Title: **Improved Halftone Image Data Hiding with Intensity Selection and Connection Selection**

Running head: *Data hiding with intensity and connection selection*

Authors: **Ming Sun Fu*, Oscar C. Au** (correspondence author = Oscar C. Au)**

Affiliation: Dept. of Electrical and Electronic Engineering

(address) Hong Kong University of Science and Technology

Clear Water Bay, Hong Kong, China.

Tel: +852 2358-7053, Fax: +852 2358-1485

Email: *fmsun@ust.hk, **eeau@ust.hk

Abstract: In this paper, we propose two novel algorithms, namely intensity selection (IS) and connection selection (CS), that can be applied to the existing halftone image data hiding algorithms DHSPT, DHPT and DHST to achieve improved visual quality. The proposed algorithms generalize the hidden data representation and select the best location out of a set of candidate locations for the application of DHSPT, DHPT or DHST. The two algorithms provide trade-off between visual quality and computational complexity. The IS yields higher visual quality but requires either the original multi-tone image or the inverse-halftoned image which implies high computation requirement. The CS has lower visual quality than IS but requires neither the original nor the inverse-halftoned images. Some objective visual quality measures are defined. Our experiments suggest that significant improvement in visual quality can be achieved, especially when the number of candidate locations is large.

List of Unusual Symbols

N	number of data bits to be embedded
M	number of pixels of which the pixel sum will be used to store one data bit
IS	intensity selection
CS	connection selection
$x(i, j)$	original multi-tone image pixel value at location (i, j)
$y(i, j)$	halftone image pixel value at location (i, j)
$con(i, j)$	measure of connectedness of pixel at location (i, j)
$\boldsymbol{A}$	locations among NM locations chosen by IS or CS, at which toggling is applied
N_i, i=0,..., 4	total number of class 1 and class 4 elements in $\boldsymbol{A}$ having i neighbors with same pixel values in the 4-neighborhood.
S_i, i=1,..., 5	five objective visual quality measures
S_1	total number of class 1 and 4 elements of $\boldsymbol{A}$
S_2	total area covered by clusters associated with class 1 and 4 elements of $\boldsymbol{A}$
S_3	average area per cluster
S_4	number of class 1 and 4 elements of A associated with clusters with size 3 or more

Number of pages: 28 (page 5 to 32)

Number of tables: 13

Number of figures: 30

Keywords: data hiding, watermark, halftone image, intensity, connection, selection, toggling, eccentricity, inverse halftoning, error diffusion, dithering

List of Tables

List of figures

Improved Halftone Image Data Hiding with Intensity Selection and Connection Selection

Ming Sun Fu, Oscar C. Au

Department of Electrical and Electronic Engineering,

Hong Kong University of Science and Technology,

Clear Water Bay, Hong Kong, China.

Tel: +852 2358-7053, Fax: +852 2358-1485, Email: fmsun@ust.hk, eeau@ust.hk

1. Introduction

Nowadays, images appear routinely in massively distributed printed matters such as books, magazines, newspapers, printer outputs and fax documents. Images appear also in widely accessible web pages, and multimedia files on the Internet and in storage media such as CD-ROM and DVD. Associated with the widespread circulation of images are issues of copyright infringement, authentication and privacy. One of the possible solutions is to embed some hidden watermarking data into the images.

Digital watermarking [22] is a process that can embed some invisible digital data called watermark into an image. There are two classes of invisible watermarks: fragile watermarks and robust watermarks. Fragile watermarks are designed to be broken easily by common image processing operations. The broken watermark serves as an indication of alteration of the original multimedia data and is useful for authentication. Major applications include tampering detection of images placed on the World Wide Web and authentication of images received from questionable sources. Robust watermarks are required to remain in the watermarked image even after it has been attacked by attackers or processed by common image processing operations such as filtering, requantization, scaling, cropping, etc. Invisible watermarks can also be classified as private and public watermarks. A private watermark uses the original image for watermark decoding while a public watermark does not.

One way to add an invisible digital watermark to an image is to insert into the least insignificant bits (LSB) of the uncompressed image [24]. Some inserts to LSB only around image contours [20]. Some hides small geometric patterns called tags in regions where the tags would be

least visible [6], such as the very bright, very dark or texture regions. Some embeds watermark in the histogram [9]. Some chooses random pairs of image points and increase the brightness of one and decrease that of the other [4]. Some adds a small positive number to random locations as specified by the binary watermark pattern and use statistical hypothesis testing to detect the presence of watermark [23]. Some uses dynamic systems (toral automorphism) to generate chaotic orbits which are dense in the spatial domain and hide the watermark at the seemingly chaotic locations [32]. Some uses predefined patterns to guide level selection in a predictive quantizer, effectively creating a watermark that resembles quantization [21]. Some embeds watermark patterns in the quantization module after discrete cosine transform (DCT) [28] or in selected blocks based on human visual models [27]. Some hides watermark in JPEG images by forcing selected DCT blocks to satisfy certain linear or circular constraint [5]. Some uses the DCT domain perceptual model to hide the maximum strength watermark for maximum robustness [19]. Some embeds in the discrete Fourier domain [26], wavelet domain [33], or other transform domains [8, 25]. Some uses spread spectrum technique in the frequency domain [10].

Many methods are invertible making it susceptible to Single-Watermarked Image Counterfeit Original (SWICO) attack [11]. It can be modified to become non-invertible to overcome SWICO, but still subject to Twin-Watermarked-Images Counterfeit-Original (TWICO) attack [12]. It can be further modified to become non-quasi-invertible to overcome TWICO. Some use hybrid methods to introduce watermarks into video [27]. They use dual watermarks, one public and one private. Two keys are used to generate the private watermark. One key is chosen by the user and thus is user dependent while the other key is signal dependent, computed from the input using some one-way hash functions. To ensure visual imperceptibility, they use frequency masking and spatial masking according to human visual system masking function. Most if not all existing image watermarking schemes use the inherent redundancy of gray scale images to embed data such as using the least significant bit to embed data in the spatial domain or modifying the images within the perceptual thresholds in the frequency domain. All these methods are designed for multi-tone images or video.

In this paper, we are concerned about invisible watermarking for a special kind of image: halftone images. Halftoning [31] is a process to change multi-tone images into 2 tone images, which

look like the original multi-tone images when viewed from a distance. Halftone images are widely used in the printing of books, magazines, newspapers and in computer printers, which are very common in our daily lives. It is often desirable to hide certain invisible meta-data within the halftone images such as company identity, owner information, creation date and time and other information for copyright protection and authentication purposes. As most halftone images are printed on papers, there are very few, if any, possible digital attacks on the watermarked halftone images though there can be many physical attacks on the papers on which the images are printed. As a result, robustness or fragileness of halftone image watermarking schemes is not particularly important for many applications. Instead, the data hiding capacity and image quality are much more important in most applications. In this paper, we assume that there are no physical attacks and the watermarked halftone images can be fully recovered without any error by scanning the printed images.

There are two main kinds of halftoning techniques, namely, ordered dithering [3] and error diffusion [14]. Ordered Dithering is a computationally simple and effective halftoning method, usually adopted in low-end printers. It compares the pixel intensities with some pseudo random threshold patterns or screens in order to determinate its 2-tone output. Table 12 shows an example of a dithering screen [7], which will be used throughout this paper. Fig. 1 shows an original 8 bit gray scale image, 'Lena', of size 512x512. Fig. 17 is the halftoned version of Lena using ordered dithering. Error diffusion is an advanced technique usually used in high-end printers. It is more complicated than ordered dithering, but it can generate halftone images with higher visual quality. It is a single pass algorithm. In error diffusion, the output is obtained by comparing the pixels with a fixed threshold. However, the halftoning error is fed back to its adjacent neighbors so that each pixel has effectively an adaptive threshold. The error feedback helps to maintain approximately equal local intensity average between the original multi-tone images and the corresponding halftone images. A commonly used error feedback kernel is the Steinberg kernel shown in Table 13. The error diffused 'Lena' with the Steinberg kernel is shown in Fig. 2. The Steinberg kernel has a small support and gives halftone images fine texture and good contrast.

It is well known that most image processing techniques such as filtering or resizing cannot be applied to a halftone image to produce another halftone image with good visual quality. Watermarking

is not an exception. Most, if not all, existing watermarking schemes for natural images as mentioned earlier cannot be applied to halftone images due to their special characteristics. The halftone image pixels take on only two values, typically 0 (black) and 255 (white), resulting in lots of high frequency noise and little intensity redundancy. Most existing watermarking schemes for natural images would generate multi-tone, instead of halftone, images.

However, there are still some existing techniques for halftone image watermarking. Some used two different dithering matrices for the halftone generation [2] such that the different statistical properties due to the two matrices can be detected. Some used stochastic screen patterns [18] and conjugate halftone screens [34] in which two screens were used to form two halftone images and the data was embedded through the correlations between two screens. The embedded pattern can be viewed when the two halftone images are overlaid. Some embedded data in the angular orientation of circularly asymmetric halftone dot patterns that were written into the halftone cells of digital halftone images [30]. One common characteristic of these methods is that they cannot embed a large amount of data without significant perceptual distortion. In [15], we proposed two methods for hiding a fairly large amount of data in halftone images without knowledge of the original multi-tone image and the halftoning method. The first method called Data Hiding Self Toggling (DHST) hides data by forcing pixels at pseudo-random locations to toggle. DHST gives poor visual quality. The second method called Data Hiding Pair Toggling (DHPT) improves on DHST by performing pair toggling rather than self-toggling. The visual quality of DHPT is quite good. In [16], we proposed an algorithm called Data Hiding Smart Pair Toggling (DHSPT) which improves on DHPT by choosing the pair-toggling partner in a smart way to reduce visual artifacts.

In this paper, we propose two methods called Intensity Selection (IS) and Connection Selection (CS) that can be applied to DHSPT, DHPT and DHST to achieve improved visual quality. The two methods allow tradeoff between visual quality and computational complexity. In Section 2, the DHST, DHPT and DHSPT will be reviewed. In Section 3, the two proposed methods, IS and CS, will be introduced. Some objective visual quality measures will also be defined. In Section 4, the simulation results will be presented and discussed.

