Model-Based Digital Image Halftoning using Iterative Reduced-Complexity Grid Message-Passing Algorithm

Phunsak Thiennviboon[a], Antonio Ortega[b] and Keith M. Chugg[b]

[a]TrellisWare Technologies, Inc., Poway, USA
[b]Dept. of Electrical Engineering, University of Southern California, Los Angeles, USA

ABSTRACT

An iterative grid message-passing algorithm for model-based digital image halftoning is introduced. Based on the standard message-passing algorithm on the grid graphical model, the algorithm is designed to suboptimally solve general two-dimensional (2D) digital least metric (DLM) problems and is found to be very successful (i.e., nearly optimal) for 2D data detection in page-oriented optical-memory (POM) systems. In contrast to many 2D (iterative) optimization techniques, this grid algorithm attempts to achieve a globally optimal solution via a local-metric computation and message-passing scheme. Using a reduced-complexity technique, the simplified grid algorithm is proposed for the halftoning problem and is shown to provide similar image quality as compared to the best halftoning algorithms in the literature. Since the grid algorithm does not exploit the properties of a specific metric, it is directly applicable to other digital image processing tasks (e.g., optimal near-lossless coding, entropy-constrained halftoning, or image/video dependent quantization).

Keywords: Model-based digital image halftoning, iterative message-passing algorithms, grid algorithm

1. INTRODUCTION

Digital image halftoning is a process of generating a binary image that, when imaged, approximates an original gray-scale image. Halftoning is important for laser printing and binary image rendering and various halftoning algorithms have been suggested in the literature. Each technique can be classified as a point process (e.g., pattern dithering[1]), a neighborhood process (e.g., error diffusion[1]), or a model-based optimization. In general, point processes are extremely simple, but the image quality is relatively poor as compared with other techniques. On the other hand, the model-based optimization methods are more complex but usually achieve the best performance.*

Model-based optimization techniques are commonly based on various searching algorithms that attempt to provide a halftone image with the best cost metric corresponding to certain system models. The most popular cost function is a square-error metric together, combined with a two-dimensional (2D) visual model.[2–5] In fact, this 2D least-square model-based halftoning is a special case of the general problem, namely the general 2D digital least-metric (DLM) problem.[6] The 2D DLM problem arises in a broad range of applications in digital imaging (e.g., processing, compression, and generation), page-oriented optical-memory (POM), and concatenated systems in digital communications. For the analagous 1D least metric problems, the Viterbi Algorithm would provide an optimal solution efficiently. The 2D DLM problem is, however, NP-hard[7] and there are no efficient approaches to conduct an exhaustive search.

In the halftoning literature, numerous authors have proposed solutions for this 2D optimization, mostly based on the application of 1D DLM algorithms or suboptimal heuristic-based 2D optimization algorithms, such as simulated annealing,[8] neural networks,[8] the Viterbi algorithm,[9] multipath tree-coding,[5] the toggle-only

Further author information:
E-mail: phunsak@trellisware.com, Telephone: 1 858 726 0130, Address: TrellisWare Technologies, Inc., Poway, CA, USA
E-mail: ortega@sipi.usc.edu, Telephone: 1 213 740 2320, Address: Signal and Image Processing Institute, Department of Electrical Engineering, University of Southern California, Los Angeles, CA, USA
E-mail: chugg@usc.edu, Telephone: 1 213 740 7294, Address: Communication Sciences Institute, Department of Electrical Engineering, University of Southern California, Los Angeles, CA, USA

*It is assumed that the model for the halftoning process takes into account all effects of the printer or screen rendering.[2]

algorithms,[2–4] and the toggle/swap algorithms.[2–4] Recently, after the decoding algorithm of turbo codes was introduced,[10] iterative message-passing algorithms have proven to be a powerful tool for many DLM problems in the communications literature.[11, 12] Such algorithms utilize a local message computation and propagation through a graphical model in order to reach near optimal solutions after several iterations. This is to be contrasted with many 2D iterative optimization techniques, e.g., simulated annealing and toggle/swap algorithm, which directly optimize the global cost metric. These message-passing approaches are variously known as belief propagation,[13] generalized distributive law (GDL),[14] and sum-product algorithms.[12]

For the halftoning problem, the iterative message passing algorithm via a serially-concatenated row-column graphical model has been introduced.[6] As a first step, it was shown that this type of algorithm is valid for the problem. Nevertheless, this iterative row-column algorithm is too complex without substantial performance. In this paper, an alternative iterative message-passing algorithm on a grid graphical model is proposed. This grid algorithm was demonstrated to provide nearly optimal performance close to that of the row-column algorithm in the POM problems (i.e., 2D equalization).[7, 11] The grid algorithm presented here achieves better performance with much less complexity than that of the row-column algorithm. The first novelty in our work is the introduction of a loopy graphical model, which we call the *grid model*, that enables the application of an iterative message-passing technique to the 2D DLM problem. We also introduce reduced-complexity techniques that prove useful for the digital image halftoning problem but could be used in other scenarios as well. The results show that the halftone image quality we achieve is comparable to that of the state-of-the-art toggle/swap scheme. Note that while our results are for the halftoning problem, the general framework we propose to solve the 2D DLM problem can be applied to other problems in digital imaging (e.g., optimal near-lossless compression, entropy-constrained optimal halftoning, or image/video dependent quantization) with appropriate changes (i.e., proper metric definition). Note that a part of this work was presented in Ref. 15 and extended to an application of image/video dependent quantization.[16]

