

New wrinkle in dirty paper techniques

Brett Bradley, John Stach
Digimarc Corporation, 19801 SW 72nd Ave, Suite 250,
Tualatin, OR, USA 97062

ABSTRACT

The many recent publications that focus upon watermarking with side information at the embedder emphasize the fact that this side information can be used to improve practical capacity. Many of the proposed algorithms use quantization to carry out the embedding process. Although both powerful and simple, recovering the original quantization levels, and hence the embedded data, can be difficult if the image amplitude is modified. In our paper, we present a method that is similar to the existing class of quantization-based techniques, but is different in the sense that we first apply a projection to the image data that is invariant to a class of amplitude modifications that can be described as order preserving. Watermark reading and embedding is done with respect to the projected data rather than the original. Not surprisingly, by requiring invariance to amplitude modifications we increase our vulnerability to other types of distortions. Uniform quantization of the projected data generally leads to non-uniform quantization of the original data, which in turn can cause greater susceptibility to additive noise. Later in the paper we describe a strategy that results in an effective compromise between invariance to amplitude modification and noise susceptibility.

Keywords: Dirty paper, watermarking, invariance

1. INTRODUCTION

For many watermarking systems, the most significant source of noise is the host media signal itself. This is due to the fact that in general the watermark reader is blind (operates without access to the original unmarked media signal) and the signal power of the host is typically much greater than that of the watermark. Recently, a class of watermarking algorithms has been developed that are capable of rejecting the host signal interference²⁻⁴. These algorithms, in part, owe their inspiration to a paper by Costa, which considers communication under the presence of noise from two disparate noise sources¹. The first of the two noise sources is completely known to the transmitter, and Costa shows that it has no effect on the overall channel capacity. From the point of view of a watermarking system, the first noise source can be thought of as the host media signal. Costa's proof is constructive: he shows how to build suitable codes, yet the result is not practical due to the fact that they require an exhaustive search. Therefore, the proposed systems, mentioned above, offer simplified versions of Costa's scheme for reasons of practicality.

The most popular simplification of Costa's scheme has resulted in extensions and modifications of quantization-based methods¹¹. In this paper we adopt the terminology used in the book by Cox, Miller, and Bloom⁷ who refer to the quantization-based methods as lattice codes. In its simplest form, a lattice code consists of a pair of quantizers. One quantizer represents a '1' bit and the other represents a '0' bit. The quantizer that is used depends upon the bit that is to be embedded in some element of the host media signal. The quantizers are offset by fifty percent (dithered) in order to minimize the required embedding distortion. The most significant enhancements to the basic quantization method include distortion compensation and spread transform modulation, which effectively trade-off embedding rate with embedding distortion⁵. It is well known that this class of techniques is vulnerable to amplitude modifications. Eggers, et al.⁶ has shown that for affine amplitude modifications, this problem can be overcome through the use of a pilot signal in order to estimate both the scaling factor and any additive offset that is applied to the embedded signal. However, turning now to the case of imagery, many amplitude modifications performed in practice are non-linear⁹.

In this paper, we propose a modification to quantization-based techniques to make them invariant to a large class of amplitude modifications; those that are order preserving. In our development we find that the proposed modification has interesting side effects. Specifically, there are inherent problems that make both embedding distortion and robustness to additive noise image dependent. We are hence required to make further modifications in order to make our technique more practical. In section two we fully describe our proposed modification to quantization based schemes, ending with

an illustration of some undesirable effects. In section three we develop a method that relaxes the inherent order preserving quantization bin structures in order to make them resemble more closely those found in uniform lattice codes. The modification results in a loss of knowledge of the exact structure of the lattice code, but this structure is recoverable through the use of training data. In section four we briefly cover several additional considerations that come into play when using order preserving lattice codes. Section five discusses the extension of our technique to lattice codes applied on pseudo-random projections. Our conclusions are presented in section six.

2. ORDER PRESERVING LATTICE CODES

In this section we explore a technique that uses lattice codes on a mapping of image pixels rather than the pixels themselves. The mapping has the property that certain common image modifications will result in the same set of mapped values, which for data-hiding purposes, will lead to a set of invariances. The types of image modifications to which our scheme is invariant can be classified as those that preserve order.

Beginning with a grayscale image, $I(x,y)$, consider its histogram, $h(b)$, where b represents a range of luminance from 0 to L_{max} . We are interested in the empirical cumulative distribution function (CDF) of $I(x,y)$, which can be obtained from the histogram:

$$F_b(B) = \frac{\sum_{l=1}^B h_l}{\sum_{l=1}^{L_{max}} h_l} \tag{1}$$

For some luminance value $B = b$, $F_b(b)$ is the percent of luminance values in the image less than or equal to b . This terminology has been derived from language used when referring to real cumulative distribution functions¹⁰, where $F_x(x)$ is the probability that the random variable X is less than or equal to x . For our purposes, $F_b(b)$ is the mapping we apply to the image pixels prior to data hiding by quantization. In the following figure we have plotted the CDF of the image "sailboats" using the solid curve.

