Watermark Rate and the Distortion of Error Diffused Halftones

Doron Shaked, Zachi Baharav
HP Laboratories Israel
HPL-98-89 (R.1)
June 26th , 2001*

E-mail: [dorons, zachi]@hp.technion.ac.il

watermarking, digital security, halftone printing, printing security, error diffusion

This report analyzes watermark embedding in Error Diffusion halftone images. A distortion measure for halftone modifications is introduced, and an embedding scheme is selected so as to minimize the proposed distortion. Next a statistical model is suggested for "natural-images", leading to a lower bound on the halftone distortion as a function of watermark rate. Bound realization is argued to be equivalent to solving a prediction problem. Accurate prediction is not possible, however a good estimator is suggested for watermark embedding. The proposed embedding procedure is compared to alternative procedures, and shown to have better distortion versus rate performance. The visual impact in real images is also demonstrated.

* Internal Accession Date Only
Approved for External Publication
HP Labs Israel, Technion City, Haifa 32000, Israel

1 Introduction

Watermarking of digital images is a process in which information is embedded in digital images, in a manner that does not damage the image quality, and is robust to non-destructive image processing procedures [2]-[6]. In other words, a good digital watermark can not be removed from an image without degrading it significantly. Applications of digital watermarking include digital security, and copy-right protection.

In this report we focus on watermarks for a special type of images, namely, printed halftone hardcopies. The idea is to watermark prints in order to authenticate a document or to identify the printer that produced it. Thus we do not intend the watermark to be robust to scanning an reprinting (the resulting new hardcopy will carry the watermark and signature of the new printer). Potential applications are related to secure printing, and include tracing illegal image documents, and embedding of digital signatures, and authenticating information in picture ID cards.

Watermark methods should enable a sufficient amount of embedded information introducing only minimal distortion to the image and its visual quality. In this report we analyze the special case of Error Diffusion halftone rendering and introduce a bound on the amount of information (rate) that can be embedded in an image if its quality is not to be reduced more than a specified (distortion) parameter.

Note that the proposed bound is different from the known rate-distortion bound in Information Theory. There the distortion relates to the same data the rate refers to (the watermark information in our case). In this report the distortion is in what might have been the channel in Information Theory (the image in our case). A rate distortion analysis of watermarking in its standard meaning was presented in [1].

In the next section a halftone distortion function is proposed, and an Error Diffusion based halftoning + mark-embedding scheme is selected so as to minimize the distortion induced by the watermark. In Section 3 a lower bound on the distortion is derived based on a statistical model for natural images. In Section 4 it is argued that achieving the lower bound may be formalized as an estimation problem. A simple estimator is proposed and its distortion performance is discussed in Section 5. Section 6 presents some examples of halftone watermarks, and demonstrates the results both numerically and visually. The report is concluded with a discussion in Section 7.

2 Distortion in Halftone Processes

This section introduces a distortion measure for halftone images, and compares a few ways to embed a watermark in images according to the expected distortion.

Suppose a halftone image is watermarked, and suppose both the mark encoder and the mark decoder have coordinated a set of image locations (pixels), whose halftone bits are the watermark content. If by some twist of luck the image halftones in those locations agree with the information that the mark encoder intended to embed in the image, then undoubtly, the mark went through without any distortion what so ever. Nevertheless, it is reasonable to assume that only part of the halftones (most likely half of them) agree with the watermark information, the rest of the halftones are necessarily forced to flip.

How much distortion should be charged for any such undesired flip? We propose to determine the distortion introduced by changes to the desired halftone pattern through a hypothetic "perceived image" $\tilde{G}$, an image that could have caused the new halftone pattern. The mark distortion is a measure $\|G - \tilde{G}\|$ of the difference between the perceived image and the original image G, where $\|\cdot\|$ is a root mean square measure. The difference $G - \tilde{G}$ is a "Salt&Pepper" type noise[1] rather than the classic "White Noise", therefore the total square error is normalized by the number of flips, rather then by the total number of pixels in the image. Thus, the distortion measure is the standard deviation of the additive noise at the flip locations.

Note that $\tilde{G}$ is not unique, namely, given the distorted halftone there are several different candidates for $\tilde{G}$. The distortion measure is defined to be the minimal distortion over all the possible $\tilde{G}$'s.

$\tilde{G}$ and consequently the halftone distortion measure depend on the halftoning process. In this report we limit ourselves to Error Diffusion. Figure 1 describes the flow chart for the basic Error Diffusion: The continuous gray value information $g_{ij} \in [0, 1]$ flows sequentially (in raster scan order). A corresponding error value e_{ij} incurred by "past" halftones (and stored in the error buffer) is added to give the "desired value" s_{ij} (the value that we would have wanted to render in the ij location). Since only halftone values (0 or 1) are possible, the closest is selected by comparing the desired value to 0.5.

[1] The difference is zero anywhere accept the flip locations, and potentially near them.

