Optimized Image Representation in Memory using Linear Arrays

Anu Suneja

Abstract - ***Various Image representation methods consume very large amounts of memory and takes large transmission time over the network. In this paper, we have tried to reduce memory requirements for storing images which can get transmitted in small time period. Initially, the preamble of a digital image as well as matrix representation of a digital image is discussed. The paper proposes an algorithm to optimize the image representation including the performance evaluation of this technique compared with the traditional techniques.***

1. INTRODUCTION

Digital images are the basis for the visualization and digital representation of designs on computers and paper [1]. These are defined in memory by a finite valued function over a finite domain. Let us assume that a digital image is a rectangular array of size M X N having domain D

D = {(r,c) | r = 0,1-------, M-1 and c=0,1,------N-1}[1]

is represented in memory by a matrix of order M X N having some integer elements g (i, j) for $0 \leq i \leq M-1, 0 \leq j \leq N-1$ where each element g (i, j) of matrix is considered as a pixel element. It represent gray level (in monochrome image) or color (in colored image) associated with pixel position (i, j). [2, 3]

There is some problem with matrix representation of an image. The problem is that it takes very large amount of memory to store any image. Since here color or gray level of each pixel is stored as individual element. Even if color of some continuous pixels is same, still it has to be stored as individual element. This problem gets worst as resolution of image get increased. Larger the memory required for an image, larger will be the time to transfer that image over network [4].

A number of compression methods have been designed to reduce transmission time for image transfer. There are two major methods to achieve it. Either use dedicated channel for image transfer or compress the image before transmitting it [5]. In this paper, we have used second method i.e. to compress the image before transmitting it. Here rather than using matrix representation, image has been represented using two 1- D arrays. This method will be more efficient if large number of continuous pixels have same color or gray level value.

2. TRADITIONAL MATRIX REPRESENTATION OF AN IMAGE

Digital Image is graphical representation of an object which is, in fact, a regular matrix that is a collection of pixel wise grey levels or intensity values [6]. According to traditional image representation method, image is stored in main memory in matrix form having dimensions R x C, where R is no. of rows and C is no. of columns or we can say resolution of screen is R x C. Value of A [i, j] = color code of pixel (i, j) [7]. Matrix representation of image results in large consumption of memory. Like if resolution of screen is R x C, range of colors provided is K and word length of system to store K color values is w bytes. This implies memory required to store an image will be=R x C x w bytes. In this image representation method, if color of some adjacent pixels is same still that same value for each pixel has to be repeated for its corresponding position in the matrix.

Resolution	**Colors used**	**Memory utilized in Traditional method(in bytes)**
8 × 8	8	128
16 × 16	16	512
64 × 64	64	8192
64 × 64	4	8192
64 × 64	8	8192
64 × 64	32	8192
64 × 64	64	8192
128 × 128	8	32768
128 × 128	16	32768
128 × 128	32	32768
128 × 128	64	32768
128 × 128	128	32768

Table 1: Memory requirement in Matrix based image representation method (assumed word length w=2 bytes)

3. PROPOSED METHOD

In proposed method, we can reduce memory requirement by taking advantage of adjacent pixels having same color. In this method, instead of using traditional 2-D array for storing pixel wise color value, we use two 1-D arrays. First 1-D array stores colors used in adjacent pixels and second array stores count of continuous pixels having same color.

If

Color[i] =n, Count[i] =m …………… (1)

implies from current to next 'm' pixels have same color value 'n'.

In proposed method memory size required is calculated as:

$m_r = (c + d) * w$ ………………….(2)

Where m_r = memory required
c = size of array color
d = size of array count
w = word length

Lecturer, Department of Computer Applications, Chitkara Institute of Engineering and Technology, Rajpura, Punjab
E-Mail: anusuneja3@gmail.com

Resolution	colors used	Memory utilized in proposed method (in bytes)
8 × 8	8	32
16 × 16	16	64
64 × 64	64	256
64 × 64	4	2056
64 × 64	8	1040
64 × 64	32	320
64 × 64	64	256
128 × 128	8	4112
128 × 128	16	2080
128 × 128	32	1088
128 × 128	64	640
128 × 128	128	512

Table 2: Memory requirement in proposed image representation method

If resolution of screen is M × N and number of colors used in image is p then memory required to store a symmetric image according to proposed method will be:

$[(M \times N)/p + p] \times w$ ……………..(3)

Where w is word length.

