bovik) in N, and it renders the technique applicable in practice.

In more consice terms, we provide a fast $O(|\mathcal{A}|^2\alpha N)$ Viterbi-type algorithm that solves the following problem. Given a sequence of finite extent $\mathbf{y} = \{y(n)\}_{n=0}^{N-1} \in \mathbf{R}^N$, find a finite-alphabet sequence $\hat{\mathbf{x}} = \{\hat{x}(n)\}_{n=0}^{N-1} \in \mathcal{A}^N$ that minimizes $d(\mathbf{x},\mathbf{y}) = \sum_{n=0}^{N-1} d_n(y(n), x(n))$ subject to $\mathbf{x}$ being locally monotonic of degree α.

A. Organization

The rest of this paper is structured as follows. In Section II, we provide some necessary definitions and a formal statement of the problem. The reader is referred to [1] and [11] and the references therein for additional background and motivation. Our fast solution is presented in Section III. A discussion on

Manuscript received September 8, 1995; revised July 26, 1996. This work was supported in part by the NSF ERC program 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. The associate editor coordinating the review of this paper and approving it for publication was Dr. A. Lee Swindlehurst.

The author is with the Institute for Systems Research, University of Maryland, College Park, MD 20742 USA (e-mail: nikos@glue.umd.edu).

Publisher Item Identifier S 1053-587X(97)01169-0.

[1] Such a connection between optimal nonlinear filtering under local syntactic constraints and Viterbi decoding algorithms has first been made in [10].

implementation complexity is also included. Some properties of locally monotonic regression are discussed in Section IV. A complete simulation experiment is presented in Section V, and conclusions are drawn in Section VI.

II. The Problem

A. Background

If $\mathbf{x}$ is a real-valued sequence (string) of length N and γ is any integer less than or equal to N, then a *segment* of $\mathbf{x}$ of length γ is any substring of γ consecutive components of $\mathbf{x}$. Let $\mathbf{x}_i^{i+\gamma-1} = \{x(i), \ldots, x(i+\gamma-1)\}$, $i \geq 0$, $i+\gamma \leq N$ be any such segment. $\mathbf{x}_i^{i+\gamma-1}$ is monotonic if either $x(i) \leq x(i+1) \leq \cdots \leq x(i+\gamma-1)$ or $x(i) \geq x(i+1) \geq \cdots \geq x(i+\gamma-1)$.

Definition 1: A real-valued sequence $\mathbf{x}$ of length N is *locally monotonic* of degree $\alpha \leq N$ (or *lomo*-α or simply *lomo* in case α is understood) if each and every one of its segments of length α is monotonic.

Throughout the following, we assume that $3 \leq \alpha \leq N$. A sequence $\mathbf{x}$ is said to exhibit an increasing (resp. decreasing) transition at coordinate i if $x(i) < x(i+1)$ (resp. $x(i) > x(i+1)$).

The following property (cf. [1]–[3]) is key in the subsequent development of our fast algorithm. If $\mathbf{x}$ is locally monotonic of degree α, then $\mathbf{x}$ has a constant segment (run of identical symbols) of length at least $\alpha - 1$ in between an increasing and a decreasing transition. The reverse is also true.

If $3 \leq \alpha \leq \beta \leq N$, then a sequence of length N that is lomo β is lomo α as well; thus, the *lomotonicity* of a sequence is defined as the highest degree of local monotonicity that it possesses [1].

An interesting property of locally monotonic regression is that it admits a maximum likelihood (ML) interpretation [1], [11]. In particular, if one chooses $d_n(y(n), x(n)) = -\log p_n(y(n) - x(n))$, where $p_n(\cdot)$ is the (independent) additive noise pdf or pmf, then locally monotonic regression of degree α may be viewed as ML over the set of all locally monotonic signals of degree α embedded in additive independent (yet not necessarily identically distributed) noise [1], [11]. This means that one may *adapt* the regression to the noise characteristics: locally monotonic regression is much more flexible than the median.

B. Digital Locally Monotonic Regression

Given $y(n) \in \mathbf{R}$, $n = 0, 1, \ldots, N-1$ and $\mathcal{A}$, which is a finite subset of $\mathbf{R}(|\mathcal{A}| < \infty)$. Let $\Lambda(\alpha, N, \mathcal{A})$ denote the space of all sequences of N elements of $\mathcal{A}$ that are locally monotonic of degree α. Digital locally monotonic regression is the following constrained optimization:

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

$$\text{subject to:} \quad \mathbf{x} = \{x(n)\}_{n=0}^{N-1} \in \Lambda(\alpha, N, \mathcal{A}). \tag{2}$$

Here, $d_n(\cdot, \cdot)$ is any per-letter distortion measure; it can be a—possibly inhomogeneous in n—metric, semimetric, or arbitrary bounded per-letter cost measure. The "sum" may also be interpreted liberally; it turns out that it can be replaced by a "max" operation to accommodate a minimax (minimize sup-error) problem formulation without affecting the structure of the fast computational algorithm, which is developed below.

