ARMY RESEARCH LABORATORY

Automated Steganography

by George Hartwig and Lisa Marvel

ARL-TR-2642 January 2002

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5067

ARL-TR-2642 **January 2002**

Automated Steganography

George Hartwig and Lisa Marvel
Computational and Information Sciences Directorate, ARL

Abstract

We outline a novel framework for automating steganographic and watermarking processes. Then, using an implementation built on this framework, we demonstrate the effect of a particular steganographic algorithm on picture quality and show how the framework may be used to generate acceptable, robust steganographic images without human intervention. The final system can serve to objectively compare the performance of various steganographic techniques.

Acknowledgments

We would like to acknowledge all of the people who have worked on the ARL steganographic project, including Frederick Brundick, Dr. Charles Retter, and Dr. Malcom Taylor. We would also like to thank Howell Caton for reviewing this report.

INTENTIONALLY LEFT BLANK.

Table of Contents

INTENTIONALLY LEFT BLANK.

List of Figures

INTENTIONALLY LEFT BLANK.

List of Tables

INTENTIONALLY LEFT BLANK.

1. Introduction

Steganographic and watermarking techniques are used to imperceptibly convey information by using various types of multimedia data as cover for the concealed communication. The inability to detect the hidden data, either visually or by computer analysis, is paramount for the successful use of these techniques. It is equally important that the intended recipient have the ability to recover the hidden data when the stegosignal (cover + hidden data) have been exposed to typical processing.

The potential applications for data hiding are numerous. Of course, the relay of hidden messages is an apparent usage, but today's technology stimulates even more subtle practices. In-band captioning, for example, can be used to embed textual or ancillary information within a cover. It can be employed to deposit creation and revision information within the cover data for the purpose of revision tracking, preventing the need to maintain two separate media. This type of consolidation could be used to join medical images with text, such as patient data, to promote patient safety and record consistency. Data hiding can also be utilized as a technique for authentication and tamper-proofing. For example, unauthorized alterations in the cover can be detected by hiding attribute information unique to the cover, such as the checksum of certain pixel values, within the cover itself. By computing the checksum at the receiver and comparing it to the extracted checksum, the receiver could determine whether or not the cover has been corrupted.

Figure 1 shows a general purpose overview of a steganographic system. A message is embedded in a digital image by the stegosystem encoder, which uses a key or password. The resulting stegoimage is transmitted over a channel to the receiver, where it is processed by the stegosystem decoder using the same key. During transmission, the stegoimage can be monitored by unintended viewers who will notice only the transmittal of the innocuous image without discovering the existence of the hidden message.

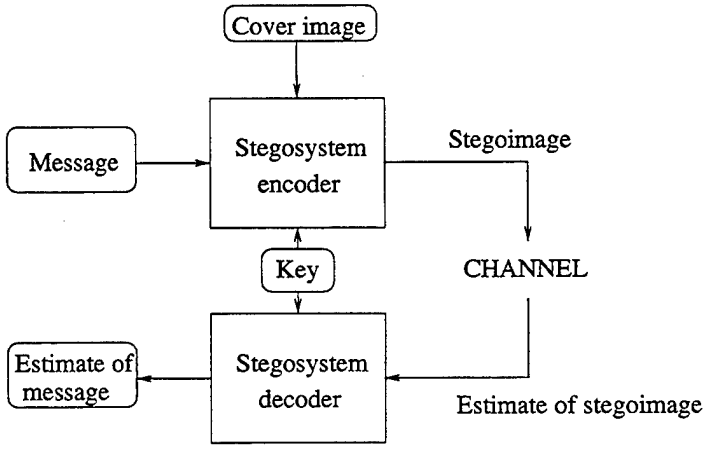


Figure 1. Overview of steganographic system.

The U.S. Army Research Laboratory (ARL) has performed research in the area of steganography and steganographic applications, such as tamper detection. For a more thorough treatment of steganography, including theoretical capacity bounds, see reference [1].

Many of the watermarking and data-hiding methods that exist today have parameters that may be adjusted to meet some criteria — robustness, payload, and/or quality. These criteria are inversely proportional to one another [2]. For instance, the greater the amount of hidden information embedded within the cover, the greater the adverse effect on the quality on the stegosignal (the cover with the message embedded within it). The same is true for robustness of the hidden information; the more robust the information is to corruption, the more the quality of the stegoimage is diminished.

Therefore, balancing the amount of hidden information, how robust the information is to removal, and the degradation of the cover become a challenging dilemma. A resolution is typically obtained iteratively via a "human in the loop" to provide quality feedback. Subsequently, an automated method of specifying all the options available to the novice user that will provide consistent performance would be desirable and would provide a method by which to objectively compare steganography and watermarking systems.

In the following section, we describe a framework for automated steganography and present the concept of a supervisor module. In section 3, the actual implementation of a stegosupervisor with a specific steganographic system is delineated. Section 4 provides some results and conclusions as well as recommendations for future work.

2. Generic Framework of Automated Steganography

In this section, we present the basic framework for automated steganography. The optimization of this multivariable problem is complex because it may be difficult to quantify such variables as robustness and quality. Furthermore, the relationship between the robustness, quality, and payload capacity of a steganographic system is nontrivial. Therefore, the general framework is based on abstract ideas.

To begin, let us define $\mathcal{S}$ as the set of steganographic methods (and all instances of operating parameters specific to each method). Then let us define $\mathcal{C}$ as a set of cover data existing in a library that contains cover data (e.g., image, audio, and video). Now $S \in \mathcal{S}$ represents a steganographic system and a single instance of its operating parameters, and $C \in \mathcal{C}$ is a single selection from the cover data library. We can also define M as the length of the message to be embedded by the stegosystem; R will represent a robustness factor indicating the resistance of the steganographic message to anticipated distortions; and Q is a quality factor indicating the imperceptibility of the message in the cover data. Without loss of generality, we can define the performance of a typical steganographic system, P, as a function of these five variables:

$$P = f\begin{pmatrix} S \\ C \\ M \\ R \\ Q \end{pmatrix}. \tag{1}$$

However, since M, R, and Q are a function of S and C,

$$M = \mathcal{M}(S,C), \tag{2}$$
$$R = \mathcal{R}(S,C), \tag{3}$$
and
$$Q = \mathcal{Q}(S,C), \tag{4}$$

P can be further quantified as

$$\begin{pmatrix} M \\ R \\ Q \end{pmatrix} = P = f \begin{pmatrix} \mathcal{M}(\mathcal{S},\mathcal{C}) \\ \mathcal{R}(\mathcal{S},\mathcal{C}) \\ \mathcal{Q}(S,C) \end{pmatrix}. \tag{5}$$

Now, to constrain the problem, let us look at a typical operational methodology. In essence, the goal of steganography is to convey some information from the sender to the receiver in a hidden way. To this end, the message and its length (M) are specified by the sender and are thus fixed as M_d. In addition, the user typically has a concept of a desired level of quality for the stegosignal, which we will call Q_d, as well as knowledge of the potential distortions stegosignal needs to overcome for proper message extraction. We denote this desired robustness as R_d.

Once the message length, robustness, and quality are fixed as M_d, R_d, and Q_d, respectively, the optimization of steganographic performance can be simplified. The goal of automated steganography is to choose $S \in \mathcal{S}$ and $C \in \mathcal{C}$ such that the relation of equation 5 produces values of $M(S,C)$, $R(S,C)$, and $Q(S,C)$ that satisfy

$$M \geq M_d, \tag{6}$$
$$R \geq R_d, \tag{7}$$
and
$$Q \geq Q_d. \tag{8}$$

Now P is limited to the manipulation of parameters for stegosystem, S, and C is limited to the selections available in a cover data library containing instances of commonly occurring cover.

Basically, the combination of equation 5 and equations 6–8 represents the search among S and C for a configuration so that M, R, and Q are lower bounded by their desired counterparts M_d, R_d, and Q_d.

Using this general framework, we can design a supervisor module to enable a notion of automated steganography, as shown in Figure 2, to produce stegosignals or to compare various steganographic techniques in an impartial manner.

