J Digit Imaging (2013) 26:65–71
DOI 10.1007/s10278-012-9472-8

Texture-Based Medical Image Compression

Vinayak K. Bairagi · Ashok M. Sapkal · Ankita Tapaswi

Published online: 2 May 2012

Abstract Image processing is one of the most researched areas these days due to the flooding of the internet with an overload of images. The noble medicine industry is not left untouched. It has also suffered with an excess of patient record storage and maintenance. With the advent of automation of the industries in the world, the medicine industry has sought to change and provide a more portable feel to it, leading to the fields of telemedicine and such. Our algorithm comes in handy in such scenarios where large amount of data needs to be transmitted over the network for perusal by another consultant. We aim for a visual quality approach in our algorithm rather than pixel-wise fidelity. We utilize parameters of edges and textures as the basic parameters in our compression algorithm.

Keywords Medical image compression · Texture · Edges · DICOM image · Telemedicine

Objective Address in This Paper The main objective is to achieve high compression using textural property of images. Even though many compression algorithms are readily available in the market, but for medical images, there is a requirement of special algorithm (discussed in paper). The proposed work in this paper is associated with texture-based medical image compression which is required for telemedicine application especially in a rural area where channel bandwidth is limited.

Essence of Paper The paper essentially is about image compression keeping visual quality into view. As is done in other compression schemes, the quality is not sacrificed at the altar of compression ratio. Our paper boasts of a dynamically varying compression ratio. This factor helps us to have a much larger compression when a series of images are involved as each image is taken for its face value. The reconstruction approach is by a prediction method which is built taking the HVS into account.

Salient Features 1. Dynamically varying compression ratio
2. Importance to visual quality
3. Reconstruction modeled on HVS

Manuscript received on 15 September 2011

V. K. Bairagi (✉)
Department of Electronics and Telecommunication,
Sinhgad Academy of Engineering,
Kondhwa (Bk.),
Pune 48, India
e-mail: vbairagi@yahoo.co.in

A. M. Sapkal
Department of Electronics & Tc Engineering,
College of Engineering,
Shivajinagar,
Pune -05, India
e-mail: amsapkal@yahoo.co.in

A. Tapaswi
Department of Electronics and Telecommunication,
VIT College of Engineering,
Vellore, India
e-mail: ankitazzz@ymail.com

Introduction

In today's world of internet and instant information, there is no scope for delay and redundancies. The relevance of the project emanates from the fact that even the most sacred medical profession has not remained untouched from the fever of technology. The fields of e-medicine and telemedicine are booming and need the fast and error-free communication of medical images to their destination to perform e-consultancy between various specialists to agree upon the correct diagnosis for the patient. Thus, the relevance of image compression remains to maintain the accuracy and visual quality of image while compressing it to reduce redundancies and fast transmission across the internet.

We use two parameters: (1) edges and (2) textures. Edges are the boundaries of objects, the boundaries of surface markings, as well as curves that correspond to discontinuities in surface orientation. Texture is that innate property of all surfaces that describes visual patterns, each having properties of homogeneity. It contains important information about the structural arrangement of the surfaces [1].

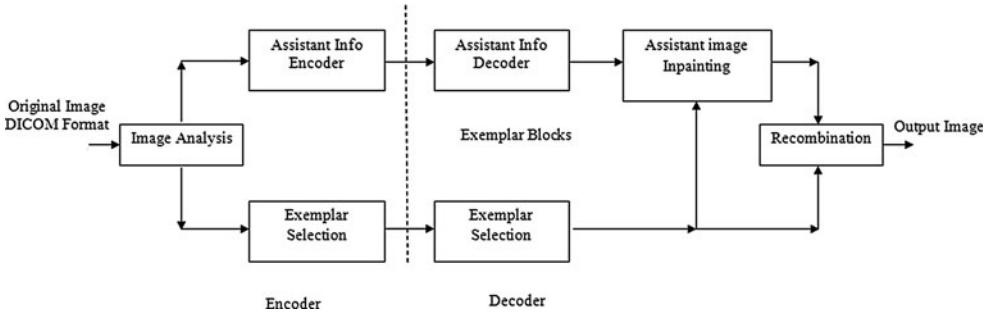

Fig. 1 Over all methodology of proposed compression system

Methodology

Image Encoder

This is performed according to the block diagram shown in Fig. 1. First, the edges of the image are extracted and coded. Then, certain blocks of the image are removed. These are called exemplars because they act as examples to the other blocks. The edges to these blocks are extracted and encoded. The rest of the image is encoded normally and then this entire information is transmitted over the channel and can be decoded at the receiver.

