Posterizing Effects can be Modeled by Point Operations

Yachin Pnueli  Alfred M. Bruckstein
Computer Science Department
Technion, Israel Institute of Technology
Haifa, Israel

March 31, 1992

Abstract

Of the available methods for digital halftoning, error diffusion algorithms are considered to provide the highest quality output. In these methods the error, produced by thresholding a given pixel, is diffused among its neighbors in some method-specific way. When only part of the error is thus diffused, the output looks sharper or more crisp. This effect, which is commonly called posterizing, (giving the output a poster like quality), was, up to now, not well understood and its merits could only be judged by heuristic methods.

In this note we claim that posterizing can be modeled as a combination of a space invariant point operation followed by the unpsterized error diffusion algorithm. We show analytical and experimental evidence supporting this claim in the one dimensional case and give experimental evidence indicating that this model can be extended to the two dimensional case as well.

1 Introduction

Digital halftoning algorithms (see [U1] for a comprehensive survey and discussion) transform a gray level picture, \( p_i \), \( 0 \leq p_i \leq 1 \) for every pixel \( i \), into a halftoned picture, \( h_i \), \( h_i \in \{0, 1\} \) where 1 is black and 0 white. Such algorithms have many applications in rendering gray-level pictures on mediums which support only binary signals, such as ink on paper or certain types of screen displays.

The algorithms collectively known as error diffusion are considered among the best available halftoning methods. In these algorithms, the earliest example of which is the Floyd and Steinberg algorithm [FS], the input picture is
scanned, pixel by pixel, in some fashion (typically raster scan) and the following operations are performed for every pixel $i$:

- Thresholding: $s_i$ - the current value of the pixel (which may be different from $p_i$ - its original value) is compared to a given threshold $\alpha$: If $s_i \geq \alpha$, $h_i = 1$, otherwise $h_i = 0$.

- Error Diffusion: $e_i$ - the error thus incurred ($e_i = s_i - h_i$), is spread among some of the yet unscanned neighbors of the pixel according to a method-specific rule.

The differences between the many such published algorithms lie either in the way in which the error is spread among the neighborhood of the given pixel (Jarvis et. al. [J], Ulichney [U1][U2] and others), or in the way in which the picture is scanned (Witten and Neal [WN] and Knuth [K], for example)\(^1\).

Of special interest is the approach pioneered in [WN] and recently improved and extended in the works of Velho and Gomes [V1], Cole [C] and Wyvill and McNaughton [WM]. In this approach the scan path is a space filling curve (Peano curve, Hilbert curve, or some composite scan) developed to the appropriate resolution. The curve is chosen so that the probability of making many steps in the same direction is small and thus the resulting halftone is devoid of directional artifacts marring raster scan methods. Given the scan path, the two dimensional halftoning problem, becomes one dimensional, with the additional advantages of reducing the memory resources needed and making parallelization easy.

A common feature to all the above algorithms is that at each diffusion step all the incurred error is spread. With each such algorithm, we can associate a posterizing algorithm with a posterizing factor $w$ ($0 < w < 1$), by spreading in each diffusion step only $w \times e_i$. The effects of this for the Floyd and Steinberg algorithm can be seen in figures 1, 2 and 3. Figure 1 is a natural scene and figure 2 a synthetic one both posterized using factors: 1.0 (no posterizing), 0.9, 0.75 and 0.5. Figures 3a and 3b show a halftoning ($w = 1.0$ and $w = 0.75$) of a synthetic picture composed of a white background, and squares colored with a gray level 0.5 plus a zero mean, equal distribution noise in the range ($-0.05, 0.05$). Figures 3c and 3d show a halftoning ($w = 1.0$ and $w = 0.75$) where the squares are at a constant gray level 0.48. In these figures the sharpening and Moiré suppression properties usually associated with posterizing can be readily seen.

In this note, which is a condensed version of [PB], our claim is that for every threshold value $\alpha$ and posterizing factor $w$ there is a function $t_{\alpha,w} : [0,1] \rightarrow [0,1]$ such that the effects of the posterizing algorithm, for every picture $[p_i]$, denoted by $ED_{\alpha,w}([p_i])$, can be modeled by $ED_{\alpha,1}(T_{\alpha,w}([p_i]))$ where $ED_{\alpha,1}$ is the unposterized version of the algorithm and $T_{\alpha,w}$ is a point operation replacing the value $p_i$ of each pixel with $t_{\alpha,w}(p_i)$. In the sequel we discuss this claim as follows:

\(^1\)Recently a few neural network variations of error diffusion have been published, see [KA] for example.
• In section 2, we prove it for the simple case of a one dimensional error diffusion algorithm operating on constant gray level pictures. We do this by showing an analogy to digital straight lines.

• In section 3, we give arguments as to why this result should hold in the more interesting case of a general picture. We use the Witten and Neal algorithm to give experimental support to this claim. We then give reasons as to why these results might extend to two dimensional algorithms as well and give experimental support (using the Floyd and Steinberg algorithm) indicating that our results hold.

• In section 4, we discuss the properties of $t_{\alpha,w}$ and how they produce the effects associated with posterizing.

2 The one dimensional constant gray level case

Let $[g]$ denote a constant gray level picture $[p_i]$ where $p_i = g$ - some gray level for every $i$.

**Theorem:** There is a function $t_{\alpha,w} : [0,1] \rightarrow [0,1]$, such that for every picture $[g]$ the pattern produced by $ED_{\alpha,w}([g])$ is the same as that produced by $ED_{\alpha,1}(t_{\alpha,w}([g]))$.

**Proof:** We use the following results:

- **Lemma 2.1:** The output of $ED_{\alpha,1}([g])$ has the same properties as the chain code of a digital straight line.

- **Lemma 2.2:** Every binary string having these properties (which we term a self similar string) can be interpreted as the output of $ED_{\alpha,1}([g'])$ for some constant gray level $g'$.

- **Lemma 2.3:** For any $0 \leq w < 1$ and $0 \leq \alpha \leq 1$, $ED_{\alpha,w}$ also produces self similar outputs when operating on a constant picture.

