Progressive Image Transmission Using Edge Detection

Awad Kh. Al-Asmari
Electrical Eng. Dept., College of Engineering
King Saud University, Riyadh 11421, Saudi Arabia
E-mail: akasmari@ksu.edu.sa

ABSTRACT

In this paper, a progressive image compression and transmission using edge detection scheme is adopted. The image is decomposed into two (primary and secondary) components. Canny method is adopted to detect the edges of the encoded image. These edges are replaced with a pre-designed nine basis nameplates. Then, the Macro edge detection technique is used to reduce the number of these nameplates and keep only the edges that are necessary for visual quality.

Eight directional predictions and interpolation technique will be performed on the encoded edges to reconstruct the first layer of the primary component at both receiver and transmitter sides. This is called the 1st stage reconstructed image, which is subtracted from the original to have the 1st stage smooth component. Then, this process will be repeated for the 2nd and the 3rd stage components.

The 3rd stage smooth component is filtered using an optimal decomposition filter and then decimated by a factor of 2. The decimated component is encoded using VQ. The decoded result of the smooth image is added to the three layers that form the primary component to have the reconstructed image. An excellent reconstructed images are found at an average of 0.179 bpp (compression ratio 45:1) and with an average PSNR of 39.45 dB. This algorithm is found to be of lower bit rate than existing image compression techniques.

Keywords: Progressive Image Transmission, Edge Detection, Vector Quantization Coding.

1. INTRODUCTION

Over the past decade, a new class of image coding schemes, generally referred to as second generation coding techniques, has been developed. Hence, the Second generation coding schemes can be expected to achieve higher compression ratios with human eye oriented fidelity criterion, which has been studied and emerged with the compression algorithms. The coding model employed for edge detection in the second generation coding schemes is based on the multi-component source model [1]. It decomposes the image into primary component, which contains the edge information, and secondary component, which represents the slow intensity variations. A lot of approaches put emphasis on the edge detection, representation and coding, which has a significant impact on the quality of the reconstructed image. Most of the edge detection schemes are thwarted by gaps in the data produced by local noise and readily follows by spurious boundaries, as well as the location errors of extracted edges, which will lead to edge discontinuity and incorrect intensity values [2]. Itoh in [3] has adopted Laplacian edge-detection method with thresholding. We find that his technique will not detect the edges for areas with small intensities variation in the image.

In this paper, we adopt Canny method for edge detection, which it depends on zero crossing method for detecting edges even at small intensities variation. The high performance of the recursive version makes it more interesting for general applications. The method can be seen as a smoothing filtering performed with a liner combination of exponential functions, followed by derivative operations. The size of the exponential filter is related to the width of gray level transition region, as well as to the noise level in the image.

Itoh has adopted the discrete cosine transform (DCT) to encode the secondary (smooth) image [3]. In this paper, the Vector Quantization (VQ) with self-generation for the codebook is used for the main nameplates. To farther reduce the average bit rate, the Huffman coding is adopted for the codebook indices. In this paper, for the test images, the primary component is reconstructed at an average bit rate of 0.072 bits/pixel (bpp) with same visual quality as that reconstructed with other existed algorithms at twice the bit rate required by our proposed technique.

The progressive image transmission approach is adopted to reconstruct the image. Edge-detection scheme involves an approximate reconstruction of the image whose fidelity is built up gradually until either the viewer decides to abort the transmission or perfect reconstruction is achieved. The applications benefiting from this are in transmitting the images over low bandwidth channels such as wireless channels and telephone lines. Examples for such applications are access to remote image databases, teleconferencing, electronic shopping, and security systems. For these applications it is important for the viewer to recognize the content as early as possible in the transmission.

In section two, the extracting of unit edges is discussed. The macro edge extracting is presented in section three. The image reconstruction process is discussed in section 4. In section 5, the codebook design for the primary and secondary component is presented. The simulation results for the proposed compression technique are discussed in section 6. The conclusion is given in section 7.

