An Iterative Algorithm for Two-Dimensional Digital Least Metric Problems with Applications to Digital Image Compression

Keith M. Chugg, Xiaopeng Chen, Antonio Ortega, Cheng-Wei Chang
Electrical Engineering–Systems Dept.
University of Southern California
Los Angeles, CA 90089-2565

Abstract
A correspondence between the problem of two-dimensional digital least-metric (DLM) fitting and data detection in serially concatenated systems in digital communication theory is described. Nearly optimal detection algorithms based on recent advances in iterative detection/decoding are applied to the DLM problem for two applications in digital image compression. The first application is least squares halftoning of digital images. The second is near-lossless (i.e., error constrained) minimum-entropy image compression. In both applications the use of the iterative algorithm yields significant improvements, measured in terms of residual metric, relative to previously suggested approaches to the DLM problem.

1 Introduction
This paper addresses the two-dimensional (2D) digital least-metric (DLM) problem which has a broad range of applications in digital imaging (e.g., processing, compression, and generation), page-oriented communications, and concatenated systems in digital communications. We define the general 2D-DLM problem, introduce an approach to its approximate solution, and demonstrate this approach in two digital image compression problems. The 2D-DLM algorithm used in this paper is based on the identification of a correspondence between the 2D-DLM problem and maximum likelihood data detection/decoding for digital communication systems using serially-concatenated systems (e.g., error correction codes) along with recent advances in decoding algorithms for such systems. Specifically, a direct correspondence between a 2D system with finite memory and a serial concatenation of two 1D systems with block interleaving is described. Iterative algorithms based on soft-in/soft-out (SISO) algorithms similar to those used in “turbo decoding” are applied [2, 3]. The concatenated model is based on that in [1], where the approach was applied to the mitigation of 2D intersymbol interference (ISI).

The class of problems we consider consists of finding, among a discrete set of choices, the intensity level for each pixel \( b(i, j) \) in an image such that a given additive cost metric over the whole image is minimized. This may be formulated as the following 2D constrained minimization problem with the additive cost function being defined by a local neighborhood of \( b(i, j), b(i, j) = \{b(i - l, j - m)\}_{l, m=\{-L_v,-L_h\}} \)

\[
\min_{\{b(i, j)\}} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda(z(i, j); b(i, j))
\]

The pixel values are discrete so that \( b(i, j) \in B \) with \( N_B = |B| \) finite. The notation \( \{b(i, j)\} \) is intended to convey minimization over the entire \( N \times N \) page. A common special case of the 2D DLM problem is the 2D digital least-squares problem where \( \lambda(z(i, j); b(i, j)) = |z(i, j) - x(b(i, j))|^2 \) and \( x(\cdot) \) is some known mapping. Our experimental results will concentrate on two practical cases where this formulation arises. The first is the problem of halftoning an image to minimize the squared error between the (filtered) image and the filtered (reconstructed) halftone. Second, we consider the problem of optimal quantization in a near-lossless 2D predictive coding scheme (i.e., one where the prediction error is quantized so as to guarantee that a maximum reconstructed error is not exceeded). The techniques we present would also be applicable to other context based adaptive quantization environments (e.g., [4]) and in general to problems where the additive cost function for a given pixel is computed based on a finite set of surrounding pixels.

In these types of problems, because admissible values for \( b(i, j) \) are finite, one could, in theory, find the best \( b(i, j) \) by exhaustive search – i.e., requiring the comparison of \( N_B \) solutions, something that is clearly impractical for typical image sizes even if there are only two admissible values (as is the case for halftoning). For the corresponding one-dimensional problem the search over the \( N_B \) possibilities can be done with complexity \( O(N_B) \) using dynamic programming (i.e., the Viterbi Algorithm (VA)). However, because of the