From lemma 2.3 we have that, for any $g$, $ED_{\alpha,w}([g])$ produces a self similar pattern. From lemma 2.2 we know that this pattern can be interpreted as the output of $ED_{\alpha,1}([g'])$ for some $g'$. By setting $t_{\alpha,w}(g) = g'$, the theorem follows.

It remains to prove these three lemmas. (For the sake of completeness we also show a fourth lemma 2.4 - that every self similar binary string can be interpreted as the output of $ED_{\alpha,w}([g'])$ for some constant gray level $g'$ and any $w$ and $\alpha$).

**Lemma 2.1:** $ED_{\alpha,1}([g])$ produces self similar patterns

Consider the operation of $ED_{\alpha,1}([g])$: At each step, $i$, the algorithm performs a threshold operation setting $h_i = 1$ if $s_i \geq \alpha$ and $h_i = 0$ otherwise. It then
performs the diffusion, setting \( s_{i+1} = p_{i+1} + e_i = g + (s_i - h_i) \). (To make the analogy to digital straight lines clearer we introduce \( e_0 \) - a zeroth error term such that \( s_1 = g + e_0 \).) Recursively substituting the above equation into itself we get:

\[
s_i = i \times g + e_k - \sum_{j=k+1}^{i-1} h_i = i \times g + e_0 - \sum_{j=1}^{i-1} h_i = i \times g + e_0 - n
\]

where \( n \) is the number of "1"s in the output signal so far.

This is exactly the process we perform when digitizing a straight line (with a slope between 0 and \( \pi/2 \)): For any grid point \( i \), we set the next symbol to 1 if \( ia + b \) is more than a distance \( \alpha \) from the line \( y = n \) and to 0 otherwise\(^2\). Since the process \( ED_{\alpha,1}([g]) \) is exactly the same as the digitization of the straight line \( y = g \times x + e_0 \), obviously, the output displays all the properties associated with digital straight lines.

There has been much research into these properties and to the question of whether a given binary sequence can be interpreted as a digital straight line. We refer the reader to [Br] for a recent work in the field which includes a survey of previous research as well. We also refer the reader to the works of Hung [Hu] and Wu [W] which we use in our proofs. In the sequel we call a string self similar (a term which we adopt from [Br]) if it can be interpreted as a digital straight line.

**Lemma 2.2**: Every self similar pattern can be produced by \( ED_{\alpha,1} \)

Let \( L \) be a self similar pattern. Using results from digital straight lines, \( L \) can be interpreted as the output of a digitization of some straight line \( y = a \times x + b \).

From the previous subsection, it can, therefore, also be interpreted as the output of \( ED_{\alpha,1}([g]) \) where \( g = a \), provided that the zeroth error term \( e_0 = b \) (since the choice of an origin does not influence the chain code of a straight line we can always find such a \( b \) in the range \([0,1])\).

The addition of an \( e_0 \neq 0 \) term might seem disturbing since we cannot claim that the exact \( L \) is produced by an \( ED_{\alpha,1} \) algorithm (for which \( e_0 = 0 \)), however this is not an actual problem: It can be shown that the patterns produced when digitizing straight lines with the same \( a \) and different \( b \)'s are identical up to a shift of some pixels to the left or right. This together with the fact that in self similar patterns "successive occurrences of a symbol are as uniformly spaced as possible\(^3\) means that two such patterns can be regarded as the same for the purpose of halftoning (where we are not interested in the value of a specific pixel but rather in the overall effect of the pattern).

\(^2\)Typically, both in error diffusion and the digitization of straight lines, only the values 0, 0.5 and 1 are used for \( \alpha \), although in principle any \( 0 \leq \alpha \leq 1 \) can be used.

\(^3\)Quoted from Freeman in [F].
Lemma 2.3: \(ED_{\alpha, w}([g])\) (0 \(\leq w < 1\)) produces a self similar pattern

We define \(\Phi(n) \equiv g \times \frac{1-w^n}{1-w}\) and \(\Psi_j^n \equiv \sum_{i=j}^{n} h_i \times w^{n-i+1}\). By continued resubstitution of the recursion formulae for \(ED_{\alpha, w}([g])\) we get that:

\[s_n = \Phi(n-i) + e_i w^{n-i} - \Psi_{i+1}^{n-1} = \Phi(n) = e_0 w^n - \Psi_1^{n-1}\]

Since the above formula is different from the formula for \(s_n\) of \(ED_{\alpha, 1}\) we cannot use direct arguments from digital straight lines and hence take a different approach.

Let \(L\) be a non-self-similar string which is presumably the output of \(ED_{\alpha, w}([g])\) for some values \(\alpha, w\) and \(g\). From [Hu] we have that \(L\) must contain an uneven pair (to be defined) and so (also from [Hu]) must contain at least one uneven pair in a normal form (also to be defined). We show that the constraints imposed by this uneven pair in normal form on the possible values of the different \(s_i\)'s cannot be satisfied and hence such a string could not be generated by \(ED_{\alpha, w}([g])\).

Notation: For \(X\) a finite binary string, let \(S(X)\) denote the number of “1”s in it and \(|X|\) the total number of symbols in it.

Definitions: We say that a binary string \(L\) contains an uneven pair, if there are \(L_1\) and \(L_2\) substrings of \(L\) such that \(|L_1| = |L_2|\) but \(S(L_1) > S(L_2) + 1\).

We say that \(L\) contains an uneven pair in normal form, if there is a string \(X\), such that both “0X0” and “1X1” are substrings of \(L\) and the portion of \(L\) between these two substrings can be partitioned into a whole number of segments of the form “1X0” (with the “1” facing the side of “0X0”).

Lemma 2.3.1: If \(L\) contains an uneven pair it also contains an uneven pair in normal form.

Proof: Follows if we first consider the uneven pairs with the minimum number of symbols and then among these the uneven pairs such that the number of symbols between the two substrings of the pair is minimum. (see appendix for complete proof).

