ISSN (Online): 2230-7893
www.IJCEM.org

Replacement of Graphic Translations with Two-Dimensional Cellular Automata, Twenty Five Neighborhood Model

Fasel Qadir[1], Shah Jahan[2] Peer M. A.[3] and Khan K. A.[4]

[1] Department of Comp. Science, University of Kashmir,
Sopore, J & K, India
faselqadir@gmail.com

[2] Department of Comp. Science, University of Kashmir,
S.P. College, Srinagar, J & K, India
sjwani2001@yahoo.com

[3] Chairman BOPEE,
Srinagar, J & K, India
drpeerma@gmail.com

[4] Department of Physics, S. P. College,
Srinagar, J & K, India
khursheedahmadk@gmail.com

Abstract

Images are being processed and manipulated at every step with the help of modern multimedia tools. In most gaming devices and cartoon series, movement of images are confined or restricted to right, left, up & down directions only. Cellular automata can be successfully applied in image processing. Cellular Automata is a methodology that uses discrete space to represent the state of each element of a domain and this state can be changed according to a transition rule. With our scheme, we are not only able to translate the image into x and y-axis, but also diagonal translations can be achieved. Uniform cellular automata rules are constructed to transform the images in all the directions.

Keywords: *Cellular Automata, Image Processing, Linear Rule.*

1. Introduction

Ulam [1] and Von Neumann [2] at first proposed Cellular Automata (CA) with the intention of achieving models of biological self-reproduction. After a few years, Amoroso, Fredkin and Cooper [3] described a simple replicator established on parity or modulo-two rules. Later on, Stephen Wolfram formed the CA theory [4],[5]. CA's are models for systems which consist of simple components and behaviour of each component is obtained and reformed upon the behaviour of its neighbors and their previous behaviour. The constructing components of these models can do robust and complicated tasks by interacting with each other. Nowadays, cellular automata are widely used in many areas of image processing such as de-noising, enhancing, smoothing, restoring, and extracting features of images.

Many versions of cellular automata (CA) with three neighborhood local state transition rule are known. In [6], algebraic properties of CA with local state transition rule number 90 are investigated. In [7] and [8], properties of CA's with cyclic boundary condition are investigated, using a polynomial expression. In [9], properties of CA's with fixed value boundary condition are investigated, using algebraic expressions. These studies mainly deal with three neighbourhood CA's. But the behaviors of CA's with five or more neighbors are full of variety [10].

With the procedures for displaying output primitives and their attributes, we can create a variety of pictures and graphs. In many applications there is also a need for altering or manipulating displays. Design applications and facility layouts are created by arranging the orientations and sizes of the component parts of the scene. And animations are produced by moving the "camera" or the objects in a scene along animation paths. Changing in orientation, size and shape are accomplished with geometric transformations that alter the coordinate descriptions of the objects. Unfortunately limited research has been done on geometrical translations based on cellular automata. P. P. Choudary et al has proposed modeling techniques for fundamental image processing, where they have applied the rules only on binary images [11], [12]. Khan has also proposed hardware architecture for digital image transformations based on cellular automata [13]. These transformations are too complicated to describe it by a small number of parameters. Here, we

ISSN (Online): 2230-7893
www.IJCEM.org

propose a new approach of modelling for transforming greyscale images along all directions by using CA linear rule. Cellular automata models having appeared as an alternative to the complex equations, having the capability to describe continuous dynamical systems. The basic idea is not to try to describe a complex system, using difficult equations but simulating this system by interaction of cells following easy rules. In other words, not to describe a complex system with complex equations but let the complexity emerge by interaction of simple individuals following simple rules. They are absolutely stable and have no rounding off errors. Moreover, border conditions in CA are straight forward, which makes them appropriate to simulate image processing and its related application areas.

2. Cellular Automata

CA model is composed of cell, state set of cell, neighbourhood and local rule. Time advances in discrete steps and the rules of the universe are expressed by a single receipt through which, at each step computes its new state from that of its close neighbours. Thus the rules of the system are local and uniform. There are one- dimensional, two-dimensional and three-dimensional CA models. For example, a simple two-state, one dimensional CA consists of a line of cells, each of which can take value ‘0’ or ‘1’. Using a local rule (usually deterministic), the value of the cells are updated synchronously in discrete time steps. With a k-state CA model each cell can take any of the integer values between 0and k-1. In general, the rule controls the evolution of the CA model.

