Fuzzy Transformation Coding/Decoding to improve capacity of LSB Image Steganography

Subin Khullar[1], Prashant Kumar[2], Rohit Wadhwa[3], Jasmeet Singh[4]
[1234]Deptt. of Information Technology, Bharati Vidyapeeth's College of Engineering, A-4 Paschim Vihar, New Delhi-63
E-mail : [1]subin.khullar@gmail.com, [2]prashant4nov@gmail.com, [3]rohit.wadhwa10787@gmail.com, [4]jasmeet.bvcoe@gmail.com

ABSTRACT
LSB Image Steganography is a very simple Steganography procedure in which only one bit (LSB) of a pixel value (of all color components-R,G,B) are used for hiding secret image. Each secret image pixel needs 8 cover image pixels. In this project we aim to improve the embedding capacity of this algorithm by compressing the secret image by using Fuzzy Transformation coding before embedding it & then applying inverse Fuzzy Transform on the extracted image to decompress it. A compression factor of 7.11 was achieved which increased the embedding efficiency to 88.88% i.e. each cover image pixel could now hide 7.11/8 secret image pixels.

KEYWORDS
LSB Image Steganography, Fuzzy Transformation, Fuzzy Sets, Cosinus Fuzzy Partitions, Image concealment/hiding, Secret Image, Cover Image, Stego Image.

INTRODUCTION

Steganography is the hiding of a piece of data which may be in the form of an image, a text file, a binary/octet file or an audio file in the pixel values of a cover image by the use of a certain hiding algorithm. We in this project hide a color image (secret image) in another color image (cover image) using modified LSB algorithm. Fuzzy Transformation (FT) is a coding/decoding algorithm which is used to compress or decompress images from Fuzzy partition sets of blocks of pixel values. This is used to compress the secret image before embedding it into the cover image so as to increase the image hiding. FT algorithm takes an image divided into 8x8 blocks as input for compression and reduces the size of each block to 3x3, the number of blocks remaining the same. Thus, 64 pixels are represented as 9 pixels giving compression ratio as 7.11. After extraction of the image during image retrieval, the extracted image is decompressed to give back the secret image of original size. The decompressed image is blurred a bit due to the approximations made during the algorithm. It can be sharpened by the use of an appropriate image sharpening or high pass filter.

STEGANOGRAPHY & FUZZY TRANSFORMATION:

i. Steganography

It is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient,

suspects the existence of the message, a form of security through obscurity. The word ***steganography*** is of Greek origin and means "concealed writing". Generally, messages will appear to be something else: images, articles, shopping lists, or some other *cover-text* and, classically, the hidden message may be in invisible ink between the visible lines of a private letter.
The advantage of steganography, over cryptography alone, is that messages do not attract attention to themselves. Plainly visible encrypted messages—no matter how unbreakable—will arouse suspicion, and may in themselves be incriminating in countries where encryption is illegal. So, whereas cryptography protects the contents of a message, steganography can be said to protect both messages and communicating parties.
Steganography includes the concealment of information within computer files. In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocol. Media files are ideal for steganographic transmission because of their large size. As a simple example, a sender might start with an innocuous image file and adjust the color of every 100th pixel to correspond to a letter in the alphabet, a change so subtle that someone not specifically looking for it is unlikely to notice it.

ii. Fuzzy Transformation:

Fuzzy Sets are sets whose elements have degrees of membership. Fuzzy sets were introduced by Lotfi A. Zadeh (1965) as an extension of the classical notion of set. In classical set theory, the membership of elements in a set is assessed in binary terms according to the ***bivalent condition*** - an element either belongs or does not belong to the set. By contrast, fuzzy set theory permits the gradual assessment of the membership of elements in a set; this is described with the aid of a membership function valued in the real unit interval [0, 1].
Let $m_1 < m_2 < ... < m_p$ be p fixed nodes of the universe, such that $m_1 = a$ and $m_p = b$, and $p \geq 3$. The set of the p fuzzy subsets $A_1, A_2, ..., A_p$, identified with their membership functions $\mu_{A1}(x), \mu_{A2}(x), ..., \mu_{Ap}(x)$ defined on the universe, form a strong uniform fuzzy partition of the universe, if they fulfill the necessary conditions of Strong Uniform Fuzzy Partitions for $k = 1, \ldots, p$. We have used ***Cosinus Fuzzy Partition*** in our project.

After partitioning the Fuzzy sets using the Fuzzy membership functions, we apply the Fuzzy Transform or Inverse Transform as shown in later in algorithm sub-section of this paper.

APPLICATIONS OF STEGANOGRAPHY:

Steganography is applicable to, but not limited to, the following areas:

1. Confidential communication and secret data storing
2. Protection of data alteration
3. Access control system for digital content distribution
4. Media Database systems

The area differs in what feature of the steganography is utilized in each system.