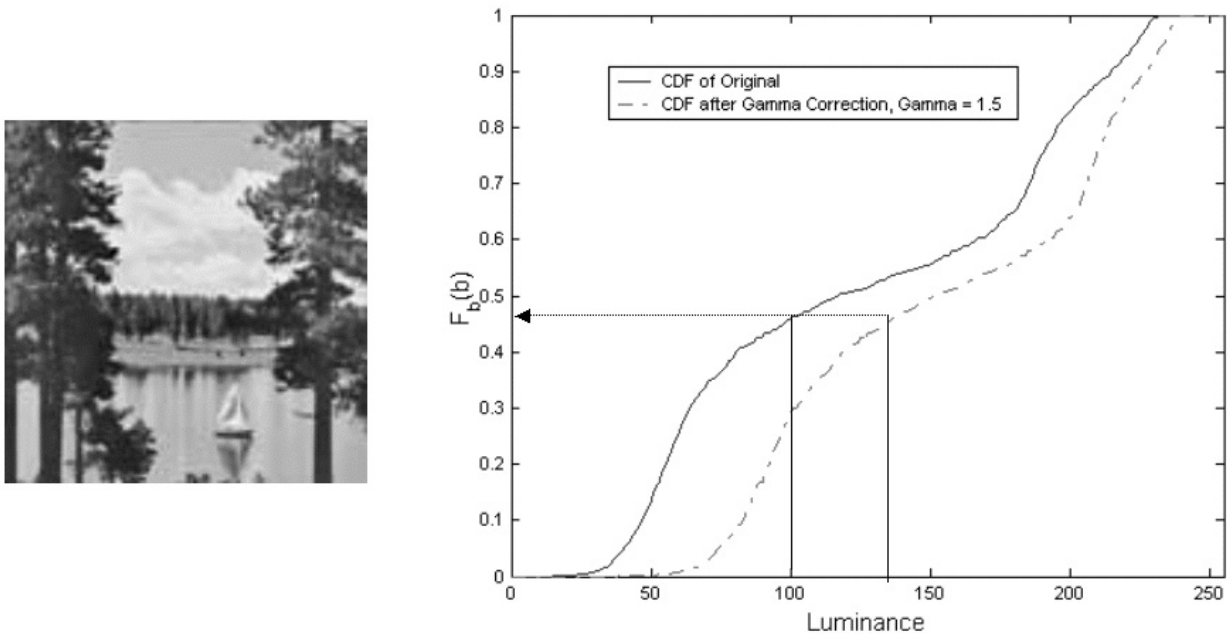


Figure 1. First subfigure is "sailboats." Second subfigure is CDF of the image before (solid curve) and after (dotted curve) gamma correction of 1.5. The plot illustrates the fact that x and $g(x)$ –the point x after gamma correction -- have the same CDF value.

A luminance value of 100 corresponds to an $F(b)$ value of approximately .46, as shown by projection onto the y-axis. The dotted curve shows the CDF of the image after a gamma correction of gamma equal to 1.5 has been applied. After the modification, luminance values of 100 become luminance values of approximately 137. It is observed by projection that the new luminance values have the same $F(b)$ values as the old luminance values. In other words, the projection is invariant to gamma correction. More generally, the projection is invariant if the following condition is met:

$$F_y(g(x)) = F_x(x) \quad (2)$$

if

$$\frac{d}{dx} g(x) > 0 : \forall x$$

Equation 2 describes the fact that the mapping $y = g(x)$ must be a monotonically increasing function, which means that in its application the ordering of elements remains the same. In addition to gamma correction, several other common image modifications satisfy this condition for example alterations involving brightness or contrast.

In our proposed data-hiding scheme, we quantize each mapped value, $F_b\{b(x,y)\}$, using a quantizer that is appropriate for the message that we wish to embed at location x,y . In a binary messaging scheme we use two uniform quantizers that are offset from each other, or dithered. The result is a scalar lattice code applied to the mapped data, $F_b\{b(x,y)\}$, rather than the luminance values, $b(x,y)$. Another way to view this process is to consider what happens to the luminance values when we quantize with respect to the mapped data. There is an equivalent set of quantization levels in the unmapped luminance domain given by:

$$b_q = F_b^{-1}(P_q) \quad (3)$$

In the above equation, the P_q represents the quantization levels of the projection $F_b(b)$. The b_q are not uniform unless the image content has a flat histogram (linear CDF). In general, the bins b_q will be close together where the CDF has steeper slope and further apart in regions of gradual slope. We show an illustration of this in the following figure:

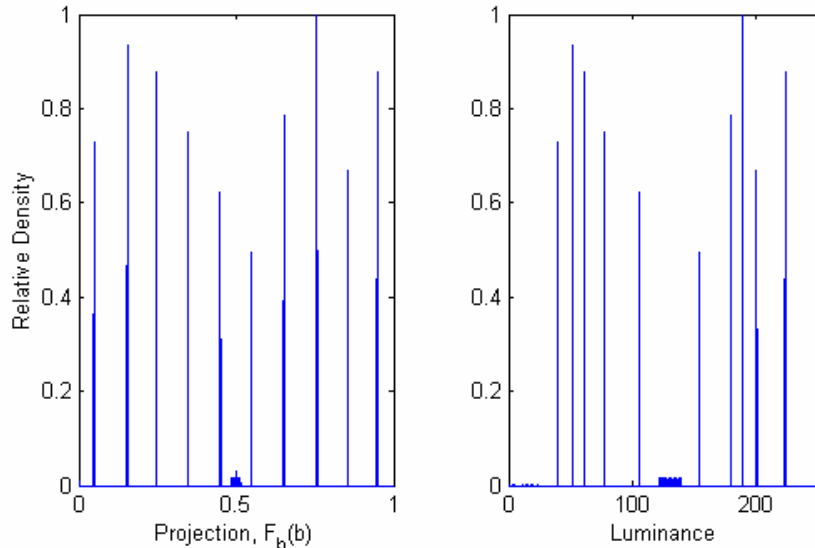


Figure 2. Right subplot shows histogram of image after embedding. Left subplot is histogram after mapping $F(x)$ has been applied.