\[ \text{The dependence is assumed to be symmetric around the current pixel for notational convenience only.} \]
lack of a natural order in two dimensions, there is no known efficient approach for conducting the exhaustive search for the 2D-DLM problem. A common approach to the 2D-DLM problem is the application of a 1D-DLM algorithm (i.e., the VA) across rows with or without decision feedback from the rows above. This decision-feedback VA (DF-VA) approach does not entirely account for the 2D nature of the cost function which may result in performance that varies with position (i.e., worse as one goes down rows due to error propagation) and/or a poor approximation to the best fit. The DF-VA approach was suggested for ISI mitigation in [5]. The use of row-column iterations of hard decisions and increased state complexity was suggested in [6] as a modification to the DF-VA with the application being deblurring of digital images. The DF-VA was utilized in [7] for minimum entropy, near-lossless image compression via a constrained error criterion. Least-squares based halftoning was suggested in [8] with a 1D cost function and a row-by-row VA. Tree-based methods using the ML algorithm have also been proposed in [9] for least-squares halftoning. The iterative SISO algorithms used in this paper were shown to substantially outperform the DF-VA in the 2D ISI mitigation problem in [1]; with the iterative SISO algorithms performance nearly coinciding with that of the exhaustive search.

2 The Iterative-MSM Digital Least-Metric Algorithm

The correspondence between a 2D system and concatenated interleaved systems follows by considering the creation of a new array of conceptual pixels $a_j(l)$ which collapses the memory corresponding to the horizontal direction (i.e., sets of $2L_h + 1$ samples are grouped as vectors)

$$a_j(l) = [b(i, j - L_h) \ b(i, j - L_h + 1) \ \cdots \ b(i, j + L_h)]^T$$

(2)

Note that $a_j(l)$ takes on $N_2^{2(2L_h+1)}$ possible values. Also, the corresponding$b(i, j)$ may be used to compute the metric $\lambda(z(i, j); b(i, j))$. If the mapping from $b(i, j)$ to $a_j(l)$ in (2) is viewed as a row-wise (1D) mapping on $b(i, j)$, the mapping from $a_j(l)$ to $\lambda(z(i, j); b(i, j))$ is viewed as a column-wise mapping of span $(2L_h + 1)$, then the correspondence to serially concatenated systems with block interleaving is complete. This correspondence suggests the application of SISO iterative detection algorithms. The approach presented in this paper is based on the Minimum Sequence Metric (MSM) SISO algorithm, which is essentially the VA with auxiliary (soft) outputs [2, 3]. We first briefly describe such a 1D SISO algorithm, followed by a description of the application to the 2D-DLM problem.

Consider a generic finite state machine (FSM) with input $d_k$, memory $L$ and output $t_k$ uniquely corresponding to a state transition $d_k^L$. The inputs to the MSM-based SISO module are the a-priori metrics associated with each $d_k$, $\mu_a(d_k)$ and with each state transition, $\mu_t(t_k)$. The outputs are related to the metric of the best data path $d_k^L$ consistent with $t_k$ (i.e., the MSM of $t_k$) and $d_k$ (i.e., the MSM of $d_k$). More precisely, for path metric $M(d_k^L) = \sum_{l=1}^{N} [\mu_t(d_{k-L}^L) + \mu_a(d_k)]$

$$I-MSM(d_k) = \min_{d_{k-L}^L} M(d_k^L) - \mu_a(d_k)$$

(3)

$$I-MSM(t_k) = \min_{d_{k-L}^L} M(d_k^L) - \mu_t(t_k)$$

(4)

where, for example, $M(d_k^L) = D$ simply denotes the metric $M(d_k^L)$ conditioned on a particular value of $d_k$. The outputs are the above real numbers for each conditional value of $d_k$ and/or $t_k$. The “soft-decision” information $I-MSM(d_k)$ can be converted to a hard decision by selecting the value of $d_k$ which minimizes that quantity. Note that, for this 1D case, this hard decision corresponds exactly to that obtained by running the VA on the DLM problem. The outputs of the SISO processor are normalized by the input metrics (i.e., $\mu_t(t)$) according to the heuristic principles of “turbo” decoding. Thus, the notation $I-MSM(\cdot)$ is used for this normalized quantity to convey the fact that this is the quantity iterated (this is also referred to as “extrinsic” information in the turbo coding literature) and MSM(\cdot) is used for the associated un-normalized quantity (i.e., this is the actual minimum sequence metric).

