ISSN 2395-1621

Data Hiding And Compression Of Images Using Smvq Algorithm

#1R.Sundaresan, #2S.NarkeesBegam, #3S.Mahalakshmi

[1]vit.sundaresh@gmail.com
[2]narkees.syed89@gmail.com
[3]mahabe49@gmail.com

[1]Assistant prof Department of BME,RVSETGI,Dindigul
[2]Assistant prof Department of ECE,RVSETGI,Dindigul
[3]Assistant prof Department of ECE,RVSETGI,Dindigul

ABSTRACT

In any communication, security is the most important issue in today‘s world. Security and data hiding algorithms have been developed, which worked as motivation for the research. A novel joint data-hiding and compression scheme for digital images using side match vector quantization (SMVQ) and image inpainting. The two functions of data hiding and image compression can be integrated into one single module seamlessly. Vector quantization is also utilized for some complex blocks to control the visual distortion and error diffusion caused by the progressive compression. The scenario of present day of information security system includes privacy, genuineness, honesty, non-repudiation. This present work focus is hiding for data technique to secure data or message with authenticity and truthfulness. The entire work has done in MATLAB. The hidden message is encrypted using a simple encryption algorithm using secret key for SMVQ and hence it will be almost unfeasible for the impostor to unhide the actual secret dispatch from the embedded cover file without knowing furtive key. Only receiver and sender know the secret key. SMVQ and inpainting technique is used for embedding and extraction method. This method could be most appropriate for hiding any secret message (text, image, audio, and video) in any standard cover media such as image, audio, video files.

ARTICLE INFO

Article History

Received: 27th December 2015

Received in revised form :29th December 2015

Accepted : 30th December , 2015

Published online :

31st December 2015

I. INTRODUCTION

1.1 OBJECTIVE

To increase data hiding capacity, communication efficiency and save the network bandwidth.

1.2 DATA HIDING

Hiding information in an inconspicuous medium is called Steganography and comes from the Greek, which literally means "Covered Writing". Data hiding, a form of steganography, represents a class of processes used to embed data, such as copyright information, into various forms of media suck as image, audio, or text with a minimum amount of perceivable degradation to the ‗host‘ signal; i.e., the embedded data should be invisible and inaudible to a human observer. It is a technology to protect contents from illegal copy, unauthorized distribution and tampering. When data is embedded, it is not written at header part but embedded directly into digital media itself by changing media contents slightly to prevent human from perceiving any difference of change. Its goal is not to restrict or regulate access to the host signal, but rather to ensure that embedded data remain inviolate and recoverable. Data embedding is a process for adding information to a digital data stream. Steganography is the term used for algorithm that modify data, using its noise component, or some other aspect of the information to convey new information. The image is altered when data is hidden inside it. In order for the data to remain a secret, the image must appear to be unchanged. In some respects, the ability to spot changes will be dependent on what the subject of the picture is.

For example, slight changes in a cloudy sky would be nearly so noticeable as changes in the portrait of someone's face. If an area of the picture is monochrome, then any changes in that region would cause the area to have a noticeable mottled appearance. If a region contains a sharp edge, altering that region would result in a noticeable softening of the edge. Thus the method for inserting the data should be a function of the appearance of the picture.Hiding a message

with steganography methods reduces the chance of a message being detected. However, if that message is also encrypted, if discovered, it must also be cracked (yet another layer of protection). Steganography does not only pertain to digital images but also to other media (files such as voice, other text and binaries; other media such as communication channels). In and of itself, steganography is not a good solution to secrecy, but neither is simple substitution and short block permutation for encryption. But if these methods are combined, much stronger encryption routines are devised.
For example (again over simplified): If a message is encrypted using substitution (substituting one alphabet with another), permute the message (shuffle the text) and apply a substitution again, then the encrypted cipher text is more secure than using only substitution or only permutation. Now, if the cipher text is embedded in an image, video and voice. It is even more secure. If an encrypted message is intercepted, the interceptor knows the text is an encrypted message. With steganography, the interceptor may not know the object contains a message, and 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 message present.

1.2.1 Features of Data hiding

Data hiding with the following features. For data invisible states that information embedded shall introduce little degradation to prevent human from noticing visually. The embedded data should be directly encoded into host signal or media, rather than into a header of signal format, so that the data remain across much data file formats. And the embedded data should be immune to modification. While inseparable marking requires the data remained embedded even after transformation of the digital contents, e.g., channel noise, filtering, resampling, cropping, encoding, printing and scanning, digital-to-analog (D/A) conversion, analog-to-digital (A/D) and compression .
In data hiding technology, asymmetrical coding of the embedded data is desirable, since the purpose of the data hiding is to keep the data in the host signal, but not necessarily to make the data difficult to access.
Error correction coding should be used to ensure data integrity. It is inevitable that there will be some degradation to the embedded data when the host signal is modified. The information is hidden directly into the host image, the data size before and after the embedding operation is consistent.

1.2.2 Data-hiding Applications

Data-hiding is now watched with interest as a new technology for protecting digital contents. Several applications of data hiding are discussed in this paragraph. The embedded data are used to place an indication of ownership in the host signal, serving the same purpose as an author‘s signature or a company logo.
A second application for data hiding is tamper-proofing. It is used to indicate that the host signal has been modified from its authorized state.
A third application, feature location, requires more data to be embedded. The embedded data are hidden inspecific locations within an image. It enables one to identify individual content features, e.g., the name of the person on the left versus the right side of an image.
The forth application , Captioning URLs, email address, or other ‘meta-content‘ can be permanently bound to image. After the hidden message has been extracted, the original can be reconstructed exactly.

1.3 IMAGE COMPRESSION

The objective of image compression is to reduce irrelevance and redundancy of the image data in order to be able to store or transmit data in an efficient form. It consist two types of compression i.e. lossy compression and lossless compression.