2. Review of DHST, DHPT and DHSPT

Halftone pixels are inherently binary in nature, taking on values of 0 and 255. It is straightforward to hide one data bit in one halftone pixel. In DHST [15], a pseudo-random number generator with a known seed is used to generate a set of N pseudo-random locations within a halftone image. Then one bit of embedded data is hidden at each pseudo-random location by forcing the halftone pixel to take on values according to the embedded data, regardless of the image content. In other words, when needed, the pixels at the pseudo-random locations are forced to toggle (called self-toggling). Such self-toggling inevitably introduce undesirable visual distortion, which appears as 'salt-and-pepper' artifacts, or local black or white clusters as shown in Fig. 2. While some clusters are small and may not be visually disturbing, some can be large and visually annoying. Some clusters may be located at locations with inherent masking properties and thus would be less visually disturbing. Others may be located at locations with little masking properties and can be very disturbing. Typically, half of the pixels at the N pseudo-random locations are already the desired values and thus no change is necessary. At the remaining half of the locations, self-toggling is performed. To read out the embedded data, the same seed is used to identify the set of pseudo-random locations and the embedded data are extracted by simply examining the halftone values at these pseudo-random locations. A typically DHST image is shown in Fig. 11 which has a lot of 'salt-and-pepper' artifacts. One reason for the low visual quality of DHST is that self-toggling introduces a distortion in the local average intensity which is disturbing to the human eye.

To improve the visual quality, the DHPT performs complementary pair-toggling at the pseudo-random locations. In complementary pair-toggling, when the pixel at a pseudo-random location (which will be called the master pixel) is forced to toggle from black to white, a neighboring complementary pixel (which will be called the slave pixel) with opposite color (white) to that of the master pixel is forced to toggle simultaneously such that the local average intensity is maintained, and vice versa. In the rare case of no slave pixel candidate being present, no complementary pair-toggling is performed and DHPT degenerates into DHST at that location. When two or more candidates for the slave pixel are found, one is chosen randomly as the slave pixel. DHPT has significant visual quality

improvement over DHST as shown in Fig. 12, with significantly fewer 'salt-and-pepper' artifacts.

The DHSPT improves upon DHPT by choosing the slave pixel in a 'smart' way such that the 'salt-and-pepper' clusters are smaller. A quantity called 'connection' is computed for each of the slave pixel candidates. The one with largest before-toggle 'connection' is chosen as the slave pixel. This quantity will be used in the proposed CS. Consider a pixel at location (m,n) and its neighbors in a 3x3 neighborhood. Let the nine pixels in the 3x3 neighborhood be $[x_1\ x_2\ x_3;\ x_4\ x_5\ x_6;\ x_7\ x_8\ x_9]$ in Matlab notation with x_5 being the pixel at location (m,n). The connection *con*(m,n) of the pixel at location (m,n) is defined as

$$con(m,n) = \sum_{i=1}^{9} w(i) f(x_5, x_i), \quad f(x,y) = \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases} \tag{1}$$

where $w(i) = 1$ for $i=1, 3, 7, 9$ and $w(i) = 2$ for $i=2, 4, 6, 8$, and $w(5) = 0$. A larger weight is given to the immediate left, right, above and below pixels because they are closer to the center pixel and are visually more significant when it has the same color as x_5. The *con* is a measure of the connectedness of the pixel at (m,n) to the surrounding neighboring pixels with the same color. For example, when the pixel at a pseudo-random location is black, the *con* of its neighboring white pixels is evaluated. The one with the largest *con* is chosen to perform toggling. After the toggling, the white neighboring pixel becomes black and its connectedness with neighboring black pixels is low, resulting in the small probability of forming a large black cluster. For the master location, since it is fixed, the freedom is limited. To minimize the *con* of the master pixel after toggling, horizontal or vertical slave neighbors are preferred over diagonal neighbors if there are two or more adjacent locations with the same *con*. The visual quality of DHSPT is significantly better than that of DHPT, as shown in Fig. 3.

3. Improved Data Hiding with Intensity and Connection Selection

In this section, we propose a novel algorithm called DHSPT-IS-orig which improves the visual quality of DHSPT by applying a technique called Intensity Selection (IS) based on the original multi-tone image. Out of several possible candidate locations, the IS would choose the best one to apply the DHSPT, according to the local brightness condition of the original multi-tone image. The IS

can also be applied to DHPT and DHST in a similar fashion to give DHPT-IS-orig and DHST-IS-orig to achieve improved visual quality.

When the original image is unavailable which is the case DHSPT, DHPT and DHST are designed for, we propose a modified version of IS based on the inverse halftoned image rather than the original image. The resulting algorithm is simply called DHSPT-IS. The combination of inverse halftoning and Intensity Selection can be applied to DHPT and DHST and the resulting algorithms are simply called DHPT-IS and DHST-IS respectively.

As inverse halftoning tends to require substantial computation which is undesirable, we also propose an alternative location selection scheme called Connection Selection (CS) which require neither the original multi-tone image nor the inverse halftoned image. The CS selects the best location based on the local connection properties of the halftone images and thus eliminates the computation associated with inverse halftoning. The resulting algorithms are called DHSPT-CS, DHPT-CS and DHST-CS. The performance of IS and CS will be compared in Section 4.

The proposed Intensity Selection will be described in Section 3.1. The proposed Connection Selection will be described in Section 3.2. Some objective visual quality measures (S_1 to S_5) for the proposed halftone image data hiding algorithms will be described in Section 3.3.

3.1 Modification of DHST, DHPT and DHSPT with Intensity Selection

To improve the visual quality of DHSPT (and DHPT and DHST also), we propose to hide one bit of the embedded data at M pseudo-random locations instead of just one location. Instead of generating N pseudo-random locations, we generate NM pseudo-random locations which are divided into N groups of M locations. Then one bit is embedded in each group of M locations by forcing the parity of the sum of the M pixels to be even or odd according to the data bit to be embedded. The parity of the M-pixel-sum is changed, when needed, by two steps. The first step is *Intensity Selection (IS)* that selects one out of the M pixels to be changed, basing the decision on the intensity condition of the original multi-tone image. The second step is the toggling of the selected pixel and one of its neighboring complementary pixels by DHSPT. This algorithm is called DHSPT-IS-orig (DHSPT with

IS based on original image). The IS can be applied to DHPT and DHST. If DHPT is used, the resulting algorithm is called DHPT-IS-orig. For DHST, the algorithm is called DHST-IS-orig.

Assume that the embedded data bits and the original halftone image pixels are statistically independent and that each embedded data bit is equally likely to be 0 or 1. Then, with a probability of 0.5, the original M-pixel-sum parity is the desired value and thus no change is needed. With a probability of 0.5, the original M-pixel-sum parity is different from the desired value. In this case, the proposed IS and DHST or DHPT or DHSPT are applied to force the M-pixel-sum parity to be the desired value. To read the embedded data, one simply uses the same random number generator and the same seed to obtain the NM pseudo-random locations. Then the embedded data bits can be extracted by examining the M-pixel-sum parity.

Let $\{(i_1, j_1), (i_2, j_2), \ldots, (i_M, j_M)\}$ be M pseudo-random locations among which one data bit needs to be hidden. These locations are typically far apart, locating at different parts of the image. Let $x(i, j)$, $y(i, j)$ be the original multi-tone image pixel value, and the halftone value at location (i, j) respectively. The M-pixel-sum parity of $\{y(i_1, j_1), y(i_2, j_2), \ldots, y(i_M, j_M)\}$ is defined as $\left(\sum_{k=1}^{M} y(i_k, j_k)\right) \bmod 2$. When the parity is different from the desired value, an odd number of the halftone values $y(i, j)$ should be toggled to change the parity. To minimize the distortion due to the changes, only one of the M pixels will be changed. We will call this pixel location (m, n).

With M locations available, the proposed IS seeks to choose the location at which the distortion due to DHSPT, DHPT or DHST would be least visually annoying. To motivate IS, we will analyze the distortion patterns of DHSPT, DHPT and DHST first. Often within a small neighborhood, the original multi-tone image pixels are approximately constant. Suppose $x(m+i, n+j) = K$ for some $0 \le K \le 255$ and $(i, j) \in \Psi$ where Ψ is a local neighborhood of (m, n). Then any good halftone method would have approximately a fraction $K/255$ of the pixels within Ψ being white and $(1-K)/255$ being black so that the local average intensity is approximately K. When K is small as in the dark regions, there are very few white pixels within Ψ. And the white pixels should be distributed evenly for good visual quality. In other words, the white pixels should be isolated pixels

for good visual quality. If the white pixels are clustered, it becomes effectively a white cluster in dark background, or a visually disturbing 'salt-and-pepper' artifact similar to those in DHST. As an example, error diffusion tends to give locally uniform textures which lead to high quality halftone images. Ordered dithering tends to give less uniform textures which lead to lower quality halftone images. Similarly, when K is large as in the bright regions, there are usually a lot of white pixels and very few black pixels within Ψ. Again, the black pixels should be isolated black pixels evenly distributed in white background for good visual quality. When K is close to 127 as in the mid-gray regions, there are significant numbers of both white pixels and black pixels. In these regions, both the black and white pixels should be evenly distributed for good visual quality. Often, checkerboard or regular patterns [13] would occur in mid-gray regions.

Consider the dark regions with small K. There are very few white pixels sparsely distributed and most of these white pixels are isolated white pixels. When DHST is applied to toggle one of the pixels, there are two scenarios. The first scenario is that DHST forces a black background pixel to be white which is very likely because there are a lot of black pixels in the dark regions. The resulting white pixel tends to be an isolated pixel which is visually pleasing because it looks like the other isolated white pixels. The resulting white pixel may also touch an existing isolated white pixel to form a cluster of 2 white pixels. This tends to be somewhat visually disturbing. But due to its small size, visual degradation is minor. The second scenario is that DHST forces a white pixel to be black. If the white pixel is an isolated white pixel (as in error diffused halftone images), the resulting black pixel merges nicely into the background and would be visually pleasing. If the white pixel is part of a white cluster (as in ordered dithered halftone images), the resulting black pixel reduces the size of the white cluster by one pixel. The resulting visual degradation is usually minor.

