Multi Image In Speech Steganography

Shaymaa Abdul Hamzh Al Garawi
Department of Computer Science, University of Babylon

Abstract.

The proposed system aims to embed multi color(256) images (two up to four image)each with size(128×128)pixels inside sound file.

The system consist of two stages: The first stage is the embedding process throw using: discrete wavelet transform technique, image coefficients transformations, similar blocks technique and key generation. The output of embedding stage are: Stego_Sound file and key sequence file. The two files are transmitted by multi media communications to the receiver.

The second stage is extracting hidden images throw using: The key sequence which has been used as an index to extract the images coefficients from the Stego_Sound file, the inverse images coefficients transformations, the inverse Discrete Wavelet Transform.

الخلاصة.

يهدف النظام المقترح إلى إخفاء صور متعددة ذوات (256) تدرج لوني(من اثنين إلى أربعة صور)وبحجم(128×128) نقطة داخل ملف صوتي.

يتألف النظام من مرحلتين: المرحلة الأولى هي عملية الإخفاء من خلال استخدام: تقنية تحويل الموجة المتقطعة، تحويل معاملات الصورة، تقنية القطاعات المتشابهة وتوليد سلسلة المفتاح. مخرجات مرحلة الإخفاء هي ملف الصوت الحامل وملف سلسلة المفتاح. يرسل الملفان بواسطة وسائط الاتصالات المتعددة إلى المستلم.

المرحلة الثانية هي مرحلة الاسترجاع من خلال استخدام: سلسلة المفتاح كدليل لاستخراج معاملات الصور من ملف الصوت الحامل، عملية معكوس تحويل معاملات الصور، معكوس تحويل الموجة المتقطع.

Introduction

Steganography is the art and science of hiding communication; a steganographic system thus embeds hidden content in unremarkable cover media so as not to arouse an eavesdropper's suspicion. In the past, people used hidden tattoos or invisible ink to convey steganographic content. Today, computer and network technologies provide easy-to-use communication channels for steganography. Essentially, the information-hiding process in a steganographic system starts by identifying a cover medium's redundant bits (those that can be modified without destroying that medium's integrity). The embedding process creates a *stego medium* by replacing these redundant bits with data from the hidden message (Anderson and Petitcolas ,1998).

2 Theoretical Review

2.1 Audio Steganography

For many signals, the low-frequency content is the most important part. It is what gives the signal its identity. The high-frequency content, on the other hand, imparts flavor or nuance. Consider the human voice. If you remove the high-frequency components, the voice sounds different, but you can still tell what's being said. However, if you remove enough of the low-frequency components, you hear gibberish.

It is for this reason that, in wavelet analysis, we often speak of approximations and details (Misiti *et al*,1997).

The approximations are the high-scale, low-frequency components of the signal. The details are the low-scale, high-frequency components.

Audio steganography, the hiding of messages in audio "noise"(and in frequencies which humans can't hear), is another area of information hiding that relies on using an existing source as a space in which to hide information. Audio steganography can be problematic, however, since musicians, audiophiles, and sound engineers have been reported to be able to detect the "high pitched whine" associated with extra high-frequency information encoded in messages. In addition to storing information in non-audible frequencies or by distorting the audible signal to include additional noise (Bennett ,2004).

2.2 Data-Hiding in Audio

Audio files can also be used to hide information. With programs such as Napster, steganography is often used to copyright audio files to protect the rights of music artists. Techniques such as least significant bit insertion, phase coding, spread spectrum coding, and echo hiding can be used to protect the content of audio files. The biggest challenge all these methods face is the sensitivity of the human auditory system or HAS (Katzenbeisser *et all*, 2000). Because the HAS is so sensitive, people can often pick up randomly added noise making it hard to successfully hide data within audio files. To fully understand the different techniques of hiding information in audio files, transmission of audio signals must first be understood. When working in audio environments, the digital representation of the audio and the transmission medium must always be considered. The digital representation of an audio file is composed of the sample quantization method and the temporal sampling rate (Bender *et all*). The most frequently used format for representing samples of high-quality digital audio is a 16-bit linear quantization (Bender *et all*). Windows Audio-Visual, or WAV, files uses this format for their files. The temporal sampling rate determines an upper bound on the usable portion of the frequency range. Common temporal sampling rates range from 8 kHz to 44.1 kHz. As sample rate increases, so does the amount of usable data space for information hiding. The transmission medium of an audio signal refers to the environment in which a signal might go through to reach its destination. Bender and his colleagues categorize the possible transmission environments into the four following groups (Bender *et all*):

- Digital end-to-end environment where the sound files are copied directly from one machine to another.
- Increased/decreased resampling environment where the signal is resampled to a higher or lower sampling rate.
- Analog transmission and resampling where a signal is converted to an analog state, played on a clean analog line, and resampled.
- “Over the air” environment where the signal is played into the air and resampled with a microphone.

By understanding the different mediums in which audio signals may travel, the appropriate technique for embedding data in audio files can be determined.

2.4 Wavelet Transformation

While DCT transformations help hide watermark information or general data, they don’t do a great job at higher compression levels. The blocky look of highly compressed JPEG files is due to the 8 x 8 blocks used in the transformation process. Wavelet transformations on the other hand are far better at high compression levels and thus increase the level of robustness of the information that is hidden, something which is essential in an area like watermarking (National Academy of Sciences,2003). This technique works by taking many wavelets to encode a whole image. They allow images to be compressed so highly by storing the high frequency “detail” in the image separately from the low frequency parts. The low frequency areas can then be compressed which is acceptable as they are most viable for compression. Quantization can then take place to compress things further and the whole process can start again if needed. A simple technique using wavelets to hide information is exactly like one of the techniques discussed in the previous section (Shoemaker, 2002). Instead of altering the DCT coefficients with pseudo noise, instead the coefficients of the wavelets are altered with the noise within tolerable levels. Embedding information into wavelets is an ongoing research topic, which still holds a lot of promise.

2.2 Image Compression

The decomposition of images using basis functions that are localized in spatial position, orientation, and scale (e.g., wavelets) has proven extremely useful in a range of applications (e.g., image compression, image coding, noise removal, and texture synthesis) (e.g.,(Shapiro, Rinaldo and Galvagno, Buccigrossi and Simoncelli, 1993,1995,1999)).

