SOME DESIGN ISSUES FOR ROBUST DATA HIDING SYSTEMS

Mahalingam Ramkumar and Ali N. Akansu

The Department of Electrical and Computer Engineering
New Jersey Institute of Technology
New Jersey Center for Multimedia Research
University Heights, Newark, NJ 07102

Abstract

We explore the intricacies of the duality of data hiding and data compression to help develop optimal data hiding techniques for images, that can reasonably resist lossy compression. The problem of efficient data hiding is split into two sub-problems - one of maximizing the *resource* - which is the *permitted distortion* of images, and the other of *efficient use of the resource* by means of sophisticated *signaling techniques.* Various options for the solutions of the former problem are proposed and their advantages and disadvantages explored. An optimal data hiding scheme making "good" use of the resource, using sophisticated signaling techniques [7, 8] is proposed.

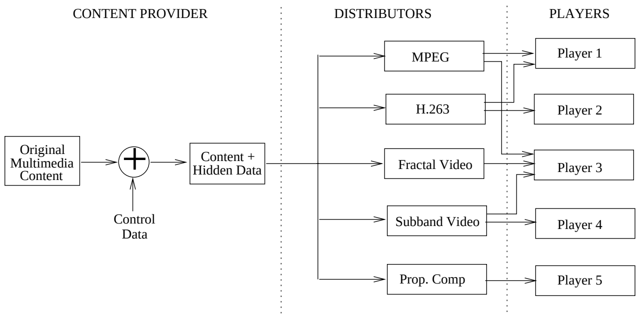

Figure 1: Block Diagram of a Video Distribution System

1. Introduction

Growing concerns over protection of intellectual property rights of digital multimedia, has resulted in an explosive growth of the field of data hiding, or multimedia steganography. Data hiding is an art of modifying the original multimedia content imperceptibly to hide some information. In this paper, we focus on data hiding applications and methods for images. We restrict ourselves to applications that only require robustness to *lossy compression.* An example of such an application is data hiding for multimedia delivery.

Data Hiding is expected to be a boon for multimedia content providers. Content providers can expect to communicate with *compliant multimedia players* through the subliminal channel provided by data hiding. This communication could control access, provide customized delivery, and provide solutions for pay-per-view implementations. A compliant multimedia player would honor an agreed upon protocol for extracting (and abiding by) the hidden control information. Figure 1 is a block diagram of a possible video delivery system. Content providers (the creators of multimedia content) can hide pertinent control information for the multimedia players and make it available for distribution. The distributors may compress the content using some standard or proprietary compression method before it reaches the end users (or their multimedia players). The content may be distributed by several distributors in different formats, understandable by different players. However, as long as all such players follow an established protocol for extracting the hidden information, *and the hidden data is able to survive all the lossy compression methods employed by the distributors*, the content providers can indirectly *control* compliant players through the hidden information. Hiding the information in the raw multimedia data ensures that the hidden data stays embedded forever in the content.

Unless the hidden data is extracted with a "reasonable degree of certainty", the compliant multimedia players may refuse to play the content. Thus intentional tampering for the purpose of removing the hidden information only serves to make that particular copy of the content unusable. On the other hand, the motivation to make it robust to *all* compression methods is to facilitate more efficient distribution of the content. Failure of the hidden data to survive a "good" compression method, makes that compression method unusable for distributing that content.

In Section 2, we investigate the inverse relationship between efficiency of lossy compression and efficiency of data hiding. Therefore efficient data hiding should utilize *holes* in the compression methods. We point out a "hole" common to *all* known compression methods, and suggest methods to utilize that hole for efficient data hiding. In Section 3 we outline the overall data hiding technique and report its performance for some test images for JPEG and SPIHT compression.

2. Compression and Data Hiding

Multimedia compression tries to convey the information content of the multimedia content in as efficient a way as possible - with the fewest number of bits. Data hiding on the other hand tries to sneak in additional bits of information into the content. As the "additional information" does noth-

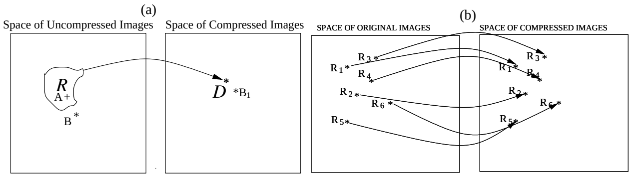

Figure 2: (a) A lossy compression - decompression sequence maps all points in the range R to a single point in the domain D (b) Known Compression Method

ing to improve the quality of the content, an ideal compressor would completely suppress the hidden information.

