ARMY RESEARCH LABORATORY

Reducing Spread Spectrum Image Steganography (SSIS) Extraction Errors With Feedback-Driven Adjustment

by Frederick S. Brundick, George W. Hartwig, Jr., and Lisa M. Marvel

ARL-TR-2889 December 2002

NOTICES

Disclaimers

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5067

ARL-TR-2889 **December 2002**

Reducing Spread Spectrum Image Steganography (SSIS) Extraction Errors With Feedback-Driven Adjustment

Frederick S. Brundick, George W. Hartwig, Jr., and Lisa M. Marvel
Computational and Information Sciences Directorate, ARL

Abstract

Steganographic techniques are useful to hide information in various types of common multimedia data for covert communication. Spread Spectrum Image Steganography (SSIS) is a data-hiding/hidden-communication method that uses digital imagery as a cover signal. This report examines the error sources in SSIS, their impact on payload throughput, and ways of minimizing these errors. We present a method that employs a feedback-driven adjustment to anticipate extraction errors and compensate for them without adversely affecting the detectability of the stegomessage. The technique, which is performed by the transmitter, does not require any additional effort on the part of the recipient. We also describe an experiment using multiple cover images to evaluate the change in error rate as a function of varying input values along with modifications to the SSIS message embedding process. In conclusion, we propose enhancements to the feedback process, enabling it to be used not only for error reduction, but also as a means of limiting the detectability of a stegomessage.

Acknowledgments

We would like to acknowledge all of the people who have contributed to the U.S. Army Research Laboratory steganographic project, particularly Dr. Charles Retter and Dr. Malcolm Taylor. We would also like to thank Ann Brodeen for reviewing this report.

INTENTIONALLY LEFT BLANK.

Table of Contents

INTENTIONALLY LEFT BLANK.

List of Figures

List of Tables

1. Introduction

In this report we examine the error sources of a blind steganographic system known as Spread Spectrum Image Steganography (SSIS) [1–6]. A blind steganographic system is one that does not require the receiver to have a copy of the cover image. SSIS is particularly interesting in that its use is difficult to detect, and it exhibits a degree of tolerance to external noise such as that introduced by communication channels or compression algorithms. To accomplish this, SSIS takes the message to be hidden, applies an error-correcting code (ECC), and uses a modulation technique to convert it into a spread spectrum signal that has a Gaussian distribution. This stegosignal is then added to the cover image and quantized to conform to the image file specifications.* The resulting stegoimage may then be sent to the receiver.

To retrieve the hidden message from the stegoimage, the receiver must generate an approximation of the original cover image. The stegosignal is then recovered by subtracting this approximation from the received image. Because of difficulties in generating an accurate approximation of the cover image, errors occur in the recovered stegosignal. This typically results in a 20%–30% bit error rate in the stegosignal. To combat these errors and others caused by external sources (e.g., noisy communication channel and data compression), a large portion of the available payload must be devoted to error-correcting codes. In the following sections we examine the details of the SSIS process and the nature of these errors. We propose techniques to minimize their occurrence, thereby decreasing the necessity of low-rate ECCs† and increasing the effective bandwidth of the cover image.

2. Steganography System—SSIS

SSIS is a data-hiding scheme that approaches steganography as a communication problem and considers the cover image as the channel *through which* the hidden information is sent. It uses an ECC to encode the message information, 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 error-correcting code, the hidden information is recoverable in cases where the image has been compressed or exposed to additive channel noise.

Figure 1 represents the processing of the SSIS embedder. Within the system, the message is optionally encrypted with key 1, encoded via a low-rate error-correcting code, and interleaved using key 2, producing the encoded message m. The sender enters key 3 into a wideband pseudorandom noise generator, producing a real-valued noise sequence n with a mean of zero and a user-specified variance. Subsequently, the modulation scheme is used to combine the message with the noise sequence, thereby producing the stegosignal s. This stegosignal is now added with the cover image f, then appropriately quantized and processed to preserve the typical dynamic range of the cover

*SSIS currently works with 8-bit gray scale images, but could easily be extended to work with color images and video as well as audio signals.

†If there are many errors, a lot of redundancy is required.

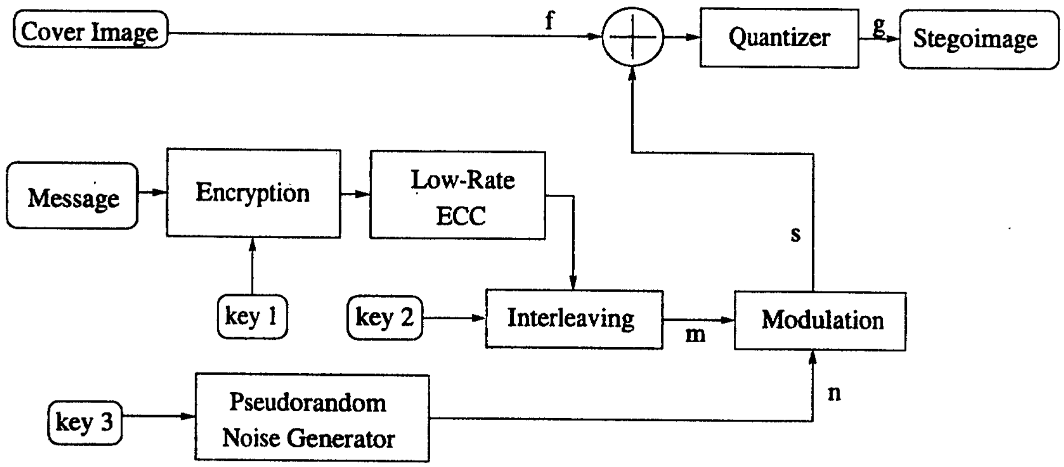


Figure 1. SSIS Embedder.

image, producing the stegoimage g. The stegoimage is then transmitted in some manner to the recipient.

Figure 2 depicts the major components of the SSIS extractor. The stegoimage is received by the recipient, who maintains the same keys as the sender and uses the stegosystem extractor to extract the hidden information. The extractor uses image filtering techniques to produce an estimate of the original cover image $\hat{f}$ from the received stegoimage g. The difference between g and $\hat{f}$ (the estimated stegosignal $\hat{s}$) is demodulated using the noise sequence n that was regenerated with key 3, producing an estimate of the interleaved and encoded message $\hat{m}$. This estimate of the message is fed into a keyed deinterleaver, decoded via the low-rate error-correcting decoder, optionally decrypted using key 1, and revealed to the recipient.

3. Error Sources

3.1 Overview

In this section, we will identify the error sources inherent in the SSIS system and attempt to quantify their effect on the overall efficiency of this steganographic algorithm. The main reason that there are errors is because this is a blind system. The original cover image is not provided to the recipient, who must construct an estimate of the original image.

There are also errors caused by external processes such as communication channel noise and data compression. They are beyond our control and are not addressed in this report. However, if we reduce the number of internal SSIS errors, then our ECC will be able to correct more external errors, resulting in a more robust system.

Figure 3 shows the images that we will use in this study along with their respective intensity histograms.* Each image is 256×256 pixels for a total of 65536 pixels. We include the histograms

*Notice that the vertical scales differ in the histograms.

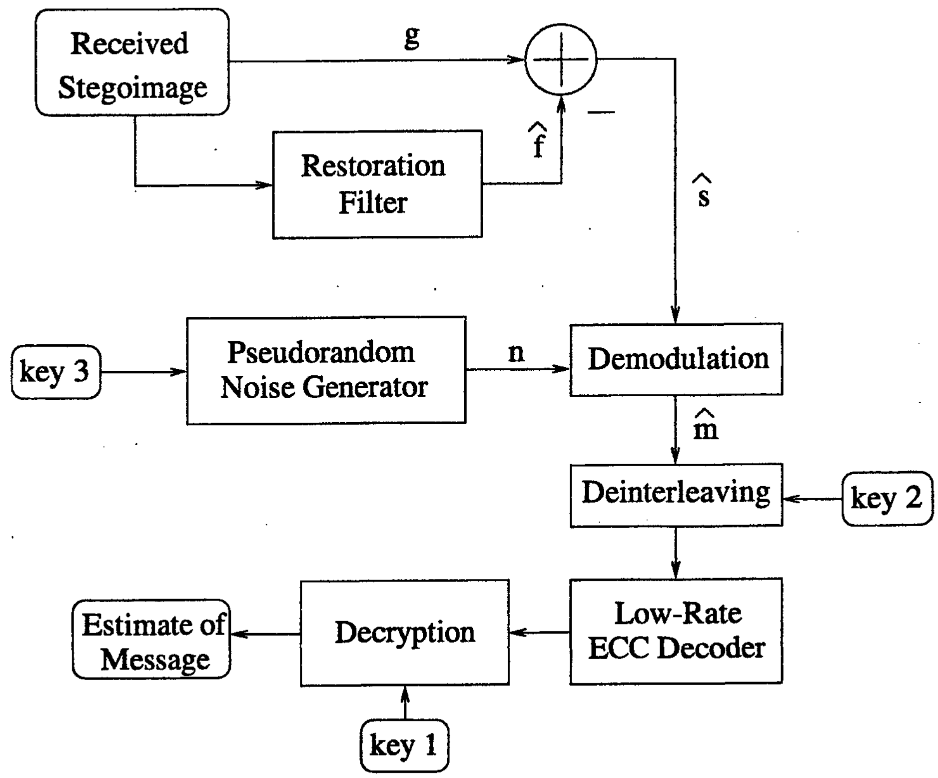


Figure 2. SSIS Extractor.