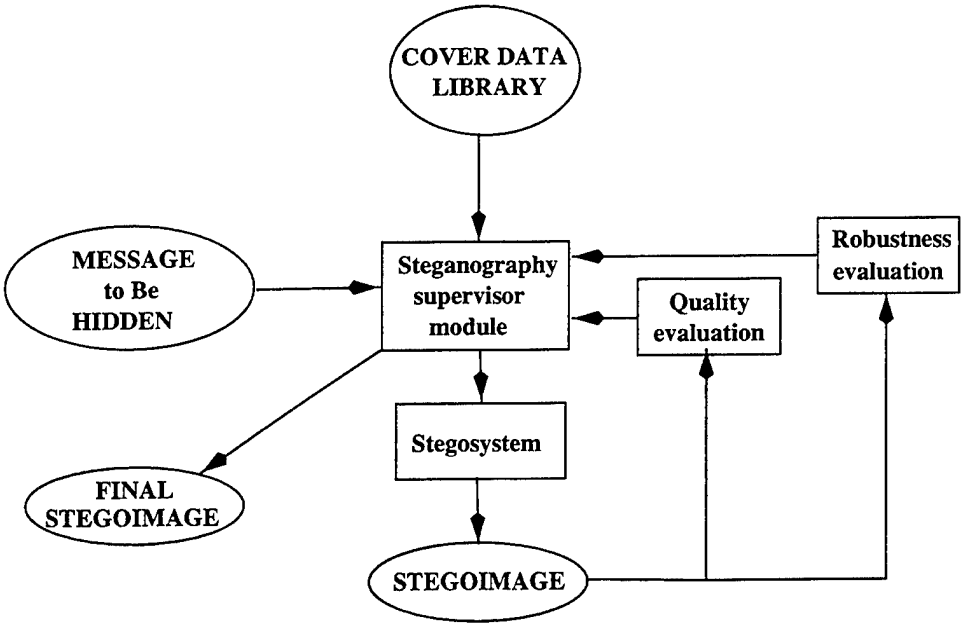


Figure 2. Automatic steganography architecture.

The supervisor module accepts the fixed length message to be encoded and, through an iterative procedure, varies the stegosystem parameters and the cover data from the cover-data library to meet a desired quality and robustness criteria. In the case of user-specified cover data, where no choice is available in cover images (the number of elements in $\mathcal{C}$ is equal to 1), only the stegosystem parameters can be adjusted.

3. Implementation

When applying steganographic techniques to hide information, we, the users, must specify the parameters that govern the performance of the steganographic system. We begin by selecting the cover data that will contain the hidden data. Once the cover type is defined, we must select the stegosystem used to embed the message. (Stegosystems are readily available for speech, image, and video cover data.) We make the assumption that the message is a stream of binary symbols that occur independently with equal probability (as is the case when the data has been efficiently compressed and/or encrypted). The payload criteria, M_d, is established by the length of this message. Next, the desired quality is designated, indicating an acceptable level of cover data distortion the user will permit. The intentional and unintentional distortions the stegosignal may incur must be determined to define the desired robustness criteria. Finally, the message is embedded, and the user must make a decision as to whether the distortions introduced by the steganographic process are too severe to be acceptable. Often this consists of a human making a quick judgement based on a comparison between the original image and the stegoimage. If the stegoimage is not acceptable, then the user must go back and begin the process again, adjusting the parameters accordingly. The purpose of our automated steganographic system is to eliminate the need

for this human feedback in the stegoprocess. In this section, we discuss an implementation of the automatic steganographic process.

For the purposes of this study, we have elected to use spread spectrum image steganography (SSIS) [3] as the stegosystem to perform the data hiding in imagery. This system will be described in more detail in a subsequent section.

As a substitute for the human in the loop that typically performs the image quality evaluation, we elected to use an accepted method that considers the human visual system. Perhaps the definitive indicator of relative image quality is the mean opinion score (MOS) [4], a value attained by averaging the numerical scores assigned by a panel of humans who view the image in question under identical conditions. However, such trials are nontrivial, and the need remains for a simple metric that is easy to calculate, simple to interpret, simple to use, and indicates how distorted an image looks to a human. We have found such a metric for monochromatic images in the CIPIC* picture quality scale (PQS) described in Miyahara et al. [5] The version 1.0 software that was used in our work may be obtained from the internet. The PQS provides a numerical output ranging from 1 to 5, with 5 assigned to an image that has no perceptible differences when compared to the original, and 1 assigned to an altered image that exhibits very annoying artifacts.† The choice of the PQS metric limits the images used in this proof of concept project to gray scale images (one byte per pixel) of size 256×256.

Unfortunately, no similar metric exists for determining robustness. So to establish the robustness evaluation process, we emulate the expected environment that the stegoimage will encounter during its life cycle. As dictated by the demonstration scenario, it is anticipated that the stegoimage will be compressed. For this aspect, standard JPEG version 6 compression is used. Also, random bit errors may be inserted in the image to simulate the bit error rate (BER) typical of various transmission media. The level of compression or noise combinations thereof can be specified in the evaluation process. Correct recovery of the message data after the distortion emulation will constitute a successful robustness evaluation with this scheme.

To tie these elements together, a stegosupervisor module was created to implement procedures for balancing the message to be transmitted, resistance to removal or distortion, and degradation of the cover image caused by adding the stegomessage. The primary factors available to the supervisor for manipulation include the selection of cover image, the power of the stegosignal imbedded in the cover image, and, since we are using SSIS, the level of the protection provided by the error control code (ECC) scheme.

3.1 Stegosupervisor

While, at first, this might seem like an optimization program for a linear programming solution, this is not the case. The interactions between the variables Q, M, and R are difficult to quantify. In particular, the Q indicator is a combination of nonlinear functions

*University of California at Davis Center for Image Processing and Integrated Computing.

†Numbers outside of this range may result from the calculations, but their meaning is unclear.

attempting to approximate the response of the human visual system, and R is a complex process of compression and channel models. So the stegosupervisor must guide an iterative scheme to produce an acceptable product, as shown in Figure 3.

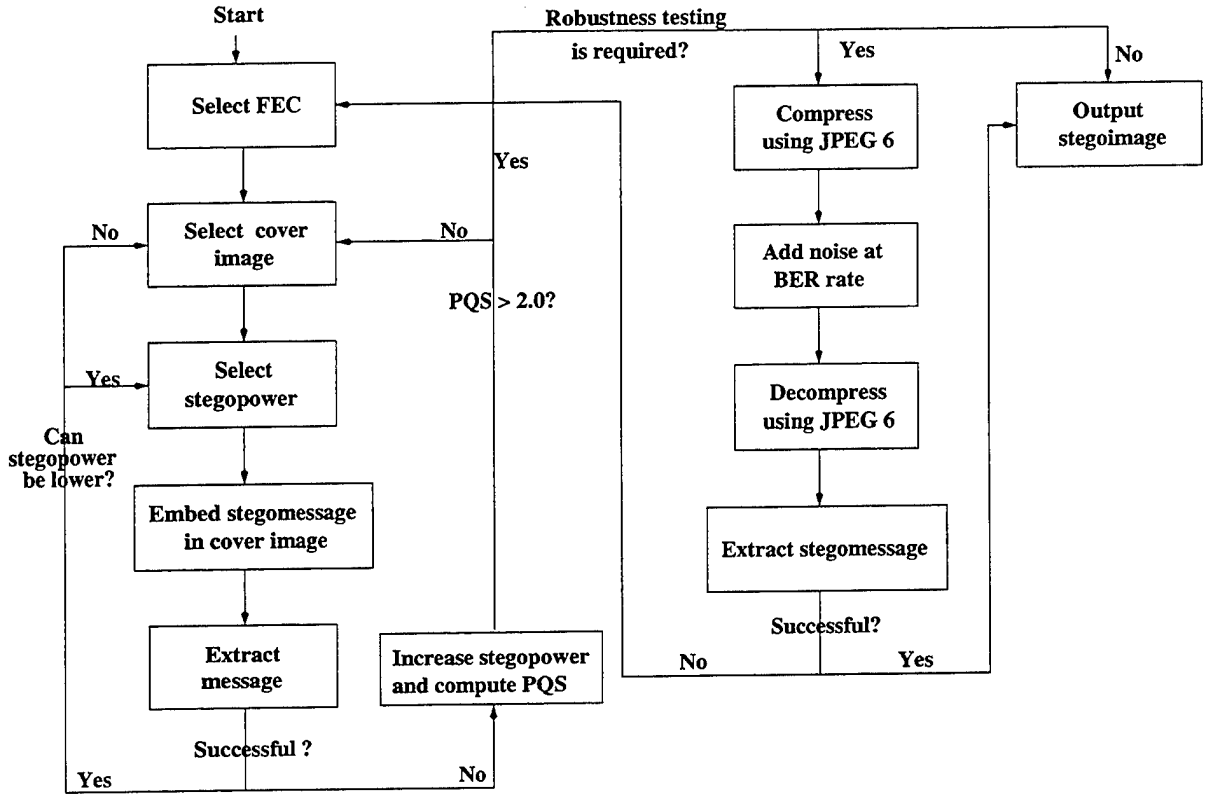


Figure 3. The automatic steganographic process.