Observe that if $3 \leq \alpha \leq \beta \leq N$, then $\Lambda(\beta, N, \mathcal{A}) \subseteq \Lambda(\alpha, N, \mathcal{A})$; thus, the above optimization is defined over an element of a sequence of nested "approximation" spaces. This means that the achievable minimum is a nondecreasing function of α.

III. Solution

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

Definition 2: Given any sequence $\mathbf{x} = \{x(n)\}_{n=0}^{N-1}$, $x(n) \in \mathcal{A}, n = 0, 1, \ldots, N-1$, define its associated *state sequence* $\mathbf{s_x} = \{[x(n), l_\mathbf{x}(n)]^T\}_{n=-1}^{N-1}$, where $[x(-1), l_\mathbf{x}(-1)]^T = [\phi, \alpha - 1]^T, \phi \in \mathcal{A}$ and, for $n = -1, \ldots, N-2$, $l_\mathbf{x}(n+1)$ is given by

$$\begin{cases} \text{sgn}(l_\mathbf{x}(n)) \cdot \min\{\text{abs}(l_\mathbf{x}(n)) + 1, \alpha - 1\} \\ \quad , x(n+1) = x(n) \\ 1, \quad x(n+1) > x(n) \\ -1, x(n+1) < x(n) \end{cases}$$

where sgn$(\cdot)$ stands for the sign function, and abs$(\cdot)$ stands for absolute value. $[x(n), l_\mathbf{x}(n)]^T$ is the state at time n, and, for $n = 0, 1, \ldots, N-1$, it takes values in $\mathcal{A} \times \{-(\alpha - 1), \ldots, -1, 1, \ldots, \alpha - 1\}$.

Clearly, we can equivalently pose the optimization (1), (2) in terms of the associated state sequence.

Definition 3: A subsequence of state variables $\{[x(n), l_\mathbf{x}(n)]^T\}_{n=-1}^{\nu}$, $\nu \leq N-1$, is *admissible* (with respect to constraint (2)) if and only if there exists a suffix string of state variables $\{[x(n), l_\mathbf{x}(n)]^T\}_{n=\nu+1}^{N-1}$ such that $\{[x(n), l_\mathbf{x}(n)]^T\}_{n=-1}^{\nu}$ followed by $\{[x(n), l_\mathbf{x}(n)]^T\}_{n=\nu+1}^{N-1}$ is the associated state sequence of some sequence in $\Lambda(\alpha, N, \mathcal{A})$.

Let $\hat{\mathbf{x}} = \{\hat{x}(n)\}_{n=0}^{N-1}$ be a solution (one always exists, although it may not necessarily be unique) of (1) and (2), and let $\{[\hat{x}(n), l_{\hat{\mathbf{x}}}(n)]^T\}_{n=-1}^{N-1}$ be its associated state sequence. Clearly, $\{[\hat{x}(n), l_{\hat{\mathbf{x}}}(n)]^T\}_{n=-1}^{N-1}$ is admissible, and so is any subsequence $\{[\hat{x}(n), l_{\hat{\mathbf{x}}}(n)]^T\}_{n=-1}^{\nu}$, $\nu \leq N-1$. The following is a key observation.

Claim 1: Optimality of $\{[\hat{x}(n), l_{\hat{\mathbf{x}}}(n)]^T\}_{n=-1}^{N-1}$ implies optimality of $\{[\hat{x}(n), l_{\hat{\mathbf{x}}}(n)]^T\}_{n=-1}^{\nu}$, $\nu \leq N-1$, among all admissible subsequences of the same length that lead to the same state at time ν, i.e., all admissible $\{[\tilde{x}(n), l_{\tilde{\mathbf{x}}}(n)]^T\}_{n=-1}^{\nu}$ 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 $\{[\tilde{x}(n), l_{\tilde{\mathbf{x}}}(n)]^T\}_{n=-1}^{\nu}$ is an admissible subsequence 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 $\{[\tilde{x}(n), l_{\tilde{\mathbf{x}}}(n)]^T\}_{n=-1}^{\nu}$ followed by $\{[\hat{x}(n), l_{\hat{\mathbf{x}}}(n)]^T\}_{n=\nu+1}^{N-1}$ is also admissible. The key point is that any suffix string of state variables making $\{[\hat{x}(n), l_{\hat{\mathbf{x}}}(n)]^T\}_{n=-1}^{\nu}$ admissible will also make $\{[\tilde{x}(n), l_{\tilde{\mathbf{x}}}(n)]^T\}_{n=-1}^{\nu}$ admissible. If $\{[\tilde{x}(n), l_{\tilde{\mathbf{x}}}(n)]^T\}_{n=-1}^{\nu}$ has a smaller cost (distortion) than $\{[\hat{x}(n), l_{\hat{\mathbf{x}}}(n)]^T\}_{n=-1}^{\nu}$, then by virtue of the fact that the cost is a sum of per-letter costs, $\{[\tilde{x}(n), l_{\tilde{\mathbf{x}}}(n)]^T\}_{n=-1}^{\nu}$ followed