1.3.1 Lossless Compression

Lossless compression is preferred for archival purposes and often for medical imaging, technical drawings, clip art, or comics.

Methods for lossless image compression:

Run-length encoding – used as default method in PCX and as one of possible in BMP, TGA, TIFF

Area image compression

DPCM and Predictive
CodingEntropy encoding
Adaptive dictionary algorithms such as LZW – used in GIF and TIFF
Deflation – used in PNG, MNG, and TIFFChain
codes

1.3.2 Lossy Compression

Lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossy methods are especially suitable for natural images such as photographs in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences may be called visuallylossless.

Methods for lossy compression:

Reducing the color space to the most common colors in the image. The selected colors are specified in the color palette in the header of the compressed image. Each pixel just references the index of a color in the color palette, this method can be combined with dithering to avoid posterization.
Chroma subsampling: This takes advantage of the fact that the human eye perceives spatial changes of brightness more sharply than those of color, by averaging or dropping some of the chrominance information in the image.
Transform coding: This is the most commonly used method. In particular, a Fourier-related transform such as the Discrete Cosine Transform (DCT) is widely used. The more recently developed wavelettransform is also used extensively, followed by quantization and entropy coding.
Fractal compression.

1.3.3 Other properties

The best image quality at a given bit-rate (or compression rate) is the main goal of image compression, however, there are other important properties of image compression schemes:

Scalability

Generally refers to a quality reduction achieved by manipulation of the bit stream or file (without decompression and re-compression). Other names for scalability are progressive coding or embedded bit streams. Despite its contrary nature, scalability also may be found in lossless codes, usually in form of coarse-to-fine pixel scans. Scalability is especially useful for previewing images while downloading them (e.g., in a web browser) or for providing variable quality access to e.g., databases. There are several types of scalability:

Quality progressive or layer progressive: The bitstream successively refines the reconstructed image.

Resolution progressive: First encode a lowerimage resolution; then encode the difference to higher resolutions.

Component progressive: First encode grey; thencolor.

Meta information:

Compressed data may contain information about the image which may be used to categorize, search, or browse images. Such information may include color and texture statistics, small preview images, and author or copyright information.

Processing power:

Compression algorithms require different amounts of processing power to encode and decode. Some high compression algorithms require high processing power. The quality of a compression method often is measured by the Peak signal-to-noise ratio. It measures the amount of noise introduced through a lossy compression of the image, however, the subjective judgment of the viewer also is regarded as an important measure, perhaps, being the most important measure.

II. IMAGE PROCESSING

Image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Image processing usually refers to digital image processing, but optical and analog image processing also are possible. The acquisition of images (producing the input image in the first place) is referred to as imaging.

Closely related to image processing are computergraphics and computer vision. In computer graphics, images are manually made from physical models of objects, environments, and lighting, instead of being acquired (via imaging devices such as cameras) from natural scenes, as in most animated movies. Computer vision, on the other hand, is often considered high-level image processing out of which a machine/computer/software intends to decipher the physical contents of an image or a sequence of images (e.g., videos or 3D full-body magnetic resonance scans).

In modern sciences and technologies, images also gain much broader scopes due to the ever growing importance of scientific visualization (of often large-scale complex scientific/experimental data). Examples include microarray data in genetic research, or real-time multi-asset portfolio trading in finance.

Digital image processing

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analogimage processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems.

Fundamental steps of digital image processing

Fundamental Steps of Digital Image Processing: There are some fundamental steps but as they are fundamental, all these steps may have sub-steps. The fundamental steps are described below with a neat diagram.

Image Acquisition:

This is the first step or process of the fundamental steps of digital image processing. Image acquisition could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves preprocessing, such as scaling etc.

Image Enhancement:

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. Such as, changing brightness & contrast etc.

Image Restoration:

Image restoration is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.

Color Image Processing:

Color image processing is an area that has been gaining its importance because of the significant increase in the use of digital images over the Internet. This may include color modeling and processing in a digital domain etc.

Wavelets and Multiresolution Processing:

Wavelets are the foundation for representing images in various degrees of resolution. Images subdivision successively into smaller regions for data compression and for pyramidal representation.

Compression:

Compression deals with techniques for reducing the storage required to save an image or the bandwidth to transmit it.

Particularly in the uses of internet it is very much necessary to compress data.

Morphological Processing:
Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape.

Segmentation:
Segmentation procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

Representation and Description:
Representation and description almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region or all the points in the region itself. Choosing a representation is only part of the solution for transforming raw data into a form suitable for subsequent computer processing. Description deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

Object Recognition:
Recognition is the process that assigns a label, such as, ―vehicle‖ to an object based on its descriptors.

Knowledge Base:
Knowledge may be as simple as detailing regions of an image where the information of interest is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an interrelated list of all major possible defects in a materials inspection problem or an image database containing high-resolution satellite images of a region in connection with change-detection applications.

Histogram Equalization
Histogram equalization (HE) is a very popular technique for enhancing the contrast of an image. Its basic idea lies on mapping the gray levels based on the probability distribution of the input gray levels. It flattens and stretches the dynamics range of the image's histogram and resulting in overall contrast improvement. HE has been applied in various fields such as medical image processing and radar image processing.

