ISSN NO. 2320-5407

Journal homepage: http://www.journalijar.com
Journal DOI: 10.21474/IJAR01

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH

RESEARCH ARTICLE

OVERLAPPED WATERMARKING FOR SECURED DATA TRANSMISSION.

R.P.Narmadha.
Asst.Professor(Senior), Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology.

Manuscript Info

Manuscript History:

Received: 17 March 2016
Final Accepted: 29 April 2016
Published Online: May 2016

Key words:
Halftoning, Look-Up-Table, Efficient Direct Binary Search, Least Mean Square, naive bayes classifier

***Corresponding Author**

R.P.Narmadha.

Abstract

A high efficiency multi-layer halftoning based watermarking which adopts noise-balanced error diffusion to achieve high embedding capacity and improve security aspect. The encoder employs an Efficient Direct Binary Search (EDBS) and Look-Up-Table (LUT) method to embed multiple watermarks. The decoder simply utilizes the Least-Mean-Square (LMS) and naive Bayes classifier to extract the embedded watermarks in multi-layer framework with self-decoding capability. The proposed watermarking preprocesses the watermark to reduce its bit depth. The 8-bit grayscale image (with 256 levels) is converted into a new image representation (with a lower bit depth) using the uniform scalar quantization with fixed divisor. The distance metric can be considered in determining the quantization. Experimental results suggest that only minimum milliseconds are required for embedding multiple watermarks.

Copy Right, IJAR, 2016,. All rights reserved.

Introduction:-

In image preprocessing, image data recorded by sensors on a satellite restrain errors related to geometry and brightness values of the pixels. These errors are corrected using appropriate mathematical models which are either definite or statistical models. Image enhancement is the modification of image by changing the pixel brightness values to improve its visual impact. Image enhancement involves a collection of techniques that are used to improve the visual appearance of an image, or to convert the image to a form which is better suited for human or machine interpretation.

Image segmentation:-

Segmentation is one of the key problems in image processing. Image segmentation is the process that subdivides an image into its constituent parts or objects. The level to which this subdivision is carried out depends on the problem being solved, i.e., the segmentation should stop when the objects of interest in an application have been isolated e.g., in autonomous air-to-ground target acquisition, suppose our interest lies in identifying vehicles on a road, the first step is to segment the road from the image and then to segment the contents of the road down to potential vehicles. Image thresholding techniques are used for image segmentation

After thresholding a binary image is formed where all object pixels have one gray level and all background pixels have another - generally the object pixels are 'black' and the background is 'white' where S(x, y) is the value of the segmented image, g(x, y) is the gray level of the pixel (x, y) and T(x, y) is the threshold value at the coordinates (x, y). In the simplest case T(x, y) is coordinate independent and a constant for the whole image. It can be selected, for instance, on the basis of the gray level histogram. When the histogram has two pronounced maxima, which reflect gray levels of object(s) and background, it is possible to select a single threshold for the entire image. A method which is based on this idea and uses a correlation criterion to select the best threshold, is described below. Sometimes gray level histograms have only one maximum. This can be caused, e.g., by inhomogeneous illumination of various regions of the image. In such case it is impossible to select a single thresholding value for the entire image and a local binarization technique must be applied. Segmentation of images involves sometimes not only the discrimination between objects and the background, but also separation between different regions. One method for such separation is known as watershed segmentation.

Classification:-

Image classification is a complex process and may be affected by many factors. The classification of document being processed is required for their efficient recognization as it reduces number of searches, easy recognization of document and also reduces the chance of error at different stages during processing. A classifier associates the document with class; labeling an observed document image according to the lass, region in to which it falls. The classification stage identifies each input document image by considering the detected features like spatial

Arrangements with respect to one another, layout of document, size of the document, color of the paper, texture. The categorization (Indexing) of images greatly enhance the performance of document by filtering out the relevant document and the class to which it belongs. Classification of main document is done first followed by the sub-sections. A class prototype is stored in knowledge base .the incoming document is assigned to one of the classes, depending on the value of measure of nearness in with the class prototype. This value is obtained by comparison of document under study and the class prototype. The document is assigned to the class with which highest value of the measurement is obtained.