Let $\mathcal{I}$ represent the space of $M \times N$ images of b bits per pixel (2^{MNb} possible images). Alternately, every point in $\mathcal{I}$ is an $M \times N$ image. As the image is represented by fewer bits in the compressed domain, many original image points are mapped by the compressor to one image point after (lossy) compression and decompression. As an example, in Figure 2 (a), all points in the range R are mapped to a single point D. Consider an image A (represented by $+$) in the region R. Let us say we want to hide one bit of information in the image A that would survive compression. The space $\mathcal{I}$ is *completely* tiled by two regions that represent 0 or 1. For example, if the image A is located in a region representing 0, it could be left intact if the bit to be hidden is 0. To hide a bit 1 however, A has to be moved to a point B (represented by $*$) which simultaneously belongs to region 1 and lies *outside the range* R, so that after compression (and decompression), the image is mapped to a different point B_1. For hiding n_b bits in an image which can survive compression, the image has to be distorted such that after decompression the image is mapped to any of 2^{n_b} possible points. In other words, the space of images has to be tiled by 2^{n_b} regions.

Now it is easy to see that no data hiding would be possible with an *ideal* compressor. If δ_t is the visual distortion permitted (δ_t *may not* be a measure of the mean square error), then there exists a finite number of points to which the original image may be "moved". However, an ideal compressor with the same threshold $\delta_c = \delta_t$ would map all such points to a *single point* in the space of decompressed images! Unless we employ different standards (a measure of δ) for the quality of the image after data hiding and that for the decompressed image, (or unless $\delta_c > \delta_t$), *no data hiding would be possible with ideal compressors.* However, practical compression techniques are not ideal. Therefore, efficient design of data hiding should utilize the holes in compression techniques.

Consider the space $\mathcal{I}$ of original images. When the compression method is known, we make use of the fact that points (or "states") R_1 to R_n are mapped to the same points R_1 to R_n in the space of decompressed images, as shown in Figure 2 (b). Therefore, the number of valid "states" of the compression method that lie within an envelope of "unnoticeable visual distortion" is a direct measure of the number of bits that can be hidden in an image. If there are P such valid "states" then $\log_2 P$ bits can be hidden in the image. If the hidden data has to survive *multiple* known compression methods, the space $\mathcal{I}$ is partitioned into *non-overlapping* regions such that each region contains at least one valid state of each compression scheme. If P_1 such states are available within the envelope of visual threshold, the $\log_2 P_1$ bits can be hidden in the image. Obviously $P_1 << P$.

If the exact effect of compression is not known (the valid states are not know *a priori*), the job of designing efficient data hiding methods warrants a totally different approach. Since one has no idea of the "valid" compression points (or valid compressed images for that particular compression method), the centers of the non-intersecting regions have to *considerably well separated* to ensure that *at least one valid compression point of all compression methods* lies in *every* non-intersecting hyper-sphere. However, the following questions arise:

- Large distance between the centers of the hyper-spheres implies that it may be necessary to introduce a significant amount of distortion to move the image to a desired "state". Is it possible to do that without affecting the visual fidelity of the image?
- Assuming it is possible to introduce a significant amount of distortion without affecting the visual fidelity to move the image A to a new point $\hat{A}$, why should a good compressor map two *visually identical* images A and $\hat{A}$ to different points in the compressed domain?

The answer to the second question is the following; All known compression methods try to minimize the *mean square error* between the original and the compressed image. In fact, the new generation of compression methods (like EZW, SPIHT and IFS (fractal) image compression) even more so than the DCT based JPEG. This is a hole common to all compression methods and can be used effectively for data hiding, if satisfactory answers to the first question exist.

2.1. Utilizing the "Hole" in Compression Schemes

It is clear that if the images can be modified considerably in the mean square sense without affecting the visual fidelity of the image, then one could achieve large separation between "states" corresponding to different bit sequences, and thus achieve robust data hiding.

