Line Drawing Algorithm on an Interleaved Grid

M.Prabukumar
Assistant Professor [Senior]
School of Information Technology & Engineering (SITE),
VIT University,
Vellore, Tamilnadu, India

ABSTRACT

In this paper Bresenham’s line drawing algorithm on interleaved grid is proposed. It uses the advantages of interleaved sampling to scan-convert the pixel on the raster with less representation error. The performance of the proposed algorithm is compared with the conventional Bresenham’s algorithm on square grid. The qualitative and quantitative analyses show that the proposed algorithm outperforms the Bresenham’s line drawing algorithm on square grid.

General Terms

Computer Graphics, Line Drawing Algorithms.

Keywords

Interleaved grids, rasterizing, scan conversion, aliasing.

1. INTRODUCTION

In typical Computer graphics, to describe a scene in terms of the basic geometric structure is known as output primitive. The most commonly known output primitives in computer graphics are point, line, circle and ellipse. Among these, the most widely used primitive is the “line”. In order to rasterizing a line segment, the intermediate position along the line path between two specified end point positions of the line has to be calculated. On a raster system, lines are positioned discretely in such a way that, at each sample position, the nearest pixel to the line can be determined. In most of the cases, the selected pixel doesn’t match with the true line position and as a result, the line has a jagged appearance. This effect is known as **“Aliasing”** [2]. Aliasing is a consequence of the representation error that occurs due to improper selection of pixels during the process of Scan conversion.

Digital Differential Analyzer (DDA [1]) and Bresenham’s line drawing algorithm are some of the initial algorithms that have been proposed to rasterizing a line segment. Till date, there has been no other line drawing algorithm that could parallel Bresenham’s algorithm’s [1] efficiency taken in terms of time complexity which owes to the fact that it uses only incremental integer calculations. This algorithm can also be adopted to display circle and other curves.

The performance of Bresenham’s line drawing algorithm can be enhanced by implementing it on an interleaved grid. The qualitative and quantitative analyses have resulted in the proposition that the Bresenham’s line drawing algorithm on an interleaved grid is much better than that of a square grid implementation. Owing to this approach, display device hardware manufacturers stand a major gain as they are being able to produce high quality computerized images with less representation error and at the same time without any compromise with the time complexity.

The primary motivation lying behind the usage of interleaved grid over a square grid system is that it is highly advantageous in many ways. Some of them include Sampling efficiency as it reduces the number of pixels being used, thus rendering the same resolution at a much better economic level; Six fold symmetry offers superior symmetry, in due course reducing the computation time for processing; Well behaved connectivity that provides less ambiguous identity of boundaries and regions corresponding to a particular pixel thus defining a definite neighborhood. Another interesting advantage is that the retina of the human eye closely resembles an interleaved grid system thus facilitating a better view of the object with an interleaved grid system [5].

The paper is organized as follows, in section 2 interleaved grid structures has been discussed. In section 3, implementation of line drawing algorithm on interleaved grid has been presented. The performance analysis of the Bresenham’s line drawing algorithm on both square and interleaved grid has been made in section 4 and finally conclusion has been drawn in section 5..

2. INTERLEAVED GRID

Computer graphics, in general, represents an object in conventional square grid. The proposed paradigm, interleaved grid is an alternative that serves the sane objective [4]. The representation of square and interleaved grid is shown above in figure 1 and figure 2 respectively. From the figure 2 it is clear that interleaved grid offers less distortion of distances than square grids, because each pixel has more non-diagonal neighbors than in a square grid. An interleaved grid has a pleasant appearance than a square grid. Comparison of certain features between square and interleaved grid is shown in table 1.[6,7]

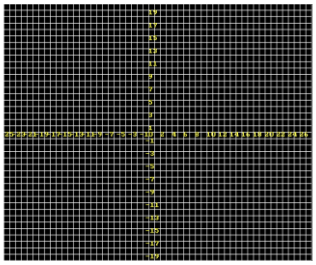


Figure 1. Square Grid

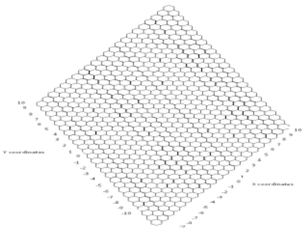


Figure 2.Interleaved Grid

Table1. Square Grid versus Interleaved Grid

Features	Type of Grid	
	Square	**Interleaved**
Number of neighboring pixels	8	6
Minimum distance of neighboring pixel distance from its center	1	1
Maximum distance of neighboring pixel distance from it center	1.414	1
Number of privileged directions of the line display	2	3
Angle between the privileged directions(degree)	90	60

