Canadian Conference on Computer and Robot Vision

ENHANCING STEGANOGRAPHY IN DIGITAL IMAGES

Abbas Cheddad, Joan Condell, Kevin Curran and Paul Mc Kevitt
School of Computing and Intelligent Systems, Faculty of Computing and Engineering
University of Ulster, Londonderry, Northern Ireland, United Kingdom
Emails: {cheddad-a, j.condell, kj.curran, p.mckevitt}@ulster.ac.uk

Abstract

The history of Steganography can be traced back to ancient civilization - the Persian and Greek conflict around 480 B.C and ancient Egyptian civilization – when Steganography was first reported to exist. Steganography is the process of hiding information in a multimedia carrier. Steganalysis, which is the official counter attack science, has defeated Steganographic algorithms whether they are based on the traditional spatial domain or the transform domain. This paper discusses the possibility of embedding data in the frames of video files. We call it adaptive as we select the specific Region of Interest (ROI) in the cover image where we can safely embed our data. We chose these regions based on human skin tone colour detection. As such the method is obviously constrained to image or video files with face instances present.

1. Introduction

Steganography is a method that involves hiding a message such as an image or an audio file in a suitable carrier. Changes are made to the carrier which represents the hidden message. If successful then no discernible change is made to the carrier. The carrier can then be sent to a receiver without anyone else knowing that it contains a hidden message. Steganography is employed in various useful applications e.g. copyright control of materials, enhancing robustness of image search engines and smart ID cards where individuals' details are embedded in their photographs. Other applications are video-audio synchronization, companies' safe circulation of secret data, TV broadcasting, Transmission Control Protocol and Internet Protocol (TCP/IP)[1] packets [1], embedding Checksum [2] etc. Petitcolas [3] demonstrated some contemporary applications one of which was in *Medical Imaging Systems* where a separation is considered necessary for confidentiality between patients' image data or DNA sequences and their captions such as Physician, Patient's name, address and other particulars. A link however, must be maintained between the two. Thus, embedding the patient's information in the image could be a useful safety measure and helps in solving such problems. An example is illustrated in Fig. 1.

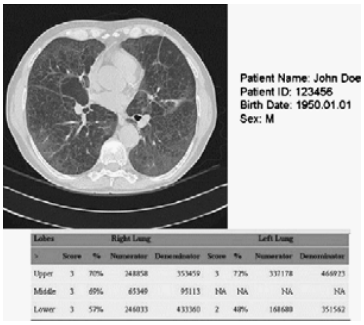

Figure 1.The DICOM SC image file that contains a screen-captured chest emphysema CAD result with the digital signature embedded [4].

Inspired by the notion that Steganography can be embedded as part of the normal printing process, Japanese firm Fujitsu[2] is pushing technology to encode data into a printed picture that is invisible to the human eye (i.e. data) but can be decoded by a mobile phone with a camera as shown in Fig. 2.

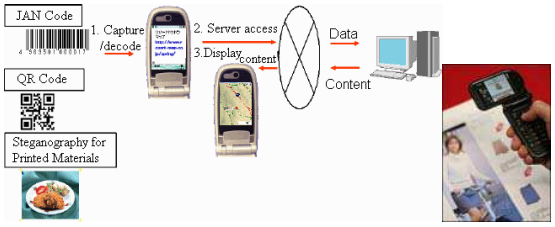

Figure 2. Fujitsu exploitation of Steganography: (left) a sketch representing the concept and (right) the idea deployed into a mobile phone.

[1] For instance a unique ID can be embedded into an image to analyze the network traffic of particular users.
[2] http://news.bbc.co.uk/go/pr/fr/-/1/hi/technology/6361891.stm Retrieved on: 15-02-2007 at: 14:17

978-0-7695-3153-3/08 $25.00 © 2008 IEEE
DOI 10.1109/CRV.2008.54

1.1 Literature Review

Existing steganography methods fall into three main categories, namely methods exploiting image format, methods embedding in the spatial domain and methods embedding in the frequency domain. Essentially, Steganography is achieved by modifying the image's Least Significant Bits (LSBs) in such a way that the carrier image remains intact visually. Fig. 3 depicts a block diagram implementing Steganography in JPEG image files exploiting the Discrete Cosine Transform (DCT).