Contrast Enhancement by Histogram Equalization
In some CT radiographs, the features of interest occupy only a relatively narrow range of the gray scale. Contrast enhancement is a method to expand the contrast of features of interest so that they occupy a larger portion of the displayed gray level range without distortion to other features and the overall image quality.
The goal of contrast enhancement techniques is to determine an optimal transformation function relating original gray level and the displayed intensity such that contrast between adjacent structures in an image is maximally portrayed. The histogram of an image represents the relative frequency of occurrence of gray levels within an image. Histogram-modeling techniques modify an image so that its histogram has a desired shape. This is useful in stretching the low-contrast levels of an image with a narrow histogram, thereby achieving contrast enhancement.
In histogram equalization (HE), the goal is to obtain a uniform histogram for the output image, so that an ―optimal‖ overall contrast is perceived. However, the feature of interest in an image might need enhancement locally. And although there was no decrease in delectability of simulated low contrast live metastases for an experienced reader, radiologists always find the appearance of the HE-enhanced images to be objectionable in that they often introduce undesirable artifacts and noise.
Adaptive Histogram Equalization (AHE) computes the histogram of a local window centered at a given pixel to determine the mapping for that pixel, which provides a local contrast enhancement. However, the enhancement is so strong that two major problems can arise: noise amplification in ―flat‖ regions of the image and ―ring‖ artifacts at strong edges. A generalization of AHE, contrast limiting AHE (CLAHE) has more flexibility in choosing the local histogram mapping function. By selecting the clipping level of the histogram, undesired noise amplification can be reduced. In addition, by method of background subtraction, the boundary artifacts can also be reduced.

III. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM
Many data-hiding schemes for the compressed codes, which can be applied to various compression techniques of digital images, such as JPEG, JPEG2000, and vector quantization (VQ). As one of the most popular lossy data compression algorithms, VQ is widely used for digital image compression due to its simplicity and cost effectiveness in implementation. During the VQ compression process, the Euclidean distance is utilized to evaluate the similarity between each image block and the codewords in the codebook. The index of the codeword with the smallest distance is recorded to represent the block. Thus, an index table consisting of the index values for all the blocks is generated as the VQ compression codes. Instead of pixel values, only the index values are stored, therefore, the compression is achieved effectively. The VQ decompression process can be implemented easily and efficiently because only a simple table lookup operation is required for each received index.
In an adaptive data hiding method for VQ compressed images which can vary the embedding process according to the amount of hidden data. In this method, the VQ codebook was partitioned into two or more subcode books, and the best match in one of the subcode books was found to hide secret data. In order to increase the embedding capacity, a VQ-based data-hiding scheme by a codeword clustering technique was used. The secret data were embedded into the VQ index table by codeword-order-cycle permutation. By the cycle technique, more possibilities and flexibility can be offered to improve the performance of this scheme.
The search-order coding (SOC) algorithm compress the VQ index table and achieve better performance of the bit rate through searching nearby identical image blocks following a spiral path. Some steganographic schemes were also used to embed secret data into SOC compressed codes.

Disadvantage

Data hiding is always conducted after image compression, which means the image compression process and the data hiding process are two independent modules on the server or sender side. Under this circumstance, the attacker may have the opportunity to intercept the compressed image without the watermark information embedded, and the two independent modules may cause a lower efficiency in applications.

3.2 PROPOSING SYSREM

A scheme not only focus on the high hiding capacity and recovery quality, but also establish a joint data-hiding and compression(JDHC) concept and integrate the data hiding and the image compression into a single module seamlessly, which can avoid the risk of the attack from interceptors and increase the implementation efficiency.

The JDHC scheme is based on SMVQ and image inpainting. On the sender side, except for the blocks in the leftmost and topmost of the image, each of the other residual blocks in raster-scanning order can be embedded with secret data and compressed simultaneously by SMVQ or image inpainting adaptively according to the current embedding bit.

VQ is also utilized for some complex residual blocks to control the visual distortion and error diffusion caused by the progressive compression. After receiving the compressed codes, the receiver can segment the compressed codes into a series of sections by the indicator bits. According to the index values in the segmented sections, the embedded secret bits can be extracted correctly, and the decompression for each block can be achieved successfully.

Advantages

High data hiding capacity Increase the implementation efficiency Avoid the risk of the attack from interceptors

Applications

Military applications
Medical applications

IV. PROJECT DESCRIPTION

4.1 INTRODUCTION

With the rapid development of Internet technology, people can transmit and share digital content with each other conveniently. In order to guarantee communication efficiency and save network bandwidth, compression techniques can be implemented on digital content to reduce redundancy, and the quality of the decompressed versions should also be preserved. Nowadays, most digital content, especially digital images and videos, are converted into the compressed forms for transmission. Another important issue in an open network environment is how to transmit secret or private data securely.

Even though traditional cryptographic methods can encrypt the plaintext into the cipher text the meaningless random data of the cipher text may also arouse the suspicion from the attacker. To solve this problem, information hiding techniques have been widely developed in both academia and industry, which can embed secret data into the cover data imperceptibly. Due to the prevalence of digital images on the Internet, how to compress images and hide secret data into the compressed images efficiently deserves in-depth study.

The JDHC scheme is based on SMVQ and image inpainting. On the sender side, except for the blocks in the leftmost and topmost of the image, each of the other residual blocks in raster-scanning order can be embedded with secret data and compressed simultaneously by SMVQ or image inpainting adaptively according to the current embedding bit.

VQ is also utilized for some complex residual blocks to control the visual distortion and error diffusion caused by the progressive compression. After receiving the compressed codes, the receiver can segment the compressed codes into a series of sections by the indicator bits. According to the index values in the segmented sections, the embedded secret bits can be extracted correctly, and the decompression for each block can be achieved successfully.

4.2 SMVQ

Side match vector quantization (SMVQ) was designed as an improved version of VQ, in which both the codebook and the subcodebooks are used to generate the index values, excluding the blocks in the leftmost column and the topmost row. SMVQ based secret hiding scheme using adaptive index. The weighted squared Euclidean distance (WSED) was utilized to increase the probability of SMVQ for a high embedding rate. In order to make the secret data imperceptible to the interceptors, hided secret data into the SMVQ compressed codes of the image by using a partially sorted codebook. The restoration of the original SMVQ-compressed image can be achieved at the receiver side. Fig.no:4.1 shows the SMVQ flowchart.