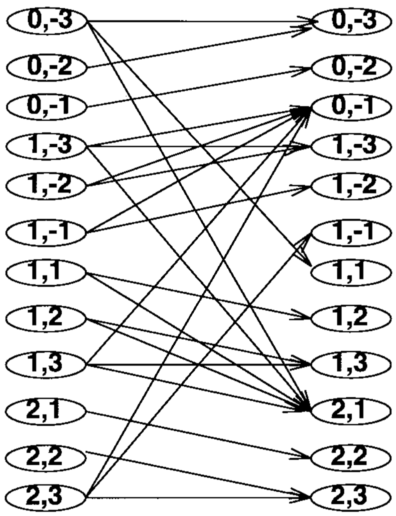


Fig. 1. Two stages of the resulting trellis for $\alpha = 4$, $\mathcal{A} = \{0, 1, 2\}$. Some states are unreachable and, therefore, not shown. Absence of an arrow indicates infinite transition cost; otherwise, the transition cost is the distance of the first variable of the receiving state from the corresponding observed symbol at this stage. Observe that the graph is sparse and regular.

by $\{[\hat{x}(n), l_{\hat{\mathbf{x}}}(n)]^T\}_{n=\nu+1}^{N-1}$ will have a smaller cost than $\{[\hat{x}(n), l_{\hat{\mathbf{x}}}(n)]^T\}_{n=-1}^{N-1}$, and this violates the optimality of the latter. □

This is a particular instance of the principle of optimality of dynamic programming [12]–[14]. 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 [15]–[17] algorithmic implementation of any digital locally monotonic regression. The costs associated with one-step state transitions need to be specified in a way that forces one-step optimality and admissibility. This specification appears in the Appendix. A formal proof can be easily constructed and is hereby omitted. C-code is available at

$$\texttt{http://www.glue.umd.edu/\~{}nikos}.$$

A simple example of the structure and connectivity of two stages of the resulting trellis is given in Fig. 1. Observe that the trellis is sparse and regular. As explained below, this fact is exploited to reduce implementation complexity.

A. Complexity

Any Viterbi-type algorithm has computational complexity that is *linear* in the number of observations, i.e., N. The number of computations per observation symbol depends on the number of states as well as state connectivity in the trellis. In the following, we derive the required number of distance (branch metric) calculations and additions per observation symbol (trellis stage) (the number of comparisons required per trellis stage is always less than this number). Each stage in the trellis has a total of $|\mathcal{A}|2(\alpha-1)$ states, which can be classified as follows:

- $|\mathcal{A}|$ state pairs of the form $([v,-1]^T, [v,1]^T), v \in \mathcal{A}$. One can easily check that the combined fun-in of each such pair (i.e., the number of states at the previous time instant from which such a pair can be reached) is $(|\mathcal{A}|-1)\alpha$. Thus, one needs $(|\mathcal{A}|-1)\alpha$ distance calculations and additions per pair, for a subtotal of $|\mathcal{A}|(|\mathcal{A}|-1)\alpha$ distance calculations and additions per stage, for this class of states.
- $|\mathcal{A}|2(\alpha-3)$ states of the form $[v,l]^T, v \in \mathcal{A}$, $1 < l < \alpha - 1$, or $-(\alpha-1) < l < -1$. Each such state can only be reached by one state, namely, $[v, l-1]^T$ if $l > 0$ or $[v, l+1]^T$ otherwise. Thus, one needs $|\mathcal{A}|2(\alpha-3)$ distance calculations and additions per stage for this class of states.
- $|\mathcal{A}|$ state pairs of the form $([v, -(\alpha-1)]^T, [v, \alpha-1]^T), v \in \mathcal{A}$. One can easily check that the combined fun-in of each such pair is 4. Indeed, a state of type $[v, \alpha-1]^T$ can only be reached from either itself or $[v, (\alpha-1)-1]^T$, and similarly, a state of type $[v, -(\alpha-1)]^T$ can only be reached from either itself or $[v, -(\alpha-1)+1]^T$. Therefore, one needs $4|\mathcal{A}|$ distance calculations and additions per stage for this class of states.

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

	$\alpha=5$	$\alpha=10$	$\alpha=15$	$\alpha=20$	$\alpha=25$	$\alpha=30$
$\|\mathcal{A}\|=2$	26	56	86	116	146	176
$\|\mathcal{A}\|=16$	1328	2688	4048	5408	6768	8128
$\|\mathcal{A}\|=32$	5216	10496	15776	21056	26336	31616
$\|\mathcal{A}\|=64$	20672	41472	62272	83072	103872	124672
$\|\mathcal{A}\|=128$	82304	164864	247424	329984	412544	495104
$\|\mathcal{A}\|=256$	328448	657408	986368	1315328	1644288	1973248

The total is $|\mathcal{A}|^2\alpha + |\mathcal{A}|(\alpha-2)$ distance calculations and additions per stage; this is tabulated in Table I for some typical parameter values, and it is of $O(|\mathcal{A}|^2\alpha)$, for a grand total of $O(|\mathcal{A}|^2\alpha N)$ for the entire regression. Clearly, $|\mathcal{A}|$ (i.e., the size of the output alphabet) is the dominating factor.