Lemma 2.3.2: \(ED_{\alpha, w}\) cannot generate an unequal pair in normal form.

Proof: Let \(L_1 = “0X0”\) and \(L_2 = “1X1”\) be an uneven pair in normal form of \(L\) such that \(i+1\) is the position in \(L\) of the first symbol of \(L_1\), \(l = |L_1| = |L_2|\) and \(m\) is the number of times the sequence “1X0” appears between \(L_1\) and \(L_2\). (we assume that \(L_1\) is before \(L_2\), the complimentary is similar).

If \(L\) is the output of \(ED_{\alpha, w}([g])\) then the following constraints must hold:

\[s_{i+t} = \Phi(l) + w e_i - \Psi_{i+1}^{l+1-1} < \alpha\]
\[s_{i+ml+t} = \Phi(ml + l) + w^{ml+l} e_i - \Psi_{i+1}^{ml+l+1-1} < \alpha\]
\[s_{i+ml+2l} = \Phi(ml + 2l) + w^{ml+2l} e_i - \Psi_{i+1}^{ml+2l-1} \geq \alpha\]
Since the uneven pair is normalized, for every $j$ in the range $i < j < i + ml + l$ we have $h_j = h_{j+i}$ with the following exceptions:

1. $h_{i+k} = 0$ but $h_{i+k+1} = 1$ for $m \geq k > 0$.

2. $h_{i+(m+1)l} = 1$ but $h_{i+k} = 0$ for $0 \leq k < m + 2$.

Therefore:

$$\Psi_{i+ml+l}^{i+ml+l+1} = w \Psi_{i+ml+i+1}^{i+ml+i+1} + w^{i+ml+i+1} + \Psi_{i+ml+i+1}^{i+ml+i+1} =$$

$$= 0 + w^{i-l-1} + \Psi_{i+l}^{i+l-1} = w^{i-l-1} + \Psi_{i+l}^{i+l-1} < 0 > w^{i} + \Psi_{i+l}^{i+l-1}$$

and hence:

$$\Psi_{i+ml+l}^{i+ml+l+1} = \Psi_{i+ml+i}^{i+ml+i} + w^{i} \Psi_{i+ml+i}^{i+ml+i} > w^{i} + \Psi_{i+ml+i}^{i+ml+i} + w^{i} \Psi_{i+ml+i}^{i+ml+i}$$

Substituting into the expression for $s_{i+ml+2l}$ and using the relation $\Phi(n + m) = \Phi(n) + w^{n} \Phi(m)$ we get:

$$s_{i+ml+2l} = \Phi(ml + 2l) + w^{i+ml+2l} e_i - \Psi_{i+l}^{i+l-1} < i$$

$$< \Phi(l) - w^{i} - \Psi_{i+l}^{i+l-1} + w^{i} (\Phi(ml + l) + w^{i+ml+i} e_i - \Psi_{i+l}^{i+l-1})$$

Using the constraint on $s_{i+ml+i}$, the last term is less than $\alpha$ and so

$$s_{i+ml+2l} < \Phi(l) - w^{i} - \Psi_{i+l}^{i+l-1} + w^{i} \alpha = \Phi(l) - \Psi_{i+l}^{i+l-1} + w^{i} (\alpha - 1)$$

Using the fact that $\alpha - 1 \leq e_i$ (see lemma 2.3.3 in the appendix) and the constraint on $s_{i+l}$:

$$s_{i+ml+2l} < \Phi(l) + w^{i} e_i - \Psi_{i+l}^{i+l-1} < \alpha$$

which contradicts the original constraint on $s_{i+ml+2l}$.

Therefore, there are no $\alpha$, $w$ and $g$ such that $ED_{\alpha,w}([g])$ produces a non self similar string $L$ as its output.

Lemma 2.4: Every self similar pattern can be produced by $ED_{\alpha,w}$ ($0 \leq w \leq 1$).

Let $L$ be a self similar binary string. We wish to show that for every $\alpha$ and $w$ there are $e_0$ and $g$ such that $ED_{\alpha,w}([g])$ with initial error $e_0$ produces $L$.

Definitions: Let $X^i$ denote $i$ concatenations of the string $X$ to itself. A string $K$ is a self similar cycle of string $L$, if $K^i$ is self similar for every $i$ and $L$ is a substring of $K^i$ for some $i$.

Proof: In lemma 2.4.1 of the appendix we show that if $L$ is self similar and finite then there is a $K$ which is a self similar cycle of it. It is therefore sufficient to show that $ED_{\alpha,w}$ can generate $K$ for some $g$ and $e_0$ with the additional
condition that the error produced when the last symbol of $K$ was generated is equal to $e_0$. The additional condition guarantees that $ED_{\alpha,w}$ can generate any number of concatenations of $K$ to itself and hence generate $L$.

To prove this we view the self similar cycle as a set of $|K|$ constraints which $g$ must satisfy for a given $\alpha$ and $w$ and show in lemma 2.4.3 of the appendix that they can be satisfied simultaneously.

3 The case of nonuniform images

The simplest approach to analyzing the general case would have been to try and show that if $ED_{\alpha,w}$, when given an initial error $e_0$ and a pixel with gray level $p_1$, produced an $h_1$ and $e_1$, then there is a $t_{\alpha,w}$ such that $ED_{\alpha,1}$ with $e_0$ operating on $t_{\alpha,w}(p_1)$ produces the same $h_1$ and $e_1$. This however leads to an equation of the form $t_{\alpha,w}(p_1) = p_1 + (w-1)e_0$ which cannot be satisfied if $t_{\alpha,w}$ is to remain independent of $e_0$. Therefore any analysis we do must deal with whole sequences of input picture, and hence must include a wide range of possible cases.

However, it is quite possible that the outputs of $ED_{\alpha,w}([p_1])$ and $ED_{\alpha,1}(T_{\alpha,w}([p_1]))$ do not produce identical outputs, but rather the same patterns of output. In this case we are faced with a more basic difficulty of defining when patterns should be considered “the same”. A way out of this difficulty could be to find a good characterization of the produced outputs (such as the self-similarity property), unfortunately no such characterization was found. We shall however explain why we expect our claim to be a useful model for this case and support this argument with experimental results.