This paper is organized as follows. In the next section, the iterative grid algorithm for the 2D DLM problems is described. The reduced complexity method to simplify the grid algorithm for digital image halftoning is proposed in Section 3. Section 4 illustrates the results and provides a discussion of the performance. Some conclusions are summarized in Section 5.

2. ITERATIVE GRID ALGORITHM

The 2D DLM problem[6] is a problem of finding the intensity level for each pixel $b_{i,j} \in \mathcal{B}$ (finite) that minimizes an additive cost metric with local dependencies $\lambda_{i,j}(\cdot)$. Specifically, the desired image is $\hat{\mathbf{B}}$ which satisfies

$$\hat{\mathbf{B}} = \arg\min_{\mathbf{B}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \lambda_{i,j}(z_{i,j}; T_{i,j}) \tag{1}$$

where $\mathbf{B} = \{b_{i,j}\}_{i,j=0}^{N-1}$ is a candidate image, $T_{i,j}$ is the value of this candidate image in the neighborhood of (i,j), and $z_{i,j}$ is the observation at location (i,j). Specifically, $z_{i,j}$ would be the (filtered) original gray-scale intensity at location (i,j) in the halftoning problem and the received intensity at location (i,j) for the image deblurring problem, respectively. In addition, the support region of $\lambda_{i,j}(z_{i,j}; T_{i,j})$ determines the number of possible values of $T_{i,j}$ – e.g., if the support region is 3×3 and $b_{i,j}$ is binary, there are 2^9 possible values of $T_{i,j}$, which we refer to as *local configurations*.

The 2D DLM problem may be written as a two-step problem of the form

$$\hat{b}_{k,l} = \arg\min_{b_{k,l}} \left[\min_{\mathbf{B}: b_{k,l}} \sum_{i,j=0}^{N-1} \lambda_{i,j}(z_{i,j}; T_{i,j}) \right] \tag{2}$$

where the *inner* minimization is conducted for all image candidates corresponding to the conditional value $b_{k,l}$. The problem of computing this inner minimization in (2) is a well-known problem in engineering and computer science (e.g., see Ref. 7, 11, 12, and 14, and references therein). Although there are many approaches to tackle this problem, the structure of the relationship of variables in (2) can be represented by a graph and algorithms

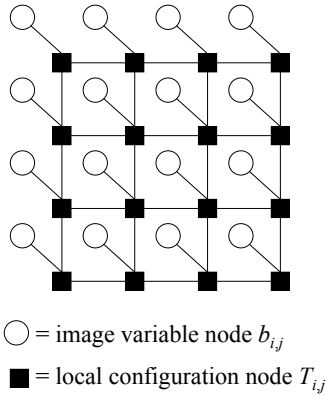


Figure 1. An example of 4×4 grid model.

based on the notion of message-passing on the graphical model may provide efficient solutions to this problem. Specifically, if the underlying graphical formulation is cycle-free, message-passing algorithms provide the desired solution. Recently, however, it has become widely appreciated that message-passing algorithms can be very effective in practice even when the underlying graphical model contains cycles (i.e., when there is no theoretical assurance of optimality).

2.1. Grid Model

A grid model is an undirected graph $G = (V, E)$ where V and E are the vertex (or node) and edge sets, respectively. Specifically, the vertex set V consists of $N \times N$ image variable nodes corresponding to $\{b_{i,j}\}_{i,j=0}^{N-1}$ and $N \times N$ local configuration nodes corresponding to $\{T_{i,j}\}_{i,j=0}^{N-1}$. The edge set E contains three different types of edges connecting two nodes together, including edges connecting node $b_{i,j}$ and node $T_{i,j}$, edges connecting node $T_{i,j-1}$ and node $T_{i,j}$, and edges connecting node $T_{i-1,j}$ and node $T_{i,j}$. Fig. 1 illustrates an example of the grid model when $N = 4$.