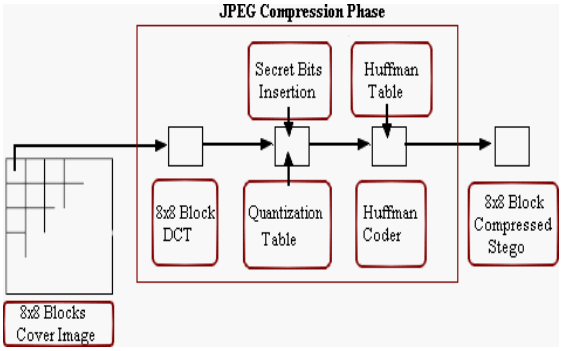

Figure 3. Data Flow Diagram showing a general process of embedding in the frequency domain.

Embedding in the spatial domain can be achieved through altering the least significant bits of the bytes of image pixel values. This process can be in a sequential fashion or in a randomised form. Algorithms based on this method have a high payload, however the method is fragile, prone to statistical attacks and sometimes visual attacks can suffice. The second type of method, the frequency domain method, is based on the embedding in the coefficient in the frequency domain (i.e., Discrete Cosine Transformation (DCT), Discrete Wavelet Transformation (DWT)). This type of technique is more robust with regard to common image processing operations and lossy compression. Another type of method is that of adaptive Steganography which adapts the message embedding technique to the actual content and features of the image. These methods can for example avoid areas of uniform colour and select pixels with large local standard deviation. Edge embedding can also be used alongside adaptive Steganography. Two of the most successful steganography tools will now be discussed.

A. S-Tools

S-Tools is a particular method which involves changing the least significant bit of each of the three colours in a pixel in a 24-bit image e.g. a 24-bit BMP file [5, 6]. The problem with 24-bit images is that they are not commonly used on the web and tend to stand out (unlike GIF, JPEG, and PNG). This feature is not helpful to Steganography. It involves a pre-processing step to reduce the number of colour entries by using a distance measurement to identify neighbour colours in terms of intensity. After this stage each colour of the dithered image would be associated with two palette entries one of which will carry the hidden data. The software for S-Tools can reduce the number of colours in the image to 256 [5]. The software uses the algorithm developed by Heckbert [7] to reduce the number of colours in an image in a way that will not visually disrupt the image [5, 8]. The algorithm plots all the colours in three dimensions (Red, Green, Blue - RGB). It searches for a collection of *n* boxes, which contains all of the colours in one of the boxes. The process starts with the complete 256*256*256 space as one box. The boxes are then recursively subdivided by splitting them in the best possible way [5]. Splitting continues until there are *n* boxes representing the space. When it is finished the program chooses one colour to represent all the colours in each box. The colour may be chosen in different ways: the centre of the box, the average box colour or the average of the pixels in the box. S-Tools, as well as other tools based on LSBs in the spatial domain, take for granted that least significant bits of image data are uncorrelated noise [9]. The system interface is easy to use. It supports a drag and drop method to load images. Once the cover image is dragged in; the system will advise the user on how much data in bytes the image can hold.

B. F5

F5 is a method which was created by Andreas Westfeld in 2001. It embeds messages by modifying the DCT coefficients. The central operation done by F5 is matrix embedding (subtraction and matrix encoding) with the aim of reducing the amount of changes made to the DCT coefficients [10]. The algorithm takes *n* DCT coefficients and hashes them to *k* bits, where *k* and *n* are computed based on the original image as well as the secret message length. If the hash value equals the message bits, then the next *n* coefficients are chosen, otherwise one of the *n* coefficients is modified and the hash is recalculated. The modifications are constrained by a threshold to estimate the acceptable hamming distance "*disT*" of *n* DCT coefficients and the original *n* DCT coefficients. This process is repeated until the hash value matches the message bits. A JAVA application version of the F5 code is publicly available [10]. F5 first recompresses the image, with a quality factor input by the user, after which the DCT coefficients are used for embedding the message. The

embedded data should be no more than 14% of the cover's size in order for it to be undetected by visual analysis [10]. F5 and S-Tools scatter the secret message over the whole carrier medium.