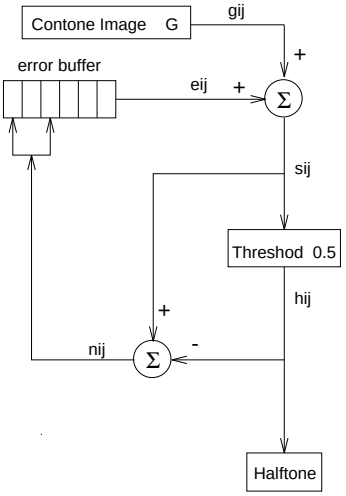

Figure 1: Flow chart of Error Diffusion halftoning.

$$h_{ij} = \begin{cases} 1 & s_{ij} \geq 0.5 \\ 0 & s_{ij} < 0.5 \end{cases} \tag{1}$$

The new error $n_{ij} = s_{ij} - h_{ij}$, incurred in obtaining h_{ij}, is "diffused" to "future" pixels (neighboring pixels who are yet to be halftoned). Floyd and Steinberg [7] have suggested a preferred diffusion filter related to the conventional raster scan, see Figure 2, where 7/16 of n_{ij} is diffused forwards, 1/16 to the location one pixel ahead in the next line, 5/16 to same position in the next line, and the rest to one position back in the next line.

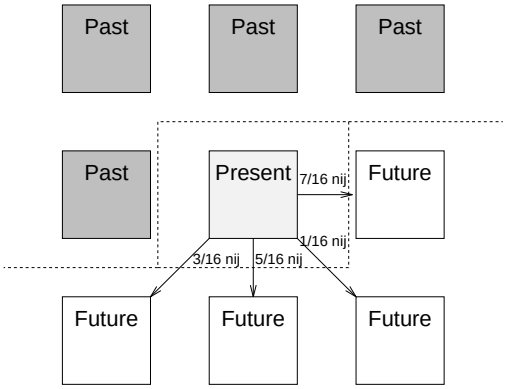

Figure 2: The Floyd and Steinberg filter for diffusing error n_{ij} to the future pixels.

Now let us determine the best way to incorporate watermark embedding into Error Diffu-

sion. Figure 3 describes flow-charts for three such schemes, in which we assume that the watermark locations have been predetermined by both the encoder and decoder. In the naive implementation of Figure 3a the watermark is imposed simply by replacing the halftone h_{ij} at the predetermined locations by the next watermark bit w_k.

Assuming the watermark embedding scheme described in Figure 3a induced a single flip at location i_0, j_0, what is the corresponding distortion? As mentioned before, there are many images that result in that flip. It is easy to see that $\tilde{G} = G \pm \delta_{i_0 j_0}$ is one of them, where $\delta_{i_0 j_0}$ is a delta function at $i_0 j_0$, and the sign '$\pm$' is determined according to the flip direction. However a unit size delta is larger than necessary. The least local error due to a flip is

$$d_{i_0 j_0} = |s_{i_0 j_0} - 0.5| \tag{2}$$

however, a local distortion that is smaller than unity necessarily causes a difference between the actual error, n_{ij}, and the same term in the halftone of $\tilde{G}$. This difference may induce changes in nearby locations or farther away in future locations. Wherever it occurs an additional distortion of $\tilde{G}$ is due. If the image is large enough all error term differences have to be accounted for somewhere in future locations, thus other possibilities for $\tilde{G}$ are of form

$$\tilde{G} = G \pm d_{i_0 j_0} \delta_{i_0 j_0} \pm (1 - d_{i_0 j_0}) \cdot \sum_{m \geq 0} \Delta_m \cdot \delta_{i_m j_m} \tag{3}$$

where locations $i_m j_m$, for $m > 0$ are in the "future" of $i_0 j_0$, and Δ_m are weights such that $\sum_{m \geq 0} \Delta_m = 1$, and $\Delta_0 \geq 0$.

The algorithm described in Figure 3b is very similar, only the watermark embedding is more sensibly located inside the feedback (directly after the thresholding). This way the Error Diffusion mechanism can be used to hide the watermark by modifying the "future" pattern. In this case it is possible to show that the images that cause this pattern are of the form

$$\tilde{G} = G \pm d_{i_0 j_0} \delta_{i_0 j_0} \mp d_{i_0 j_0} \cdot \sum_{m \geq 0} \Delta_m \cdot \delta_{i_m j_m} \tag{4}$$

Here also $\sum_{m \geq 0} \Delta_m = 1$, but now $\Delta_0 \leq 0$. To compare the two algorithms consider that $d_{i_0 j_0}$ is distributed in $[0, 1]$, with a strong bias towards the lower values, and thus the second algorithm is preferred.