Each edge in G is labelled by the overlap (or mutual information) between information in two connected nodes, e.g., the label $L(e)$ of edge $e = (b_{i,j}, T_{i,j})$ connecting node $b_{i,j}$ and node $T_{i,j}$ is $b_{i,j}$. On the other hand, the label of edge $e = (T_{i,j-1}, T_{i,j})$ and $e = (T_{i-1,j}, T_{i,j})$ are $L(e) = T_{i,j-1} \cap T_{i,j} \stackrel{\Delta}{=} r_{i,j}$ and $L(e) = T_{i-1,j} \cap T_{i,j} \stackrel{\Delta}{=} c_{i,j}$, respectively, where $\cap$ represents an intersection operation between two sets. Moreover, a local function is assigned to each node providing a priori knowledge on each possible value of (a set of) variables corresponding to the particular node. For this grid model, the local function of the node $T_{i,j}$ is $\psi_{T_{i,j}}(T_{i,j}) = \lambda_{i,j}(z_{i,j}; T_{i,j})$ for each possible value of $T_{i,j}$ and the local function of the node $b_{i,j}$ is $\psi_{b_{i,j}}(b_{i,j}) = 0$ (i.e., no a-priori knowledge is assumed) for all possible values of $b_{i,j}$.

The relationship of this 2D problem representing by the grid model is that the region corresponding to the local configuration $T_{i,j}$ is *connected* by a rook movement in a chess.[7] In other words, if we start from one pixel in this region, we can move to all other pixels in the region with only upward or downward movement.

2.2. Messages, Message Updating, and Message Fusion

A message is a measure of quality providing an updated or a posteriori knowledge on each possible value or configuration of (a set of) variables pending at each edge. Let $\mu_{u,v}(L(e) = m)$ be a *directed* message for $L(e) = m$, at edge $e = (u, v)$, from node u to node v. Moreover, let $s \in \mathcal{S}$ be a specific configuration of a set of variables corresponding to the node u where $\mathcal{S}$ is a set of all possible configurations of s. The standard message-passing algorithm[†] consists of two basic operations, namely *message updating* and *message fusion*,[7, 12, 14]

[†]The message-passing algorithm considered in this paper is based on the minimization-summation (or min-sum) operation. Other operators are possible if they satisfy certain mathematical properties.[7, 11, 12, 14]

described in the following.

$$\mu_{u,v}(m) = \min_{\mathcal{S}:\, m} \left(\psi_u(s) + \sum_{w \in \text{ne}(u)\setminus\{v\}} \mu_{w,u}(n) \right) \qquad \text{(Message Updating)} \tag{3}$$

$$\sigma_u(s) = \psi_u(s) + \sum_{w \in \text{ne}(u)} \mu_{w,u}(n) \qquad \text{(Message Fusion)} \tag{4}$$

where $L(e=(u,v))=m$ in (3), $L(e=(w,u))=n$ in (3) and (4) uniquely determined by the configuration s, and ne(u) is the set of neighbor nodes connecting to node u. Note that $e=(u,v)=(v,u)$ since the graph is undirected. The addition and minimization operations in (3) and (4) sometimes are referred as the *combining* (or *product*) and *marginalization* operations in the communications literature.[7, 11, 14] Considering the message fusion in (4), a node output $\sigma_u(s)$ is computed which provides a posteriori knowledge on each possible configuration s of variables at node u. In other words, the decision $\hat{u}$ on the value of variables at node u is determined by thresholding the node output as follows.

$$\hat{u} = \arg\min_{s \in \mathcal{S}} \sigma_u(s). \tag{5}$$

Applying the standard message passing technique is equivalent to having a processing node corresponding to each node in Fig. 1. The role of this node, when it is activated, is to accept messages from its neighboring processing nodes, and to return a similar message to each of those nodes. $T_{i,j}$ and binary $b_{i,j}$ and the node processor at node $T_{i,j}$. Each of the four edge labels from neighbor local configuration nodes of $T_{i,j}$ comprises 6 binary pixels, so that each takes on 64 conditional values (or configurations) and the edge label from $b_{i,j}$ comprises one binary pixel with two possible values. Then the message updating at node $T_{i,j}$ can be described as follows.

$$\begin{aligned} \text{M}[T_{i,j}] &= \lambda_{i,j}(z_{i,j}; T_{i,j}) + \mu_{b_{i,j},T_{i,j}}(b_{i,j}) \\ &\quad + \mu_{T_{i-1,j},T_{i,j}}(c_{i,j}) + \mu_{T_{i+1,j},T_{i,j}}(c_{i+1,j}) + \mu_{T_{i,j-1},T_{i,j}}(r_{i,j}) + \mu_{T_{i,j+1},T_{i,j}}(r_{i,j+1}) \end{aligned} \tag{6}$$

$$\mu_{T_{i,j},u}(m) = \min_{T_{i,j}:\, m} \text{M}[T_{i,j}] - \mu_{u,T_{i,j}}(m) \tag{7}$$

where each of the combined conditional variables on the right-hand side of (6) are determined uniquely by the configuration $T_{i,j}$ considered, $m=L(e=(u,T_{i,j}))$ where $u \in \{b_{i,}, T_{i-1,j}, T_{i+1,j}, T_{i,j-1}, T_{i,j+1}\}$ in (7), and the minimization (or marginalization) in (7) is conducted for all local configurations $T_{i,j}$ corresponding to the edge variable with value m. Note that $M[T_{i,j}]$ takes on $2^9=512$ possible local configurations. Figs. 2(a) and 2(b) illustrate the combining and marginalization processes in (6) and (7), respectively, with $u=T_{i,j+1}$. The subtraction in (7) is used to create the so-called *extrinsic information* in the turbo decoding literature.[7, 10, 11]