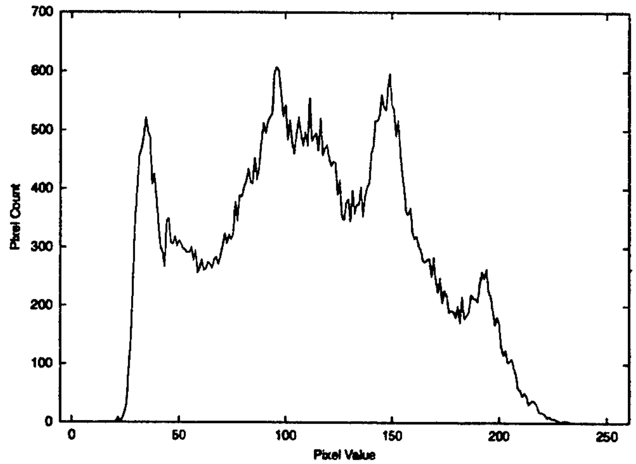
because some of the techniques that can be applied to mitigate errors have profound effects on the distribution of pixel values in a stegoimage and are unacceptable since they may lead to detection. SSIS requires that the embedded signal added to the cover image must have the same characteristics as noise inherent to the image, namely low-power white Gaussian noise. Any modifications made to the system must maintain the goal of undetectability or it is not successful steganography.

3.2 Quantization and Clipping

The 8-bit gray scale images have pixels with integer values in the range 0–255, while the stegosignal consists of a stream of floating point values. To make the stegoimage conform to the format specifications of a typical image, a quantizer is used as shown in Figure 1. Roundoff errors occur in the process of quantizing the floating point stegoimage values. This, at least in work to date, has not proven to be a measurable source of error.

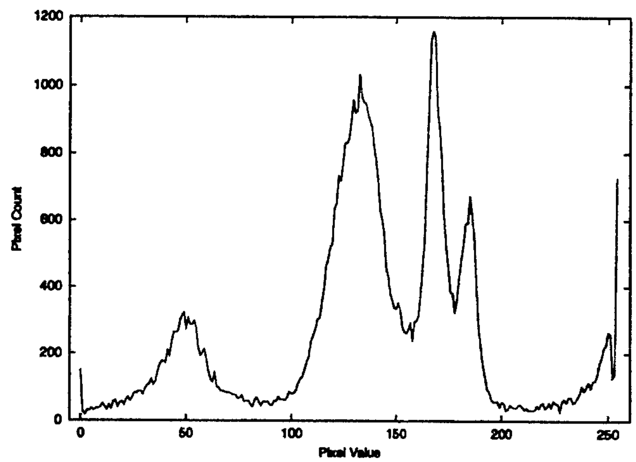
Clipping occurs when the added stegosignal causes the pixel value to exceed the permitted values of the image file format (e.g., 0–255). The number of pixels affected by this type of error is dependent upon both the cover image and the variance of the noise. For instance, among our test images the Eiger image is much more susceptible to clipping than the Barbara image because it has many pixels near the extremes of the allowable range. The M2 is even more susceptible as the histograms in Figure 3 demonstrate with the tall spikes near the range endpoints of 0 and 255.

(a) Original Barbara Image



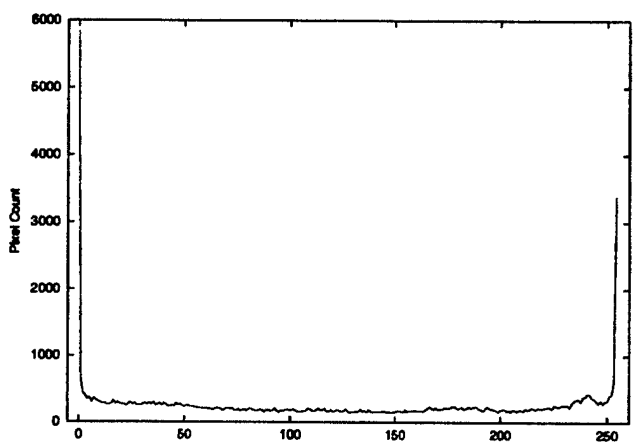
(b) Barbara Image Histogram

(c) Original Eiger Image



(d) Eiger Image Histogram

(e) Original M2 Image



(f) M2 Image Histogram

Figure 3. Sample Images.

3.3 Filtering

The major source of extraction errors is imperfect estimation of the original cover image. If the restoration filter could perfectly reproduce the original cover image, the only errors in the extracted noise would be quantization and clipping errors.

We add white Gaussian noise—such as the arbitrarily chosen sequence plotted in Figure 4—to the cover image to embed the steganographic message. To recover the stegosignal, a low-pass filter is used to generate an estimate of the cover image. Unfortunately, low-pass filters cannot differentiate between the high-frequency component due to the stegosignal and naturally occurring high frequencies in the cover image (e.g., edges). This results in a large number of errors when the stegosignal is extracted.

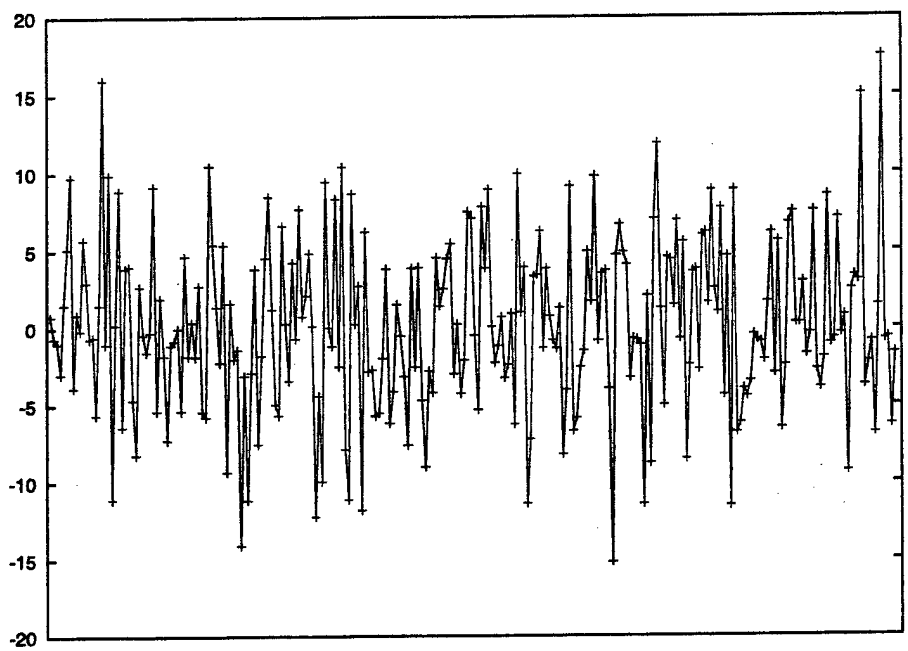
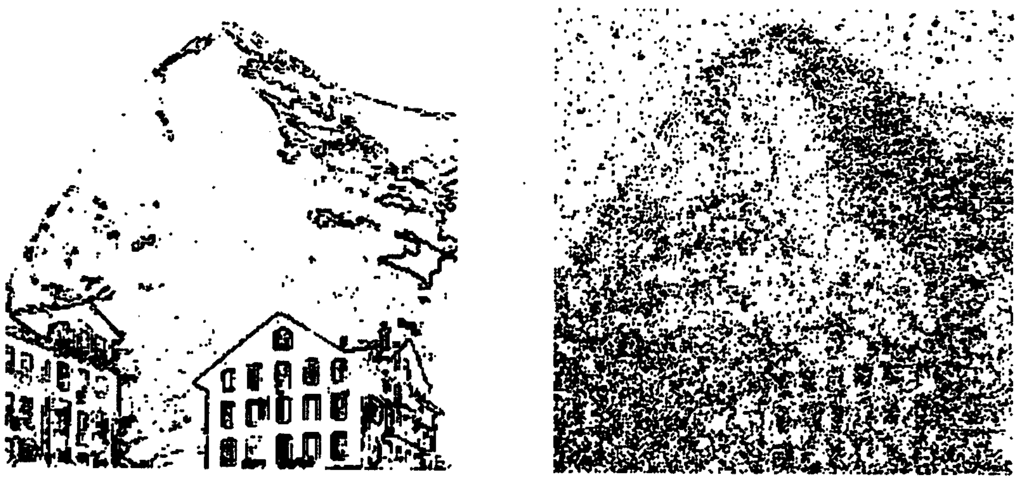