The subplot on the left shows the distribution of projection values after quantization when the quantizer for embedding a bit value of '1' is used for all pixels. The quantization levels are uniform; they begin at $F_b(b) = 0.05$ and continue with a step-size of 0.1. They are set coarsely for demonstration purposes. We use the same "sailboats" image in this example as we used in Fig. 1. The dependency upon the CDF of the corresponding quantization levels in the luminance domain is depicted in the subplot on the right. The quantization levels are narrowly spaced in regions where the CDF increases most rapidly (luminance of around 50 and 200). Conversely, the quantization levels are more spread out in CDF regions of small slope.

2.1 Inherent Drawbacks

There are several drawbacks with the proposed approach if it is implemented without further modifications. Once again referring to figure 2, we see that the required embedding distortion for regions where the quantization bins are widely spaced can be quite large. There are some visible artifacts in the figure at luminance values close to zero and around one hundred twenty, in that they do not belong to a peak. For our example implementation, we set a maximum on the allowed magnitude change of luminance. If the change required for quantization is too large, we simply do not alter the pixel in question. Hence, the previously mentioned artifacts are the result of not performing the quantization operation in cases where the resulting embedding distortion would be impermissible. Of course, one can reduce the effect of this particular problem by decreasing the bin step-size in the mapped domain. However, this exacerbates the effect of another problem. In areas where the image CDF is increasing rapidly, the quantization bins in the luminance domain will be close together, which leads to poor noise immunity. In other words, it will take very little distortion to cause an error in these regions. A related problem has to do with the overall span of luminance values in an image. For a fixed number of levels in the projection domain, there will be poor noise immunity if the overall luminance span is small. Conversely, there will be data embedding problems if the luminance span is too large. In the next section we describe a modification to the proposed algorithm that mitigates these problems.

3. ADAPTIVE CDF LATTICE CODES

As discussed above, the spacing of quantization bins in the luminance domain when uniform in the mapped domain will sometimes lead to unsatisfactory behavior. Ideally, one would alter the location of these bins by considering the image CDF. In the extreme, one would let the CDF dictate bin assignments in such a way that the result would be uniform bin spacing in the luminance domain, a typical lattice code structure. In doing so, we would remain invariant to luminance shifts due to the fact that the bin starting locations would be governed by the projection $F_b(b)$. Of course, we would lose invariance to other types of amplitude modifications. We seek a means to go between the two extremes.

Our solution to the aforementioned problem adapts bin spacing from a uniform projection domain starting point in a step-wise fashion. We do this so that the bin spacings are amenable to easy recovery should the image be subjected to a variety of amplitude modifications. Initial quantization levels in the projection domain are chosen at a finely granulated level. The spacing is chosen so that required luminance changes will not be too large in CDF regions of small slope.

The next step is to apply a process of quantization level coarsening. The goal of this process is to remove some quantization levels in regions where they are too close together. Ideally, we should achieve an overall bin configuration that is much more uniform. We treat the set of two quantizers, representing different message bits, together so that the dithering configuration remains intact*. Beginning with the lowest two quantization bins in each quantizer, we consider the additional embedding distortion that would arise if we removed half of the levels. We refer to this as pruning. We will describe shortly the mechanism used for pruning. If the additional embedding distortion is deemed satisfactory, we perform the pruning. We proceed in an analogous fashion throughout the entire range of quantization bins. An illustration of the described process is shown in the following figure.

*By dithering we mean the consistent alteration of symbols 'circle' and 'square' representing symbols '0' and '1', respectively

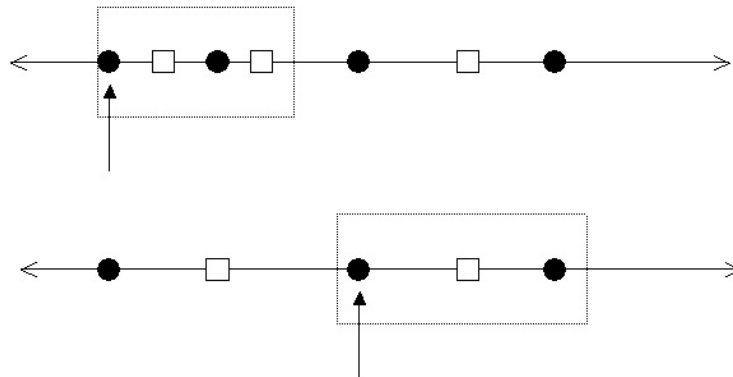


Figure 3. Quantization level coarsening procedure: shown in luminance domain where the quantization centroids are generally non-uniform.