The worst-case storage requirements of digital locally monotonic regression are $O(|\mathcal{A}|\alpha N)$, but actual storage requirements are much more modest due to path merging.

Computational complexity being $O(|\mathcal{A}|^2\alpha N)$ means (as we will soon see in the simulation section) that in a serial software implementation, one may obtain an exact optimal solution in the order of a couple of minutes for long observation sequences. In addition, the algorithm, being a Viterbi-type technique, has strong potential for hardware implementation. The availability of VLSI Viterbi decoding chips, as well as several dedicated multiprocessor architectures for Viterbi-type decoding, makes fast digital locally monotonic regression a realistic alternative to standard nonlinear (e.g., median) filtering, at least for moderate values of $|\mathcal{A}|, \alpha$. In the binary case, current Viterbi technology [18]–[23] can handle 2^{12} states. Hardware capability is continuously improving at a rather healthy pace. Viterbi-type filtering techniques, like the one described here, will certainly benefit from these developments.

IV. Some Properties of Locally Monotonic Regression

From the viewpoint of nonlinear filtering theory, digital locally monotonic regression is not technically a filter due to the possibility of multiple minima. However, all these minima are equivalent in terms of distortion cost, and it is standard practice in Viterbi decoding to invoke some tie-breaking strategy to obtain a unique solution. This way, we also obtain a unique input/output operator, and we may refer to digital locally monotonic regression as a *filter*. From a traditional nonlinear filtering perspective, it is of interest to investigate whether this filter is *idempotent* (converges to a fixed point in one step) [24], *self-dual* (in the binary case, it treats an "object" and its "background" in a balanced fashion) [24], and/or *increasing* (order-preserving) [24]. The median is self-dual and increasing but not idempotent. Idempotence is obviously a desirable property.[2] Self-duality is usually desirable. The increasing property facilitates mathematical analysis, yet it may often be hard to justify. One may easily show (along the lines of [10]) the following proposition:

Proposition 1: If $d_n(\cdot,\cdot)$ is a distance metric, then digital locally monotonic regression is idempotent. The result is also true under the relaxed condition that $\forall n \in \{0, 1, \ldots, N-1\}$, $d_n(\cdot,\cdot)$ achieves its minimum value if and only if its arguments are equal.

Proposition 2: If $d_n(y,x) = d_n(|y-x|), n = 0, 1, \ldots, N-1, \forall y, x$, then, without loss of optimality, digital locally monotonic regression can be designed to be self-dual by means of special choice of tie-breaking strategy [10]. In particular, the result holds for l_1, l_2 distance metrics.

However, the most interesting observation has to do with whether or not digital locally monotonic regression is increasing. To see this, it is convenient to reproduce a few definitions.

Definition 4: $\mathbf{y}_1 \leq \mathbf{y}_2$ if and only if $y_1(n) \leq y_2(n),\ \forall n \in \{0, 1, \ldots, N-1\}$.

Definition 5: A filter f is *increasing* if and only if $\mathbf{y}_1 \leq \mathbf{y}_2 \Longrightarrow f(\mathbf{y}_1) \leq f(\mathbf{y}_2),\ \forall \mathbf{y}_1, \mathbf{y}_2 \in \mathbf{R}^N$.

Proposition 3: Regardless of choice of tie-breaking strategy, digital locally monotonic regression is not increasing.

Proof: Counter-examples can be constructed for binary variables, in which case, a signal is locally monotonic of degree α if and only if it is piecewise-constant and the length of its smallest piece is greater than or equal to $\alpha - 1$. Such a counterexample (for $\alpha = 6$) can be found in [25]. □

Therefore, under mild conditions, digital locally monotonic regression is idempotent and self-dual but not increasing.

V. Simulation

An experiment with a real human ECG signal is given in Figs. 2 and 3. Fig. 2 depicts a portion of a human ECG signal from the Signal Processing Information Base (SPIB) at spib.rice.edu. Fig. 3 depicts the result of locally monotonic regression under the l_1 distance and for $\alpha = 5$.

[2]Note that although the median is not idempotent, the median root is.

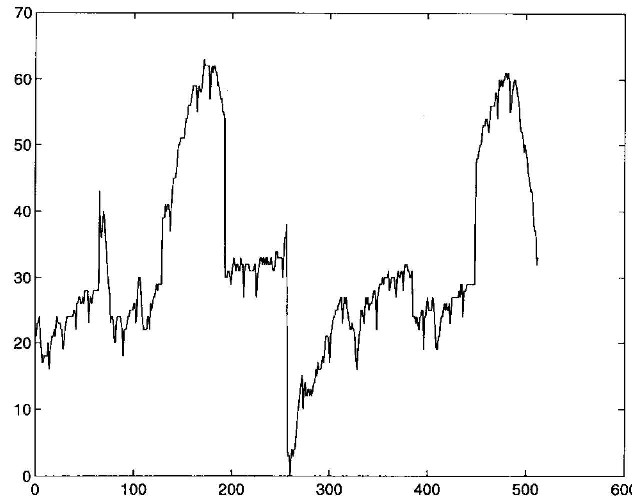


Fig. 2. Portion of human ECG from the signal processing information base.