Figure 4. White Gaussian Noise, Variance = 30.

In order to explain the extraction details, we must first show how the modulation process works. A matrix of white Gaussian noise, g_1, is generated with key 3, then transformed using a piecewise linear modulation into a second matrix, g_0, in such a way that the minimum distance between each pair of values is maximized. The stegosignal takes on the noise value g_1 when the stegomessage bit is a 1, and g_0 when the bit is a 0. When the estimated stegomessage is extracted, the algorithm determines whether an extracted noise value is closer to the g_1 value or the g_0 value and sets the message bit to 1 or 0, respectively.

To graphically display where extraction errors are occurring, binary error maps may be generated. A white pixel is present when the extracted message bit matches the corresponding bit in the original message, while a black pixel indicates an extraction error. An algorithm may be run on a cover image to locate the edges in the image as shown in Figure 5a.* The error map in Figure 5b shows that errors are not uniformly distributed throughout an image, but are more likely to occur along edges and other high-contrast areas. For instance, the transition between the mountain and the sky is clearly visible, as are the windows in the building.



(a) Detected Edges (b) Error Map

Figure 5. Edge Detection and Error Map.

4. Error Correction

4.1 Objective

The stegomessage which is to be added to a cover image is first encoded with the ECC bits, resulting in a larger sequence, and then padded to make the stegomessage the same size as the cover image. The number of bits in the stegomessage is the same as the number of pixels in the cover image because noise is added to every pixel in the cover image to minimize detectability. The amount of error-correcting overhead required is generally proportional to the number of errors that are expected. A larger noise variance results in fewer extraction errors but also a lower quality (noisier) stegoimage, while a smaller variance produces a visibly better stegoimage with more errors. If the number of internal errors could be reduced, a smaller variance could be used for the same size payload, a larger payload could be embedded using the same variance, or the message could be better protected against external errors.

*The edge detection technique used is to threshold the Pythagorean sum of two Sobel gradient operators at 90 degrees to each other.

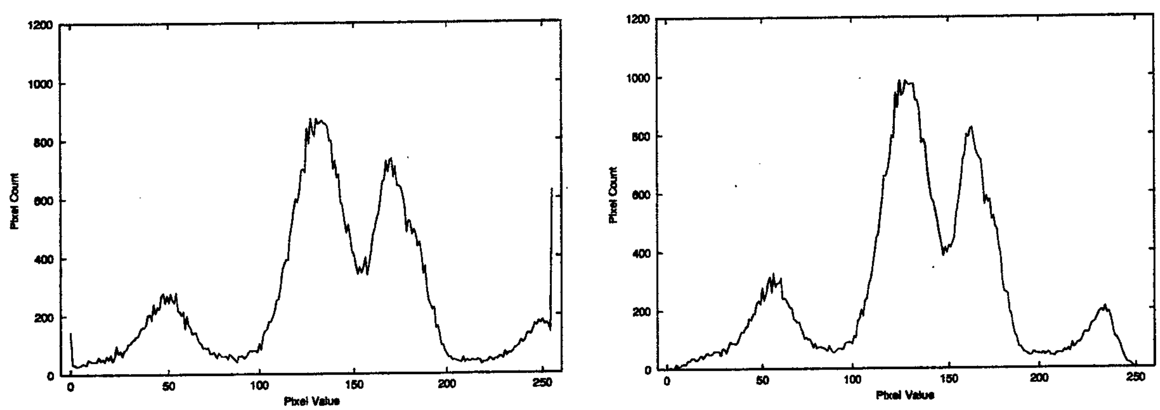
4.2 Minimizing Clipping Errors

4.2.1 Clipping vs. Scaling

The goal with both clipping and scaling is to keep the pixels in the stegoimage within the allowable value range of 0–255. With clipping, the cover image and stegosignal are added together, and pixels that exceed the range are replaced with 0 or 255 accordingly. This results in spikes at the ends of the intensity histogram as shown in Figure 6a.

We may prevent out-of-range values by scaling the cover image before adding the stegosignal to it. The scale factor is chosen such that the pixel values of the stegoimage include the full range of values from 0 to 255. Scaling has the side effect of causing the intensity histogram to taper off at the ends as in Figure 6b.

A third approach is to combine the two methods. The scale factor could be chosen such that some of the stegoimage pixels extend beyond the allowable extreme values, requiring them to be clipped. However, the scale factor could be adjusted to cause the intensity histogram of the stegoimage to be comparable to the original histogram, minimizing detectability of the stegomessage. This technique was not tested in our experiments.



(a) Clipped (b) Scaled

Figure 6. Histograms of Clipped and Scaled Stegoimages.

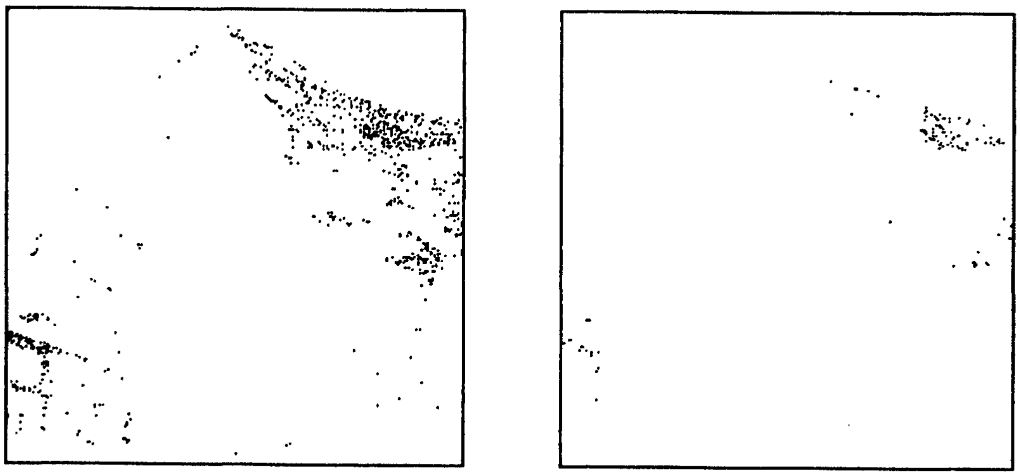
4.2.2 Clipping

In order to estimate the impact of clipping on the extraction process, a stegoimage with a noise variance of 60 was created using the Eiger cover image.* The estimated message was extracted from the stegoimage by subtracting the original cover image. This is the same as using a perfect filter, and all errors are now caused by quantization and clipping. In this example, 161 pixels

*This value may be too large for practical use because of detectability issues, but the goal was to intentionally cause clipping.

underflowed and 674 pixels overflowed, for a total of 835 clipped pixels, while only 326 of these caused an actual extraction error (for an error rate of $326/65536 = 0.5\%$).

The process was then repeated using an alpha-trimmed filter to generate the estimated cover image. The pixels that were clipped were compared with the pixels that produced extraction errors. Figure 7a shows all pixels where clipping was necessary to keep them within the allowable range. (The pixels in the building outline in the lower left corner were less than 0, while others were greater than 255.) The picture in Figure 7b shows the location of clipped pixels that caused an error in the extracted stegomessage. Of the 835 clipped pixels, 108 of them caused an error. This would appear to be better than using the original cover image instead of applying a filter, but this was only one source of errors. An additional 16373 errors were caused by the filtering process. Since the total number of errors was 16481 (25%), clipping had a negligible impact (0.66% of all errors).



