Engineering Letters, 13:3, EL_13_3_1 (Advance online publication: 4 November 2006)

StegAnim-A Novel Information Hiding Technique using Animations

Gopalakrishna Reddy Tadiparthi and Toshiyuki Sueyoshi

***Abstract*— Literature in the field of information hiding exhibits a number of steganographic techniques that conceal data effectively. But, conventional steganography utilizes data embedding in some form or the other that is not robust to simple manipulations of the stego-object. Since there are a lot of contemporary steganographic software available, there is a need for measuring the robustness the algorithms used by those software. This paper proposes a novel technique to hide information using animations as cover-object. We also propose a measure for the robustness of the steganographic technique. Experimental results demonstrate the robustness of the proposed technique when applied to digital image steganography. Steganalytic attacks include resizing, sharpening, blurring, and insertion of noise into the stego-object.**

***Index Terms*— Animation, Robustness, Steganography.**

I. Introduction

Information hiding can be broadly divided into two categories: Steganography and Watermarking. Steganography is the art and science of communicating in a way which hides the existence of the communication. While steganography focuses on undetectable communication, watermarking focuses on reliable transmission of the message.

The goal of watermarking is to protect copyrighted multimedia files [1, 2]. Watermarks or copyright notice are used to identify the file as an intellectual property. They both differ in purpose, specifications and detection/extraction methods [3]. In watermarking, the object to be transmitted is the cover signal and embedded data which provides copyright protection. In steganography, the object to be transmitted is the embedded message and the cover signal that serves as a carrier. Therefore, watermark can be visible and there is no need to hide the presence of a watermark. Removal of a watermark renders the host signal useless. So, robustness against malicious attack and signal processing is a primary concern for watermarking. The necessary condition for a steganographic algorithm is to avoid the detection of the embedded message algorithmically or with the help of senses. Over the last few years, steganography has been studied in the framework of computer science. Several algorithms have been developed to hide secret messages in innocent looking data.

Steganography differs from cryptography, because it does not conceal the communication itself. However, it scrambles the data to prevent eavesdroppers understanding the content. Both of these techniques may be considered complementary and orthogonal. Firewalls can be configured to easily detect the presence of cryptographic technique in a given data set arriving at one of the ports. Most good cryptographic tools also produce data that looks almost perfectly random. It can be said that they are trying to hide the information by disguising it as random noise. On the other hand, many steganographic algorithms are not trivial to break even after learning that there is hidden data to find. Cryptographic authentication protocols cannot solve all the issues related to authentication. Cryptographic authentication deals with authenticating the sender of the message over insecure channels. However, once the message (image) is decrypted, the image is unprotected and can be copied and further distributed. This led us to believe that a data hiding model has to be designed in a manner which it uses neither cryptography nor error control coding for robustness.

Steganalysis is the detection of the presence of hidden information in a given multimedia file. It involves two techniques: visual and statistical analysis. Visual analysis uses human eye to detect the presence of hidden information. These techniques can also use some signal processing algorithms (decomposing the image into bit planes) to facilitate the detection. Visual inspection can succeed when secret data is inserted in relatively smooth areas with pixel values near saturation [3]. A relatively powerful detection tool is with the help of statistical analysis. Embedding data in a cover multimedia file changes the statistical behavior of that file. Each steganographic algorithm changes statistical properties of the cover-media in a different way. The insignificantly small universal methods developed to detect embedded stego-data are generally less effective than the steganalytic methods aimed at specific steganographic algorithms [3].

The steganographic terminologies (embedded data, cover, and stego-object) used in this study agree with those outlined by [4]. By definition, information hiding hides a message (the embedded message) under a cover message to yield the stego-message. In robust image steganography, a message is embedded into the image in a robust manner. The robustness in

Manuscript received May 25, 2006.

T. Sueyoshi is a professor at the Department of Management, New Mexico Institute of Mining and Technology, Socorro, NM 87801 USA. (phone: 505-835-6452; fax: 505-835-5498; e-mail: toshi@nmt.edu). He also is a visiting professor at National Cheng Kung University, Department of Industrial and Information Management, Tainan, Taiwan.

G. R. Tadiparthi is a Ph.D. candidate at Department of Computer Science, New Mexico Institute of Mining and Technology, Socorro, NM 87801 USA. (e-mail: gtadiparthi@computer.org).

this study implies the ability to survive common image processing operations, such as lossy compression, filtering, noise adding, geometrical transformations, etc. Classical paintings can be studied for authenticity using sophisticated experimental techniques. There are many techniques available that can alter the image. A visible signature in the corner of the image can be easily replaced or removed with advanced image processing software packages, such as PhotoShop, PaintShop Pro, etc. Additional information in the image header of the image file can be erased or changed, as well. In other words, any attempt to authenticate the digital image by appending information may fail. The embedded information is transparent to the human eye, but it should be detectable, using a detection algorithm, when a secret key is available.

This article is organized as follows: Section II reviews the previous works in this field. Section III describes the proposed model. Section IV discusses the measurement of robustness. Section V provides the experiments and the results obtained in a simulated study. Section VI concludes this paper and describes possible future extensions of the proposed method.

II. Previous Works

The goal of steganography is covert communication. The most general steganographic model presented by Simmons is the prisoners' problem [5]. In this problem, two persons in the jail plan to make an escape together. A warden monitors any communication between them. Thus, they must hide the messages concerning escape plan in another innocuous-looking media. An assumption in this model is that both the sender and receiver must have shared some secret information before imprisonment. The prisoners' problem is classified into secret key steganography. When there is no prior information shared by two communication parties, it is classified into pure steganography. If the sender knows the public key of the receiver, the steganographic protocol is called “public key steganography”[6] [7]. The warden may be passive, that is, he only observes the passing messages. If the warden detects an occurrence of covert communication, the whole purpose of using steganography is defeated. The analysis of different data hiding techniques can be found in Bender [8].

A fundamental requirement of a steganographic system is that the hidden message carried by stego-media should not be detected or noticed when the stego-media is observed casually. The requirements of a steganographic algorithm are described in Venkatraman et al [9] and Chen and Wornell [10]. Chen and Wornell [10] state that there are three conflicting goals to any information hiding technique: maximizing capacity, minimizing distortion between cover-object and stego-object, and maximizing robustness. Information hiding models should be perceptually transparent. Most of the steganographic techniques take advantage of the limitations of human auditory system and human visual system. The data embedded by the sender should not significantly change the characteristics of the cover-object. Steganographic capacity is the amount of data that can be embedded in a cover-object relative to the original size. This data and cover image should withstand any kind of simple transformations and filtering techniques. The steganographic technique should also be tamper proof. There should be an indication if the stego-object has been modified from its created state. Mechanisms should be provided to find out the possible noise or spikes in the transmission medium.

