Abstract—Image Compression is an important issue in Internet, mobile communication, digital library, digital photography, multimedia, teleconferencing and other applications. Application areas of Image Compression would focus on the problem of optimizing storage space and transmission bandwidth. In this paper, a modified form of skip line encoding is proposed to further reduce the redundancy in the image. The performance is found to be better than the skip-line encoding.

Index Terms—binary images, image compression, RLE encoding, skip-line encoding.

I. INTRODUCTION

Digital images require large number of bits to represent it and image compression is used to minimize the amount of memory required to represent an image. The objective of image compression is to reduce redundancy of the image data in order to store or transmit data in an efficient form. Image compression techniques are broadly classified into lossy compression and lossless compression. In lossless compression, the reconstructed image is same as the original image as there is no loss of information during compression and decompression process. In lossy compression, some loss of information is permitted without affecting the visual quality of the image. Benefits of image compression includes low storage space requirements, low communication bandwidth requirements, quicker sending and receiving of images, and less time on image viewing and loading.

II. RELATED WORKS

A good number of image compression methods has been proposed by different researchers based on techniques like dividing image into blocks [1, 2, 3, 4, 5, 6, 7], compressions based on edge detection [8, 9], chain coding [10, 11], skip line encoding [12, 13] etc. These methods exploit the spatial redundancy or correlation between pixels in the image. There are a number of compression techniques performed in frequency domain or other transformed domains. Some of the work belong to the following four categories are briefly reviewed in the following:

i. Block truncation coding
ii. Compression Based on Edge or Block Detection
iii. Chain Coding
iv. Skip Line Encoding

A. Block Truncation Coding

The basic Block Truncation Coding (BTC) is proposed by Edward J. Delp [1]. It is a lossy fixed length compression that uses a Q level quantization to quantize a local region of image. BTC preserves the sample mean and standard deviation of a gray scale image. The first step of the algorithm is to divide image into non overlapping rectangular regions. Two luminance values are selected to represent each pixel in the block. These values are chosen such that the sample mean and standard deviation of the reconstructed block are identical to those of the original block. Decompression algorithm knows whether a pixel is brighter or darker than the average by knowing the bit map for each block. Simplicity in implementation is the main feature of BTC. Disadvantages are that the boundaries of adjacent blocks can sometimes be visible. In some images edges may appear to be ragged despite being sharp and some sloping gray levels may exhibit false contours.

O. R. Mitchell et al. proposed Multilevel Graphics Representation Using Block Truncation Coding [2]. They proposed a modified version of a non information preserving coding technique known as BTC. This method is suitable for the coding of images in which many gray levels are present. In multilevel graphics, a low pass (smoothing) operation followed by thresholding at the sample mean gives a bit plane which results in a reconstructed image of lowest mean square error. The method is based on local binarization of non overlapping blocks in the graphics image such that the mean and sample variance in each block is preserved.

A New Version of Region Based BTC is proposed by J. Polec et al. [3]. It is a content related image compression technique. Here they used a multi level thresholding algorithm and median filtering to divide the image into quasi homogeneous regions. A region based truncation code related to BTC is applied to interiors of each segment.

C.K. Yang et al. [4] proposed a technique for improving block truncation coding by line and edge information and adaptive bit plane selection. A set of line and edge bit plane is defined corresponding to the visual continuity and discontinuity constraints. The line and edge bit planes are defined independently of the image to be coded. The set of the entire best match predefined bit plane is classified and those types which occur more frequently in the set are picked out. Each block is finally coded by the values of the mean, the standard deviation and the index of the best match.
predefined bit plane. The compression results with good bit rates and reasonable reconstructed image quality.

Some of the recent works based on Block Truncation Coding incorporating fuzzy edges and Huffman codes are those proposed by T. M. Amarunnishad et al. [5, 6, 7]. In [5] and [6], BTC performance is enhanced employing fuzzy edge operator. In [7], BTC performance improvement is attempted making use of Huffman codes to compress bit planes numbers. In all the cases, substantial improvements in compression performances are reported.

B. Compression Based on Edge or Block Detection

Aggoun and El-Mabrouk [8] presented an image compression algorithm using local edge detection. In this image is classified into visually active and visually continuous blocks which are then coded individually. It is based on a local histogram analysis. The differential image is obtained by applying a gradient operator. This image is divided into 4 x 4 non overlapping blocks and local histogram analysis is used to distinguish between visually continuous and visually active blocks. This algorithm works well for images with limited amount of texture and suitable to wireless image communications, where low power consumption is highly desirable.

U.Y. Desai et al. [9] proposed Edge and Mean Based Image Compression algorithm for applications with very low bit rate. Spatial redundancy is reduced by extracting and encoding edge and mean information. Sobel operator is used for edge detection. A thinning operation is done to produce edges that are single pixel wide.

This edge and mean based algorithm produces good quality images at very low bit rates. Sharp edges are very well preserved. It does not perform any texture coding, so there is some loss of texture.

C. Chain Coding

Lossless Chain Coder for Gray Edge Images is proposed by R. Redondo et al. [10]. This method provides lossless compression rates higher than the commonly used lossless methods. The method is based on searching pixels whose intensity is greater than zero and then storing the amplitude of their intensities and required movements to reach them. The image is scanned from left to right and from top to bottom until a non zero pixel is found. This pixel is called head of the chain. The searching process is performed clockwise and all stored pixels are marked for avoiding multiple coding. Huffman prefix codes and arithmetic codes are applied after the processing in order to obtain the final compressed code stream.