(a) All Clipped Pixels (b) Clipping Errors

Figure 7. Clipped Pixels.

The M2 image was subjected to the same test. A stegoimage was generated using the same variance of 60, then the estimated message was extracted with an alpha-trimmed filter. Because of the cover image's large number of pixels at the allowable pixel value extremes, 3850 pixels underflowed and 2219 overflowed. Of these 6069 pixels, 1412 caused an extraction error. There were a total of 21942 errors (33%), with clipped pixels accounting for 6.4% of the errors. While this is a factor of 10 larger than the impact that clipping had on the Eiger image, if the clipping errors could be removed, the total number of errors would be reduced to 31%, a small improvement in error rate.

The pixel intensity distribution of the Barbara image is the opposite of that of the M2 image. The values are concentrated in the middle of the range, with extreme values of 13 and 235. These are far enough from the allowable values of 0 and 255 that pixels in the stegoimage never had to be clipped. The Barbara stegoimage had the smallest number of extraction errors: 15577 or 24%.

The clipping errors occurred predominantly where clipped pixels were clustered together, and not where they were widely scattered. This is reasonable because the type of image estimation filter that was used, an alpha-trimmed mean filter, deletes the extreme values. The alpha-trimmed mean filter examines a 3 × 3 block of pixels and computes the new pixel value $\hat{y}$ with the equation

$$\hat{y} = \frac{1}{N-2l} \sum_{i=1+l}^{N-l} x_i, \tag{1}$$

where x_i are the pixels in the filter window, sorted by increasing magnitude [7]. In our experiment, the parameter N was set to 9 and l was set to 1. By definition, a clipped pixel will be an extreme value and will therefore be ignored, thus having no impact on the filtered image. On the other hand, a group of clipped pixels will not all be deleted, and the average will be affected.

Another reason the clipping did not cause errors is the binary nature of the extraction process. As explained in section 3.3, the value of an estimated message bit is determined by comparing the estimated stegosignal value with the corresponding pair of white Gaussian noise values. If the estimated noise from a clipped pixel, regardless of its precise value, remains closer to the correct noise value, the bit will still be chosen correctly.

This is shown graphically in Figure 8. The original value for this pixel was 254 and the message bit was a zero. Adding g_0 (7.433) resulted in 261, which was clipped to 255. Because the stegoimage pixel's value was reduced, the filtered image's value was too low (225.6), resulting in a higher estimated noise of 29.4. However, any value greater than the midpoint of the g_1–g_0 interval ($\bar{g} = 2.0$) will cause zero to be chosen as the estimated message bit.

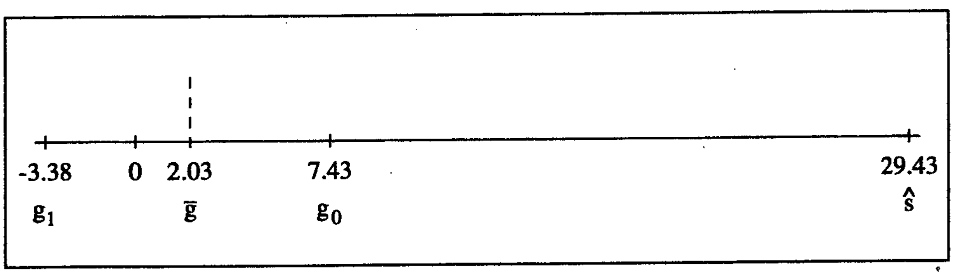


Figure 8. Clipping Without Error.

4.2.3 Scaling

As an alternative to clipping, it is possible to scale the pixel values of the cover image such that when the stegosignal is added to the scaled cover image, the stegoimage pixel values will include—but not exceed—the allowable extreme values. Tests were conducted to quantify the impact of scaling the cover image instead of clipping the stegoimage. Given a cover image and a stegosignal, the SSIS program adds them together and determines the extreme floating point stegoimage values S_{max} and S_{min}. The scale factor (s_f) is calculated using

$$s_f = \frac{255}{S_{max} - S_{min}}. \tag{2}$$

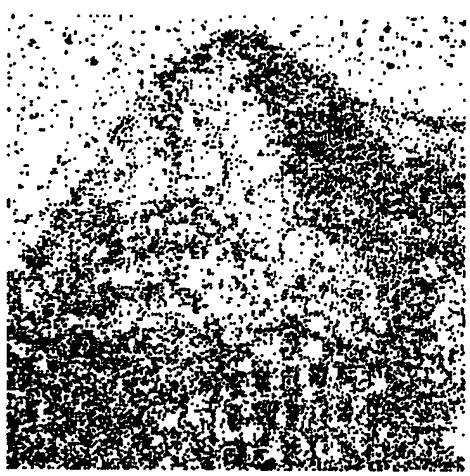
The original cover image is scaled, the same stegosignal is added to it, and the result is quantized to produce the stegoimage.

The intensity histograms of a clipped stegoimage and the stegoimage generated by scaling may be drastically different, possibly making it more apparent that a stegomessage has been hidden in the image. Figure 6 shows histograms for stegoimages that were generated using the Eiger cover image, whose original histogram is shown in Figure 3. Notice that the spikes occurring at 0 and 255 in the clipped image are absent in the scaled image.

As Table 1 shows, the number of extraction errors from a clipped stegoimage is comparable to the number of errors made when the cover image is scaled. The error maps in Figure 9 contain no discernible differences except for the errors that were caused by clipped pixels. Therefore, if the intensity histogram of the cover image resembles a scaled stegoimage rather than a clipped stegoimage, the cover image may be scaled to reduce detectability without seriously affecting the number of extraction errors.

Table 1. Errors From Clipping vs. Scaling

	Eiger Number of Errors			M2 Number of Errors		
Variance	Clipped	Scaled	Change	Clipped	Scaled	Change
10	22465	22469	4	25993	25050	-943
30	18918	18936	18	23674	22551	-1123
60	16481	16663	182	21942	20724	-1218
130	13704	14145	441	19797	18651	-1146



(a) Clipped

(b) Scaled

Figure 9. Error Maps of Clipped and Scaled Stegoimages.

4.3 Filtering Errors

Several different filters have been incorporated into the SSIS program to compare their effectiveness. These filters are an alpha-trimmed filter, a Huber filter, a mean filter, and a median filter [8, 9]. The researcher may optionally provide a prefiltered image. A variety of different external filtering techniques were tried, in the hope that an existing filter could be used, but all of the ones tested gave much higher error rates than the original filters. Unfortunately, a perfect filter does not exist, and the development of a better filter is beyond the scope of this project.

As mentioned earlier, all images contain noise, and it is this property that makes SSIS feasible. If the original noise could be removed before the stegosignal is added, it is hypothesized that fewer extraction errors should occur. This prefiltering technique was tested by using an alpha-trimmed filter on the cover image before generating the stegoimage, then using an alpha-trimmed filter again to extract the estimated message. The error rate was reduced by 50%, but the stegoimage, as shown in Figure 10, was unacceptable because the prefiltering smoothed out the edges, making the new image very blurry. This example shows that image quality and detectability of the hidden message must be considered when modifications are made to the SSIS process.

Figure 10. Prefiltered Stegoimage.

4.4 Improved Error Correction

When a stegoimage is created with the SSIS process, errors occur because of clipping, the inability to perfectly recreate the cover image, and external sources such as channel noise and data compression. To address these errors, the inclusion of error-correcting code in the stegomessage is needed. If the correcting ability of the selected code is powerful enough, it is possible for the original embedded message to be recovered even though the estimated stegomessage contains errors.

The low-rate ECCs used with SSIS are capable of correcting an estimated message that contains a relatively high percentage of errors. However, this capability is at the expense of a large amount of overhead, reducing the size of the payload. The codes developed for SSIS are shown in Table 2.