Petitcolas et al [11] classify the steganography into only two categories: linguistic and technical steganography. This is not a sufficient classification given the amount of techniques available in the literature. The different kinds of steganographic techniques found in the literature can be broadly classified as follows:

A. Bit-wise Embedding

The early technique of embedding data was to hide data in the bit planes of images. Chandramouli and Memon [12] provide a complete analysis of the LSB (Least Significant Bit) based steganography techniques and suggest improvements to the simple algorithm by proposing adaptive steganographic technique. Kawaguchi and Eason [13] describe the BPCS (Bit-Plane Complexity Segmentation) based steganography. In that they embed data in varying bit-planes from MSB (Most Significant Bit) to LSB and achieve higher steganographic capacity. Fridrich [14] proposes a steganographic method for embedding messages in palette-based images. The pixels for hiding are chosen randomly using a pseudo-random number generator with the key as a seed. There have been many techniques for hiding information or messages in images in such a manner that the alterations made to the image are perceptually indiscernible. Some of the approaches include Least significant bit insertion (LSB), Masking and filtering, and Transform techniques [15]. LSB techniques embed the bits of the message directly into least significant bit plane of the cover-image in a pre-defined order. Masking and filtering is similar to paper watermarks. This technique is restricted to 24 bit and gray scale images. These techniques perform analysis of the image, thus embed the information in significant areas so that the hidden message is more integral to the cover-image than just hiding it in the noise level. Transform techniques embed the message by modulating coefficients in a transform domain, such as the Discrete Cosine Transform (DCT) used in JPEG compression, Discrete Fourier Transform, or Wavelet Transform, These methods hide messages in significant areas of the cover-image, which make them more robust to attack. Transformations can be applied over the entire image, to block through out the image, or other variants. Sanford et al [16] explain a data embedding method that hides data in the noise component of BMP images.

B. Video Steganography

Noda et al [17] explain video steganography by using image steganographic techniques. They use BPCS steganography combined with wavelet compression. Chae and Manjunath [18] use an embedding scheme based on texture masking and lattice structure. They use the block DCT (Discrete Cosine

Transforms) in individual video frames for embedding data. Westfeld and Wolf [19] describe a steganographic technique used in a video conferencing system. It is a DCT based lossy compression mechanism. George et al [20] analyze the spread spectrum technique when applied to images and video. The spread spectrum method has the advantage that the water-mark extraction is possible without using the original unmarked image.

C. *Audio Steganography*

Inoue and Matsumoto [21, 22] developed a steganography technique for Standard MIDI Files (SMF) using the redundancy of the description of note events in SMF. Tachibana et al [23, 24] propose a steganographic technique for hiding in MPEG Advanced Audio Coding (AAC) using a two-dimensional pseudo-random array.

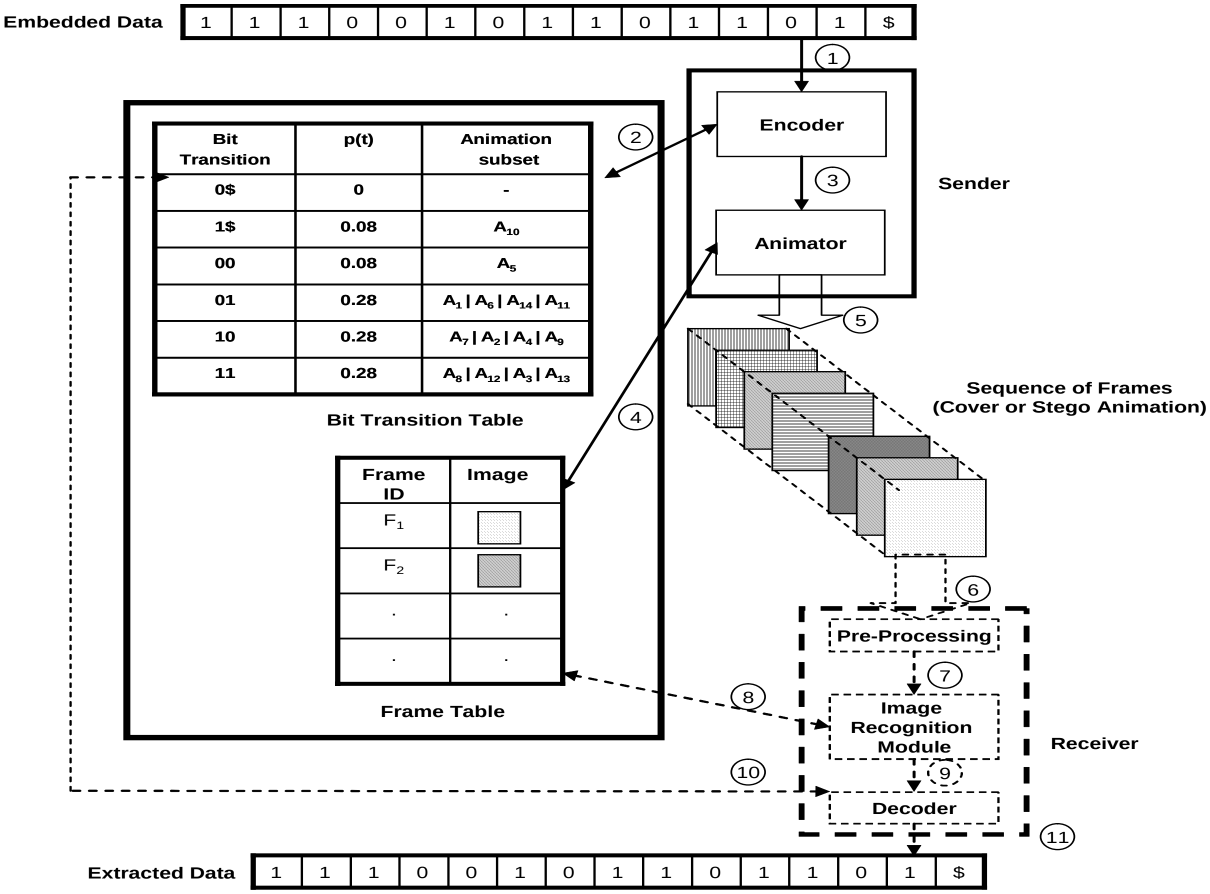
D. *Layered Steganography*

Ratan and Madhavan [25] use a combination of signal processing, cryptography, and steganography to increase the security of information. Sung et al [26] use cryptography, error coding, and encoding in tandem to provide a secure steganographic technique.

Steganographic capacity is assuming an important role in the quality metric of any steganographic software. Chandramouli and Memon [27] define steganographic security and capacity in terms of the steganographic detection algorithm used. Nozaki et al [28] use color BMP images for demonstrating a large capacity steganography using color images. This is outdated because not many people use BMP images on the Internet now days because of the large storage space occupied by them. Lee and Chen [29] use a variable-sized LSB insertion technique to maximize the capacity. Tolba et al [30] presents a cover-screw algorithm based on wavelet based fusion and achieve high capacity image steganography.