Considering the node $b_{i,j}$, since there is only one edge connecting node $b_{i,j}$ and node $T_{i,j}$. The directed message from node $b_{i,j}$ to node $T_{i,j}$ is $\mu_{b_{i,j},T_{i,j}}(b_{i,j}) = \psi_{b_{i,j}}(b_{i,j})$ and never changes during the processing. In addition, the node output and decision of $b_{i,j}$ are

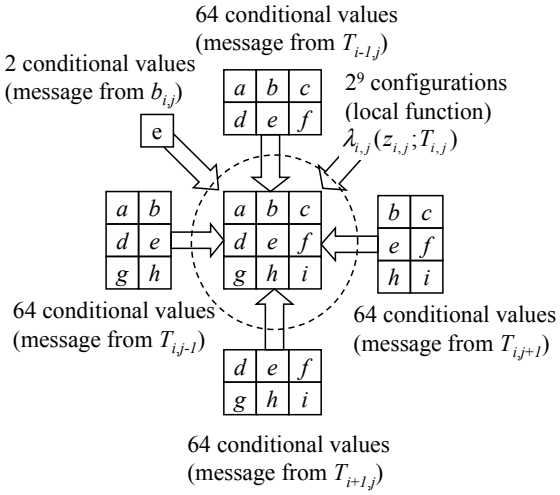
$$\sigma_{b_{i,j}}(b_{i,j}) = \psi_{b_{i,j}}(b_{i,j}) + \mu_{T_{i,j},b_{i,j}}(b_{i,j}) \quad \text{and} \tag{8}$$

$$\hat{b}_{i,j} = \min_{b_{i,j} \in \mathcal{B}} \sigma_{b_{i,j}}(b_{i,j}), \tag{9}$$

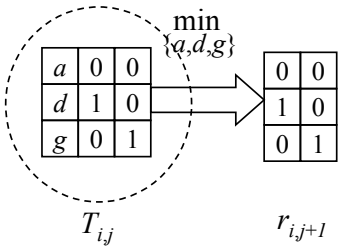
respectively.

2.3. Iterative Grid Algorithm

A grid algorithm is a standard message-passing algorithm on the grid model. It consists of the basic operations defined in (6)-(9). The algorithm is described by specifying a sequence or schedule of node processor activations (or node activations) in order to update (directed) messages exchanged among $\{T_{i,j}\}_{i,j=0}^{N-1}$ ((6)-(7)) and a stopping condition to generate the final decision on the image using (8)-(9). Since the graph is loopy, a schedule of node activations for message updating is iterated after achieving a certain pattern. By repeated activation of these



(a) Combining messages consistent with $T_{i,j}$.



(b) Example of marginalization before subtraction.

Figure 2. Examples of combining and marginalization operations for 3×3 support region $T_{i,j}$.

processing nodes, the global configuration (i.e., $\mathbf{B}$) is eventually accounted via these edge messages. In simple terms, after several iterations, local decisions take into account global costs. Note that since this graph is not cycle-free, the algorithm is not guaranteed to yield an optimal solution. Nevertheless, it was demonstrated that this suboptimal algorithm provides a near-optimal solution in the 2D data detection in the POM systems.[7, 11]

3. SIMPLIFIED ITERATIVE GRID ALGORITHM FOR DIGITAL IMAGE HALFTONING

In this paper, we consider the least-squares optimization corresponding to the halftoning problem, defined as

$$\hat{\mathbf{B}} = \arg\min_{\mathbf{B}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (z_{i,j} - \mathcal{X}(T_{i,j}))^2 \tag{10}$$

where $b_{i,j}$ is the intensity level of halftone image, $z_{i,j}$ and $\mathcal{X}(T_{i,j})$ are filtered versions of gray-scale and halftone images, respectively, $\mathcal{X}(T_{i,j}) = b_{i,j} * h_{i,j}$ and $z_{i,j} = y_{i,j} * h'_{i,j}$, where $y_{i,j}$ is the original gray-scale image[‡]. The filters $h_{i,j}$ and $h'_{i,j}$ are 2D filters representing the visual models and they could be different in order to control the sharpness of the halftone image. All images are $N \times N$ and black (white) pixels are represented by 1 (0) intensity level. Therefore, the grid algorithm is applicable with, however, an exponential complexity of the size of $T_{i,j}$ or the size of the support for the filter $h_{i,j}$. This requires a number of reduced-complexity techniques to enable its use with sufficiently large filters.