In SMVQ, a main codebook is required to encode the blocks in the first row and first column, and a subcodebook is required to encode the rest of the blocks. The subcodebook is a subset of the main codebook. SMVQ is based on the concept that the pixels of the top row in the current block are correlated closely with those of the bottom row in the upper block, and the pixels of the first column in the current block are correlated closely with those of the right column in the left block. The blocks in the first row and first column are encoded using VQ. During this process, the main codebook is fully searched to find the best representative codeword to replace the original blocks. The blocks of the first row and first column must be encoded accurately since these blocks are used to predict future blocks

4.3 IMAGE INPAINTING

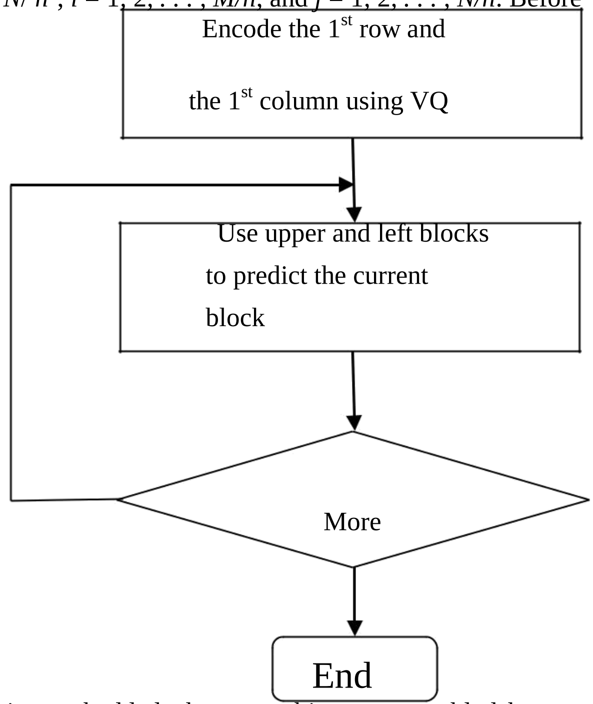
The concept of image inpainting is inherited from the ancient technique of manually repairing valuable artworks in an undetectable manner. Inpainting for digital images has found applications in such areas as repairing of damaged photographs, filling in or removing chosen areas, and wiping off visible watermarks. Image inpainting can generate or create image regions that initially do not exist at all, based on the useful information in the close neighborhood.

Currently, there are mainly three classes of the image inpainting methods, i.e., partial differential equation (PDE) based methods, interpolation-based methods, and patch-based methods. The PDE-based inpainting methods often propagate the available information of gray values automatically from surrounding areas into region Ω to be inpainted along a specific direction. There are several mathematical physics models that can be used for PDE-based inpainting, such as the fluid dynamics model and the heat transfer model. Different PDE models correspond to the different methods of information propagation.

4.4 IMAGE COMPRESSION AND SECRET DATA EMBEDDING

As an extension of VQ, SMVQ was developed to alleviate the block artifact of the decompressed image and increase the compression ratio, because the correlation of neighboring blocks is considered and the indices of the subcodebooks are stored.

In our scheme, the sender and the receiver both have the same codebook ψ with Wcodewords, and each codeword length is n^2. Denote the original uncompressed image sized $M \times N$ as **I**, and it is divided into the non-overlapping $n \times n$ blocks. For simplicity, assume that M and N can be divided by n with no remainder. Denote all k divided blocks in raster scanning order as $\mathbf{B}_{i,j}$, where $k = M \times N/ n^2$, $i = 1, 2, \ldots, M/n$, and $j = 1, 2, \ldots, N/n$. Before



being embedded, the secret bits are scrambled by a secret key to ensure security.

The blocks in the leftmost and topmost of the image **I**, i.e., $\mathbf{B}_{i,1}$($i = 1, 2, \ldots, M/n$) and $\mathbf{B}_{1,j}$ ($j = 2, 3, \ldots, N/n$), are encoded by VQ directly and are not used toembed secret bits. The residual blocks are encoded progressively in raster-scanning order, and their encoded methods are related to the secret bits for embedding and the correlation between their neighboring blocks. The flowchart of the processing for each residual block is shown in Fig.no: 4.2.

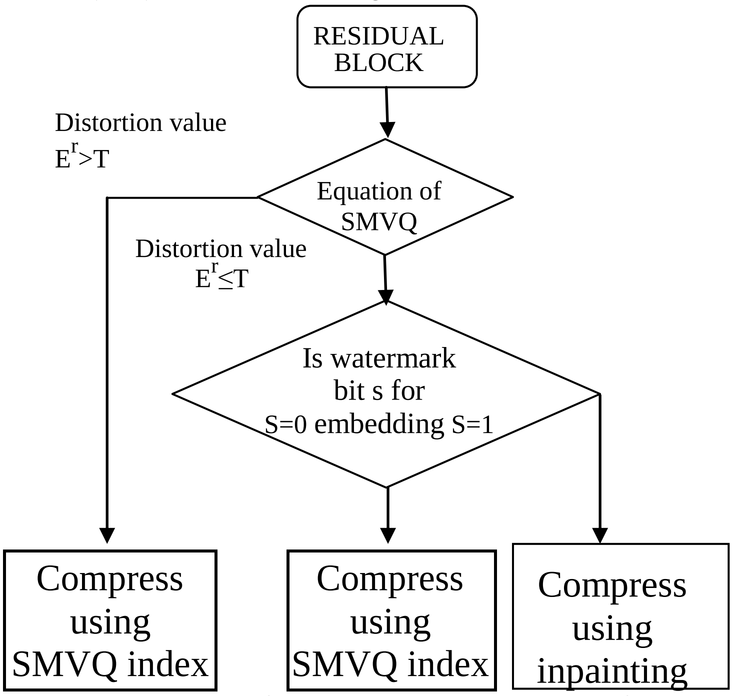


Fig.no:4.2.Flowchart of compression and secret data embedding for each residual block.

