M.Tech.(Computer Science) Dissertation Series

Properties of a Digital Circle and Efficient Computation of its Area

A dissertation submitted in partial fulfillment of the requirements for the M.Tech.(Computer Science) degree of the Indian Statistical Institute

By

Sahadev Bera

Roll No : CS0601

Under the supervision of

Prof. Bhargab B. Bhattacharya

Indian Statistical Institute
203, B.T. Road
Kolkata - 700108

INDIAN STATISTICAL INSTITUTE
203, B.T.Road
Kolkata - 700108

Certificate of Approval

This is to certify that the dissertation thesis titled "Properties of a Digital Circle and Efficient Computation of its Area" submitted by Mr. Sahadev Bera, in partial fulfillment of the requirements for the M. Tech.(Computer Science) degree of the Indian Statistical Institute, Kolkata, embodies the work done under my supervision.

Prof. Bhargab B. Bhattacharya
Advanced Computing Microelectronics Unit
Indian Statistical Institute, Kolkata - 700108

Acknowledgment

I take my pleasure in thanking Prof. Bhargab B. Bhattacharya for his valuable guidance through the dissertation. His pleasant and encouraging words have always kept my spirits up. I would also like to express my sincere gratitude to Dr. Partha Bhowmick and Dr. Arijit Bishnu and the faculty members of the ACMU for their kind help.

I take this opportunity to thank my classmates and seniors of the institute for their heartiest inspiration.

Sahadev Bera
M.Tech.(Computer Science),
Indian Statistical Institute, Kolkata - 700108

Abstract

In this thesis we report some new properties, observations, and algorithms on a digital circle, and their applications to the analysis of geometric information embedded in a digital image. In particular, we develop a new method of detecting the centre, radius and area of a digital circle, based on its number-theoretic interpretation in the digital space. As a sequel, this also leads to a new, efficient and superior method of approximating the value of π, compare to the existing area-based approaches.

Diss/08/02/213

Key Words:

Digital Geometry, Digital Circle, Computation of Pi, Tangent in Digital Space

Contents

Chapter 1

Introduction

1.1 Digital Geometry

Digital Geometry is a new branch of Computer Science. It deals with the geometric features embedded in the digital picture of an object.

Definition 1 *Digital Geometry is the study of geometric or topologic properties of sets of pixels or voxels. It often attempts to obtain quantitative information about objects by analyzing digitized (2D or 3D) pictures in which the objects are represented by such sets [3].*

1.2 Why Digital Geometry?

Consider the problem of drawing an object on computer screen. All real objects lie in Euclidean space, and are spatially continuous. But a computer screen with a rectangular array of pixels is discrete as it is a finite subset of $\mathbb{Z}^2$. Thus, for digital imaging, we have to map the real continuous objects in the Euclidean space to finite discrete sets in the digital space so that screen image appears to be visually same as the real object. On the other hand, given a digital image, one may want to know whether or not some curves (say circle, ellipse etc.) are present in the digital image, and if so, what the features are. In other words, we may need to extract the underlying geometric information from the digital image. Thus for computer graphics, computer vision, and for image processing, we require the knowledge of the geometry of real objects in digital space.

1.3 Main Aspects of Study

(i) Constructing digitized representation of real objects, with emphasis on precision and efficiency.
(ii) Study the properties concerning digital straightness, digital curves, digital convexity, digital planarity, etc.
(iii) Recognition of real objects and extraction of their features (area, length, curvature, surface area, volume, etc.) from a given digital image.

1.4 Applications

The knowledge of digital geometry applies in the area of computer graphics, pattern recognition, image analysis, medical imaging, robot vision, video game, animation etc., to name a few.

Chapter 2

Grids and Digitization

Definition 2 *A colour digital picture can be thought as an array of triples (R,G,B) of integers ranging from 0 to G_{Max}.*

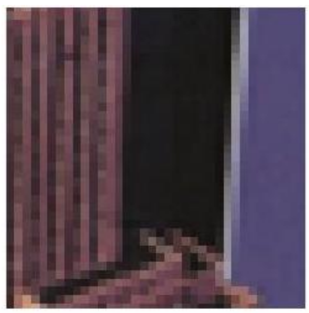

Fig. 2.1: An RGB picture

An RGB picture is composed of three single-valued channels, which can be shown as gray-level pictures.