A **discrete cosine transform** (**DCT**) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering, from lossy compression of audio (e.g. MP3) and images (e.g. JPEG) (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential equations. The use of cosine rather than sine functions is critical for compression, since it turns out (as described below) that fewer cosine functions are needed to approximate a typical signal, whereas for differential equations the cosines express a particular choice of boundary conditions.

In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample. There are eight standard DCT variants, of which four are common.

The most common variant of discrete cosine transform is the type-II DCT, which is often called simply "the DCT". Its inverse, the type-III DCT, is correspondingly often called simply "the inverse DCT" or "the IDCT". Two related transforms are the discrete sine transform (DST), which is equivalent to a DFT of real and *odd* functions, and the modified discrete cosine transform (MDCT), which is based on a DCT of overlapping data.

Wavelet Transform:-

The wavelet transform is similar to the Fourier transform (or much more to the windowed Fourier transform) with a completely different merit function. The main difference is this: Fourier transform decomposes the signal into sines and cosines, i.e. the functions localized in Fourier space; in contrary the wavelet transform uses functions that are localized in both the real and Fourier space. Generally, the wavelet transform can be expressed by the following equation:

$$F(a,b) = \int_{-\infty}^{\infty} f(x)\, \psi^{*}_{(a,b)}(x)\, \mathrm{d}x \quad \text{.......(1)}$$

Where the * is the complex conjugate symbol and function ψ is some function. This function can be chosen arbitrarily provided that it obeys certain rules.

As it is seen, the Wavelet transform is in fact an infinite set of various transforms, depending on the merit function used for its computation. This is the main reason, why we can hear the term “wavelet transform” in very different situations and applications. There are also many ways how to sort the types of the wavelet transforms. Here we show only the division based on the wavelet orthogonality. We can use *orthogonal wavelets* for discrete wavelet transform development and *non-orthogonal wavelets* for continuous wavelet transform development. These two transforms have the following properties:

1. The discrete wavelet transform returns a data vector of the same length as the input is. Usually, even in this vector many data are almost zero. This corresponds to the fact that it decomposes into a set of wavelets (functions) that are orthogonal to its translations and scaling. Therefore we decompose such a signal to a same or lower number of the wavelet coefficient spectrum as is the number of signal data points. Such a wavelet spectrum is very good for signal processing and compression, for example, as we get no redundant information here.
2. The continuous wavelet transform in contrary returns an array one dimension larger than the input data. For a 1D data we obtain an image of the time-frequency plane. We can easily see the signal frequencies evolution during the duration of the signal and compare the spectrum with other signals spectra. As here is used the non-orthogonal set of wavelets, data are highly correlated, so big redundancy is seen here. This helps to see the results in a more humane form.

Discrete Wavelet Transform:-
The discrete wavelet transform (DWT) is an implementation of the wavelet transform using a discrete set of the wavelet scales and translations obeying some defined rules. In other words, this transform decomposes the signal into mutually orthogonal set of wavelets, which is the main difference from the continuous wavelet transform (CWT), or its implementation for the discrete time series sometimes called discrete-time continuous wavelet transform (DT-CWT).

The wavelet can be constructed from a scaling function which describes its scaling properties. The restriction that the scaling functions must be orthogonal to its discrete translations implies some mathematical conditions on them which are mentioned everywhere, e.g. the dilation equation

$$\phi(x) = \sum_{k=-\infty}^{\infty} a_k \phi(Sx - k) \quad \text{........(2)}$$

where S is a scaling factor (usually chosen as 2). Moreover, the area between the function must be normalized and scaling function must be orthogonal to its integer translations, i.e.

$$\int_{-\infty}^{\infty} \phi(x)\, \phi(x+l)\, \mathrm{d}x = \delta_{0,l} \quad \text{.......(3)}$$

After introducing some more conditions (as the restrictions above does not produce a unique solution) we can obtain results of all these equations, i.e. the finite set of coefficients a_k that define the scaling function and also the wavelet. The wavelet is obtained from the scaling function as N where N is an even integer. The set of wavelets then forms an orthonormal basis which we use to decompose the signal.