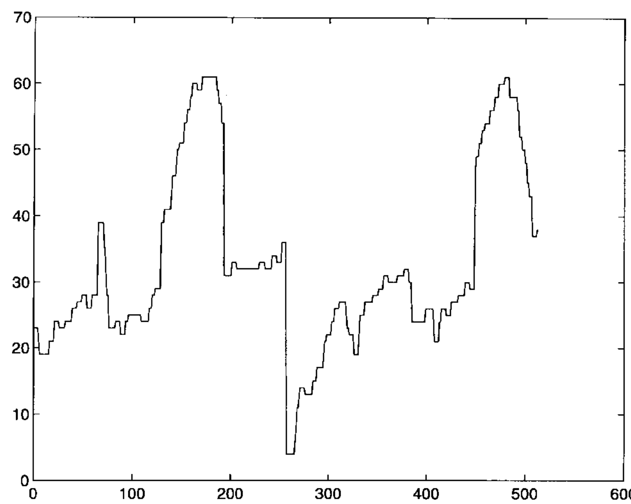


Fig. 3. Output of digital locally monotonic regression of degree $\alpha = 5$.

Nonlinear smoothing of edge signals embedded in noise is one of the prime applications of median-type filtering. Therefore, it is of interest to present simulation results on locally monotonic regression applied to synthetic noisy edge-ramp signals. Fig. 5 depicts such an input signal. This particular signal has been generated by adding i.i.d. noise on synthetic "true" noise-free test data, which is depicted in Fig. 4. Observe that the noise-free test data is almost everywhere locally monotonic up to a certain degree but not purely locally monotonic; therefore, the true signal itself will suffer some distortion when subjected to locally monotonic regression. As noted earlier, the degree of this distortion is an increasing function of α.

The noise has been generated according to a uniform distribution, and most of the data points are contaminated. Our goal here is to present a balanced experiment that is not overly in favor of the approach; thus, we do not use our prior knowledge of the noise model to match the regression to the noise characteristics, which is certainly a possibility (cf. [1], [11] and our earlier discussion. By proper choice of $d_n(\cdot,\cdot)$, locally monotonic regression can be tailored to provide maximum

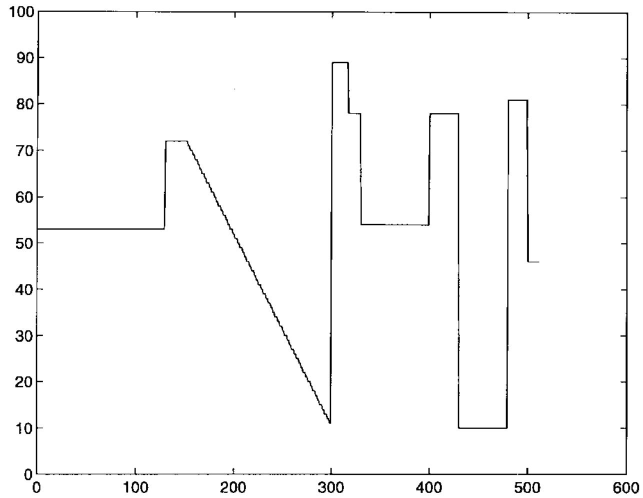


Fig. 4. "True" noise-free test data.

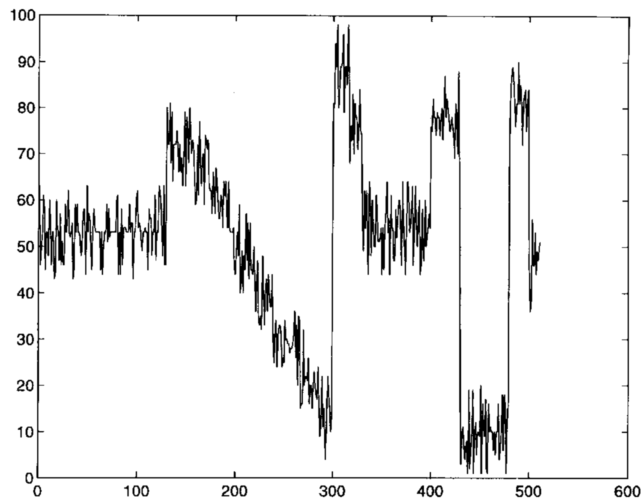


Fig. 5. Input sequence $\{y(n)\}_{n=0}^{511}$.