Image Restoration

Image restoration is performed on the basis of edge-based inpainting and texture synthesis [2]. Edges represent discontinuities while texture represents continuity. First, the edges are generated and thickened and then the filling in of these edges is done on the basis of the exemplar textures and thus the image is regenerated. And the removed blocks of the normally coded image are filled in. The normally coded image is decoded normally. While regenerating texture the patch farthest from the edge will be generated so as not to interfere with the edge regeneration [2].

Encoder Side

An uncompressed, i.e., a Digital Imaging and Communications in Medicine (DICOM), image is taken and processing is done on that. The first step is to read the image in MATLAB and detect its edges using some suitable algorithms. Here, we have selected Canny edge detection technique for this purpose.

Figure 2 shows the block diagram-wise representation of the encoder system.

Edge Detection

Edges are the high-frequency information present in an image. The method to obtain edges is to get their gradient. Many operators have been given to date for edge detection such as Sobel, Prewitts, Roberts, etc. Out of all of them, Canny is by far the best. Developed by John Canny in 1986, the Canny edge detector uses a multistage algorithm to detect a wide range of edges in images. Canny's aim was to discover the optimal edge detection algorithm. In this situation, an "optimal" edge detector means:

- *Good detection*: The algorithm should mark as many real edges in the image as possible.
- *Good localization*: Edges marked should be as close as possible to the edge in the real image.
- *Minimal response*: A given edge in the image should only be marked once, and where possible, image noise should not create false edges.

The disadvantages are as follows:

- *Noise influence:* The area surrounding the boundary of image should ideally have no edges. But due to noise, edges are detected in the surrounding.
- *Threshold level*: Every image has a different level of thresholding to have proper edge information. It is not possible to set a variable threshold in Canny. Especially due to this disadvantage, we cannot rely on Canny.

Thus, we need to define the threshold for each level. As we want to make our scheme as user independent as possible, we go in for dynamic thresholding [14]. We have implemented

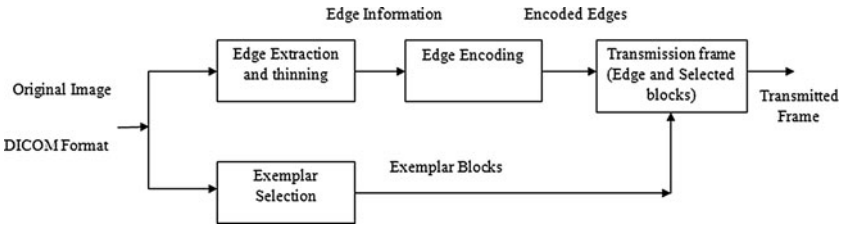

Fig. 2 Encoder of compression system

the edge detection algorithm proposed in [3]. To overcome the disadvantages in other algorithms, we use a topological-based algorithm. Input image is first smoothed by a two-dimensional isotropic Gaussian filter to avoid noise. For every pixel, Laplacian and θ values are found, where θ is the direction of the gradient. Spatially adopted threshold is implemented on pixels with non-maximum gradients to prevent missing edges. Edges extracted with this algorithm are often more than 1 pixel in width. So, edge thinning is required and implemented as proposed in [4].

Block Classification

Exemplar selection is performed to remove some blocks from an original image and remaining ones are regarded as exemplars. The exemplar selection is conducted at non-overlapped 8×8 block and classified into structural or textural. In specific, if a block contains more than one quarter pixels which are located within a short distance from edges, it will be regarded as structural, otherwise textural [3, 15].

Structural Exemplar Selection The selection of exemplars from structural blocks is accomplished in two steps. First, some blocks are regarded as *necessary* because inpainting can hardly restore them. Second, some *additional* blocks are selected in order to improve the visual quality of restored images. The blocks located at the end points or conjunctions of edges are regarded as necessary blocks because they contain the transition between different image partitions and thus are hard to be restored by inpainting [5]. For a circle edge, two necessary blocks are identified in its inner and outer regions. These two are selected to provide exemplars for inpainting. Moreover, additional structural blocks are selected to represent local variations. If a block contains obvious variation, it has the priority to be preserved. In practice, each block is assigned a variation parameter Vi defined as [3]:

$$\mathbf{Vi = w1\ Var(Bi) + \quad w2 \sum_{Bj \in \mu 4(Bi)} | E(Bi) - E(Bj)|,} \tag{1}$$

where w1 and w2 are weighting factors, μ4() indicates four neighbors of a certain block, and Var() and *E*() are the variance and the mean of pixel values, respectively [5].