Once a message is presented to the stegosupervisor, a default SSIS ECC code is selected, and the message is encoded and interleaved. Several ECC codes are available for use with the SSIS system, as shown in Table 1 [1]. This table lists four different Reed-Solomon (RS) codes catagorized by the number of output bits and input bits (i.e., (31,8)) for the RS codes, along with their binary expansion [6, 7]. For each of these codes, the average percentage of bits the code is able to correct is indicated as the BER correction capability. The payload in bits per pixel (bpp) is also shown, as well as the information rate in percent of total bits.

Although the current SSIS ECCs are the binary expansions of RS codes used here, any ECC can be used within SSIS. In fact, convolutional codes have also been implemented within SSIS.

With an ECC selected and the message length known, it is possible to select a cover image that is large enough to contain the ECC corrected message. A library of cover images should contain images that are obtained locally either via a scanner or digital camera. These images should be used without replacement. The same image appearing multiple times with different noise signatures would certainly raise suspicions. If images widely available over the web are used, the comparison of stegoimages with the originals would again raise the suspicions of someone attempting to detect the covert stegosignal.

Table 1. Binary expansion of RS codes.

Original RS Code	Binary Code	BER Correcting Capability	Payload (bpp)	Information Rate (bits)
(31,8)	(155,40)	0.12	0.2581	0.0323
(63,6)	(378,36)	0.21	0.0952	0.0119
(127,5)	(889,35)	0.27	0.0393	0.0049
(255,4)	(2040,32)	0.34	0.0156	0.0019

With the type ECC and cover image selected, the only remaining SSIS parameter selection is the a power of the steganographic signal (stegopower) in which to embed the message. For our system, we have selected to start the iterative process with a relatively high stegopower of 80. A stegoimage is then constructed, and immediately thereafter, an attempt is made to extract the message. If successful (i.e., the original and extracted message compare perfectly), the stegopower is lower a fixed amount and the process is repeated. Eventually, the message cannot be successfully extracted. At this point, the stegopower reverts to the last successful value, and a new stegoimage is constructed. Then a PQS value is computed. If the value is less than our designated threshold (i.e., 4), a new cover image is selected, and we start over; otherwise, we proceed to the robustness testing.

As mentioned in the previous section, robustness testing is very straightforward, and additional noise and/or compression may be applied to the stegoimage under consideration. After the noise and/or compression has been applied, an attempt is made to extract the original message. If successful, the stegoimage is output; otherwise, a new cover image is selected, and the process starts anew.

Noise is added by generating a uniform sequence of random numbers between 0 and 1. A bit stream of ones and zeros are then computed by testing the random numbers against the specified BER. If the number is less than the BER, the bit is set to 1; otherwise, it is set to 0. The data stream and the error stream are then combined using the "exclusive-or" operator to produce an output where approximately the specified number of bits are in error.*

To apply Joint Photographic Experts Group (JPEG) [8] compression to the stegoimage, the version 6 software obtained from the Independent JPEG Group's (IJG) web site, <ftp://ftp.uu.net/graphics/jpeg/>, was used. If low quality factors were used, the loss of high frequency components would destroy the stegomessage beyond recovery; therefore, quality values must be greater than 85.

In some cases, it may not be possible to obtain a successful conclusion, resulting in a stegoimage that doesn't meet all desired attributes. This may be particularly true in the case where the cover library contains a single element. In such situations, the stegosupervisor will produce a "best effort" stegoimage and report the result.

*With most image formats, the introduction of errors into the header will produce catastrophic errors when handling the image. We therefore limit errors to the data portion of the image.

3.2 Stegosystem—SSIS

For this implementation, we used a steganographic system developed at ARL called SSIS [3, 9, 10, 12, 11]. SSIS is a data-hiding scheme that approaches steganography as a communication problem, considering the cover image as the channel through which the hidden information is sent. It uses error-control coding to encode the message information and then employs a noise modulation technique to construct a signal that appears as white Gaussian noise. This noise signal is then added to the original image to construct an image containing the hidden message. The data is recovered using channel (image) estimation techniques. By selecting the appropriate embedded signal power and ECC, the hidden information is recoverable in cases where the image has been compressed via low levels of image compression or exposed to additive channel noise.

Figure 4 represents the processing of the SSIS encoder. Within the system, the message is optionally encrypted with key 1 and then encoded via a low-rate error-correcting code, producing the encoded message, m. The sender enters key 2 into a wideband pseudorandom noise generator, producing a real-valued noise sequence, n. Subsequently, the modulation scheme is used to combine the message with the noise sequence, thereby composing the embedded signal, s, which is then input into an interleaver using key 3. This signal is now added with the cover image, f, to produce the stegoimage, g, which is appropriately quantized and clipped to preserve the typical dynamic range of the cover image. The stegoimage is then transmitted in some manner to the recipient.

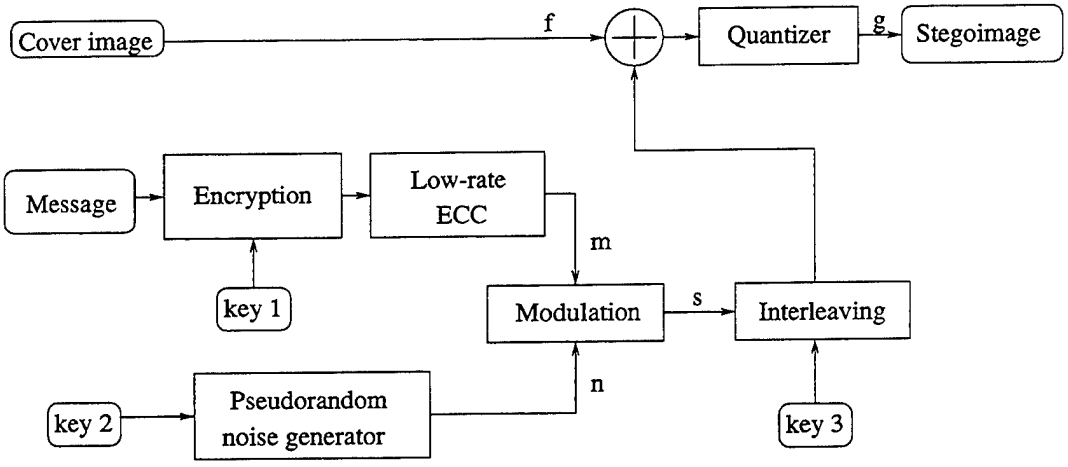


Figure 4. SSIS encoder.

Figure 5 depicts the major components of the SSIS decoder. At the receiver end, the stegoimage is received by the recipient, who maintains the same keys as the sender, and uses the stegosystem decoder to extract the hidden information. The decoder uses image restoration techniques to produce an estimate of the original cover image, $\hat{f}$, from the received stegoimage, $\hat{g}$. The difference between $\hat{g}$ and $\hat{f}$ is fed into a keyed deinterleaver to construct an estimate of the embedded signal, $\hat{s}$. With key 2, the noise sequence, n, is regenerated, the encoded message is then demodulated, and an estimate of the encoded message, $\hat{m}$, is

constructed. The estimate of the message is then decoded via the low-rate error-control decoder, optionally decrypted using key 1 and revealed to the recipient.