To limit the complexity of message updating, let $T'_{i,j}$ be a 3-bit L-shape pattern $\{b_{i-1,j}, b_{i,j-1}, b_{i,j}\}$ with 8 possible configurations. The square-error metric is then modified to be

$$\hat{\mathbf{B}} \quad = \quad \arg\min_{\mathbf{B}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (\check{z}_{i,j} - \mathcal{X}'(T_{i,j}))^2 \tag{11}$$

where,

$$\mathcal{X}'(T_{i,j}) \quad = \quad h_{1,0} b_{i-1,j} + h_{0,1} b_{i,j-1} + h_{0,0} b_{i,j}. \tag{12}$$

[‡]The notation "$*$" represents the two-dimensional convolution, i.e., $x_{i,j} * y_{i,j} = \sum_m \sum_n x_{i-m,j-n}\, y_{m,n}$.

Therefore, the grid algorithm is constructed based on the 2D DLM problem in (11). This modified metric is accomplished by using the most updated decision of the intensity levels, $\hat{b}_{(i,j)}$, to modify the *current* observed image (called a "decision-feedback" operation), i.e.,

$$\check{z}_{i,j} \quad = \quad z_{i,j} - \check{h}_{i,j} * \hat{b}_{i,j}, \tag{13}$$

where $\check{h}_{i,j} = h_{i,j}$ except for $\check{h}_{1,0} = \check{h}_{0,1} = \check{h}_{0,0} = 0$ and $\hat{b}_{i,j}$ is obtained from (8) and (9) assuming $\psi_{b_{(i,j)}}(0) = \psi_{b_{(i,j)}}(1) = 0$, i.e.,

$$\hat{b}_{i,j} \quad = \quad \arg \min_{b_{i,j} \in \{0,1\}} \mu_{T_{i,j},b_{i,j}}(b_{i,j}). \tag{14}$$

Note that, for this 3-bit L-shape support region, all messages are labelled with 1-bit symbol. In other words, $L(e = (b_{i,j}, T_{i,j})) = b_{i,j}$, $L(e = (T_{i,j-1}, T_{i,j})) = b_{i,j-1}$ and $L(e = (T_{i-1,j}, T_{i,j})) = b_{i-1,j}$. Together with a metric normalization by subtraction the metric of zero-bit message out of all possible values of message (i.e., 0 or 1) after message-updating process, the metric corresponding to zero-bit is always zero. Therefore, all messages are represented by one number and the number of metric additions in (6) over 8 possible configurations is reduced. The minimization operation in (14) is also simplified, so that it becomes a *sign* selection. In addition, since the value of $\hat{b}_{i,j}$ is either 0 or 1, the changing value $\hat{b}^{old}_{i,j} - \hat{b}^{new}_{i,j}$ is either 0, +1 or -1 and the updated observed data $\check{z}_{i+m,j+n}$ is obtained through the decision feedback by adding the coefficient $\pm\check{h}_{m,n}$ or doing nothing (i.e., $\hat{b}^{old}_{i,j} - \hat{b}^{new}_{i,j} = 0$). Hence, only the square-metric operation ($|\cdot|^2$) requires a multiplication operation[§]. Finally, the mapping table from the 8 configurations of $T_{i,j}$ to $\mathcal{X}'(T_{i,j})$ in (12) can be computed in advance and stored in the table during the operation.

Let us define each (i,j)-th node activation corresponding to the node $T_{i,j}$ as an operation of message updating in (6)-(7) for a particular set of out-going messages, bit decision on $b_{i,j}$ in (14), and decision feedback to $\check{z}_{i,j}$ influenced by the new value of $b_{i,j}$ in (13). Then, the simplified algorithm is conducted by running a sequence of node activations as follows. For each iteration, nodes $T_{i,j}$'s are activated row-wise left-to-right and right-to-left from top-to-bottom, and then column-wise top-to-bottom and bottom-to-top from left-to-right. Note that only directed messages influencing a future change of messages are updated. This will be run iteratively until a stopping condition is reached (usually with a fixed number of iterations). This activation schedule will be referred as the *row-column* schedule. This row-column schedule is only an example of possible activation schedules. In fact, it is necessary to redefine the schedule based on the real setting of a particular application (e.g., a type of initial images) in order to provide the best complexity and convergence characteristics.

In conclusion, the (outgoing) message updating from each node $T_{i,j}$ requires 8 multiplications (for the squaring operations), at most 30 additions and at most six 4-way compare/select operations. The decision feedback for each $\hat{b}_{i,j}$ requires at most $|T_{i,j}| - 3$ additions where $|T_{i,j}|$ is a number of bits involving in the local configuration $T_{i,j}$. Note that most of all operations in message-updating and decision feedback can be done in parallel. Each node $T_{i,j}$ is activated 4 times to update its outgoing messages in each iteration due to the row-column schedule. The complexity of the associated look-up-table (LUT) is assumed to be negligible. Nevertheless, the number of operations can be further reduced by exploiting or changing the structure of the algorithm, e.g., the activation schedule and the squaring operation.