3. LINE DRAWING ALGORITHM ON AN INTERLEAVED GRID

Consider a line segment whose slope is in the range of $0 \le m \le 1$. From the figure 3, it is clear that if (x, y) is the last pixel drawn then the next pixel to be drawn is either (x + 1, y) or (x, y + 1). In most of the cases, the selected pixel does not have an actual representation, because the pixel selected by the mathematical equation cannot be addressed on the screen if it is a floating point number. This error is called **rasterization error** and is denoted by ε. It ranges between -05 to 0.5. While selecting the next pixel from the current pixel, if the difference between the new value of y (i.e. y + ε + m) and the current value of y is less than 0.5, the pixel is (x + 1, y) is selected, otherwise the selected pixel is (x, y + 1). The updated error value depends on the pixel selected. If we select (x + 1, y) then the updated error value is $\varepsilon_{new} = (y + \varepsilon + m) - y$, otherwise, it is $\varepsilon_{new} = (y + \varepsilon + m) - (y + 1)$.Thus arises the following algorithm that avoids rounding operation.

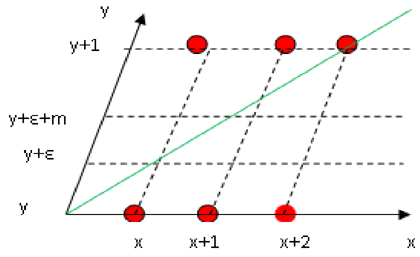


Figure 3: Positive slope line on Interleaved Grid

Algorithm (a):

```
Step1: Read the two endpoints of the line (x1, y1) & (x2, y2)

Step2: Initialize $\varepsilon := 0$; $y := y1$.

Step3: For x = x1 to x2

    Plot(x, y)

        If $((\varepsilon + m) < (0.5))$

        $x := x + 1$;

        $\varepsilon := \varepsilon + m$;

        Else

        $y := y + 1$;

        $\varepsilon := \varepsilon + m - 1$;

        Endif

    Endfor.
```

The algorithm (a) uses floating point calculations. To avoid this, let us multiply Δx and 2 on both sides of the decision criterion used in algorithm (a).

$((\varepsilon + m) * 2 * \Delta x < (1 / 2) * 2 * \Delta x)$

$(2 * \Delta x * \varepsilon) + (2 * \Delta y) < \Delta x$

Replace $\Delta x * \varepsilon$ by ε` in the above equation we get

$2(\varepsilon` + \Delta y) < \Delta x$

The updated decision criterion for the error in the form of ε` is derived from the equations

$\varepsilon = \varepsilon + m$ and

$\varepsilon = \varepsilon + m - 1$.

We get,

$\varepsilon * \Delta x = \varepsilon * \Delta x + \Delta y$

$\varepsilon` = \varepsilon` + \Delta y$

$\varepsilon * \Delta x = \varepsilon * \Delta x + \Delta y - \Delta x$

$\varepsilon` = \varepsilon` + \Delta y - \Delta x$

Using this new error value ε`, the modified Bresenham's line drawing algorithm is presented below:

Algorithm (b):

```
Step1: Read the two endpoints of the line (x1, y1) & (x2, y2)

Step2: Initialize $\varepsilon` := 0$; $y := y1$.

Step3: For x = x1 to x2
```

```
        Plot(x, y)
                If $(2 (\varepsilon` + \Delta y) < \Delta x)$
                $x: = x + 1;$
                $\varepsilon`: = \varepsilon` + \Delta y;$
                Else
                $y: = y + 1;$
                $\varepsilon` : = \varepsilon`+ \Delta y - \Delta x;$
                Endif
        Endfor.
```

Similarly, consider a line whose slope is between 0 and − 1. From the figure 4, it is clear that the plotted point at (x, y) using the line-drawing algorithm, the possible point for the next plot is between (x + 1, y − 1*)* and (x+1, y). That leads to an error (ε), related with *y*, which can be minimized by proper selection of the next point. So plot *(x*+1*, y)* if y − (y + ε + m) is less than 0.5 otherwise plot *(*x+1, y −1*)*. The error update rule is delicately different for the negative slope line. If plotting *(*x+1*,* y*)* the *new* value of error is $\varepsilon_{new} = (y + \varepsilon + m) - y$, otherwise it is $\varepsilon_{new} = (y + \varepsilon + m) - (y - 1)$. This gives the following algorithm which avoids rounding operation.