A CA is a 4-tuple {L, S, N, F}: where L is the regular lattice of cells, S is the finite state of cells, N is the finite set of neighbors indicating the position of one cell related to another cells on the lattice N, and F is the function which assigns a new state to a cell where $F:S^{|N|} \rightarrow S$.

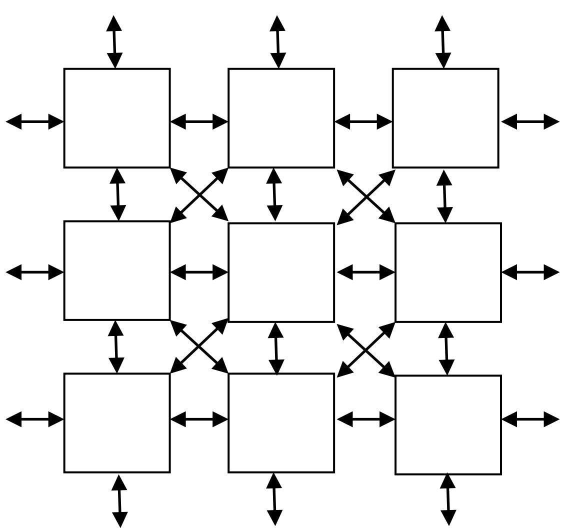

Fig. (1): Network Structure of 2DCA

As the image is a two dimensional, here we use 2DCA model. In a 2DCA the cells are arranged in a two dimensional grid with connections among the neighboring cells, as shown in the figure (1). The central box represents the current cell (that is, the cell being considered) and all other boxes represent the eight nearest neighbours of that cell. The structure of the neighbours mainly includes Von Neumann neighbourhood and Moore neighbourhood are shown in figure–(2):

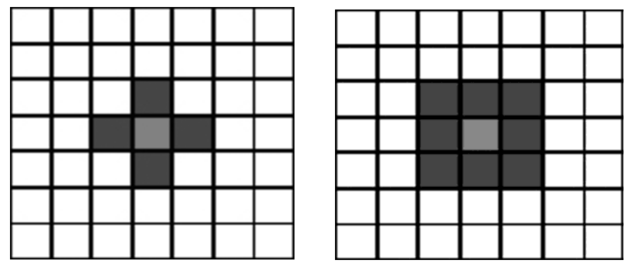

Fig. (2) : structure of neighbourhoods
(a) Von Neumann neighbourhood
(b) Moore neighbourhood

Von Neumann neighbourhood, four cells, the cell above and below, right and left from each cell is called von Neumann neighbourhood of this cell. The radius of this definition is 1, as only the next layer is considered. The total number of neighbourhood cells including itself is five as shown in the equation (1) [14]:

$$N(I,j) = \{ (k,l) \in L : |k-i| + |l-j| \leq 1 \} \quad (1)$$

where k is the number of states for the cell and l is the space of image pixels. Besides the four cells of von Neumann neighbourhood, moore neighbourhood also includes the four next nearest cells along the diagonal. In this case, the radius r=1 too. The total number of neighbour cells including itself is nine all as shown in the equation (2) [14]:

$$N(I,j) = \{ (k,l) \in L : \max (|k-i|,|l-j|) \leq 1 \} \quad (2)$$

The state of the target cell at time t+1 depends on the states of itself and the cells in the neighbourhood at time t. In case of moore neighborhood, the next state (t+1) is given by:

$$S_{i,j}(t+1) = f(S_{i-1,j-1}(t) \oplus S_{i-1,j}(t) \oplus S_{i-1,j+1}(t) \oplus S_{i,,j-1}(t) \oplus S_{i,j} \oplus S_{i,j+1}(t) \oplus S_{i+1,j-1}(t) \oplus S_{i+1,j}(t) \oplus S_{i+1,j+1}(t) \quad (3)$$

Since operation $\oplus$ in Eq.(3) is logical Exclusive-OR.

3. Geometrical Translation - an overview

Suppose that the task is to translate a two dimensional points with coordinates (x,y) to a new location (x',y') by using the displacement (dx, dy). The translation is easily accomplished by using the equations:

$$x' = x + dx$$
$$y' = y + dy \qquad (4)$$

The translation equation (4) can be expressed as a single matrix equation by using column vectors as:

$$P = \begin{bmatrix} x \\ y \end{bmatrix}, P' = \begin{bmatrix} x' \\ y' \end{bmatrix}, T = \begin{bmatrix} dx \\ dy \end{bmatrix} \qquad (5)$$

Then, two dimensional translation equation in the matrix form can be expressed more concisely as:

$$P' = P + T \qquad (6)$$

We can translate an object by applying equation (3) to every point of the object. Because each line in an object is made up of an infinite number of points, however this process will take an infinite long time. Fortunately, we can translate all the points on a line by translating only the lines end points and by drawing a new line between the translated points; this also true of scaling (stretching) and rotation. Translation is a rigid-body transformation that moves objects without deformation, that is every point on the object is translated by the same amount.