The application of the iterative detection algorithm to the general 2D-DLM problem is as shown in Figure 1. An MSM algorithm is run down each column, for which the input symbols are $a_j(l)$. During the first iteration, the a-priori metrics for $a_j(l)$ are set to zero. The transition metrics for the column processing are $\lambda(z(i, j); a_j(l-1))_l^{L_h-L_c}$ for each iteration. The MSM($a_j(l)$) is the best metric (distortion) that can be obtained for each value of $a_j(l)$ considering the $j^{th}$ column of the image. The value of I-MSM($a_j(l)$) is computed during this column processing for each pixel location and used as the transition metric for an MSM processor operating on the $j^{th}$ row; thus “folding-in” information for the entire $j^{th}$ column. This row processor, using zero a-priori metrics for $b(i, j)$, outputs a new version of I-MSM($a_j(l)$) (i.e., the I-MSM of the FSM transitions). The row SISO processor can also produce the soft information on its input, namely I-MSM($b(i, j)$), which may be minimized over the possible values of $b(i, j)$ to provide the approximate solution to the 2D-DLM problem. This describes one iteration. Future iterations are conducted by setting the a-priori (data) metrics for the $j^{th}$ column processor to I-MSM($a_j(l)$) as obtained from the output of the $j^{th}$

Note that we are considering columns of $a_j(l)$, thus corresponding to $2L_h + 1$ pixels in the original image. By doing so we ensure that the computed metrics are unaffected by pixels outside the column.

---

$^2$The notation $V_{b_i}^{b_2}$ is used as shorthand for $\{v_k\}_{k=-b_2}^{b_1}$. 

$^3$Note that we are considering columns of $a_j(l)$, thus corresponding to $2L_h + 1$ pixels in the original image. By doing so we ensure that the computed metrics are unaffected by pixels outside the column.
row processor of the previous iteration. The processing described above is precisely the iterative decoding algorithm for serial concatenated systems described in [3], generalized to an arbitrary metric as in [2], and applied to the 1D equivalent concatenated model as developed in [1]. This approach, while suboptimal, has been noted to perform remarkably well for concatenated 1D systems in [3].

3 Example Applications in Digital Image Compression

We consider two applications of the proposed 2D optimization approach. Note that in both cases our goal is to illustrate the improvement achievable when using algorithms that more fully exploit the 2D nature of the problems, rather than to present a complete solution for the applications at hand. It is assumed that the processing used to approximately solve the DLM component of the compression problem operates effectively as an isolated “black box” with any associated improvements preserved by suitable design of the other components of the compression algorithms (e.g., metrics, filter coefficients, etc.). Thus, we compare the performance of the proposed approach to that of the often-used DF-VA.

The iterative algorithm described in Section 2 is more complex than the standard DF-VA approach in several ways. The MSM algorithm is essentially equivalent to a forward and backward VA, thus the overall complexity is that of running an \(N_B^{2L_v}\)-state and an \(N_B^{2L_v(2L_u+1)}\)-state VA over a sequence of length \(2N^2I\), where \(I\) is the number of iterations. The DF-VA is equivalent to running an \(N_B^{4L_vL_u}\)-state VA over a sequence of length \(N^2\). Thus, the iterative MSM approach is a factor of \(2I\left[N_B^{2L_v(2L_u+1)} + N_B^{2L_v}\right] \approx 2IN_B^{2L_v}\). The factor of two is due to the addition of a backward recursion, and the factor \(N_B^{2L_v}\) represents the fact the proposed algorithm does not use “state reduction” via decision feedback as does the DF-VA. The row or column processing for the proposed algorithm can proceed entirely in parallel, whereas the DF-VA must be run one row at a time to provide the decision feedback. There are differing memory requirements as well (e.g., the soft information requires more memory, but storing survivors is not necessary as in the DF-VA). Even without any iterations, passing soft information typically provides a significant increase in performance which is greater than that achieved by increasing the state complexity of the DF-VA [6].