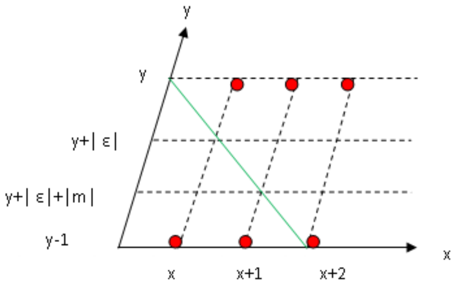


Figure 4: Negative slope line on Interleaved Grid.

Algorithm (c):

Step1: Read the two endpoints of the line (x1, y1) & (x2, y2)

Step2: Initialize ε: = 0; y: = y1.

Step3: For x=x1 to x2

```
        Plot(x, y)
                If $((\varepsilon + m) + 0.5 > 0)$
                $x: = x + 1;$
                $\varepsilon: = \varepsilon + m;$
                Else
                $x: = x + 1;$
                $y: = y - 1;$
                $\varepsilon: = \varepsilon + m + 1;$
                Endif
        Endfor.
```

This algorithm has floating point operations and to alleviate it, we follow the same procedure that had been described for positive slope (multiply Δx and 2 on both sides of the decision criterion). Using the new error value of ε` the modified Bresenham's line drawing algorithm for negative slope is presented below.

Algorithm (d):

Step1: Read the two endpoints of the line (x1, y1) & (x2, y2)

Step2: Initialize ε`: = 0; y: = y1.

Step3: For x=x1 to x2

```
        Plot(x, y)
                If $(2(\varepsilon` + \Delta y) + \Delta x > 0)$
                $x: = x + 1;$
                $\varepsilon`:= \varepsilon` + \Delta y;$
                Else
                $x: = x + 1;$
                $y: = y - 1;$
                $\varepsilon`: = \varepsilon`+ \Delta y + \Delta x;$
                Endif
        Endfor.
```

4. PERFORMANCE ANALYSIS

The performance of Bresenham's line drawing algorithm on an interleaved grid is compared qualitatively and quantitatively with the conventional Bresenham's line drawing algorithm on square grid. To provide fair comparison between the two algorithms, the interleaved grid area is made equal to square grid area on the screen. The lines considered to perform the qualitative analysis are in the orientation from 0^0 to 360^0 with respect to the x-axis. The comparison is performed qualitatively by plotting lines on the screen (as shown in figure 5 and figure 6) and quantitatively by computing root mean square error (RMSE) (as shown in table 3 and figure 7).

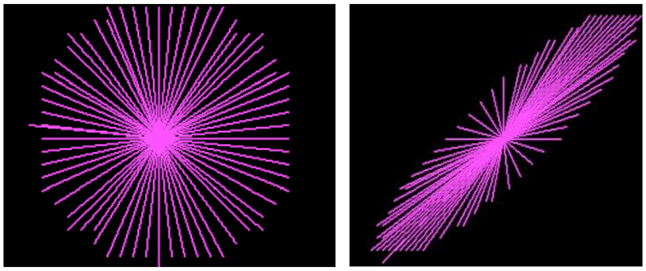
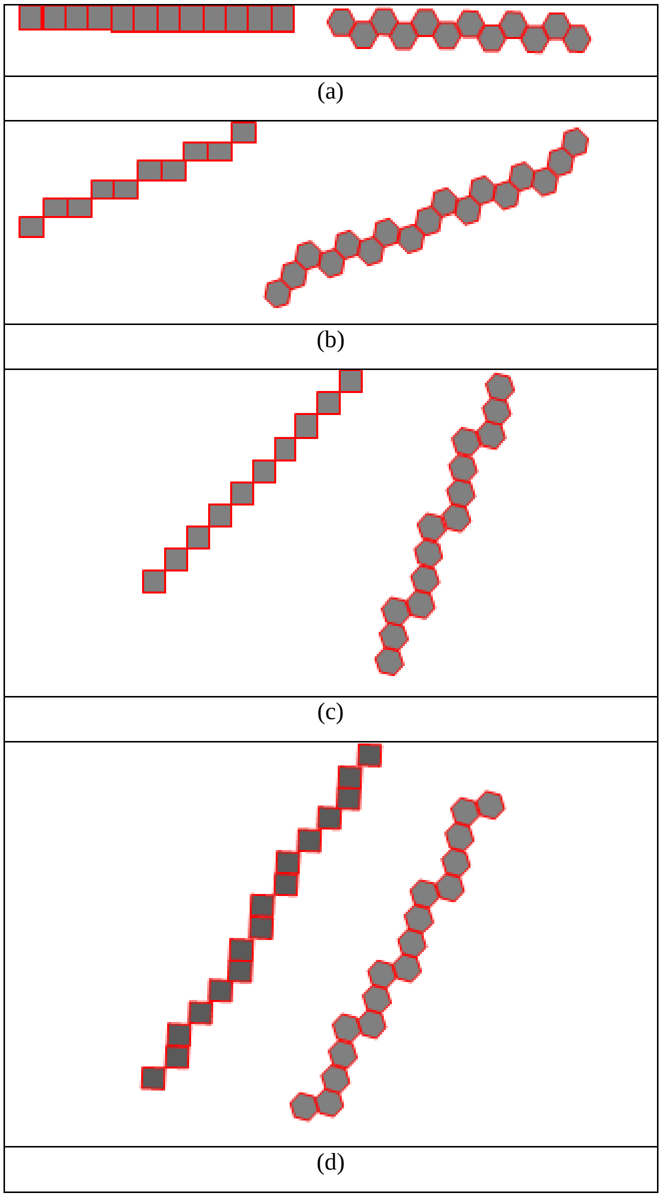