We conclude this section with a summary of the drawbacks of current techniques shown in Table 1.

Table 1. Drawbacks of current methods.

Method	Limitation
File formatting techniques (i.e., Header and EXIF embedding)	▪ Large payload but easily detected and defeated ▪ Not robust against lossy compression and image filters ▪ Resaving the image destroys totally the hidden data
Direct spatial LSB techniques	▪ Large payload but often offset the statistical properties of the image ▪ Not robust against lossy compression and image filters
Transform domain techniques	▪ Less prone to attacks than the former methods at the expense of capacity ▪ Breach of second order statistics ▪ Cannot resist attacks based on multiple image processing techniques

1.2 Steganalysis

Steganalysis is the science of attacking Steganography in a battle that never ends. It mimics the already established science of Cryptanalysis. Note that a Steganographer can create Steganalysis merely to test the strength of their algorithm. Steganalysis is achieved through applying different image processing techniques e.g., image filtering, rotating, cropping, translating, etc. More deliberately Steganalysis can involve coding a program that examines the stego-image structure and measures its statistical properties e.g., first order statistics (histograms) or second order statistics (correlations between pixels, distance, direction). Apart from many other advantages higher order statistics, if taken into account before embedding, can improve the signal-to-noise ratio when dealing with Gaussian additive noise.

2. The Adaptive Approach – 'Steganoflage'

Most, if not all, of the Steganographic algorithms consider the entire image file to embed the data aimed for concealment. Our motivation is based on promising results we obtained during our initial experiments. We noticed that by embedding in certain region of images (i.e., human skin) can assure us higher performance for the following reasons:

1- Human skin and faces in particular are generally the core element in images and thus in video files, automatically directing our algorithm to non-smooth regions where embedding is desirable.
2- The algorithm resists cropping attacks as it is unlikely that human faces will be deliberately cropped.
3- Exploiting success in Biometric fields makes it feasible to resist rotation attacks, since detecting human eyes would give our algorithm excellent reference points to correct and restore the initial angle.

This adaptive image content Steganography method for sequences of images (e.g. Video images) is shown in Fig. 4. We can use colour space transformations to detect and track any presence of human skin tone. The latter emerged from the field of Biometrics, where the threefold RGB matrix of a given image is converted into different colour spaces to yield distinguishable regions of skin or near skin tone. The next section will define the colour transformation used in the proposed 'Steganoflage' approach.

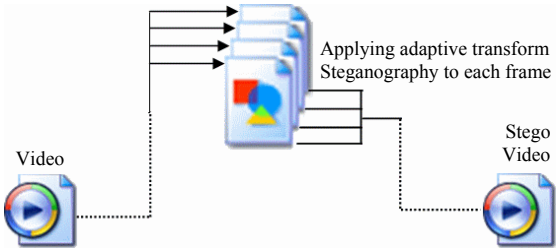

Figure 4. General overview of the proposed scheme – 'Steganoflage'.

2.1. Brief Definition of Colour Transformation Used

Colour images comprise of three channels of colours, namely Red (R), Green (G) and Blue (B). There are two types, one of which is called indexed colour. Its RGB map is represented by three vectors where each intensity pixel points to one raw (R_i,G_i,B_i) where i denotes the i^{th} entry in the 3D colour matrix. Another type is known as true colour where each channel is referenced by a matrix having the same

image dimension. The three matrices (RGB) are then blended to yield a true colour output.

YCbCr (*Y*: intensity, *Cb*: Chromatic blue and *Cr*: Chromatic red) is a well known transformation colour space. This kind of transformation is applied in image compression [11] and in object detection (human skin tone) [12]. This colour transformation invokes a rounding operator to perform a series of arithmetic operations. This phenomenon classifies it as a non-reversible lossy system. This loss of information happens only during the first transformation cycle (Fig. 5) and will show a slight effect, if any, on any additional cycles (Fig. 6). Figure 5 shows an example of this with the RGB triplet for the 'Lady' image (shown in Figure 7). Even though the changes in the first transformation cycle are small their effect appears in a reasonable compression (Fig. 5). Cycle 2 introduces no further image degradation except the slight error appearing in the Green and Blue (G, B) channels (Fig. 6). All additional cycles produce no changes. These observations were based on our initial experiments and were highlighted by Domanski and Rakowski in their work [11].