4. Cellular Automata Linear Rules

Many versions of cellular automata (CA) with three neighborhood local state transition rule are known. In [16], algebraic properties of CA with local state transition rule number 90 are investigated. In [17] and [18], properties of CA's with cyclic boundary condition are investigated, using a polynomial expression. In [19], properties of CA's with fixed value boundary condition are investigated, using algebraic integers. These studies mainly deal with three neighbourhood CA's. But the behaviors of CA's with nine or more neighbors are full of variety.

A rule is the "program" that governs the behaviour of the system. All cells apply the rule over and over, and it is the recursive application of the rule that leads to the remarkable behaviour exhibited by many CA's. In 2-D twenty-five neighborhood CA the next state of a particular cell is affected by the current state of itself and twenty-four cells in its nearest neighborhood also referred as extended moore neighborhood as shown in Figure-(3). A specific rule convention that is adopted here is given by [15].We use their model as reference and modify it so as to study CA based image processing.

Therefore, 2^{25} = 33554432 possible states exist. Each of **33554432** states can produce a 1 or a 0 for the centre cell in the next generation. Hence, $\mathbf{2^{33554432}}$ possible rules exist. A comprehensive study of all rules in higher dimensional automata is thus not easily possible. However, in this paper we will mainly concentrate on the primary rules.

The central box represents the current cell and all other boxes represent the eight nearest neighbours of that cell. Each box contains the rule number as well as the pixel location associated with that rule. In case, the next state of a cell depends on the present state of itself and/or its one or more neighbouring cells (including itself), the rule number will be the arithmetic sum of the numbers of the relevant cell.

65536 **(i-2,j-2)**	**32768** **(i-2,j-1)**	**16384** **(i-2,j)**	**8192** **(i-2,j+1)**	**4096** **(i-2,j+2)**
131072 **(i-1,j-2)**	**16** **(i-1,j-1)**	**8** **(i-1,j)**	**4** **(i-1,j+1)**	**2048** **(i-1,j+2)**
262144 **(i,j-2)**	**32** **(i,j-1)**	**1** **(i,j)**	**2** **(i,j+1)**	**1024** **(i,j+2)**
524288 **(i+1,j-2)**	**64** **(i+1,j-1)**	**128** **(i+1,j)**	**256** **(i+1,j+1)**	**512** **(i+1,j+2)**
1048576 **(i+2,j-2)**	**2097152** **(i+2,j-1)**	**4194304** **(i+2,j)**	**8388608** **(i+2,j+1)**	**16777216** **(i+2,j+2)**

Fig. (3): 2DCA Rule Convention using extended Moore neighborhood

5. Image Transformations: 2DCA based Approach

Images are being processed and manipulated at every step in modern multimedia tools. In most gaming devices and cartoon series, movement of images are confined or restricted to right, left, up & down directions only. With our scheme, we are not only able to translate the image into x and y-axis, but also diagonal translations can be achieved. The angle description of the proposed method is shown in the figure (21).

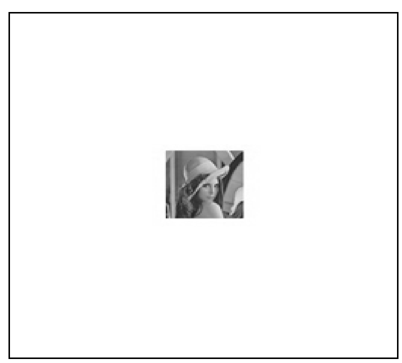

Fig. (4): Original Image whose size is 256*256.

Fig. (5): Application of Rule 2 (t=100) and Rule 1024 (t=50), it moves the Original image 100 pixels to its left. In other words, these rules move the original image at an angle of 180^o.

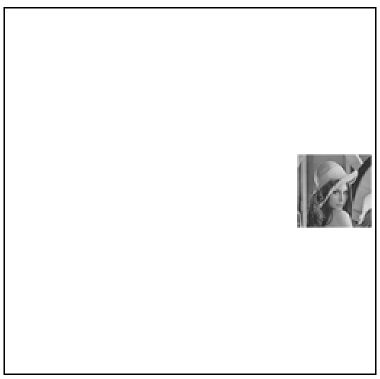

Fig. (6): Application of Rule 32 (t=100) and Rule 262144 (t=50), it moves the Original Image 100 pixels to its right. In other words, these rules move the original image at an angle of 360^o.