Figure 5a.Square Grid

Figure 5b.Interleaved Grid

Figure 5. Line orientation plot (0 to 360 degree)

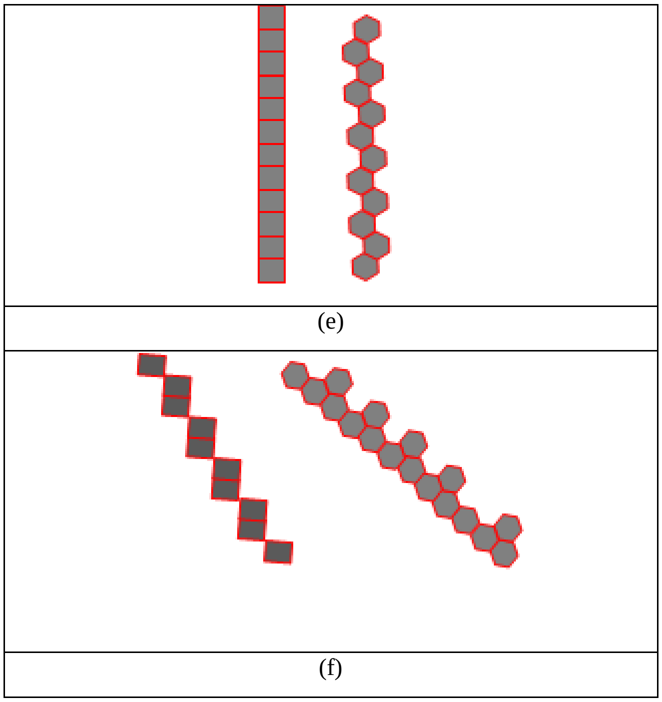


(a)

(b)

(c)

(d)



(e)

(f)

Figure 6: Magnified view of the lines on square and interleaved grid, the left figure is on square grid and the right one is on interleaved grid with line orientation (a) zero degree, (b) 30 degree, (c) 45 degree, (d) 60 degree, (e) 90 degree, (f) 120 degree.

Table 2: Qualitative analysis square vs Interleaved Grid

S.No	Angle(in degree)	Type of Grid		Good in
		Square grid	Interleaved grid	
1	0	Smoothness high	Smoothness low	square grid
2	30	More ragged	Less ragged	Interleaved grid
3	45	Less ragged	More ragged	square grid
4	60	More ragged	Less ragged	Interleaved grid
5	90	Smoothness high	Smoothness low	square grid
6	120	More ragged	Less ragged	Interleaved grid

The result of the Qualitative analysis is tabulated as above. It clearly reflects the superior efficiency of an interleaved grid system over a square grid system.

Now, moving on to the Quantitative analysis, we can actually observe the difference between values predicted by the proposed algorithm and the values actually observed from the line equation y=mx+c. It can be obtained from the root mean square deviation (RMSD) or root mean square error (RMSE) [8, 9, and 10].It is frequently used to measure or to judge the efficiency of any model. It is given by

Table 3: RMSE in square and interleaved grid.