The decomposition is based on separable quadrature mirror filters (QMFs) (Vaidyanathan, Vetterli, Simoncelli and Adelson,1987,1987,1990) is employed. As illustrated in Figure 1, this decomposition splits the frequency space into multiple scales and orientations. This is accomplished by applying separable lowpass and highpass filters along the image axes generating a vertical, horizontal, diagonal and lowpass subband. For example, the horizontal subband is generated by convolving with the highpass filter in the horizontal direction and lowpass in the vertical direction, the diagonal band is generated by convolving with the highpass filter in both directions, etc. Subsequent scales are creating by recursively filtering the lowpass subband. The vertical, horizontal, and diagonal subbands at scale
i =1, …, n are denoted as Vi(x, y), Hi(x,y), and Di(x, y), respectively. Shown in Figure 2 is a three-level decomposition of a "disc" image.

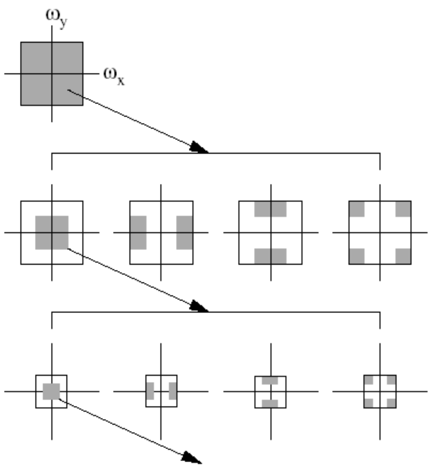


Figure 1: An idealized multi-scale and orientation decomposition of frequency space. Shown, from top to bottom, are levels 0,1, and 2, and from left to right, are the lowpass, vertical, horizontal, and diagonal subbands.

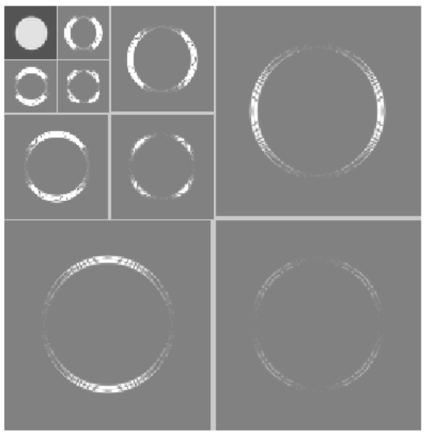
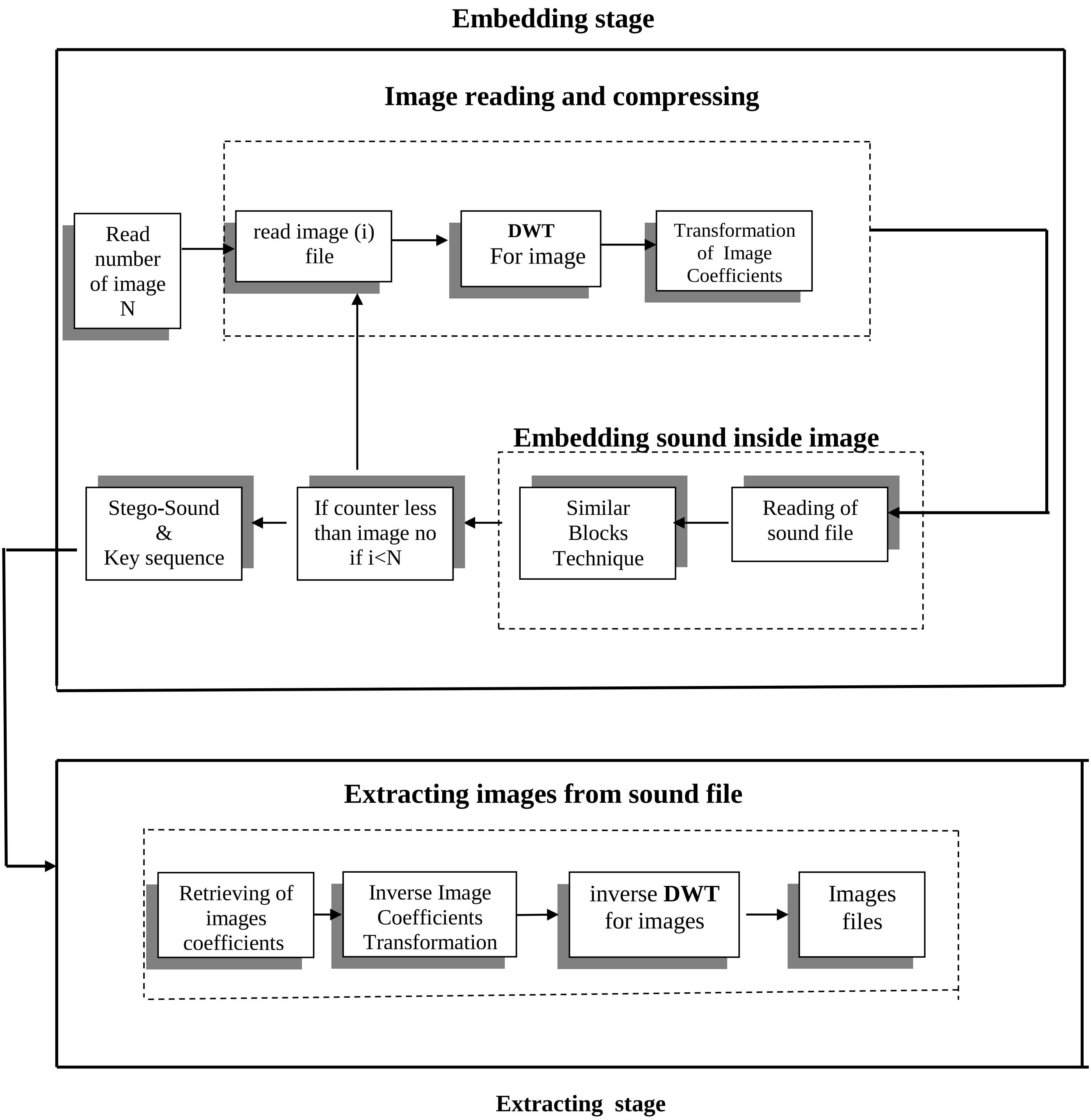


Figure 2: Shown are the absolute values of the subband coefficients at three scales and three orientations for a "disc" image. The residual lowpass subband is shown in the upper-left corner.