2. Extracting unit edges

The edges will be detected by using Canny method. The scheme for unit edge detection is shown in Figure 1. A sub-blocks of 5x5 will be taken from the macro edge image and correlated with the first eight nameplates, which are shown in Figure 2 according to the correlation equation given below

$$R_n\ (x,y) = \sum_j \sum_k \lambda\ (x+j, y+k)\ t_n\ (j,k) \qquad (1)$$

Where:

n : is an integer number equal to 0,1,2, ...7.

J,K : is an integer number depends on the size of the nameplate equal to 0,1,..4.

λ : is the sub-block taken from the image after thresholding.

t_n : is one of the eight nameplates.

Subsequently, if there exits n such that R_n (x,y) greater than or equal to a predefined threshold (P_{th}), a unit edge will be detected and replaced with the nameplate that satisfy this criteria.

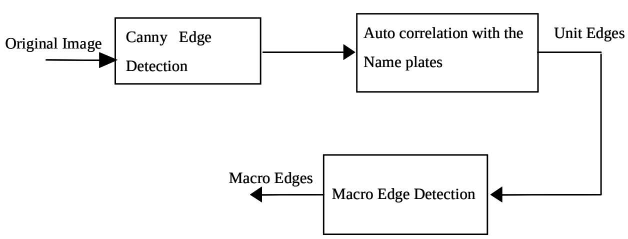


Figure 1. The unit edge detection process.

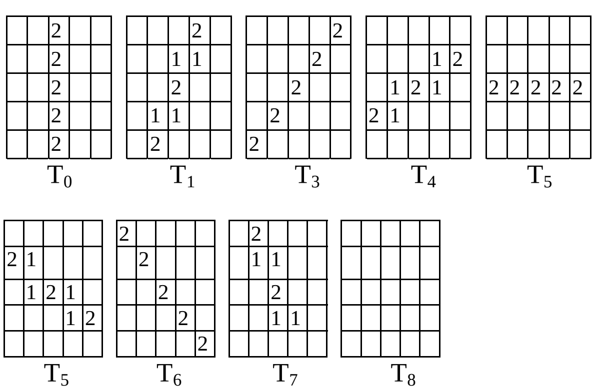


Figure 2. Eight directional small segment patterns.

3. Extracting Macro edge

After extracting unit edges, macro edge detection is performed. These macro edges are defined into sixteen directions i.e. in steps of 11.25 degrees. Let W_N be an element vector whose direction is N as shown in equations 3 and 4; respectively. The aim of detecting Macro edges is to reduce the number of gaps, to give smoothing for interpolation process, and to reduce the number of edges in the unit edge image to be encoded. Also, to keep these edges which are necessary for vision.

$$W_N=(Wx_N,Wy_N) \text{ where } N=0,1,2,\dots,15 \qquad (3)$$

These directions are:

$$\begin{array}{llll} W_0 = (0,4) & W_1 = (-1,4) & W_2 = (-2,4) & W_3 = (-3,4) \\ W_4 = (-4,4) & W_5 = (-4,3) & W_6 = (-4,2) & W_7 = (-4,1) \\ W_8 = (4,0) & W_9 = (4,1) & W_{10} = (4,2) & W_{11} = (4,3) \\ W_{12} = (4,4) & W_{13} = (3,4) & W_{14} = (2,4) & W_{15} = (1,4) \end{array} \qquad (4)$$

The starting point of the detection can be defined as a pixel on which a flag of any bit plane is active. The direction for the detection process is defined as N=2n, where n is equal to the nameplate number. Since it is risky to predetermine the direction of the macro edge before the search operation, the most likely direction among three possible directions is selected. There are N, N-1, N+1 directions. In each possible direction, it is determined at each connection point, which is located every four pixels along each direction whether, the macro edge is connected or not. The starting point in the direction N is denoted by (X_S,Y_S), and the connection point by (X^i_c,Y_c^i). The relation ship between the connection point and the starting point will be given by the following equations:

$$\begin{aligned} X_c^{\ i} &= X_s + iWx_N \\ Y_c^{\ i} &= Y_s + iWy_N \end{aligned} \qquad (5)$$