When DHPT or DHSPT are applied to the dark regions with small K, there are two scenarios. The more likely scenario is again that DHPT or DHSPT forces a black (master) background pixel to be white. If no white slave pixel candidate is found, the resulting white master pixel is an isolated white pixel like the other isolated white pixels and is visually pleasing. If a white slave pixel is identified, the DHPT and DHSPT would force it to be black in the pair-toggling. Since this is a dark region with sparsely distributed white pixels, it is highly likely that this white slave pixel is an isolated

white pixel. Then the application of DHPT or DHSPT is equivalent to moving this isolated white pixel locally from the slave pixel location to the master pixel location. If the white pixel at the new location is isolated, it is visually pleasing. If it touches another isolated white pixel, the resulting 2-pixel cluster is rather small and would result in minor visual degradation. The second scenario is the less likely case of DHPT or DHSPT forcing a white pixel to be black. As this is a dark region, it is highly likely that the white master pixel is an isolated white pixel and that a black neighboring slave pixel can be identified and forced to be white in the pair toggling of DHPT and DHSPT. This effectively is the movement of the white pixel locally from the master pixel location to the slave pixel location. If the new white pixel is an isolated white pixel, it is visually pleasing. If it touches another isolated white pixel, the 2-pixel cluster can result in minor visual degradation. In other words, the DHST, DHPT and DHSPT most likely do not result in large visually disturbing clusters in the dark regions.

Similar arguments can be applied to bright regions with large K . In other words, the DHST, DHPT and DHSPT most likely do not result in large visually disturbing clusters in bright regions also.

Consider the mid-gray regions with K close to 127. There are typically many black pixels and white pixels both of which somewhat evenly distributed, forming some regular structure or texture. When DHST is applied, it may force a black pixel to be white or a white pixel to be black depending on the data bit to be embedded. Regardless of the color, the result is that the regular structure is locally punctured resulting usually in a visually disturbing large cluster ('salt-and-pepper' artifact). The cluster size can range from 2 to more than 10. The clusters in real images often contain 3, 4 or 5 pixels which are large enough to be visually disturbing, especially against a regular background. And since the background is mid-gray, both black and white clusters are visually disturbing. When DHPT is applied, the regular pattern is still punctured but the resulting 'salt-and-pepper' clusters tend to be 1 pixel smaller than those from DHST due to the pair-toggling resulting in improved visual quality. When DHSPT is applied, the slave pixel is chosen such that the resulting 'salt-and-pepper' clusters are as small as possible. As a result, DHSPT tends to give improved visual quality over DHPT in the mid-gray regions. Consider an example of a white pixel being forced to be black in a checkerboard pattern. The DHST would result in a 5-pixel black cluster in the shape of a

star and would be visually disturbing due to its large size. Both DHPT and DHSPT would result in one 4-pixel black cluster and an adjacent 4-pixel white cluster. These are large clusters disturbing to the human eyes. In other words, the 'salt-and-pepper' clusters due to DHST, DHPT and DHSPT in the mid-gray areas tend to be large resulting in significant visual degradation.

While DHST, DHPT and DHSPT introduce distortions inevitably whenever toggling is applied, the resulting 'salt-and-pepper' clusters tend to be larger and visually more disturbing in mid-gray area than in bright or dark areas. As a result, our proposed Intensity Selection (IS) tries to select locations that are either very bright or very dark for best visual quality.

Assume that the original multi-tone pixels $\{x(i_1, j_1), x(i_2, j_2), \ldots, x(i_M, j_M)\}$ are available. We define the instantaneous eccentricity at location (m,n) as

$$eccentricity(m,n) = |x(m,n) - 127|$$

When eccentricity is small, the pixel is mid-gray and close to 127. When eccentricity is large, the pixel is either very bright or very dark. In IS, we choose the location with the largest eccentricity among the M candidates. The proposed IS is very simple in terms of computational complexity. It has a drawback that it requires the original multi-tone images. When this Intensity Selection is followed by DHSPT, DHPT and DHST, the corresponding algorithms are called DHSPT-IS-orig, DHPT-IS-orig, and DHST-IS-orig respectively.

But DHST, DHPT and DHSPT are designed for cases in which the original halftone images are not available. In those cases, the proposed IS cannot be applied due to the absence of the original image. Here we also propose a modified version of IS which does not need the original images.

The modification is to perform inverse halftoning on the halftone image to obtain an estimate of the original multi-tone image, and then perform IS based on the estimated multi-tone image. While inverse halftoned images are not perfect, our experiments suggest that they are good enough for the proposed IS. The eccentricity tends to be insensitive with respect to inverse halftoning distortion. One price of the modification is the large computational requirement of inverse halftoning.

While there are many existing inverse-halftoning methods such as projection onto convex set (POCS) [29], wavelet-based methods[35] and edge-preserving filtering[1], we find in our experiments

that simple lowpass filtering is sufficient to give good enough estimated values for IS. And lowpass filtering has the lowest complexity among all inverse-halftoning methods. The algorithm that applies inverse halftoning, IS and DHSPT is simply called DHSPT-IS. When the same is applied to DHPT and DHST, they are called DHPT-IS and DHST-IS respectively.

3.2 Modification of DHST, DHPT and DHSPT with Connectivity Selection

In the previous section, we propose to perform inverse halftoning before applying Intensity Selection when the original multi-tone image is absent. However, the inverse halftoning tends to have relatively large computational complexity even for the lowpass filtering. In this section, we propose a new location selection method called Connectivity Selection (CS) which does not require the original multi-tone image or the inverse halftoned image to select the best location out of the M candidate locations. The CS will replace the IS in DHSPT-IS, DHPT-IS and DHST-IS to give DHSPT-CS, DHPT-CS and DHST-CS.

The motivation of Connectivity Selection is that the major distortion of DHSPT, DHPT and DHST are in the form of 'salt-and-pepper' noise, and thus a good location should result in the smallest possible 'salt-and-pepper' clusters.

Consider $\{(i_1, j_1), (i_2, j_2), \ldots, (i_M, j_M)\}$, the M pseudo-random locations among which one data bit needs to be hidden. In the proposed CS, we compute the connection *con*(i,j) defined in Eqn. 1 for each of the M locations and choose the location with the largest connection. Then DHSPT, DHPT or DHST is applied to the chosen location. In this way, the connection of the chosen location is the smallest possible among the M candidates, after the self-toggling. And the size of the resulting 'salt-and-pepper' artifact should be very small, if not the smallest.

In bright regions, there are many white pixels of which the *con* tends to be larger than the *con* in mid-gray regions. Similarly, the *con* of black pixels in dark regions tends to be larger than the *con* in mid-gray regions. As a result, there is a tendency for CS to select locations in the bright and dark regions, similar to those chosen by IS.

3.2 Objective Visual Quality Measures

The peak-signal-to-noise ratio (PSNR) is not suitable for halftone images because all halftone images have low PSNR due to the dominating high frequency halftone noise. A good-looking halftone image may have equally low PSNR as a bad-looking halftone image. Thus we do not use PSNR in this paper. In this section, we analyze the distortion characteristics of DHSPT, DHPT and DHST and define five scores S_1, S_2 S_3, S_4 and S_5 as objective visual quality measure for the algorithms.

Let A be those locations among the NM pseudo-random locations which are chosen by IS or CS and at which toggling is applied. Then the cardinality of A is less than or equal to N. Typically the cardinality is about $N/2$. As toggling is applied to the locations within A and their nearby neighbors, there is no change to the rest of the image. In other words, there is no need to measure the distortion of the rest of the image. It suffices to measure the distortion around the locations within A. The distortion due to the proposed algorithm appears mainly in the form of 'salt-and-pepper' artifacts due to local clusters of pixels as in Fig. 11. Many clusters are formed by 2 to 5 connected pixels of identical color, some white and some black. Large clusters are visually more disturbing than small ones. Thus a good way to measure visual quality is to measure the amount and the size of the 'salt-and-pepper' clusters. To do so, we classify the elements of A into four classes:

1. black halftone pixel in bright region ($y(m,n)=0$, $x(m,n)>127$)
2. white halftone pixel in bright region ($y(m,n)=255$, $x(m,n)>127$)
3. black halftone pixel in dark region ($y(m,n)=0$, $x(m,n)\leq 127$)
4. white halftone pixel in dark region ($y(m,n)=255$, $x(m,n)\leq 127$)

Both class 1 and class 4 pixel clusters can be visually disturbing, while class 2 and class 3 pixel clusters are significantly less disturbing. Class 1 corresponds to originally white halftone pixels in bright regions being forced to be black by the proposed data hiding algorithms. They can be visually disturbing due to the huge contrast between the new pixel color (black) and the background color (bright). Similarly, class 4 pixels can be disturbing. On the contrary, class 2 corresponds to originally black pixels in bright background being forced to be white. Given they are in bright regions, the black pixels are likely to be isolated black pixels. They tend not to be visually disturbing because of the nice

blending of the new color (white) into the background color (bright). Similarly, class 3 pixels tend to be significantly less visually disturbing than class 1 and class 4 pixels.