These reduced-complexity techniques mentioned above could be applied to other 2D DLM problems without a constraint on any cost-function formulation. In fact, the decision feedback is a common technique to reduce the number of patterns for many searching algorithms (e.g., the Viterbi algorithm and the forward-backward algorithm) in the communications literature (see Ref. 7 and 11 and references therein). Although various reduced-complexity patterns are possible, the 3-bit L-shape pattern is chosen since it greatly simplifies the structure of messages while allowing messages passing in two dimensions. It also reduces the computational complexity and the storage required for this simplified grid algorithm. As will be seen in the next section, this simplified pattern yields a very good performance. On the other hand, the details of how to efficiently implement the decision-feedback operation significantly depend on the structure of a particular cost function. In other words,

[§]Note that further simplification on the square operation is possible.

Figure 3. Simplified grid algorithm (10th iteration).

Table 1. Comparison of square-error cost metric for various test images (random initial image).

Algorithm	Lena	Elaine	Fishing Boat	21 level step wedge	Stream and bridge	Man	General test pattern
Simplified grid algorithm	2.60E-4	2.38E-4	2.68E-4	2.31E-4	2.94E-4	2.56E-4	1.46E-3
Toggle/swap[2–4]	1.39E-4	1.20E-4	1.45E-4	1.30E-4	1.75E-4	1.43E-4	1.29E-3
Toggle-only[2–4]	4.17E-4	4.07E-4	4.46E-4	3.56E-4	4.88E-4	3.97E-4	1.55E-3
Floyd-Steinberg error diffusion[1]	3.47E-4	2.17E-4	4.23E-4	1.61E-4	6.17E-4	3.56E-4	1.41E-3

the linear nature of the 2D convolution and the binary intensity values (i.e., either 0 or 1) are belong to this specific 2D DLM problem. Note that the normalization operation is a general method and essential for every practical implementation.

4. RESULTS AND DISCUSSION

To evaluate the performance of the proposed simplified iterative grid algorithm we present several experimental results in this section. First, we consider the 512×512 ($N = 512$) gray-level Lena image. The filters $h_{i,j}$ and $h'_{i,j}$ are zero-mean Gaussian[2] with standard deviation $\sigma = 1.5$ (truncated to 9×9) and $\sigma = 0.9$ (truncated to 5×5), respectively. The initial binary image is randomly generated and all messages are initiated with zero metric. Figs. 3 show results of the grid algorithm at the 10th iterations and Fig. 4 shows the image obtained using Floyd-Steinberg error diffusion. Using the same metric definition and initial random image, the images obtained using the toggle-only and toggle/swap techniques are shown in Figs. 5 and 6, respectively. In this case, the perceptual quality of the halftone images obtained from the toggle/swap and grid algorithms is somewhat better than that of the images obtained from toggle-only and error diffusion (at least when they are printed with a laser printer). Specifically, the toggle-only image is "rougher" while error diffusion yields an image with more blurring and more unwanted artifacts. At the same time, the image obtained from error diffusion is smoother which may make it more visually pleasing, especially on a monitor screen. This is partly because the random

Figure 4. Floyd-Steinberg error diffusion.

Table 2. Square-error cost metric for various test images (error-diffusion initial image).

Algorithm	Lena	Elaine	Fishing Boat	21 level step wedge	Stream and bridge	Man	General test pattern
Simplified grid algorithm	1.90E-4	1.56E-4	2.05E-4	1.39E-4	2.81E-4	2.06E-4	1.37E-3

initial halftone image is very grainy. Moreover, the performance of the three optimization-based techniques is dependent on the choice of filter. Thus, it appears that the chosen Gaussian filter and random binary image tend to create somewhat grainy halftone images, and therefore alternative filters and initial binary images may be needed to obtain better perceptual results for this particular problem.

We also summarize our results for several other test images in in Table 1. On these images similar observations can be made with respect to perceptual quality. In general, the grainy effect is more noticeable in the flat and smooth regions than in regions with complicated texture. On the other hand, if there are artifacts due to error diffusion, these are visible in the flat and smooth regions. In terms of the square-error cost metrics, Table 1 provides cost metric values at convergence for each of the various techniques[¶] for several test images[‖]. These metrics seem to roughly correspond to the subjective quality (i.e., the better halftone has the lower metric) and the convergence characteristics for all techniques (except error diffusion, which does not explicitly optimize a cost funcion). It is difficult, however, to derive any strong conclusions from the differences between these absolute values. In particular, error diffusion metrics can sometimes be surprisingly low for this particular metric. This obviously indicates that the other techniques do not find a global optimal (i.e., algorithms are trapped in local minima) with the chosen random initialization. This result also seems to indicate that further work may be needed to define efficient and perceptually meaningful cost functions (i.e., in our case filters) and/or better initial binary images to be used in the context of a DLM halftoning problems. As an example, Table 2 illustrates cost metric values when the error diffusion images are used as initial images for the grid algorithm. While the numbers clearly show that our algorithm reduces the overall cost metric with respect to the initial condition

[¶]Floyd-Steinberg error diffusion is a one-pass non-iterative algorithm.