The connection will be extended up to the point if it is active or one of its neighboring pixels are active. After calculating the three coordination points in X-axis and Y-axis using equation 5, the Euclidean distance will be calculated to choose the longest path. The criterion for the connection is that the Macro edge is extended up to the connection point if a flag in the bit planes n, (N-1)/2, (N+1)/2 is active on the connection point or in the vicinity of the eight neighboring pixels. Based on calculating the Euclidean distance, the longest will be chosen among the three candidate macro edges. On the other hand, the nameplates perpendicular to the macro edge detected will be neutralized [3].

4. Image Reconstruction

At the receiver, the transmitted indices of the primary component will be decoded. The mean of the original image, which represents the DC of the original signal, will be transmitted to be used for the reconstruction of the image. The received 1 and 2 of the encoded sub-block will be substituted by half of the mean for 2 and the mean for 1. After replacing this in the whole image, two operations will follow. These operations are interpolation and eight directional predictions. For interpolation an active pixel either half of the mean or the mean will be interpolated with another active pixel on the three directions 0 degrees, 45 degrees, and 90 degrees. To smoothen the image, eight directional prediction will be performed.

Let V_i be a reference pixel and α_i be the distance from the predicted pixel which are denoted by ψ, the gray level obtained at the predicted pixel is given by equation 6. This process will be performed recursively to the whole image. At the transmitter, the difference between the original image and the primary reconstructed component will be obtained. Then, the previous process will be repeated on the difference image (smooth component).

$$\psi = \frac{\sum_{i=0}^{7} V_i \alpha_i^{-1}}{\sum_{i=0}^{7} \alpha_i^{-1}} \qquad (6)$$

For further simplification, the processed components are denoted as follows:

e_i : is the i^{th} stage encoded image.
R_i : is the i^{th} stage reconstructed image.
S_i : is the i^{th} smooth component.
Where, i=1, 2, 3,…..N.

In this paper, the number of stages (N) is selected to be 3. This selection is based on the simulations for different values of N. For good quality image, reconstruction, and low complexity encoder, N=3 is found to be a suitable selection. The number of stages (N) is plotted against the SNR for Wltr01 image of size 512 x 512. Increasing the number of stages to have N = 10, will give maximum PSNR around 25 dB and the image quality is not acceptable. The PSNR at N = 3 is about 20 dB. For N > 3 the rate of change of SNR is very small and its around 1 dB per stage. This will increase both the (delay) transmission time and the algorithm complexity.

The overall scheme for encoding the edge and reconstruction the primary component of the image at the transmitter is shown in Figure 3.

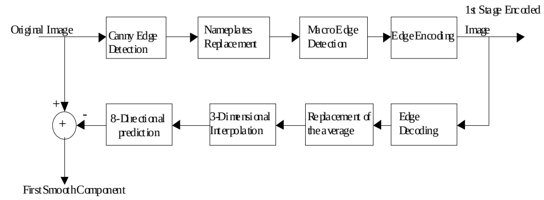
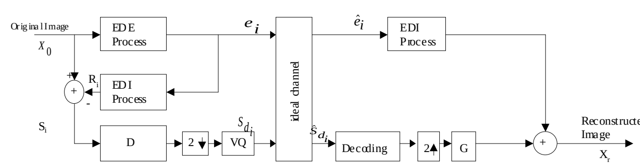


Figure 3. The edge detection encoding/decoding process.

A. Decimation of the smooth component

By progressively extracting the edge of an image, the smooth component will almost approach the DC of the original image. This will make the pixels closer to each other in their values in a certain region. Therefore, the decimation and interpolation technique can be used in this case. There are many techniques in the literature used for image decomposition. In this case, the concept of pyramidal coding will be adopted. For this kind of image, there is an optimal choice which is well suited for an image of DC nature.