In this method, image is not represented pixel wise, so pixel position is calculated using formula:

a) $x=x+1$ if cnt mod r =0 ……………(4)
$x=x$ otherwise
b) $y=0$ if $y=r-1$ ……………(5)
$y=y+1$ otherwise

In proposed method, unlike matrix representation of image co ordinates (x, y) of pixel are not known rather color c_i and number of adjacent pixel cnt_i having same color c_i is given. Therefore, some algorithm is required to find pixel position from the given array C and count.

ALGORITHM:

```
Step 1) Set x: = -1, y: = -1, cnt: = 0
Step 2) Repeat Step 3 for i = 0 to nc -1
Step 3) Repeat Step 4 for k = 0 to count[i]-1
Step 4    if (cnt mod r==0) then
              x=x+1
[End of step 4 If statement]
Step 5) if (y==r-1) then
              y = 0
              Else
              y=y+1
[End of step 5 if statement]
Step 6) Drawpixel(x, y, color [I])
Step 7) cnt = cnt+1
[End of step 3 for loop]
[End of step 2 for loop]
Step 8) Exit
```

4. COMAPRISON OF PROPOSED METHOD WITH TRADITIONAL MATRIX REPRESENTATION:

Where in traditional method we have to store color value for each pixel according to resolution of screen, in proposed method, we just store the number of pixels that have color value color[i], color [i+1] and so on.

Proposed representation of an 8 X 8 sized image with 8 colors:

```
color[8]={0,1,2,3,4,5,6,7};
count[8]={8,8,8,8,8,8,8,8};
```

Traditional representation of an 8 X 8 sized image with 8 colors:

```
color [8][8]=  {{0, 0, 0, 0, 0, 0, 0, 0},
               {1, 1, 1, 1, 1, 1, 1, 1},
               ……………………
                    {7, 7, 7, 7, 7, 7, 7, 7}};
```

Representation of an image acc. to proposed method will consume less memory than traditional method .This is because here image is stored color wise and not pixel wise.

To draw an image of resolution 8 X 8 where 8 colors are used in symmetric way, Memory required to store such image in matrix form will be 128 bytes. Whereas, if we store same image acc. to proposed method we need only 32 bytes. It will not only save memory but also save execution time. Time taken to draw above image represented in traditional method is noted to be 9844429.978022 ns where as in proposed method time is noted to be 9844442970.615385 ns.

As resolution get increased more and more memory is saved. Also less the frequency with which color get changed more the memory will be saved in proposed method as compare to traditional method as shown below:

frequency of change in color	2	4	8	16
Memory Saved (in %)	0	50	75	87.45

Table 3: Memory saved with change in color

Proposed method not only saves memory, but it also save execution time as shown:

Resolution	colors used	Execution Time	
		Traditional	Traditional
8 X 8	8	954442686.2	954442625.7
16 X 16	16	954442774	954442729.6
64 X 64	64	954442861	954442823.2
128 X 128	128	954442975.3	954442899.7
64 X 64	16	954443059.5	954443029.2
64 X 64	32	954443126.5	954443098.2
64 X 64	64	954442861	954442823.2
128 X 128	16	954443200.3	954443167.2
128 X 128	32	954443273	954443227.7
128 X 128	64	954443334.5	954443307.1
128 X 128	128	954443388.3	954443357.1

Table 4: Comparison of execution time taken for Traditional and proposed method

5. CONCLUSION

In this paper digital images are represented using color codes of continuous pixels, which is different from matrix based representation that was traditionally used. It leads to reduction in memory requirement for storing an image. Also it saves image transmission time over the network. Comparison of memory utilized in traditional and proposed method has been made using charts.

6. REFERENCES

[1] http://web.arch.usyd.edu.au/~mary/DECO2101/ Date accessed November 10th, 2008.