The quantizer is represented in the luminance domain where the levels are in general non-uniform. The vertical arrow in the top subfigure points to the first member of the group under consideration for pruning -- encapsulated by the dotted lines. In order to prune the group, we would like to replace the set of four quantization levels in the group with a single pair of quantization levels. This is done by leaving the first quantization level unchanged, removing the 2nd and 4th levels, and changing the identity of the third from a circle to a square. Next, we determine the additional embedding distortion that is required to adopt this new quantizer configuration. If, for example, the maximum embedding distortion remains below a threshold after the modification, we keep the new configuration. Otherwise, we do not make the change. In the second subfigure we have the result of the described alteration when it is deemed distortion satisfactory. The dithering pattern remains intact and the change affects only local positions within the quantizer. As mentioned previously, we then proceed to the next group of four levels, which is indicated by the vertical arrow pointing to the first member of the new group in the second subfigure. Whether pruning takes place or not, the next group that we consider begins with the symbol two symbols to the right of the first symbol in the previous group. Therefore, if for the case of the example just described we had not carried out the pruning process, the next group would begin with the second circle from left in the first subfigure of figure 3.

After pruning, the quantization levels in the luminance domain should be of both a more appropriate spacing throughout and appear more uniform. We can, of course, perform subsequent iterations of pruning as necessary. We have provided an example of the result of two iterations of the described process in the figure below. We have applied our algorithm to the "sailboats" image. Beginning with uniform quantization levels in the projection domain, the quantization levels for one of the quantizers in the luminance domain are shown in the leftmost subplot in the figure. The initial quantization levels are disparate, to say the least. Referring to the second subplot, which is the result after one iteration of coarsening, we see a substantial improvement. After two, the quantization levels are much closer to uniform (furthest right subplot).

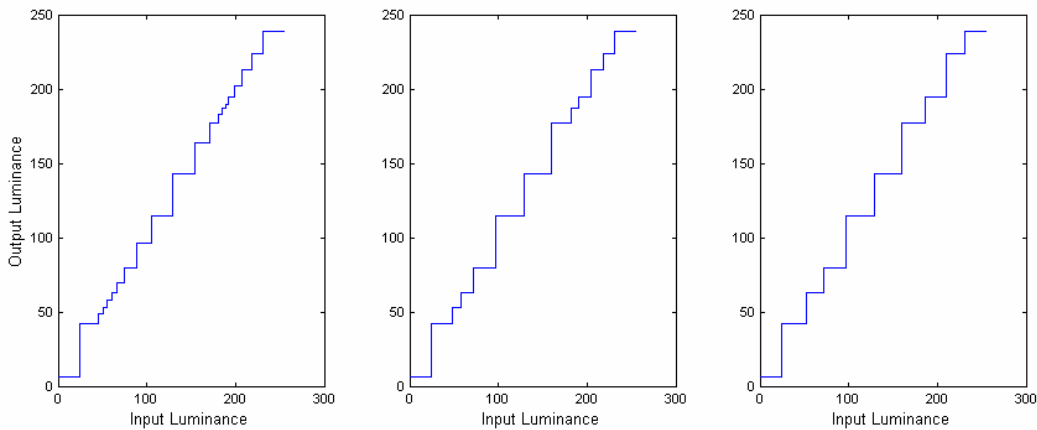


Figure 4. Effect of Quantization level coarsening after two iterations

3.1 Quantization Level Recovery

The benefit of uniform quantization in the projection domain is invariance to amplitude modifications that preserve luminance ordering. By pruning quantization levels using the process described above, we lose a priori knowledge of what the exact quantization levels are if the image amplitude is modified. However, these original levels are not difficult to recover for two reasons. Qualitatively speaking, many amplitude modifications of the type we are concerned with will not drastically modify the shape of the image CDF[†]. We expect that neighboring regions in the luminance domain will be affected in more or less the same way. Therefore, since it operates locally, the process of pruning initial quantization levels in the modified image should result in close to the same set of final levels that were arrived upon with respect to the initial image. The other reason we can recover the initial levels is a result of the coarsening design itself.

Recovering the initial quantization levels can be construed as a hypothesis-testing problem. Suppose we are presented with a watermarked image that may have been altered by a process that preserves pixel ordering, such as contrast adjustment. The first step in watermark reading, as well as embedding, is to create the image CDF. From this point, we establish the initial set of relatively finely granulated levels that are uniform in the mapped domain. It should be stressed that the reader and the embedder use the same initial spacing of levels in this domain. Here, the reader is asked to behave as if it were the embedder. The reader conjectures as to whether the embedder pruned quantization bins in the original image prior to embedding, keeping in mind that the quantization bins in the luminance domain may have moved somewhat relative to their embedding positions. Again, the procedure is carried out beginning with the lowest quantization levels in groups of four. Assume for a moment that we allow only one stage of coarsening. For each group, there are two possible scenarios. Either, we left the four levels alone, or we merged them. In many instances, one of the two scenarios will be remote and we need not consider it further. However, in many other cases it will be difficult to establish which case is valid without the use of training data.

[†] An example of an exception to this general statement is histogram equalization.