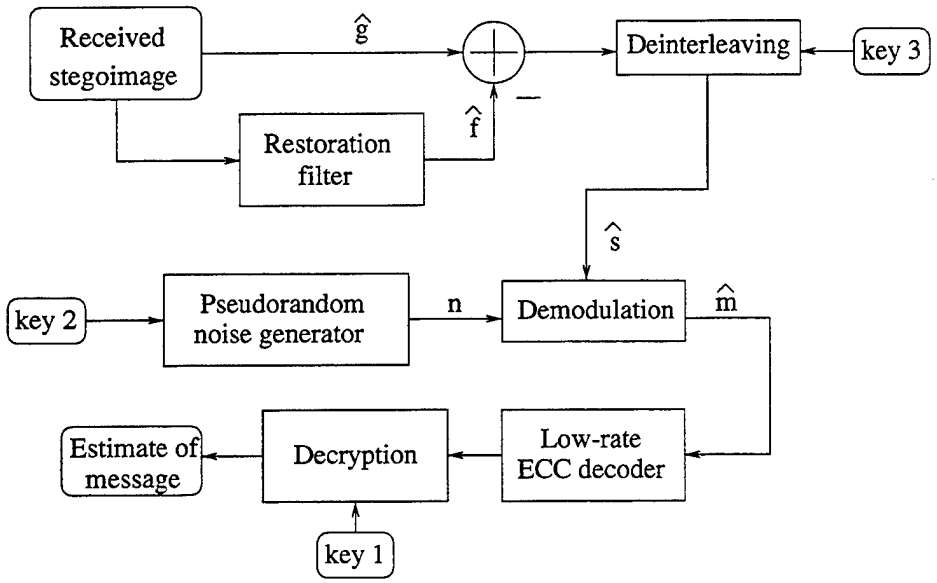


Figure 5. SSIS decoder.

For an elaborate treatment of all subprocesses of the SSIS algorithm, see reference [11].

The parameters of SSIS that correspond to vector S in equation 1 include two elements that are the variance of the additive signal power, denoted as σ_s^2, and the particular ECC that is used to encode the message. These two parameters will be varied by the automated steganography supervisor module.

3.3 Picture Quality Measurement

One of the problems encountered in using the blind steganographic process is balancing the amount of information that can be hidden in an image and the apparent quality of the resulting image. Since the hiding process embeds the data to be hidden in white Gaussian noise and then adds to the cover image, it would be advantageous to have a metric that could be used to predict when the level of added noise becomes perceptible to humans. Since images are created for humans, quality criteria such as the mean squared error (MSE) and peak signal to noise ratio (PSNR) that rely strictly on statistical information fail to correlate with the way a human would judge the image [13]. The traits of the human visual system (HVS) most often mentioned with respect to image quality include: luminance sensitivity—this is primarily due to the light-adaptive processes of the retina; frequency sensitivity—the design of the HVS optics and neural processing impose limits on how data is perceived and processed; and signal content sensitivity—the fact that sensitivity to variations is a function of signal content is due to the manner that the brain processes visual information [14, 15]. The MSE and the PSNR are defined by equations 9 and 10, respectively. In these equations, x_i is the value of the a pixel from the original image and $\hat{x}_i$ is the value from the corresponding

reconstructed pixel.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2, \tag{9}$$

$$PSNR = 10\log_{10}\frac{255^2}{MSE}. \tag{10}$$

An accepted method of assessing image quality that does consider the HVS is to assemble a panel of subject matter experts and have them rate the images. Perhaps the definitive indicator of relative image quality is known as the MOS [16]. * The MOS impairment scale is shown in Table 2, and the experimental conditions are shown in Table 3.

Table 2. ITU-R BT.500 MOS impairment scale.

Scale	Impairment
5	Imperceptible
4	Perceptible, but not annoying
3	Slightly annoying
2	Annoying
1	Very annoying

Table 3. ITU-R BT.500 experimental image rating conditions.

Condition	Value
Ratio of viewing distance to picture height	4
Room illumination	None
Peak luminance on the screen	42.5 (cd/m^2)
Lowest luminance on the screen	0.23 (cd/m^2)
Time of observation	Unlimited
Number of observers	9 (expert observers)

For examples of the experimental techniques used to determine subjective image quality see the papers by Van Dijk et al. [17] and Perlmutter et al. [18]. Such experiments are nontrivial, and the need remains for a simple metric that is easy to calculate, simple to interpret, and simple to use and that indicates how distorted an image looks to a human.

Such a metric has been found for monochromic images, in the CIPIC PQS described in Miyahara et al. [5]. The brief description of the PQS presented here closely follows the much

*Initially issued in 1974 by The International Radio Consultive Committee (CCIR) as Recommendation 500, this document covers methods of subjective testing of electronically displayed images.

more detailed description provided in the referenced paper. For the version 1.0 software that was used in our work, see reference [5].

The PQS is a metric that attempts to measure the relative distortion between an original image and one that has been manipulated or distorted in some way while taking into account how the HVS would perceive that distortion. To achieve this goal, the PQS is based on five factors that attempt to characterize the distortion (see table 4). Now we discuss each of the distortion factors in more detail. The cover image used in the following discussions is a monochromic 256 $\times$ 256 version of the well known "Barbara." The steganographic process has added white Gaussian noise, with a variance of 60 to the cover image.

Table 4. PQS distortion factors.

Factor	Meaning
F 1	The ITU-T Rec.J.61 television noise weighting function [16].
F 2	A factor that attempts to account for HVS characteristics including contrast sensitivity, frequency sensitivity, and perceptual thresholds.
F 3	This factor characterizes linear errors such as those occurring between blocks in discrete cosine transformation compression methods.
F 4	This factor represents correlated errors since they are much more noticeable than random errors.
F 5	This factor accounts for errors that occur near high-contrast boundaries. These errors are attenuated by the HVS.

Distortion factor F_1 is based on the ITU-T Rec.J.61* [16] television noise weighting standard. First, we calculate the error map, $e(m,n)$,

$$e_i(m,n) = i(m,n) - \hat{i}(m,n), \tag{11}$$

where $i(m,n)$ are the pixel values from the original image and $\hat{i}(m,n)$ are the values from the distorted image. Then,

$$f_1(m,n) = \left[e_i(m,n) * w_{tv}(m,n)\right]^2, \tag{12}$$

where the $*$ represents the convolution operation and w_{tv} is the spatial weighting value corresponding to the frequency weighting defined by ITU-Rec.J.61 to be

$$W_{tv}(f) = \frac{1}{1+(f/f_c)^2}, \; f = \sqrt{u^2+v^2}, \tag{13}$$

with a 3-dB cutoff frequency $f_c = 5.56$ *cpd* at a viewing distance of four times the picture height. Finally, the single distortion factor F_1 is computed as

$$F_1 = \frac{\sum_{m,n} f_1(m,n)}{\sum_{m,n} i^2(m,n)}. \tag{14}$$

*Previously known as CCIR 567-1.

A graphical interpretation of $f_1(m,n)$ obtained when the PQS value is computed for the test image described earlier is being compared to our original image is shown in Figure 6. The factor values have been scaled to enhance visibility but the structure of the white Gaussian noise (WGN) added as part of the stegoprocess is clearly visible.

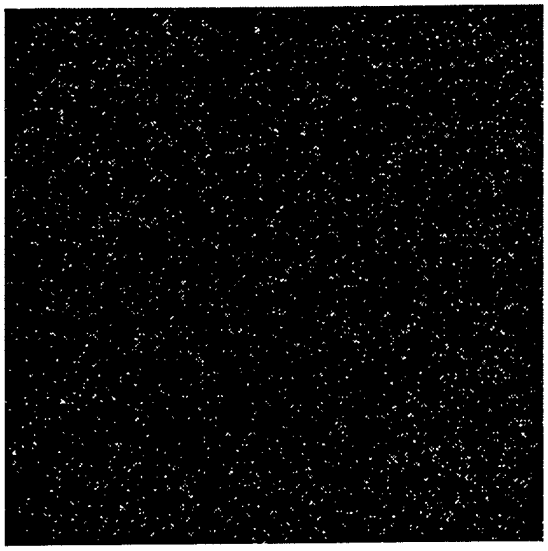


Figure 6. PQS factor $f_1(m,n)$.