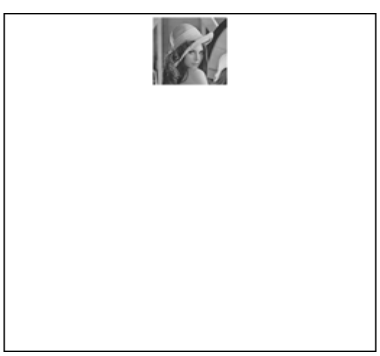

Fig. (7): Application of Rule 128 (t=100) and Rule 4194304 (t=50), it moves the Original Image 100 pixels to its top. In other words, these rules move the original image at an angle of 90^o.

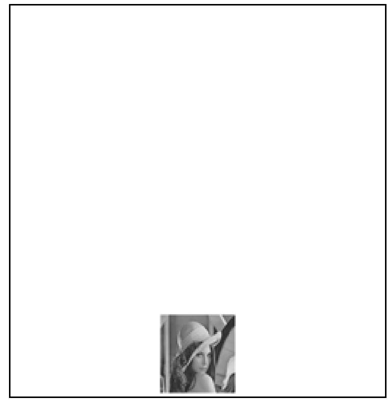

Fig. (8): Application of Rule 8 (t=100) and Rule 16384 (t=50), it moves the Original Image 100 pixels to its bottom. In other words, these rules move the original image at an angle of 270^o.

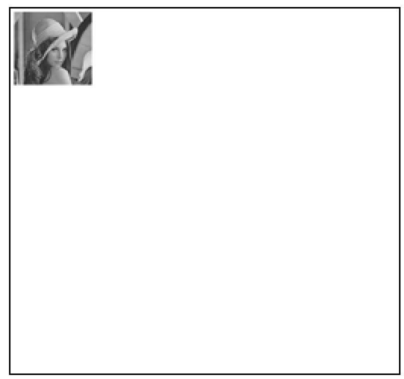

Fig. (9): Application of Rule 256 (t=100) and Rule 16777216 (t=50), it moves the Original image 100 pixels to its top-left. In other words, it moves the original image at an angle of 135^o.

Fig. (10): Application of Rule 64 (t=100) and Rule 1048576 (t=50), it moves the Original Image 100 pixels to its top-right. In other words, it moves the original image at an angle of 45^o.

Fig. (11): Application of Rule 4 (t=100) and Rule 4096 (t=50), it moves the Original Image 100 pixels to its bottom-left. In other words, these rules move the original image at an angle of 225^o.

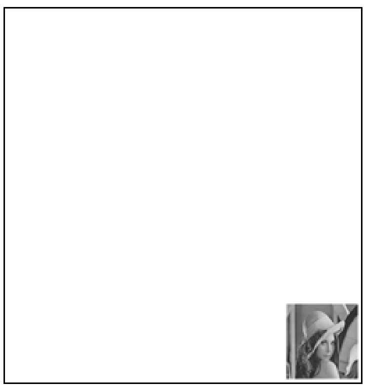

Fig. (12): Application of Rule 16 (t=100) and Rule 65536 (t=50), it moves the Original Image 100 pixels to its bottom-right. In other words, it moves the original image at an angle of 315^o.

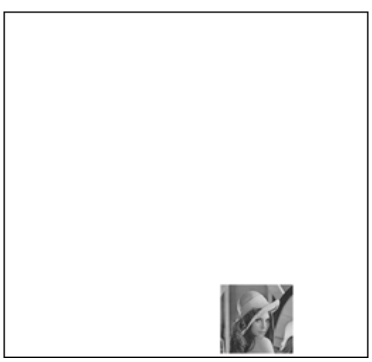

Fig. (13): Application of Rule 32768 (t=50), it moves the Original Image 100 pixels between its Bottom and Bottom-Right. In other words, it moves the original image at an angle of 292.5^o.