Iterature review:-

A. Model based halftoning using direct binary search

In this work, M. Analoui and J. P. Allebach[12] proposed a new method to generate halftone images which are visually optimized for the display device. The algorithm searches for a binary array of pixel values that minimizes the difference between the perceived displayed continuous-tone image and the perceived displayed halftone image. The algorithm is based on the direct binary search (DBS) heuristic. Since the algorithm is iterative, it is computationally intensive. This limits the complexity of the visual model that can be used. It also impacts the choice of the metric used to measure distortion between two perceived images. In particular, use a linear, shift- invariant model with a point spread function based on measurement of contrast sensitivity as a function of spatial frequency. The non-ideal spot shape rendered by the output devices can also have a major effect on the displayed halftone image. This source of non-ideality is explicitly accounted for in our model for the display device. By recursively computing the change in perceived mean-squared error due to a change in the value of a binary pixel it achieves a substantial reduction in computational complexity. The effect of a trial change may be evaluated with only table lookups and a few additions.

B. Efficient model based halftoning using direct binary search

In this paper, R. Ulichney [13] the direct binary search (DBS) algorithm is an iterative method which minimizes a metric of error between the grayscale original and halftone image. This is accomplished by adjusting an initial halftone until a local minimum of the metric is achieved at each pixel. The metric incorporates a model for the human visual system (HVS). In general, the DBS time complexity and halftone quality depend on three factors: the HVS model parameters, the choice of initial halftone, and the search strategy used to update the halftone. Despite the complexity of the DBS algorithm, it can be implemented with surprising efficiency. It also demonstrates how the algorithm exploits the model for the HVS to efficiently yield very high quality halftones.

C. Impact of hvs models on model-based halftoning

A model for the human visual system (HVS) by S. H. Kim and J. P. Allebach [6] is an important component of many halftoning algorithms. Using the iterative direct binary search (DBS) algorithm, the halftone texture quality provided by four different HVS models that have been reported in the literature has been compared. Choosing one HVS model as the best for DBS, then develop an approximation to that model which significantly improves computational performance while minimally increasing the complexity of the code. By varying the parameters of this model, find that it is possible to tune it to the gray level being rendered, and to thus yield superior halftone quality across the tone scale. Then developed a dual-metric DBS algorithm that effectively provides a tone-dependent HVS model without a large increase in computational complexity.

D. The void-and-cluster method for dither array generation

This paper proposed by R. A. Ulichney [13] explains Halftoning to two or more levels by means of ordered dither has always been attractive because of its speed and simplicity. However, the so-called recursive tessellation arrays in wide use suffer from strong periodic structure that imparts an unnatural appearance to resulting images. A new method for generating homogeneous ordered dither arrays is presented. A dither array is built by looking for voids and clusters in the intermediate patterns and relaxing them to optimize isotropy. While the method can be used for strikingly high quality artifact-free dithering with relatively small arrays, it is quite general; with different initial conditions the familiar recursive tessellation arrays can be built. This paper presents the algorithm for generating such arrays. Example images are compared with other ordered dither and error diffusion-based techniques.

E. A survey of techniques for the display of continuous tone pictures on bilevel displays

Many displays are basically bi-level in nature with individual display cells, all of the same size, arranged in a rectangular array. J. F. Jarvis, C. N. Judice, and W. H. Ninke [5] present a survey of processing techniques for presenting continuous tone still images on such displays. Four techniques are covered in detail while several others are covered briefly. All the techniques achieve the subjective effect of continuous tone by properly controlling only the spatial density of bi-level display states.The processing techniques consist of dividing an image into picture elements and comparing the intensity of each element with a threshold value. If the element intensity is greater than the threshold, the corresponding display cell is set to the bright state; otherwise, the cell is set to the dark state. In the *ordered-dither* technique, the threshold is spatially dependent, i.e., it is determined only by the coordinates of the element being processed. In the remaining three techniques, *constrained average, dynamic threshold,* and *minimized average error,* the threshold is determined by values of elements close to the one currently being processed. The latter three thus require more processing and more storage than the first, but allow some edge emphasis. Images processed by all the described techniques are exhibited.