Denote the current processing block as $\mathbf{B}_{x,y}$ ($2 \le x \le M/n$, $2 \le y \le N/n$), and its left and up blocks are$\mathbf{B}_{x,y-1}$and$\mathbf{B}_{x-1,y}$, respectively.

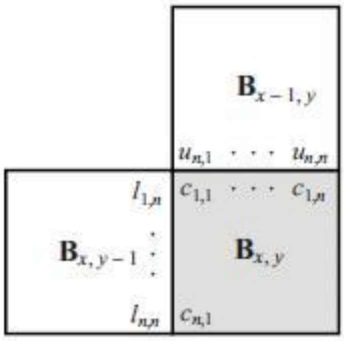


Fig.no:4.3.Illustration of the prediction based on left and up neighboring pixels.

As shown in Fig.no:4.3.$c_{p,1}$($1 \le p \le n$) and $c_{1,q}$($2 \le q \le n$) represent the $2n - 1$ pixels in the left and upperborders of $\mathbf{B}_{x,y}$. The n pixels in the right border of $\mathbf{B}_{x,y-1}$ and the n pixels in the bottom border of $\mathbf{B}_{x-1,y}$ are denoted as $l_{p,n}$ ($1 \le p \le n$) and $u_{n,q}$ ($1 \le q \le n$), respectively. Similar with SMVQ, the $2n-1$ pixels in the left and upper borders of $\mathbf{B}_{x,y}$ are predicted by the neighboring pixels in $\mathbf{B}_{x,y-1}$ and

$\mathbf{B}_{x-1,y}$:$c_{1,1} = (l_{1,n}+u_{n,1})/2$, $c_{p,1}=l_{p,n}(2 \le p \le n)$, and$c_{1,q}=$

$u_{n,q}(2 \le q \le n)$. Instead of all n^2pixels in$\mathbf{B}_{x,y}$, only these$2n - 1$ predicted pixels are used to search the codebook ψ.

After transforming all Wcodewords in the codebook ψ into the $n \times n$ matrices, the mean square error (MSE) E^w is calculated between the $2n - 1$ predicted pixels in $\mathbf{B}_{x,y}$ with the corresponding values of each transformed codewordC^w sized $n \times n$. The Rcodewords with the smallest MSEs, i.e., E^w, are selected to generate one subcodebook$\Theta_{x,y}$ for the block $\mathbf{B}_{x,y}$ ($R < W$). Suppose that, among the Rcodewords in $\Theta_{x,y}$, the codeword indexed λ has the smallest MSE, i.e., E^r , with all n^2 pixels in $\mathbf{B}_{x,y}$ ($0 \le \lambda \le R - 1$). If the value of $E^r$is greater than a pre-determined threshold T for distortion control, it implies that the current residual block $\mathbf{B}_{x,y}$ locates in a relatively complex region and it has lower correlation with its neighboring blocks.

Under this circumstance, in order to achieve better decompression quality, the standard, block independent VQ with codebook ψ is used to compress the block $\mathbf{B}_{x,y}$, and no secret bits are embedded. Otherwise, if $E^r \leq T$, it implies that the current residual block $\mathbf{B}_{x,y}$ locates in a relatively smooth region and it has higher correlation with its neighboring blocks. Thus, in this condition, SMVQ or image inpainting is adaptively utilized to compress the block $\mathbf{B}_{x,y}$ according to the secret bit *s* for embedding, which results in the shorter index length and the success of secret data hiding.

Note that, if VQ is adopted, an indicator bit, i.e., 0, should be added before the compressed code of the VQ index for $\mathbf{B}_{x,y}$. If not, the indicator bit, i.e., 1, is added as the prefix of the compressed code for $\mathbf{B}_{x,y}$. As for the block $\mathbf{B}_{x,y}$, if its $E^r$is not greater than the threshold$T$and thecurrent secret bit *s* for embedding is 0, SMVQ is utilized to conduct compression, which means that the index value λ occupying $[\log_2 R]$ bits is used to represent the block $\mathbf{B}_{x,y}$ in the compressed code. Because the codeword number R in subcodebook$\Theta_{x,y}$ is less than the codeword number W of the original codebook , the length of the compressed code for $\mathbf{B}_{x,y}$ using SMVQ must be shorter than using VQ. On the other hand, if $E^r \leq T$ and the current secret bit *s* for embedding is 1, the image inpainting technique is used.

Image inpainting can recover the image structural information effectively when the processed region is not too large. Evidently, if $E^r \leq T$, it implies that $\mathbf{B}_{x,y}$ locates in a relatively smooth region. Thus, it is suitable to conduct image inpainting in the compression for $\mathbf{B}_{x,y}$ under this condition. In our scheme, a PDE-based image inpainting method using the fluid dynamics model is adopted. Denote $\mathbf{B}_{\chi}$as the region including the current block$\mathbf{B}_{x,y}$that needscompression by inpainting and the available neighboring region of $\mathbf{B}_{x,y}$. Let $B_{\chi}(\xi, \eta)$ be the gray value of $\mathbf{B}_{\chi}$ in the coordinate (ξ, η). The Laplacian$B_{\chi}(\xi, \eta)$ is used as a smoothness measure of the region $\mathbf{B}_{\chi}$.

By analogizing the inpainting process as the fluid flowing and imitating the practice of a traditional art professional in the manual retouching, details in the unknown region may be created through propagating the available information in the surrounding areas into the unknown region along isophote directions. Clearly, variations in image gray values are minimal along the isophote directions. The scalar product in the above equation indicates projection of the smoothness change onto the direction of isophote. By using the finite difference method, discretized iteration algorithm to solve the PDE is obtained. Information propagation of this inpainting model finishes until the gray values in $\mathbf{B}_{x,y}$ reach stable state.