Note, that in one block, the different partitions separated by edges are independent in calculating the variance and the mean; then, the resulting parameters of different partitions are summed up to get the total variation parameter of the block. Given an input threshold, the blocks with higher variation parameters will be selected as exemplars.

Textural Exemplar Selection The necessary textural blocks are selected at the border of textural regions. In specific, if a textural block is next to a structural one, it is considered as necessary. Such blocks are preserved because they can clearly separate textural blocks from structural ones and thus facilitate texture synthesis at the decoder. Additional blocks are also selected according to their variation parameters which can be calculated. The textural blocks with higher variation parameters will be preserved as exemplars. The exemplars are selected according to the statistical model of textures as given in [3]. In medical images, there is no other influence of other parameters on textures [5].

Texture Blocking and Texture Classification

The block classification and separation is done in this process. For these, following steps are followed:

1. The noise surrounding main subject body in the image is removed.
2. The image is resized and divided in blocks of 8×8 pixels.
3. These 8×8 pixel blocks are considered for texture blocking.
4. The texture pattern of first block is compared with the rest of the blocks and the image is checked for repetitive texture pattern.
5. Each texture pattern is assigned a code/number. If the texture pattern is repeated, then no new code/number is assigned to it.
6. This process is repeated for all the 8×8 blocks, covering complete image.
7. Thus, the image is represented in terms of texture codes.
8. A lookup table is created which consists of texture pattern block and code/number assigned to it.
9. The image represented with texture pattern code/number is encoded with algorithm.
10. In transmission packet, we pass:
    (a) Image dimensions
    (b) RLE-encoded image data
    (c) Texture pattern lookup table
11. At the decoder, we first separate the data from received packet.
12. Then, the RLE data of image were decoded, restoring the image's code matrix.
13. Then, a cell matrix is formed with each cell of 8×8 pixel size.
14. According to the texture pattern code in image's code matrix, respective texture pattern is placed in the cell matrix with the help of lookup table.

Image dimensions	Cropped image dimension	Noise removed re-cropped dimension	RLE encoded structural information	Selected textural Blocks +index of the blocks

Fig. 3 Structure of transmission packets

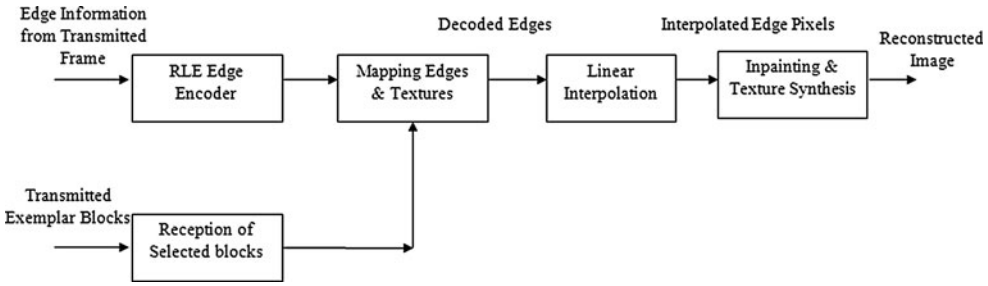

Fig. 4 Decoder block diagram

The cell matrix is then converted to normal matrix form. This matrix represents the decoded image. The decoded image can be viewed, and the image can be saved in any desired format.

Another approach was to classify texture blocks and making a database of the different texture sample blocks. In texture blocking, we extract texture patterns from the image and use these patterns to encode (compress) and restore the image [6]. But as we go for such approach, the size of the database is huge and defeats the entire purpose of image compression. Also, there are very few repeating blocks in a medical image. Another approach would be to utilize the symmetry in an image and only encode half the image about the axis of symmetry.