3.1 LS-based Halftoning

For the halftoning of images, \(b(i,j)\) represents the binary halftoned image and \(z(i,j)\) is the gray-scale original. The potential utility of the proposed 2D-DLM algorithm is illustrated by using a simple LS metric

\[
\lambda(z(i,j); b(i,j)) = [z(i,j) - h(i,j)*b(i,j)]^2
\]

where the low-pass filter \(h(i,j)\) is zero for \(|i| > L_v\) or \(|j| > L_h\), and \(b(i,j)\) is a candidate binary halftone image. More effective metrics are considered in [9]. A 512 \times 512, 8-bit gray-level version of the lenna image was halftoned using the iterative algorithm (\(I = 3\) iterations) and the DF-VA; the results are shown in Figure 2. A 3 \times 3 filter was used with values (scanned left-to-right from top-to-bottom): (0.2219, 0.1439, 0.0355), (0.1439, 0.0980, 0.0306), (0.0355, 0.0306, 0.0174). This filter was inspired from the filter used in [9] without any effort to optimize. The resulting metric was approximately 30% less for the proposed approach than for the DF-VA. The proposed approach also yields significantly better perceptual quality.

One perceptually undesirable property of the halftones is the distinctive patterns in areas of constant gray-level. The “soft-in” nature of the SISO processing allows one to control these patterns by biasing the algorithm toward a particular reference image. For example, an initial bias toward a gray pattern suitable for a particular imaging device may be appropriate.
Random biasing can also be used to alleviate the patterns. For example, on the first iteration we may set the a-priori metrics $\mu_d(a_j(i)) = \sum_{i=-L}^{L} \mu(b(i,j+1))$ where $\mu(b(i,j) = 0)$ is a random number uniformly distributed in $[0, r]$ and $\sum_{B=0,1} \mu(b(i,j) = B) = r$, and $r$ is a designable parameter. This has the effect of biasing toward a random halftone to alleviate the distinctive patterns. This provides a clouding of the halftone which is partially removed with each iteration. Figure 3 shows a halftoned version of lenna using $I = 3$ iterations and random biasing with $r = 3 \times 10^4$.

### 3.2 Near-Lossless Compression

Imposing a constraint of a maximum pixel error of $\pm d$ levels is a useful tool for near-lossless image coding [10, 7]. This constraint was used within a context-adaptive DPCM coder in [10, 7]. Given the local neighborhood (the context) a uniquely defined probability model is chosen for the prediction error and one selects the quantizer level which minimizes the entropy while preserving the $\pm d$ distortion constraint. Since choices for the current pixel affect future contexts a corresponding 2D-DLM problem can be formulated in this case where the memory corresponds to the DPCM predictor and the metric is the total entropy of the prediction error. Specifically, the metric is defined by

$$\lambda(z(i,j); b(i,j)) = -\log P(h(i,j) \times b(i,j))$$

where $b(i,j) = x(i,j) - \hat{x}(i,j)$ being the difference in the original and encoded image. Under the near-lossless constraint that each pixel in the encoded im-

Figure 2: Halftoning results for (a) the DF-VA and (b) the iterative algorithm at 150 dpi.

Figure 3: Halftoning results for the iterative algorithm at 150 dpi with random biasing.
There are several possible roles for these techniques. It is performed many times (e.g., image compression applications). Second, for applications where encoding is performed once and decoding is performed many times (e.g., image databases), these algorithms represent practical approaches when performance is at a premium. Finally, these approaches may become practical for real-time applications as the available processing power increases.

In order to more accurately assess the practical improvements, these iterative DLM algorithms must be integrated into complete compression algorithm designs. This process will be aided by the development of reduced complexity approximations that will allow the use of metric functions with larger supports.

### Table 1: Near lossless coding results for $d = 1$, i.e., the maximum absolute value of the pixel error is 1. $M$ indicates the number of probability models used, $2M + 1$.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>lenna.256</th>
<th>boat.256</th>
<th>house.256</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSM (M=0)</td>
<td>2.42753</td>
<td>2.44261</td>
<td>1.80555</td>
</tr>
<tr>
<td>DF-VA (M=0)</td>
<td>4.57418</td>
<td>4.61178</td>
<td>3.33947</td>
</tr>
<tr>
<td>MSM (M=3)</td>
<td>2.33708</td>
<td>2.37737</td>
<td>1.71897</td>
</tr>
<tr>
<td>DF-VA (M=3)</td>
<td>4.34798</td>
<td>4.44904</td>
<td>3.04858</td>
</tr>
</tbody>
</table>

4For $d = 1$, the prediction error can be constrained to even values to reduce the prediction error variance.

### References