The smooth component will be decimated first by a minimum average filter which has been introduced first in [4]. The design of this filter is based on the idea that the ideal decimation filter is a low pass filter with a cut off frequency at π/2, unity gain in the passband (PB). The ideal interpolation filter is the same as the decimation filter, except it must have a gain of two in the pass band (PB). A constraint is introduced to insure that the decimation filter will have an ideal DC response. It is found that the size of the interpolation filter is 3 x 3 with center coefficient equal to unity. This property together with the fact that the decimation filter is a unit impulse will ensure that the pixels sub-sampled from the original image in the decimation process are interpolated to their original values in the interpolation process. Figure 4 shows the overall encoding/decoding process for the proposed algorithm.



D : Decimation filter G: Interpolation filter

Figure 4. The encoder/decoder of the proposed algorithm.

Where,
EDE : is the edge detection and encoding process.
EDI : is the edge decoding and interpolation process.
e_i : is the i^{th} stage encoded image.
$\hat{e}_i$: is the i^{th} received primary component.
R_i : is the i^{th} stage reconstructed primary component

S_i : is the i[th] smooth component.
S_{di} : is the decimated smooth component.
$\hat{S}_{d_i}$: is the received smooth component.
X_r : is the reconstructed image.

The decimated smooth component (S_{di}) will be encoded using VQ. In this figure, the primary component is the combination of the three stages reconstructed images. The indices are decoded to reconstruct the decimated smooth component and then interpolating it to be filtered, using the optimal interpolation filter, to have the same size as the primary component. Then, the decoded smooth and primary components are added to form the reconstructed image (X_r) as shown in Figure 4.

5. Codebook Design for Primary and Smooth Components

Eight different test images are used to detect the macro edges which are used to design the codebook for the primary component. Two neighbor nameplates are assumed to be the codeword of the designed codebook. The size of the codeword will be a sub-block of size 5x10 from any two combinations of the nine nameplates. The size of the codebook will be 9^2, which are self generated from the original nameplates. A sub-block of 5x10 from the test image will be taken, and then will be compared with the eighty one possible combinations of the codebook. One of the codebook entries will give an exact representation of the input sub-block. However, the rate is reduced by using Huffman code concept. This can be achieved by designing a counter to calculate how many times each possible combinations of the nameplate occur. The eight images are used to design the Huffman code. From the simulation results, it is found that most of the nameplates combinations are of zeros or ones occurrences. Therefore, this codebook can be reduced by discarding those combinations with number of count less than or equal to 1. As a result of this, the entries of the codebook is reduced to be 16 codewords only, which will reduce the bit rate requirement for the first stage encoded image to be about (4/50) bpp. This rate can be further reduced by designing a Huffman code for the 16 code vectors.

The Difference image or the smooth component is decimated by 2 and coded by using LBG algorithm. The procedure includes first initializing the codebook and application of Lyloid (LBG) algorithm to the initialized codebook. Initializing the codebook is a crucial part in designing any codebook. A large number of training set of vectors are taken to represent the smooth images (Primary Components). The codebook size is selected to be 256 vectors with codeword (vector) size of 16 pixels each.

6. Simulation results

In this paper, The test images are of dimension 512 x 512 at 8 bpp each. The original images are listed in Figure 5 (a) - (c). These images are Inventors, Ksu-boy2, Wltr01; respectively. The proposed algorithm has been applied using different edge detection techniques such as Sobel, Roberts, Prewitt, and Canny approaches. Canny edge detection method has the highest performance in terms of PSNR (dB), compression ratio, and visual quality. Figure 5 (d) – (f) shows the 3[rd] stage reconstructed primary image (R_3) at an average PSNR of 15.70 dB and at an average bit rate of 0.072 bpp. Table 1 shows the PSNR (dB), and the bit rate for the primary and the reconstructed images.