The log error in the linear predictor is then given by:

3 The System Framework

the block diagram below illustrates the suggestion system(see Figure (3.1)).



Figure(3.1) illustrates the block diagram of suggestion system

3.1 Reading and compressing of the Image Files

suggestion system image data is read, the image have (256)color level with size(128), then the data is compressed to reduce image file size by using discrete wavelet transform, the latter divides image data into approximations and three details (horizontal, vertical and diagonal)of level one. discrete wavelet transform is applied again on approximations from level one to get approximation and three details (horizontal, vertical and diagonal)of level two. to retrieve an original image, zeros are substituted instead of the real values of details (horizontal, vertical and diagonal) of level two and the process is applied again on level one to get the original image with lossy compression.

3.2 Transformation of Image Coefficients

The result approximations are called coefficients, while sound data are between[-1,1] ,the coefficients must be transformed between[-1,1] to facility embedding process, the transform consist of many steps, it can be represented as an algorithm as below:

Image Coefficients Transformation Algorithm

Input: Image Coefficients (ARRAY).
Output: Image Coefficients (cof).

Step1: arrange image coefficients volatilizing.

```
for i⟵1 to length(ARRAY)-1
    for j ⟵ i+1 to length(ARRAY)
        if ARRAY(i)>ARRAY(j)
        BEGIN
            temp⟵ARRAY(i)
            ARRAY(i) ⟵ARRAY(j)
            ARRAY(j) ⟵temp
        End
```

Step2: Calculate the median.

```
ln⟵length(ARRAY)
  if ln mod 2==0
  BEGIN
        xbar1⟵ARRAY(ln/2)
        xbar2⟵ARRAY((ln/2)+1)
        xbar ⟵ (xbar1+xbar2)/2
  END
  ELSE
  xbar=round(ln/2)
```

Step3: If the median of coefficients consist of two value then the average must be found.

```
xbar← (xbar1+xbar2)/2
```

Step4: Subtract the average of median from all the image coefficients to get the negative values.

```
for i←1  to dim_image
   for j←1 to dim_image
    begin
         cA1_Scale(i,j) ← COF_A(i,j)-xbar
         ARRAY1(k) ← ARRAY(k)-xbar
          k←k+1
   end
```

Step5: Calculate the Maximum value of image coefficients from step4 And Put in (Max).

```
max←cA1_Scale(1)
for i←1 to dim_image
      for j←1 to dim_image
          if cA1_Scale(i,j)>max
          max←cA1_Scale(i,j)
```

Step6: Multiply the coefficients which result from step5 **:** to transform image coefficient between[-1,1] as sound data file .

```
  for i←1 to dim_image
     for j←1 to dim_image
        cof(i,j) ← cA1_Scale(i,j)*1/max;
```

3.3 Reading of Sound File

After selecting many sectors of an audio recording (songs)for the songbird Om Kalthom and transform their sectors properties(11025 Hz, wave file pcm, 16 bit, mono), (see table 1.1).

discrete wavelet transform is applied on sound file to get approximation and details. The benefit of this step is to embed image coefficients in details coefficients of sound coefficients especially in large sound files, table(1.1) illustrates sound files and their properties.

Table(1.1) illustrate sound files and their properties

No	Sector	Properties	Sample
1	ولد الهدى فالكائنات ضياء	11025Hz, wave file pcm, 16bit, mono	118483
2	يوم يتيه على الزمان صباحه ومساءه	11025Hz, wave file pcm, 16bit, mono	175693
3	يا من له عز الشفاعة وحده …	11025Hz, wave file pcm, 16bit, mono	…..
4	دين يشيد آية في آية لبناته …	11025Hz, wave file pcm, 16bit, mono	405504
5	والله جل جلاله البناء	11025Hz, wave file pcm, 16bit, mono	106470
6	بك يا ابن عبد الله قامت سمحة ..	11025Hz, wave file pcm, 16bit, mono	…
7	ما جئت بابك مادحا بل داعيا…	11025Hz, wave file pcm, 16bit, mono	…
8	دانتك في الخلق العظيم شمائل…	11025Hz, wave file pcm, 16bit, mono	…

3.4 Similar Blocks Technique

To satisfy our target of embedding process we are going to divide image coefficients into equal blocks without remainder, either(16×16) or (32×32)Because image size equal(128×128). In addition divide sound coefficients(details)into equal blocks, the last block include remainder, it's treated before embedding process.

The first block of image coefficients is saved as an array with two dimension called image array (block1), the first block of sound coefficients(details) is saved as an array with two dimension called sound array(block2) also. The absolute difference summation is calculated between each point of (block1) with the opposite point of (block2) to get the average, the last result is divided by the dimensions. If the average equals zero, sound array(block2) number is saved because it is the best block to embed image array(block1) in it, otherwise the test is transmitted into the subsequent block, the average is calculated, if the new average is less than the last value, the new block will be saved as the best block until access to the last block of sound coefficients.

After that, the search operation is started to assign all image array(block1) in sound coefficients. This process is iterated on all image blocks reaching to the last block of image, at the last sound file will be carry image coefficients, serial key will be created which represents locations of blocks(image coefficients). Many images are selected to make similar blocks process, the result sound file is saved in wave configuration, serial key is saved in separate file, the two are transmitted to receiver later by internet using E-mail.