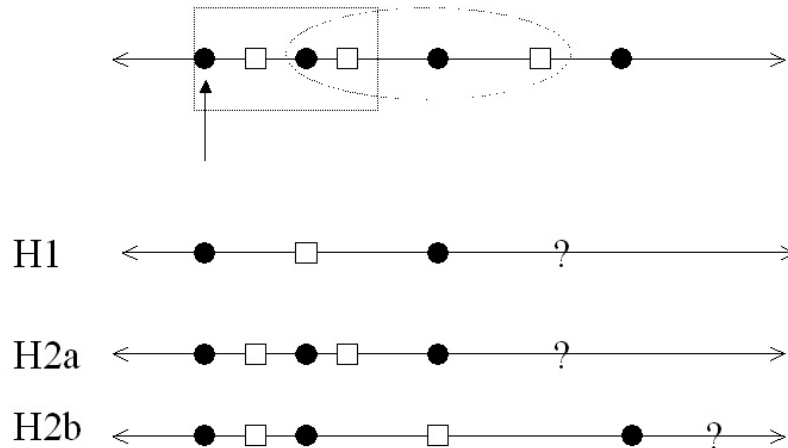


Figure 5. Quantization level recovery process

We illustrate the quantization level recovery process in figure 5. The top subfigure is the same one that was used in figure 4 with one modification. We have drawn a dotted ellipse around an additional group to show how it contributes to the decision process. The next subfigure, labeled H1, is the configuration after coarsening occurs. The question mark to the right of the depicted quantization levels is present to indicate that any additional quantization levels in the quantizer have no bearing on the decision at hand. The last two subfigures represent the case where we do not perform coarsening on the first group. The two competing hypotheses here consist of whether coarsening on the next group occurs, which is circumscribed by the ellipse in the top subfigure. The quantizer H1 differs from both H2a and H2b, which both have the bin pattern *circle, square, circle* over the same range that H1 has the pattern *circle, square*. Training data could be used to make a decision between whether H1 is in force versus H2 in considering only the mentioned set of levels. Training data is simply a pre-specified subset of the total data to be embedded where each symbol is known. As such, it would be embedded using the same set of quantizers as the regular information-carrying data. There are at least several ways to use the training data in order to recover the proper quantization bin configuration. One such method is comprised of decoding any training data that happens to belong to the region spanned by the first group of bins using each of the hypothesized bin configurations. In doing so, we can determine the most likely quantizer configuration by measuring the resulting error rate. Moreover, by considering training data from additional levels (up to the second circle in H1) we could further bolster our decision. Supposing for a moment that H1 is representative of what actually occurred at the embedder, it is apparent that H2a is much closer to the true configuration than H2b. It is therefore much more likely that H2a will be chosen in the event of an error than H2b. As with embedding, the described process continues over the range of the quantizer.

There are a couple of considerations that will slightly modify the process just described. One of these is the fact that we will often like to perform an additional iteration of coarsening to our quantization levels. Doing so will grow the number of hypotheses we must consider in each region of the quantizer. Another issue lies in the fact that we can use a priori information in the coarsening process in addition to training data. Earlier we stated that we need not consider hypotheses that are too remote. Rather than making a binary decision in this regard we can incorporate our estimate of the prior probability for each hypothesis into the overall decision resulting in an a posteriori determination.

4. OTHER CONSIDERATIONS

4.1 Luminance Span Accommodations

Some images are inherently unsuited to data embedding using the described algorithm. As a severe example, take an image that spans just a few gray levels of total contrast. The CDF of this image changes very quickly from zero to one, yet our procedure begins with a prescribed set of projection domain quantization levels. The coarsening method described above will partially serve to alleviate this problem, but it will take many iterations of the procedure in order to whittle away the number of quantization levels to the scant few that appropriately match the image content.

A further enhancement to our basic algorithm involves assessing the total contrast range prior to assigning initial quantization levels in the projection domain. An easy way to define this concept is in terms of the image CDF. We define the total contrast as the difference in luminance between the corresponding points in luminance where the CDF is equal to x_1 and where the CDF is equal to $1-x_2$, where x_1 and x_2 are small and typically $x_1 = x_2$. The foregoing description applied to the image "sailboats" is illustrated in figure 6. From our measure of total contrast, we assign an initial spacing of quantization levels in the projection domain. In order to make the problem tractable, we would prefer that the same set of initial quantization levels are chosen during watermark reading as was used for watermark embedding even if a fairly significant amplitude modification has taken place. Suppose, for example, we consider only two cases. We will begin with many quantization levels for embedding when an image has large total contrast, and we will use relatively few quantization levels when the image has small total contrast. If the wrong set of initial quantization levels is chosen when reading the watermark, we can use the training data to rectify the problem.