Experiment:-

In the fig 1, it shows the entire process of the project. First login to the process then give the image as the input, select the required part from the given image to embed the data. Encrypt the data that we want to embed into the image. Then spread spectrum the given image to embed the data. Finally will get the output as watermarked image. Our experiment has been explained as follows:-

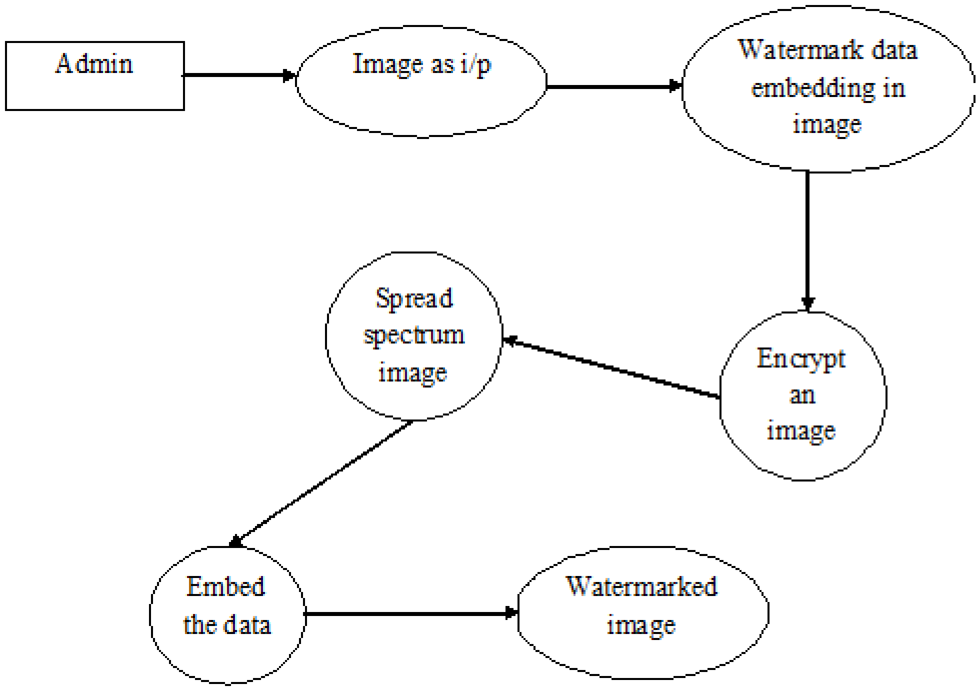

Fig 1: System Architecture Diagram

A. water mark data embedding in image:-

The proposed watermarking preprocesses the watermark to reduce its bit depth. The 8-bit grayscale image (with 256 levels) is converted into a new image representation (with a lower bit depth) using the uniform scalar quantization with fixed divisor. The distance metric can be considered in determining the quantization. The proposed watermarking preprocesses the watermark to reduce its bit depth. The 8-bit grayscale image (with 256 levels) is converted into a new image representation (with a lower bit depth) using the uniform scalar quantization with fixed divisor. The distance metric can be considered in determining the quantization.

B. Encoding scheme

This method requires a single grayscale host image, multiple grayscale watermarks with different resolutions, a pseudo key, and compressed tables. The watermark images are firstly permuted using a specific pseudo key. The watermark images are embedded in sequential manner.

The embedding process can be formally defined by Grayscale Watermark Format Conversion and Hidden-Layer Data Hiding LUT consists of several tables containing the halftone patterns for watermark encoding. The reference table is accessed using the watermark angle information for transforming the original grayscale host image into the watermarked halftone image. The main goal of incorporating the stacking constraint during LUT table construction is to ensure the blue noise distribution.

C. Spread spectrum the image:-

Technique basics:-

The core of SSIS is a spread spectrum encoder. These devices work by modulating a narrow band signal over a carrier. The carrier's frequency is continually shifted using a pseudorandom noise generator feeded with a secret key. In this way the spectral energy of the signal is spread over a wide band, thus decreasing its density, usually under the noise level. To extract the embedded message, the receiver must use the same key and noise generator to tune on the right frequencies and demodulate the original signal.