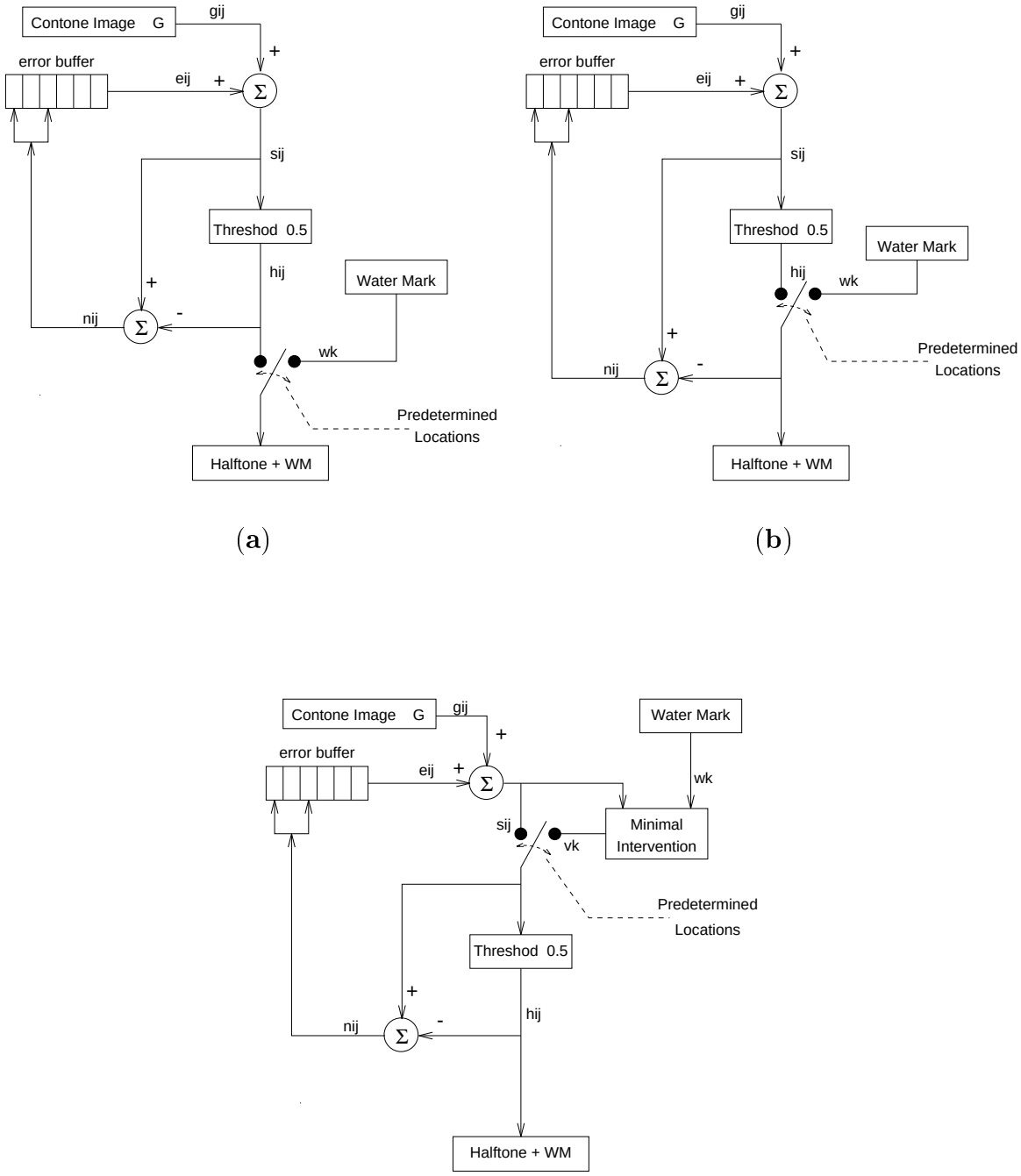

Figure 3: Watermark embedding schemes: a) Naive watermarking. b) Post-threshold watermarking. c) Pre-threshold watermarking.

In Figure 3c the watermark embedding is pushed back even further to before the thresholding. In this implementation, the mark is embedded by replacing s_{ij} at the predetermined locations with minimal intervention values v_k, that are equal to s_{ij} in case of concensus (i.e. the next halftone would by w_k anyway), or else $\epsilon > 0$ close to 0.5, with $\epsilon \to 0$.

$$v_k = \begin{cases} s_{ij} & \text{concensus}(s_{ij}, w_k) \\ 0.5 + \epsilon & \text{otherwise} \begin{cases} w_k = 1 \\ w_k = 0 \end{cases} \\ 0.5 - \epsilon & \end{cases} \tag{5}$$

with: $\text{concensus}(s_{ij}, w_k) \stackrel{\triangle}{=} ((s_{ij} < 0.5) \text{ and } (w_k = 0)) \text{ or } ((s_{ij} \geq 0.5) \text{ and } (w_k = 1))$

In this scheme the distortion may be attributed to the present location with no diffusion artifacts, and it is easy to see that the image

$$\tilde{G} = G \pm d_{i_0 j_0} \delta_{i_0 j_0} \tag{6}$$

results in the required halftone pattern. Naturally this is the preferred halftoning+marking scheme.

Figure 4 presents examples of watermark embedded halftones using the above watermark embedding algorithms. Shown are a halftoned image of constant original tone (without watermark) in Figure 4a, the flip locations in Figure 4b, and the results of inducing a flip at the designated positions using the three watermark embedding algorithms in 4c-e.

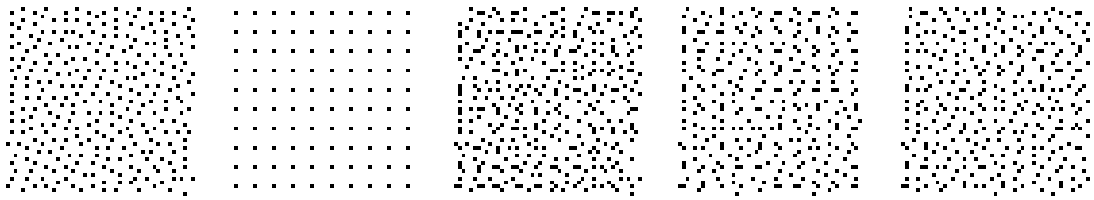

Figure 4: Test of watermarking schemes: a) Error diffusion with no watermark. b) Flip locations. c) Naive watermarking. d) Post-threshold watermarking. e) Pre-threshold watermarking.