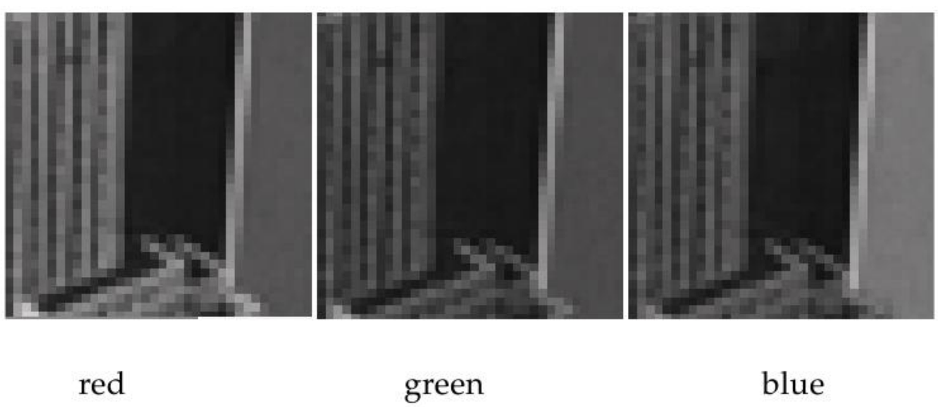

Fig. 2.2: 3 components of a color in RGB space.

Where gray level = integer between 0 and G_{max}
Defaults: $G_{max} = 255$ (one byte), 0 = black, G_{max} =white

2.1 Grid points and grid cells

In 2D, the grid point set is $\mathbb{Z}^2$, and in 3D, the grid point set is $\mathbb{Z}^3$. A grid vertex is shifted by $(0.5, 0.5)$ with respect to a grid point in 2D and by $(0.5, 0.5, 0.5)$ in 3D. A pair of adjacent grid vertices defines a grid edge. A grid square is defined by four grid edges that form a square, and a grid cube in 3D is defined by six grid squares that form a cube.

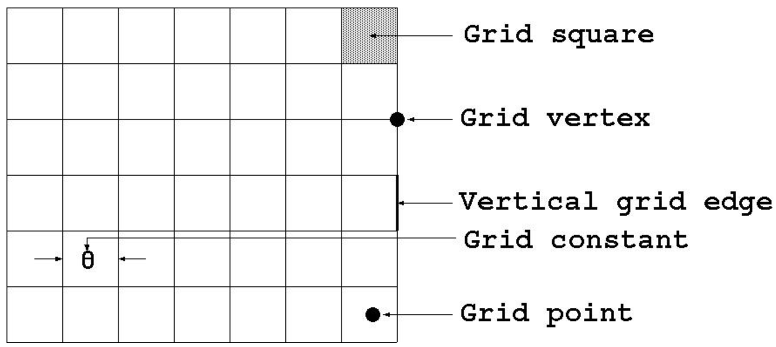

Fig. 2.3: A regular orthogonal grid in the plane.

Definition 3 *A grid cube is also called a 3-cell; a grid square is a 2-cell; a grid edge is a 1-cell; and a grid vertex is*

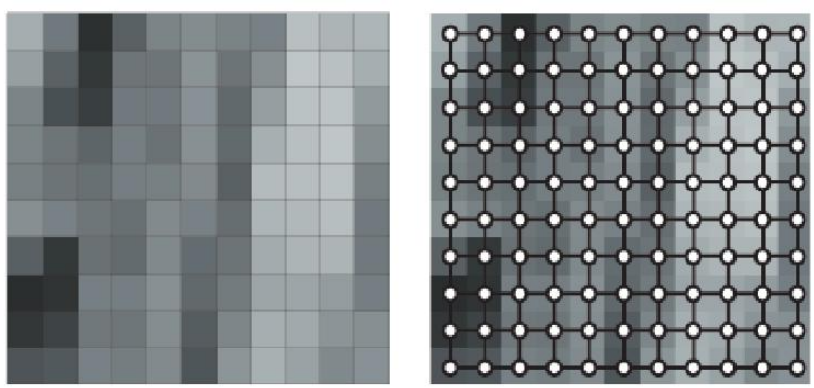

Fig. 2.4: A digital image and its digitized grid structure.