A lot of work has been done in the field of steganalysis using machine learning algorithms. Berg et al. [31] use decision tree, naïve bayes and neural networks for detecting hidden messsages in images. Chae and Manjunath [32] develop a steganographic approach for gray scale images using a discrete wavelet transform that is robust to low-pass filtering and lossy compression. Cox et al [33, 34] develop a information hiding technique using spread spectrum techniques that can be applied to audio, image, video and multimedia data that is robust to common signal and geometric distortions. They modify the spread spectrum method for hiding in multimedia. They also conclude that hiding should be performed in perceptually significant components of the multimedia signal. Cvejic and Seppanen [35] hide bits into higher LSB layers of audio files resulting in increased robustness against noise addition or MPEG compression. Fridrich and Goljan [36] describe self-embedding images that can recover portions of the image that have been cropped out, replaced, damaged, or otherwise tampered. Chen and Wornell [37] propose the use of Quantization Index Modulation (QIM) to achieve provably good rate-distortion-robustness performance. Lee and Chen [38] developed a robust steganographic model for images. In this model, they used ECC (Error Control Coding) to achieve robustness. It was observed that if a powerful capability of error-control code is used, then the payload of the message is less. (Payload is the amount of embedded data). This boiled down to choosing a proper ECC scheme for such a model. Cryptographic technique was also employed to make it more secure. Chang [39] suggests that to improve robustness, it may be necessary to reduce the size of embedded data and embed it multiple times under different parts of selected coefficients, where each embedding responds to a particular attack in a different way. Thus, they suggested using redundancy as a factor to achieve robustness. Smith and Comiskey [40] discovered that direct sequence might be less vulnerable to intentional removal, and wins in terms of computational complexity. Hwang [41] proposed a robust algorithm that answered the problem inherent in information hiding techniques for digital images of alignment during data extraction. This alignment problem can be solved by a search approach where a combination of coarse orientation detection, random search and gradient descent methods are employed. When used in conjunction with the Patch Track algorithm, it is possible to create an information hiding and retrieval system that is robust towards rotation, cropping and noise. Sung et al [26] proposed a model that uses animations using CFGs (Context Free Grammars), but adds cryptography to maintain the integrity and error control coding layer to account for the robustness. Basically, this means not only adding another layer but also more processing power and time consuming.

Steganalysis is the art of discovering the existence of hidden information. Jajodia, Johnson, Pfitzman, and Westfeld [4-6]were the first to work on stegananalysis. Johnson and Jajodia [15] identify some characteristics of stego-images that are created by specific image steganographic systems. To remove all possible embedded messages, an active warden may be allowed to slightly modify the data being sent between prisoners. An example of mild modification performed by the active warden is to replace the words with some close synonyms in the mail documents. If the carrier of secret messages is an image, any low-pass filters can be utilized for obviating covert communication. It is worthy to note that the primary goal of an active warden is to avoid covert communication taking place. On the other hand, in the real world a passive warden or monitor makes an attempt to catch unknown criminals from their communication to a known criminal. Opposite to the goal of steganalysis, the requirements of a steganographic system include not only imperceptibility but also undetectability by any steganalysis tool. When examined by an active warden, the hidden message should be robust against any possible modification. Budhia and Kundur [42] present steganalysis technique for digital video sequences based on the collusion attack and pattern recognition. They use the redundant information present in the temporal domain to detect cover messages in the form of Gaussian watermarks. Hesse et al [43] provide a framework for network based IDS (Intrusion Detection System) to detect steganographic communication. Li et al [44] explains an adaptive steganographic technique that escapes the contemporary steganalytic techniques. Avcibas et al [45, 46] use various image quality metrics to identify the presence of hidden information. They use analysis of variance (ANOVA) techniques to identify the quality measures. They build a classifier based on multivariate regression analysis. Wang and Wang [3] discuss contemporary tools of steganography and also discusses the commonly used steganalytic techniques.



Bit Transition	p(t)	Animation subset
0$	0	-
1$	0.08	A_{10}
00	0.08	A_5
01	0.28	$A_1 \mid A_6 \mid A_{14} \mid A_{11}$
10	0.28	$A_7 \mid A_2 \mid A_4 \mid A_9$
11	0.28	$A_8 \mid A_{12} \mid A_3 \mid A_{13}$

Bit Transition Table

Frame ID	Image
F_1	░
F_2	▒
.	.
.	.

Frame Table

Figure 1: StegAnim-Proposed steganographic model

III. Proposed Model

Our proposed data hiding model consists of hiding data using an animated set of images as cover. The proposed algorithm is illustrated in Figure 1. The embedded data is a binary string. The key is in the form of two lookup tables: Frame Table and Bit Transition Table. The creation of these lookup tables is explained in this section.

Animation, when applied to images, is defined as moving diagrams that are made up of a series of images that represent a distinct narrative unit [47]. Each of the images should be usually connected either by unity of location or time. The individual image used in an animation is a frame. Every animation follows a set of rules or grammar that governs the way the sequence is arranged. These rules can be in the form of regular expressions or can be represented by Finite Automata. Let F represent the set consisting of the frames that can be used in this animation. i.e., $\boldsymbol{F} = \{\boldsymbol{F}_1, \boldsymbol{F}_2, ..., \boldsymbol{F}_{\boldsymbol{q}}\}$. Let 'A' represent the regular expression for the animation. 'A' can contain any number of frames and it represents the order in which the image frames are played. For example $A = F_1 (F_3F_2\ F_3)^* F_3F_2$ is a valid representation of animation.

A. Construction of Frame Table

Figure 2 explains the creation of a frame table. The frame table consists of a frame id and corresponding picture image. This is a tabular representation of the set F described above. We use a table for ease of querying the data during the data hiding process. The original animation sequence is passed through a frame grabber. The function of the frame grabber is to capture each frame/image in the animation sequence. These frames are inputted to a hamming neural network to identify the unique set of frames used in this animation. The frame table consists of identification frame numbers denoting a frame.

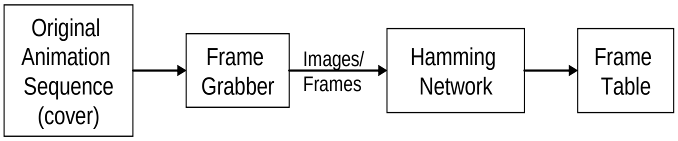


Figure 2: Frame Table creation

B. Construction of Bit Transition Table

The example illustrates that the data to be hidden is in the form of a binary string, i.e., it contains 0's and 1's. We append an end of string character, $, to the binary string. Therefore 6 bit transitions possible. Let T represent the set of possible bit transitions. For a binary string, T = [10 11, 0$, 1$]. For a general case, where the string is of base n, the number of transitions possible is 2^n+n. Let p(t) represent the probability of each transition. Let N be the length of the string.

$$\boldsymbol{p(t)} = \frac{\textbf{No. of t}}{\textbf{N}} \quad \forall \boldsymbol{t} \in \boldsymbol{T} .$$

Now, create m distinct subsets of the animation using the same grammar that was used to generate the original cover animation. Please note that m >= N. i.e., create m disjoint subsets of animation 'A'. A = A_1U A_2U…U $A_{m.}$ Each disjoint animation subset is made up of a sequence of frames. For example, A_1=F_1 F_3 F_2 F_5 F_4. Allocate to each transition a rounded value of (m x p(t)) number of animation subsets.