Note that the naive watermark embedding algorithm in Figure 4c produces poor results. Better results are produced by the other two algorithms in Figures 4d and 4e, which seem

to have almost equivalent quality. The reason for the good quality in Figure 4d is that the error (4) of Post-threshold watermarking is predominantly high frequency. This reduces the visible noticeability, especially in a halftone. The frequency factor is not reflected in the proposed distortion measure.

3 Bounding the Distortion

In this section we use a statistical model for natural images to deduce a lower bound on the watermark distortion.

Images have approximately constant gray value almost anywhere, hence a random family of images having a constant gray value with a small additive noise, is a reasonable model for small neighborhoods of pixels. The random image $I = A + N$ is composed of a constant image A with a gray value uniformly distributed in $[0, 1]$, and a White Noise image N with a small standard deviation $\sigma \sim 0.01$. It is hopped that the statistics of this model are similar to those obtained on collections of natural images.

The probability distribution P_d of the minimal distortion d_{ij} in (2) is presented in Figure 5. It shows that d_{ij} is much more often smaller than 0.5 than otherwise, which justifies the preference of the second watermark embedding scheme over the first one.

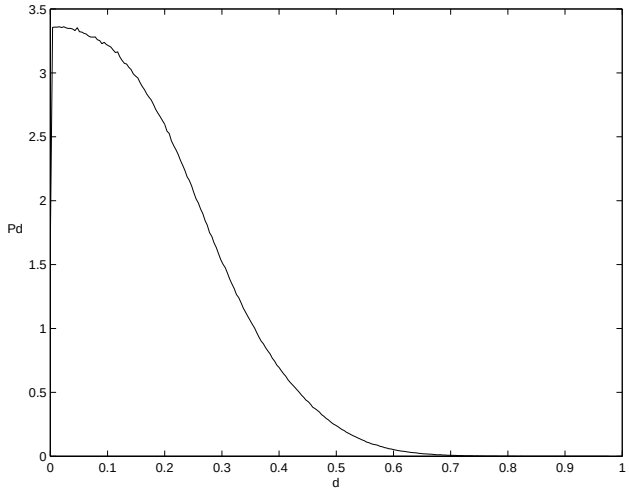

Figure 5: Probability distribution of d_{ij}.

Suppose it were possible to attach the set of watermark locations to any halftone image. This situation would have given the watermark encoder the freedom to embed the water-

mark wherever it does the least harm. To make things more realistic suppose also that the watermark encoder has to decide on a location before it is aware of the bit it has to place there (otherwise the distortion would have been zero for practically any rate). In this hypothetic situation the best embedding strategy is to place the information in those locations for which $d_{ij} = 0$. If there are enough such locations, the mark has incurred zero distortion, however, if more information bits are needed, locations for which $d_{ij} > 0$ have to be employed as well. Alternative embedding strategies can not be better then this one, and will necessarily result in a worse (or at best - equal) distortion for any given rate.

In the above embedding strategy, larger rates, employ larger chunks from the lower part of P_d so that the rate is their 0-order moment, and the resulting distortion is the square root of their normalized 2nd-order moment. Thus the distortion versus rate bound is the following parametric curve:

$$(D(d), R(d)) = \left(\sqrt{\frac{\int_0^d \alpha^2 P_d(\alpha) d\alpha}{\frac{1}{2} \cdot \int_0^d P_d(\alpha) d\alpha}}, \quad \int_0^d P_d(\alpha) d\alpha \right) \tag{7}$$

where $P_d(\cdot)$ is the probability distribution function presented in Figure 5, and d the marginal distortion (i.e. the maximal distortion embedded at a single location). The graph of (7) is the dotted graph in Figure 6. Any distortion smaller or rate larger than this curve are not possible.

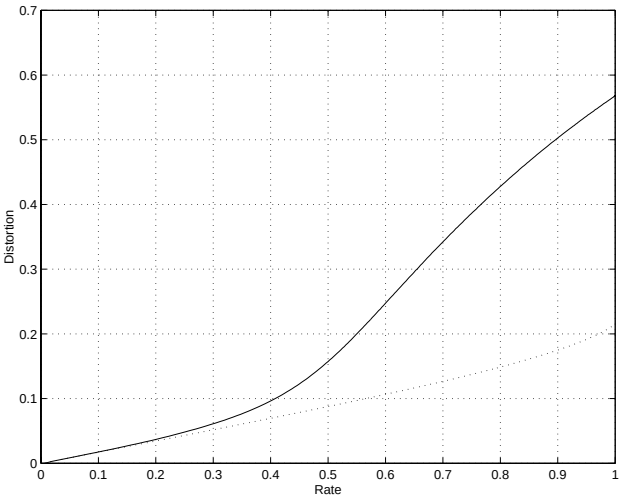

Figure 6: Lower distortion versus rate bounds. Dotted graph - expected from the statistics of unmarked halftones. Solid graph - accounting for the effect of the distortion on the statistics.