Edge Encoding

The structural information of an image is nothing but edges. Instead of transmitting whole edge matrix which takes large time and space, we transmit the coded information and decode it at the receiver side. Edge encoding and decoding is done using one of the simplest yet effective method known as RLE. We have proposed this step to reduce the size of the edge information by utilizing the redundancy in the edges. Edges are represented by only two levels, i.e., 0 and 1, hence we get good encoding using the RLE. RLE is a very simple form of data compression in which runs of data, i.e., the sequences in which the same data value occurs in many consecutive data elements, are stored as a single data value and count, rather than as the original run. This is most useful on data that contains many of such runs and therefore can be successfully implemented for our requirement.

Textural Block Passing

The textural blocks selected are transmitted as it is. Implementation of any encoding technique gives variation in gray scale values. So to preserve required information, they are transmitted completely.

Arithmetic Encoding

The exemplar blocks transmitted by us in our methodology are sent without any compression. The values in the blocks

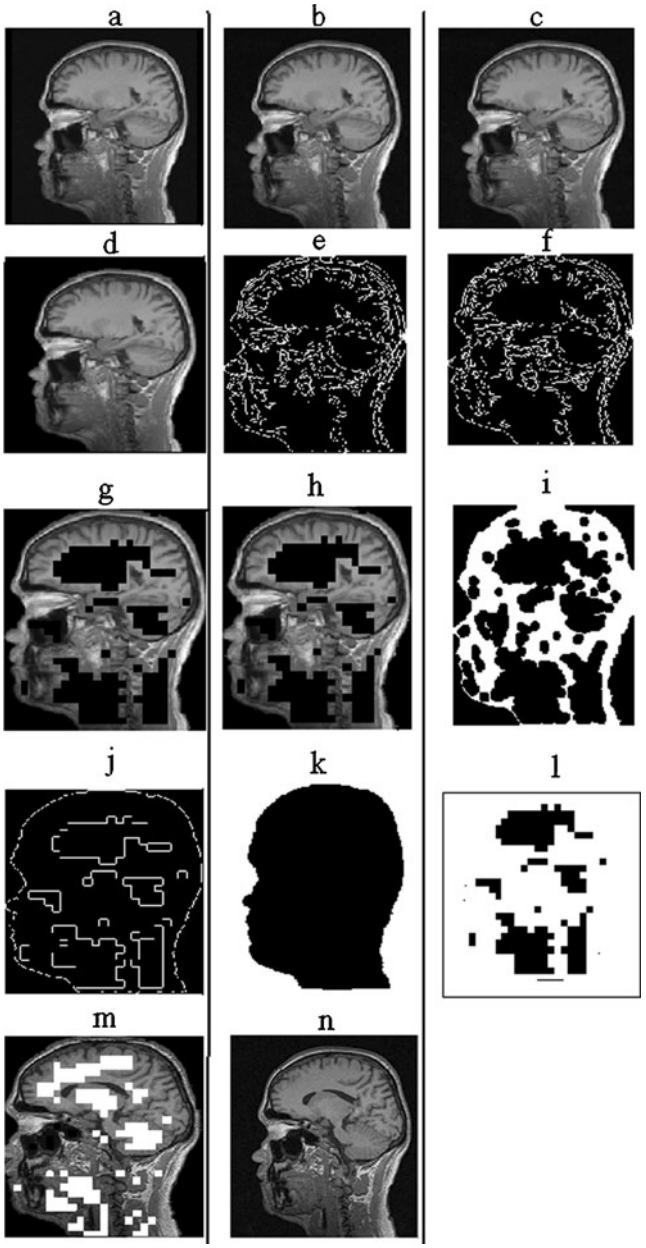

Fig. 5 Various in-between stages. Left to right and top to bottom in zigzag manner: **a** original image, **b** cropped image, **c** image smoothening, **d** noise removal, **e** structural information, **f** RLE edge decoding, **g** mapping of edges and texture, **h** linear interpolation, **i** influencing region in an image, **j** morphological information, **k** background variation, **l** selection of patches, **m** coloring the selected patches [13], and **n** final restored image