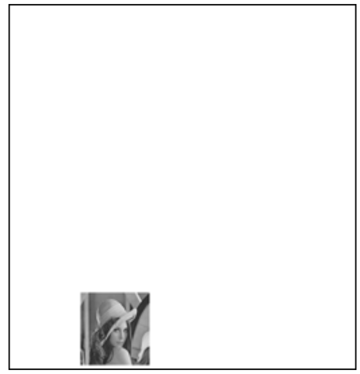

Fig. (14): Application of Rule 8192 (t=50), it moves the Original Image 100 pixels between its Bottom and Bottom-Left. In other words, it moves the original image at an angle of 247.5^o.

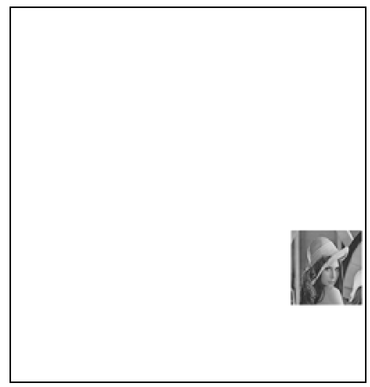

Fig. (15): Application of Rule 131072 (t=50), it moves the Original Image 100 pixels between its Right and Bottom-Right. In other words, it moves the original image at an angle of 337.5^o.

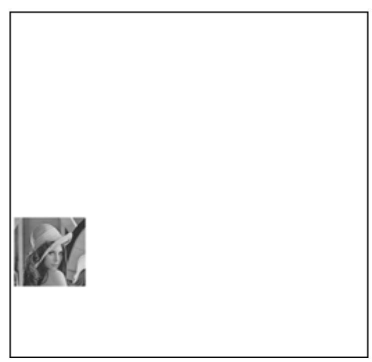

Fig. (16): Application of Rule 2048 (t=50), it moves the Original Image 100 pixels between its Left and Bottom-Left. In other words, it moves the original image at an angle of 202.5^o.

Fig. (17): Application of Rule 524288 (t=50), it moves the Original Image 100 pixels between its Top-right and Right. In other words, it moves the original image at an angle of 22.5^{o}.

Fig. (18): Application of Rule 512 (t=50), it moves the Original Image 100 pixels between its Top-left and left. In other words, it moves the original image at an angle of 157.5^{o}.

Fig. (19): Application of Rule 2097152 (t=50), it moves the Original Image 100 pixels between its Top and Top-Right. In other words, it moves the original image at an angle of 67.5^{o}.

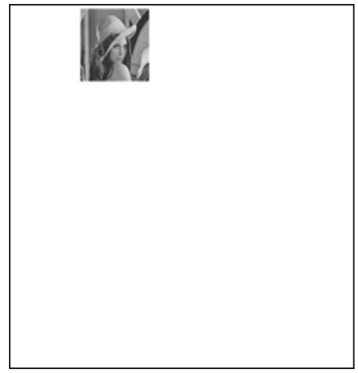

Fig. (20): Application of Rule 8388608 (t=50), it moves the Original Image 100 pixels between its Top and Top-Left. In other words, it moves the original image at an angle of 112.5^{o}.

6. Conclusions

In most gaming devices and cartoon series, movement of images are confined or restricted to right, left, up & down directions only. With our scheme, we are not only able to translate the image into x and y-axis, but also diagonal translations can be achieved. Uniform cellular automata rules are constructed with periodic boundary condition to transform the images in all the directions. Also we can translate The proposed technique can be extended to games and animation.

References

[1] S. Ulam, "Some Ideas and Prospects in Biomathematics", Annual Review of Biophysics and Bioengineering, 1963, pp. 277-292.

[2] J. Von Neumann, "Theory of Self-Reproducing Automata", University of Illinois Press, 1966.

[3] S. Amoroso, G. Cooper, " Tessellation Structures for Reproduction of Arbitrary Patterns", J. Comput. Syst. SCI, 1971, pp. 455-464.

[4] S. Wolfram, "Statistical Mechanics of Cellular Automata", Rev. Mod. Phys. 1983, pp. 601-644.

[5] S. Wolfram, "Computation Theory of Cellular Automata", Commun. Math. Phys., 1984, pp. 15-57.

[6] O. Martin, A. M. Odlyzko, S.Wolfram.: Algebraic properties of cellular automata. Commun. Math. Phys. 93, 219-58 (1984).

[7] P. Guan, Y. He.: Exact results for deterministic cellular automata with additive rules. J. of Statist. Phys. Vol. 43. os. 3/4, 463-478 (1986).

[8] E. Jen.: Cylindrical cellular automata. Commun. Math. Phys, 118, 569-590 (1988).

[9] M. Nohmi.: On a polynomial representation of finite linear cellular automata. Bull. of Informatics and cybernetics, 24 o. 3/4, 137-145 (1991).