Our argument is based on the analysis of what happens to a constant gray level pattern, if at some point in the sequence we add a disturbance by replacing $p_1 = g$ with $p_1 = g + \delta$, where $\delta$ is some small value. It is easy to see that such a disturbance decays exponentially (at a rate determined by $w$), and hence either has no effect at all or else this effect is localized at the disturbed pixel and its first few neighbors (localized in the sense that a few pixels away from the disturbance the pattern is the same as the original except possibly that it is shifted a few pixels to the left or right).

To get a feeling of what happens consider the case of $g = \alpha = 0.5$ which generates the pattern 1010101010... For a $w$ close to 1, a small negative disturbance at the third pixel changes the output pattern to 1010101010.... Note that, although the specific output value of all pixels after the third have changed as a result of the disturbance, the pattern is the same except at the third pixel itself.

If we view the input picture as a set of constant gray level patches connected by edge zones, we can claim that inside a constant level patch our model holds as proven and in the edge zones we can model the transition from one patch to the other as a disturbance on the values of the initial pixels in the patch. As we argued above the effects of such disturbances are limited to the edge zones only.
and hence are insignificant when we consider the global picture. This is especially true since some deviation from any constant pattern is expected to happen at the edge anyway, and the exact nature of this deviation is unimportant as long as the eye interprets it as an edge.

It is still left to show that such a view is consistent with actual pictures. We believe it is. When we consider "strong" edges (were the difference in gray level between adjacent pixels is large), we usually find that the number of such pixels is negligible (as can be seen in figure 4a), and when we consider "weak" edges, although it is no longer true that the edge pixels are a negligible fraction of the picture (as can be seen in figures 4b and 4c), most of them have disturbances which are too small to be noticed.

Further support for this model comes from the dependence of $t_{\alpha,w}$ on $w$ (discussed in section 4). We show there that as $w$ approaches 1, $t_{\alpha,w}$ approaches the identity function. Hence even though the rate of decay of a disturbance slows as $w$ approaches 1, the total effect in this range remains small, since the difference between $ED_{\alpha,w}$ and $ED_{\alpha,1}(T_{\alpha,w})$ becomes less significant.

Finally, to give support to our claim we present experimental results. We use the recently published "Geof - scan" algorithm [WM] algorithm. This algorithm scans the input using a simple type of space filling curve and is thus a one dimensional error diffusion algorithm. We halftone a picture using this algorithm with posterizing factors of 1, 0.9, 0.75 and 0.5 (figures 5a-d). Then we experimentally calculate the transformations $t_{\alpha,w}$ (see next section) for these posterizing factors and halftone the picture using $ED_{\alpha,1}(T_{\alpha,w})$. The results can be seen in figures 6a-c. We believe they convincingly demonstrate our claim.

Following our success in the one dimensional case, we were tempted to extend our results to the two dimensional case (which includes most useful error diffusion algorithms). Unfortunately, as we could not find the two dimensional equivalent of the self similarity property, we cannot give any analytical support to our claim even in the case of constant gray level pictures.

However, since qualitatively the effects of posterizing in two dimensions seem to be the same as those of one dimension (compare figures 1a-d and 5a-d) we thought it worth while to try our model experimentally even without direct analytic support. We used the Floyd Steinberg algorithm. As in the one dimensional case we let it run with different posterizing factors on constant gray level pictures and used these results as the experimental values of $t_{\alpha,w}$. We then invoked $ED_{\alpha,1}(T_{\alpha,w})$ on the input picture. The result can be seen in figures 7a-c. We believe that these pictures when compared to the corresponding pictures in figures 1a-d are close enough to support our claim that the effects of an $ED_{\alpha,w}$ algorithm (even if not identical) can at least be modeled by an $ED_{\alpha,1}(T_{\alpha,w})$ algorithm.
4. The properties of $t_{\alpha,w}$ - the transformation function

We present the following results: Experimental results (plots of $g$ versus $t_{\alpha,w}(g)$) for the one dimensional case, analytical results for the one dimensional case and experimental results for the Floyd and Steinberg case.

4.1 Experimental results - 1D

Figures 8a-c are graphs of $g$ versus $t_{\alpha,w}(g)$ for different values of $w$ and a constant $\alpha$. Figures 8d-f are graphs for different values of $\alpha$ and a constant $w$. From these figures we can see the dominant properties of $t_{\alpha,w}$:

1. The very light gray levels get transformed to 1 and the very dark to 0.
2. In between, the transformation is shaped as a type of unequal quantization.
3. At a finer level we see, however, that there are no vertical "steps" in the function (this can be proven analytically form the fact that every self similar pattern has a corresponding gray level for any $\alpha$ and $w$).
4. The plateaus of $t_{\alpha,w}$ are larger for values of $g$ which are simple fractions with small denominators.
5. The transformation gets more pronounced for smaller $w$ (at $w = 0$ it is a threshold step function) and approaches identity as $w$ approaches 1.
6. For the same $w$ and different values of $\alpha$, the graph looks the same but gets shifted to the left or to the right.

The first four observations can be summed up by observing that $t_{\alpha,w}$ can be regarded as a special case of the Devil's staircase fractal [M]. We state the properties of this fractal more rigorously in the next sub-section. Here we describe only how its observed properties as outlined above explain the effects associated with posterizing:

Pushing the very low and very high levels to 0 and 1 respectively as seen in observation 1 has the effect of increasing the slope of the transition graph and so enhancing the sharpness of the picture (see the "scale adjustment graph" in [U1] chapter 1, for example - figure 9 is a sketch of this graph). Also it is well known that sharpness can be achieved in halftoning if we reduce the number of distinct gray levels which we try to render as seen in observation 2: as the distance between consecutive input gray levels grows we have to scan less output pixels.