Table 2. Error-Correcting Codes for 256 × 256 Images

Binary Code	Error Correcting Capability	Payload (bpp)	Payload (bytes)
(155,40)	0.12	0.2581	2114
(378,36)	0.21	0.0952	779
(889,35)	0.27	0.0393	321
(2040,32)	0.34	0.0156	127

The "Binary Code" column shows the total number of bits that are needed to store the given number of message bits to repair the error rate in the "Error Correcting Capability" column. The payload is shown in bits per pixel (bpp), while the payload in bytes is the ratio times the number of pixels in the cover image divided by eight.* For example, if the expected error rate is 12%, every 40 bits of payload will be expanded into 155 bits, giving a payload ratio of $40/155 = 0.2581$. Our test images are 256 × 256 (65536 pixels), resulting in a payload of 2114 bytes.

Due to the SSIS modularity, the error-correcting code that is used, like the various filter algorithms, is independent of the SSIS process. Preliminary work has been performed to use side information—namely, the fact that many errors occur along edges—to improve the number of errors that a given set of codewords may correct [10].

5. Feedback-Driven Adjustment

5.1 Basic Technique

The previous section shows that there are no simple solutions to reduce the number of errors when an embedded message is extracted from a stegoimage. Our approach is based on the premise that we can generate a stegoimage and determine where restoration errors occur at the transmitter. By adjusting the magnitude of the cover image at those locations where errors occur, we may "nudge" the estimated stegosignal value over the threshold in the correct direction. In Figure 8, if $\hat{s}$ was to the left of the threshold ($\bar{g}$), we would attempt to nudge it to the right so that it would become closer to g_0 than g_1, and a zero bit would be correctly chosen.

5.2 The Feedback Embedding Process

The feedback process starts with a stegoimage as produced by the SSIS process. A stegoimage is composed of two parts: a cover image and a stegosignal. Using the notation from Figures 1 and 2, this may be represented by

$$g = f + s \tag{3}$$

*There are eight bits in a byte.

and

$$\hat{s} = g - \hat{f}. \tag{4}$$

Our goal is to modify the cover image f in such a way that the filtered image $\hat{f}$ is closer to the actual cover image, resulting in an estimated stegosignal $\hat{s}$ with fewer errors.

We begin by generating a stegoimage and clipping or scaling as required. Instead of sending this stegoimage to its intended recipient, we extract the estimated stegosignal and compare it with the original stegosignal. If the estimated bit and original bit match (both are 1 or both are 0), an extraction error has not occurred and we do nothing. If they differ (an extraction error did occur), we adjust this cover pixel value to eliminate this error by adding the difference between the midpoint of the two possible Gaussian noise values (g_1 and g_0) and the estimated stegosignal to the corresponding pixel in the cover image:

$$f' = f + \frac{g_1 + g_0}{2} - \hat{s}. \tag{5}$$

After the entire cover image is modified, it is added to the original stegosignal,

$$g' = f' + s, \tag{6}$$

the result is quantized, clipped or scaled, and the final stegoimage is transmitted.

5.3 The Extraction Process

An advantage of the feedback process is the fact that the recipient of the stegoimage is unaware that the cover image was modified before the stegomessage was embedded. He receives the stegoimage, then extracts, decodes, and decrypts the message using the three keys. The transmitter has anticipated the errors that the extractor would originally have made and has compensated for them. The recipient needs no additional information to benefit from this procedure.

5.4 Limitations

As can be seen by the equation for the alpha-trimmed mean filter (equation 1), its use imposes limitations on the effects of the nudging. Once a pixel's value becomes the local maximum or minimum, the alpha-trimmed filter ignores the value. In addition, the modified pixel may be subject to clipping or scaling.

When one pixel is modified, all eight of its neighbors in the approximation may be affected because of the filtering techniques used. All corrections are made based on the extracted values and not the new values. It is possible that the correction made to one pixel will be offset by the changes made to one or more of its neighbors.

6. Experiment and Results

The original SSIS configuration produced clipped stegoimages, and the alpha-trimmed filter was used to extract the embedded message. Therefore, this combination was used as the baseline case for the analysis.

6.1 Stegoimage Generation

6.1.1 Experimental Design

The cover images that were used in the experiment are shown in Figure 3. Four factors were selected to construct the stegoimages:

- cover image
- noise variance
- clipping or scaling and optional nudging
- filter used for feedback

For every combination of image and variance shown in Table 3, 18 stegoimages were generated as enumerated in Table 4. The same message (m) and random number seed for modulation (key 3) were used to create the various stegoimages. The "standard" stegoimage was produced by clipping, the same stegosignal was added to a scaled cover image, and the four nudging techniques were used with each of the four filters.

6.1.2 Stegoimages

As mentioned previously, the extreme pixel values in the Barbara image are well within the image format range, so the stegoimages never had to be clipped or scaled. In addition, the changes caused by nudging were small enough that the modified stegoimage remained within the range. Therefore, only five different stegoimages were created as shown in Figure 11. The original cover image is included so the quality of the images may be compared.

For both the Eiger and M2 cover images, scaling the cover image flattened the tails of the intensity histogram so that nudging did not cause pixels to go out of bounds. Therefore, the scaled/nudged/clipped and scaled/nudged/scaled techniques were replaced with scaled/nudged. Figures 12 and 13 show the original cover image, clipped stegoimage, stegoimage based on the scaled cover image, and three nudged stegoimages using the alpha-trimmed filter for Eiger and M2 images, respectively.

Table 3. Image and Variance Factors

Image	Variance
Barbara	10
Eiger	30
M2	60

Table 4. Experimental Factors for Stegoimage Generation

Technique	Feedback Filter
clip	NA
scale	NA
C/N/C	alpha-trimmed
C/N/C	Huber
C/N/C	mean
C/N/C	median
C/N/S	alpha-trimmed
C/N/S	Huber
C/N/S	mean
C/N/S	median
S/N/C	alpha-trimmed
S/N/C	Huber
S/N/C	mean
S/N/C	median
S/N/S	alpha-trimmed
S/N/S	Huber
S/N/S	mean
S/N/S	median

Note: NA = not applicable
Legend: C = clip, N = nudge, S = scale

(a) Cover Image

(b) Normal Stegoimage

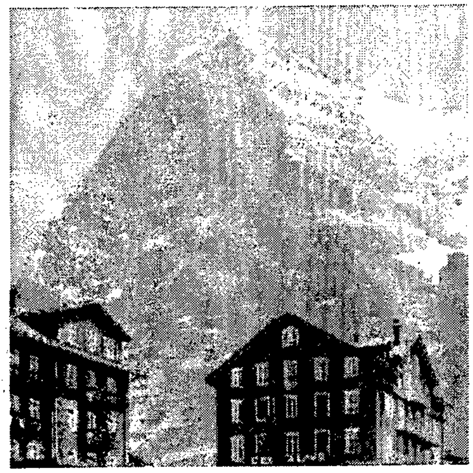
(c) Nudged, Alpha-trimmed Filter

(d) Nudged, Huber Filter

(e) Nudged, Mean Filter

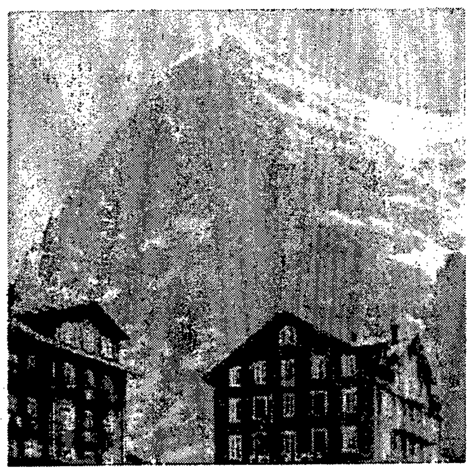
(f) Nudged, Median Filter

Figure 11. Barbara Images, Variance = 30.