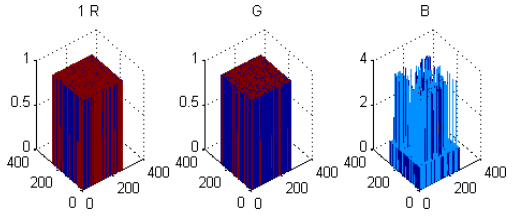

Figure 5. Cycle 1 - RGB triplet of Lady.
Signal error introduced by converting true colour images into *YCbCr* colour space.

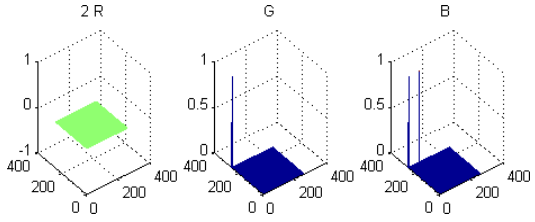

Figure 6. Cycle 2. RGB triplet of Lady.
Signal error introduced by converting the resulting images from Cycle 1 into *YCbCr* colour space.

Figure 7. 'Lady' image.

Embedding in such a colour space is meaningless if we do not correct these errors and restore the original values of the original image when we inverse transform the *YCbCr* colours to RGB colour space. In other words, we transform only the series of pixels which will carry our hidden data. By this means we can increase the PSNR value by minimizing the errors caused by the transformation.

2.2 Methodology

We anticipate that Computer Vision can play a key role in our work. Successful face localization algorithms in colour images exploit the fact that human skin tone can be localized within a certain range in the transform colour domain (i.e. RGB to YC_bC_r, HSV or Log-opponent). Steganography can benefit from this in such a way that permits us to track and embed into the edge of sequential appearances of human skin in the frames (e.g., faces in crowd, an athlete exercising, etc). We can also adjust the human skin tone values, within the permissible value ranges, to embed secret data without introducing artefacts on the carrier image.

Video files indexing and content based retrieval applications have attracted a lot of attention during the last few years and they are still areas of active research. The core of our proposal is to find salient spatial features in image frames. We perform skin tone detection to embed secret data in videos for the following reasons:

- When the embedding is spread on the entire image (or frame), scaling, rotation or cropping will result in the destruction of the embedded data because any reference point that can reconstruct the image will be lost. However, skin tone detection in the transformed colour space ensures immunity to geometric transforms.
- Our suggested scheme modifies only the regions of the skin tone in the colour transformed channel. This is done for imperceptibility reasons.
- The skin-tone has a centre point at C_b, C_r components. It can be modelled and its range is known statistically, therefore, we can embed safely while preserving these facts. Moreover, no statistical breach occurs whether it is of first order or second order type.
- If the image (or frame) is tampered with by a cropping process, it is more likely that our selected region will be in the safe zone, as human faces generally demonstrate the core elements in any given image and thus could

be considered protected areas (e.g. in portraits).

- Our Steganographic proposal is consistent with the object based coding approach followed in MPEG4 and MPEG7 standards (the concept of Video Objects (VOs) and their temporal instances, Video Object Planes (VOPs) is central to MPEG video) [13].
- Intra-frame and Inter-frame properties in videos provide a unique environment to deploy a secure mechanism for image based Steganography. We could embed in any frame (e.g., 100) an encrypted password and a link to the next frame holding the next portion of the hidden data in the video. Note this link does not necessarily need to be in a linear fashion (e.g., frames 100→12→3...→n).

Videos are one of the main multimedia files available to public on the internet thanks to the giant free web-hosting companies (e.g., YouTube, Google Videos, etc). Every day a mass of these files is uploaded online and human factors are usually present.