[10] S. Inokuchi, K. Honda, H, Y. Lee, T. Sato, Y. Mizoguchi and Y. Kawahara.: On reversible cellular automata with finite cell array. Proceedings of Fourth International Conference on Unconventional Computation, LNCS 3699. (2005).

[11] Choudhury P. P, Nayak B. K, Sahoo S.,"Efficient Modelling of some Fundamental Image Transformations. *Indian Statistical Institute* Tech.Report *No*. ASD/2005/4, 13,2005.

[12] Choudhury P. P et al "theory and application of two dimensional, null boundary, nine neighborhood, cellular automata linear rules" Cornell University Library, 15-aprail, 2008.

[13] A. R. Khan, "Image Processing hardware based on cellular automat", journal of Innova Ciencia, Vol 4, No. 2;Feb 2012.

[14] Songtao Liu, Hongquan Chen, Shaoqing Yang, "An effective filtering algorithm for image salt pepper noises based on cellular automata", congress on image and signal processing, IEEE, 2008.

[15] A. R. Khan and P. P. Choudhury, "VLSI architecture of CAM", *International jour of computer math applications,* vol 33, pp 79-94, 1997.

[16] O. Martin, A. M. Odlyzko, S.Wolfram.: Algebraic properties of cellular automata. Commun. Math. Phys. 93, 219-258 (1984).

[17] P. Guan, Y. He.: Exact results for deterministic cellular automata with additive rules. J. of Statist. Phys. Vol. 43. Nos. 3/4, 463-478 (1986)

[18] E. Jen.: Cylindrical cellular automata. Commun. Math. Phys, 118, 569-590 (1988).

[19] M. Nohmi.: On a polynomial representation of finite linear cellular automata. Bull. of Informatics and cybernetics, 24 No. 3/4, 137-145 (1991)

First Author received the Bachelor degree from Kashmir University in 2005. He has completed his master's degree in computer applications from Rajive Gandhi Technical University and master's degree in mathematics from Barkatullah University. Currently, he is pursuing his PhD from the department of computer science, university of Kashmir. His research interests include cellular automata, Quantum dots, image processing and cryptography.

Second Author received the Masters degree from Kashmir University in 1989. Currently, he is pursuing his PhD from the department of computer science, university of Kashmir. He is an Associate Professor of the Department of Electronics/IT, S. P. College, Srinagar. His research interests include cellular automata, image processing and cryptography.

Third Author

Fourth Author received the Bachelor and Master degree from Kashmir University. In 2005, he received the PhD degree from Department of Computer Science, Kashmir University. In 2003 and 2004, he went to IIT Kharagpur respectively as a visiting scholar. These days, he is working as Associate professor and Head at P. G. Department of Physics, S. P. College Srinagar. His research interests include cellular automata, Quantum Dots and Image Processing.

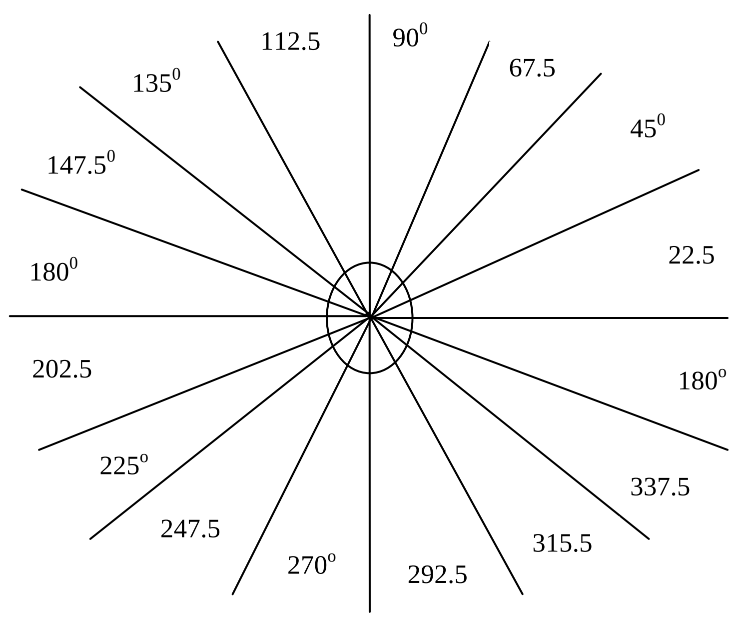

Figure (4): Angle Description of the proposed method