E. Tamir et al. [11] proposed an efficient chain code encoding for segmentation-based image compression. In this, regions and lines are obtained as result of image segmentation. Lines and contours of uniform regions are encoded using chain code. The chain code is obtained in a way that is efficient with respect to bit rate and produces lossless contour and line encoding.

D. Skip Line Encoding

Skip-n-line encoding is proposed by A.A. Moinuddin, et al, in 1997 [12].

If there is a high degree of correlation between successive scan lines, then there is no need to code each of them. Only one of them needs to be coded and the others may be skipped. While decoding, skipped lines are taken to be identical to the previous line. Lossless compression technique (Run length encoding) is used to compress the stored scan line. Skip one line encoding (S1LC) means if two successive scan lines are similar then code only one of them and skip the next line. That means in the best case only half of the total scan lines need to be coded. An example for using S1LC is shown in figure 1 and figure 2 shows its decompression process. Similarly S2LC means code one line and skip next 2 lines. Here in the best case only one third of the total scan lines need be run length coded.

In 2010, Hao Sung et al. [13] proposed a Skip-line with threshold algorithm for Binary image compression. In this, skip_n_line encoding is modified by adding a threshold function. Three functions are devised to find the correlation between two scan lines.

Basic XOR function is used to find out how many bits are different between two scan lines.

$$\sum_{M}^{M} XOR( Line_{N,M} , Line_{N+1,M} )$$

Where N is the row and M is the column.

The above gives the number of bits that are different between two scan lines.

© 2013 ACEEE
DOI: 03.LSCS.2013.2.560
‘Swinging XOR Function’ added ranging into the XOR function. This algorithm skips lines with steep slopes and replaces them with vertical lines to get better compression.

\[ f = \sum \text{XOR}((\text{Line}_{N,M}), (\text{Line}_{N+1,M+1})) \] (2)

‘Seesaw’ XOR Function separately compares left side and right side ranges, and larger value is taken.

**E. Run Length Encoding**

This method is developed in the 1950’s and images with repeating intensities are represented as run length pairs. Each pair specifies the start of new intensity and the number of consecutive pixels with that intensity. This is an effective encoding technique for binary image compression.

Eg: let 1 denotes a white pixel and 0 denotes a black pixel

\[
\begin{align*}
1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
\end{align*}
\]

can be encode as \{(1,3),(0,2),(1,5)\}

**III. MODIFIED SKIP LINE ENCODING**

In Ref. [12, 13], scan lines are skipped only if there is sufficient similarity between successive scan lines. In the proposed modified approach, scan line can be skipped if there is similar scan line anywhere in the image.

In figure 3, line 1 and line 2 are similar. So there is no need to store both the lines 1 and 2. When comparing with line 3, it is different from line 2 so store line 3 and we can skip line 4. Next we compare line 4 and 5 and are different. In the modified approach, if we are about to store a new scan line we check whether a similar line is already stored or not. If it is already stored we update the index array which indicates the scan line to be substituted at the time of decoding.

**IV. RESULTS OF EXPERIMENTS**

Modified approach is implemented using Matlab and tested on binary images. Sample images used are given in Figure 5. The percent compression ratio defined as given below is used to evaluate the performance of the approach:

\[ \% \text{Compression ratio} = \left( \frac{\text{original size} - \text{compressed size}}{\text{original size}} \right) \times 100 \]

The compression ratios obtained for the three images after skip line encoding and modified skip line encoding are given in Table 1. The modified approach provides improvements in compression ratio for all the three tested images.

**Complexity analysis:**

On an average we need to compare a line with r/2 lines, where r is the number of lines (rows) in the image. Each comparison takes O(c) time, where c is the number of columns in the image. So, the complexity is O(r^2c).

**V. CONCLUSION**

Though the storage device price decreases and size increases, the storage requirements are still increasing as the present day images are of huge size. Research leading to improved techniques providing even a small increase in compression ratio compared to the existing approaches is still important for saving storage space and bandwidth. In this paper, a modified form of lossless skip line encoding is proposed to further reduce the redundancy in the image. Improvement is achieved by employing a global skip-line technique instead of the local skip-line. The compression performance is found to be better than the conventional skip-line encoding.

To improve the compression ratio further, the skip line encoding can be made lossy, by permitting loss of data during compression to approximate a row of an image with a curve. Application of suitable reconstruction technique can make the images more acceptable with a good amount of increase in compression ratio.

**REFERENCES**

Table I: Performance comparison

<table>
<thead>
<tr>
<th>Image</th>
<th>Image size (Bytes)</th>
<th>% Compression ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>skip line encoding</td>
</tr>
<tr>
<td>Figure 5.(a)</td>
<td>9600</td>
<td>90.4</td>
</tr>
<tr>
<td>Figure 5.(b)</td>
<td>12288</td>
<td>92.4</td>
</tr>
<tr>
<td>Figure 5.(c)</td>
<td>50625</td>
<td>96</td>
</tr>
</tbody>
</table>