Usage of the $RGB \rightarrow YCbCr$ transformation is twofold; first to segment homogeneous objects in the cover image namely human skin regions in this study, and second to embed our data using the red chrominance (*Cr*). The *YCbCr* space can remove the correlation of R, G, and B in a given image. This phenomenon is what interests us as less correlation between colours means less noticeable distortion. In our approach, the concentration on skin tone is motivated by some interesting applications of the final product. For instance, to combat the use of forged passport documents or national identity cards, a security measure would be to embed individuals' information in their photos. This can also reduce the cost of chip production since many contemporary identity cards use chips. Moreover, it enhances portability as the decoding phase can take place anywhere using a tiny applet application installed on mobile devices. There are different algorithms exploiting different colour spaces to detect human skin tone in colour images [12, 14].

Our algorithm starts first with the segmentation of probable human skin regions:

$$C = Bck \cup (\bigcup_{i=1}^{n} S_i), \qquad (1)$$
$$where: S_i \cap S_j = \varnothing (\forall i \neq j)$$

In Equation (1) *C* denotes the cover image, *Bck* background regions and (S_1 , $S_{2,...,}$ S_n) are connected subsets that correspond to skin regions.

Based on our experiments we found that embedding into these regions produces less distortion to the carrier image compared to embedding in a sequential order or even in a noise-like fashion as adopted by S-Tools. In addition to this, our algorithm yields a robust output against reasonable noise attacks and translation (Fig. 8). Note that application of noise results for S-Tools and F5 are not shown here in this case as they cannot tolerate any kind of alteration to the Stego image.

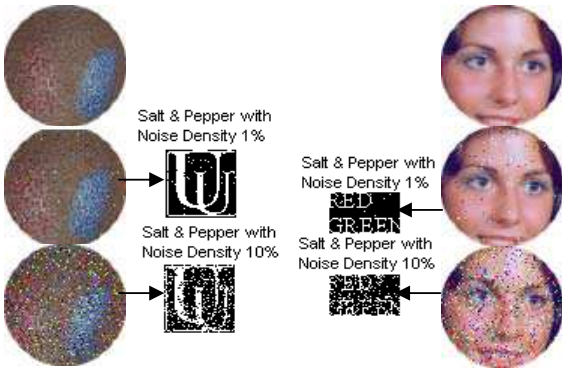

Figure 8. Results of applying Salt & Pepper noise.

Figures 9 and 10 show how the proposed method preserves the quality of the original image. Table 2 shows the in comparison of our approach to F5 and S-Tools which are known as strong algorithms. Table 2 was generated using the images shown in Figures 9 and 10. F5 and S-Tools are available online [15]. S-Tools performance was discussed previously [16].

Future work plans to for address geometric rotation distortion. Robustness against noise is due to our selection of featured points, surviving *YCbCr* compression, which are salient dot patterns in images. The resistance to geometric distortion is feasible since unlike S-Tools and F5, when we select skin tone blobs we can detect eye coordinates which act as our reference points to recover the initial position and orientation and thus make our method invariant to both rotation and translation. The next section highlights the statistical measurement used to contrast our work to S-Tools and F5.

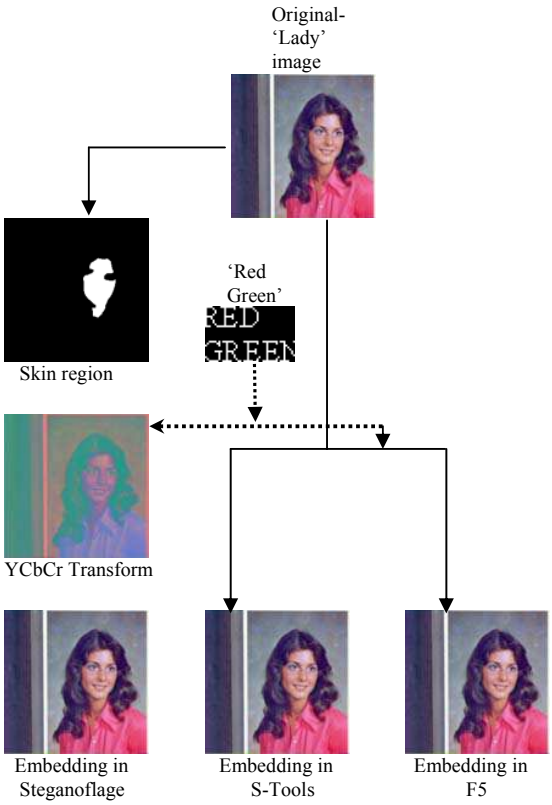

Figure 9. Embedding results. Original image (top), Binary template to hide "Red Green" (47x29) (middle) and results of each tool (bottom). Stagenoflage's stages are shown in the leftmost column.