Distortion factor F_2 is a more complete single channel model of visual perception in that both a correction incorporating Weber's law and a correction for the frequency sensitivity of the HVS are considered. To calculate the PQS first, both $\hat{i}$ and i are transformed using the following approximation to the Weber-Fecher law for contrast sensitivity:

$$x(m,n) = k * i(m,n)^{\frac{1}{2.2}}, \tag{15}$$

and

$$\hat{x}(m,n) = k * \hat{i}(m,n)^{\frac{1}{2.2}}, \tag{16}$$

where k is a scaling constant and the exponent is a middle number from a range of common values [19]. The contrast adjusted error image is given by

$$e(m,n) = x(m,n) - \hat{x}(m,n). \tag{17}$$

The weighting of errors to account for sensitivity to spatial frequencies is then calculated based on a measure contrast sensitivity function as

$$S(\omega) = 1.5e^{-\sigma^2\omega^2/2} - e^{-2\sigma^2\omega^2}, \tag{18}$$

where

$$\sigma = 2,\ \omega = \frac{2\pi f}{60},\ f = \sqrt{u^2+v^2}, \tag{19}$$

u represents the horizontal frequencies and v represents the vertical frequencies in cycles per degree (*cpd*).

At higher frequencies, the measured response depends on the angle of approach and is given by

$$S_a(u,v) = S(\omega)O(\omega,\theta), \tag{20}$$

where

$$O(\omega,\theta) = \frac{1 + e^{\beta(w-w_0)}\cos^4 2\theta}{1 + e^{\beta(w-w_0)}} \tag{21}$$

and

$$\theta = \tan^{-1}(u/v),\ \beta = 8,\ and\ f_0 = 11.13\ cpd. \tag{22}$$

So generating the frequency weighted error $e_w(m,n)$ is simply a matter of filtering the contrast adjusted error $e(m,n)$ with $S_a(u,v)$.

Using these contrast and frequency weight error maps, we then calculate the remaining distortion factors.

Frequency weighted errors below a given threshold T are discarded. Therefore,

$$f_2(m,n) = I_T(m,n)\left[e_w(m,n) * s_a(m,n)^2\right], \tag{23}$$

and

$$F_2 = \frac{\sum_{m,n} f_2(m,n)}{\sum_{m,n} i^2(m,n)}. \tag{24}$$

$I_T(m,n)$ is an indicator function for perceptibility. A graphical interpretation of $f_2(m,n)$ obtained when the PQS value is computed for the example image is shown in Figure 7. Here we see that much of the noise shown in Figure 6 has been discarded as imperceptible to the human observer.

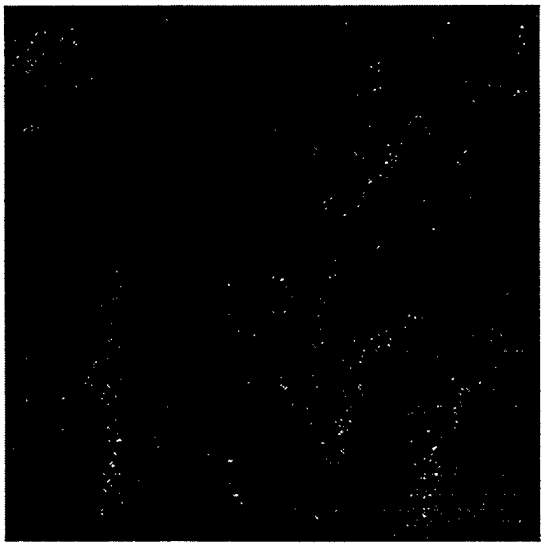


Figure 7. PQS factor $f_2(m,n)$.

The distortion factors F_3, F_4, and F_5 are designed to evaluate the effect of structured and correlated errors. F_3 detects the end of block discontinuities found at the boundaries of transform blocks such as those visible in images compressed with JPEG using a low-quality factor.

$$f_{3h}(m,n) = I_h(m,n)\Delta_h^2(m,n), \text{ and } f_{3v}(m,n) = I_v(m,n)\Delta_v^2(m,n), \tag{25}$$

where

$$\Delta_h(m,n) = e_w(m,n) - e_w(m,n+1) \text{ and } \Delta_v(m,n) = e_w(m,n) - e_w(m+1,n). \tag{26}$$

I_h and I_v are indicator functions that select those differences that span the horizontal and vertical boundaries, respectively.

$$F_{3h} = \frac{1}{N_h}\sum_{(m,n)} f_{3h}(m,n) \text{ and } F_{3v} = \frac{1}{N_v}\sum_{(m,n)} f_{3v}(m,n), \tag{27}$$

where N_h and N_v are the number of pixels selected by the horizontal and vertical indicator functions. Finally,

$$F_3 = \sqrt{F_{3h}^2 + F3_{3v}^2}\ . \tag{28}$$

A graphical interpretation of $f_3(m,n)$ computed for the example image is shown in Figure 8.

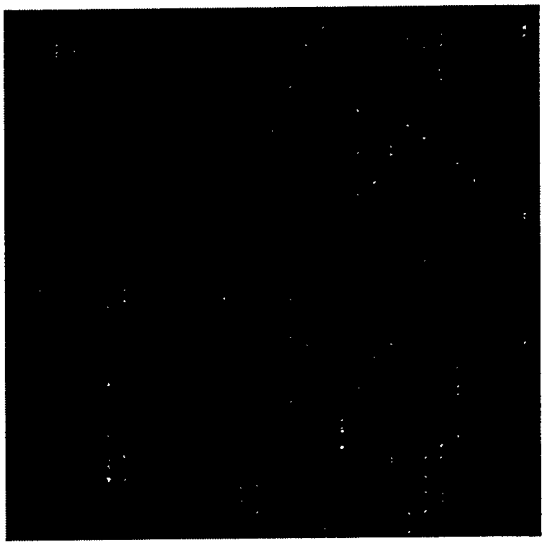


Figure 8. PQS factor $f_3(m,n)$.

Since the HVS notices errors that exhibit spatial correlation much more than random noise, we calculate local spatial correlations and then sum them over the entire image.

$$f_4(m,n) = \sum_{(k,l)\in W} \mid r(m,n,k,l) \mid^{0.25}, \tag{29}$$

where the local correlation is

$$r(m,n,k,l) = \frac{1}{n-1}\left[\sum e_w(i,j)e_w(i+k,j+l) - \frac{1}{n}\sum e_w(i,j)\sum e_w(i+k,j+l)\right] \quad (30)$$

and where the sums are computed over a 5×5 window centered at (m,n). The set of correlation lags, W, includes all of the unique lags $\mid k \mid, \mid l \mid \leq 2$ except those made redundant because of symmetry and the $0,0$ one which is simply the variance of the error values in the window. The exponent serves to de-emphasizes the relative magnitude of the errors. Finally, the factor is calculated as

$$F_4 = \frac{1}{MN}\sum_{m,n} f_4(m,n). \quad (31)$$

A graphical interpretation of $f_4(m,n)$ is shown in Figure 9.

Figure 9. PQS factor $f_4(m,n)$.

Distortion factor F_5 is designed to account for the visual masking of errors by the underlying image signal. In particular, it measures distortions that occur in the vicinity of high-contrast transitions and is measured in both the horizontal and vertical directions. First, masking factors are calculated as

$$S_h(m,n) = e^{-0.04V_h(m,n)} \text{ and } S_v(m,n) = e^{-0.04V_v(m,n)}, \quad (32)$$

where V_h is computed from the original image data,

$$V_h = \frac{\mid i(m,n-1) - i(m,n+1) \mid}{2}, \text{ and } V_v = \frac{\mid i(m-1,n) - i(m+1,n) \mid}{2}. \quad (33)$$

The per pixel error as masked by the underlying signal is

$$f_5 = I_M(m,n) \mid e_w(m,n) \mid (S_h(m,n) + S_v(m,n)), \quad (34)$$

where $I_M(m,n)$ is a selector function * used to select the pixels that are close to the intensity transitions. Finally, for the image, we calculate

$$F_5 = \frac{1}{N_K} \sum_{m,n} f_5(m,n), \tag{35}$$

where N_K is number of pixels selected by the function $I_M(m,n)$. A graphical interpretation of $f_5(m,n)$ for the example image is shown in Figure 10.