Consequently, the structural and geometric information of the block $\mathbf{B}_{x,y}$ can be recovered effectively without serious blurring on edges. Consequently, when s=1, in order to indicate that block $\mathbf{B}_{x,y}$ is processed by inpainting and differentiate from the index λ produced by SMVQ, the index value R occupying $\log_2(R + 1)$ bits is used as the compressed code of $\mathbf{B}_{x,y}$ $(R > \lambda)$. For simplicity, assume that $\log_2(R + 1)$ is an integer and $[\log_2 R] \equiv \log_2(R + 1)$. After the current block $\mathbf{B}_{x,y}$ is processed, the following block in raster-scanning order repeats the above procedure.

Note that each processed block should be substituted with its corresponding decompressed result, i.e., VQ codeword, SMVQ codeword, or inpainting result, for the success of progressive mechanism. The whole procedure of image compression and secret data embedding finishes until all residual blocks are processed.

Then, the compressed codes of all image block are concatenated and transmitted to the receiver side.

4.5IMAGE DECOMPRESSION AND SECRET DATA EXTRACTION

After receiving the compressed codes, the receiver conducts the decompression process to obtain the decoded image that is visually similar to the original uncompressed image, and the embedded secret bits can be extracted either before or during the decompression. Because the $(M + N - n)/n$ blocks in the leftmost and topmost of the image needto be used in the decompression for other residual blocks, they should be first decompressed by their VQ indices retrieved from the image compressed codes. Each VQ index of these pre-decom locks occupies log2W Segmentation bits. Then, the $k-(M + N - n)/n$ residual blocks are processed block by block in raster-scanning order.

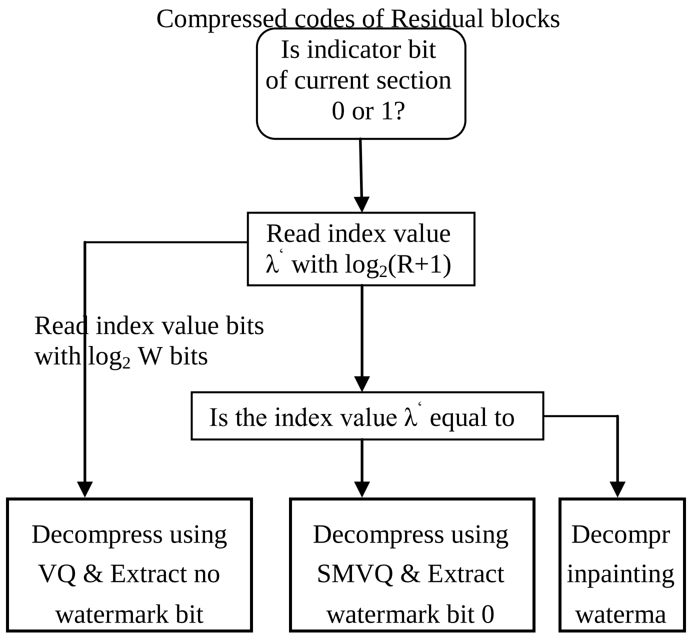


fig.no:4.4. Flowchart of decompression and secret data extraction for each residual block.

Fig.no:4.4.shows the flowchart of decompression and secret bit extraction for each residual block. To conduct the decompression and secret bit extraction of each residual block, the compressed codes are segmented into a series of sections adaptively according to the indicator bits. Explicitly, if the current indicator bit in the compressed codes is 0, this indicator bit and the following $\log_2 W$ bits are segmented as a section, which means this section corresponds to a VQ compressed block with no embedded secret bit.

The decimal value of the last $\log_2 W$ bits in this section is exactly the VQ index that can be used directly to recover the block. Otherwise, if the current indicator bit is 1, this indicator bit and the following $\log_2(R + 1)$ bits are then segmented as a section, which means this section corresponds to an SMVQ or inpainting compressed block. Denote the decimal value of the last $\log_2(R+1)$ bits in this section as λ'. Under this circumstance, if λ' is equal to R, it implies that the residual block corresponding to this section was compressed by inpainting and that the embedded secret bit in this block is 1.

Otherwise, if λ' ∈ [0, $R - 1$], it implies that the block corresponding to this section was compressed by SMVQ and that the embedded secret bit is 0. If the current segmented section corresponds to an inpainting compressed block $\mathbf{B}_{inp}$, the available information of its neighboring decompressed blocks are utilized to conduct recovery by the same inpainting technique used in the compression process. If the current segmented section corresponds to an SMVQ compressed block $\mathbf{B}_{smq}$, SMVQ index value, i.e., λ', is used to recover this block with the assistance of its left and upper decompressed blocks. Using the same prediction method described in Subsection A, the $2n-1$ pixels in the left and upper borders of $\mathbf{B}_{smq}$ are estimated by the neighboring pixels in its left and upper decompressed blocks.

Similarly, the MSEs are calculated between these $2n - 1$ predicted pixels in $\mathbf{B}_{smq}$ with the corresponding values of all *W*codewords in the codebook. Then, the *R*codewords with the smallest MSEs are chosen to generate a subcodebook. Finally, the codeword indexed λ' in the generated subcodebook is used to recover the block $\mathbf{B}_{smq}$. After all the segmented sections in the compressed codes complete the above described procedure, the embedded secret bits can be extracted correctly, and the decompressed image $\mathbf{I}_d$ can be obtained successfully.

Due to the decoding of the compressed codes, the decompressed image $\mathbf{I}_d$ doesn‘t contain the embedded secret bits any longer. Note that the process of secret bit extraction can also be conducted independently, which means that the receiver can obtain all embedded bits by simply segmenting and analyzing the compressed codes without the decoding. Therefore, besides the image compression, the proposed scheme can achieve the function of data hiding that can be used for covert communication of secret data.

The sender can transmit the secret data securely through the image compressed codes, and the receiver can extract the hidden secret data effectively from the received compressed codes to complete the process of covert communication. Additionally, because the secret data extraction in our scheme can be conducted independently with the decompression process, the receiver can obtain the secret bits at any time if he or she preserves the compressed codes. The scheme can also be used for the integrity authentication of the images, in which the secret bits for embedding can be regarded as the hash of the image principle contents.