One solution to this problem (of trying to introduce as much distortion as possible without affecting the visual fidelity) is to use good models of "visual thresholds" (for example, see Ref. [1]) to embed the hidden bits. Many data hiding methods that utilize these models have been proposed. However, a main draw back of these methods is that well defined visual threshold models (say in the DCT or wavelet domain) also suggest the compression techniques means to improve their performance. Thus when one uses these models to add significant amount of signature energy to certain coefficients of the image, a better compression technique which may evolve in the future may also make use of these visual thresholds to perhaps *quantize those coefficients more coarsely.* In other words *utilizing such visual threshold models indirectly amounts to utilizing holes that can be easily "plugged" in the future.* One of the main advantages of data hiding is that the hidden data stays with the content *for ever.* As compression techniques improve in the future, content *distribution* becomes more efficient. But

Original Image　　After Histogram Modification　　After StirMark

Figure 3: Left : Original Goldhill image. Center: Goldhill image obtained by modifying the histogram Right : After StirMark

if the hidden data is not able to survive those compression methods, the content loses its value. Therefore, more useful data hiding techniques should utilize holes which are very difficult to plug.

Figure 3 depicts the original 256×256 Goldhill image, its histogram reshaped version, and image after StirMark [2] (StirMark is a watermark attack software that introduces imperceptible geometric distortions in the image). Though the second and third images are very close to the original in visual fidelity, their PSNRs are 20 and 19 dB respectively! It is clear that significant amounts of distortion (in the MSE sense) can be tolerated as long as the introduced distortion *only modifies the histogram or introduces small geometric distortions, or perhaps, both.* Hence if we are able to embed the hidden data by introducing geometric distortions and / or histogram modification, a large separation between different "states" can be obtained.

However, things may not be as simple as it seems at first glance. Let $\mathcal{H}(I)$ be a function of the histogram of the pixels of an image I. If we try to embed data by *specifying* $\mathcal{H}(I)$ [3], the hidden data will *not* be robust to compression. Even small modifications in the MSE (like what may typically be introduced by lossy compression) can change the histogram significantly. Similarly, if $\mathcal{G}(I)$ is a function of some geometric features [1] of the image I, $d(\mathcal{G}(I), \mathcal{G}(I_1))$ may be large even if $d(I, I_1)$[2] is small. Just as introduction of small geometric distortions can cause a significant change in the MSE, introduction of small distortions in the MSE may cause significant changes to $\mathcal{G}(.)$. This is the reason that the watermarking technique proposed by Rongen et. al [4] is robust to StirMark, but not very robust to JPEG compression). In order achieve robustness to compression, the well separated "states" (corresponding to the bit sequence to be embedded) have to be *specified first.* Then geometric distortions and / or histogram modifications have to be introduced to move the image close to the specified state. However, there may not be a simple or even methodical way to do this. But if such a method can be found and implemented with reasonable degree of computational complexity,[3] it promises to be an excellent solution to the problem of robust data hiding.

Until such a solution can be found, there exists another practical solution to introduce a large amount of distortion in the image without affecting its visual fidelity. Figure 4 (left) shows the original 256×256 Boats image. The second image (center, 14.1 dB PSNR) was derived by retaining the DFT phases of the original image and choosing random magnitudes. In spite of the very low PSNR of the image, we see that a significant amount of "information" about the original image is preserved. The third image (right, 15.6 dB PSNR) was derived by retaining the magnitudes of the DFT coefficients of the original image but choosing the DFT phases randomly. Even though the PSNR of the third image is 1.5 dB better than that of the second, the resulting image conveys almost no information about the original. This illustrates the well known fact that the human visual system (HVS) is much more sensitive to DFT phase than DFT magnitudes [5].

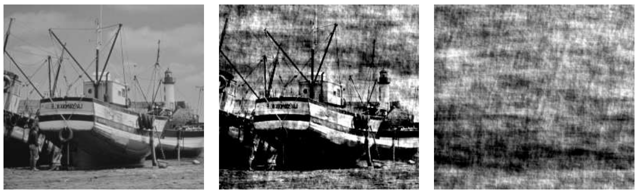

Figure 4: Left : Original Boats image. Center: Boats image obtained by retaining the DFT phases of the original image and choosing random magnitudes (PSNR 14.1 dB). Right: Image obtained by retaining DFT magnitudes of the original and choosing random DFT phases (PSNR 15.6 dB).