If a class 1 pixel is an isolated black pixel, it does not look bad because halftoning would produce isolated black pixels naturally, though not necessarily at that location. However, if it is connected to many other black pixels to form a large cluster, it would look strange and annoying because halftoning algorithms would not generate such clusters naturally in the bright regions. Similar comments apply to class 4 pixels as well. We thus define the following five scores for the class 1 and class 4 elements of $\boldsymbol{A}$:

$$S_1 = \sum_{i=0}^{4} N_i \qquad S_2 = \sum_{i=0}^{4} (i+1)N_i \qquad S_3 = \frac{S_2}{S_1} \qquad S_4 = \sum_{i=2}^{4} N_i \qquad S_5 = \sum_{i=0}^{4} iN = S_2 - S_1$$

where N_i is the total number of the class 1 and class 4 elements in $\boldsymbol{A}$ having i neighbors with same pixel values in the 4-neighborhood (left, right, center, top, bottom) after data hiding is performed. The N_o corresponds to the number of visually pleasing isolated class 1 or class 4 elements in $\boldsymbol{A}$. The S_1 gives the total number of class 1 and class 4 elements of $\boldsymbol{A}$. The S_2 gives the total area covered by the clusters associated with the class 1 and class 4 elements of $\boldsymbol{A}$. The S_3 gives the average area per cluster. The S_4 is the number of class 1 and class 4 elements of $\boldsymbol{A}$ associated with clusters of size 3 or more, which is useful because clusters of size 1 or 2 are not very visually disturbing. The S_5 is a perceptual measure with a linear penalty model. It gives a zero penalty score to isolated black or white pixels which look visually pleasing. It gives scores of 1, 2, 3 and 4 for clusters of size 2, 3, 4 and 5 respectively. In general, algorithms with smaller scores of S_1, S_2, S_3, S_4 and S_5 are better.

4. Results and Discussions

The proposed algorithms are tested with five 512x512 test images: 'Lena', 'Pepper', 'Boat', 'Barbara' and 'Harbor'. These five images are halftoned by error diffusion using the Steinberg kernel in Table 12 and by order dithering using the '8x8 dispersed dot' screen in Table 11. All the proposed algorithms are tested on both the error diffused images and the ordered dithered images. For the case

of M=2 in IS or CS, the algorithms are called IS2 or CS2 respectively. Similarly, for the case of M=4, the algorithms are called IS4 or CS4. For example, DHSPT-CS4-orig is the DHSPT-CS-orig with M=4. The scores (S_1 to S_5) of the algorithms are shown in Tables 1 to 10. Due to limited space, the complete set of results is shown only for 'Lena' in Tables 1 and 2. The corresponding images are shown in Fig. 1 to 22. Selected results of the other images are shown in Tables 3 to 10. Selected images of 'Pepper' are shown in Figs. 23 to 30.

The 'Lena' halftoned by error diffusion with the Steinberg kernel is shown in Fig. 1. In Fig. 3, the DHSPT is applied to hide 4096 bits of embedded data. The 4096 bits are a relatively large amount of data to be embedded. They are used to test the algorithms under stress. While the DHSPT gives good visual quality, there are still quite a few 'salt-and-pepper' artifacts, some black and some white. According to Table 1, there are 1336 (S_1) class 1 and class 4 elements of $\boldsymbol{A}$. These class 1 and class 4 clusters occupy a total area (S_2) of 2805 pixels. On the average, each cluster contains 2.10 pixels (S_3) which is quite large. Only 455 (S_4) out of the 1336 'salt-and-pepper' clusters are of size 3 or larger which are visually disturbing. The S_5 is 1469 which is quite large. The corresponding error image is shown in Fig. 4, which shows that the error pixels usually occur in pairs and they are somewhat uniformly distributed within the image.

When the proposed Intensity Selection based on the original multi-tone image is applied in DHSPT-IS2-orig with M=2, there is a drastic decrease in the amount of 'salt-and-pepper' artifacts. The improved visual quality is confirmed by the significantly reduced average cluster size (S_3) from 2.10 to 1.76 (16% reduction). Although there is an increase of class 1 and class 4 clusters (S_1) from 1336 to 1482, the number of large clusters (S_4) is reduced drastically from 455 to 267 (41% reduction). When inverse halftoning using simple lowpass filter is applied before intensity selection is applied in DHSPT-IS2, the resulting image is shown in Fig. 5. Without the information of the original multi-tone image, the visual quality of DHSPT-IS2 is surprisingly close to that of DHSPT-IS2-orig, with much fewer 'salt-and-pepper' artifacts than DHSPT according to Table 1. In fact, the S_2, S_3, S_4 and S_5 of DHSPT-IS2 are slightly lower than those of DHSPT-IS2-orig, with S_1 essentially the

same. This suggests that IS depends mainly on the low frequency information of the multi-tone image which is retained after halftoning and inverse halftoning.

When M is increased from 2 to 4 in DHSPT-IS4-orig, there is further reduction in the 'salt-and-pepper' artifacts. Table 1 shows that the average cluster size (S_3) and the number of large clusters (S_4) reduces significantly compared with DHSPT-IS2-orig. Again, DHSPT-IS4 is slightly better than DHSPT-IS4-orig. The image of DHSPT-IS4 is shown in Fig. 6 which has fewer 'salt-and-pepper' artifacts than DHSPT-IS2. The error image of DHSPT-IS4 in Fig. 9 shows that IS tends to avoid the mid-gray regions and concentrate on the very bright and very dark regions as designed.

When Connection Selection is used instead of IS to achieve lower computational complexity in DHSPT-CS2 with M=2, there is observable drop in visual quality. The image of DHSPT-CS2 is shown in Fig. 7 which has slightly fewer 'salt-and-pepper' artifacts than DHSPT but considerably more than DHSPT-IS2, which are confirmed by Table 1. The average cluster size (S_3) and the number of large clusters (S_4) of DHSPT-CS2 are lower than DHSPT but higher than DHSPT-IS2. Both S_2 and S_5 of DHSPT-CS2 are higher than DHSPT and DHSPT-IS2. It appears that, for M=2, the DHSPT-CS2 achieves lower computational complexity than DHSPT-IS2 at the expense of considerably lower visual quality. And the visual quality of DHSPT-CS2 is not much better than DHSPT.

. When M is increased from 2 to 4 in DHSPT-CS4 as shown in Fig. 8, the visual quality is significantly improved over DHSPT-CS2. This is confirmed by the greatly improved S_3, S_4 and S_5. The visual quality of DHSPT-CS4 is better than DHSPT-IS2 but worse than DHSPT-IS4, as confirmed by Table 1. The error image in Fig. 10 shows that CS tends to avoid the mid-gray region and concentrate on the very bright and very dark regions similar to IS, though to a lesser degree.

When IS and CS are applied to improve upon DHPT and DHST, significant gain is possible also. The 'Lena' with 4096 bits hidden by DHST and DHPT are shown in Figs. 11 and 12 respectively. There are obviously lots of 'salt-and-pepper' artifacts in DHST and fewer in DHPT. The visual quality of DHST is worse than DHPT, which in turn is worse than DHSPT. This is reflected in Table 1. For both DHST and DHPT, the average cluster size is reduced when IS is applied, with more

reduction for larger M. Again, IS based on the original image is not better than IS based on inverse halftoned image. When CS is used instead of IS, observations similar to those for DHSPT applies. The CS2 is worse than IS2, which in turn is worse than CS4. And IS4 has better quality than any other algorithms tested. The DHST-IS4 and DHPT-IS4 in Figs. 13 and 14 are better than the DHST-CS4 and DHPT-CS4 in Figs. 15 and 16, which in turn are better than DHST and DHPT in Figs. 11 and 12 respectively.

Similar observations can be made for the effect of the algorithms on the order dithered 'Lena' and other images. The images related to order dithered 'Lena' are shown in Figs. 17 to 22, and the statistics are reported in Table 2. The results of IS4 and CS4 in 'Pepper', 'Boat', 'Barbara' and 'Harbor' are shown in Tables 3 to 10, with some selected images shown in Figs. 23 to 30. Similar observations can be made for the effect of the algorithms on these other test images. The average over Tables 1 to 10 of DHSPT, DHSPT-CS4 and DHSPT-IS4 are shown in Table 11. These show that DHSPT-CS4 is a significantly improvement over DHSPT, and DHSPT-IS4 is even better than DHSPT-CS4.

In general, the proposed IS and CS can improve the visual quality of the existing algorithms without reducing the data hiding capacity. The visual quality refers to both subjective visual quality, and objective scores S_1 to S_5. Both IS and CS perform better with larger M, when there are more locations to choose from. In our experiments, the CS2 improves the visual quality slightly. The IS2 gives higher visual quality than CS2 at the expense of higher computational complexity. The CS4 is better than IS2, and IS4 is even better than CS4. In our experiments, all the five scores of DHSPT-IS4 are considerably lower than DHSPT-CS4.

5. Conclusion

In this paper, we propose the Intensity Selection (IS) to select one of M possible locations to apply DHSPT, DHPT or DHST. Our experiments suggest that IS based on the original multi-tone image has essentially the same visual quality as IS based on the inverse halftoned image. In the experiments, IS is found to be effective in improving significantly the subjective and objective visual

quality of DHSPT, DHPT and DHST without affecting the data hiding capacity. Usually, the visual quality of IS is better for larger M. Since the computational complexity of inverse halftoning is large, we also propose the Connection Selection (CS) to do the selection without the need of the original images or the inverse halftone images. Consequently, the computational complexity of CS is significantly lower than IS. In our experiments, the CS is found to be able to improve the subjective and objective visual quality of DHSPT, DHPT and DHST without affecting the data hiding capacity. Its visual quality can be considerably lower than IS for the same M, though the difference becomes smaller for larger M. The proposed IS and CS provide a tradeoff between visual quality and computational complexity.

6. Acknowledgement

This work is funded by an RGC CERG grant.