The construction of the bit transition table can be understood with the help of the following example. Let us consider the bit string to be embedded as 11100101101101. Append a '$' at the end of the string, the bit string becomes 11100101101101. Now calculate the probability by the method described above. The probability values are described in Figure 1. Let us assume that we have created m=14 subsets of animation. Each transition is allocated a rounded (14 * p(t)) number of animation subsets.

C. Embedding Secret Data

The goal of this process is to encode data into predefined animations. In our model, the sender encodes data and recreates the stego-animation in the same manner as the cover-animation. The steps denoted in Figure 1 are described as follows:

1. The secret data (to be hidden) is fed to the encoder – one bit at a time.

2. The encoder looks up the bit sequence in the Bit Transition table and chooses the corresponding animation subset. If there is more than one animation subset associated with a transition, the encoder can choose one of the subset in a random manner. The animation subsets are passed on to the next module Animator.

3. The Animator receives each animation subset, a sequence of frame IDs, from the Encoder. It concatenates the complete frame sequence in the order in which it was received from the encoder.

4. The Animator looks up the corresponding picture image from the frame table for each frame it obtains.

5. The Animator creates an animation by concatenating all the picture images in a sequence. This animation obtained is the stego-animation that contains the embedded data concealed in the form of an encoding.

D. Extraction

The following steps illustrate the process of recovering the hidden data in the animation. It should be noted that the bit transition table and the frame table have to be sent to the receiver on a separate channel. The frame table and bit transition table are like keys for the extraction of the hidden data. Without the key, the extractor will not be able to obtain any meaningful data from the animation.

6. The stego-animation is passed to the "pre-processing" module frame by frame. The pre-processing consists of grabbing each frame from the animation and converting it into the matrix format required by the hamming network in the image recognition module.

7. The image recognition module is made up of a hamming neural network. The number of input nodes in the hamming networks is equal to the total number of pixels of each image. The number of output nodes is the same as the number of distinct frames as recorded in the frame table.

8. The image recognition module identifies each frame supplied to the module by reverse-lookup from the frame table. It maps each picture image to the frame id.

9. The image recognition module supplies the decoder with the sequence of frame ids obtained from the frame table in the given order.

10. The decoder forms the disjoint animation subsets from the frame sequence and then extracts the bit transitions by searching the Bit-transition table from the frame ids.

11. The data is finally outputted by concatenating all the bit sequences.

IV. Robustness

Robustness is the ability of the embedded message to be recovered even after the stego-object has been modified so that the minimum fidelity of the stego-object is maintained with respect to the cover-object. To measure the robustness, the object is subjected to some of the object-processing operations. The following procedure is used to measure the robustness of a steganographic algorithm. The robustness of embedded data is thus described in terms of a low rate of bit errors.

Hopper *et al* [48] define robust steganography to be a model between the sender and the adversary in the steganography in which the adversary is allowed to do some pre-defined limited alterations to the stego-object. Craver [7] defines a robust channel as one whose content cannot be altered without making unreasonably drastic changes to the stego-object (i.e., requiring a malicious, instead of an active warden). Also, the most robust algorithm is the one that can extract the original embedded data with a certain level of change to the stego-object. Such a technique is the current need for the science of Steganography.

Steganography algorithms provide stealth and security to information. The degree of stealth and security is hard to measure. One way to judge the strength of a steganographic algorithm is to imagine different attacks and then assess whether the algorithm can successfully withstand them. This approach is far from perfect, but it is the best available. Anticipating all possible attacks is nearly impossible. An approach for measuring the robustness is proposed below:

1. Create a repertoire of common attacks to be performed on each stego-object. (Attack$_1$,Attack$_2$… Attack$_n$)
2. Obtain a list of commonly available contemporary steganographic software that is to be compared for robustness. (Software$_1$, Software$_2$, … Software$_m$)
3. Apply each of the n attacks on all of the stego-object created by m software and do the following for each of the extracted data obtained.
4. Take the XoR of the original embedded bits and extracted bits, where XoR stands for Exclusive OR.
5. Use the following formula to calculate values of the robustness of the bits for the corresponding attack. A = (the number of 1s in the XoRed result / the total number of bits) x 100 % and RF (Robustness Factor) is measured by

$$\text{RF of the } i^{th} \text{ software} = \sum_{j=1}^{n} A_{ij} .$$

Here, n stands for the total number of attacks.

Represent the obtained values in the form of Figure 3.

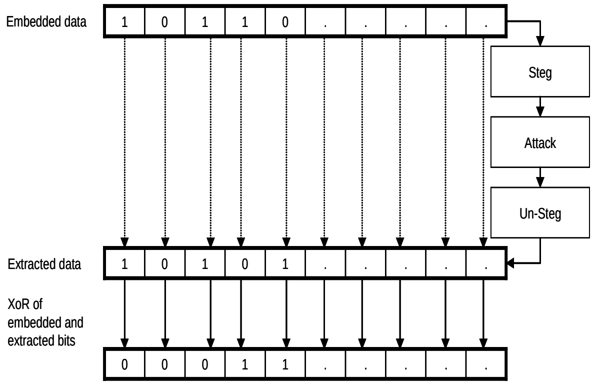


Figure 3: Explaining the measurement of robustness factor

V. Experiments and Simulation Results

A. *Software*

The proposed algorithm, documented in Figure 1, was compared with the following software in terms of robustness: BMP Secrets [49], Contraband 9g [50], Hide in Picture [51], ImageHide [52], InfoStego [53], SecurEngine [54], Steghide [55], Stools [56], ThirdEye [57]. All these software were chosen to perform steganography in BMP images. They are all freely available tools. A detailed description of the software is available in the appendix.

B. *Test Images Used*

Table 1 lists test images used in this experiment.

Table 1: Test Images used in the experiments

1

2

3

4

5

6

C. *Attacks performed*

A 100 bit input file was used to hide inside the carrier file. The following attacks were carried out on the stego-image.

1) *Resizing*

The size of an image is related to many factors. Primarily image size is measured in pixels. The size of an image can be assumed to be denoted by the width and height of the image measured in pixels. This can make the image larger or smaller. The original size of the image was 640 x 480 pixels.