Thus if the data embedding is done in the magnitude DFT domain (the "states" are specified by their magnitude DFT coefficients - embedding the data changes the magnitudes of the DFT coefficients of the original image, but leaves the phase intact) a significant amount of distortion (in the MSE sense) can be introduced without affecting the visual fidelity of the image. In addition, unlike the use of well defined visual threshold models, this is not a hole that is capable of being easily "plugged" in the future (compression techniques that utilize the DFT and quantize the magnitudes coarsely and the phases finely have been proposed earlier, but have not been effective [6]).

3. The Data Hiding Scheme

Figure 5 show the block diagrams of data embedding and the data detection schemes. The overall data hiding technique consists of a combination of transforms $\mathcal{T}$, to obtain $\mathbf{c} = \mathcal{T}(I, \mathcal{K})$, where $\mathcal{K}$ is the *private key*, and $\mathbf{c} \in \Re^D$. The bit sequence to be embedded, viz. $\mathbf{b}$, is mapped to a signature sequence $\mathbf{s} = \mathcal{S}(\mathbf{b}, \mathcal{K}) \in \Re^D$. The signature sequence is embedded in the coefficients $\mathbf{c}$ to obtain $\hat{\mathbf{c}} = \mathcal{E}(\mathbf{c}, \mathbf{s})$. The modified coefficients $\hat{\mathbf{c}}$ then is converted to the stego-image $\hat{I} = \mathcal{T}^{-1}(\hat{\mathbf{c}}, \mathcal{K})$. The stego-image may undergo lossy compression and is received as $\tilde{I}$ by the receiver. The coefficients $\tilde{\mathbf{c}} = \mathcal{T}(\tilde{I}, \mathcal{K})$ is extracted and $\tilde{\mathbf{s}} = \mathcal{D}(\tilde{\mathbf{c}})$ is used by the signaling scheme to estimate the bit sequence as $\tilde{\mathbf{b}} = \mathcal{S}^{-1}(\tilde{\mathbf{s}}, \mathcal{K})$. The pairs $(\mathcal{E}, \mathcal{D})$ and $(\mathcal{S}, \mathcal{S}^{-1})$ taken together, define a sophisticated signaling scheme employing a *floating signal con-*

[1]For example, in [4] the function depends on the spatial location of "salient" features

[2]$d(.)$ is an arbitrary distance measure.

[3]Computational complexity of the data embedding algorithm may not be a serious limitation - data embedding is done only once for each content.

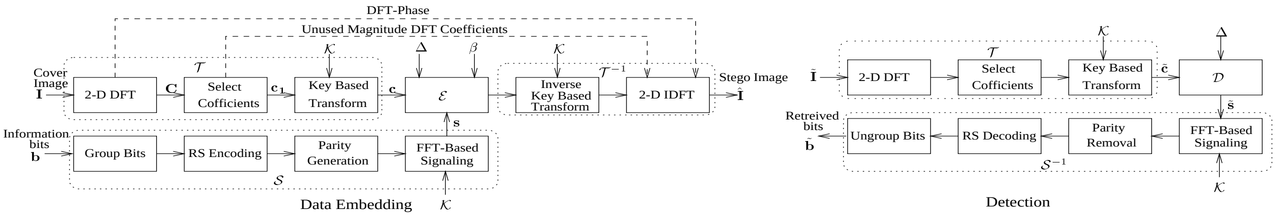

Figure 5: Block Diagram of Data Embedding and Detection

stellation [7, 8].

Let I be the original $M \times N$ image, and $I \overset{\mathcal{F}}{\Longleftrightarrow} \mathbf{I}$, where $\overset{\mathcal{F}}{\Longleftrightarrow}$ stands for 2-D DFT pairs. $\mathbf{I}$ has 4 real coefficients and $MN-4$ complex coefficients. Only half ($D_0 = \frac{MN-4}{2}$) coefficients however, have unique magnitudes. Let $\mathbf{C} \in \Re^{D_0}$ be a vector of the unique magnitudes of the complex DFT coefficients of $\mathbf{I}$. Every image can be represented as a point in the D_0-dimensional space of its magnitude DFT coefficients. The D_0 magnitude DFT coefficients serve as the carriers for the subliminal communication. However, as high frequency DFT coefficients may not be able to survive lossy compression, we shall use only a subset $\mathbf{c}_1 \in \Re^D$ of $\mathbf{C}$ for data hiding.