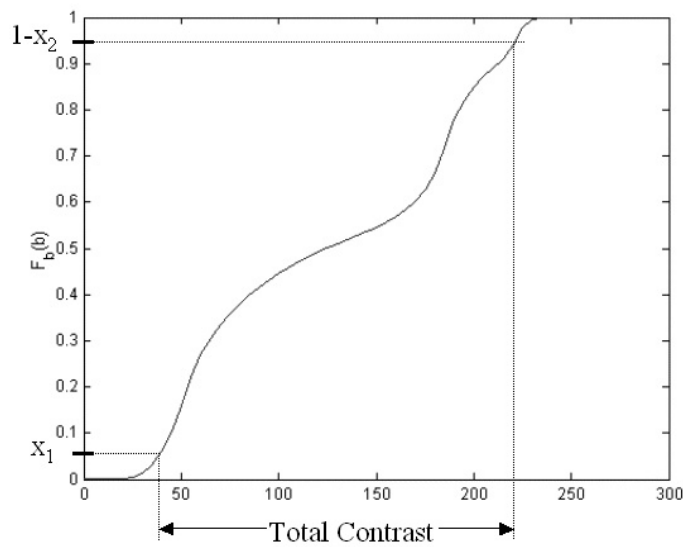


Figure 6. Illustration of total contrast definition

4.2 Distortion Compensation

Distortion compensation has interesting implications when applied to our basic method. Recall from the work of Chen and Wornell⁵ that when applied to any given element, distortion compensation results in a weighted combination of that element before and after quantization. The weighting parameter, which we refer to as alpha, has a range between zero and one, where a zero value means that no image alteration is performed and a one means that full quantization is used. In a uniform quantizer decreasing alpha increases bin sizes, which reduces the overall probability of error in more severe additive noise environments. A full treatment of this topic is beyond the scope of this paper, but we list several references^{5, 7, 8}. Due to the fact that we have, in general, non-uniform quantizers, we can treat distortion compensation in a different way. Put simply, we choose to make the amount of distortion compensation used commensurate with the size of each bin. Where the bins are close together we use no distortion compensation, set alpha equal to one (we could use a default maximum value for alpha instead of one). Where the bins are spread further apart, we decrease alpha accordingly. By adding this additional feature, we mitigate the problem illustrated in figure 2. Specifically, the artifacts, as we have called them, as a result of not embedding due to the large bin spacing will be reduced.

4.3 Contouring Reduction

In their implementation of the Scalar Costa Scheme, Eggers, et.al use a key in order to securely embed the watermark⁸. Visually, their addition has the effect of removing the contouring that would otherwise be a result of the quantization process. We refer to contouring as the visual effect that is achieved when a relatively small number of the total possible gray levels are populated by pixels. In brief, the key is used to partition the image pixels into possibly many different groups. The pixels within each group use a quantizer that is offset from the original quantizer by some fraction of the bin size. Each group uses a different fractional offset. As a set they are typically equal-spaced between 0 and 1. Provided enough distinct groups are used there should be no observed quantization upon embedding, especially when distortion compensation is used.

In our proposed method, we can employ the key to similar effect. However, there is some ambiguity in terms of how we proceed because our quantizers are in general non-uniform. We define quantizer offsets in the projection domain and maintain the assumption that, for each group, the fractional offset is between zero and one. To adjust a fractional offset from the original quantizer configuration, we move each bin center the desired fractional amount to the higher of the two adjacent bin centers. One potential issue with this approach is that the coarsening procedure applied to generate the original quantizer configuration may not match the quantizer for a group with a relatively large fractional offset. For this reason it may be better to define an offset range centered about the original quantizer with fractional shifts in either direction.

4.4 Training Data

There are potentially two places where training data is required. We have already mentioned that training data is used to determine how quantization levels are configured prior to reading the watermark. Another possible area of use involves a reference for the original CDF, itself. Through the process of watermark embedding it is possible that the original CDF will be altered. If, for example, quantization without distortion compensation and a security key is used during embedding, the CDF will change to resemble a staircase. When the underlying CDF is altered, we must reserve a subset of the image pixels to use as unembedded reference values. It is the CDF of the reference values rather than that of the total image that is used to govern quantization bin assignments. In implementations that use fractional offset, key-controlled quantizers and/or distortion compensation, the reference set is potentially unnecessary because the original CDF is altered insignificantly during the embedding process.

5. ORDER PRESERVING SPREAD TRANSFORM LATTICE CODES

To round out our discussion of lattice codes applied to image data that has undergone the mapping $F(b)$, we discuss the implications of extending our ideas to the case where quantization occurs after the data has been projected onto a pseudo-random vector⁵. A general description of the process is as follows. We partition the image into vectors of length L , comprised of pixels from pseudo-random locations throughout the image. Focusing on just one of the resulting vectors, we apply the mapping function, $F(b_i)$, to each element of the vector. To establish the required watermark, we project this new mapped vector onto a pseudo random sequence. The result is quantized using one of two uniform quantizers, depending upon which message we want to embed. The watermark is embedded such that the spread transform of the mapped data achieves the quantization target. Mathematically, this can be expressed as follows:

$$\sum_i F(b_i + w_i) s_i = P_Q \quad (4)$$

In this equation, s_i is the pseudo-random spreading vector and P_Q is the target projection. The equation expresses the fact that the watermark, w_i , is added to the original luminance data. Another way to view this expression more explicitly shows how the watermark is generated to achieve the desired result:

$$\sum_i [F(b_i) + d_i] s_i = P_Q \quad (5)$$

In the above equation, the projection of each component of the original data is modified additively, which matches the procedure applied when using regular spread transform lattice codes. The relation between the two expressions is:

$$w_i = F^{-1}(F(b_i) + d_i) - b_i \quad (6)$$

Depending upon the image histogram, this may lead to too much embedding distortion for some of the b_i .

A graphical illustration of some of the key issues associated with the described process is shown below.