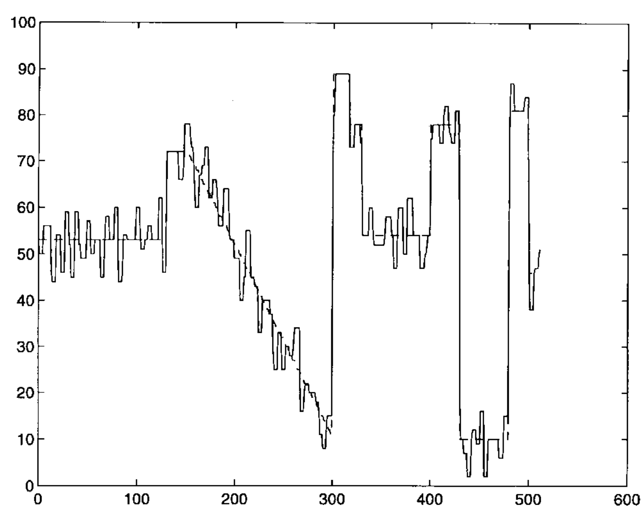


Fig. 6. Output of digital locally monotonic regression of degree $\alpha = 5$.

likelihood (ML) estimates). This is consistent in spirit with our earlier choice of noise-free test signal, which is not purely locally monotonic, and the fact that in practice, one rarely

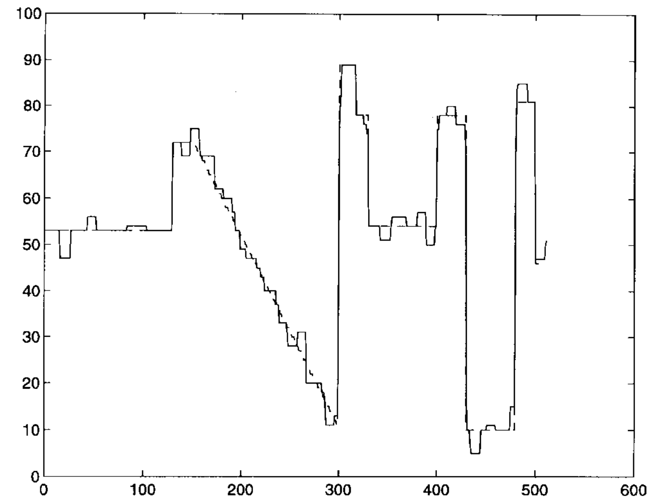


Fig. 7. Output of digital locally monotonic regression of degree $\alpha = 10$.

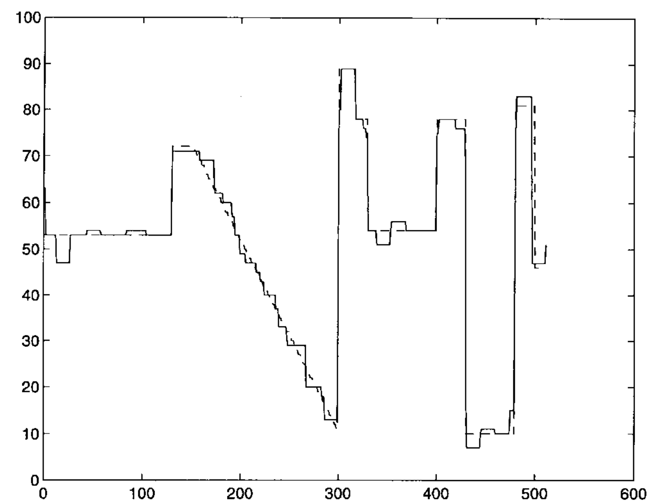


Fig. 8. Output of digital locally monotonic regression of degree $\alpha = 15$.

has complete knowledge of noise statistics, and therefore, the user community will probably opt for using e.g., tried-and-true l_1, l_2 distance metrics.

The noise-free test data of Fig. 4 is also overlaid on subsequent plots. This is meant to help the reader judge filtering quality. For this example, we blindly choose $d_n(y(n), x(n)) = |y(n) - x(n)|$, $\forall n \in \{0, 1, \ldots, N-1\}$, $\mathcal{A} = \{0, \ldots, 99\}$, and $N = 512$. The resulting optimal approximation for $\alpha = 5, 10, 15, 20$, and 25 is depicted in Figs. 6–10, respectively. The results are very good. The overall run time is approximately equal to 2 min for $\alpha = 15$, $N = 512$, $|\mathcal{A}| = 100$, on a SUN SPARC 10, using simple C-code developed by the author. Much better benchmarks may be expected for smaller alphabets and/or by implementing the algorithm in dedicated Viterbi hardware, e.g., for $|\mathcal{A}| = 32$ and everything else as above, the overall run time is approximately 12 s for a throughput of 42 32-ary symbols/s.

VI. Conclusion and Further Research

Motivated in part by the work of Restrepo and Bovik [1], our own earlier work in [10], and the fact that in

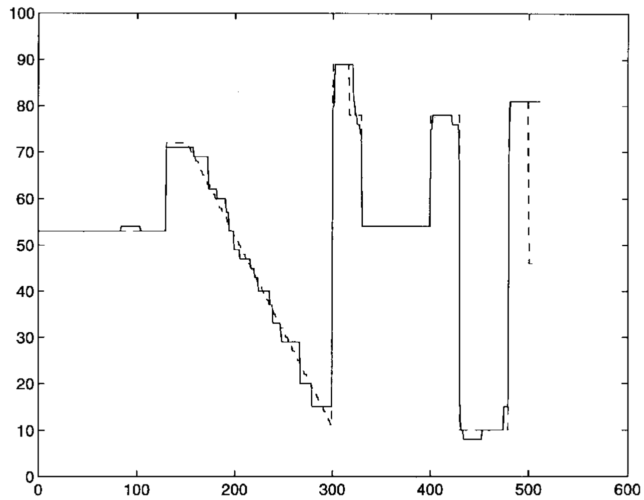


Fig. 9. Output of digital locally monotonic regression of degree $\alpha = 20$.