Table 2: A result summary on resizing (downsizing = 90%)

Software	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Average
BMP Secrets	87	64	65	82	79	73	75
Contraband	75	62	61	60	60	60	63
Contraband-Hell	75	43	45	47	39	49	49.67
Hide In Picture	12	16	19	29	24	31	21.83
ImageHide	8	31	6	8	9	14	12.67
InfoStego	31	9	13	8	6	14	13.50
Securengine	12	29	9	13	18	6	14.50
Steghide	12	32	25	6	14	10	16.50
Stools4	5	6	8	12	19	3	8.83
Third Eye	21	33	23	12	70	6	27.50
Proposed Model	100	100	100	100	100	100	100

TABLE 3: A RESULT SUMMARY ON RESIZING (4 TYPES OF RESIZING)

Software	80% x 80% (512 x 384)	90% x 90% (576 x 432)	110% x 110% (704 x 528)	120% x 120% (768 x 576)	Mean
BMP Secrets	67.83	75.00	84.00	59.45	71.57
Contraband	59.50	63.00	74.00	23.33	54.96
Contraband-Hell	46.83	49.67	60.00	47.33	50.96
Hide In Picture	9.00	21.83	33.00	3.00	16.71
ImageHide	9.50	12.67	36.67	7.00	16.46
InfoStego	7.83	13.50	53.33	9.33	21.00
Securengine	1.50	14.50	74.76	1.00	22.94
Steghide	4.17	16.50	37.50	3.33	15.38
Stools4	1.67	8.83	49.00	2.00	15.38
Third Eye	10.67	27.50	45.50	5.00	22.17
Proposed Model	100.00	100.00	95.50	83.33	94.71

The original image was downsized and upsized and checked for performance. Table 2 documents a result on RF (%) applied to 6 different images in a case of 90% reduction. Each number listed in Table 2 indicates an average of 100 attacks (i.e., resizing). The number of the last column of Table 2 is an average of the 6 different images of Table 1. Table 3 summarizes a condensed result on such RF scores, considering 4 different types of resizing (80%, 90%, 110% and 120%). The number listed in the last column of Table 3 indicates an average of each software's RF, where the average is measured under 2400 different combinations [= 100 (vulnerable points) x 6 (images) x 4 (types of resizing)]. A finding from the two tables is that the proposed algorithm outperforms the other software when downsized. See the last row of Table 3. However, a similar result cannot be observed when it is upsized. The simulation result is due to the fact that the pre-processing module (Step 7) in the proposed algorithm checks to examine whether the size is same as the original one stored in the frame table. See Figure 1. If it is not, a resizing algorithm is first applied to the conversion of the original size as stored in the database.

Since the resizing algorithm incorporated in our model is important in terms of robustness, we need to descript it in detail. The standard approach for resizing robustness is called "bicubic interpolation," and it estimates the colour at a pixel in the destination image by an average of 16 pixels surrounding the closest corresponding pixel in the image. There are two methods in common usage for interpolating the 4x4 pixel, cubic B-Spline and a cubic interpolation function, the B-spline approach will be discussed here. The diagram below introduces the conventions and nomenclature used in the equations. It is desirable to determine the colour of every point (i',j') in the final (destination) image. There is a linear scaling relationship between the two images, in general a point (i',j') corresponds to a non integer position in the original (source) image [58]. This is position is given by x = iw'/w and y = j h' / h. The nearest pixel coordinate (i,j) is the integer part of x and y, dx and dy, in the diagram is the difference between those. That is, dx = x – I and dy = y - j.

Unfortunately, the other software examined in this study does not have this type of checking capability and hence this study finds the simulation results documented in Tables 2 and 3.

2) *Sharpening and Blurring*

Filters alter each pixel's color based on its current color and the colors of any adjacent pixels. The results can vary from a minor adjustment of a single characteristic to a total alteration of an image. The Sharpen effects are filters that produce the opposite effect of the Blur commands by increasing the contrast between adjacent pixels where there are significant color contrasts, usually at the edges of objects. They lighten the light pixels and darken the dark pixels. Sharpen More applies the Sharpen effect with more intensity. These effects can be used to fix photographs that are slightly out of focus.

The Blur effects are filters that smooth transitions and decrease contrast by averaging the pixels next to hard edges of defined lines and areas where there are significant color transitions [59]. The Gaussian blur is one kind of blur filter that uses a mathematical formula to create the effect of looking through an out-of-focus lens [60].

These effects are measured in the length of the matrices used for the filters. They are usually 3x3, 5x5 or 7x7 etc. So, a filter aperture of 3 means that the matrix is a 3 x 3 matrix. The maximum filter accepted for the images in our experiments is 31 x 31.

The original image was sharpened and blurred to examine the performance of different software. Table 4 documents such a result on RF (%) applied to 6 different images in a case of filter aperture = 10. Each number listed in Table 4 indicates an average of 100 attacks (i.e., sharpening/blurring). The number of the last column of Table 4 is its average applied to the 6 different images. Table 5 summarizes a condensed result on such RF scores, considering 5 different filter apertures (FA = 3, 10, 17, 24, and 31). The number listed in the last column of Table 5 indicates an average of each software's RF, where the average is measured under 3000 different combinations [= 100 (vulnerable points) x 6 (images) x 5 (filter apertures)]. A finding from the two tables is that the proposed algorithm outperforms the other software in the sharping changes. See the results in the first three columns (FA=3, 10 and 17) of Table 5. However, an opposite result was observed when the image was more blurry (FA = 24 and 31).

TABLE 4: A RESULT SUMMARY ON SHARPENING AND BLURRING (FILTER APERTURE = 10)

Software	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Average
BMP Secrets	84.00	74.00	86.00	77.00	83.00	78.00	80.33
Contraband	70.00	69.00	88.00	72.00	70.00	75.00	74.00
Contraband-Hell	66.00	68.00	73.00	67.00	75.00	71.00	70.00
Hide In Picture	24.00	0.00	0.00	0.00	0.00	0.00	4.00
ImageHide	1.00	4.00	6.00	31.00	4.00	11.00	9.50
InfoStego	0.00	0.00	1.00	1.00	5.00	7.00	2.33
Securengine	4.00	0.00	4.00	0.00	0.00	14.00	3.67
Steghide	0.00	9.00	15.00	18.00	12.00	0.00	9.00
Stools4	0.00	0.00	0.00	0.00	3.00	0.00	0.50
Third Eye	3.00	4.00	0.00	4.00	0.00	0.00	1.83
Proposed Model	83.00	74.00	94.00	95.00	92.00	99.00	89.50

TABLE 5: A RESULT SUMMARY ON SHARPENING AND BLURRING (FIVE FILTER APERTURES)