3.5 Retrieving of Image Coefficients

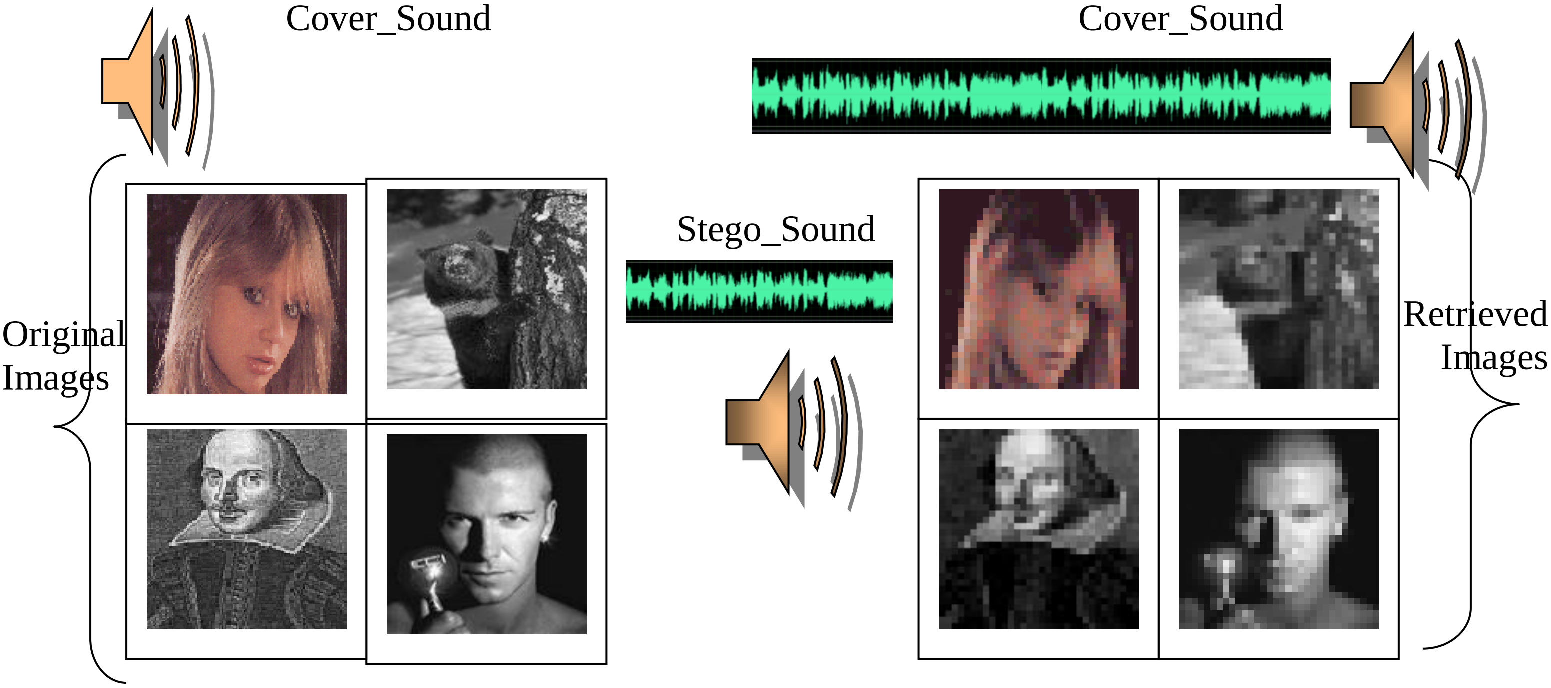
The receiver has to divide sound coefficients (details) into equal blocks, as it's divided in similar blocks stage. Compare each block number of sound coefficients(details) with the first value of serial key, if the equality condition is satisfied then all image coefficients will be assigned in an array, if the equality condition is not satisfied then it will be transmitted to the next block of sound coefficients(details) reaching to sound block which carries image coefficients until serial key finishing.

3.6 Inverse Image Coefficients Transformation

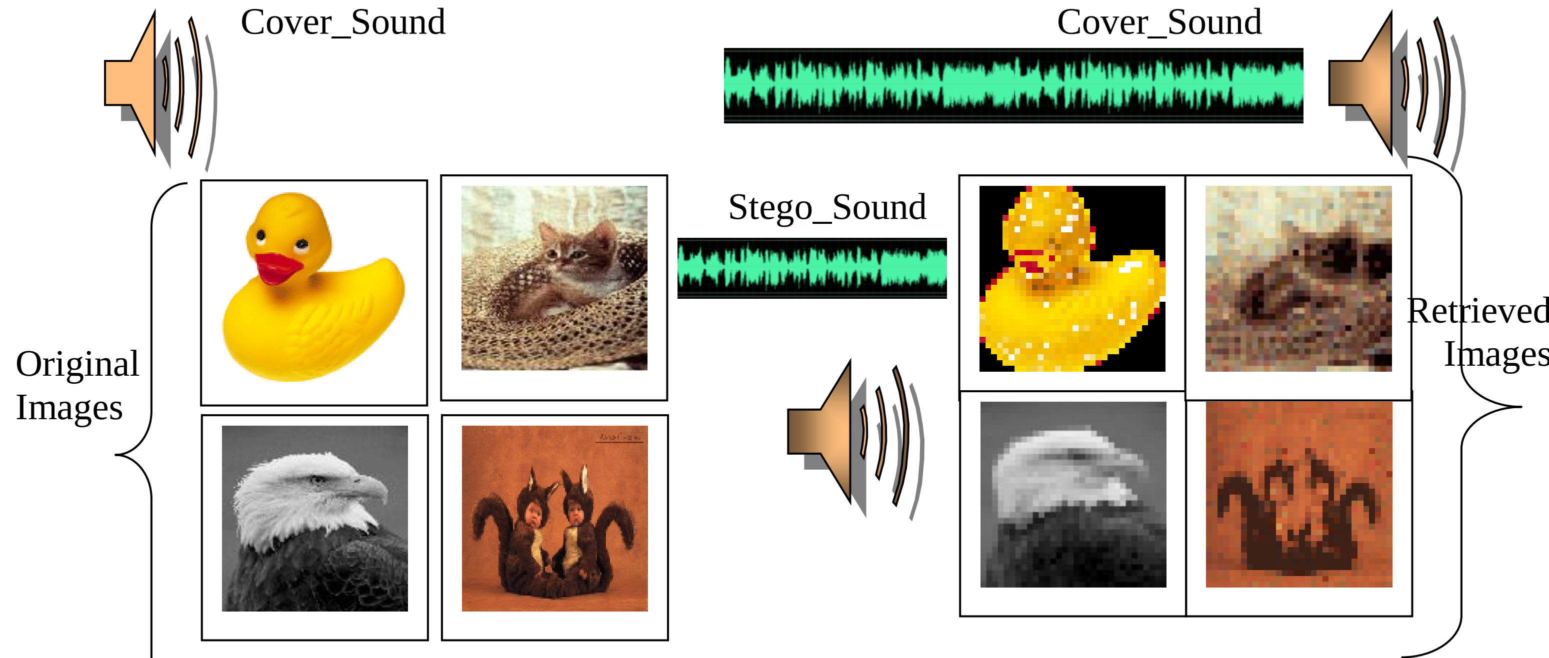
Image coefficients are transformed by multiplying them with max value to make coefficients in original range, then the median value will be subtracted from all values of image coefficients(the max value and median value are sent with key).