---

4. this fact is somewhat surprising since it indicates that $\alpha$ has an effect on the output when posterizing, where as it is well known that it has no effect on regular halftoning (specific pixels do change values but the overall picture looks the same).
in order to detect a change from one "rendered gray level" to another (there are analogous results for digital straight lines as well).

The third observation is the one that reduces the formation of false contours, as compared to other methods which also create sharpness effects (such as bit reduction via threshold methods) since although the transition from one rendered level to the next is steep - it is never completely vertical.

The fourth observation which maps similar values of \( g \) to the same value \( t_{\alpha, w}(g) \) and which is strongly biased towards gray levels which have simple rendering patterns is responsible to the Moiré suppression, since many times Moiré patterns can be traced either to a slight noise added on top of a constant gray level or to a gray level which cannot be rendered on the pixel grid without aliasing effects (see figure 3). This property, however, also explains an undesirable property of too much posterization (aside from the obvious - too sharp and hence too flat appearance): Since the transformation "suppresses" more complex patterns and enhances the "simple" patterns it has a tendency to prefer simple repetitive textures over untextured "Blue noise" patterns \(^5\) and since these are usually more desirable, too much posterizing can reduce the quality of the output.

The fifth observation is just a statement of the fact that \( w \) is the posterizing factor and hence its value determines the strength of the transformation.

The sixth observation is interesting in the sense that if we want to stress or amplify a certain range of the histogram of the picture we can do so by using the appropriate threshold \( \alpha \) (of course this effect becomes more pronounced as \( w \) gets smaller). However if we accept a common image processing paradigm that in a "good" picture the center of gravity of the histogram is at 0.5, this effect becomes meaningless. Another way of saying the same thing is that we can perform a histogram balancing operation on the picture before halftoning and then disregard this effect and use only \( \alpha = 0.5 \).

4.2 Analytical results - 1D

In our study of \( t_{\alpha, w} \), we calculate the values of \( g \) for which the output of \( ED(\alpha, w) \) produces certain patterns and hence \( t_{\alpha, w}(g) \) should equal to the gray level for which \( ED(\alpha, 1) \) produces these patterns. We move from simple patterns to more complex. We complete the analytical picture by stating the results of \([FC]\) which give analytically the global properties of \( t_{\alpha, w}(g) \).

The simplest patterns are cyclic patterns of only "0" or "1". Simple calculations on geometric series show that \( t_{\alpha, w}(g) = 0 \) for every \( g \) such that \( g < (1 - w)\alpha \) and \( t_{\alpha, w}(g) = 1 \) for every \( g \) such that \( g \geq (1 - w)\alpha + w \). This also gives us \( w \) as the length of the entire subrange of the gray scale on which \( t_{\alpha, w} \) produces nontrivial outputs.

The next more complex patterns, are patterns of \( k - 1 \) zeros followed by a

\(^5\) A term adopted from Ulrich [12].
one. To find which gray levels produce such an output pattern for \( ED_{\alpha,w} \), we solve the following set of equations:

For every \( i \) such that \( nk + 1 \leq i \leq nk + k - 1 \):

\[
\Phi(i) + w^i e_{nk} < \alpha
\]

and for \( nk + k \):

\[
\Phi(i) + w^k e_{nk} \geq \alpha.
\]

We use the same technique as in section 2.4 of setting \( e_0 = e_{nk} \) to guarantee cyclic behavior and get the solution:

\[
(1 - w)\alpha + (1 - w) \frac{w^k}{1 - w^k} \leq g < (1 - w)\alpha + (1 - w) \frac{w^{k-1}}{1 - w^k}
\]

Thus, any \( g \) within this range produces a cyclic pattern of \( k - 1 \) zeros followed by a one.

If we take a yet more complex pattern, such as the \( l - 1 \) appearances of a pattern of \( k - 1 \) zeros followed by a one, followed by a single appearance of \( k \) zeros followed by a one, we get a solution of the following form:

\[
(1 - w)\alpha + \frac{w^k(1 - w)(1 - w^{k+l})}{(1 - w^k)(1 - w^{k+l})} \leq g < (1 - w)\alpha + \frac{w^k(1 - w)(1 - w^{k+l+k})}{(1 - w^k)(1 - w^{k+l+k})}
\]

Similar results can be obtained by recursively looking at more complex patterns (the self similarity property guarantees that each step of the analysis is identical except that the element with which we are dealing is a string and not an individual symbol). However, for extracting the global properties of \( t_{\alpha,w} \), we can directly use the recent results of Feely and Chua [FC]. They analyze exactly the same system in the totally different context of \( \Sigma - \Delta \) modulation with integrator leak: Feely and Chua show that for every cyclic self similar pattern \( P \) of length \( L \) there is a segment \([a, b] \subset [0, 1]\) with

\[
b - a = \frac{2w^{L-2}(1 - w)^2}{1 - w^L}
\]

such that \( P \) can be obtained by \( ED_{0.5,w}(g) \) for every \( g \in [a, b] \).

From these results we can extract the following properties of \( t_{\alpha,w} \) which are the characteristics of our special variant of the Devil's staircase:

- \( (1 - w)\alpha \) acts as an origin of the transformation which is otherwise independent of \( \alpha \) - this is consistent with observation 6 of the previous section.

- \( t_{\alpha,w} \) maps segments of \([0, 1]\) into distinct points in \([0, 1]\).

- The length of a segment which is mapped to a given point \( g \) is exponentially dependent on the complexity of the pattern produce by \( ED_{0.5,1}(g) \). This has long been known to be related to the continuous fraction expansion of the value of \( g \) (see [Ch] or [S] for example). A direct outcome of this observation is that the largest values are associated with \( g \)'s which are simple fractions with small denominators such as: 0, 1, 1/2, 1/3,
2/3, 1/4, 3/4, 1/5, 2/5 etc. corresponding to the cyclic patterns "0", "1", "01", "001", "110", "0001", "1110", "00001", "01001", etc. This property is consistent with observation 4 above.