Table 1: The bit rate and the PSNR (dB) for the primary and reconstructed images.

Image	Inventors	Ksu-boy2	Wltr01	Average
Primary image (R_3) bpp	0.076	0.072	0.069	0.072
Primary image (R_3) PSNR (dB)	14.71	15.51	17.43	15.70
Reconstructed Image (X_r) bpp	0.186	0.175	0.176	0.179
Reconstructed Image (X_r) PSNR (dB)	36.32	39.29	42.75	39.45

Figure (g) – (k) shows the reconstructed smooth component. Figure (i) – (m) shows the reconstructed images at an average bit rate of 0.179 bpp (compression ratio of 45:1) and with an average PSNR of 39.45 dB. The visual quality of the reconstructed images are excellent as shown in Figure 5.

A. compression with other algorithms

The performance of the proposed algorithm is compared with other works using edge detection concept for compression. The comparison is in terms of the PSNR (dB), bit rate (bpp), visual quality, and the relative complexity. Kumar *etc* in [5] used fractal based coding scheme suitable for progressive transmission in four stages. For Lena image of size 512 x 512, the first stage is transmitted with a bit rate of 0.29 bpp and PSNR of 23.35 dB, the second stage is transmitted with bit rate of 0.39 bpp and PSNR of 27.7 dB and the total reconstructed image is reconstructed at a bit rate of 0.71 bpp and PSNR of 30.7 dB. The complexity of this scheme is considered to be high. Itoh in [3] has suggested a scheme based on edge detection concept suitable for progressive image transmission. The smooth component is transmitted using DCT of variable block size. For Lena image, the primary component is reconstructed at a bit rate of 0.048 bpp and the image is reconstructed at a bit rate of 0.23 bpp and