7. References

1. O.C. Au, M.S. Fu, et. al., "Hybrid Inverse Halftoning using Adaptive Filtering", Proc. of IEEE Int. Sym. On Circuits and Systems, Vol. 4, Jun. 1999, pp. 259-262.
2. Z. Baharav, D. Shaked, "Watermarking of Dither Halftoned Images", *Proc. of SPIE Security and Watermarking of Multimedia Contents*, Vol. 1, Jan. 1999, pp. 307-313.
3. B. E. Bayers, "An Optimum Method for Two Level Rendition of Continuous Tone Pictures," *Proc. of IEEE Int. Communication Conf.*, Vol. 1, Jun. 1973, pp. 2611-2615.
4. W. Bender, et al., "Techniques for data hiding", *Proc. of SPIE*, Vol. 2420, Feb. 1995, pp.164-173.
5. A. G. Borg, I. Pitas, "Image Watermarking using DCT Domain Constraints", *Proc. of IEEE Int. Conf. on Image Processing*, Vol. 3, Sept. 1996, pp. 231-234.
6. G. Caronni, "Assuring Ownership Rights for Digital Images", *Proc. of Reliable IT Systems*, 1995.
7. L.M. Chen, H.M. Hang, "An Adaptive Inverse Halftoning Algorithm", *IEEE Trans. on Image Processing*, Vol. 6, No. 8, Aug. 1997, pp.1202-1209.
8. H. Choi, et. al., "Robust Sinusoidal Watermark for Images", *Electronics Letters*, Vol. 35, No. 15, Jul. 1999, pp. 1238-1239.
9. D. Coltuc, P. Bolon, "Watermarking by Histogram Specification", *Proc. of SPIE*, Vol.3657, Jan. 1999, pp.252-263.
10. I.J. Cox, et al., "Secure Spread Spectrum Watermarking for Multimedia", *IEEE Trans. of Image Processing*, Vol. 6, No. 12, Dec. 1997, pp. 1673-1687.

11. S. Craver, et al., "Resolving Rightful Ownerships with Invisible Watermarking Technique: Limitation, Attacks, and Implications", *IEEE Journal on Selected Areas in Communication*, Vol. 16, No. 4, May 1998, pp 573-586.
12. S. Craver, et al., "On the Invertibility of Invisible Watermarking Techniques", *Proc. of IEEE Int. Conf. on Image Processing*, Vol. 1, Oct. 1997, pp. 540-543.
13. Z. Fan, R. Eschbach, "Limit Cycle Behavior of Error Diffusion", *Proc. of IEEE Int. Conf. on Image Processing*, Vol.2, Nov. 1994, pp.1041-1145.
14. R.W. Floyd, L. Steinberg, "An Adaptive Algorithm for Spatial Grayscale," *Proc. Int. Sym. of SID*, 1975, pp. 36-37.
15. M.S. Fu, O.C. Au, "Data Hiding for Halftone Images", *Proc. of SPIE Conf. On Security and Watermarking of Multimedia Contents II*, Jan. 2000.
16. M.S. Fu, O.C. Au, "Data Hiding by Smart Pair Toggling for Halftone Images", to appear in *Proc. of IEEE Int. Conf. On Acoustics, Speech and Signal Processing*, Jun. 2000.
17. S.W. Kim, et al., "Image Watermarking Scheme using Visual Model and BN Distribution", *Electronic Letters*, Vol. 35, No. 3, Feb. 1999, pp.212-214.
18. K.T. Knox, "Digital Watermarking Using Stochastic Screen Patterns", *United States Patent Number 5,734,752.*
19. C. Podilchuk, W. Zeng, "Perceptual watermarking of still images" Proc. of IEEE First Workshop on Multimedia Signal Processing, Vol. 1, Jun. 1997, pp.363-368.
20. B.M. Macq, J.J. Quisquater, "Cryptology for digital TV Broadcasting", *Proc. of the IEEE*, Vol. 83, No. 6, Jun 1995, pp. 944-957.
21. K. Matsui, K. Tanaka, "Video-Steganography", *IMA Intellectual Property Project Proceedings*, Vol. 1, 1994, pp. 187-206.
22. F. Mintzer, et al., "Effective and Ineffective Digital Watermarks", *Proc. of IEEE Int. Conf. on Image Processing*, Vol. 3, Oct. 1997, pp. 9-13.
23. N. Nikolaidis, I. Pitas, "Copyright Protection of Images using Robust Digital Signatures", *Proc. of IEEE Int. Conf. on Acoustics, Speech, Signal Processing*, Vol. 4, May 1996, pp. 2168-2171.
24. R.G. van Schyndel, et al., "A Digital Watermark", *Proc. of IEEE Int. Conf. On Image Processing*, Vol. 2, Nov. 1994, pp. 86-90.
25. K. Seok, Y. Aoki, "Image Data Embedding System for Watermarking using Fresnel Transform", *Proc. of IEEE Int. Conf. On Multimedia Computing and Systems*, Vol. 1, Jun. 1999, pp. 885-889.
26. V. Solachidis, I. Pitas, "Circularly Symmetric Watermark Embedding in 2-D DFT Domain", *Proc. of IEEE Int. Conf. On Acoustics, Speech, Signal Processing*, Vol. 6, Mar. 1999, pp.3469-3472.
27. M.D. Swanson, et. al., "Multiresolution Scene-based Video Watermarking using Perceptual Models", *IEEE J. on Selected Areas in Communications*, Vol. 16, No. 4, May 1998, pp. 540-550.
28. H. Ogawa H, et. al. "Digital Watermarking Technique for Motion Pictures based on Quantization", *IEICE Trans. on Fundamentals of Electronics Communications & Computer Sciences*, Vol.E83-A,

No.1, Jan. 2000, pp.77-89.

29. N.T. Thao, "Set Theoretical Inverse Halftoning", *Proc. of IEEE Int. Conf. On Image Processing*, Vol. 1, Oct. 1997, pp. 783-786.
30. R. T. Tow, "Methods and Means for Embedding Machine Readable Digital Data in Halftone Images", *United States Patent Number 5,315,098.*
31. R. A. Ulichney, *Digital Halftoning*, Cambridge, MA, MIT Press, 1987.
32. G. Voyatzis, I. Pitas, "Applications of Toral Automorphisms in Image Watermarking", *Proc. of IEEE Int. Conf. On Image Processing*, Vol. 2, Sept. 1996, pp. 237-240.
33. H.J.M. Wang, et. al., "Wavelet based Blind Watermark Retrieval Technique", *Proc. of SPIE*, Vol. 3528, Nov. 1998, pp.440-451.
34. S. G. Wang, "Digital Watermarking Using Conjugate Halftone Screens", *United States Patent Number 5,790,703.*
35. Z. Xiong, et. al., "Inverse Halftoning Using Wavelets", *Proc. of IEEE Int. Conf. on Image Processing*, Vol. I, Oct. 1996, pp 569-572.

	N_0	N_1	N_2	N_3	N_4	S_1	S_2	S_3	S_4	S_5
DHST	113	420	429	284	90	1336	3826	2.86	803	2490
DHST-CS2	215	645	605	262	48	1775	4608	2.60	915	2833
DHST-CS4	358	1004	509	101	13	1985	4362	2.20	623	2377
DHST-IS2	198	607	496	160	22	1483	3650	2.46	678	2167
DHST-IS4	306	901	369	42	2	1620	3393	2.09	413	1773
DHST-IS2-orig	194	599	491	169	29	1482	3686	2.49	689	2204
DHST-IS4-orig	289	892	364	57	5	1607	3418	2.13	426	1811
DHPT	331	492	361	152	0	1336	3006	2.25	513	1670
DHPT-CS2	574	701	416	84	0	1775	3560	2.01	500	1785
DHPT-CS4	967	782	210	26	0	1985	3265	1.64	236	1280
DHPT-IS2	550	645	256	32	0	1483	2736	1.84	288	1253
DHPT-IS4	866	658	90	6	0	1620	2476	1.53	96	856
DHPT-IS2-orig	532	639	264	47	0	1482	2790	1.88	311	1308
DHPT-IS4-orig	831	662	102	12	0	1607	2509	1.56	114	902
DHSPT	425	456	352	103	0	1336	2805	2.10	455	1469
DHSPT-CS2	715	642	363	55	0	1775	3308	1.86	418	1533
DHSPT-CS4	1157	648	160	20	0	1985	3013	1.52	180	1028
DHSPT-IS2	675	555	228	24	1	1483	2570	1.73	253	1087
DHSPT-IS4	1057	491	70	2	0	1620	2257	1.39	72	637
DHSPT-IS2-orig	665	550	231	35	1	1482	2603	1.76	267	1121
DHSPT-IS4-orig	1027	495	79	6	0	1607	2278	1.42	85	671

Table 1 Scores (S_1 to S_5) of various algorithms for error diffused 'Lena' (Steinberg)

	N_0	N_1	N_2	N_3	N_4	S_1	S_2	S_3	S_4	S_5
DHST	317	160	173	402	265	1317	4089	3.10	840	2772
DHST-CS2	556	289	272	440	154	1711	4480	2.62	866	2769
DHST-CS4	730	541	322	276	51	1920	4137	2.15	649	2217
DHST-IS2	490	286	279	338	65	1458	3576	2.45	682	2118
DHST-IS4	577	534	345	132	8	1596	3248	2.04	485	1652
DHST-IS2-orig	459	280	279	317	99	1434	3619	2.52	695	2185
DHST-IS4-orig	553	486	362	154	14	1569	3297	2.10	530	1728
DHPT	477	171	404	264	1	1317	3092	2.35	669	1775
DHPT-CS2	845	271	439	156	0	1711	3328	1.95	595	1617
DHPT-CS4	1271	322	276	51	0	1920	2947	1.53	327	1027
DHPT-IS2	776	279	338	62	3	1458	2611	1.79	403	1153
DHPT-IS4	1111	345	132	8	0	1596	2229	1.40	140	633
DHPT-IS2-orig	739	277	318	99	1	1434	2648	1.85	418	1214
DHPT-IS4-orig	1039	362	153	15	0	1569	2282	1.45	168	713
DHSPT	477	169	401	261	9	1317	3107	2.36	671	1790
DHSPT-CS2	845	269	441	154	2	1711	3332	1.95	597	1621
DHSPT-CS4	1271	319	276	53	1	1920	2954	1.54	330	1034
DHSPT-IS2	776	277	339	61	5	1458	2616	1.79	405	1158
DHSPT-IS4	1111	345	132	6	2	1596	2231	1.40	140	635
DHSPT-IS2-orig	739	276	317	96	6	1434	2656	1.85	419	1222
DHSPT-IS4-orig	1039	361	152	16	1	1569	2286	1.46	169	717