A truly secure data hiding scheme, should be difficult to crack even if every step of the algorithm for data hiding is public. In this case, the only 'secret' should be the key $\mathcal{K}$ The security can be vastly improved by using a key based transform before data embedding (and therefore before detecting). In the proposed scheme, we use a simple key based transform based on cyclic all-pass filters.

Let $\mathbf{h} \overset{\mathcal{F}}{\longleftrightarrow} \mathbf{H}$ where $\mathbf{h} \in \Re^N$ is cyclic all-pass (or $|H(k)| = 1\, \forall k$). As all cyclic shifts of $\mathbf{h}$ are orthogonal, they form a a basis for $\Re^N$. The phases ϕ_n, $n = 0, 1, \ldots, N-1$ of the elements of $\mathbf{H}$ can be arbitrary, we have N degrees of freedom for choice of the vector $\mathbf{h}$ with mutually orthogonal circular shifts. For real $\mathbf{h}$ we have $\frac{N}{2} - 1$ phase values which can be arbitrarily chosen. Thus a pseudo-random all pass sequence of length N can be generated from a pseudo-random (uniformly distributed between π and $-\pi$) sequence of length $\frac{N}{2} - 1$. The pseudo random sequence, would be generated from the key $\mathcal{K}$. If

$$
\begin{aligned}
\phi_k &= \begin{cases} 0 \text{ or } \pi & k = 0, k = \frac{N}{2} \\ \theta_k & k = 0 \cdots \frac{N}{2} - 1 \\ -\theta_{N-k} & k = \frac{N}{2} + 1 \cdots N-1 \end{cases} \\
H(k) &= \cos(\phi_k) + \mathrm{i} \sin(\phi_k), k = 0 \cdots N-1, \qquad (1)
\end{aligned}
$$

where $\mathrm{i} = \sqrt{-1}$, and $\theta_k, k = 1 \cdots \frac{N}{2} - 1$ are randomly distributed between π and $-\pi$, then $\mathbf{h} = \mathcal{F}^{-1}(\mathbf{H})$, is a cyclic all-pass sequence. A transform employing the $\mathbf{h}$ and all its cyclic shifts as its basis can be easily implemented by cyclic correlation. If $\mathbf{x} \in \Re^N$ is a vector of coefficients, the corresponding transform coefficients $\mathbf{X}$ can be obtained as $\mathbf{X} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{x}).\mathcal{F}(\mathbf{h}))$, and the inverse transform as $\mathbf{x} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{X}).\mathcal{F}^*(\mathbf{h}))$.

The selected magnitude DFT coefficients $\mathbf{c}_1$ undergo a key-based transform to obtain the vector $\mathbf{c}$. The first 3 blocks in the block diagram represent the operation $\mathbf{c} = \mathcal{T}(I, \mathcal{K})$.

The signaling technique $\mathcal{S}$ is used to generate the signature sequence $\mathbf{s} = \mathcal{S}(\mathbf{b})$, from the bit sequence $\mathbf{b}$ to be embedded. The signaling scheme uses an a symbol based signaling technique that permits use of large signal dimensions while still maintaining a reasonable computational complexity, due its FFT-based implementation. The signaling scheme employs orthogonal cyclic all-pass filters as signature sequences (of lengths which are powers of 2). If $\mathbf{s}_k, k = 1 \cdots n$ is a segment of $\mathbf{s}$ and $\mathbf{s}_k \in \Re^{L_k}, L_k = 2^{p_k - 1}$, maximally separable signature sequences $\mathbf{s}_k^l, l = 1 \cdots 2^{p_k}$, corresponding to p_k bits, are obtained as L_k orthogonal sequences and their negatives. The L_k orthogonal sequences are obtained as cyclic shifts of a cyclic all-pass filter (see Ref. [7] for details). Each segment $\mathbf{s}_k$ is therefore used for hiding p_k bits or a p_k-bit symbol (in our implementation $p_k = p \forall k$).