3.7 Compression Inverse

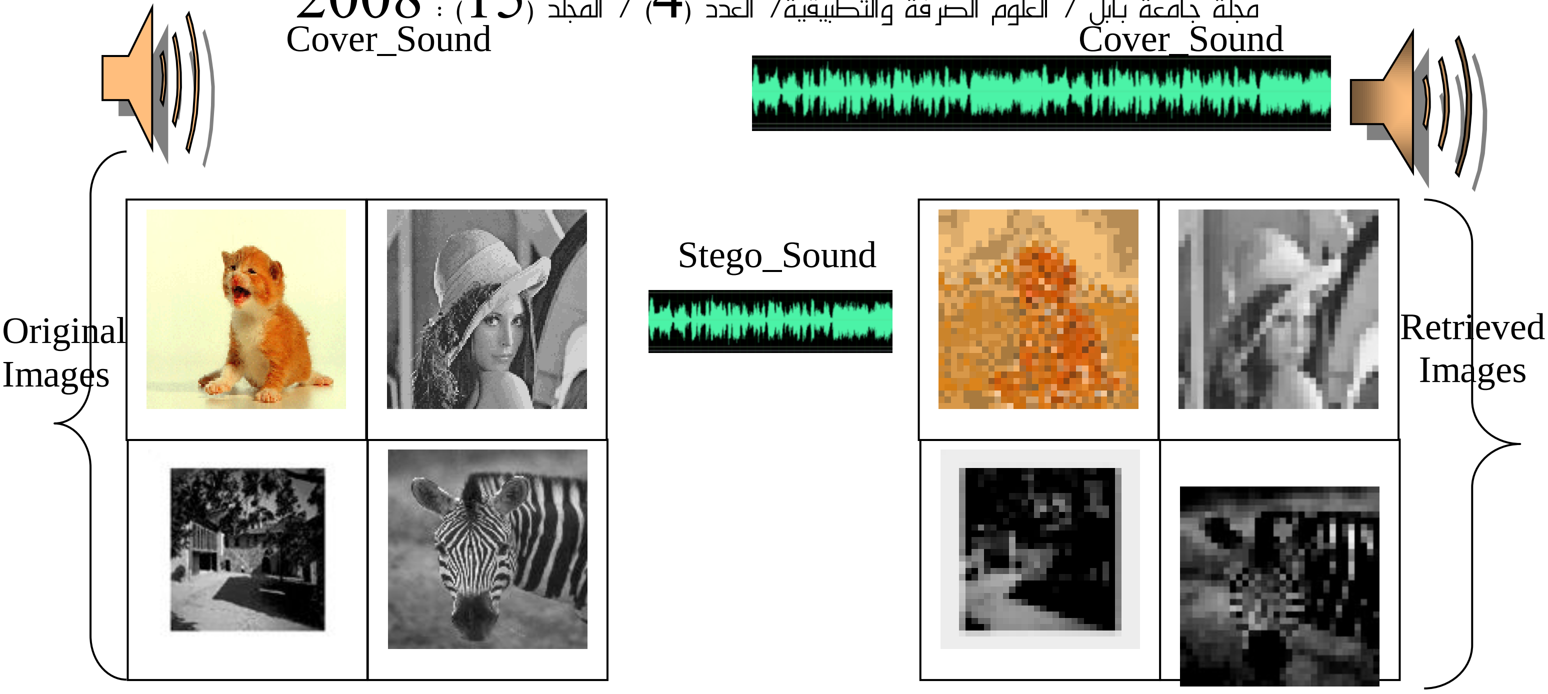
Image coefficients of level one are reconstructed by adding array of zeros to image coefficients of level two for three vectors(H,V,D), instead of the original values. In the same way image data are reconstructed by adding array of zeros to image coefficients of level one for three vectors(H,V,D), instead of the original value. The resulting image will be lossy image.