Table 2 Scores (S_1 to S_5) of various algorithms for order dithered 'Lena'

	N_0	N_1	N_2	N_3	N_4	S_1	S_2	S_3	S_4	S_5
DHST	132	414	570	265	47	1428	3965	2.78	882	2537
DHPT	361	564	435	67	1	1428	3067	2.15	503	1639
DHPT-CS4	962	775	214	11	0	1962	3198	1.63	225	1236
DHPT-IS4	850	578	163	9	0	1600	2531	1.58	172	931
DHSPT	403	637	325	61	2	1428	2906	2.04	388	1478
DHSPT-CS4	1094	719	143	6	0	1962	2985	1.52	149	1023
DHSPT-IS4	961	523	108	7	1	1600	2364	1.48	116	764

Table 3 Scores of various algorithms for error diffused 'Boat' (Steinberg)

	N_0	N_1	N_2	N_3	N_4	S_1	S_2	S_3	S_4	S_5
DHST	326	181	218	516	115	1356	3981	2.94	849	2625
DHPT	507	218	513	116	2	1356	2956	2.18	631	1600
DHPT-CS4	1363	333	276	21	0	1993	2941	1.48	297	948
DHPT-IS4	1117	330	177	16	3	1643	2387	1.45	196	744
DHSPT	507	216	506	115	12	1356	2977	2.20	633	1621
DHSPT-CS4	1363	333	274	22	1	1993	2944	1.48	297	951
DHSPT-IS4	1117	329	171	23	3	1643	2395	1.46	197	752

Table 4 Scores of various algorithms for order dithered 'Boat'

	N_0	N_1	N_2	N_3	N_4	S_1	S_2	S_3	S_4	S_5
DHST	201	346	538	260	44	1389	3767	2.71	842	2378
DHPT	375	561	370	83	0	1389	2939	2.12	453	1550
DHPT-CS4	1050	709	224	13	0	1996	3192	1.60	237	1196
DHPT-IS4	956	563	113	3	0	1635	2433	1.49	116	798
DHSPT	451	558	324	55	1	1389	2764	1.99	380	1375
DHSPT-CS4	1160	666	162	8	0	1996	3010	1.51	170	1014
DHSPT-IS4	1062	485	87	1	0	1635	2297	1.40	88	662

Table 5 Scores of various algorithms for error diffused 'Pepper' (Steinberg)

	N_0	N_1	N_2	N_3	N_4	S_1	S_2	S_3	S_4	S_5
DHST	371	168	210	458	166	1373	3999	2.91	834	2626
DHPT	538	211	456	165	3	1373	3003	2.19	624	1630
DHPT-CS4	1399	258	287	21	0	1965	2860	1.46	308	895
DHPT-IS4	1250	274	159	1	0	1684	2279	1.35	160	595
DHSPT	539	209	448	170	7	1373	3016	2.20	625	1643
DHSPT-CS4	1400	257	288	20	0	1965	2858	1.45	308	893
DHSPT-IS4	1250	274	158	1	1	1684	2281	1.35	160	597

Table 6 Scores of various algorithms for order dithered 'Pepper'

	N_0	N_1	N_2	N_3	N_4	S_1	S_2	S_3	S_4	S_5
DHST	164	461	513	258	56	1452	3937	2.71	827	2485
DHPT	428	546	372	103	3	1452	3063	2.11	478	1611
DHPT-CS4	1149	715	166	15	0	2045	3137	1.53	181	1092
DHPT-IS4	1030	572	88	5	0	1695	2458	1.45	93	763
DHSPT	496	527	345	76	8	1452	2929	2.02	429	1477
DHSPT-CS4	1280	627	127	11	0	2045	2959	1.45	138	914
DHSPT-IS4	1196	421	74	4	0	1695	2276	1.34	78	581

Table 7 Scores of various algorithms for error diffused 'Barbara' (Steinberg)

	N_0	N_1	N_2	N_3	N_4	S_1	S_2	S_3	S_4	S_5
DHST	388	204	179	426	218	1415	4127	2.92	823	2712
DHPT	587	183	418	213	14	1415	3129	2.21	645	1714
DHPT-CS4	1483	277	228	20	2	2010	2811	1.40	250	801
DHPT-IS4	1320	249	123	6	1	1699	2216	1.30	130	517
DHSPT	592	177	408	202	36	1415	3158	2.23	646	1743
DHSPT-CS4	1485	274	225	26	0	2010	2812	1.40	251	802
DHSPT-IS4	1320	249	123	3	4	1699	2219	1.31	130	520

Table 8 Scores of various algorithms for order dithered 'Barbara'

	N_0	N_1	N_2	N_3	N_4	S_1	S_2	S_3	S_4	S_5
DHST	81	266	472	392	108	1319	4137	3.14	972	2818
DHPT	194	416	530	178	1	1319	3333	2.53	709	2014
DHPT-CS4	603	830	519	47	0	1999	4008	2.01	566	2009
DHPT-IS4	569	623	314	21	2	1529	2851	1.86	337	1322
DHSPT	252	471	444	149	3	1319	3137	2.38	596	1818
DHSPT-CS4	712	851	401	35	0	1999	3757	1.88	436	1758
DHSPT-IS4	652	622	240	14	1	1529	2677	1.75	255	1148

Table 9 Scores of various algorithms for error diffused 'Harbor' (Steinberg)

	N_0	N_1	N_2	N_3	N_4	S_1	S_2	S_3	S_4	S_5
DHST	203	78	81	547	339	1248	4485	3.59	967	3237
DHPT	272	86	543	341	6	1248	3467	2.78	890	2219
DHPT-CS4	826	264	705	80	0	1875	3789	2.02	785	1914
DHPT-IS4	676	275	488	13	7	1459	2777	1.90	508	1318
DHSPT	276	78	539	331	24	1248	3493	2.80	894	2245
DHSPT-CS4	833	243	715	83	1	1875	3801	2.03	799	1926
DHSPT-IS4	676	272	491	12	8	1459	2781	1.91	511	1322

Table 10 Scores of various algorithms for order dithered 'Harbor'

	S_1	S_2	S_3	S_4	S_5
DHSPT	1367	3006	2.21	550	1639
DHSPT-CS4	1975	3109	1.58	306	1134
DHSPT-IS4	1616	2378	1.48	175	762

Table 11 Summary of DHSPT, DHSPT-CS4 and DHSPT-IS4 over 10 tables

0	32	8	40	2	34	10	42
48	16	56	24	50	18	58	26
12	44	4	36	14	46	6	38
60	28	42	20	62	30	44	22
3	35	11	43	1	33	9	41
51	19	59	27	49	17	57	25
15	47	7	39	13	45	5	37
63	31	45	23	61	29	43	21

Table 12 '8x8 dispersed-dot' ordered dithering screen matrix

		7
3	5	1

Table 13 Steinberg kernel

Fig. 1 Lena halftoned by error diffusion (Steinberg)

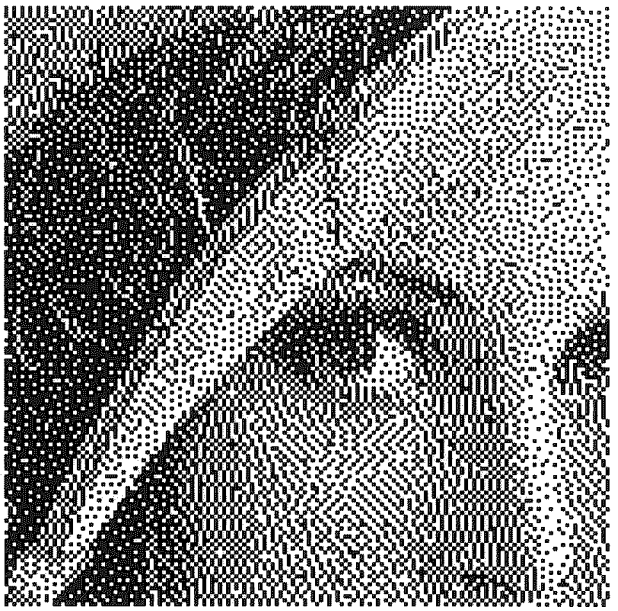

Fig. 2 'Salt-and-pepper' artifacts due to self-toggling

Fig. 3 Lena with 4096 bits (Steinberg, DHSPT)

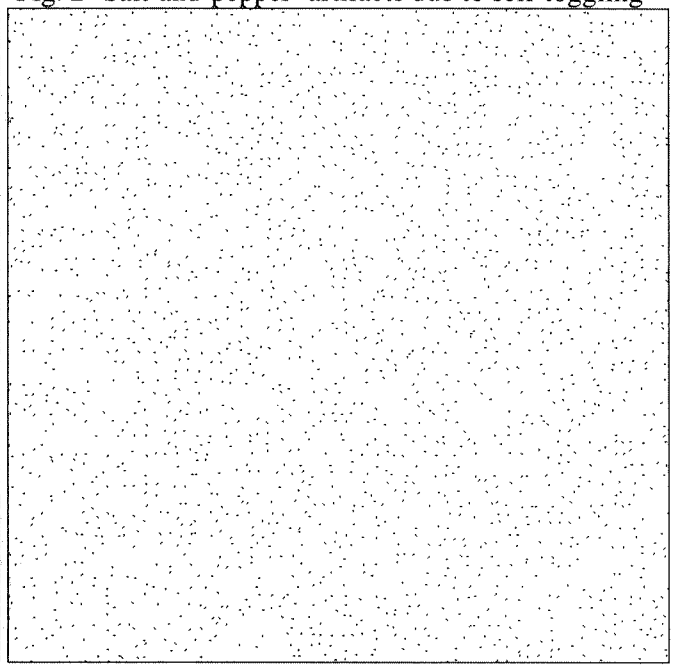

Fig. 4 Error image for DHSPT (Steinberg)

Fig. 5 Lena with 4096 bits (Steinberg, DHSPT-IS2)

Fig. 6 Lena with 4096 bits (Steinberg, DHSPT-IS4)