2.3 Performance Measure

As a performance measurement for image distortion, the well known Peak-Signal-to-Noise Ratio (PSNR), which is classified under the difference distortion metrics, is applied on the Stego and the Original images. It is defined as:

$$PSNR = 10\log_{10}\left(\frac{C_{\max}^2}{MSE}\right) \qquad (2)$$

where MSE denotes Mean Square Error given as:

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left(S_{xy} - C_{xy}\right)^2$$

and $C_{\max}$ holds the maximum value in the original image, for example:

$$C_{\max} \le \begin{cases} 1 & \textit{in double precision intensity images} \\ 255 & \textit{in 8-bit unsigned integer intensity images} \end{cases}$$

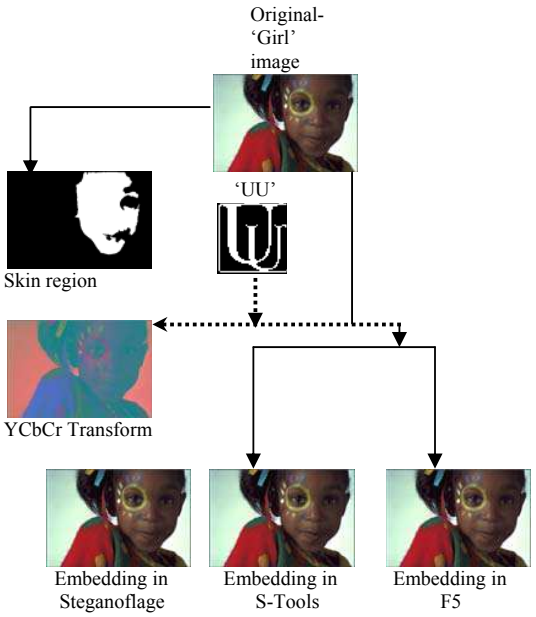

Figure 10. Embedding results. Original image (top), Binary template to hide "UU" (48x47) (middle) and results of each tool (bottom). Stagenoflage's stages are shown in the leftmost column.

In Equation (2) x and y are the image coordinates, M and N are the dimensions of the image, S_{xy} is the generated Stego image and C_{xy} is the cover image.

The PSNR is often expressed on a logarithmic scale in decibels (dB). PSNR values falling below 30dB indicate a fairly low quality (i.e., distortion caused by embedding can be obvious). A high quality Stego should strive for 40dB and above.

2.4 Experimental Results

Several simulations were performed to evaluate the performance of the proposed system. A set of RGB images were used for this purpose. Due to limited space we only show examples of two images with two different binary secret images (Fig. 9 and Fig. 10).

Experimental results are promising (Table 2). Notice that our method maintains higher PSNR despite the size of embedded bits which is more than that of S-Tools. Nevertheless, the deficiency of the proposed framework is the limited payload caused by our careful selection of salient points that can resist YCbCr compression and because of the region based spirit that the method follows. An obvious alternative would be to target video files, and this is the main focus of our current work.

Table 2. Comparison of Performance with Proposed Method (Steganoflage), S-Tools and F5.

Lady vs. Red Green

	PSNR	*Embedded Bits*	*Size Original /Stego*
Steganoflage	67.326	1368	192/ 102
S-Tools	64.192	1204	192/ 192
F5	45.119	19792	51.6/ 48.2

Girl vs. UU

	PSNR	*Embedded Bits*	*Size Original /Stego*
Steganoflage	73.335	2264	598/604
S-Tools	72.009	1198	1120/ 1120
F5	47.384	27088	281/ 271