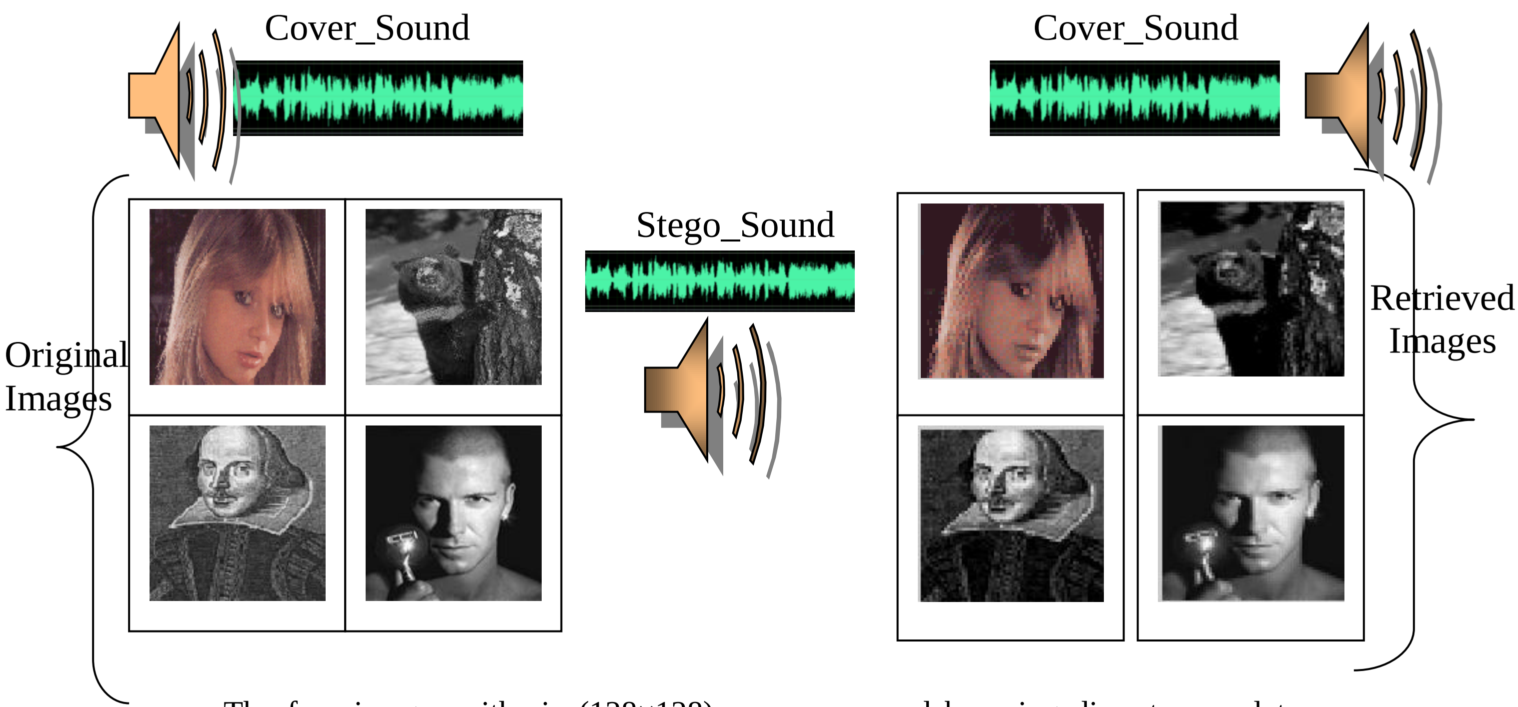
The four images with size(128×128) are compressed by using discrete wavelet transform for two levels. Its coefficients are transformed, similar blocks technique is applied on them, the size of each block which is embedded equal (16×16). The resulting coefficients equal (0.25) of the original coefficients, the serial key for each image equal (4 element), the total serial key for four image equal (16 elements).



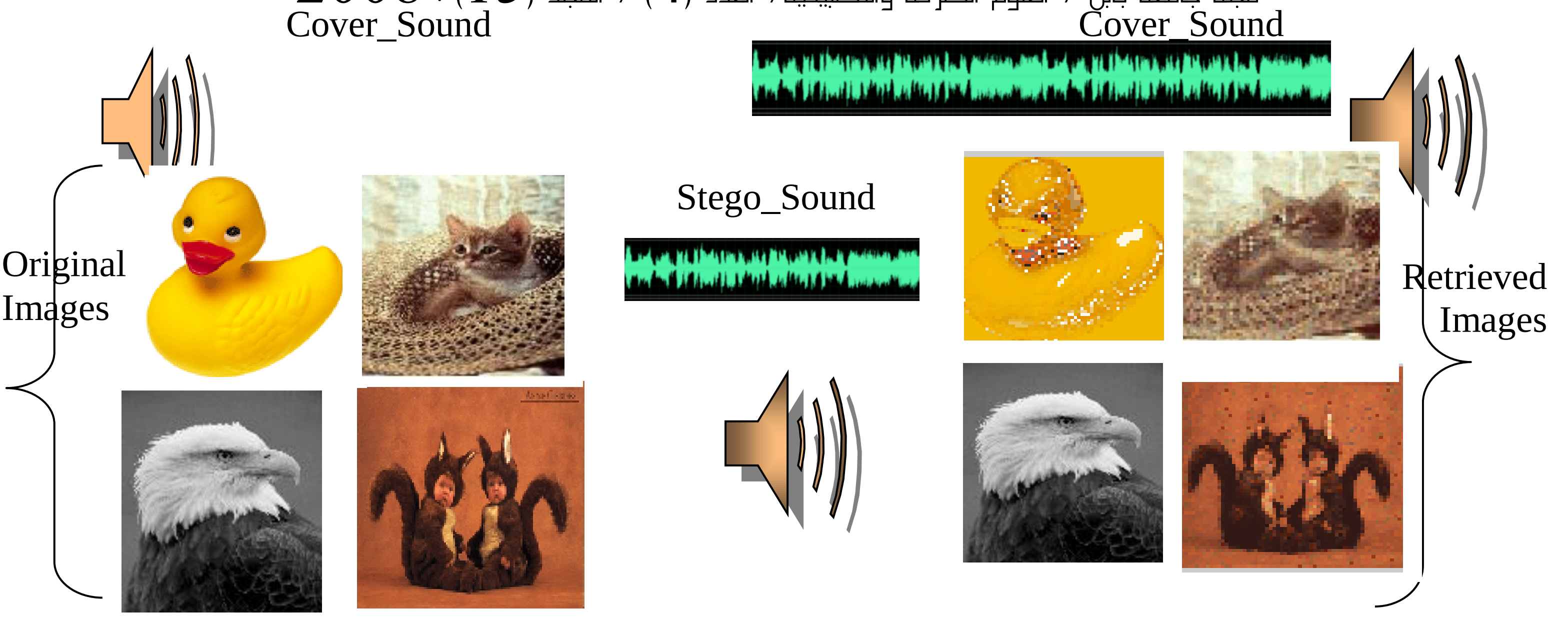
The four images with size(128×128) are compressed by using discrete wavelet transform for two levels. Its coefficients are transformed, similar blocks technique is applied on them, the size of each block which is embedded equal (16×16). The resulting coefficients equal (0.25) of the original coefficients, the serial key for each image equal (4 element), the total serial key for four image equal (16 elements).