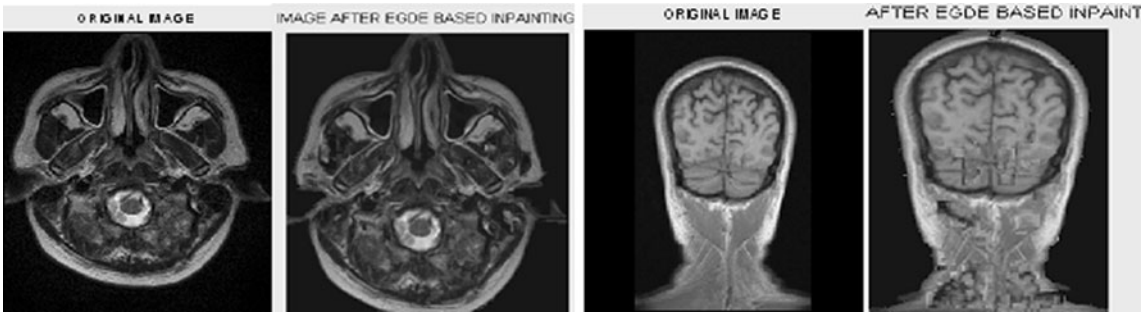

Fig. 6 **a**, **b** Results on various images. Note that the values given are for the image used in stepwise execution explanation as shown in Fig. 5 and verified for few more images as shown in Fig. 6

range from 0 to 255, hence we use 8 bits to represent them. As an advancement to our method, we would like to propose the utilization of arithmetic coding as an encoding scheme for the transmitted exemplars. Arithmetic coding is a lossless variable length entropy encoding scheme used for data compression. Arithmetic coding scheme utilizes the repetition of the values as a measure of encoding. The values which are repeated most require the least number of bits when encoded. The main feature differentiating the arithmetic coding scheme from other entropy encoding schemes is that it treats an entire message as a whole and does not separate a message into components to be replaced by codes rather it encodes the message together [7].

Transmission Frame

A transmission frame is constructed in which all the information to be sent is put in a single frame as shown in Fig. 3. At the decoder side, we perform image restoration based on the exemplar information available with us from the transmission frame. The pixels on the edge are restoration using linear interpolation, and the pixels near the edge are restoration using edge-based inpainting approach (EBI). Also, to restore the central pixels, we utilize texture synthesis approach [3].

Table 1 Specifications of system with respect to results on image shown in Fig. 6a

Description	Value
Input image size	256×240 pixels, 177 KB
Cropped image size	255×239 pixels, 68.2 KB
Noise removed image size	255×239 pixels, 50.9 KB
Re-cropped image size	240×208 pixels, 50.4 KB
Edge information	240×208 pixels, 4.9 KB
Exemplar information	1×34,368
Tx frame	1×13, 44.8 KB
Encoder time	14.2273 s
Decoder time	99.6375 s

Decoder Side

RLE edge decoder is similar to encoder. Figure 4 shows the blocks on the decoder side. It receives the encoded information and converts it to the original format thereby enabling restoration [3].

Blocks Restoration

Likewise, transmitted blocks are restored with the help of index received at decoder. The decoding of block makes restoration possible. The edges and texture restored in the decoders are merged. So, we get a layout for restoring the remaining information blocks. When transmitting the exemplar blocks, we also transmit the index of the selected blocks to ease us in the mapping of the texture and edge blocks at the decoder side [3].

Linear Interpolation

When the edges are received and mapped, we only know a certain number of pixels on the edges because we are

Fig. 7 Graphical user interface of application

passing only a limited number of exemplar blocks. To restore the remaining pixels on the edges, we go for linear interpolation. Interpolation is a technique for approximation of the gray scale value of unknown pixels by taking into account the known discrete samples. It is an approximation technique, hence we do not obtain exact values of the pixels, but as we are aiming for visual quality, we do not pay much heed to such issues [8].

Inpainting and Texture Synthesis

Different from the previous work on inpainting or synthesis, our image restoration method tries to fully utilize the transmitted edges. In specific, EBI is performed to restore the pixel values on the edges [9, 10]. This method makes use of discrete wavelet transform and neighboring the edges and texture synthesis is utilized for the restoration of textural regions [3]. Any DICOM image can be inputted with a bit depth of 16 bits and the output format is bit stream [16].

Results and Observations

The proposed scheme is tested on several images (more than 300) (available from private domain database as well as form public domain images), as obtained from online databases [11, 12]. According to our observation, the tumor portion of the image is not lost at all because the edge of the tumor is stored because it has high variation value and the textural blocks are synthesized. Thus, mainly the location and size of the tumor are not hampered at all (Figs. 5 and 6).

Transmission frame and other details of compression system are shown in Table 1. It is clear that compression is achieved and is equals to 25.31 % of original file size. MATLAB has many advantages over other programming languages. MATLAB facilitates us to build a friendly graphical user interface. We have made our process a standalone application. The GUI window is shown in Fig. 7.