[‖]All test images can be found at `http://sipi.usc.edu/services/database/Database.html`

Figure 5. Toggle-only algorithm at convergence (after 25 iterations).

(error diffusion), it can also be observed that the images at convergence also "inherit" some characteristics of the initial error diffusion image, especially the textured appearance in the flat and smooth regions (see Figure 7).

In terms of complexity, the cost metric from the grid algorithm can be further reduced by increasing the number of searched patterns (i.e., changing the 3-bit L-shape to other patterns) together with choosing an appropriate sequence of node activations. As for the toggle-only and toggle/swap schemes, the direct binary search (DBS) approach[4] was simulated. To the best of our knowledge, the toggle/swap scheme is the highest performance algorithm known. When comparing the relative complexities of the grid and DBS techniques, both methods were implemented without filter- or image-dependent optimizations. Thus there is no special initialization in either technique and there is no exploration on the potential complexity reductions that may arise from symmetries in the values of the filter coefficients. For the DBS approach this is called the *original version*.[4] Under these conditions and the random initial image, the complexities of the grid and toggle/swap approaches are comparable, with the grid technique being generally faster than toggle/swap and about the same speed as toggle-only schemes. Note that this complexity does not include the complexity of the table initialization for both techniques. In addition, without special structure of the filters,[4] the complexity of the table initialization of the grid algorithm is much lower than that of toggle-only and toggle/swap algorithms. As an example, the square-error cost metric (averaged per pixel)** and the running time (after initialization)†† (using the random initial image) for the Lena image are shown in Figs. 8 and 9 as a function of the number of iterations.

Clearly, the grid algorithm converges faster than both toggle-only and toggle/swap with initial random image, while error diffusion remains the fastest technique. This convergence characteristic of the grid algorithm was found to be roughly the same for other test images in our experiment. In terms of the cost metric, for this Lena image, the toggle/swap technique is better than the grid algorithm and both of them are better than the toggle-only approach. Although the cost metric from the error diffusion is very close to that of the grid and

**Square-error cost metric (per pixel) between halftone and original gray-scale images is performed over all pixels in the image except the first 5 pixels from the image boundary.

††Running time is measured as the CPU time difference (Sun Ultra-30) without counting the required running time for initialization, e.g., look-up-table (LUT) initialization and pre-filtering.

Figure 6. Toggle/swap algorithm at convergence (after 19 iterations).

Figure 7. Simplified grid algorithm with error diffusion initial image (at convergence).

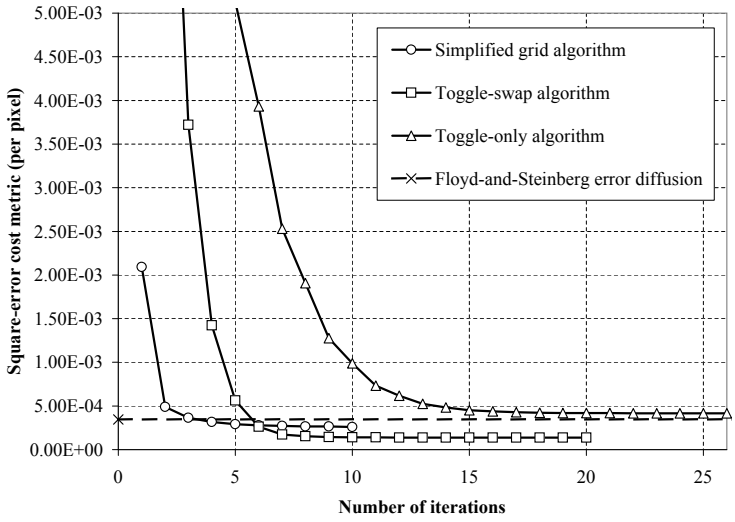


Figure 8. Square-error cost metric (per pixel) vs. number of iterations.

toggle/swap techniques, the halftone image quality obtained from error diffusion is lower quality than that of grid and toggle/swap methods.

A number of techniques have been introduced to accelerate the DBS operation.[4] Most of these techniques (e.g., choosing a good initial image before the start of the iteration) can be used within the grid framework as well so that the grid algorithm is expected, once filter- and image-dependent optimizations have been incorporated, to be operated at least at comparable speed. However, for the good initial image providing the fast convergence, the activation schedule of the grid algorithm is needed to be redefined (e.g., using a forward-backward serpentine scan instead of the row-column schedule) in order to remove the unnecessary complexity. In addition, other techniques can be used to reduce complexity of the grid algorithm. In fact, it is straightforward to show that the error diffusion[1] and 1D LSMB[9] can be derived from this grid algorithm. Finally, it is worth noting again that the algorithm introduced herein is applicable to general 2D DLM problems with roughly similar complexity regardless of the metric being used. The DBS techniques, however, are designed to exploit special properties of additive quadratic metrics. If alternative metrics were required (e.g., to produce an optimal halftone under an entropy constraint) the complexity of the algorithm would increase substantially. The application of the grid algorithm to other 2D DLM problems is an interesting area for future research (see Ref. 16 for an extended work using the grid algorithm).