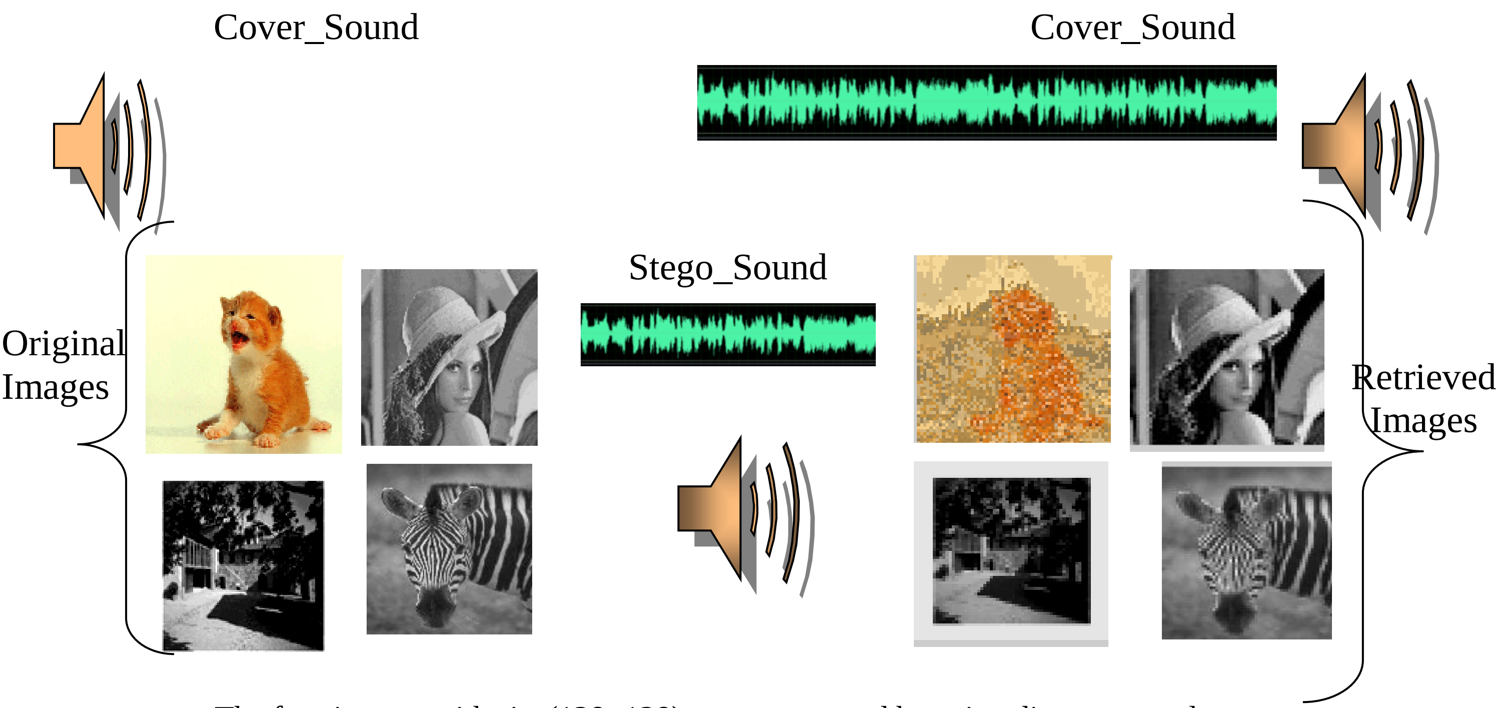
The four images with size(128×128) are compressed by using discrete wavelet transform for two levels. Its coefficients are transformed, similar blocks technique is applied on them, the size of each block which is embedded equal (16×16). The resulting coefficients equal (0.25) of the original coefficients, the serial key for each image equal (4 element), the total serial key for four image equal (16 elements).



The four images with size(128×128) are compressed by using discrete wavelet transform for one level. Its coefficients are transformed, similar blocks technique is applied on them, the size of each block which is embedded equal (32×32). The resulting coefficients equal (0.5) of the original coefficients, the serial key for each image equal (4 element), the total serial key for four image equal (16 elements).



The four images with size(128×128)pixels are compressed by using discrete wavelet transform for one level. Its coefficients are transformed, similar blocks technique is applied on them, the size of each block which is embedded equal (32×32). The resulting coefficients equal (0.5) of the original coefficients, the serial key for each image equal (4 element), the total serial key for four image equal (16 elements).



The four images with size(128×128) are compressed by using discrete wavelet transform for one level. Its coefficients are transformed, similar blocks technique is applied on them, the size of each block which is embedded equal (32×32). The resulting coefficients equal (0.5) of the original coefficients, the serial key for each image equal (4 element), the total serial key for four image equal (16 elements).

Results

Table(4.1)illustrates three tests between original and retrieved image

No	Image Name	Level	Block	Psnr	Snr	Rms	Result Image	Key
1	Sh15.bmp	2	16	32.152	2.4584	39.618	Shosho.bmp	4
2	Bear.bmp	2	16	34.397	4.5128	23.627	Zozo.bmp	4
3	Shik.bmp	2	16	31.521	1.7589	45.808	Shkshk.bmp	4
4	Beckham1	2	16	35.917	5.9921	16.649	Bobo.bmp	4
5	Ducky.bmp	2	16	33.379	3.1291	29.864	Dudu.bmp	4
6	Baz.bmp	2	16	31.803	3.0616	42.935	Ba2ba2.bmp	4
7	Eagle.bmp	2	16	35.518	6.4564	18.253	Eg.bmp	4
8	Men3.bmp	2	16	33.303	4.5552	30.393	mm.bmp	4
9	Kit2.bmp	2	16	28.029	1.5174	102.37	Kitkit.bmp	4
10	Lenna.bmp	2	16	33.919	5.431	26.375	Lolo.bmp	4
11	Belmont.bmp	2	16	31.701	3.4117	43.947	Blmnt.bmp	4
12	Zebra.bmp	2	16	29.773	0.9485	68.512	Zzz3.bmp	4

The table(4.1) illustrates peak signal to noise ratio between original image and retrieved image which is between(28.029-35.917), root mean square error between(16.649-102.37), and signal to noise ratio between(0.9485-6.4564).