Chapter 3

Digital Straight Line

3.1 Digital Straightness

Definition 4 *A digital arc is called 'straight' if it is the digitization of a straight line segment [3].*

Definition 5 *A set S of grid points is called digitally straight if there is a continuous line whose digitized set contains S.*

Definition 6 *A DSL is the digitized set of an Euclidean straight line.*

Definition 7 *A DSS is the digitized set of an Euclidean straight line segment.*

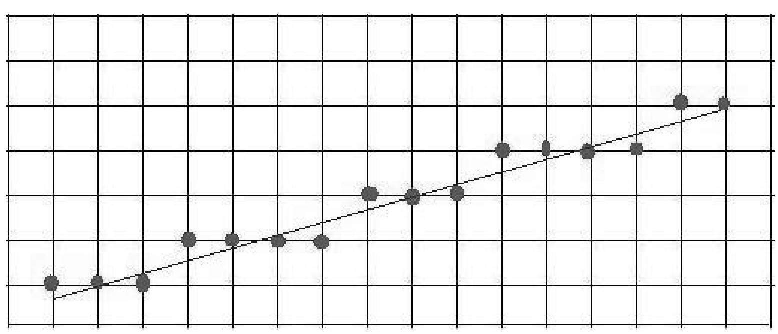

Fig. 3.1: Digital straight line segment or DSS.

Consider the digitization of rays

$$\gamma_{\alpha,\beta} = \{(x, \alpha x + \beta) : 0 \leq x < +\infty\}$$

in the set $\mathbb{N}^2 = (i, j) : i, j \in \mathbb{N}$ for all grid points with non-negative integer coordinates in the plane. As a simplification we assume that $0 \leq \alpha \leq 1$; this is possible due to the symmetry of grid. Such a ray generates a sequence of intersection points $p_0, p_1, p_2, \cdots$ of $\gamma_{\alpha,\beta}$ with the vertical grid lines at $n \geq 0$. Let $(n, I_n) \in \mathbb{Z}^2$ be the grid point nearest to p_n. (If there are two nearest grid points, we take the upper one.) The floor function $\lfloor . \rfloor$ specifies the largest integer not exceeding a given real. Formally,

$$I_{\alpha,\beta} = \{(n, I_n) : n \geq 0 \wedge I_n = \lfloor \alpha n + \beta + 0.5 \rfloor\},$$

and $i_{\alpha,\beta} = i_{\alpha,\beta}(0)i_{\alpha,\beta}(1)i_{\alpha,\beta}(2)\cdots$ is a digital ray with slop α and intercept β, where differences between successive I_n's define chain codes:

$$i_{\alpha,\beta}(n) = I_{n+1} - I_n = 0 if I_n = I_{n+1}, 1 if I_n = I_{n+1} - 1, \tag{3.1}$$

Code 0 interpreted as a horizontal grid increment and 1 specifies a diagonal increment in the grid $\mathbb{N}^2$; see Figure 3.2

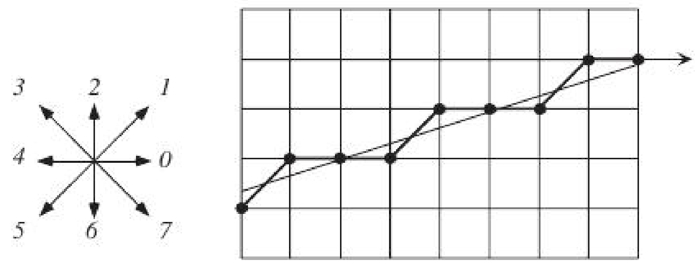

Fig. 3.2: Segment of a digital ray, defined by grid-intersection digitization (as calculated by the Bresenham algorithm).

Definition 8 *Tow digital line segments are called digitally collinear if there exits a common Euclidean straight line whose digitized set contains the sets of digital line segments.*

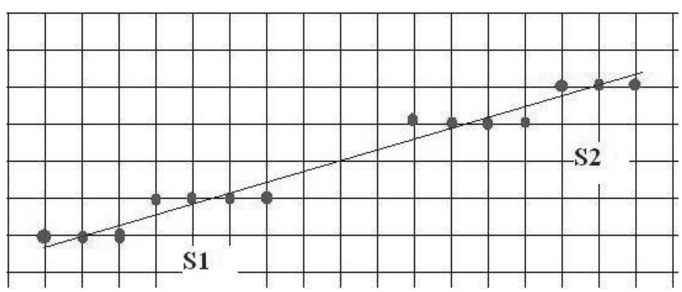

Fig. 3.3: Digitally collinear straight line segments.