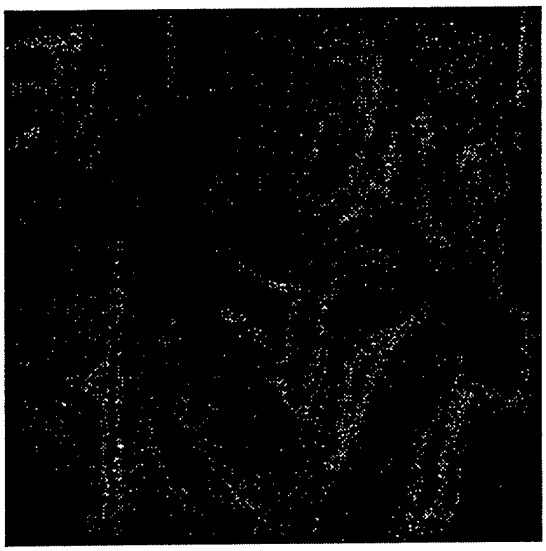


Figure 10. PQS factor $f_5(m,n)$.

The PQS is a linear combination of the five distortion factors,

$$PQS = w_0 + \sum_{j=1}^{6} w_j F_j, \tag{36}$$

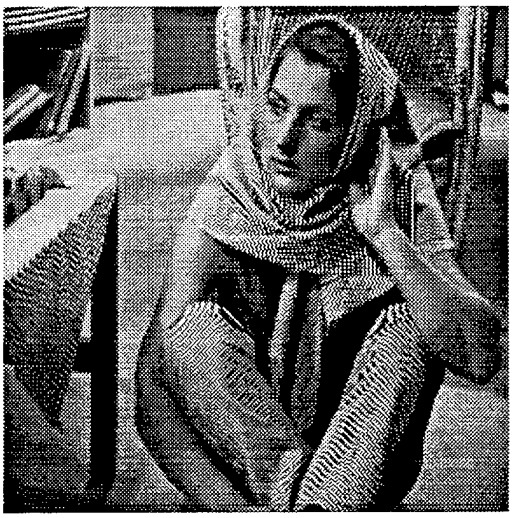
where the w_j are the partial regression coefficients obtained from a multiple regression analysis to fit equation 36 to experimental results obtained using the mean opinion scores of observers on a set of test images. The expected ranges, R, of the weighted factors and the experimentally determined w_j are shown in Table 5. If the absolute value of factor F_i, when multiplied by its corresponding weight, w_j, exceeds the range shown in the table, then the results are less reliable since the computed values fall outside of the values computed with the test set.

Figure 11 shows some examples of PQS measurements. In this figure, the top left-hand image is the original. The other three are stegoimages, with increasing amounts of noise as introduced by the SSIS process.

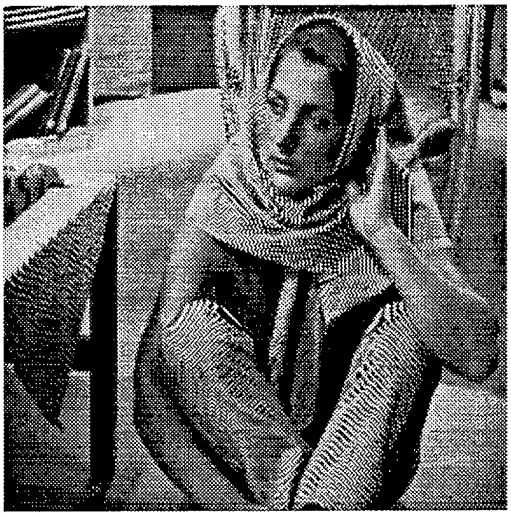
*The selector function in this case computes the 3×3 Kirsch edge response and selects those values that exceed a threshold of 400.

Table 5. Expected factor ranges and weights.

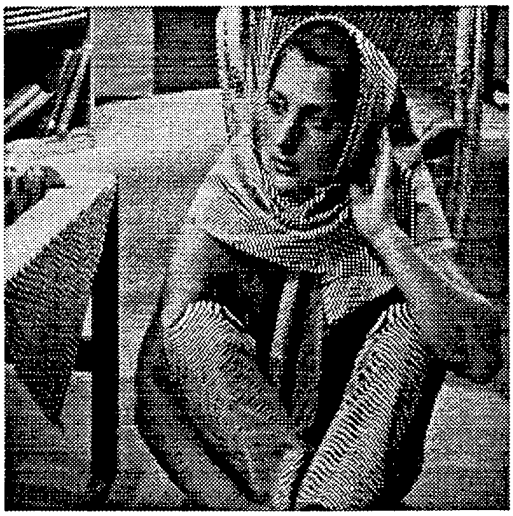
j	0	1	2	3	4	5
Range		.1	.1	1.5	3.5	3.5
w_j	5.797	0.035	.044	.01	-.132	-.135



Barbara - Original Image



Variance = 5.0 ; PQS ~ 4



Variance = 20.0 ; PQS ~ 3

Variance = 60.0 ; PQS ~ 2

Figure 11. Sample PQS measurements.

3.4 Robustness Evaluator—Channel Models

Currently, robustness is evaluated by applying expected distortions and testing if the embedded message can be extracted. It should be remembered that our goal here is to embed a message in a cover image so that it is undetectable. Although stegoimages produced by the SSIS process are tolerant of minor manipulations such as compression with high-quality factors or moderate noise in the transmission process, stegoimages produced by this system are not tolerant of large manipulations. High levels of compression by lossy algorithms, large amounts of cropping, or certain kinds of filtering will result in the inability to correctly recover the embedded message.

In this system, we have included an implementation of JPEG V6 and, if desired, the stegoimage will be compressed using a specified quality factor. The JPEG file is then decompressed and processed by the SSIS decoder. If the message is successfully recovered, then the stegoimage is deemed acceptable by meeting the desired robustness. Similarly, a noisy channel can be approximated by the use of a binary symmetric channel model. In this model, each bit is changed from a 0 to a 1, or from a 1 to a 0, according to a probability of error ϵ. Conversely, the probability that the bit will be unchanged is $1 - \epsilon$. Epsilon is commonly known as the BER of the binary symmetric channel. BERs typically range from around 10^{-5} for fiber optic cables to 10^{-3} for radio channels. Errors according to the specified BER can be added to the stegoimage and the distorted image tested, as in the compression case. The final result is a "pass" or "fail." The image is either robust or not at the specified distortion rated with the given stegoparameters.

3.5 Performance

To evaluate the performance of the automated steganography algorithm, we implemented a stegosupervisor program to explore the operation of the algorithm using a limited suite of images as the cover data library. This library, composed of six gray scale, 256×256 images, is shown in Figure 12.

In Figure 13, we show the effect on PQS obtained by increasing the SSIS parameter, σ_s^2, for six cover images within the demonstration library.

In the following, we present a brief tour through the operation of the stegosupervisor. Invoked with the following command line

```
ss -m msg_file -i ../IMAGES -s stg_out.pgm -q 90 -b 4
```

where "msg_file" contains the message to be hidden;

```
"Ask not what your country can do for you, but
 ask what you can do for your country." "The rain
 in Spain falls mainly in the plain."
```

"../IMAGES" is the path to the directory that contains the potential cover images, "stg_out.pgm" is the name of the final stegoimage, and "-q 90 -b 4" specify a quality factor for JPEG compression and a BER of 10^{-4}, respectively, to be used for robustness testing.

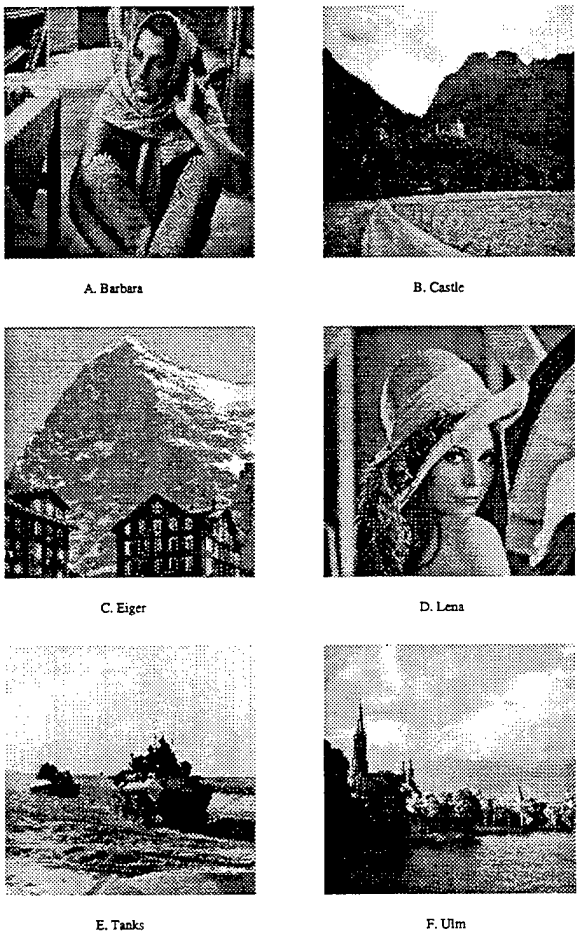


Figure 12. Library images.