Evidently, the bound (7) depends on the statistics of d_{ij}, however since d_{ij}'s distribution changes once the watermarking starts to add distortion, the real bound on the distortion for a given rate has to be computed from the modified distribution. The solid graph in Figure 6 was computed from distributions of d_{ij} obtained by inducing a random flip (with probability 0.5) in locations for which d_{ij} was smaller then a predefined threshold. The probability distributions for threshold values 0, 0.12, 0.24, ...0.96 are presented in Figure 7.

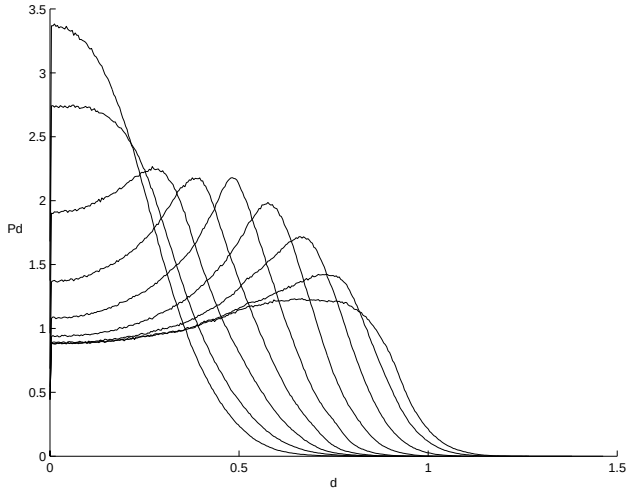

Figure 7: Probability distributions for d_{ij} at various rates of watermarking.

This bound is fantastic, since it implies that one can achieve a rate of 0.005 (i.e. a watermark information bit for every 200 pixels) with no distortion at all, and a rate of 0.05 (a bit in 20 pixels) with a distortion of 0.01 (i.e. no more than the White Noise we expect to see in natural images; $\sim \pm 4$ gray levels in the 255 scale). Note that White Noise occurs on all the pixels whereas in this case, the distortion occurs only in the flip locations. A distortion of 0.01 is therefore an additive Salt&Pepper noise (zero in most places) with a standard deviation of 0.01 on the flip locations (one of every 40 locations).

This brings up the questions: Is this bound tight? Can we somehow achieve it in practice? The answer is unfortunately No, however we can get quite close to it, as is shown in the following sections.

4 Estimating the Diffusion Error

In this section a method is suggested for selecting locations in which a watermark can be embedded so as to minimize the distortion. Achieving the lower bound described above is equivalent to an estimation problem to which the proposed method is a straight forward and easily implementable solution.

Since the situation described above in which watermark embedded locations are independently communicated to the decoder is not realistic some other indicators of the mark locations should be available to the mark decoder. Better embedding locations are characterized by low d_{ij}, thus any indicator achieving the lower bound is an accurate predictor of low d_{ij}'s. Since natural images include a small but non-ignorable amount of randomness, an accurate prediction of d_{ij} is impossible. Realistic implementations of the watermark embedding should therefore include estimators of low d_{ij}'s. The quality of the watermark will depend on the estimation quality.

In order to maintain the watermark integrity, and since in reality d_{ij} can not be accurately predicted in the decoder, the encoder has to employ the same estimator to determine the mark locations it uses. Thus we arrive at the watermark encoder flow chart of Figure 8. Note that the data the estimator can use is limited to the data available in the decoder at the given stage which is the causal[2] halftone image. In other words d_{ij}'s should be estimated from the causal halftone pattern.

Since images are usually locally constant (the image model we use is even more restricted in that sense), the average of the causal halftone neighborhood of a pixel should give the estimator a good approximation of the local gray value in the original image. Having that information it can emulate the halftoning algorithm and estimate d_{ij}. This process is best explained by an example: Suppose a White pixel (marked by an X) has the causal halftone neighborhood described in Figure 9. The decoder has to decide whether d_{ij} was large or small. To that end consider that a low d_{ij} means that, both the Black and the White halftones were reasonable possibilities for h_{ij}, and that the decision that was finally made (White$\equiv 1$) could have been different if s_{ij} were a little smaller than it really was[3]. For Figure 9a it is evident that a Black halftone was not a reasonable possibility at that position, therefore

[2] The term causal refers to the pixels that were previously processed.

[3] s_{ij} should have been, at least, d_{ij} smaller.

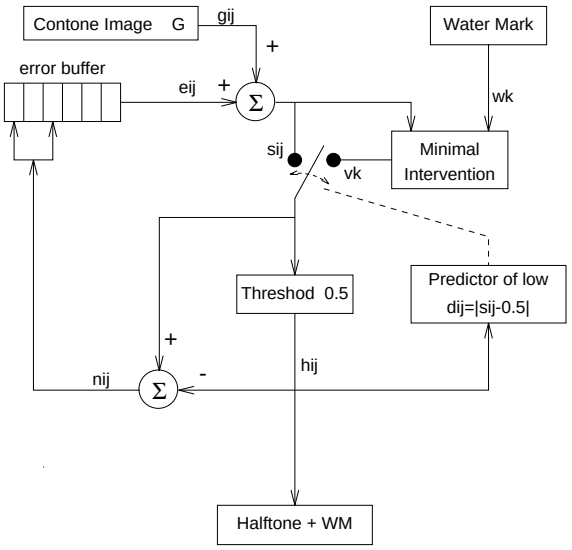

Figure 8: Flow chart of watermark encoder.

the estimator should indicate that d_{ij} was large. On the other hand, in Figure 9b both halftones seem to be a reasonable possibility for the given location, and thus in this case, the estimator should indicate that d_{ij} had a small value, and that the White value of the halftone is a watermark. A similar analysis for a different purpose appears in [8].