- Note that a consequence of this is that only a single point is mapped by $t_{a,w}$ to each irrational number. Since the continuous fraction expansion of an irrational is infinite - the pattern it produces is non-cyclic and hence only a segment of infinitesimal size is mapped to it.

### 4.3 Experimental results - 2D

Figures 10a-c are graphs of $g$ versus $t_{a,w}$, for different values of $w$ and a constant $a$, for the Floyd Steinberg algorithm. Apart from the one obvious difference that these transformations are not monotonic at certain places, the graphs appear very similar to the graphs of the one dimensional case. This is consistent with the observation that posterization has qualitatively the same effect in both cases.

At first glance, the nonmonotonicity of the transformation appears to be a computational mistake, since it implies that we can find two gray levels $g_1$ and $g_2$ such that $g_1 < g_2$ but the output of $ED_{a,w}(g_1)$ contains more white pixels than $ED_{a,w}(g_2)$, however this can actually happen:

For example, taking the gray levels $g_1 = \frac{23}{55}$ and $g_2 = \frac{24}{55}$ and a picture of size $512 \times 512$, we find that for $g_1$ we have 45,620 "1" pixels and for $g_2$ only 43,435.

The reaction to this nonmonotonicity can be either to claim that it is an integral part of posterizing in two dimensions\(^6\) and try and find some useful features of it, or else to view it as an unwanted side-effect of the algorithm we use. In this case if we precede the unposterized error diffusion with a point operation in which the nonmonotonicities where smoothed out, we should expect to obtain better looking outputs than the "raw" posterized versions. Figures 11a-b are such outputs, and it is possible to claim that they do look better, although they are far from being conclusive evidence.

### 5 Discussion

There is a trend in the field of halftoning algorithms to try and analyze algorithms in the form of some picture processing operations followed by a neutral halftoning algorithm. This trend has the advantage that it allows us to integrate knowledge from other fields of picture processing when choosing a halftoning strategy. We showed in this note that the effects of posterizing in the one dimensional case fall neatly into this trend and can be interpreted as a space invariant point operation followed by the unposterized algorithm. We

\(^6\)At least is not specific to the Floyd Steinberg algorithm as it remains when we replace its filter of $7/16, 1/16, 5/16, 3/16$ with the more symmetric $3/8, 1/4, 3/8$ filter.
discussed some of the properties of this point operation and showed how they explain the effects of posterizing. We also found experimentally that the results can be extended (at least as a conceptual model) to the two dimensional case as well. An interesting extension of this work would to be to check whether a transformation with the following properties would produce better results than posterizing in the two dimensional case:

- The transformation is monotonic
- The size of the interval $(g_1, g_2)$ which gets transformed to $g$ depends exponentially on the 2-D area of the pattern used by $EU_{a1}$ to render $g$.

Positive results would indicate that the properties outlined for the one dimensional case are indeed fundamental to the posterizing effect, besides giving us a better "posterized" halftoning algorithm.

An interesting side effect of our research was the exposition of the wide applicability of the error diffusion equations or algorithm. In this note we showed the intimate relation between the unposterized error diffusion algorithm and the chain code of digital straight lines. While we prepared this version of the manuscript the works of Gray [G] and Feely and Chua [FC] came to our attention showing the applicability of the exact same algorithm in the field of analog to digital conversion. This was especially interesting since the phenomena of leaky integrator in $\Sigma - \Delta$ modulation (which is a result of the nonideal behavior of circuits) is the exact counterpart of error diffusion with posterizing (which is a design parameter for the halftoning algorithm). It seems that many of our results were independently developed there using a different point of view and motivation.

References:


Appendix

Lemma 2.3.1: If \( L \) contains an uneven pair then it also contains an uneven pair in normal form.

Proof: First we prove that there is an uneven pair of the form: "0X0" and "1X1" and then we prove that there is such an uneven pair for which the substring between the members of the pair can be partitioned into a whole number of segments of the form "1X0" (with the "1" facing the side of "0X0").

Let \( L_1 = (s_1, s_2, \ldots, s_m) \) and \( L_2 = (t_1, t_2, \ldots, t_m) \) be the uneven pair such that \( |L_1| = |L_2| \) is minimum and \( S(L_1) > S(L_2) + 1 \). Clearly \( s_1 \) must be a "1" and \( t_1 \) a "0" otherwise \( (s_2, \ldots, s_m) \) and \( (t_2, \ldots, t_m) \) would form a shorter uneven pair. By a similar argument \( s_m \) must be a "1" and \( t_m \) a "0".

Let \( 1 < i < m \) be the first index for which \( s_i \neq t_i \). If \( s_i = \text{"1"} \) then \( (s_1, \ldots, s_i) \) and \( (t_1, \ldots, t_i) \) form a shorter uneven pair. If \( s_i = \text{"0"} \) then \( (s_i, \ldots, s_m) \) and \( (t_i, \ldots, t_m) \) form a shorter uneven pair.

Consider now only minimal uneven pairs as above and let \( L_1 = (s_1, s_2, \ldots, s_m) \) and \( L_2 = (t_1, t_2, \ldots, t_m) \) be an uneven pair such that the size of the substring separating them in \( L \) is minimal. If \( L_1 \) is adjacent to \( L_2 \) then we are done (if they overlap, we can parse them into \( T_1 \), \( IT \) and \( T_2 \) were \( IT \) is the part that overlaps and then \( T_1 \) and \( T_2 \) form an uneven pair contradicting the minimality of \( L_1 \) and \( L_2 \)).

It is left to show that if the sequence of symbols between them is not in the required format then the length from the first symbol of \( L_1 \) to the last symbol of \( L_2 \) cannot be minimum. Consider a sliding window of length \( m \), where \( m \) is the number of symbols in the uneven segments. We start with this window located on \( L_1 \) and move it one symbol at a time towards \( L_2 \). At the first step our window drops a "1" and picks up the symbol following \( L_1 \). This symbol must be a "0" otherwise our window and \( L_2 \) form a closer minimal uneven pair.