The receiver can calculate the hash of the principle contents for the decompressed image, and then compare this calculated hash with the extracted secret bits (embedded hash) to judge the integrity of the received compressed codes and the corresponding decompressed image. If the two hashes are equal, it means the image is authenticated. Otherwise, the received compressed codes must be tampered.

V. SYSTEM REQUIREMENTS

5.1 HARDWARE REQUIREMENTS

The minimal hardware requirements are as follows,

System : Dual core processor
Hard Disk : 160 GB
RAM : 2 GB

5.2 SOFTWARE REQUIREMENTS

The minimal software requirements are as follows,

Os : Windows
Language : Mat lab
Tool Box : Signal Processing tool box

5.3 MATLAB

5.3.1 Introduction

MATLAB® is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numerical computation. Using MATLAB, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and FORTRAN.

You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computationalbiology. Add-on toolboxes (collections of special-purpose MATLAB functions) extend the MATLAB environment to solve particular classes of problems in these application areas.

MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications.

5.3.2 Key Features

High-level language for technical computing
Development environment for managing code, files, and data
Interactive tools for iterative exploration, design, and problem solving
Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration
2-D and 3-D graphics functions for visualizing data
Tools for building custom graphical user interfaces
Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java, COM, and Microsoft Excel

5.3.3 The MATLAB Language

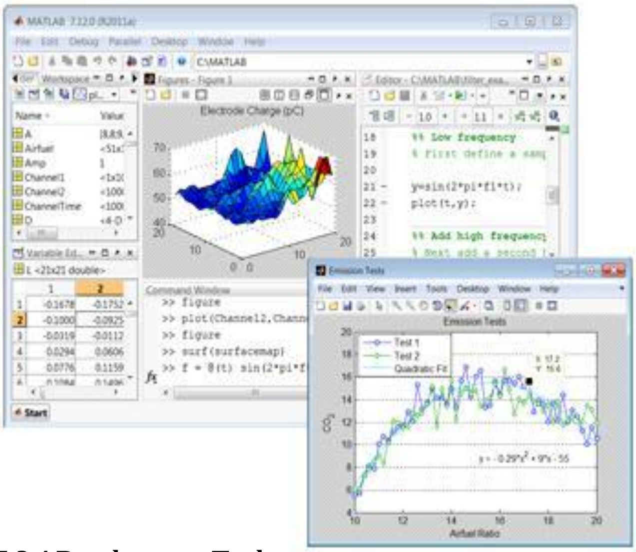
The MATLAB language supports the vector and matrix operations that are fundamental to engineering and scientific problems. It enables fast development and execution.
With the MATLAB language, you can program and develop algorithms faster than with traditional languages because you do not need to perform low-level administrative tasks, such as declaring variables, specifying data types, and allocating memory. In many cases, MATLAB eliminates the need for _for‘ loops. As a result, one line of MATLAB code can often replace several lines of C or C++ code.

At the same time, MATLAB provides all the features of a traditional programming language, including arithmetic operators, flow control, data structures, data types, object-oriented programming (OOP), and debugging features.

MATLAB lets you execute commands or groups of commands one at a time, without compiling and linking, enabling you to quickly iterate to the optimal solution.
For fast execution of heavy matrix and vector computations, MATLAB uses processor-optimized libraries. For general-

purpose scalar computations, MATLAB generates machine-code instructions using its JIT (Just-In-Time) compilation technology.

This technology, which is available on most platforms, provides execution speeds that rival those of traditional programming languages.



5.3.4 Development Tools

MATLAB includes development tools that help you implement your algorithm efficiently. These include the following:

MATLAB Editor - Provides standard editing and debugging features, such as setting breakpoints and single stepping

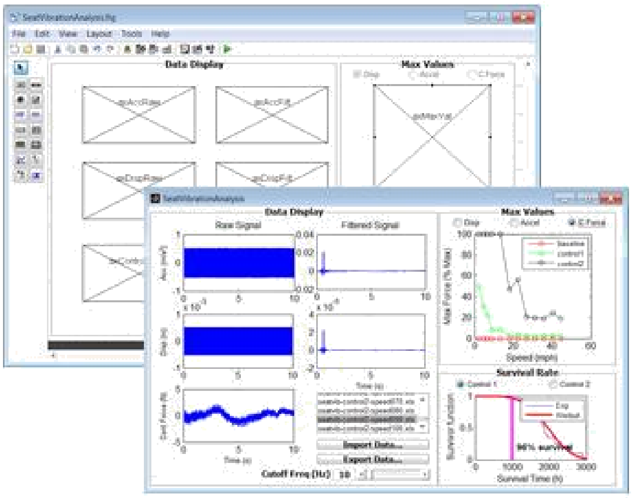
Code Analyzer - Checks your code for problems and recommends modifications to maximize performance and maintainability

MATLAB Profiler - Records the time spent executing each line of code

Directory Reports - Scan all the files in a directory and report on code efficiency, file differences, file dependencies, and code coverage

5.3.5 Designing Graphical User Interfaces

You can use the interactive tool GUIDE (Graphical User Interface Development Environment) to layout, design, and edit user interfaces. GUIDE lets you include list boxes, pull-down menus, push buttons, radio buttons, and sliders, as well as MATLAB plots and ActiveX controls. Alternatively, you can create GUIs programmatically using MATLAB functions.



5.3.6 Analyzing and Accessing Data

MATLAB supports the entire data analysis process, from acquiring data from external devices and databases, through preprocessing, visualization, and numerical analysis, to producing presentation-quality output.