Complexity The computational complexity describes how complicated a process can be. To find the computational complexity of a system, we must know about hardware and software complexity of that system. Since our approach is based on software only, its computational complexity depends on the code and the number of loops, conditional statements, etc. For a system like this, it is very tough to predict the computational complexity. Time complexity is defined as the time required in executing a process. This time can be found out by using tic-toc method. This is the easiest method for finding time required to run each code.

Conclusions

A compression approach towards visual quality rather than pixel-wise fidelity has been pursued. It takes an advantage of inpainting as well as texture parameters. Some blocks are intentionally removed during encoding, while simple edges are transmitted as assistant information. To fully utilize the edges, we design edge-based inpainting and adopt texture synthesis to restore the removed blocks. This algorithm has been implemented by us in a generalized way to compress any given medical image. It can be further extended to a series of images utilizing their common features as a basis of compression along with already used parameters. Also, this method can be made intelligent and maybe utilized for region of interest detection based on texture information leading to further compression. Further, the exemplar blocks can be encoded thus leading to more compression ratio. Also, as the techniques for better restoration come up, we need to include these in our scheme because we are dealing with medical image where no restoration is good enough and we need to always do better. The noise removal technique used for background noise removal needs to be improved to act more efficiently.

Acknowledgments This work was financially supported in part by the Pune University, Pune, India under research grant EN-61-2007. The authors also would like to thank the Sinhgad General Hospital, Pune for the valuable help and support and the authors of the references which have been used, as well as the reviewers of the paper.

References

1. Panayiotis D. Korfiatis, Anna N. Karahaliou, Alexandra D. Kazantzi, "Texture-Based Identification and Characterization of Interstitial Pneumonia Patterns in Lung Multidetector CT", IEEE Trans Inf Technol Biomed, 14(3): 675–680, 2010.
2. Zongben Xu and Jian Sun, "Image Inpainting by Patch Propagation Using Patch Sparsity", IEEE Trans Image Process, 19(5):1153–1165, 2010.
3. Dong Liu, Xiaoyan Sun and Feng Wu, "Edge-Based Inpainting And Texture Synthesis For Image Compression", Conf Proc ICME 2007, pp 1443–1446, 2007.
4. Available at http://homepages.inf.ed.ac.uk/rbf/HIPR2/thin.htm. Accessed May 2011.
5. Dong Liu, Xiaoyan Sun, Feng Wu, Shipeng Li and Ya-Qin Zhang "Image Compression With Edge Based Inpainting", IEEE Trans. Circuits Syst. Video Technol, 17(10):1273–1287, 2007.
6. P. P´erez, M. Gangnet, and A. Blake, "Poisson image editing," in *Proc. ACM Siggraph*, pp. 313–318, 2003.
7. X. Sun, F. Wu, and S. Li, "Compression with vision technologies," in *Proc. Picture Coding Symposium, PCS*, pp 1–5, 2006.
8. M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. ACM Siggraph*, 35 (4): 417–424, 2000.
9. S. D. Rane, G. Sapiro, and M. Bertalmio, "Structure and texture filling-in of missing image blocks in wireless transmission and compression applications," *IEEE Trans. Image Process.*, 12 (3):296–303, 2003.

10. M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Process.*, 12(8):882–889, 2003.
11. Computer Vision Group [online]. Available at, http://decsai.ugr.es/cvg/index2.php. Accessed Aug 2011.
12. The National Library of Medicine [NLM] [online]. Available at, http://www.nlm.nih.gov. Accessed July 2011.
13. C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Trans. Image Process.*, 10(8):1200–1211, 2001.
14. R. Medina-Carnicer, A. Carmona-Poyato, R. Muñoz-Salinas, and F. J. Madrid-Cuevas, "Determining Hysteresis Thresholds for Edge Detection by Combining the Advantages and Disadvantages of Thresholding Methods", IEEE Trans Image Process, 19(1):165–173, 2010.
15. S. Ebenezer Juliet, D. Jemi Florinabel, "Efficient block prediction-based coding of computer screen images with precise block classification", IET Image Process, 5(4):306–314, 2011.
16. V. Kwatra, A. Sch¨odl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: image and video synthesis using graph cuts," in *Proc. ACM Siggraph*, pp. 277–286, 2003