Redundant signaling for error correction uses a combination of parity introduction and Reed-Solomon coding. A sequence of d-bit symbols D_1 to D_n is encoded using Reed-Solomon encoding over $\mathcal{GF}(2^d)$, with block size of $2^d - 1$ (if $n < 2^d - 1$, the "shortened" code is implemented by zero-padding $D_1 \cdots D_n$ to length $2^d - 1$). The RS encoded sequence of d-bit symbols is then "appended" with q-parity bits to produce a p-bit symbol sequence, where $p = d + q$. From p-bit symbol sequence, the signature sequence $\mathbf{s}$ is derived as $\mathbf{s} = [\mathbf{s}_1 \cdots \mathbf{s}_n]$. Signaling with parity is especially useful for very low SNR data hiding (if ρ in Eq(2) is very small). This would result in reduced overall capacity and therefore large segment lengths L_k (or large p). If p is very high, and $q = 0$ (or $d = p$), large alphabet sizes (2^p) should be used for RS encoding / decoding, which may be prohibitively expensive. However, for the FFT-based signaling scheme introduction of parity can be done efficiently. If q parity bits are introduced, then the RS-coding/decoding needs to work only with alphabet sizes of 2^{p-q}. Though the introduction of parity is sub-optimal, it may be necessary if computational complexity is an issue.

The signature sequence $\mathbf{s}$ is embedded by the block $\mathcal{E}$ to obtain $\hat{\mathbf{c}} = \mathcal{E}(\mathbf{c}, \mathbf{s})$. The embedding block $\mathcal{E}$ utilizes a continuous periodic function to *tile* the space of the coefficients $\mathbf{c}$ with the signal constellation defined by $\mathcal{S}$. The block $\mathcal{E}$ is characterized by parameters Δ (the period of the periodic function) and β (maximum permitted distortion) of each coefficient. The periodic function used by $\mathcal{E}$ and $\mathcal{D}$, and the distribution of the distortion introduced (to the host signal) by embedding, viz, $f_E(e)$ are shown in Figure 6.

Optimal values of Δ and β are chosen depending on the variance σ_ν^2 of additive noise ν in the channel (compression / decompression, where $\tilde{\mathbf{c}} = \hat{\mathbf{c}} + \nu$), and the *permitted distortion* γ of the original image due to data hiding. It can be

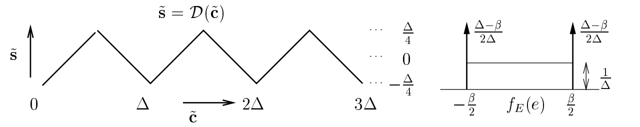

Figure 6: Periodic function employed by $\mathcal{E}$ and $\mathcal{D}$ and distribution of the distortion introduced

shown that the optimum values are obtained by maximizing the normalized correlation of $\mathbf{s}$ and $\tilde{\mathbf{s}}$, given by

$$\rho = \frac{2\sum_{i=0}^{\infty}\int_{\frac{i\Delta}{2}}^{\frac{(i+1)\Delta}{2}}(-1)^i(\frac{(2i+1)\Delta}{4} - z)f_Z(z)\mathrm{d}z}{\sqrt{2\sum_{i=0}^{\infty}\int_{\frac{i\Delta}{2}}^{\frac{(i+1)\Delta}{2}}(\frac{(2i+1)\Delta}{4} - z)^2 f_Z(z)\mathrm{d}z}} \qquad (2)$$

where, (if the additive noise in the channel is Gaussian),

$$f_Z(z) = \frac{\beta}{\Delta\sqrt{2\pi\sigma_\nu^2}}\mathrm{e}^{\frac{-z^2}{2\sigma_\nu^2}} + \frac{1}{2\Delta}\left\{\mathrm{erf}\left(\frac{z+\frac{\Delta-\beta}{2}}{\sqrt{2}\sigma_\nu}\right) - \mathrm{erf}\left(\frac{z-\frac{\Delta-\beta}{2}}{\sqrt{2}\sigma_\nu}\right)\right\}$$

where erf(.) denotes the *Gaussian error function*. The maximization is subject to the constraint $\gamma^2 = \frac{\beta^2}{12\Delta}(3\Delta - 2\beta)$. (see Ref. [8] for details).

For detection the coefficients $\tilde{\mathbf{c}}$ (which is $\hat{\mathbf{c}}$ corrupted with noise in the channel) is obtained after 2-D DFT and choosing selected magnitude DFT coefficients to undergo the Key-based transform. The signature $\tilde{\mathbf{s}} = \mathcal{D}(\tilde{\mathbf{c}})$ is extracted and the p-bit symbol sequence is decoded by the FFT-based signaling scheme. The q parity bits in each symbol are removed and RS-Decoding is performed to get the the d-bit symbol sequence from which the bit sequence $\tilde{\mathbf{b}}$ is extracted.