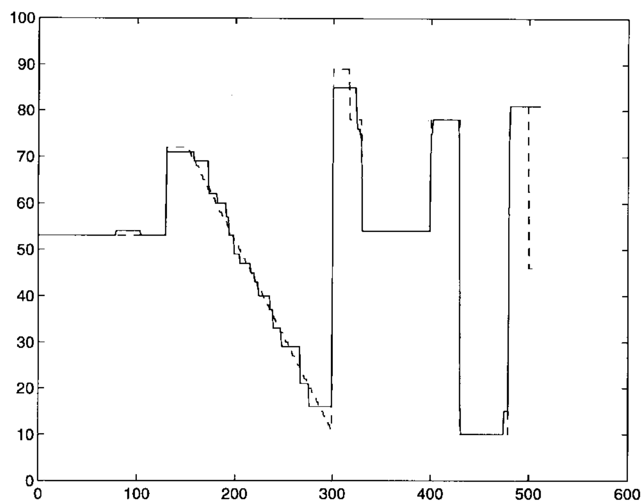


Fig. 10. Output of digital locally monotonic regression of degree $\alpha = 25$.

practice, one usually deals with digital (finite-alphabet) data, we have posed the problem of digital locally monotonic regression in which the output symbols are drawn from a finite alphabet as a natural optimal counterpart of median filtering of digital signals. Capitalizing on a connection between optimal nonlinear filtering under local syntactic constraints and Viterbi decoding algorithms, which has first been made in [10], we have provided a fast $O(|\mathcal{A}|^2 \alpha N)$ algorithm that computes any such regression, where

- $|\mathcal{A}|$ size of the digital output alphabet
- α lomo-degree
- N sample size.

This is *linear* (as opposed to *exponential* in the work of Restrepo and Bovik) in N, and it renders the technique applicable in practice.

The connection between optimal nonlinear filtering under local syntactic constraints and Viterbi decoding algorithms seems to be strong and pervasive; it appears to provide a unifying framework for the efficient computation of a rich class of nonlinear filtering techniques, some of which were oftentimes deemed impractical, due to their complexity. This key element certainly deserves further investigation, and several threads are currently being pursued.

Appendix
Specification of One-Step State Transition Costs for Digital Locally Monotonic Regression

Here, $c(s_{\mathbf{x}}(n) \rightarrow s_{\mathbf{x}}(n+1))$ denotes the cost of a one-step state transition, $s_{\mathbf{x}}(n) = [x(n), l_{\mathbf{x}}(n)]^T$, and $\vee, \wedge$ denote logical OR and AND, respectively. The required specification follows:

if :

$$(l_{\mathbf{x}}(n+1) = 1) \wedge (x(n) < x(n+1)) \wedge [(l_{\mathbf{x}}(n) > 0) \vee (l_{\mathbf{x}}(n) = -(\alpha - 1))]$$

/* To make an increasing transition, one of two things must hold: **either** you're currently in the midst of an increasing trend, **or**, if in the midst of a decreasing trend, you've just completed a constant run of at least $\alpha - 1$ symbols following the latest decreasing transition. */

$$\vee$$

$$(l_{\mathbf{x}}(n+1) = -1) \wedge (x(n) > x(n+1)) \wedge [(l_{\mathbf{x}}(n) < 0) \vee (l_{\mathbf{x}}(n) = \alpha - 1)]$$

/* Similarly, to make a decreasing transition, one of two things must hold: **either** you're currently in the midst of a decreasing trend, **or**, if in the midst of an increasing trend, you've just completed a constant run of at least $\alpha - 1$ symbols following the latest increasing transition. */

$$\vee$$

$$(1 < l_{\mathbf{x}}(n+1) < \alpha - 1) \wedge (x(n) = x(n+1)) \wedge (l_{\mathbf{x}}(n+1) = l_{\mathbf{x}}(n) + 1)$$

/* If you are in a constant run following an increasing transition, and you receive one more identical symbol, then the only thing you are allowed to do is increment your counter */

$$\vee$$

$$(-(\alpha - 1) < l_{\mathbf{x}}(n+1) < -1) \wedge (x(n) = x(n+1)) \wedge (l_{\mathbf{x}}(n+1) = l_{\mathbf{x}}(n) - 1)$$

/* Similarly, if you are in a constant run following a decreasing transition, and you receive one more identical symbol, then the only thing you are allowed to do is decrement your counter */

$$\vee$$

$$(l_{\mathbf{x}}(n+1) = \alpha - 1) \wedge (x(n) = x(n+1)) \wedge [(l_{\mathbf{x}}(n) = \alpha - 1) \vee (l_{\mathbf{x}}(n) = (\alpha - 1) - 1)]$$