Figure 9: Pixels in causal neighborhoods indicating d_{ij} was probably: a) large. b) small.

The above example does not constitute a real estimator of d_{ij}, nor is such an estimator necessary. All one needs is an estimation of whether d_{ij} is smaller than a predefined threshold value which determines the rate (and the distortion), or not. Since operating an estimator on-line is time consuming, it is wiser to do the estimation off-line and pick up a list of all the neighborhoods which estimate a small d_{ij}. This is described in the next section.

5 Determining Watermark Location

This section describes and analyses a causal estimator of low d_{ij} values, allowing for consistent determination of watermark embedding locations in both the encoder and the decoder. Its

derivation was based on statistics obtained from a database approximating our model of natural images.

The estimation was performed off-line as follows: Natural images are represented by a set of 249 images of constant gray level $\{4, 5, \ldots 251\}$ with additive independent noise distributed uniformly in the range $\{-4, 4\}$. The size of all the images is 500×500. This set was halftoned without any watermark. Resulting "natural halftones" were analyzed to find a set of *predominant neighborhoods* (neighborhoods with more than 50 occurrences). Then, the sample average of d^2 was calculated for each predominant neighborhood. Each value represents the expected average square distortion incurred if the corresponding neighborhood is added to the estimator's lookup table. Finally, the predominant neighborhoods were ordered on a list in an increasing distortion order. Partial lists off the top of this list were used for estimator lookup tables. More entries in the lookup table correspond to higher rate (and higher distortion).

All the above was done for three different neighborhood sizes: 4, 11, and 21. The neighborhoods are depicted in Figure 10. Note that the neighborhoods are causally nested in each other.

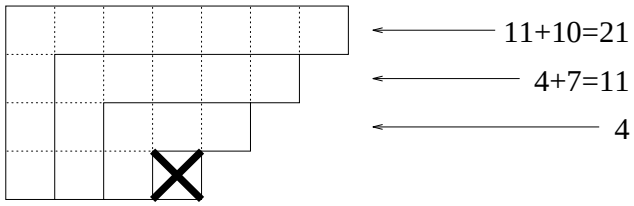

Figure 10: The three neighborhoods used in the simulations.

The solid graph in Figure 11 is a section of the (solid) lower bound in Figure 6. The other graphs describe the rate and distortion performance of the proposed encoder for the three neighborhood sizes, each parameterized by a growing size of estimator lookup table.

Note that if it were true that by inspecting causal neighborhoods one can accurately predict the value of d_{ij}, then all the occurrences of a given neighborhood would have corresponded to a single value of d, and the lower bound of Section 3 would have been reached. In practice all predominant neighborhoods have typical narrow d_{ij}-histograms, which get narrower as the size of the neighborhood increases. The width converges at a neighborhood size of 21 bits. The narrowest d_{ij}-histograms are however not narrow enough to be considered accurate

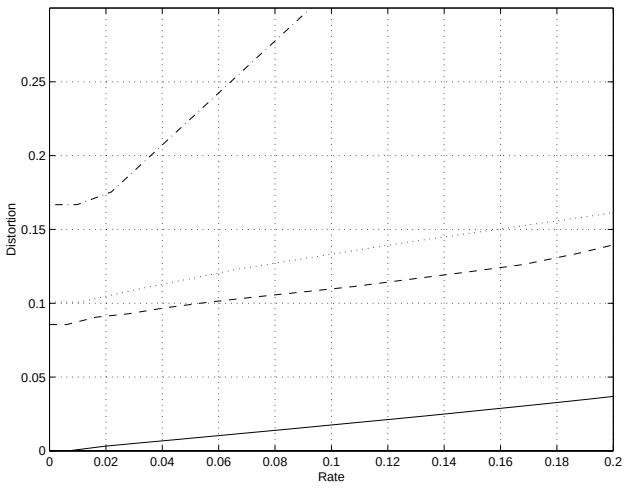

Figure 11: Distortion versus rate graphs: Solid - lower bound. Dash-dot - estimator size 4. Dot - estimator size 11. Dash - estimator size 21.

predictors of d_{ij}. This fact is responsible for the gap between the best estimator performance and the lower bound in Figure 11.

The performance of the proposed estimators does not converge to the lower bound, however since accurate prediction of d_{ij} is not possible the lower bound can not be reached and the estimators might yet be close to optimal. The proposed estimators might have been considered optimal if the following assumtions were true:

1. The statistics of unmarked halftones holds for watermark distorted halftones.
2. The database for the statistics was large enough.
3. The estimation can be based only on causal neighborhood information.