Software	FA = 3 Sharpest, Least Blur	FA = 10	FA = 17	FA = 24	FA = 31 Most Blur, Least Sharp	Mean
BMP Secrets	85.50	80.33	75.50	70.00	23.00	66.87
Contraband	74.17	74.00	70.34	55.00	2.30	55.16
Contraband-Hell	70.83	70.00	65.67	34.00	1.20	48.34
Hide In Picture	5.67	4.00	3.67	0.00	0.00	2.67
ImageHide	11.17	9.50	3.33	0.00	0.00	4.80
InfoStego	9.50	2.33	2.00	0.00	0.00	2.77
Securengine	4.67	3.67	1.50	0.00	0.00	1.97
Steghide	12.00	9.00	4.00	2.00	0.00	5.40
Stools4	0.67	0.50	0.00	0.00	0.00	0.23
Third Eye	2.50	1.83	0.00	0.00	0.00	0.87
Proposed Model	100.00	89.50	75.50	0.00	0.00	53.00

The original image was sharpened and blurred to examine the performance of different software. Table 4 documents such a result on RF (%) applied to 6 different images in a case of filter aperture = 10. Each number listed in Table 4 indicates an average of 100 attacks (i.e., sharpening/blurring). The number of the last column of Table 4 is its average applied to the 6 different images. Table 5 summarizes a condensed result on such RF scores, considering 5 different filter apertures (FA = 3, 10, 17, 24, and 31). The number listed in the last column of Table 5 indicates an average of each software's RF, where the average is measured under 3000 different combinations [= 100 (vulnerable points) x 6 (images) x 5 (filter apertures)]. A finding from the two tables is that the proposed algorithm outperforms the other software in the sharping changes. See the results in the first three columns (FA=3, 10 and 17) of Table 5. However, an opposite result was observed when the image was more blurry (FA = 24 and 31).

TABLE 6: A RESULT SUMMARY ON RANDOM NOISES (20%)

Software	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Average
BMP Secrets	12	21	32	73	34	12	31
Contraband	25	51	39	59	41	63	46
Contraband-Hell	9	5	72	59	43	67	43
Hide In Picture	74	49	37	13	36	68	46
ImageHide	77	42	79	54	14	58	54
InfoStego	54	5	70	55	52	29	44
Securengine	39	23	65	51	58	62	50
Steghide	42	27	14	50	43	11	31
Stools4	25	72	32	29	33	36	38
Third Eye	56	41	7	51	78	43	46
Proposed Model	6	33	82	46	90	31	48

3) Random Noise

Random noise was inserted in the form of random dots using the RAND( ) function of excel. A pseudo random number was generated between 0 and 307199 and the image was deformed in each corresponding location. This attack was done to examine how the random noises influence on the performance of software.

TABLE 7: A RESULT SUMMARY ON RANDOM NOISES (FIVE DIFFERENT NOISES)

Software	20%	40%	50%	60%	80%	Mean
BMP Secrets	31	34	18	16	6	21.00
Contraband	46	46	27	14	5	27.60
Contraband-Hell	43	47	32	15	4	28.20
Hide In Picture	46	35	28	10	6	25.00
ImageHide	54	39	19	17	6	27.00
InfoStego	44	32	21	12	5	22.80
Securengine	50	34	22	13	6	25.00
Steghide	31	51	32	10	6	26.00
Stools4	38	38	19	12	5	22.40
Third Eye	46	47	33	14	4	28.80
Proposed Model	48	52	41	16	4	32.20

Table 6 documents such a result on RF (%) applied to 6 different images when 20% of the total bits are generated randomly. Each number listed in Table 6 indicates an average of 100 attacks (i.e., adding of random noises). The number of the last column of Table 6 is its average applied to the 6 different images. Table 7 summarizes a condensed result on such RF scores, considering 5 different percentages of random noises (20%, 40%, 50%, 60%, 80%). The number listed in the last column of Table 7 indicates an average of each software's RF, where the average is measured under 3000 different combinations [= 100 (vulnerable points) x 6 (images) x 5 (percentages)]. A finding from the two tables is that the proposed algorithm slightly outperforms the other software in the addition of random noises.

VI. CONCLUSIONS

This study proposed a novel steganographic algorithm based on animations. The basic rationale for using animations as cover was justified in the robustness factor achieved using this proposed algorithm. As mentioned previously, it is widely known that many conventional steganography are not robust enough because they are susceptible to destroy an embedded data in the stego-object when simple manipulations to the stego-object are performed.

In contrast to cryptography, where the "enemy" is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by a cryptosystem, the goal of steganography is to hide messages inside other "harmless" messages in a way that does not allow any "enemy" to even detect that there is a second secret message present. Most good cryptographic tools also produce data that looks almost perfectly random. It can be said that they are trying to hide the information by disguising it as random noise. On the other hand, many steganographic algorithms are not trivial to break even after knowing that there is hidden data to find. Cryptographic authentication protocols cannot solve all the issues related to authentication. Cryptographic authentication deals with authenticating the sender of the message over insecure channels. However, once the message (image) is decrypted, the image is unprotected and can be copied and further distributed. This led us to believe that a data hiding model has to be designed that neither uses cryptography nor error control coding for robustness.

The metric of any steganographic algorithm is measured by its capacity, security, and robustness. Our algorithm lacks in the factor that it takes more animation frames to increase the capacity. This is a shortcoming of our approach. One way to overcome this would be to use the embedding methodologies to hide data within each frame. Future work is needed in this direction to carefully combine embedding into each frame and still keep the high robustness factor that has been achieved. Also, our approach is limited in the fact that it is more suitable to live animation plays, i.e., a live telecast of images would be the most appropriate. An extension of this model to the recorded video formats is an important future research work.

We proposed a metric for robustness of a steganographic algorithm. This technique is more resistant to attacks with respect to other chosen freeware software. We employ a set of known attacks on the stego-object to document that the proposed method is more robust to attacks from the adversaries. The experiments related to (a) resizing, (b) sharping/blurring and (c) random noises confirm that the proposed algorithm is more robust than the other 10 commonly available steganographic software.

Finally, it is hoped that this study makes a small contribution on the development of steganography. We look forward to seeing further research extensions, as discussed in this article.

VII. Appendix

A. *BMP Secrets*

This program uses cryptographic and compression algorithms in addition to using bit-insertion for hiding the data. It is difficult to decode information without password than to decode simply encoded data. The simplest method to hide information in BMP files is to put it in the most insignificant bits of each pixel. When you hide small files in large images, you can not distinguish the original from the picture with data. However, using this method, you can not replace a lot of information, because the human eye will distinguish changes. The program uses original steganography method developed by Parallel Worlds that allows up to 65% embedding capacity of the true-color BMP file with your data. It is vulnerable when converted to lossy formats. The hidden information is lost when the image is subjected to any image processing operation.

B. *Contraband 9g*

Contraband encodes data and embeds BMP file without changing change size or format of the BMP. The 4-digit code is used to play around with bits in such a way that it's very hard to notice whether or not data has been concealed in a picture.