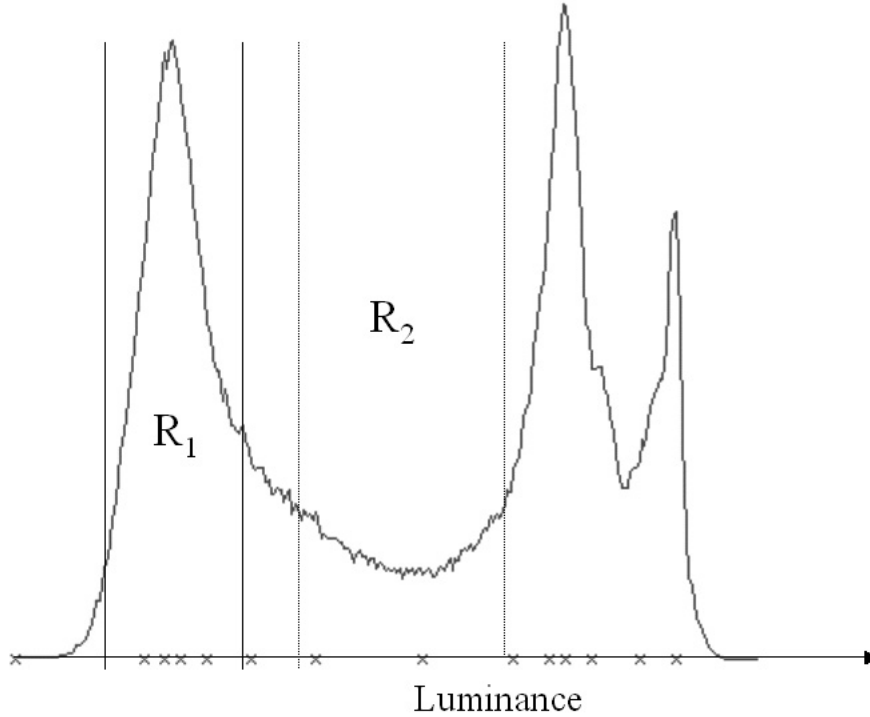


Figure 7. Histogram of image "sailboats" The area under the curve is constant between any consecutive pair of x 's.

This figure is the histogram of the image "sailboats" before embedding. There are a series of ' x ' marks on the axis labeled 'luminance'. These ' x ' marks are uniformly spaced in the projection $F(b)$, but they are clearly not uniformly spaced with respect to the image luminance data. For the purpose of embedding using the spread transform process described above, we pseudo-randomly draw L pixels that are distributed according to the histogram. Some such pixels will be drawn from the region labeled, R_2 . It is observed that in this region we must make relatively large changes in luminance in order to alter the corresponding projection value. Referring again to equation 5, we see that there are many ways to choose each of the d_i . Typically, the parameter is made constant for simplicity, but in this situation we may want to make d_i small for image pixels drawn from R_2 . Of course, we will have to compensate by making d_i larger in regions like R_1 , where the projection values change much more quickly.

As with the scalar implementation, the quantized projection version of our scheme is more sensitive to additive noise in some regions of luminance than others. In region R_1 , the $F(b)$ value will change quickly with small amounts of additive noise. The same amount of noise is seen to have little effect in region R_2 . By equalizing the various regions with respect to additive noise, we expect to achieve better results. Equalization is accomplished through weighted projections.

$$\sum_i a_i [F(b_i) + d_i] s_i = P_{Q,W} \quad (7)$$

We have modified equation 5 so that each component is weighted by a_i , which depends upon the region that contains each b_i . For example, we propose to use smaller weights for a region like R_1 and larger weights for those with characteristics like R_2 . If we have chosen the weights properly, the weighted projection, $P_{Q,W}$, will change fairly evenly for equal luminance changes across different regions of the image. Also, observe that for a proper choice of weights, it becomes less of a problem to make d_i in equation 7 constant for all i . It is worth pointing out at this time that the embedder and reader should use the same set of weights. Otherwise, the effective noise produced by using the wrong set of weights will undermine their use altogether.

If the weights are to be derived from the data, itself, we must find a way to consistently obtain the same weights that were used during embedding when detecting the watermark even if the image is altered in various ways. In the top-left subplot of figure 8, we show the histogram of an image within the luminance range of fifty to two hundred. The vertical dotted lines partition the histogram into regions of constant thickness in the projection domain. Based upon our arguments above, we would like to assign a weight to each region that varies inversely with the average density. However, there is some difficulty in accomplishing this task due to the fact that the image may undergo an amplitude modification that changes the character of the histogram. The effect of just such a transformation can be seen in the lower left subplot of the same figure. We have applied a gamma correction of 1.5 to the original image. The two nearly equal histogram peaks in the former subplot are now of markedly different heights. By designing weights that are a smoothly varying function of the image histogram, we are bound to get different results when trying to recover the weights if the amplitude is modified.