5. CONCLUSION

In this paper, a novel iterative message-passing algorithm using the grid model was developed for the general 2D DLM problem. While many 2D iterative optimization techniques utilize a global cost metric, our approach relies on local metric computation and message propagation to achieve a suboptimal solution. As an example of an application of this algorithm, digital image halftoning was considered, and reduced-complexity techniques such as a reduced pattern and decision feedback were proposed. The complexity was further reduced by exploiting the structure of the proposed activation schedule and normalization. Our results show that the reduced-complexity version of the algorithm provides a halftone image with quality comparable to that of the DBS algorithm. Because there are no constraints on the metric definition, this algorithm can be applied to many 2D DLM problems in the image processing literature.

REFERENCES

1. R. A. Ulichney, *Digital Halftoning*, MIT Press, Cambridge, MA, 1987.

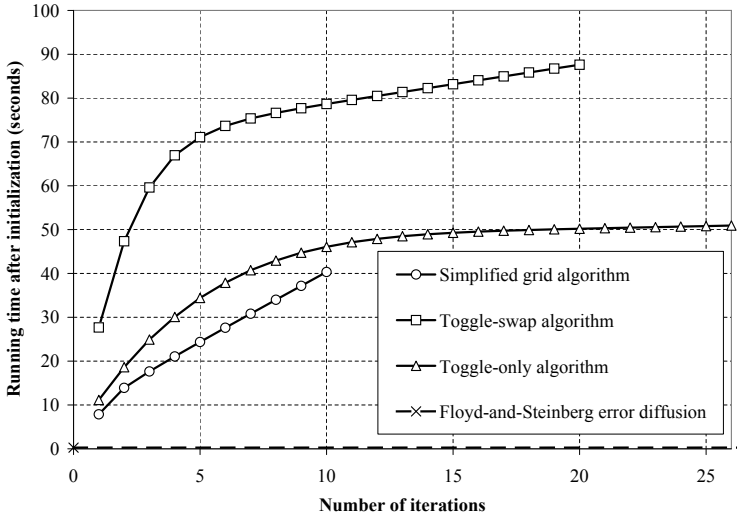


Figure 9. Running time after initialization (seconds) vs. number of iterations.

2. T. N. Pappas and D. L. Neuhoff, "Least-squares model-based halftoning," *IEEE Trans. Image Processing* **8**, pp. 1102–1116, 1999.
3. D. J. Lieberman and J. P. Allebach, "A dual interpretation for direct binary search and its implications for tone reproduction and texture quality," *IEEE Trans. Image Processing* **9**, pp. 1950–1963, 2000.
4. D. J. Lieberman and J. P. Allebach, "Efficient model based halftoning using direct binary search," *IEEE Trans. Image Processing* , submitted.
5. P. W. Wong, "Entropy-constrained halftoning using multipath tree coding," *IEEE Trans. Image Processing* **6**, pp. 1567–1579, 1997.
6. K. M. Chugg, X. Chen, A. Ortega, and C.-W. Chang, "An iterative algorithm for two-dimensional digital least metric problems with applications to digital image compression," *Proc. ICIP'98 (Chicago, Illinois)*, pp. 722–726, 1998.
7. P. Thiennviboon, *Graphical Models for Iterative Data Detection*, Ph.D. dissertation, University of Southern California, Los Angeles, CA, 2002.
8. R. Geist, R. Reynolds, and D. Suggs, "A markovian framework for digital halftoning," *ACM Trans. Graph.* **12**, pp. 136–159, 1993.
9. D. L. Neuhoff, T. N. Pappas, and N. Seshadri, "One-dimensional least-squares model-based halftoning," *J. Opt. Soc. Amer. A* **14**, pp. 1707–1723, 1997.
10. C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: turbo-codes," *Proc. ICC'93 (Teneva, Switzerland)*, pp. 1064–1070, 1993.
11. K. M. Chugg, A. Anastasopoulos, and X. Chen, *Iterative Detection*, Kluwer Academic Publishers, 2001.
12. F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory* **47**, pp. 498–519, 2001.
13. J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, San Mateo, CA, 1988.
14. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Trans. Inform. Theory* **46**, pp. 325–343, 2000.
15. P. Thiennviboon, A. Ortega, and K. M. Chugg, "Simplified grid message-passing algorithm with application to digital image halftoning," *Proc. ICIP'01 (Thessaloniki, Greece)*, pp. 1061–1064, 2001.
16. P. Sagetong and A. Ortega, "Message-passing algorithm for two-dimensional dependent bit allocation," *Proc. SPIE (Santa Clara, USA)*, 2003.