C. *Hide In Picture*

HIP (Hide In Picture) conceals files inside bitmaps, using a password. It is not possible to get the hidden file back (or to even be sure there is a file in the picture) without the correct password.

D. *ImageHide*

ImageHide embeds text in images. Encryption and decryption of data is supported. There is no increase in image size. Image looks the same to normal paint packages. Load and save to files. Get past all the mail sniffers.

E. *InfoStego*

Info Stego is a tool that protects private information, communication and legal copyright using information watermark and data encryption technology by embedding data into any other picture, sound, video etc.

F. *Securengine*

SecurEngine hides confidential documents in jpg, bmp, wav and txt files carrier. It also supports encryption algorithms like Blowfish, Gost, Vernam, Cast256, and Mars. It can build a Self Decrypting Archives to transfer some confidential files over internet without the need of SecurEngine to decrypt them.

G. *Steghide*

Steghide is a steganography program that is able to hide data in various kinds of image- and audio-files. The respectively sample-frequencies are not changed thus making the embedding resistant against first-order statistical tests.

H. *Stools*

Stools is perrhaps the most widely recognized steganography tool available today. S-Tools hides the secret message within the cover file via random available bits. These available bits are determined through the use of a pseudo-random number generator. This non-linear insertion makes the presence and extraction of secret messages more difficult. S-Tools makes use of the concept of least significant bit (LSB). It takes the image palette, finds the least significant bit of each byte, and attempts to reconstruct the cover file inserting the bits of the secret message into these LSBs. As mentioned above, S-Tools will insert the bits in a non-linear fashion. S-Tools can embed data in the LSBs of audio files (.wav), in the LSBs of the RGB color values in graphic files (.bmp), or in free sectors of disks. Additional encryption methods like DES and IDEA are provided for added security.

I. *ThirdEye*

Most computer data has to be 100% accurate in order to function correctly, but digitally sampled data (data that is analog in nature but represented in digital form in some way for the purpose of storing on a computer) need not be. Examples of sampled data are images, sounds, video, etc. By making subtle alterations to sampled data it is possible to conceal information whilst retaining nearly all the content of the original sample. Images on a computer are digital approximations of analog data. To represent images digitally, they are sampled- i.e. they are approximately represented as numbers! Since there is an approximation anyway, changing a number here and there will not affect the image much. This is the principle behind the working of ThirdEye.

REFERENCES

[1] H. Berghel and L. O'Gorman, "Protecting ownership rights through digital watermarks," IEEE Comput., vol. 29, pp. 101-103, 1996.

[2] N. Memon and P. Wong, "Protecting digital media content," Communications of the ACM, vol. 41, pp. 34-43, 1998.

[3] H. Wang and S. Wang, "Cyber Warfare: Steganography vs. Steganalysis," in Communications of the ACM, vol. 47, 2004, pp. 76-82.

[4] B. Pfitzmann, "Information Hiding Terminology," presented at Proceedings of First Workshop of Information Hiding, Cambridge, UK, 1996.

[5] G. J. Simmons, "The Prisoner's Problem and the Subliminal Channel," presented at Proceedings of CRYPTO '83, 1984.

[6] S. Katzenbeisser and F. A. P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking: Artech House, 2000.

[7] S. Craver, "On Public-key Steganography in the Presence of an Active Warden," presented at Proceedings of 2nd International Workshop on Information Hiding, Portland, Oregon, USA, 1998.

[8] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," IBM Systems Journal, vol. 35, pp. 313-336, 1996.

[9] VenkatramanS, A. Abraham, and M. Paprzycki, "Significance of steganography on data security," presented at Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on, 2004.

[10] B. Chen and G. W. Wornell, "Quantization Index Modulation Methods for Digital Watermarking and Information Embedding of Multimedia," The Journal of VLSI Signal Processing, vol. 27, pp. 7-33, 2001.

[11] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding - a survey," Proceedings of the IEEE, vol. 87, pp. 1062-1078, 1999.

[12] R. Chandramouli and N. Memon, "Analysis of LSB based image steganography techniques," presented at IEEE International Conference on Image Processing (ICIP), Oct 7-10 2001, Thessaloniki, 2001.

[13] E. Kawaguchi and R. O. Eason, "Principle and applications of BPCS-steganography," Proceedings of SPIE - The International Society for Optical Engineering, vol. 3528, pp. 464-473, 1999.

[14] J. Fridrich, "A New Steganographic Method for Palette-Based Images," presented at Final Program and Proceedings: IS and T's 52nd Annual Conference, Apr 25-28 1999, Savannah, GA, United States, 1999.

[15] N. F. Johnson and S. Jajodia, "Steganalysis: The Investigation of Hidden Information," IEEE, pp. 113-116, 1998.

[16] M. T. I. Sandford, J. N. Bradley, and T. G. Handel, "Data embedding method," presented at Integration Issues in Large Commercial Media Delivery Systems, Oct 23-24 1995, Philadelphia, PA, USA, 1996.

[17] H. Noda, T. Furuta, M. Niimi, and E. Kawaguchi, "Video steganography based on bit-plane decomposition of wavelet transformed video," presented at Security, Steganography, and Watermaking of Multimedia Contents VI, Jan 19-22 2004, San Jose, CA, United States, 2004.

[18] J. J. Chae and B. S. Manjunath, "Data hiding in video," IEEE International Conference on Image Processing, vol. 1, pp. 311-315, 1999.

[19] A. Westfeld and G. Wolf, "Steganography in a video conferencing system," Lecture Notes in Computer Science, 1525 ed, 1998, pp. 32.

[20] M. George, J.-Y. Chouinard, and N. Georganas, "Digital watermarking of images and video using direct sequence spread spectrum techniques," Canadian Conference on Electrical and Computer Engineering, vol. 1, pp. 116-121, 1999.

[21] D. Inoue and T. Matsumoto, "A scheme of Standard MIDI Files steganography and its evaluation," presented at Security and Watermarking of Multimedia Contents IV, Jan 21-24 2002, San Jose, CA, United States, 2002.

[22] D. Inoue, M. Suzuki, and T. Matsumoto, "Detection-Resistant Steganography for Standard MIDI Files," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E86-A, pp. 2099-2106, 2003.

[23] R. Tachibana, "Two-dimensional audio watermark for MPEG AAC audio," presented at Security, Steganography, and Watermaking of Multimedia Contents VI, Jan 19-22 2004, San Jose, CA, United States, 2004.

[24] R. Tachibana, S. Shimizu, S. Kobayashi, and T. Nakamura, "An audio watermarking method using a two-dimensional pseudo-random array," Signal Processing, vol. 82, pp. 1455-1469, 2002.

[25] R. Ratan and C. E. Veni Madhavan, "Steganography based information security," IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India), vol. 19, pp. 213-219, 2002.