(a) Cover Image

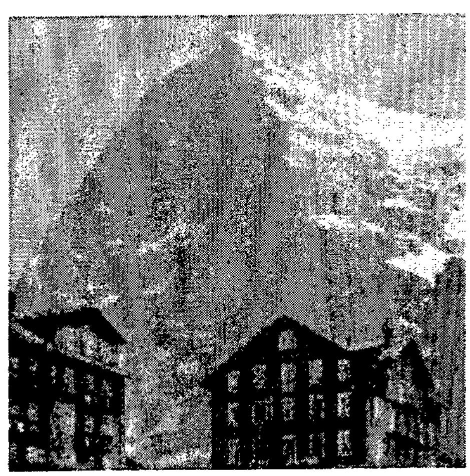
(b) Clipped Stegoimage



(c) Scaled Stegoimage

(d) C/N/C Stegoimage

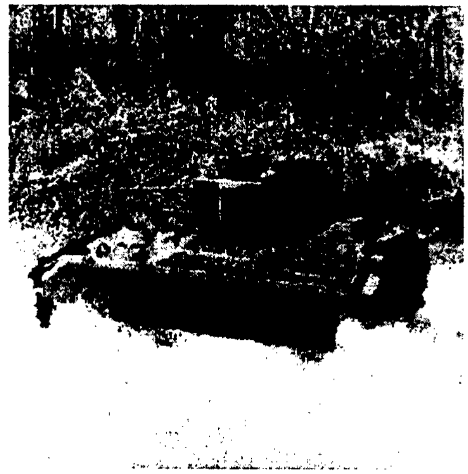
(e) Scaled, Nudged Stegoimage



(f) C/N/S Stegoimage

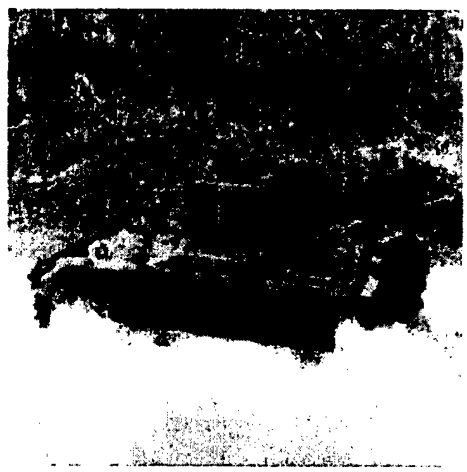
Figure 12. Eiger Images, Variance = 30.

(a) Cover Image



(b) Clipped Stegoimage

(c) Scaled Stegoimage



(d) C/N/C Stegoimage

(e) Scaled, Nudged Stegoimage

(f) C/N/S Stegoimage

Figure 13. M2 Images, Variance = 30.

6.2 Message Extraction

6.2.1 Barbara Cover Image

To compare the effectiveness of the four filters, the estimated stegomessage was extracted from the "normal" Barbara stegoimage (created with every noise variance). For the four nudged stegoimages, the filter used for extraction was the same used when the stegoimage was created.* The stegoimages and extraction filters are listed in Table 5, while the noise variances are in Table 3.

Table 5. Stegoimage Extraction Combinations for Barbara

Stegoimage	Extraction Filter
normal	alpha-trimmed
normal	Huber
normal	mean
normal	median
nudged (a)	alpha-trimmed
nudged (h)	Huber
nudged (u)	mean
nudged (m)	median

Legend: a = alpha-trimmed, h = Huber
u = mean, m = median

6.2.2 Eiger and M2 Cover Images

The Eiger and M2 cover images, unlike the Barbara image, produced stegoimages that needed to be clipped or scaled. The estimated stegomessage was extracted from both the clipped and scaled stegoimages with each of the four filters. The unique stegoimages from Table 4 were clipped/nudged/clipped, clipped/nudged/scaled, and scaled/nudged. As was done with Barbara, the extraction filter used for each stegoimage was the same as the feedback filter. The 20 combinations of stegoimage and extraction filter are shown in Table 6.

6.3 Results

Each of the 20 (8 for Barbara) estimated messages, $\hat{m}$, was compared with the original message, m, and the error rate was computed. The error rates for all the data (including all three noise variances) ranged from 13% to 40%. The relative change in error rate was computed for all of the cases by comparing them with the clipped and alpha-trimmed baseline case. The results for a variance of 30 are shown graphically in Figures 14–16.

*A test was conducted where two different filters were used and is discussed in section 6.4.

Table 6. Stegoimage Extraction Combinations for Eiger and M2

Stegoimage	Extraction Filter
clipped	alpha-trimmed
clipped	Huber
clipped	mean
clipped	median
scaled	alpha-trimmed
scaled	Huber
scaled	mean
scaled	median
C/N/C (a)	alpha-trimmed
C/N/C (h)	Huber
C/N/C (u)	mean
C/N/C (m)	median
C/N/S (a)	alpha-trimmed
C/N/S (h)	Huber
C/N/S (u)	mean
C/N/S (m)	median
S/N (a)	alpha-trimmed
S/N (h)	Huber
S/N (u)	mean
S/N (m)	median

Legend: C = clip, N = nudge, S = scale
a = alpha-trimmed, h = Huber
u = mean, m = median

The curves for the Eiger image are clustered into three sets. Starting with the bottom curves, the clipped and scaled error rates are nearly equal, with the scaled errors slightly worse than the base case. The three mean filters are nearly identical, and the median filter gives worse results. When the stegoimage is clipped, nudged, and clipped again, the error rate is reduced by about 27%. The improvements from scaled and nudged images are just below the clipped values. Clipping, nudging, and scaling generates stegoimages whose extraction error rate improvements are around 40%, a significant amount.

The results for the Barbara image are similar to the middle and bottom Eiger curves. Nudging reduces the error rate by around 25%.

The M2 curves are like the Eiger curves, except scaling is slightly better than clipping instead of being worse. The pair of nudged curves are near 30% and the clipped/nudged/scaled curve varies from 50% to 60% improvement.

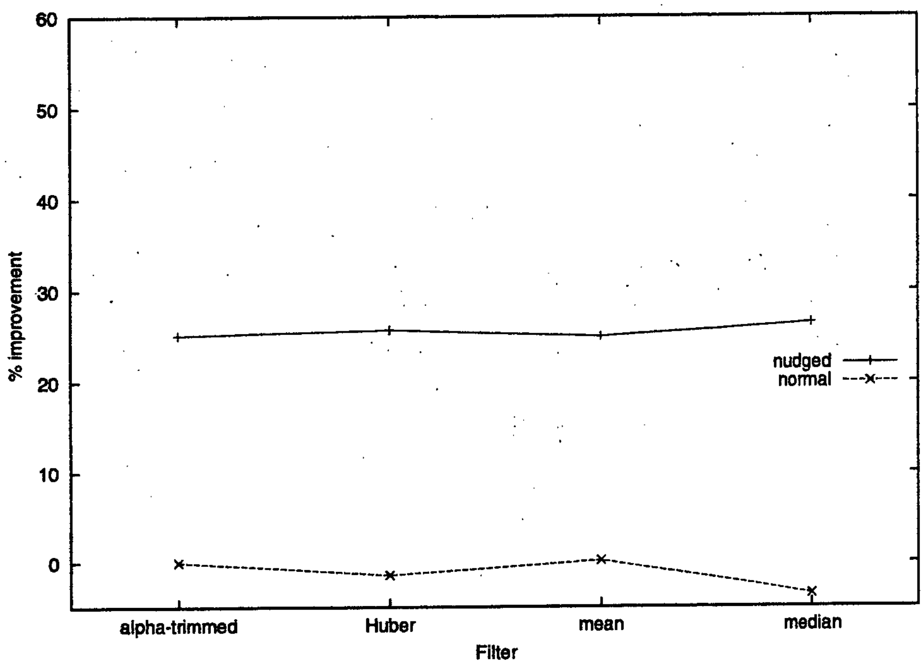


Figure 14. Improvement in Barbara Error Rate.

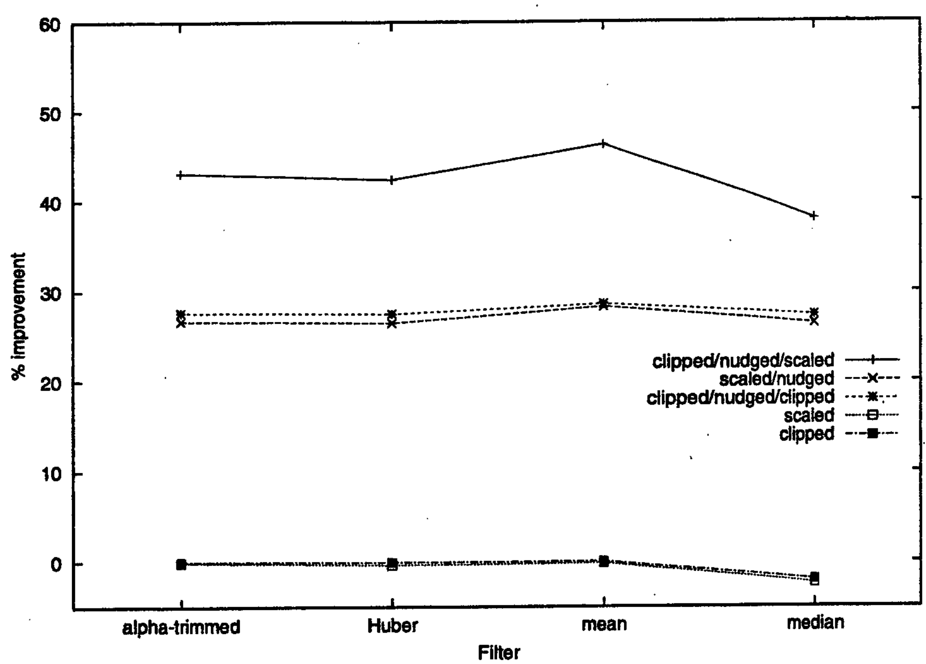


Figure 15. Improvement in Eiger Error Rate.