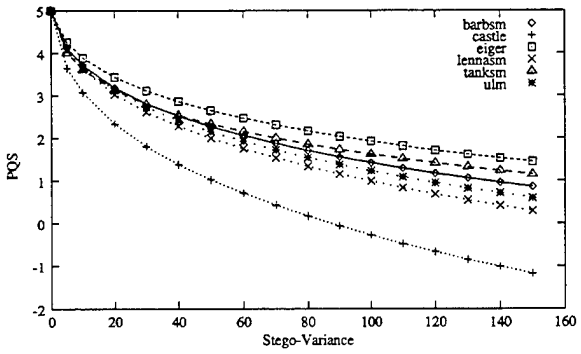


Figure 13. Sample PQS measurements.

The input message is found to be 136 bytes long, so the (889,35) ECC can be used, resulting in an encoded message length of 27636 bits. With this determined, the cover image library is searched, and available images are catalogued.

```
barbsm.pgm
rows =256 cols = 256
   size = 65536 bytes
castle.pgm
rows =256 cols = 256
   size = 65536 bytes
eiger.pgm
rows =256 cols = 256
   size = 65536 bytes
lennasm.pgm
rows =256 cols = 256
   size = 65536 bytes
tanksm.pgm
rows =256 cols = 256
   size = 65536 bytes
ulm.pgm
rows =256 cols = 256
   size = 65536 bytes
```

This task is trivial at this point since our picture quality metric restricts us to 256×256 images. Therefore, the stegosupervisor selects the first image encountered (Figure 14A).

At this point, the stegosupervisor program applies the error control code, interleaves the resulting code words, and embeds the stegomessage into the cover image. For this initial try, a stegopower of 80 is used, and the minimum PQS value we will accept is 2.

```
encode255 GI.255.4.1.n10 < tf1NIaObo > tf2OIaObo
intrleav 65536 < tf2OIaObo > tf3PIaObo
stegoder -e -key 1024 ../IMAGES/barbsm.pgm tf3PIaObo tf4QIaObo 0 80
```

We then extract the image and, if successful, the stegopower is lowered and the process repeated until the message cannot be retrieved.

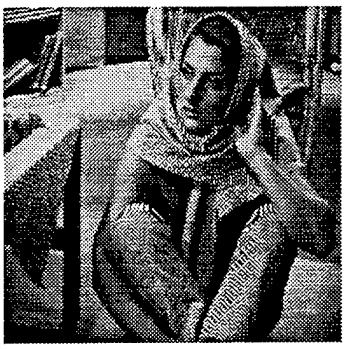
```
stegoder -e -key 1024 ../IMAGES/barbsm.pgm tf3C.aquc tf4D.aquc 0 80

stegoder -d -key 1024 tf4D.aquc msg_test 0 80

intrleav 65536 d < msg_test > di_msg_test
factor = 5 period = 65535
decode127 ../BIN/HI.127.5.41.n6 500 < di_msg_test > di_fec_mt
asc2bin < di_fec_mt | ../BIN/no_z > recovered_msg
 comparing embedded message with recovered message = 0

stegoder -e -key 1024 ../IMAGES/barbsm.pgm tf3C.aquc tf4D.aquc 0 75
stegoder -d -key 1024 tf4D.aquc msg_test 0 75
 comparing embedded message with recovered message = 0

stegoder -e -key 1024 ../IMAGES/barbsm.pgm tf3C.aquc tf4D.aquc 0 70
```



A. Original Barbara Image

B. Calculated Stegoimage

C. Decompressed Stegoimage

Figure 14. Demonstration images.

```
stegoder -d -key 1024 tf4D.aquc msg_test 0 70
 comparing embedded message with recovered message = 0

stegoder -e -key 1024 ../IMAGES/barbsm.pgm tf3C.aquc tf4D.aquc 0 65
stegoder -d -key 1024 tf4D.aquc msg_test 0 65
 comparing embedded message with recovered message = 0

stegoder -e -key 1024 ../IMAGES/barbsm.pgm tf3C.aquc tf4D.aquc 0 60
stegoder -d -key 1024 tf4D.aquc msg_test 0 60
 comparing embedded message with recovered message = 0

stegoder -e -key 1024 ../IMAGES/barbsm.pgm tf3C.aquc tf4D.aquc 0 55
stegoder -d -key 1024 tf4D.aquc msg_test 0 55
 comparing embedded message with recovered message = 0

stegoder -e -key 1024 ../IMAGES/barbsm.pgm tf3C.aquc tf4D.aquc 0 50
stegoder -d -key 1024 tf4D.aquc msg_test 0 50
 comparing embedded message with recovered message = 0

stegoder -e -key 1024 ../IMAGES/barbsm.pgm tf3C.aquc tf4D.aquc 0 45
stegoder -d -key 1024 tf4D.aquc msg_test 0 45
 comparing embedded message with recovered message = 0

stegoder -e -key 1024 ../IMAGES/barbsm.pgm tf3C.aquc tf4D.aquc 0 40
stegoder -d -key 1024 tf4D.aquc msg_test 0 40
```

```
recovered_msg msg_file differ: char 31, line 1
 comparing embedded message with recovered message = 256

"Ask not what your country can%jor you, but
ask what you can do for your country." "The rain
in Spain falls mainly in the plain."
```

Then, the stegopower is increased to the last successful value, and a new stegoimage constructed. A PQS value is computed for that image.

```
stegoder -e -key 1024 ../IMAGES/barbsm.pgm tf3C.aquc tf4D.aquc 0 45

pqs ../IMAGES/barbsm.pgm tf4D.aquc > pqs.out

Found an image we are happy with at least so far. 'barbsm.pgm'
 PQS = 2.392330 and stegovariance = 45
```

Since the PQS criterion is met with the value 2.39, shown in Figure 14B, we proceed with the robustness testing. Had the PQS value been less than the threshold, we would select a new cover image and restart the process. If the stegoimage will not encounter any distortions or noise in being transmitted to its final destination, such as the case where only lossless compression will be used over transmission protocols that guarantee no errors, robustness testing is not needed and may be skipped.

```
cjpeg -grayscale -quality 90 -progressive -restart 1 tf4D.aquc > tf4D.aquc.jpeg
cp tf4D.aquc.jpeg tmp.jpeg;
noise 121321 0.000100 0 < tmp.jpeg > tf4D.aquc.jpeg
djpeg -grayscale -pnm tf4D.aquc.jpeg > tf4D.aquc_rec.pgm
Corrupt JPEG data: 18 extraneous bytes before marker 0xd0
stegoder -d -key 1024 tf4D.aquc msg_test 0 45
 comparing embedded message with recovered message = 0
"Ask not what your country can do for you, but
ask what you can do for your country." "The rain
in Spain falls mainly in the plain."
```

The decompressed JPEG image to which noise was added is shown in Figure 14C. Note that although the added noise caused the JPEG program some problems, as evidenced by the artifacts in the images, the stegosystem ECC enabled the correct message to be recovered.

We have shown the successful generation of a stegoimage. Two other outcomes are possible. First, it may happen that the message is too large to be embedded in the available cover images. In this case, the program has no choice other than to admit failure. A second possibility occurs when a message can be embedded but the PQS criteria cannot be met. In this case, a the stegoimage that came closest to meeting PQS criteria and also allowing successful recovery of the message is output. Cases where the robustness test fails also result in a "best effort" being output.