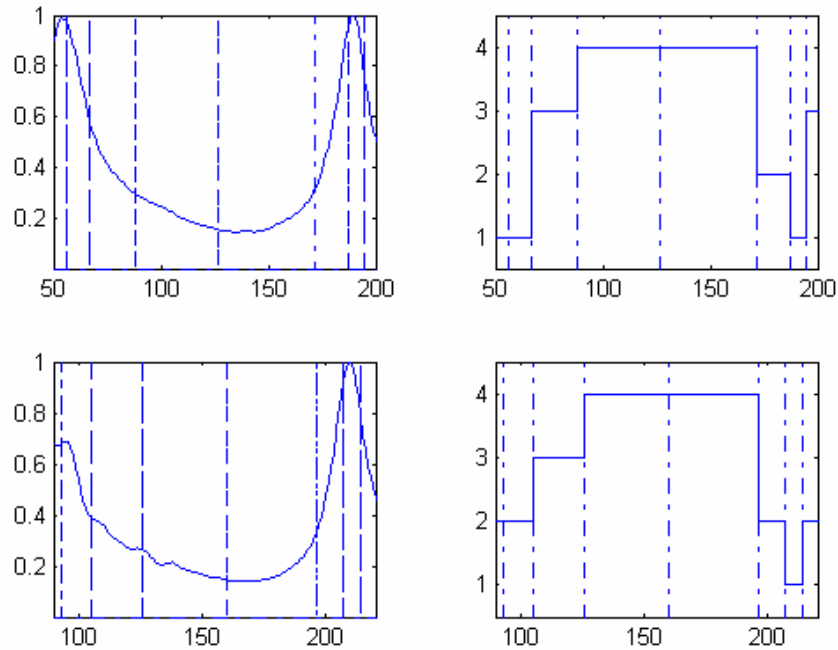


Figure 8. Illustration of process for determining projection weights

The process of determining weights for projections begins by partitioning the projection axis, $F(b)$, into N fixed regions. The regions are of constant thickness in the projection domain. The partitioning is done with respect to the projection because we require that the region boundaries remain constant if an order preserving transformation, $y = g(b)$, is applied to the image luminance. An example of the partitioning applied to the histograms described in the previous paragraph is shown in figure 8. To further expedite our process of weight determination we quantize with respect to the y-axis in addition to the x-axis. The average density in each of the N regions is computed. The region that has the maximum average density is assigned a weight of 1. This weight is the minimum for all regions as it is expected to have the worst noise immunity. The other regions are assigned weights relative to the first weight assignment. An example of this process is depicted in the top-right subplot of figure 8. In each region we have quantized the weight to one of four levels, which is observed to vary inversely with the average density. After gamma correction with gamma equal to 1.5, the same process is applied. The weights match the set used for embedding in all but two out of the seven regions shown. However, we can do better.

By using training data we can refine the weight estimates obtained using the process described above. Training data are defined with respect to the quantized projection subspace; one element of the training dataset consists of L components

of luminance data. Training data is embedded in exactly the same fashion as the information carrying data, according to equation 7. On the watermark detection side, once an initial estimate of the weights is obtained for each region, we apply the corresponding projections to the training dataset. We call the set of projections our reference set. An error rate can be assigned to the reference set since we know what the training dataset is. Next, we vary the weights in each region both one step up, and one step down. For each weight variation we reapply the projections and compare with the reference set. If fewer errors are obtained, we replace the old weight with the new, and call the new set of projections the reference set. In this way, we proceed through all regions and obtain the correct set of weights.

6. CONCLUSIONS

We have presented a novel modification to watermarks based upon lattice codes that renders them invariant to many types of amplitude modifications. Quantization bins in our scheme are defined with respect to the image CDF, which maps order preserving image transformations to the same location. However, we have shown that in its pure form the modification leads to problems with both embedding distortion and noise immunity that are image dependent. We have disclosed several ways through which invariance can be retained and distortion problems can be reduced involving the use of training data. Finally, we have extended our ideas to the case where lattice codes are applied to data that has first undergone a spread transform.

REFERENCES

1. M.M.H. Costa, "*Writing on dirty paper*," IEEE Transactions on Information Theory 29, pp. 439-441, May 1983.
2. B. Chen and G.W. Wornell, "*Provably robust digital watermarking*," in Proceedings of SPIE: Multimedia Systems and Applications II (part of Photonics East '99) vol. 3845, pp. 43-54, (Boston MA, USA), September 1999.
3. B. Chen and G.W. Wornell. "*An information-theoretic approach to the design of robust digital watermarking systems*." IEEE Transactions on Acoustics, Speech, and Signal Processing, 1999.
4. I.J. Cox, M.L. Miller, and A.L. McKellips, "*Watermarking as communications with side information*," Proceedings of the IEEE, Special Issue on Identification and Protection of Multimedia Information 87, pp.1127-1141, July 1999.
5. B. Chen and G.W. Wornell. "*Preprocessed and postprocessed quantization index modulation methods for digital watermarking*." In Security and Watermarking of Multimedia Contents II, volume SPIE-3971, pp. 48-59, 2000.
6. J.J. Eggers, R. Bauml, and B. Girod. "*Estimation of amplitude modifications before scs watermark detection*." In Security and Watermarking of Multimedia Contents IV, volume SPIE-4675, pp. 387-398.
7. I.J. Cox, M.L. Miller, and J.A. Bloom, *Digital Watermarking*, Chapter 5, Academic Press, 2002
8. J. Eggers, B. Girod, "*Informed Watermarking*", Chapter 3, Kluwer Academic Publishers, 2002.
9. W.K. Pratt, *Digital Image Processing*, Chapter 10, John Wiley and Sons, 1991.
10. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 1991.
11. E. H. Adelson, U.S. Patent No. 4,939,515, September, 1988.