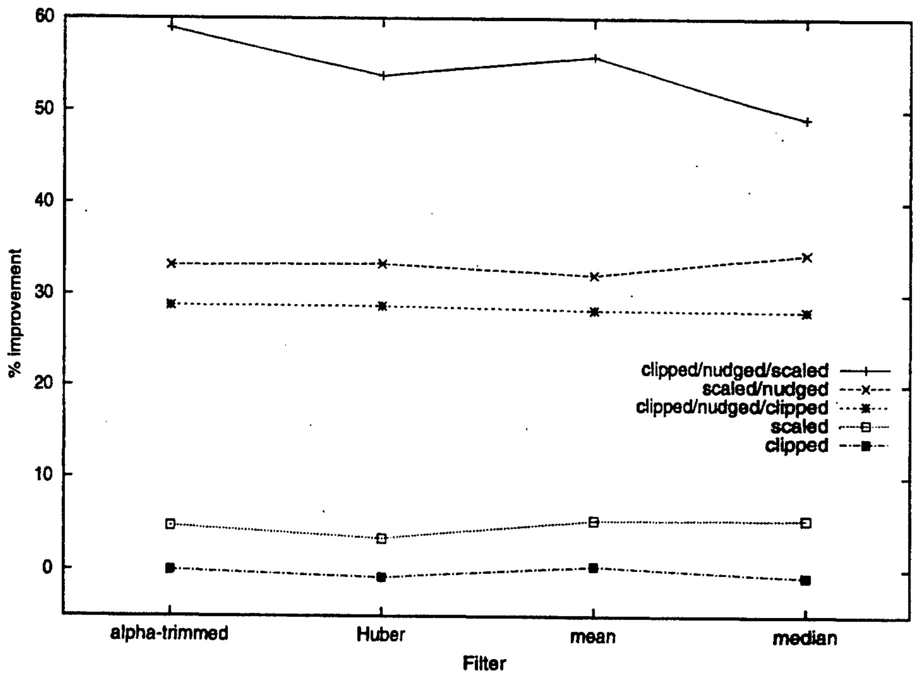


Figure 16. Improvement in M2 Error Rate.

To show the significance of noise variance, the error rate improvements are plotted for the Eiger image as a function of variance in Figure 17. The clipped and scaled curves are omitted because they remain near zero for all three variances. Likewise, the scaled/nudged curves are very close to the clipped/nudged/clipped curves and are omitted for clarity.

In our experiments, using a median filter with the clipped/nudged/scaled algorithm always resulted in an error rate that, while much better than the base case, was always worse than that obtained by other filters. The larger the noise variance, the smaller the amount of improvement. With the exception of the median filter, increasing the variance always resulted in an improvement in the relative error rate. However, the relative improvement for the clipped/nudged/clipped algorithm went down for all filters as the variance increased.

Table 7 may be used to resolve these issues. It shows information about stegoimages created from the Eiger cover image as described in section 6.1.1. The third column contains the number of pixels that were clipped when the initial stegoimage was created. The fourth column lists the number of pixels in the cover image that were nudged, which is the same as the number of extraction errors. The C/N/C columns contain the number of pixels that were clipped after the nudging was performed and the number of errors made when the recipient extracted the message. The corresponding extraction errors from the clipped/nudged/scaled stegoimage are shown in the last column.

A larger variance means more noise, resulting in more clipped points, but the number of errors is reduced because the noise is extracted more accurately from the stegoimage. The number of points that are clipped after nudging is slightly higher than the number before nudging. Pixels that were clipped the first time are likely to be outside of the allowable range again, along with a few points that were nudged past the range endpoints.

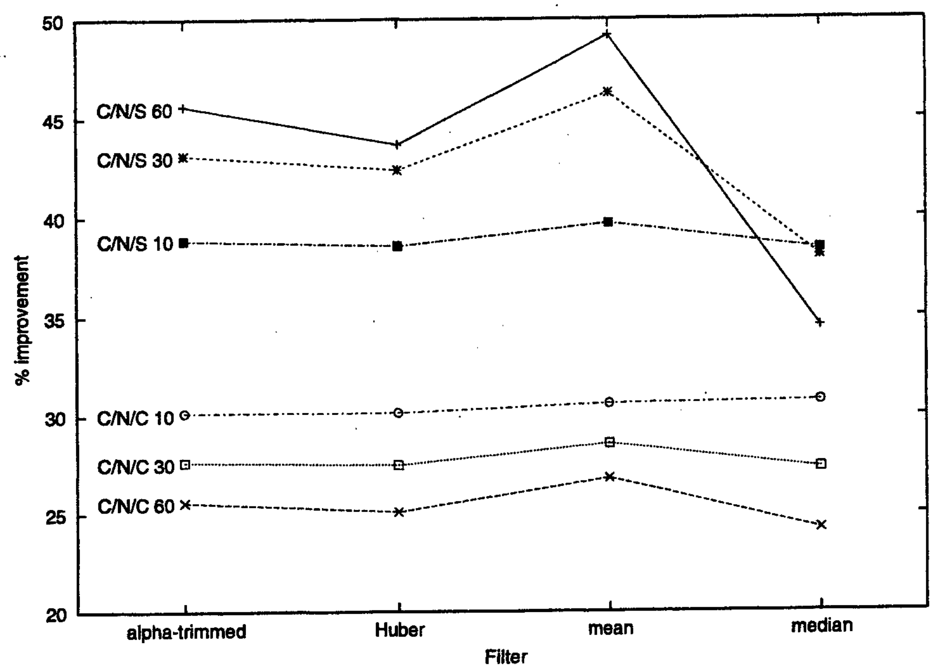


Figure 17. Improvement in Error Rate vs. Variance in Eiger Image.

Table 7. Errors for Eiger With Nudging Plus Clipping or Scaling

				C/N/C		C/N/S
Variance	Filter	Clipped	Nudged	Clipped	Errors	Errors
10	alpha-trimmed	407	22465	408	15692	13743
	Huber	407	22478	410	15690	13804
	mean	407	22573	408	15591	13552
	median	407	22839	417	15562	13843
30	alpha-trimmed	643	18918	653	13691	10761
	Huber	643	18919	656	13720	10899
	mean	643	18932	647	13516	10164
	median	643	19300	658	13751	11723
60	alpha-trimmed	835	16481	843	12266	8962
	Huber	835	16470	844	12346	9284
	mean	835	16479	840	12065	8375
	median	835	16949	852	12492	10795

Clipping after nudging dilutes the effectiveness of the nudging because there is a limit to the amount that a pixel may be nudged until it gets clipped. The number of extraction errors for a given variance is reduced when the nudged image is clipped, but the relative reduction decreases as the variance is increased because of this limiting factor. On the other hand, scaling does not have this restriction, allowing the error rate improvement to increase with the variance.

6.4 Experimental Excursion

To explore the system response to filter interactions, an experiment was conducted where every possible combination of filters was used in the feedback embedding process and during extraction. As shown in Figure 18, each set of four points represents stegoimages that were generated using one filter. For each of the four stegoimages, all four filters were used to extract the estimated message; the order of the extraction filters is the same as the labels on the x-axis. The improvement in error rate was computed relative to the base case where the estimated message was extracted from a clipped stegoimage using an alpha-trimmed filter.