Definition 9 *Two digital line segments S_1 and S_2 are called digitally parallel if there exits two parallel Euclidean straight lines A_1 and A_2 such that digitized set of A_i for $i = 1, 2$ contains S_i.*

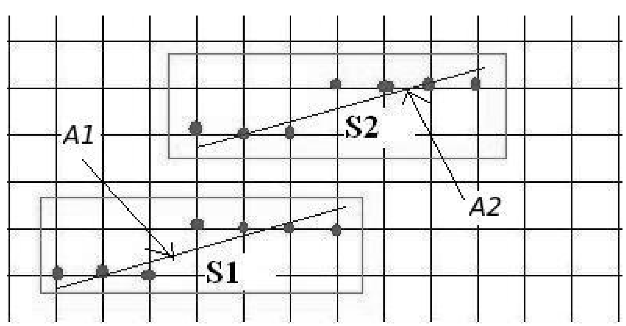

Fig. 3.4: Digitally parallel straight line segments.

Chapter 4

Digital Circle

Drawing circle, ellipse and free-form curves are quite frequent in most CAD packages. For drawing a circle, two parameters; the coordinates of the center and the radius of the circle are required as inputs. In this chapter, we explore the Bresenham scan conversion technique, which is widely used in the CAD graphics packages [6]. A new technique based on a number-theoretic concept as described in [1] is introduced for generating a digital circle of large radius efficiently.

4.1 Bresenham's Digital Circle Drawing Algorithm

The scan conversion technique for circle drawing has some advantages. A circle itself has eight-way symmetry. Consider the Figure 4.1 where a point P on the boundary is reflected seven times. As

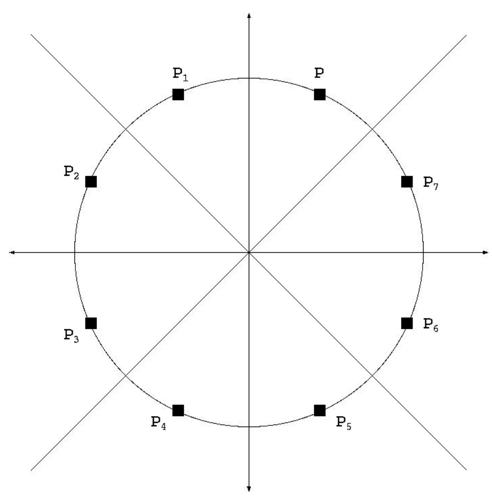

Fig. 4.1: The eight-way symmetry of a circle(the point P is reflected seven times).

a result, scan conversion of one-eight of the circle is enough to generate its remaining seven octants. Bresenham has implemented the line scan conversion algorithm using midpoints to generate the circle. For a circle with radius r and origin at $(0, 0)$, the equation is given by

$$f(x, y) = x^2 + y^2 - r^2 \tag{4.1}$$

Referring to Figure 4.2 for the current midpoint M_c

$$f(M_c) = f\left(x+1, y-\frac{1}{2}\right) = (x+1)^2 + \left(y-\frac{1}{2}\right)^2 - r^2 \tag{4.2}$$

if $f(M_c) < 0$, E is chosen, else SE is selected. Let us consider these two cases separately. If E is

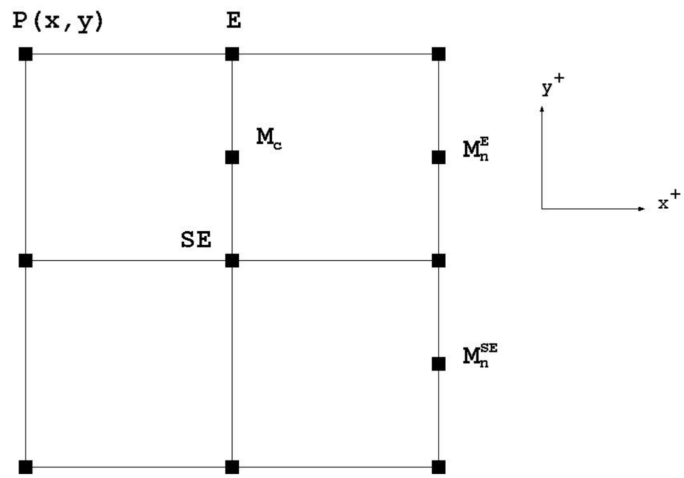

Fig. 4.2: The pixel grid to select the next scan converted point on the circle (from the point $P(x, y)$, the next selectable point is E or SE. The current and the next two possible midpoints are M_c and M_n^E, M_n^{SE}, respectively).