4. Conclusion

In this report, we have presented a novel framework for automated steganography. After outlining the generic problem, the problem was constrained, and assumptions were made based upon common steganographic operating scenarios. Finally, a prototype implementation was demonstrated that establishes the basic soundness of the technique. However, much work remains to be done. Perhaps the most important would be the development of a more versatile picture quality metric, particularly one that would allow for different image sizes.

The SSIS stegosupervisor could be enhanced by precomputing many of the factors now obtained by iteration, such a PQS. Since the modulation of the random values does not effect the statistics of the Gaussian noise, an arbitrary signal with the desired power may be used to calculate a PQS for an cover image; thus, given a required PQS threshold, a capacity for a given cover image may be determined. This capacity must then be divided between the message to be embedded and the error control code.

INTENTIONALLY LEFT BLANK.

5. References

1. Marvel, L. M. "Image Steganography for Hidden Communication." ARL-TR-2200, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, April 2000.
2. Smith, J. R., and B. O. Comisky. "Modulation and Information Hiding in Images." *Information Hiding, First International Workshop*, vol. 1174 of *Lecture Notes in Computer Science*, pp. 207-226, edited by R. Anderson, Berlin: Springer-Verlag, 1996.
3. Marvel, L. M., C. G. Boncelet, Jr., and C. T. Retter. "Spread Spectrum Image Steganography." *IEEE Transactions on Image Processing*, vol. 8, no. 8, pp. 1075-1083, August 1999.
4. International Telecommunication Union Radiocommunication Bureau. "ITU-R BT.500 Methodology for the Subjective Assessment of the Quality of Television Picture." <http://ext-www-proxy.itu.ch/itudoc/itu-r/rec/bt/index.html>, November 1997.
5. Miyahara, M., K. Kotani, and V. R. Algazi. "Objective Picture Quality Scale (PQS) for Image Coding." *IEEE Transactions on Communications*, vol. 46, no.9, pp. 1215–1226, September 1998, <http://info.cipic.ucdavis.edu/scripts/reportPage?96-12.>, 1996.
6. Retter, C. T. "Binary Weight Distributions of Low Rate Reed-Solomon Codes." ARL-TR-915, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, December 1995.
7. Retter, C. T. "Decoding Binary Expansions of Low-Rate Reed-Solomon Codes Far Beyond the BCH Bound." *Proceedings of the 1995 IEEE International Symposium on Information Theory*, p. 276, British Columbia: Whistler, September 1995.
8. Wallace, G. "The JPEG Still Picture Compression Standard." *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, April 1991.
9. Marvel, L. M., C. G. Boncelet, Jr., and C. T. Retter. "Reliable Blind Information Hiding for Images." *Information Hiding, Second International Workshop*, vol. 1525 of *Lecture Notes in Computer Science*, edited by D. Aucsmith, Berlin: Springer-Verlag, 1998.
10. Katzenbeisser, S., and F. Petitcolas, editors. *Information Hiding: Techniques for Steganography and Digital Watermarking.* Norwood, MA: Artecg House, Inc., 2000.
11. Marvel, L. M., C. G. Boncelet, Jr., and C. T. Retter. "Spread Spectrum Image Steganography." ARL-TR-1698, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, June 1998.
12. Boncelet, C. G. Jr., L. M. Marvel, and C. T. Retter. "Spread Spectrum Image Steganography (SSIS)." U.S. Provisional Patent 06/082634, April 1998.
13. Bernd, B. "What's Wrong With Mean-Squared Error?" *Digital Images and Human Vision*, pp. 207–220, edited by A. B. Watson, Cambridge, MA: MIT Press, 1993.
14. Bradley, A. P. "A Wavelet Visible Difference Predictor." *IEEE Transactions on Image Processing*, vol. 8, no. 9, p. 718, May 1999.

15. Daly, S. "The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity." *Digital Images and Human Vision*, pp. 179–206, edited by A. B. Watson, Cambridge, MA: MIT Press, 1993.

16. International Telecommunication Union Radiocommunication Bureau. ITU-T Rec. J.61, "Transmission Performance of Television Circuits Designed for Use in International Connections." <http://www.itu.int/itudoc/itu-t/rec/j/index.html>, June 1990.

17. Van Dijk, A. M., J. B. Martens, and A. B. Watson. "Quality Assessment of Coded Images Using Numerical Category Scaling." *Proceedings of the IEEE 1995 SPIE*, vol. 2451, pp. 90–101, 1995.

18. Perlmutter, S. M., P. C. Cosman, R. M. Gray, et al. "Image Quality in Lossy Compressed Digital Mammograms." *Signal Processing - Special Issue on Medical Image Compression*, vol. 59, pp. 189–210, June 1997.

19. Jain, A. K. "Fundamentals of Digital Image Processing." Englewood Cliffs, NJ: Prentice Hall, pp. 51–52, 1989.

List of Abbreviations

CCIR	International Radio Consultative Committee
CIPIC	Center for Image Processing and Integrated Computing
ECC	Error Control Code
HVS	Human Visual System
JPEG	Joint Photographic Experts Group
MOS	Mean Opinion Score
MSE	Mean Square Error
PQS	Picture Quality Scale
PSNR	Peak Signal to Noise Ratio
SSIS	Spread Spectrum Image Steganography

INTENTIONALLY LEFT BLANK.

NO. OF COPIES	ORGANIZATION
2	DEFENSE TECHNICAL INFORMATION CENTER DTIC OCA 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	HQDA DAMO FDT 400 ARMY PENTAGON WASHINGTON DC 20310-0460
1	OSD OUSD(A&T)/ODDR&E(R) DR R J TREW 3800 DEFENSE PENTAGON WASHINGTON DC 20301-3800
1	COMMANDING GENERAL US ARMY MATERIEL CMD AMCRDA TF 5001 EISENHOWER AVE ALEXANDRIA VA 22333-0001
1	INST FOR ADVNCD TCHNLGY THE UNIV OF TEXAS AT AUSTIN 3925 W BRAKER LN STE 400 AUSTIN TX 78759-5316
1	US MILITARY ACADEMY MATH SCI CTR EXCELLENCE MADN MATH THAYER HALL WEST POINT NY 10996-1786
1	DIRECTOR US ARMY RESEARCH LAB AMSRL D DR D SMITH 2800 POWDER MILL RD ADELPHI MD 20783-1197
1	DIRECTOR US ARMY RESEARCH LAB AMSRL CI AI R 2800 POWDER MILL RD ADELPHI MD 20783-1197

NO. OF COPIES	ORGANIZATION
3	DIRECTOR US ARMY RESEARCH LAB AMSRL CI LL 2800 POWDER MILL RD ADELPHI MD 20783-1197
3	DIRECTOR US ARMY RESEARCH LAB AMSRL CI IS T 2800 POWDER MILL RD ADELPHI MD 20783-1197
	ABERDEEN PROVING GROUND
2	DIR USARL AMSRL CI LP (BLDG 305)

NO. OF COPIES	ORGANIZATION
2	COMMANDER US ARMY CECOM RDEC STCD AMSEL RD ST SP P VAN SYCKLE C TZATZALOS FORT MONMOUTH NJ 07703-5202
	ABERDEEN PROVING GROUND
16	DIR USARL AMSRL CI N RADHAKRISHNAN J GANTT AMSRL CI C J GOWENS AMSRL CI CN G RACINE L MARVEL (5 CPS) C RETTER AMSRL CI CT F BRUNDICK G HARTWIG (5 CPS)

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 2002	October 1999–October 2000

4. TITLE AND SUBTITLE
Automated Steganography

5. FUNDING NUMBERS
611102.AH48

6. AUTHOR(S)
George Hartwig and Lisa Marvel

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: AMSRL-CI-CT
Aberdeen Proving Ground, MD 21005-5067

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-2642

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

10.SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

We outline a novel framework for automating steganographic and watermarking processes. Then, using an implementation built on this framework, we demonstrate the effect of a particular steganographic algorithm on picture quality and show how the framework may be used to generate acceptable, robust steganographic images without human intervention. The final system can serve to objectively compare the performance of various steganographic techniques.

14. SUBJECT TERMS
steganography, PQS, watermarking, images, automatic

15. NUMBER OF PAGES
35

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

INTENTIONALLY LEFT BLANK.