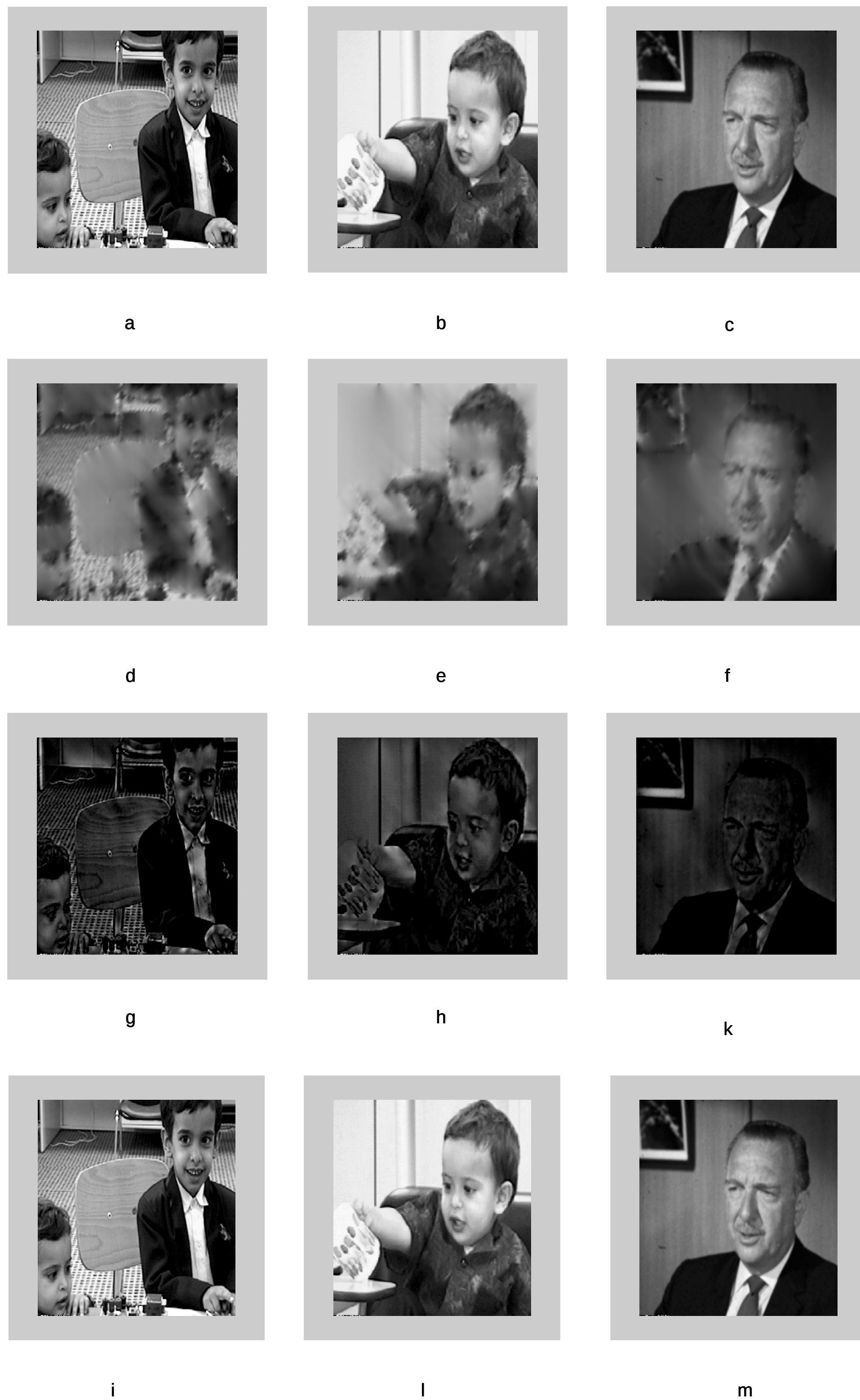


Figure 5. The original Images: (a) Inventers, (b) Ksu-boy2, and (c) Wltr01;
The reconstructed primary component (R3): (d) – (f);
The reconstructed smooth component (S3): (g) – (h);
The reconstructed images (Xr): (i) – (m).

PSNR of 38 dB. The relative complexity of this scheme is high because of the DCT transform.

The proposed algorithm is also tested for Lena image. The 1st stage of the primary component is reconstructed at a bit rate of 0.0226 bpp and PSNR of 17 dB. This bit rate is half of that required in [3]. The visual quality of the primary component compared to that in [3] is the same. The image is reconstructed at a bit rate of 0.177 bpp and PSNR of 39 dB. Also, the visual quality of the reconstructed image is very high. The proposed algorithm has low relative complexity compared to the other schemes.

7. Conclusions

In this paper, we have described a progressive hybrid approach for compressing gray scale images based on Canny edge detection method. Reconstruction is achieved by adding the compressed primary component to the compressed secondary component. We have suggested using the VQ and Huffman code to encode the nameplates of the primary component. The smooth component has been decimated using a minimum average entropy filter and then encoded using VQ approach. The compression ratio and the image quality of this algorithm is competitive to other existing methods given in the literature [3] and [5]. The compression ratio for our algorithm is about 45:1 (0.180 bpp).

References

[1] J. K. Yan and D. J. Sakrison, "Encoding of images based on a two component source model", IEEE Trans. Commun. vol COM-25, pp.1315-1322, Nov.1997.

[2] M. Kunt, A. Ikonomo polos, and M. Kocher, "Second Generation Image coding techniques", Proc. IEEE, vol.73, pp. 549-574, April, 1985.

[3] Y. Itoh, "An Edge Oriented Progressive Image Coding", IEEE Trans. Circuit and Systems for Video Technology, vol. 6., no.2, April, 1996.

[4] D. Houlding and J. Vaisey, " Low entropy image pyramids for efficient lossless coding ", IEEE Trans., Image Processing, vol. 4, on. 8, August, 1995.

[5] S. Kumar, R. C. Jain, "Low Complexity Fractal-based Image Compression Techniques", IEEE Trans., Image Processing, April, 1997.