chosen, for the next midpoint M_n^E,

$$f(M_n^E) = f\left(x+2, y-\frac{1}{2}\right) = (x+2)^2 + \left(y-\frac{1}{2}\right)^2 - r^2 \tag{4.3}$$

Therefore, in this case, the increment is

$$\triangle E = f(M_n^E) - f(M_c) = 2x + 3 \tag{4.4}$$

If SE is chosen, again for the next midpoint M_n^{SE},

$$f(M_n^{SE}) = f\left(x+2, y-\frac{3}{2}\right) = (x+2)^2 + \left(y-\frac{3}{2}\right)^2 - r^2 \tag{4.5}$$

Therefore, the increment in this case is

$$\triangle SE = f(M_n^{SE}) - f(M_c) = 2x - 2y + 5 \tag{4.6}$$

Now the problem remain to detect the first midpoint which should be at $(1, r - \frac{1}{2})$ for a circle of radius r and with the at center $(0, 0)$. Then

$$f\left(1, r-\frac{1}{2}\right) = 1 + r^2 - r + \frac{1}{4} - r^2 = \frac{5}{4} - r \tag{4.7}$$

Therefore, the complete pseudocode for drawing a circle is given as follows: Note that the scan conversion is done only for the upper octant of the first quadrant. The remaining portion of the circle is drawn using its property of eight-way symmetry.

```
function BresenhamCircle(int r, BYTE colour){
        // Assume centre is at origin
           x = 0;
           y = r;
           d = 5/4 - r; //d is a decision variable
           OnCirclePoint(x, y, colour);
           while(y>x){ // for upper octant of the first quadrant
               if(d < 0){ // E is selected
                  d = d + 2x + 3;
                  x = x = 1;}
               else { // SE is selected
                  d = d + 2(x - y) + 5;
                  x = x + 1;
                  y = y - 1; }
            OnCirclePoint(x, y, colour);
}
```

The routine OnCirclePoint(x, y, colour) selects seven more points corresponding to the eight-way reflection of the point (x,y). The following pseudocode explains this:

```
function OnCirclePoint(x, y, colour){
         OnPixel( x,  y, colour);
         OnPixel(-x,  y, colour);
         OnPixel( x, -y, colour);
         OnPixel(-x, -y, colour);
         OnPixel( y,  x, colour);
         OnPixel(-y,  x, colour);
         OnPixel( y, -x, colour);
         OnPixel(-y, -x, colour);
}
```

4.2 Number-theoretic Construction of Digital Circle

This method is an alternative way of constructing digital circle developed by Bhowmick and Bhattacharya [1]. Like the Bresenham's algorithm, this is also an integer domain algorithm. But to draw a circle of large radius this is more suitable and take less computation time than Bresenham,s algorithm. The first octant of a digital circle consists of grid points arranged in consecutive horizontal rows as shown in Figure 4.3. This method uses a number-theoretic property of digital circle that describes the sequence of the number of grid points appearing consecutively. Starting from first row recursively, it

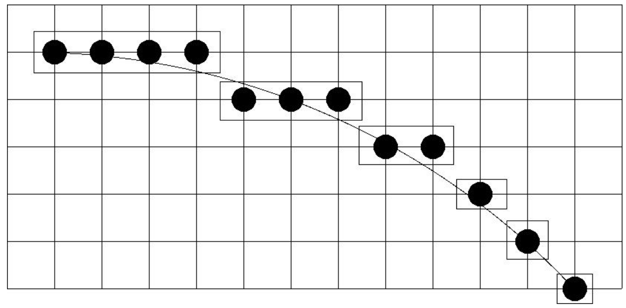

Fig. 4.3: Number-theoretic interpretation of a digital circle.

calculates the length of the next row by the following relation,

$$l_0 = r - 1 \tag{4.8}$$

and

$$l_k = 2r - 2k \tag{4.9}$$

where r is the radius of the digital circle, and k is the number of horizontal row, $k \geq 1$.
The pseudocode of the algorithm DCS (digital circle using square numbers) is described in Figure 4.4 [1].

```
Algorithm DCS (int r){
int i=0,j=r,s=0,w=r-1;
int l=w<<1;
while(j>=i){
      do{ include_8_sym_points(i,j);
          s=s+i;
          i++;
          s=s+i;}while(s<=w);
      w=w+l;
      l=l-2;
      j--;}}
```

Fig. 4.4: Algorithm DCS.