3. Conclusion

Digital Steganography is a fascinating scientific area which falls under the umbrella of security systems. We have presented in this work some background discussions on algorithms of Steganography deployed in digital imaging. The emerging techniques such as DCT, DWT and Adaptive Steganography are not an easy target for attacks, especially when the hidden message is small. That is because they alter bits in the transform domain, thus image statistics' distortion is kept to a minimum. Generally these methods tend to have a lower payload compared to spatial domain algorithms. In short there has always been a trade off between robustness and payload. In this paper we propose a new colour image Steganography method – Steganoflage [17] – which outperforms S-Tools and F5 in many ways. A number of experiments were discussed and a table of comparative results was given. It should be borne in mind that the core element in our algorithm is skin tone detection, thus the input image or video files must be in RGB colour and human oriented. In future work, we are considering the use of face features as reference points to recover from any rotational distortion.

References

[1] Johnson, N. F. and Jajodia, S., (1998). Exploring Steganography: Seeing the Unseen. IEEE Computer, 31 (2): 26-34.

[2] Bender, W., Butera, W., Gruhl, D., Hwang, R., Paiz, F.J. and Pogreb, S., (2000). Applications for Data Hiding. IBM Systems Journal, 39 (3&4): 547-568.

[3] Petitcolas, F.A.P., (2000). "Introduction to Information Hiding". In: Katzenbeisser, S and Petitcolas, F.A.P (ed.) Information hiding Techniques for Steganography and Digital Watermarking. Norwood: Artech House, INC.

[4] Zheng Zhou, (2007). Data security assurance in CAD-PACS integration. Computerized Medical Imaging and Graphics 31: 353–360.

[5] Wayner, P., (2002). Disappearing Cryptography. 2nd ed. USA: Morgan Kaufmann Publishers.

[6] Johnson N, F., Zoran D, Sushil J, Information Hiding., (2001). Steganography and Watermarking – Attacks and Countermeasures. Kluwer Academic Publishers.

[7] Heckbert P, (1982) Colour Image Quantization for Frame Buffer Display. In Proceedings of SIGGRAPH 82: 297-307.

[8] Martin, A., Sapiro, G. and Seroussi, G., (2005). Is Image Steganography natural? IEEE Trans on Image Processing, 14 (12): 2040-2050.

[9] Westfield A and Pfitzmann, A., (1999). Attacks on Steganographic Systems Breaking the Steganography Utilities EzStego, Jsteg, Steganos and S-Tools and Some Lessons Learned. Proc of Third International Workshop on Information Hiding, IH'99 Dresden Germany, September / October, Computer Science 1768. pp. 61- 76.

[10] Westfeld, A. (2001). F5-steganographic algorithm: High capacity despite better Steganalysis. Lecture Notes in Computer Science 2137. Springer Verlag, pp. 289-302.

[11] Domanski, M and Rakowski, K. (2001). Near-Lossless Colour Image Compression for Multimedia Applications. Proceedings of EURASIP ECMCS01. Budapest: pp. 181-184.

[12] Hsu R.-L., Abdel-Mottaleb M. and Jain A. K. (2001). Face detection in color images. Proc, International Conference on Image Processing (ICIP01), Greece, pp. 1046-1049.

[13] Puri, A and Eleftheriadis, A. (1998) MPEG-4: An object-based multimedia coding standard supporting mobile applications. Mobile Networks and Applications. Springer Netherlands, 3 (1): 5–32.

[14] Osamu Ikeda. (2003). Segmentation of Faces in Video Footage Using HSV Color for Face Detection and Image Retrieval. Proceedings, International Conference on Image Processing (ICIP03). Volume 3, 14-17 Sept. pp.III-913-916.

[15] F5: http://wwwrn.inf.tu-dresden.de/~westfeld/f5.html
S-Tools: ftp://ftp.funet.fi/pub/crypt/mirrors/idea.sec.dsi.unimi.it/code/s-tools4.zip.

[16] Cheddad, A., Condell, J., Curran, K and Mc Kevitt, P. (2007) A Comparative Analysis of Steganographic Tools. Proceedings of the Seventh IT&T Conference. Institute of Technology Blanchardstown, Dublin, Ireland. Pp 29-37.

[17] Steganoflage: http://www.infm.ulst.ac.uk/~abbasc/