ALGORITHM:

i. Fuzzy Transformation Pseudo code:

Input: Data Image P(i,j) an N x M grayscale or color(repeat all steps for each RGB component of pixel) image.

Output: Compressed Data Image R(k,l) an n x m image.

Algorithm:

Step 1: Find Fuzzy Relation R: (i, j)∈{1,…,N} ×{1,…,M} → R(i,j) ∈ [0,1] where

R (i,j) = P(i,j)/255 for 256 gray levels.

Step 2: Divide R (i,j) into blocks of 8 x 8 pixels and call it $R_B(i,j)$

Step 3: Define Uniform Fuzzy Partitions A of (1,…, N) and B of (1,…, M).

Step 4: Let R_B the corresponding fuzzy relation (submatrix of R) of sizes N(B)×M(B), coded to a block F_B of sizes n(B) × m(B) (with 3 ≤ n(B) « N(B), 3 ≤ m(B) « M(B)) via the discrete FTR [F_{kl}^B] defined as:

$$F_{kl}^{B}=\frac{\sum_{j=1}^{M(B)}\sum_{i=1}^{N(B)}R_B(i,j)A_k(i)B_l(j)}{\sum_{j=1}^{M(B)}\sum_{i=1}^{N(B)}A_k(i)B_l(j)}$$

for every (k,l)∈{1,…,n(B)}×{1,…,m(B)}. We use the following functions for{A_1,…,$A_{n(B)}$} & {B_1,…,$B_{m(B)}$} :

$$A_1(x)=\begin{cases}0.5(1+\cos\frac{\pi}{h}(x-x_1)) & \text{if } x\in[x_1,x_2]\\ 0 & \text{otherwise}\end{cases}$$

$$A_k(x)=\begin{cases}0.5(1+\cos\frac{\pi}{h}(x-x_k)) & \text{if } x\in[x_{k-1},x_{k+1}]\\ 0 & \text{otherwise}\end{cases}$$

$$A_n(x)=\begin{cases}0.5(1+\cos\frac{\pi}{h}(x-x_n)) & \text{if } x\in[x_{n-1},x_n]\\ 0 & \text{otherwise,}\end{cases}$$

where n = n(B), k = 2,…, n, h = (N(B) - 1)/(n - 1), x_k = 1+ h·(k-1) and

$$B_1(y)=\begin{cases}0.5(1+\cos\frac{\pi}{s}(y-y_1)) & \text{if } y\in[y_1,y_2]\\ 0 & \text{otherwise}\end{cases}$$

$$B_t(y)=\begin{cases}0.5(1+\cos\frac{\pi}{s}(y-y_t)) & \text{if } y\in[y_{t-1},y_{t+1}]\\ 0 & \text{otherwise}\end{cases}$$

$$B_m(y)=\begin{cases}0.5(1+\cos\frac{\pi}{s}(y-y_m)) & \text{if } y\in[y_{m-1},y_m]\\ 0 & \text{otherwise,}\end{cases}$$

where m = m(B), t = 2,…, m, s = (M(B) − 1)/(m− 1), y_t = 1+ s·(t − 1).

Step 5: $F_{kl}^B(i,j)$ values are multiplied with 255and the blocks are recombined before writing them to the final compressed image.

ii. Inverse Fuzzy Transformation Pseudo code:

Input: Compressed Data Image P'(i,j) an n x m grayscale or color(repeat all steps for each RGB component of pixel) image.

Output: Decompressed Data Image P(k,l) an N x M image.

Algorithm:

Step 1: Find Fuzzy Relation R: (i, j)∈{1,…,n} ×{1,…,m} → R(i,j) ∈ [0,1] where

R (i,j) = P'(i,j)/255 for 256 gray levels.

Step 2: Divide R (i,j) into blocks of n(B) x m(B) pixels and call it $R_B(i,j)$

Step 3: Define Uniform Fuzzy Partitions A of (1,…, n) and B of (1,…, m). The functions A and B are same as defined above.

Step 4: Each compressed block $R_B(i,j)$ is decoded to a block $P^F_{n(B)m(B)}$ of sizes N(B) x M(B) by using inverse discrete FTR defined as:

$$P^{F}_{n(B)m(B)}(i,j)=\sum_{k=1}^{n(B)}\sum_{l=1}^{m(B)}F_{kl}^{B}A_k(i)B_l(j)$$

for every (i, j)∈{1,…,N(B)}×{1,…,M(B)}. F_{kl}^B(i,j) is same as defined above.

Step 5: $P_{kl}^B(i,j)$ values are multiplied with 255 and the blocks are recombined before writing them to the final decompressed image.

iii. LSB Steganography for RGB images:

We have used the standard LSB Steganography algorithm for RGB images wherein the R, G, B values of a pixel of secret image are hidden in the R, G, B pixel values in the cover image and 8 pixels in cover image are required to hide one pixel of secret image. The only modification that we have made to this is that we have also embedded the width and the height of the secret image in the first 16 diagonal pixels in the Red & Green color components of the cover image respectively so that while extraction of the image we can find out secret image dimensions.