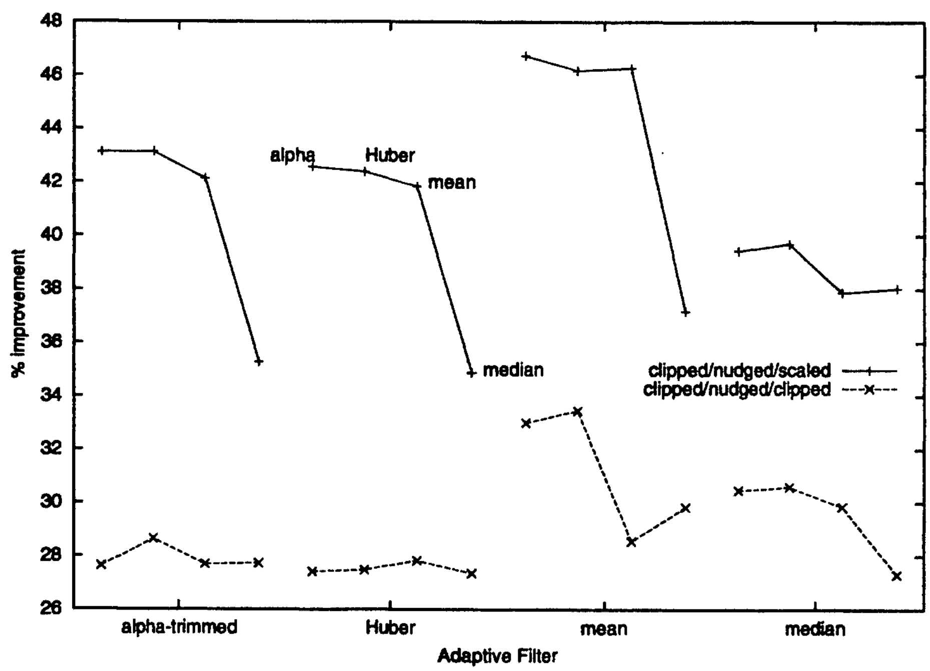


Figure 18. Improvement in Error Rate vs. Filter Combination in Eiger Image.

When the clipped/nudged/clipped algorithm was used, the median filter proved to be effective both as the feedback filter—especially when a different filter was used for extraction—and as the extraction filter. The best improvement was obtained by using a mean filter in nudging and a Huber filter for extraction. The median filter was very poor for extraction with clipped/nudged/scaled. The other filters produced very little variation in error rates.

7. Conclusions and Future Work

We have examined the sources of extraction errors in the SSIS system. Clipping accounted for a surprisingly small number of errors, even with a cover image that had a large number of pixels near the range endpoints. Scaling the cover image to prevent clipping gave mixed results and was comparable to the error rate measured when clipping was used. Several different filters were used to produce an estimate of the original cover image, also with little impact on the error rate.

Nudging the cover image in anticipation of extraction errors resulted in a measurable improvement in the error rate without seriously degrading the quality of the stegoimage. The technique that we developed, while heuristic, added very little to the processing time. If the intensity histogram of a cover image has spikes near the range endpoints, then the sequence of clipping, nudging, and clipping should produce an acceptable stegoimage whose message may be extracted with fewer errors. If the histogram tapers off toward the extremes, then scaling and nudging will produce an equally good stegoimage. The choice of which method to use depends only on the desired shape of the histogram to minimize detection of the stegoimage and can therefore be automated.

The current feedback-driven adjusting technique could be expanded into an iterative process, but care must be taken not to degrade the stegoimage. The image is important information in its own right, not just a means of conveying a hidden message. If the cover image is drastically modified, especially without adequate constraints, the presence of a stegomessage in the resultant stegoimage may be detectable.

If the embedder knows which filter the recipient will be using to extract the message, he could use each filter in the feedback process. The message would be extracted a second time, and the stegoimage that produced the fewest errors would be transmitted.

To maintain image quality, limits should be imposed on nudging. One possible constraint is the local variance of a pixel, while another is the intensity histogram of the entire image. Perhaps "good" pixels could be nudged to balance the changes made to error-causing pixels. The concept of nudging could be applied, not only to reduce the number of extraction errors, but also to minimize the detection of stegomessages.

These experiments have shown that nudging is a viable way of reducing the error rate of the steganography extraction process. Further research and analysis must be performed to refine the process and adapt it to the problem of minimizing detectability.

INTENTIONALLY LEFT BLANK.

8. References

1. 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.
2. 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.
3. Katzenbeisser, S., and F. Petitcolas (eds.). *Information Hiding: Techniques for Steganography and Digital Watermarking*. Norwood, MA: Artecg House, Inc., 2000.
4. Marvel, L. M., C. G. Boncelet, Jr., and C. T. Retter. "Methodology of Spread-Spectrum Image Steganography." ARL-TR-1698, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, June 1998.
5. Boncelet, C. G., Jr., L. M. Marvel, and C. T. Retter. "Spread Spectrum Image Steganography (SSIS)." U.S. Provisional Patent 06/082634, April 1998, final patent application filed February 1999, revised August 2002.
6. Brundick, F. S., and L. M. Marvel. "Implementation of Spread-Spectrum Image Steganography." ARL-TR-2433, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, March 2001.
7. Bednar, J. B., and T. L. Watt. "Alpha-Trimmed Means and Their Relationship to the Median Filters." *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 1, pp. 145–153, February 1984.
8. Palmieri, F., and C. G. Boncelet, Jr. "Ll-Filters—A New Class of Order Statistic Filters." *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 5, pp. 691–701, May 1989.
9. Gonzalez, R. C., and R. E. Woods. *Digital Image Processing*. Reading, MA: Addison-Wesley Publishing Co., 1993.
10. Marvel, L. M., and C. T. Retter. "The Use of Side Information in Image Steganography." Proceedings of the IEEE 2000 International Symposium on Information Theory and Its Applications, Honolulu, HI, November 2000.

INTENTIONALLY LEFT BLANK.

Glossary

alpha-trimmed mean filter an averaging filter that deletes a specified number of extreme pixel values

clipping replacing pixels that exceed the allowable range with the corresponding minimum or maximum value

cover image the original image into which a stegomessage will be embedded

error map a graphical way of showing which message bits were incorrectly extracted

error rate the number of errors divided by the total number of bits in the stegomessage

error-correcting code transforms data and adds redundancy so that it may be checked for errors and corrected

estimated message the sequence of message bits extracted from a stegoimage before it has been error-corrected

external error an error not caused by the SSIS process (e.g., channel compression)

extraction or internal error an error caused within the SSIS process from incorrectly extracting a message bit

intensity histogram a plot showing the distribution of pixel values in an image

interleave rearrange or shuffle bits to distribute errors

low-rate ECC an error-correction code with a low payload to ECC overhead ratio

nudging adjusting the magnitude of the cover image at those locations where extraction errors occur

piecewise linear modulation transforms one Gaussian sequence into another while maximizing the minimum distance between the two

quantize convert floating point pixel values into the integers 0–255

scaling multiplying the cover image by a value less than 1 to keep the stegoimage pixels within the allowable range

stegoimage the final image produced by the SSIS embedder

stegomessage the sequence of bits used to produce a stegosignal

stegosignal the sequence of floating point values to be added to a cover image

white Gaussian noise noise with a constant power spectral density of $N_o/2$. SSIS represents this as Gaussian random variables with $\mu = 0$ and $\sigma^2 = N_o/2$.

INTENTIONALLY LEFT BLANK.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 2002

3. REPORT TYPE AND DATES COVERED
Final, October 2000–July 2002

4. TITLE AND SUBTITLE
Reducing Spread Spectrum Image Steganography (SSIS) Extraction Errors With Feedback-Driven Adjustment

5. FUNDING NUMBERS
611102.AH482TEPE1

6. AUTHOR(S)
Frederick S. Brundick, George W. Hartwig, Jr., and Lisa M. 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-2889

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)*

Steganographic techniques are useful to hide information in various types of common multimedia data for covert communication. Spread Spectrum Image Steganography (SSIS) is a data-hiding/hidden-communication method that uses digital imagery as a cover signal. This report examines the error sources in SSIS, their impact on payload throughput, and ways of minimizing these errors. We present a method that employs a feedback-driven adjustment to anticipate extraction errors and compensate for them without adversely affecting the detectability of the stegomessage. The technique, which is performed by the transmitter, does not require any additional effort on the part of the recipient. We also describe an experiment using multiple cover images to evaluate the change in error rate as a function of varying input values along with modifications to the SSIS message embedding process. In conclusion, we propose enhancements to the feedback process, enabling it to be used not only for error reduction, but also as a means of limiting the detectability of a stegomessage.

14. SUBJECT TERMS
steganography, capacity, hidden communication, implementation, error

15. NUMBER OF PAGES
33

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

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

INTENTIONALLY LEFT BLANK.