In practice non of the assumptions is true:

1. As already mentioned in Section 3 watermarking changes the statistics of the halftones, hence, the estimators are based on an analysis of the wrong family of halftones. A better estimation is expected with lookup tables compiled for the required rate. This was not done here since obtaining the optimal list for a given rate is a delicate feedback optimization, which is out of scope for this project.
2. The database represents well the statistics of small neighborhoods. For larger neighborhoods it might be insufficient. A larger database might improve the performance

of large-neighborhood estimation, and push the performance convergence higher than 21 bit neighborhoods.

3. The causal neighborhood is not the only perceivable simple information one can use for estimation. For example one can use information about the previous flip locations (so as to spread them evenly in the image). Figure 12 is an example of the possible advantage of that kind of information. The dash-dot graph in Figure 12 describes a non-estimator method using predefined locations that are randomly distributed, whereas the predefined locations for the dotted graph were evenly spaced in a blue noise distribution.

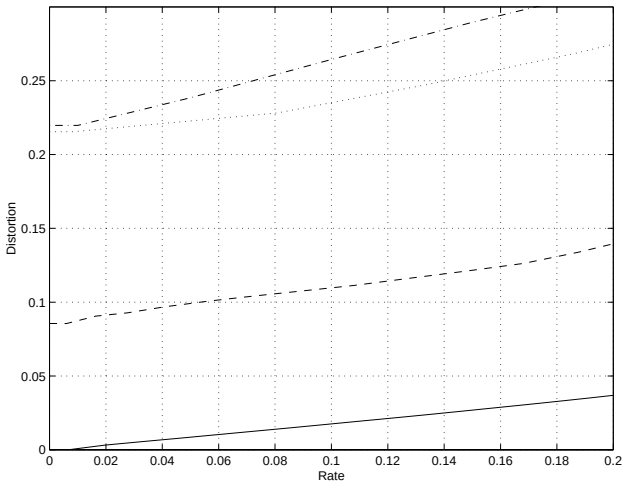

Figure 12: Distortion versus rate graphs: Solid - lower bound. Dash - estimator size 21. Dash-dot - Random locations. Dot - Evenly spaced locations.

To conclude the presentation of the proposed estimator note that:

- Comparing the performance of the 21bit estimator and the two non-estimator watermark encoders in Figure 12, positions it well between those and the lower bound.
- The proposed 21 bit watermark encoder predicts impressive watermark rates for a relative modest distortion, for example: Rate 0.01 (one bit in every 100 pixels) for a distortion of about 0.09 (an additive shot noise with $\sigma = 23$ gray levels on the flip locations - one in every 200 pixels).

6 Simulation Results

Up to this point, the analysis and the comparisons were done on a family of images describing an abstract image model. Its important to verify that the conclusions apply to real images. This is done in the following section.

Figure 13 shows the lower bound, and the performance curve of the 21 bit estimator. Those are derived from simulations based on an image model and were already presented above. Additional points marked by X's denote the rate and distortion of a watermark embedded in a real image by the proposed 21 bit estimator. Different points correspond to a varying size of estimator look up table, taken off the top of the neighborhood list produced for the natural image model. As can be seen the performance predictions based on the natural image model are relevant for real images.

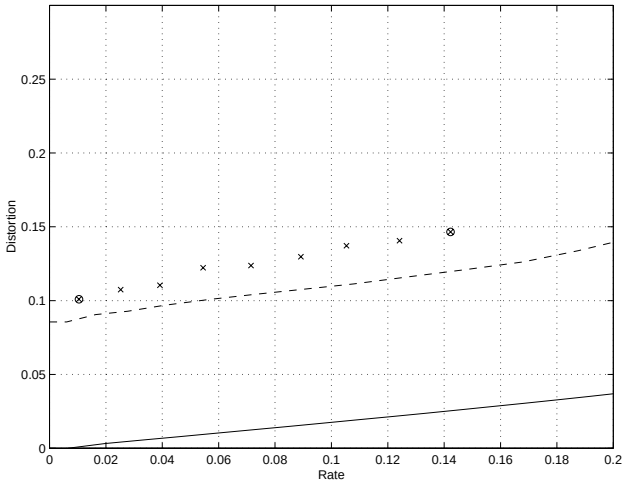

Figure 13: Distortion versus rate graphs: Solid - lower bound. Dash - estimator size 21. X marks - performance of estimator size 21 on a real image. Circles - instances presented in Figure 14.

Figure 14 presents a part of the Lena image, halftoned without a watermark, and with two rates of watermark (circled out in Figure 13). Each of the halftone images is rendered at 150dpi. In Figures 14b(c) the watermark encoder used an estimator lookup table of 1500(13,500) entries, and embedded 2,729(37,299) bits of watermark information, i.e. a bit for every 96.1(7.0) pixels, or a rate of 0.010(0.142). The standard deviation of d_{ij} at the 1,388(18,503) flip locations was 25.7(37.4) gray levels, or a distortion of 0.101(0.147) in the normalized scale used above.

The difference between Figures 14a and 14b is practically imperceptible, whereas the distor-

(**a**) (**b**) (**c**)

Figure 14: a) No watermark. b) Watermark Rate 0.010. c) Watermark Rate 0.142.

tion in in Figure 14c is visible. Another insight may be gained by comparing the watermarked halftones in Figure 15. All three images have a watermark rate of 0.04 (a bit for every 25 pixels). In Figure 15a the watermark was embedded by the proposed estimator (size 21), in Figure 15b the watermark was embedded in (blue-noise) evenly distributed predefined locations, and in Figure 15c the locations were chosen randomly. The respective distortion measures are 0.110, 0.200, and 0.204.