[2] B.Chanda,D. Dutta Majumder,"Digital Image Processing and Analysis", PHI publications, 2005, pages 73-80.

[3] Jain, Anil.K., "Fundamentals of digital Image Processing", PHI ,2005,pages 4-6.

[4] Paul Haeberli,http://www.graficaobscura.com/matrix/index.html/ Date accessed December, 2008.

[5] B.Chanda,D. Dutta Majumder,"Digital Image Processing and Analysis", PHI publications, 2005, pages 145-165.

[6] https://gvsig.org/web/docusr/userguide-gvsig-1-1/anexos/glosario-en Date accessed December, 2008.

[7] C. Gonzalez Rafel,"Digital Image Processing", Pearson education, 2005, pages76-84.

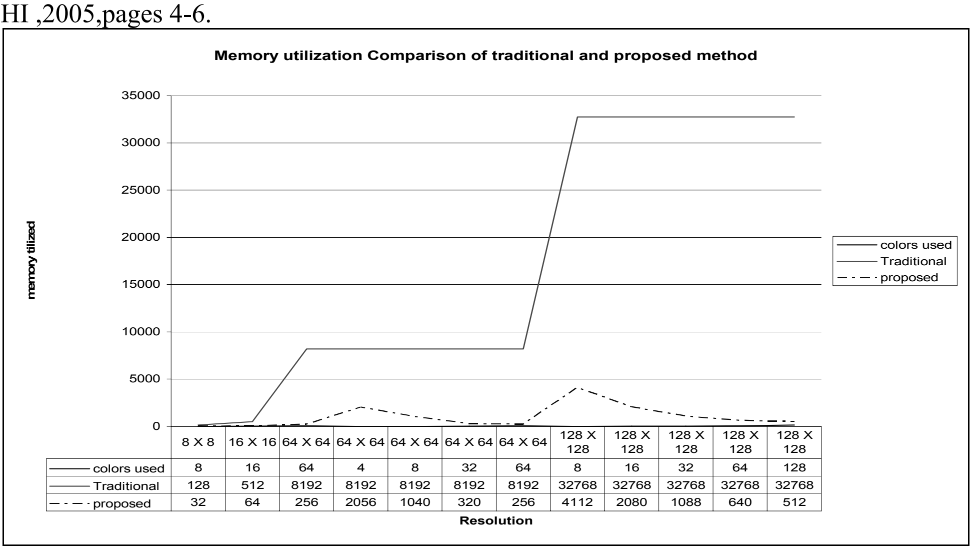

Figure 1: Comparison of memory utilized Traditional and proposed image representation method

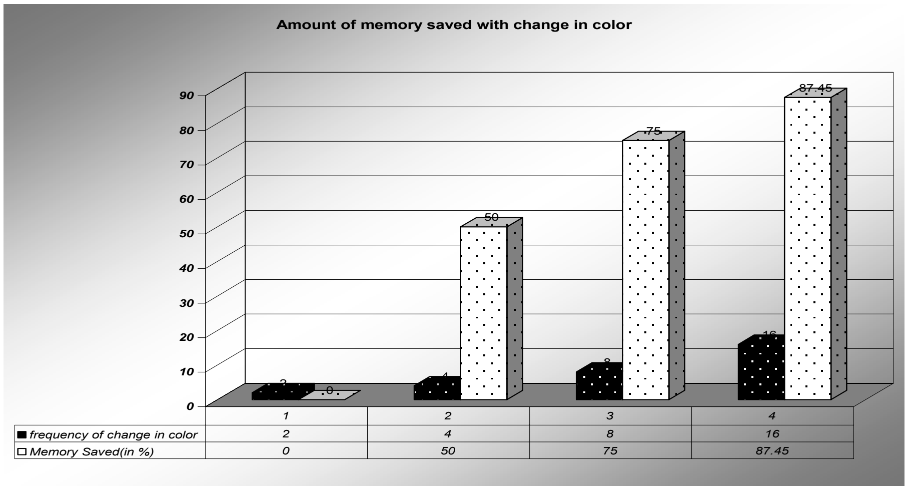

Figure 2: Comparison of memory saved with change in color for Traditional and proposed method