RESULTS OBTAINED

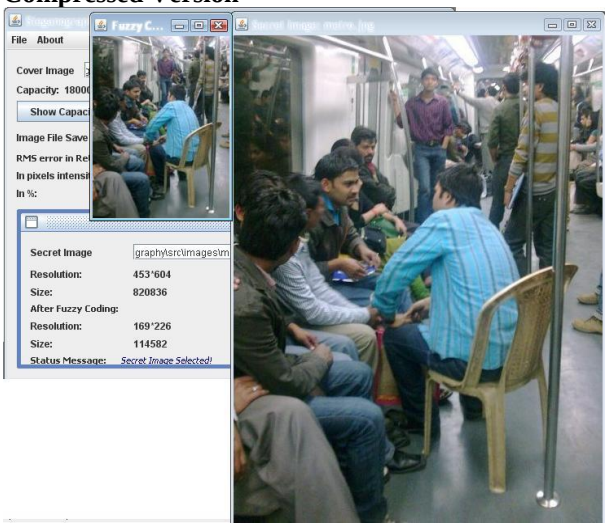
The extracted compressed secret image was exactly the same as the embedded compressed secret image. But, when it was decompressed and compared to the original secret image an

average RMS error of 4.172% & mean Quantization error of 2.484% was observed for a varied set of secret images taken. The quantization error can be reduced by the use of a sharpening filter that would unblur the extracted decompressed image. Other noise components could also be reduced by the use of a locally adaptive smoothening filter.
The entire project was made in NetBeans IDE 6.8 using Java Swings API (JApplet for GUI).

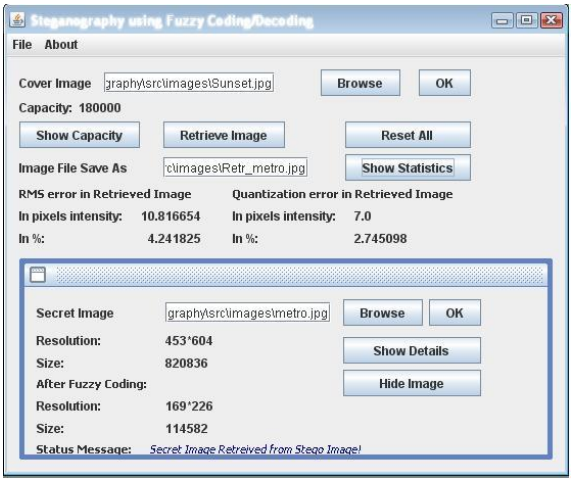
SCREEN SHOTS:

Screen Shot 1: Selecting Cover Image

Screen Shot 2: Selecting Secret Image & Viewing Its Compressed Version



Screen Shot 3: GUI After Retrieving Secret Image Showing Statistics



Screen Shot 4: Original V/s Retrieved Secret Image

ACKNOWLEDGEMENTS
Our sincere thanks to Prof. (Dr.) N.D. Kaushika & Prof. V.K. Mittal for suggesting the problem & for constant expert guidance throughout the project. We would also like to thank our project guide Ms. Komal Buddhiraja for her support & guidance which made this project possible.

FUTURE SCOPE
Future scope of this project lies in the application of Fuzzy Transform Coding/Decoding to increase the embedding capacity of other Image Steganography algorithms specially those operating in the frequency domain like JSteg & then comparing the results with that of LSB.

CONCLUSION
Thus, we in this project improved the image hiding capacity of LSB Image Steganography by using Fuzzy Transform Coding/Decoding. A compression ratio of 7.11 was achieved which improved the embedding efficiency from a just 12.5% (1/8) to 88.88% (7.11/8).

REFERENCES & BIBLIOGRAPHY

[1] Zahra Toony, Hedieh Sajedi & Mansour Jamzad, "A High Capacity Image hiding Method based on Fuzzy Image Coding/Decoding", 14th International CSI Computer Conference (CSICC'09) pgs: 512 – 523 ,©2009 IEEE.

[2] Ferdinando Di Martino, Vincenzo Loia, & Salvatore Sessa, "Direct and Inverse Fuzzy Transforms for Coding/Decoding Color Images in YUV Space", Journal of Uncertain Systems Vol.3, No.1, pp.11-30, 2009, © 2009 World Academic Press, UK.

[3] I. Perfilieva, *"Fuzzy transforms. Institute for Research and Applications of Fuzzy Modelling"*, University of Ostrava.

[4] Loquin Kevin, "Fuzzy histograms and density estimation".

[5] Nick Efford, "Digital Image Processing – A practical introduction using Java", Pearson Education, ISBN: 81 – 7808 – 074 – 5, ©2000 Pearson Education.