Table(4.2)illustrates three tests between original and retrieved image

No	Image Name	Level	Block	Psnr	Snr	Rms	Result Image	Key
1	Sh15.bmp	1	32	35.748	6.5424	17.309	len.bmp	4
2	Bear.bmp	1	32	35.619	6.0416	17.832	dbdob.bmp	4
3	Shik.bmp	1	32	32.321	2.2492	38.103	Shk1.bmp	4
4	Beckham1	1	32	38.124	10.052	10.016	Bk1.bmp	4
5	Ducky.bmp	1	32	28.281	1.6357	96.609	Dk.bmp	4
6	Baz.bmp	1	32	33.227	5.0363	30.931	Bzon.bmp	4
7	Eagle.bmp	1	32	32.19	3.75	39.271	suker.bmp	4
8	Men3.bmp	1	32	34.899	6.6454	21.045	two.bmp	4
9	Kit2.bmp	1	32	29.49	1.7638	73.12	nono.bmp	4
10	Lenna.bmp	1	32	32.644	3.3839	35.376	Len1.bmp	4
11	Belmont.bmp	1	32	33.425	5.0874	29.552	Bl.bmp	4
12	Zebra.bmp	1	32	34.467	4.7023	23.246	Zbr.bmp	4

The table(4.2) illustrates peak signal to noise ratio between original image and retrieved image which is between(28.281-38.124), root mean square error between(10.016-96.609), and signal to noise ratio between(1.6357-10.052).

Discussion

Table(4.1) and table(4.2) illustrate three types of tests. Tests are applied on the original images and results images after applying Discrete Wavelet Transform(DWT), coefficients transformation, similar blocks technique, extracting images by Inverse Image Coefficients Transformation, inverse DWT.

The results of peak signal to noise ratio(PSNR) in table(4.1) were between (28.029-35.917), these results relatively are good in compare with the last searches results(Garawi , 2006).

The results of peak signal to noise ratio(PSNR) in table(4.2) were between (28.281-38.124), these results were better than the results on table(4.1), because applying Discrete Wavelet Transform(DWT) on the images was for one level only, this made the amount of compression less more than the last.

Root mean square error(RMS) proportion to (PSNR). (RMS) are between(10.016-96.609) in table(4.1), and between(10.016-96.609) in table (4.2).

Signal to noise ratio(SNR) oppositely proportion to(PSNR). (SNR) is between(0.9485-6.4564) in table(4.1), and between(1.6357-10.052) in table (4.2).

The key is a short easily exchangeable random number sequence(Kremer, 2004). So the smaller key leads to better results. Therefore, the increasing of embedding block size reduces serial key.

Table(4.1), after applying DWT for two level, the size of each image is(32×32)coefficients. At embedding stage, the size of each embedding block is (16×16)coefficients. Therefore, coefficients are divided into four blocks, each block is represented by a value. This value represents embedding location. Therefore, each image against it four values represent key sequence.

Table(4.1), after applying DWT for one level, the size of each image become (64×64)coefficients. At embedding stage, if the size of each embedding block is (16×16)coefficients then the serial key becomes contain from eight values. Therefore, the size of each embedding block is (32×32)coefficients, coefficients are divided into four blocks, each block is represented by value. This value represents embedding location. Therefore each image against it four values represent key sequence.

Conclusions

1- The selected images must be different(contrast) to make the distribution of images coefficients as a normal distribution in details coefficients of file sound at embedding process . Because if the selected images are similar, the distribution process will be localized in limited region in details coefficients of file sound, this will be make the distortion in sound undistributed and this results gibberish sound, from the other hand this will make the stego_sound suspicious from the attackers.
2- Applying Discrete Wavelet Transform on images for two or more levels will reduce the size of image coefficients, and will be (0.25) or less than the original image file, that means increasing compression efficiency. From the other hand the serial key will be very short especially when the size of embedding block equals (16×16) or (32×32).
3- To make the cover(sound file) not suspicions from the attacks, it is preferred selecting sound file contains noise to make images coefficients intersect with the details coefficients of file sound, the effect of images coefficients(distortion)on file sound will be un sensible.
4- Increasing number of selected images more proportion to effects on the cover (sound). because embedding a great amount of image coefficients in details coefficients of sound file results suspicious stego_sound from the attackers.

Reference

Al-Garawi, SH. 2006**, "Speech In Image Steganography",** M.Sc thesis, Babylon University.

Anderson R. J. and Petitcolas F.A.P., 1998**, "On the Limits of Steganography"**, *J. Selected Areas in Comm.*, vol. 16, no. 4, , pp.474–481.

Bender, W., Gruhl, D., Lu, A., Morimoto, N., IBM Systems Journal, **“Techniques for Data Hiding”**, http://www.research.ibm.com/journal/sj/mit/sectiona/bender.pdf.

Bennett K., 2004, **"Linguistic Steganography"**, center for education and research in information assurance and security, department of linguistics purdue university west laffayette in 47906, kbennett @cerias.purdue.edu.

Buccigrossi R.W. and Simoncelli E.P., 1999**, "Image compression via joint statistical characterization in the wavelet domain"**. *IEEE Transactions on Image Processing*, 8(12):1688ñ1701.

Katzenbeisser, Stefan and Petitcolas, Fabien A.P., 2000, **"Information Hiding: Techniques for Steganography and Digital Watermarking",** Artech House, Inc., Norwood, MA,.

Kremer, J. August 13, 2004, **"Steganography"**, http://www.com/TEAM/S/ steganography.html.

Misiti M. *et al* , 1997 , **"Wavelet Toolbox"**, MathWorks, Inc.

National Academy of Sciences, 2003, **"How do Wavelets work?",** National academy of ciences,http://www.beyonddiscovery.org/content/view.page.asp?I=1956,

Rinaldo R. and Calvagno G., 1995, **"Image coding by block prediction of multi resolution sub-images",** *IEEE Transactions on Image Processing*, 4(7):909ñ920,.

Shapiro J., 1993**, "Embedded image coding using zerotrees of wavelet coefi_cients"**. *IEEE Transactions on Signal Processing*, 41(12): 3445ñ3462.

Shoemaker, C. 2002, **"Hidden Bits: A Survey of Techniques for Digital Watermarking**",http://www.vu.union.edu/~shoemakc/watermarking/watermarking.html#watermarkobject,Virtual Union,

Simoncelli E.P. and Adelson E.H.**,** 1990, **"Subband image coding**", chapter Subband transforms, pages 143ñ192. Kluwer Academic Publishers, Norwell, MA.

Vaidyanathan P.P., 1987, **"Quadrature mirror _lter banks"**, M-band extensions and perfect reconstruction techniques. *IEEE ASSP Magazine*, pages 4ñ20.

Vetterli M.**,** 1987**, "A theory of multirate _lter banks"**, *IEEE Transactions on ASSP*, 35(3):356ñ372.