After the first step, in every step but the last our window must drop on the left the same symbol it picks up on the right: Otherwise if it drops a "0" and picks up a "1" then the window and \( L_2 \) form a closer uneven pair, and if it drops a "1" and picks up a "0" then the window and \( L_1 \) form a closer uneven pair.

Lemma 2.3.3: For any \( i : \alpha - 1 \leq e_i < \alpha \).

Proof: The lemma is trivially true for \( i = 0 \) (since \( e_0 = 0 \)). Assume it is true for every pixel up to some value \( j \), then

\[ w(\alpha - 1) + g \leq s_j < wa + g \]

If \( s_j \geq \alpha \) then \( e_j = s_j - 1 \) and hence

\[ \alpha - 1 \leq e_j < wa + g - 1 < \alpha \]

else \( s_j < \alpha \) and then \( e_j = s_j \) and hence

\[ \alpha \geq e_j \geq w(\alpha - 1) + g \geq \alpha - 1 \]
Lemma 2.4.1: If \( L \) is self similar and finite then there is a \( K \) which is a self similar cycle of it.

Proof: We follow the work of Wu. In [W], Wu presents and algorithm for testing the self similarity of strings. His algorithm recursively groups symbols of the string into super symbols and succeeds only when it is left either with a single super symbol or with any number of identical super symbols. It is easy to see that when his algorithm succeeds we can ungroup its super symbols recursively and construct a \( K \) as required by this lemma. Since \( L \) is self similar and so must pass Wu's algorithm, such a \( K \) must exist.

Technical Lemma 2.4.2: Let \( L = (s_1, s_2, \ldots, s_{\text{final}}) \) be a self similar string. Let \( L_1 = (s_i, s_{i+1}, \ldots, s_{i+1}) \) and \( L_2 = (s_j, s_{j+1}, \ldots, s_{j+1}) \) be substrings of it such that \( s_{i+1} = 0 \) and \( s_{j+1} = 1 \) and \( L_1 \) differs from \( L_2 \) by at least one symbol. Let \( x_0 \leq l \) be the minimal number such that \( s_{i+1} - x_0 \neq s_{j+1} - x_0 \). Then, \( (1 - w)w^{x_0} \leq \psi^{i+1}_l - \psi^{j+1}_l \).

Proof: Clearly, \( s_{i+1} - x_0 = 1 \) otherwise the segments \((s_{i+1} - x_0, \ldots, s_{i+1})\) and \((s_{j+1} - x_0, \ldots, s_{j+1})\) form an uneven pair. If \( x_0 = l \) or the remainder \((s_i, \ldots, s_{i+1} - x_0) = (s_j, \ldots, s_{j+1} - x_0) \) then

\[
\psi^{i+1}_l - \psi^{j+1}_l = w^{x_0} \geq (1 - w)w^{x_0}
\]

and our lemma is proved.

Otherwise let \( y_0 \) be the next minimal number such that \( s_{i+1} - y_0 \neq s_{j+1} - y_0 \). Now \( s_{i+1} - y_0 = 0 \) else we have an uneven pair in \((s_{i+1} - y_0, \ldots, s_{i+1})\) and \((s_{j+1} - y_0, \ldots, s_{j+1})\). Therefore:

\[
\psi^{i+1}_l - \psi^{j+1}_l = w^{y_0} - w^{y_0} + w^{y_0 + 1} (\psi^{i+1}_l - \psi^{j+1}_l)
\]

If \( y_0 = l \) or the remaining parts of \( L_1 \) and \( L_2 \) are identical, we are done (since the last term is then zero). Else there is an \( x_1 \) which is the minimal number such that \( s_{i+1} - x_1 \neq s_{j+1} - x_1 \), and we can apply the same argument above recursively, until eventually we reach the left end of \( L_1 \) and \( L_2 \). We get that

\[
\psi^{i+1}_l - \psi^{j+1}_l = \sum_{k=0}^{n} (w^{x_k} - w^{y_k}) \geq \sum_{k=0}^{n} (w^{x_k}(1 - w)) \geq w^{x_k}(1 - w)
\]

Since each \( x_k \) is at least \( x_k + 1 \).

Lemma 2.4.3: Let \( K \) be a self similar cycle of the self similar string \( L \), then the set of \( k = |K| \) constraints it imposes on \( g \) (assuming that \( K \) was generated by \( E D_{\alpha, w}((g)) \) for a given \( \alpha \) and \( w \)) can be satisfied simultaneously.

Proof: By using for \( e_0 \) a value such that \( e_0 = e_k \) we guarantee that every \( i \)th constraint is identical to the \( i + k \)'th constraint. We can therefore prove the lemma for the set of constraints implied by the \( n \)th appearance of \( K \) for any \( n \) we choose. Consider such a set of constraints:

\[
s_{nk+1} = \Phi(nk + 1) + w^{nk+1}e_0 - \psi^{nk}_1 (<, >) \alpha
\]
\[ s_{n+k+i} = \Phi(nk+i) + w^{nk+i}e_0 - \Psi_{1}^{nk+i+1} (\leq, \geq) \alpha \]

\[ s_{n+k+k} = \Phi(nk+k) + w^{nk+k}e_0 - \Psi_{1}^{nk+k} (\leq, \geq) \alpha \]

where \( (\leq, \geq) \) indicates \( \leq \) or \( \geq \) depending on the value of \( h_{n+k+i} \).