Line Orien tation	RMSE		Line Orient ation	RMSE	
	Square	Interle aved		Square	Interlea ved
0	0	0	180	0	0
5	0.3175	0.5771	185	0.3175	0.5774
10	0.3419	0.3922	190	0.3239	0..5253
15	0.4088	0.4629	195	0.3528	0.5
20	0.432	0.3651	200	0.3928	0.3922
25	0.625	0.433	205	0.4543	0.5345
30	0.5092	0.4472	210	0.5777	0.5976
35	0.559	0.433	215	0.5777	0.5976
40	0.7126	0.5	220	0.6602	0.5976
45	0.686	0.4851	225	0.6831	0.4472
50	0.8144	0.6614	230	0.7702	0.4629
55	0.8385	0.6124	235	0.9243	0.5345
60	0.8385	0.6124	240	0.9243	0.5345
65	1.25	0.559	245	1.0221	0.5345
70	1.089	0.6831	250	1.1767	0.6202
75	1.3628	0.8452	255	1.5877	0.7071
80	1.7097	0.9199	260	2.7798	1.206
85	3.1754	1.5546	265	2.7798	1.206
90	0	0	270	0	0
95	3.1754	1.5546	275	2.7798	1.206
100	3.1754	1.5546	280	1.5877	0.7071
105	1.7097	0.9199	285	1.1767	0.6202
110	1.3628	0.8452	290	1.0221	0.5345
115	1.0801	0.6831	295	0.9165	0.5164
120	1.0221	0.8018	300	0.9108	0.5164
125	0.9165	0.5774	305	0.9108	0.5164
130	0.9108	0.6124	310	0.6831	0.4472
135	0.8144	0.6614	315	0.7126	0.6124
140	0.6831	0.5164	320	0.6831	0.6325
145	0.6831	0.4472	325	0.5092	0.5164
150	0.5777	0.4629	330	0.5092	0.5164
155	0.5092	0.4472	335	0.432	0.5164
160	0.4543	0.378	340	0.4088	0.4629
165	0.3922	0.3922	345	0.3419	0.4804
170	0.3528	0.4082	350	0.3175	0.5774
175	0.3089	0.603	355	0.3175	0.5774

$$\sqrt{\sum_{i=1}^{n}(y-ya)^2/n},$$

Where, y is the value of y coordinate obtained from line equation y=mx+c

ya is the value of y coordinate obtain from line drawing algorithm

n is the length of the line segment.

The root mean square error (RMSE) for the line orientation from zero degree to three sixty degrees at every five degree difference using Bresenham's line algorithm on square and interleaved grid are tabulated in table (3) and plotted in figure (7).From the figure it is clear that Bresenham's line algorithm on interleaved grid has less RMSE than Bresenham's line algorithm on square grid. The performance of Bresenham's line drawing algorithm on interleaved grid is **66.66 %** better than square grid implementation in terms of RMSE. From the table (2) it is clear that overall performance is good in interleaved grid than conventional square grid.

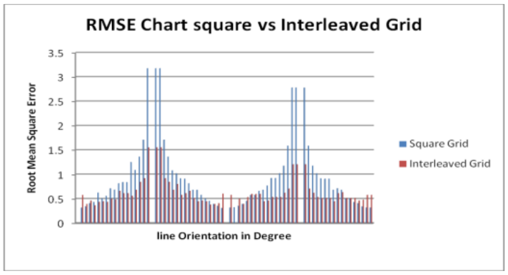


Figure 7: RMSE chart for square and interleaved grid

The theoretical computation and execution time analysis are given in table 4 and table 5 respectively. To reduce impacts of multicasting and multiprogramming on execution time, the average execution time is calculated. The theoretical computation is same for both types of grids, the execution time analysis make this statement true. From the table 4 and figure 8, it is clear that execution time for the line algorithm on interleaved grid is comparable with square grid implementation.