(**a**) (**b**) (**c**)

Figure 15: Watermark rate = 0.04: a) Estimator size 21. b) Evenly spaced locations. c) Random locations.

Note also that whereas the distortion in Figure 14c gives the impression of an additive White Noise, the Salt&Pepper artifacts of the distortion are more visible in Figures 15bc (because of the occasional really bad locations chosen by the non-estimator embedding methods).

7 Discussion

This discussion is devoted to some of the assumptions that were made in this report and to practical implementation notes.

- The absence of watermark distortion in the analysis, points at the most significant practical consideration regarding this research. It has been implicitly assumed that individual halftone dots may be identified in the printed image without any error. This is, however, not true. In practice printers might mess up the alignment, and cover up neighboring locations, in addition, several mechanical deficiencies may occasionally destroy halftone patterns. All this does not necessarily prevent watermarking, it simply calls for some practical implementation measures, mainly exploiting the vast amount of information bits for redundant coding. Practical considerations of that sort are, however, out of the scope of this report.
- Since the error type is Salt&Pepper rather than White Noise, it is not possible to characterize it properly in a single distortion term. The full characterization of the noise should include two terms: The density of the noisy locations (in our case half the rate), and the error variance at the noisy locations. Since we describe distortion versus rate (rate is the first term), we normalize the square distortion by the number of flip locations (thus obtaining the second term), rather than by the size of the image which is commonly done when assuming White Noise.
- The distortion measure (bundled with a few other assumptions) was checked in a (non-extensive) comparison of the visual quality of the alternative embedding routines of Figure 3, (see Figure 4). Nevertheless, there is room for improved measures especially by considering the frequency content of the error.
- The distortion measure is based on the distortion between two con-tone images: The original image and an image that would have caused the distorted halftone. As was pointed out in the text, there are many such images, and one needs to consider only the one with the smallest distortion. The minimization has been performed with an implicit assumption that the difference is causal, i.e, the difference occurrs at the flipped halftone location and possibly in "future" pixels. This assumption is intuitive and reasonable, but not necessary.
- Results obtained by the proposed image model apply for real images. There are however better models.

- The image model used in practice was a limited set of 500×500 images. This set is sufficient for most of the statistical test that were needed, however for large neighborhood sizes it may be stretching the limits. This fact may explain some of the results obtained for large estimator neighborhood sizes (general perfomance may be better, and specifically, convergence may occur at estimator sizes that are larger than obtained).
- The watermark embedding and decoding complexity has not been treated in this report. Necessarily larger neighborhoods as well as longer lookup tables require more computations.
- The lower bound is not tight. Being equivalent to a predictor justifies its use as a goal for practical embedding algorithms. Nevertheless, an alternative tight lower bound might yet exist.
- As mentioned in the text, compiling the estimator lookup tables is a delicate feedback optimization: Any choice of neighborhoods modifies the error statistics for the selected neighborhoods as well as others. The compromise made by borrowing lookup tables from the no-distortion statistics is probably responsible for a collapse of the estimator performance in high rates. This phenomenon was not presented in the graphs. It occurs only in rates that are out of the rate-range referred to in this report (those rates are not interesting for practical purposes).
- A better performance is possible if error correction codes are used in the watermark. Consider the possibility for the encoder to bluff occasionally, when it notices that embedding a certain watermark will introduce a very large error. If the decoder can corrects the resulting errors in the watermark it might be used to reduce the distortion (with a necessary reduction in the rate due to the error correction). The error correction possibility was not explored in this report.

8 References

[1] Z. Baharav, and D. Shaked, "Watermarking of Dither Halftoned Images", *HP Labs TR*, HPL-98-32, February 1998.

[2] W. Bender, D. Gruhl, N. Morimoto, A. Lu, "Techniques for Data Hiding", *IBM Systems Journal*, Vol. 35, pp. 313–335, 1996.

[3] J. T. Brassil, S. Low, N. F. Maxemchuk, L. O'Gorman, "Electronic Marking and Identification Techniques to Discourage Document Copying", *IEEE J. on Selected Areas in Communications*, Vol. 13, pp. 1495–1504, 1995.

[4] A. M. Bruckstein, and T. J. Richardson, "Holographic Transform Domain Image Watermarking Method", *Bell-Labs Technical Report*, 1997.

[5] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking Multimedia", in *IEEE Trans. on Image Proc.*, Vol. 6, pp. 1673–1687, 1997.

[6] I. J. Cox, and M. L. Miller, "A Review of Watermarking and the Importance of Perceptual Modeling", in *Proc. Electronic Imaging 97*, February 1997.

[7] R. Floyd and L. Steinberg. "An adaptive algorithm for spatial gray scale", in *SID Symposium*, pages 36–37, 1975.

[8] P. W. Wong, "Entropy-Constrained halftoning, Using Multipath Tree Coding", IEEE Trans. on Image Processing, Vol. 6, pp. 1567–1579, 1997.