A casual observer won't be able even to detect the hidden communication, since it is under the noise level. The SSIS encoder adds more steps in order to push spread spectrum to its limits:

1. It optionally encrypts the message *m* to be embedded with *key1*, getting *e*

2. The data stream passes through a Low-Rate ECC (Error Correction Code) encoder, to acquire better robustness against destruction attacks and unwanted noise, becoming *c*.
3. Spread spectrum modulation, using a pseudorandom noise generator fed with *key2*, and get *s*
4. An inter leaver and spatial spreader processes *s* using *key3* obtaining *i*
5. The output of the inter leaver is added to the image *f*, getting g
6. A. quantization process is used to preserve the initial dynamic range of the cover image. We'll call it still *g*. It is assumed that the stego-image is sent through a noisy channel to the receiver and will become *g'*

D. **Decoding scheme:-**
A 3-level decoding scenario is proposed. The marked image repeatedly down-sampled to yield each embedded layer. All marked image samples are block-wisely transformed to the frequency domain using FFT. Since the watermark is encrypted, the pseudo decryption key is required to obtain the plaintext. The extracted watermarks are further extracted another set of NB watermarks, which were embedded by the NBEDF algorithm. The transformed blocks are fed into the naïve Bayes classifier for watermark extraction.

Results & conclusion:-
In this paper, the LUT strategy is deployed to efficiently embed multiple watermarks into a set of multi-scale watermarks, which are then embedded into a host halftone image to significantly improve the embedding capacity and reliability. Experimental results suggest that only 8.4 milliseconds are required for embedding multiple watermarks into a 512 × 512 image and 2MB is required to store the proposed compressed reference table. For the decoder, the LMS-trained filters are used to reduce the dimension of the features, and the parametric modeling is further applied to precisely model the distribution of the features. The use of the naive Bayes classifier yields a CDR of 99.32%. The decoded watermarks can be further overlapped for obtaining additional watermarks using the NBEDF self-decoding strategy.

References:-

1. M. Analoui and J. P. Allebach, Aug. 1992, “Model-based halftoning using direct binary search,” Proc. SPIE, Human Vis., Vis. Process., Digit. Display III, vol. 1666, pp. 96–108,.
2. R. W. Floyd and L. Steinberg, 1975 “An adaptive algorithm for spatial grayscale,” in SID 75 Dig., Soc. Inf. Display, pp. 36–37.
3. M. S. Fu and O. C. L. Au, May 2000, “Hiding data in halftone image using modified data hiding error diffusion,” Proc. SPIE, Vis. Commun. Image Process., vol. 4067, pp. 1671–1680.
4. J. R. Goldschneider, E. A. Riskin, and P. W. Wong, “Embedded color error diffusion,” in Proc. IEEE Int. Conf. Image Process.,vol.1. Sep. 1996, pp. 565–568.
5. J. F. Jarvis, C. N. Judice, and W. H. Ninke, 1976, “A survey of techniques for the display of continuous tone pictures on bilevel displays,” Comput. Graph. Image Process., vol. 5, no.1, pp. 13–40.
6. S. H. Kim and J. P. Allebach, Mar. 2002 “Impact of HVS models on model-based halftoning,” IEEE Trans. Image Process., vol. 11, no. 3, pp. 258–269.
7. D. E. Knuth, Oct. 1987 “Digital halftones by dot diffusion,” ACM Trans. Graph., vol. 6, no. 4, pp. 245–273.
8. D. J. Lieberman and J. P. Allebach, Oct. 1997 “Efficient model based halftoning using direct binary search,” in Proc. IEEE Int. Conf. Image Process., vol. 1., pp. 775–778.
9. D. J. Lieberman and J. P. Allebach, Nov. 2000, “A dual interpretation for direct binary search and its implications for tone reproduction and texture quality,” IEEE Trans. Image Process., vol. 9, no. 11, pp. 1950–1963.
10. Mese and P. P. Vaidyanathan, Apr. 2000, “Optimized halftoning using dot diffusion and methods for inverse halftoning,” IEEE Trans. Image Process., vol. 9, no. 4, pp. 691–709.
11. P. Stucki, 1981, “MECCA—A multiple-error correcting computation algorithm for bilevel image hardcopy reproduction,” IBM Thomas J. Watson Res. Center, IBM Res. Lab., Zürich, Switzerland, Res. Rep. RZ1060.
12. R. Ulichney,1987, Digital Halftoning. Cambridge, MA, USA: MIT Press.
13. R. A. Ulichney, 1993, “Void-and-cluster method for dither array generation”, Proc. SPIE, Human Vis., Vis. Process., Digit. Displays IV, vol. 1913, pp. 332–343, Sep.