Fig. 7 Lena with 4096 bits (Steinberg, DHSPT-CS2)

Fig. 8 Lena with 4096 bits (Steinberg, DHSPT-CS4)

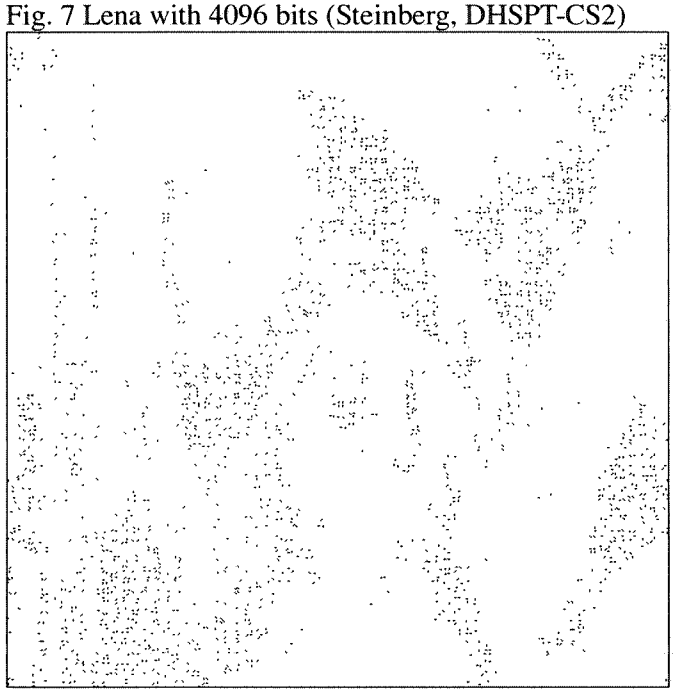

Fig. 9 Error image for DHSPT-IS4 (Steinberg)

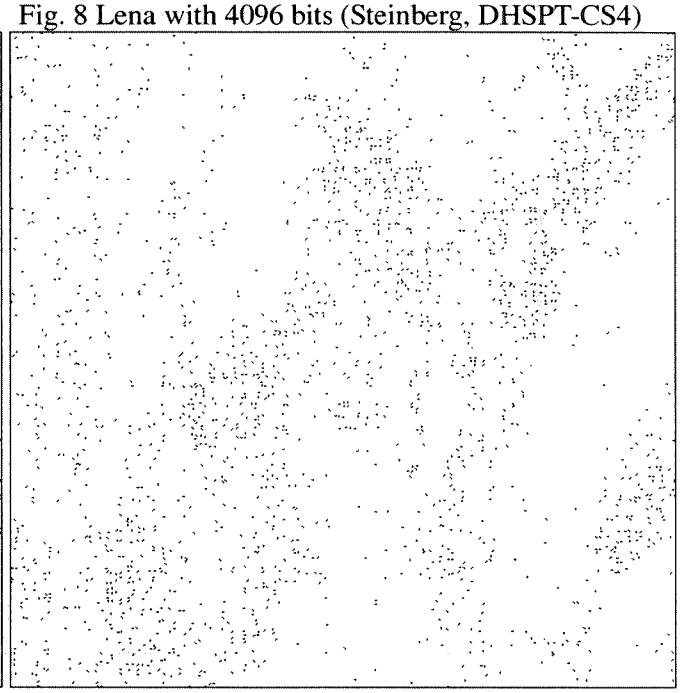

Fig. 10 Error image for DHSPT-CS4 (Steinberg)

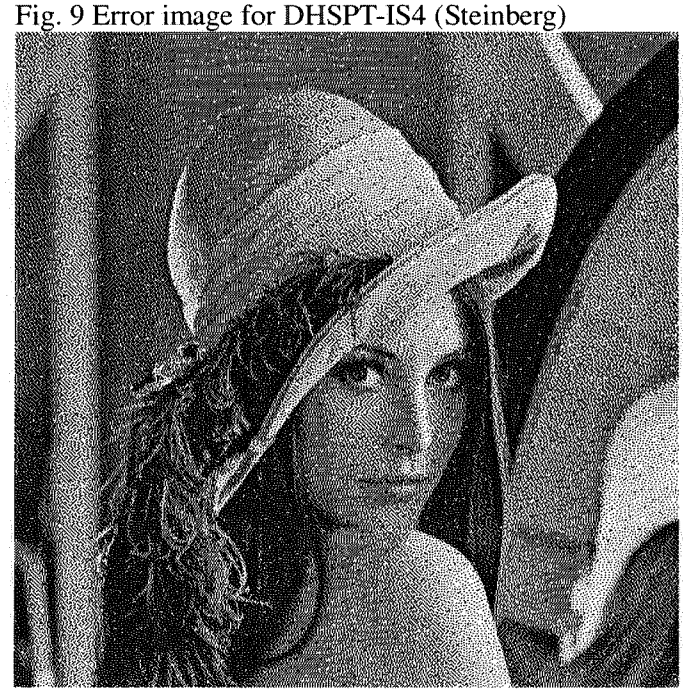

Fig. 11 Lena with 4096 bits (Steinberg, DHST)

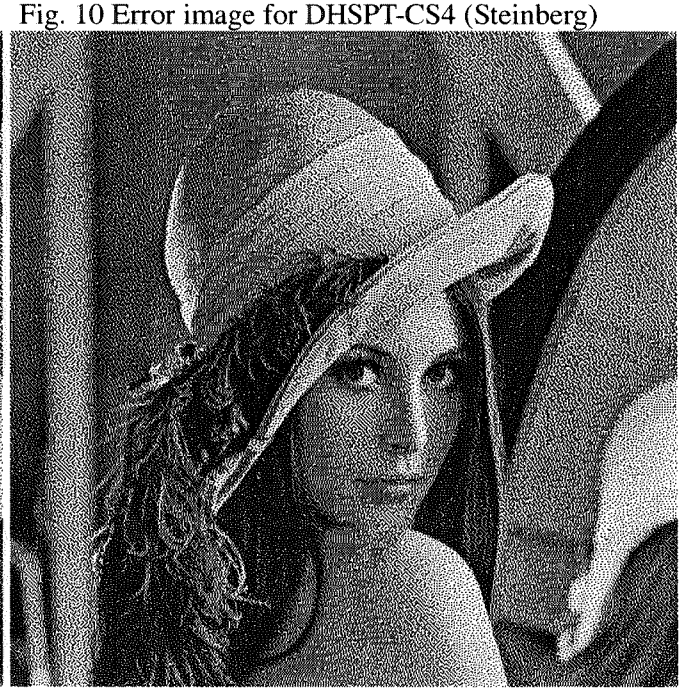

Fig. 12 Lena with 4096 bits (Steinberg, DHPT)

Fig. 13 Lena with 4096 bits (Steinberg, DHST-IS4)

Fig. 14 Lena with 4096 bits (Steinberg, DHPT-IS4)

Fig. 15 Lena with 4096 bits (Steinberg, DHST-CS4)

Fig. 16 Lena with 4096 bits (Steinberg, DHPT-CS4)

Fig. 17 Lena halftoned by order dithering

Fig. 18 Lena with 4096 bits (dither, DHSPT)

Fig. 19 Lena with 4096 bits (dither, DHSPT-IS2)

Fig. 20 Lena with 4096 bits (dither, DHSPT-IS4)

Fig. 21 Lena with 4096 bits (dither, DHSPT-CS2)

Fig. 22 Lena with 4096 bits (dither, DHSPT-CS4)

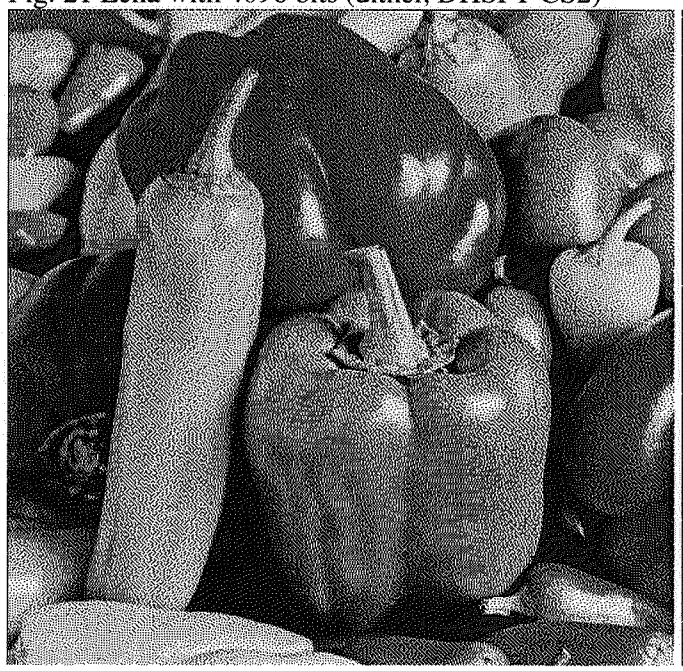

Fig. 23 Pepper halftoned by error diffusion (Steinberg)

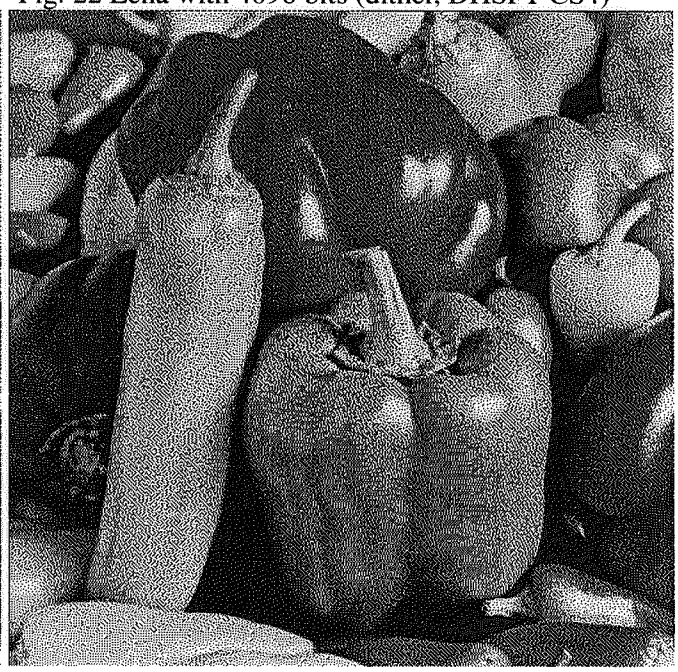

Fig. 24 Pepper with 4096 bits (Steinberg, DHSPT)