Table 4: Theoretical Computational Load (N is the length of the line segment)

Arithmetic operation	Square	Interleaved
Number of addition	2N	2N
Number of Subtraction	N	N
Number of Comparison	N	N

Table 5: Execution Time Analysis of Square vs Interleaved Grid

Line Orientation	Average execution time(seconds)		Line Orientation	Average execution time(seconds)	
	Square	Interleaved		Square	Interleaved
0	0.031	0.032	180	0.015	0.031
5	0.016	0.031	185	0.031	0.015
10	0.031	0.015	190	0.032	0.032
15	0.031	0.032	195	0.015	0.015
20	0.016	0.015	200	0.032	0.032
25	0.016	0.032	205	0.015	0.015
30	0.015	0.015	210	0.016	0.016
35	0.031	0.016	215	0.015	0.015
40	0.013	0.015	220	0.016	0.016
45	0.016	0.016	225	0.016	0.016
50	0.016	0.016	230	0.016	0.016
55	0.016	0.016	235	0.016	0.016
60	0.016	0.016	240	0.015	0.015
65	0.015	0.015	245	0.015	0.015
70	0.015	0.015	250	0.015	0.015
75	0.015	0.015	255	0.015	0.015
80	0.015	0.015	260	0.016	0.016
85	0.015	0.015	265	0.016	0.016
90	0.016	0.016	270	0.016	0.016
95	0.016	0.016	275	0.016	0.016
100	0.016	0.016	280	0.016	0.016
105	0.016	0.016	285	0.016	0.016
110	0.016	0.016	290	0.015	0.015
115	0.015	0.015	295	0.016	0.016
120	0.015	0.015	300	0.015	0.015
125	0.016	0.016	305	0.015	0.015
130	0.015	0.015	310	0.016	0.032
135	0.032	0.016	315	0.031	0.015
140	0.015	0.031	320	0.016	0.032
145	0.016	0.016	325	0.031	0.031
150	0.031	0.031	330	0.031	0.031
155	0.016	0.016	335	0.047	0.031
160	0.015	0.016	340	0.031	0.031
165	0.016	0.031	345	0.031	0.032
170	0.031	0.031	350	0.031	0.031
175	0.032	0.032	355	0.016	0.031

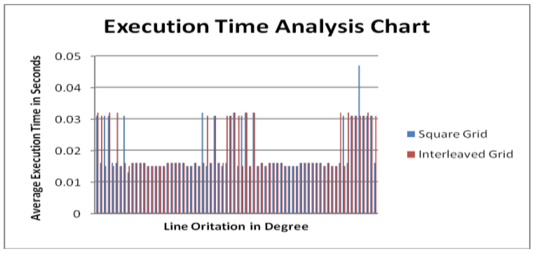


Figure 8: Execution Time Analysis Chart square vs Interleaved Grid.

5. CONCLUSION

Scan conversion of given two dimensional line is implemented effectively and efficiently using Bresenham’s line drawing algorithm on interleaved grid structure. The qualitative and quantitative analysis concludes that the generated line using Bresenham’s algorithm on interleaved grid is better then conventional square grid implementation. The theoretical Computational Load and the Execution Time analysis further strengthen the above point.

6. REFERENCES

[1] Bresenham, J.E. *Algorithm for computer control of a digital plotter*, IBM Systems Journal, January 1965, pp. 25-30.

[2] J. D. Foley, A. van Dam, S. K. Feiner, and J. H. Hughes, *Computer graphics: Principles and practice*, Second edition, Addison-Wesley (1990).

[3] Donald Hearn, and M. Pauline Baker, *Computer graphics, C Version*, Second edition, Prentice-Hall (2005).

[4] R.M.Mersereau, "The processing of hexagonally sampled two dimensional signals," *Proc. IEEE* vol. 67 no. 6 pp.930-949, June 1979.

[5] R.C.Staunton, "Hexagonal image sampling, a practical proposition," *Proc. SPIE* vol. 1008 pp. 23-27, 1989.

[6] Luszak.E and Rosenfeld.A, Distance on a hexagonal grid, *IEEE Transactions on Computers,* 25(3), 532-533(1968).

[7] Wuthrich, C.A. and Stucki, P, An algorithm comparison between square- and hexagonal based grids. *Graphical Models and Image Processing*, **53**(4), 324–339. (1991).

[8] Bimal kumar Ray, An alternative approach to circle drawing, Journal.Indian Institude of Science, 86,617-623(2006).

[9] Root mean square deviation, Internet source: http://en.wikipedia.org/wiki/Root_mean_square_deviation.html

[10] Least Squares Fitting, Internet source: http://mathworld.wolfram.com/LeastSquaresFitting.html.

AUTHORS PROFILE

Prof. Prabukumar.M is working as a Assistant Professor (Senior) in School of Information Technology and engineering of VIT University, Vellore, India. He is a life time member in computer society of India. His areas of interests includes Computer graphics, image processing, Video Image Processing, Computer Vision, Digital Signal Processing, Digital Systems Design and Computer Networks.