5.3.7 Data Analysis

MATLAB provides interactive tools and command-line functions for data analysis operations, including:

Interpolating and decimating

Extracting sections of data, scaling, and averaging

Thresholding and smoothing

Correlation, Fourier analysis, and filtering

1-D peak, valley, and zero finding

Basic statistics and curve fitting

Matrix analysis

5.3.8 Data Access

MATLAB is an efficient platform for accessing data from files, other applications, databases, and external devices. You can read data from popular file formats, such as Microsoft Excel; ASCII text or binary files; image, sound, and video files; and scientific files, such as HDF and HDF5. Low-level binary file I/O functions let you work with data files in any format. Additional functions let you read data from Web pages and XML.

You can call other applications and languages, such as C, C++, COM objects, DLLs, Java, FORTRAN, and Microsoft Excel, and access FTP sites and Web services. Using Database Toolbox™, you can also access data from ODBC/JDBC-compliant databases.

5.3.9 Visualizing Data

All the graphics features that are required to visualize engineering and scientific data are available in MATLAB. These include 2-D and 3-D plotting functions, 3-D volume visualization functions, tools for interactively creating plots, and the ability to export results to all popular graphics formats. You can customize plots by adding multiple axes; changing line colors and markers; adding annotation, LaTEX equations, and legends; and drawing shapes.

2-D Plotting

You can visualize vectors of data with 2-D plotting functions that create:

Line, area, bar, and pie charts

Direction and velocity plots

Histograms

Polygons and surfaces

Scatter/bubble plots

Animation

VI. RESULT AND DISCUSSION

Experiments were conducted on a group of gray-level images to verify the effectiveness of the scheme. The uncompressed color image database (UCID) that contains 1338 various color images with sizes of 512 × 384 was also adopted .

The luminance components of the color images in this database were extracted and used in the experiments. The performances of compression ratio, decompression quality, and hiding capacity for the proposed scheme were evaluated. All experiments were implemented on a computer with a 3.00 GHz AMD Phenom II processor, 2.00 GB memory, and Windows 7 operating system, and the programming environment was Matlab 7.

The compression ratio C_R can be calculated. Peak signal-to-noise ratio (PSNR) was utilized to measure the visual quality of the decompressed images I_d. The image quality is better if its PSNR value is higher.

I (x, y) and I_d (x, y) are the pixel values at coordinate (x, y) of the original uncompressed image I and the decompressed image I_d, respectively Besides PSNR, the structural similarity (SSIM) was also used to assess the visual quality of the decompressed image.

The measure of SSIM was developed based on the characteristics of the human visual system (HVS), which integrated the information of structure, luminance and contrast synthetically for the image quality assessment. The standard SMVQ method has the exactly same compression ratio with the proposed scheme and the scheme. However, the standard SMVQ method cannot carry secret information within its compressed codes. Our scheme not only can carry a large amount of secret bits within the compressed codes, but also achieves higher decompression quality than the standard SMVQ method and due to the satisfactory recovery property of image inpainting.

VII.CONCLUSION

A joint data-hiding and compression scheme by using SMVQ and PDE-based image inpainting method proposed here. The blocks, except for those in the leftmost and topmost of the image, can be embedded with secret data and compressed simultaneously, and the adopted compression method switches between SMVQ and image inpainting adaptively according to the embedding bits. VQ is also utilized for some complex blocks to control the visual distortion and error diffusion.

On the receiver side, after segmenting the compressed codes into a series of sections by the indicator bits, the embedded secret bits can be easily extracted according to the index values in the segmented sections, and the decompression for all block scan also be achieved successfully by VQ, SMVQ, and image inpainting. The experimental results show that our scheme has the satisfactory performances for hiding capacity, compression ratio, and decompression quality. Furthermore, the proposed scheme can integrate the two functions of data hiding and image compression into a single module seamlessly.

REFERENCES

[1] . Pennebaker and J. L. Mitchell, The JPEG Still Image Data Compression Standard. New York, NY, USA: Reinhold, 1993.

[2] Announcing the Advanced Encryption Standard (AES), National Institute Of Standards & Technology, Gaithersburg, MD, USA, Nov. 2001.

[3] R. L. Rivest, A. Shamir, and L. Adleman, ―A method for obtaining digital signatures and public-key cryptosystems,‖ Commun. ,ACM, vol. 21, no. 2, pp. 120–126, 1978.

[4] C. C. Chang, T. S. Chen, and L. Z. Chung, ―A steganographic method based upon JPEG and quantization table modification,‖ Inf. Sci., vol. 141, no. 1, pp. 123–138, 2002.

[5] H. W. Tseng and C. C. Chang, ―High capacity data hiding in JPEG-compressed images,‖ Informatica, vol. 15, no. 1, pp. 127–142, 2004.

[6] P. C. Su and C. C. Kuo, ―Steganography in JPEG2000 compressed images,‖ IEEE Trans. Consum. Electron., vol. 49, no. 4, pp. 824–832, Nov. 2003.

[7] C. C. Chang and W. C. Wu, ―Fast planar-oriented ripple search algorithm for hyperspace VQ codebook,‖ IEEE Trans. Image Process., vol. 16, no. 6, pp. 1538–1547, Jun. 2007.

[8] C. C. Chang, W. L. Tai, and C. C. Lin, ―A reversible data hiding scheme based on side match vector quantization,‖ IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 10, pp. 1301–1308, Oct. 2006.

[9] C. Qin, S. Wang, and X. Zhang, ―Simultaneous inpainting for image structure and texture using anisotropic heat transfer model,‖ Multimedia Tools Appl., vol. 56, no. 3, pp. 469–483, 2012.

[10] T. F. Chan and J. Shen, ―Nontextureinpainting by curvature-driven diffusions,‖ J. Vis. Commun. Image Represent., vol. 12, no. 4, pp. 436–449, 2001.

[11] A. Criminisi, P. Perez, and K. Toyama, ―Region filling and object removal by exemplar-based image inpainting,‖ IEEE Trans. Image Process., vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[12] G. Schaefer and M. Stich, ―UCID—An uncompressed color image database,‖ Proc. SPIE, vol. 5307, pp. 472–480, Jan. 2004.