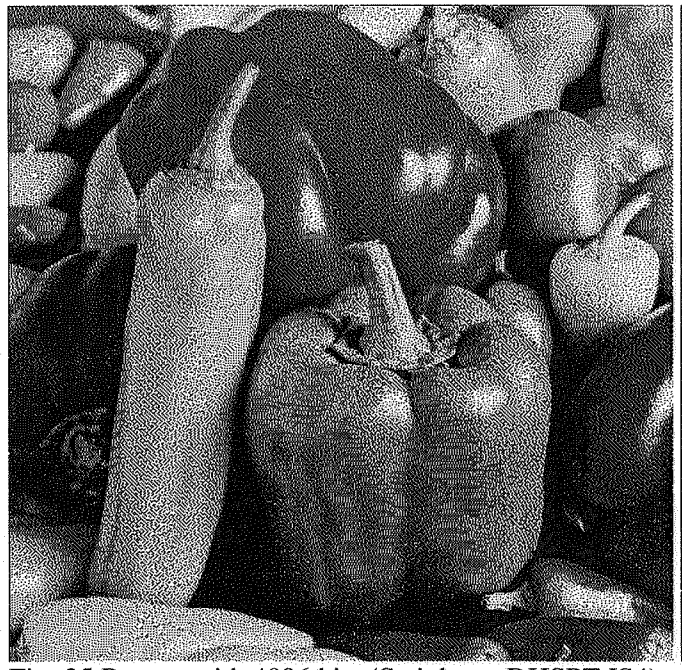

Fig. 25 Pepper with 4096 bits (Steinberg, DHSPT-IS4)

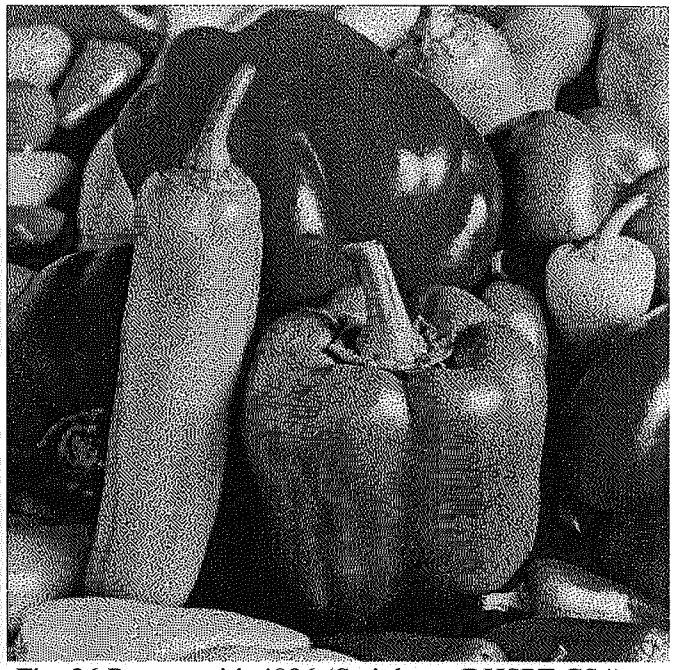

Fig. 26 Pepper with 4096 (Steinberg, DHSPT-CS4)

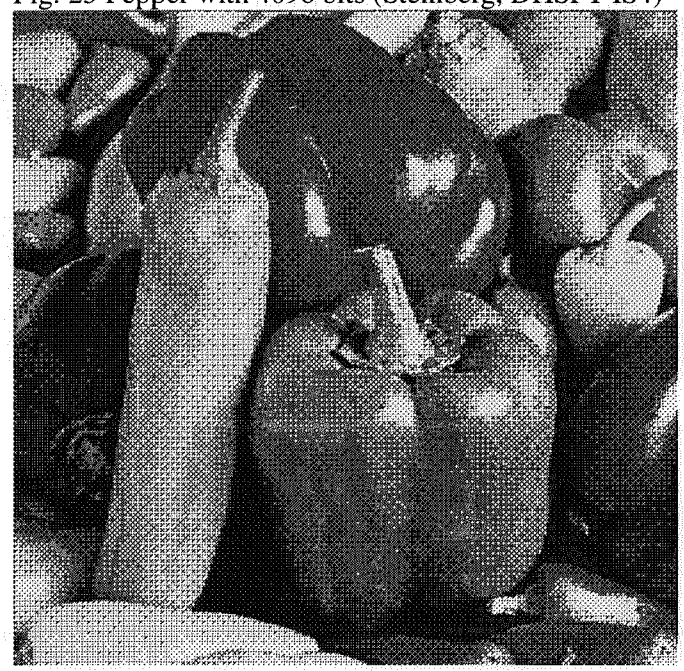

Fig. 27 Pepper halftoned by ordered dithering

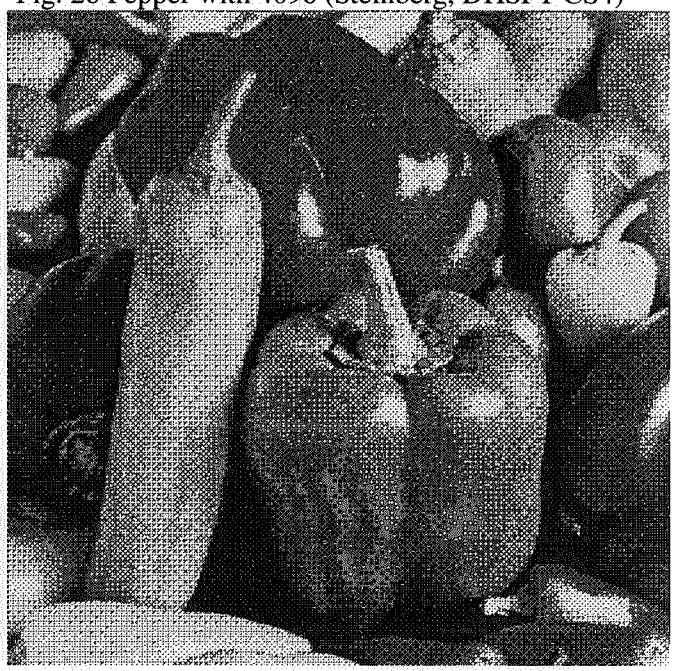

Fig. 28 Pepper with 4096 bits (dither, DHSPT)

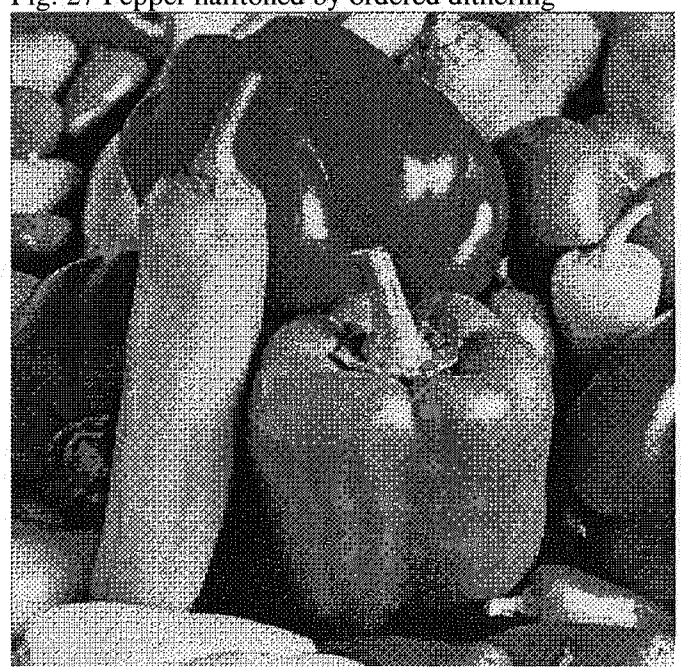

Fig. 29 Pepper with 4096 bits (dither, DHSPT-IS4)

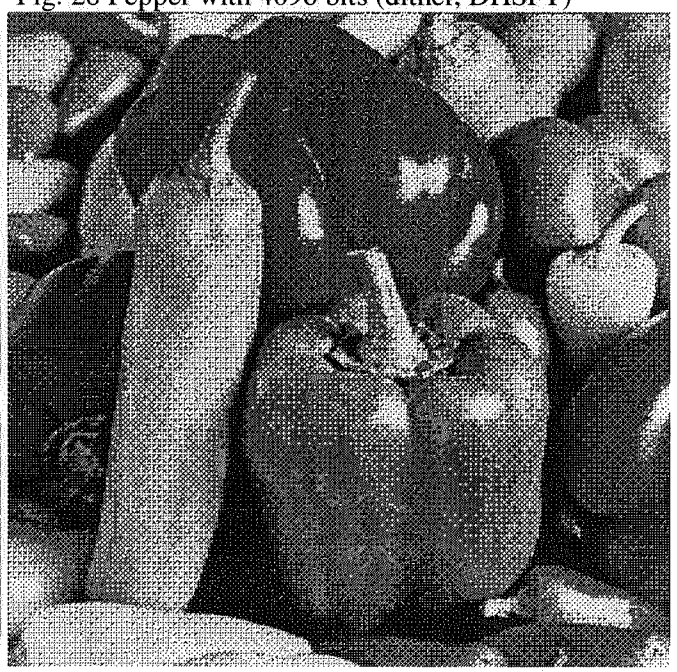

Fig. 30 Pepper with 4096 bits (dither, DHSPT-CS4)