Figure 7 shows the performance of the data hiding scheme for several test images undergoing JPEG (at various quality factors), and SPIHT compression (at different bit-rates). From applying JPEG at at quality factors of 75, 65, 55 and 50 respectively, it was found that the resulting images on an average were compressed to 1.35, 1.25, 1.15 1.10 and 1.0 bpp respectively. So in the figure, the X-axis for both plots (JPEG and SPIHT) is an indication of the bit-rate of the compression method employed.

4. Conclusions

In this paper, we have proposed an efficient algorithm for data hiding schemes for images, for which robustness to compression is the key issue. The over-all problem of data hiding is split into two sub-problems - one of maximizing the resource (or the *permitted distortion* of the host signal), and the other of optimal utilization of the resource by means of *efficient signaling* techniques. Many solutions to the former problem have been addressed, (the latter - efficient signaling strategies - are addressed in [7, 8]), and choices for the over-all data hiding scheme are made on the basis of many trade-offs between computational complexity, and efficiency of the data hiding scheme.

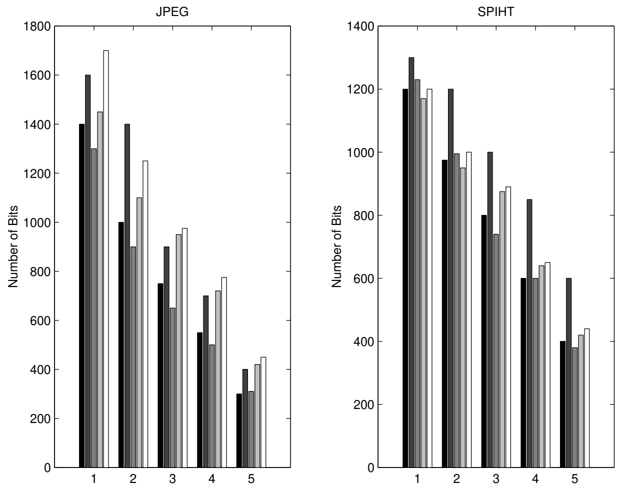

Figure 7: Plots of Achieved Data Hiding Capacities for JPEG (Left) and SPIHT (Right) Compression for 5 256×256 test images (Lena, Barbara, Boats, Goldhill and Girl). JPEG compression scenarios 1 - 5 correspond to quality factors 75, 65, 55, 50 and 40 respectively. SPIHT compression scenarios 1 - 5 correspond to 1.35, 1.25, 1.15, 1.10 and 1.0 bpp respectively.

5. References

[1] G.E. Legge, J.M. Foley, "Contrast Masking in Human Vision", *Journal of Optical Society of America*, **70**, No 12, pp 1458-1471, 1980.

[2] M. Kutter and F. A. P. Petitcolas. "A Fair Benchmark for Image Watermarking Systems",Proceedings of Electronic Imaging '99, Security and Watermarking of Multimedia Contents, San Jose, California, vol. **3657**, pp. 226-239, January 1999.

[3] D.Coltuc, P.Bolon, "Watermarking By Histogram Specification", Proceedings of SPIE, Security and Watermarking of Multimedia Contents, San Jose, CA, Vol **3657**, pp 252-263, January 1999.

[4] P.M.J.Rongen,M.J.J.J.B. Maes, C.W.A.M. van Overveld, "Digital Image Watermarking by Salient Point Modification: Practical Results," Proceedings of SPIE, Security and Watermarking of Multimedia Contents, San Jose, CA, Vol **3657**, pp 273-282, January 1999.

[5] A.V. Oppenheim, J.S. Lim,"The Importance of Phase in Signals", *Proceedings of the IEEE*, vol **69**, No 5, pp 529 - 541, May 1981.

[6] W.A. Pearlman, R.M. Gray, "Source Coding of the Discrete Fourier Transform", *IEEE Trans. on Information Theory*, vol **IT-24**, pp 683 - 692, Nov 1978.

[7] M.Ramkumar, A.N. Akansu, "FFT-Based Signaling for Multimedia Steganography", Submitted to ICASSP-2000.

[8] M.Ramkumar, A.N. Akansu, "Self-Noise Suppression Schemes for Multimedia Steganography", SPIE, International Workshop on Voice, Video and Data Communication, Multimedia Applications, Boston, September 1999.