Substituting for \( e_0 \) the value

\[ e_0 = \frac{g}{1-w} - \frac{\Psi_{1}^{k}}{1-w^k} \]

which is the solution to the equation \( e_0 = e_k = \Phi(k) + w^k e_0 - \Psi_{1}^{k} \) we get:

\[ s_{n+k+1} = \Phi(nk+1) + w^{nk+1} \left( \frac{g}{1-w} - \frac{\Psi_{1}^{k+1}}{w(1-w^k)} \right) - \Psi_{1}^{nk} (\leq, \geq) \alpha \]

\[ s_{n+k+i} = \Phi(nk+i) + w^{nk+i} \left( \frac{g}{1-w} - \frac{\Psi_{1}(k)}{w(1-w^i)} \right) - \Psi_{1}^{nk+i-1} (\leq, \geq) \alpha \]

\[ s_{n+k+k} = \Phi(nk+k) + w^{nk+k} \left( \frac{g}{1-w} - \frac{\Psi_{1}^{k+1}}{w(1-w^k)} \right) - w\Psi_{1}^{nk+k-1} < \alpha \]

Solving for \( g \) we get:

\[ g (\leq, \geq) (1-w)(\alpha + \Psi_{1}^{k} \frac{w^{nk}}{1-w} + \Psi_{1}^{nk}) \]

\[ \cdots \]

\[ g (\leq, \geq) (1-w)(\alpha + \Psi_{1}^{k} \frac{w^{nk+i-1}}{1-w} + \Psi_{1}^{nk+i-1}) \]

\[ \cdots \]

\[ g < (1-w)(\alpha + \Psi_{1}^{k} \frac{w^{nk+k-1}}{1-w} + \Psi_{1}^{nk+k-1}) \]

Let \( kn + \text{min} \) be the equation of the form \( g < z_{kn+j} \) such that \( z_{kn+\text{min}} \) is the smallest among all such equations. Let \( kn + \text{max} \) be the equation of the form \( g \geq z_{kn+j} \) such that \( z_{kn+\text{max}} \) is the largest among all such equations.

Clearly if there is a \( g \) which satisfies these two equations then it also satisfies the entire set of \( k \) equations. Since \( H \) is a self similar cycle, we can always choose \( K \) such that the index \( \text{min} \) comes before the index \( \text{max} \). Let \( \Delta = \text{max} - \text{min} \).

It is left to show that \( z_{kn+\text{min}} > z_{kn+\text{max}} \). Substituting for the \( x \)'s:

\[ (1-w)(\alpha + \Psi_{1}^{k} \frac{w^{nk+\text{min}-1}}{1-w} + \Psi_{1}^{nk+\text{min}-1}) > \]
\( > (1 - w)(\alpha + \psi_1^k \frac{w^{nk+\text{max}-1}}{1 - w} + \psi_1^{nk+\text{max}-1}) \)

Deleting common terms:

\[ \psi_1^k \frac{w^{nk+\text{min}-1}}{1 - w} + \psi_1^{nk+\text{min}-1} > \psi_1^k \frac{w^{nk+\text{max}-1}}{1 - w} + \psi_1^{nk+\text{max}-1} \]

From the definition of \( \Psi \) we get:

\[ \psi_1^{nk+\text{max}-1} = \psi_1^{nk+\text{min}+\Delta-1} = \psi_1^{nk+\text{max}-1} + w^{nk+\text{min}-1} \Psi_1^{\Delta} \]

and by rearranging the terms we get that it is sufficient to prove that:

\[ \psi_1^{nk+\text{min}-1} - \psi_1^{nk+\text{max}-1} >= (w^{\text{max}} - w^{\text{min}}) \psi_1^k \frac{w^{nk-1}}{1 - w} + w^{nk+\text{min}-1} \Psi_1^{\Delta} \]

The strings \( L_1 = (h_{\Delta+1}, ..., h_{nk+\text{max}-1}) \) and \( L_2 = (h_1, ..., h_{nk+\text{min}-1}) \) have the same length. Clearly \( h_{nk+\text{min}} = 0 \) and \( h_{nk+\text{max}} = 1 \) as can be seen from the \( < \) and \( \geq \) signs in their respective equations. If \( n > 1 \) this also implies that they are not identical. Therefore they meet the conditions of lemma 2.4.2 and hence there is an \( x (x \leq k) \) such that:

\[ \psi_1^{nk+\text{min}-1} - \psi_1^{nk+\text{max}-1} >= w(x)(1 - w) \]

and so by selecting a sufficiently large \( n \) we get:

\[ \psi_1^{nk+\text{min}-1} - \psi_1^{nk+\text{max}} >= w(x)(1 - w) >
\]

\[ (w^{\text{max}} - w^{\text{min}}) \psi_1^k \frac{w^{nk-1}}{1 - w} + w^{nk+\text{min}-1} \psi_1^{\Delta} \]

and so our set of equations contains no contradiction.
Figure 1: The Floyd Steinberg Algorithm

a. Posterizing factor $w = 1.0$

b. Posterizing factor $w = 0.9$

c. Posterizing factor $w = 0.75$

d. Posterizing factor $w = 0.5$
Figure 2: The Floyd Steinberg Algorithm

a. Posterizing factor $w = 1.0$

b. Posterizing factor $w = 0.9$

c. Posterizing factor $w = 0.75$

d. Posterizing factor $w = 0.5$
Figure 3: The Floyd Steinberg Algorithm

a. Noise picture - Posterizing factor $w = 1.0$

b. Noise picture - Posterizing factor $w = 0.75$

c. Aliasing picture - Posterizing factor $w = 1.0$

d. Aliasing picture - Posterizing factor $w = 0.75$
Figure 4: Pixels differing from neighbors

(a) 0.2 difference - 1.8% of total picture

(b) 0.1 difference - 9.2% of total picture

(c) 0.05 difference - 25.6% of total picture
Figure 5: 1-D error diffusion using the Geof scan

a. Posterizing factor $w = 1.0$

b. Posterizing factor $w = 0.9$

c. Posterizing factor $w = 0.75$

d. Posterizing factor $w = 0.5$
Figure 6: Using the transformation for the Geof scan

a. Transformation for $w = 0.9$

b. Transformation for $w = 0.75$

c. Transformation for $w = 0.5$
Figure 7: Using the transformation for the F-S algorithm

a. Transformation for $w = 0.9$

b. Transformation for $w = 0.75$

c. Transformation for $w = 0.5$
Figure 8: The 1-D transformation function $t(g)$
Figure 9: Example of a tone scale adjustment graph

Figure 10: The Floyd Steinberg transformation function t(g)
Figure 11: Using monotonic trans. for the F-S algorithm

a. Transformation for $w = 0.75$

b. Transformation for $w = 0.5$