[26] A. H. Sung, G. R. Tadiparthi, and S. Mukkamala, "Defeating the Current Steganalysis Techniques (Robust Steganography)," presented at ITCC, 2004.

[27] R. Chandramouli and N. D. Memon, "Steganography Capacity: A Steganalysis Perspective," presented at Security and Watermarking of Multimedia Contents V, Jan 21-24 2003, Santa Clara, CA, United States, 2003.

[28] K. Nozaki, M. Niimi, R. O. Eason, and E. Kawaguchi, "Large capacity steganography using color BMP images," presented at Proceedings of the 1998 3rd Asian Conference on Computer Vision, ACCV'98, Hong Kong, Hong Kong, 1998.

[29] Y. K. Lee and L. H. Chen, "High capacity image steganographic model," IEE Proceedings: Vision, Image and Signal Processing, vol. 147, pp. 288-295, 2000.

[30] M. Fahmy Tolba, M. Al-Said Ghonemy, I. A.-H. Taha, and A. S. Khalifa, "High capacity image steganography using wavelet-based fusion," presented at Proceedings - ISCC 2004, Ninth International Symposium on Computers and Communications, Jun 28-Jul 1 2004, Alexandria, Egypt, 2004.

[31] G. Berg, I. Davidson, M.-Y. Duan, and G. Paul, "Searching for hidden messages: automatic detection of steganography," presented at Fifteenth Annual Conference on Innovative Application of Artificial Intelligence, Acapulco, Mexico, 2003.

[32] J. J. Chae and B. S. Manjunath, "A robust embedded data from wavelet coefficients," presented at Storage and Retrieval for Image and Video Databases VI, Jan 28-30 1998, San Jose, CA, United States, 1998.

[33] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," IEEE Transactions on Image Processing, vol. 6, pp. 1673-1687, 1997.

[34] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure, imperceptable yet perceptually salient, spread spectrum watermark for multimedia," presented at Proceedings of the 1996 Southcon Conference, Jun 25-27 1996, Orlando, FL, USA, 1996.

[35] N. Cvejic and T. Seppanen, "Increasing robustness of LSB audio steganography using a novel embedding method," presented at International Conference on Information Technology: Coding Computing, ITCC 2004, Apr 5-7 2004, Las Vegas, NV, United States, 2004.

[36] J. Fridrich and M. Goljan, "Images with self-correcting capabilities," IEEE International Conference on Image Processing, vol. 3, pp. 792-796, 1999.

[37] B. Chen and G. W. Wornell, "Provably robust digital watermarking," Proceedings of SPIE - The International Society for Optical Engineering, vol. 3845, pp. 43-54, 1999.

[38] Y. K. Lee and L. H. Chen, A Secure Robust Image Steganographic Model, 2000.

[39] L. W. Chang, "Issues in Information Hiding Transform Techniques," Naval Research Lab, Center for Computer High Assurance Systems, Washington DC A849104, May 2002 2002.

[40] J. R. Smith and B. O. Comiskey, "Modulation and Information Hiding in Images," presented at Proceedings of First International Workshop on Information Hiding, 1996.

[41] R. Hwang, "A Robust Algorithm for Information Hiding in Digital Pictures," in M.Eng Thesis. Cambridge, MA: MIT, 1999.

[42] U. Budhia and D. Kundur, "Digital video steganalysis exploiting collusion sensitivity," presented at Sensors, and Command, Control, Communications, and Intelligence (C31) Technologies for Homeland Security and Homeland Defense III, Apr 12-16 2004, Orlando, FL, United States, 2004.

[43] D. Hesse, J. Dittmann, and A. Lang, "Network based intrusion detection to detect steganographic communication channels - on the example of images," presented at Euromicro Conference, 2004. Proceedings. 30th, 2004.

[44] G. Li, N. Memon, and R. Chandramouli, "Adaptive steganography," presented at Security and Watermarking of Multimedia Contents IV, Jan 21-24 2002, San Jose, CA, United States, 2002.

[45] I. Avcibas, N. Memon, and B. Sankur, "Steganalysis using image quality metrics," IEEE Transactions on Image Processing, vol. 12, pp. 221-229, 2003.

[46] I. Avcibas, N. Memon, and B. Sankur, "Steganalysis based on image quality metrics," presented at 2001 IEEE fourth Workshop on Multimedia Signal Processing, Oct 3-5 2001, Cannes, France, 2001.

[47] animation, "Computer Desktop Encyclopedia," vol. 2005, 2005.

[48] N. J. Hopper, J. Langford, and L. V. Ahn, "Provably Secure Steganography," presented at Advances in Cryptology: CRYPTO, 2002.

[49] V. Chekh and A. Grushetsky, "Secret Information in BMP images- BMP Secrets," http:www.pworlds.com/products/secrets.html, 2001.

[50] J. B. Thyssen and H. Zimmerman, "Contraband 9g," http://www.stegoarchive.com, 1999.

[51] D. T. Figueiredo, "Hide In Picture (HIP) Version 2.0," http://www.brasil.terravista.pt/Jenipabu/2571/e_hip.htm, 2001.

[52] Dancemammal.com, "ImageHide," http://prem-01.portlandpremium.co.uk/p128/imagehide.htm.

[53] A. Labs, "InfoStego Personal Edition," http://www.antiy.net/infostego/index.htm, 2001.

[54] A. Pinet, "SecurEngine 4.0," http://securengine.isecurelabs.com/, 2003.

[55] S. Hetzl, "Steghide," http://steghide.sourceforge.net/, 2003.

[56] A. Brown, "Steganography tools for Windows - Stools 4.00," ftp://ftp.demon.net/pub/mirrors/crypto/idea/code/s-tools4.zip, 2002.

[57] S. K. Popuri, "The Third Eye Version 1.0," http://www.webKclub.com, 2002.

[58] P. Bourke, "Bicubic Interpolation for Image Scaling," http://astronomy.swin.edu.au/~pbourke/colour/bicubic/, 2001.

[59] J. S. Inc., "Paint Shop Pro," http://jasc.com, 2003.

[60] Webmonkey, "The Web Developer's Resource - Glossary," http://hotwired.lycos.com/webmonkey/glossary/gaussian.html, 2003.

Toshiyuki Sueyoshi is a full professor for Department of Management at New Mexico Institute of Mining and Technology in USA. He is also a visiting full professor for Department of Industrial and Information Management at National Cheng Kung University in Taiwan. He obtained his Ph.D. from University of Texas at Austin. He has published more than 150 articles in international journals.

Gopalakrishna Reddy Tadiparthi received the B.E. degree in Computer Science from University of Madras, India, in 1999 and worked as a Network Engineer at Satyam Infoway Limited (SIFY), India till 2002. He received the M.S. degree in Computer Science from New Mexico Institute of Mining and Technology in 2003 and is currently a Ph.D. candidate in the Department of Computer Science.