/* The only way you can reach a positive full count of $\alpha - 1$ is to either have a positive full count or be just one sample short of a positive full count **and** receive one more identical symbol */

$$\vee$$

$$(l_{\mathbf{x}}(n+1) = -(\alpha - 1)) \wedge (x(n) = x(n+1)) \wedge [(l_{\mathbf{x}}(n) = -(\alpha - 1)) \vee (l_{\mathbf{x}}(n) = -(\alpha - 1) + 1)]$$

/* The only way you can reach a negative full count of $-(\alpha - 1)$ is to either have a negative full count or be just one sample short of a negative full count **and** receive one more identical symbol */

$$\begin{aligned}\textbf{then}: & c\big([x(n), l_{\mathbf{x}}(n)]^T \rightarrow [x(n+1), l_{\mathbf{x}}(n+1)]^T\big) \\ & = d_{n+1}(y(n+1), x(n+1)) \\ \textbf{else}: & c\big([x(n), l_{\mathbf{x}}(n)]^T \rightarrow [x(n+1), l_{\mathbf{x}}(n+1)]^T\big) \\ & = \infty. \end{aligned} \tag{3}$$

REFERENCES

[1] A. Restrepo and A. C. Bovik, "Locally monotonic regression," *IEEE Trans. Signal Processing*, vol. 41, pp. 2796–2810, Sept. 1993.

[2] S. G. Tyan, "Median filtering: Deterministic properties," in *Two-Dimensional Digital Signal Processing II: Transforms and Median Filters*, T. S. Huang, Ed. Berlin: Springer-Verlag, 1981, pp. 197–217.

[3] 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.

[4] 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.

[5] T. A. Nodes and N. C. Gallagher, "Median filters: Some modifications and their properties," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 739–746, 1982.

[6] N. C. Gallagher Jr., "Median filters: A tutorial," in *Proc. IEEE Int. Symp. Circ., Syst.*, 1988, vol. ISCAS-88, pp. 1737–1744.

[7] A. C. Bovik, T. S. Huang, and D. C. Munson, "A generalization of median filtering using linear combinations of order statistics," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1342–1349, 1983.

[8] ——, "The effect of median filtering on edge estimation and detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 181–194, Mar. 1987.

[9] S. T. Acton and A. C. Bovik, "Nonlinear image estimation using piecewise and local image models," *IEEE Trans. Image Processing*, submitted for publication.

[10] N. D. Sidiropoulos, "The viterbi optimal runlength-constrained approximation nonlinear filter," *IEEE Trans. Signal Processing*, vol. 44, pp. 586–598, Mar. 1996.

[11] A. Restrepo and A. C. Bovik, "Statistical optimality of locally monotonic regression," *IEEE Trans. Signal Processing*, vol. 42, pp. 1548–1550, June 1994.

[12] R. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton Univ. Press, 1957.

[13] R. Bellman and S. Dreyfus, *Applied Dynamic Programming*. Princeton, NJ: Princeton Univ. Press, 1962.

[14] S. Dreyfus and A. Law, *The Art and Theory of Dynamic Programming*. New York: Academic: 1977.

[15] 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.

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

[17] B. Sklar, *Digital Communications*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[18] H.-L. Lou, "Implementing the viterbi algorithm," *IEEE Signal Processing Mag.*, vol. 12, no. 5, pp. 42–52, 1995.

[19] 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.

[20] P. G. Gulak and E. Shwedyk, "VLSI structures for viterbi receivers: Part I—General theory and applications," *IEEE J. Select. Areas Commun.*, vol. 4, pp. 142–154, Jan. 1986.

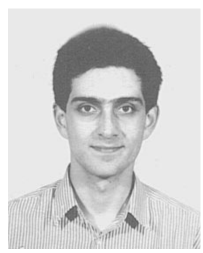
[21] S. Kubota, S. Kato, and T. Ishitani, "Novel viterbi decoder VLSI implementation and its performance," *IEEE Trans. Commun.*, vol. 41, pp. 1170–1178, Aug. 1993.

[22] K. K. Parhi, "High-speed VLSI architectures for Huffman and Viterbi decoders," *IEEE Trans. Circuits Syst. II*, vol. 39, pp. 385–391, June 1992.

[23] 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, pp. 616–624, Mar. 1992.

[24] H. J. A. M. Heijmans, *Morphological Image Operators*. Boston, MA: Academic, 1994.

[25] N. D. Sidiropoulos, "The viterbi optimal runlength-constrained approximation nonlinear filter," in *Proc. Int. Symp. Mathematical Morphology*, Atlanta, GA, May 1996.



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

From 1988 to 1992, he was a Fulbright Fellow and a research assistant at the Institute for Systems Research (ISR) of the University of Maryland. From September 1992 to June 1994, he served his military service with the Hellenic Air Force. From October 1993 to June 1994, he also was a member of the technical staff of the Systems Integration Division, G-Systems Ltd., Athens, Greece. In 1994, he joined the Institute for Systems Research, University of Maryland, as a Postdoctoral Research Associate. In 1996, he was promoted to the rank of Assistant Research Scientist. He teaches regularly in the Department of Electrical Engineering at the University of Maryland as an adjunct professor. His primary 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 a registered P.E. in Greece.