Chapter 5

Tangent to a Digital Circle

5.1 Basic Concepts

In this chapter we investigate the intersection properties of a digital circle with a digital straight line, and then explore certain properties of a tangent to a digital circle. Some of these properties observed here also hold for any digital curve. The digital representation of a curve is different from Euclidean representation though visually they are more or less the same. But from a mathematical point of view, there are significant differences between them. We introduce two new theorems and some definitions of digital curves.

Theorem 5.1.1 *For any curve in Euclidean space there is a unique curve in Digital space.*

Proof. Let the round function be given by [4]:
$round(x) = \lfloor (x + 0.5) \rfloor - - - (1)$
Let $f(x, y) = 0$ be the curve in the Euclidean space. Let corresponding to $x = x_1$(integer), the value of y be y_1(real). So we have to use round function to get an integer value.
Now the round function (1) returns a unique value for y_1, say y_r. So for each point (x, y) for integer x there exists a unique grid point (x, y_r). As all the grid points corresponding to the curve $f(x, y) = 0$ in Euclidean space are unique, the digital representation of f is unique.
□

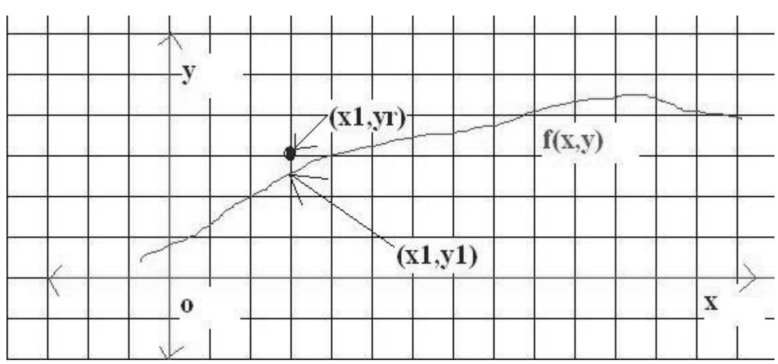

Fig. 5.1: Digitization of a curve.

Theorem 5.1.2 *For a digital curve there exits more than one (actually infinite number of) curves in the Euclidean space of same type (i.e. if the curve is a circle then circles of different radii etc.) whose* *digital representations are same as the given digital curve, i.e., for any digital curve in the Digital space its Euclidean representation is not unique.*

Proof. Let S be the set of grid points of a digital curve and $f(x, y) = 0$ be a curve of the same type in Euclidean space whose digital representation is S. So we have to show f is not unique.

(i) Lower wall case

(a) Let for $x = X$(integer) the value y be $(Y + 0.5)$ i.e. f passes through $(X, Y + 0.5)$; then the corresponding grid point is $(X, Y + 1)$ and the lower wall contains $(X, Y + 0.5)$ shown in Figure 5.2. So there is no curve between f and the lower wall i.e. there is no lower envelope.

(b) Let for $x = X$(integer), the value $y \in (Y + 0.5, Y + 1.5)$; then the corresponding grid point is $(X, Y + 1)$ and the lower wall contains $(X, Y + 0.5)$. So any point (x_1, y_1) such that $x_1 = X$ and $y_1 \in [Y + 0.5, y]$ has the same digital representation i.e. f can pass through any of those points (x_1, y_1). So the lower envelope exists with non zero width.

(ii) Upper wall case (the curve can never touch the upper wall)

(a) Let for $x = X$(integer), the value y be $(Y + 0.5)$; then the corresponding grid point is $(X, Y + 1)$ and the upper wall contains $(X, Y + 1.5)$. So any point (x_1, y_1) such that $x_1 = X$ and $y_1 \in [Y + 0.5, Y + 1.5)$ has the same digital representation, i.e., f can pass through any of those points (x_1, y_1). So the upper envelope of width $(Y + 1.5) - (Y + 0.5) = 1$ exists.

(b) Let for $x = X$(integer), the value $y \in (Y + 0.5, Y + 1.5)$; then the corresponding grid point is $(X, Y + 1)$ and the upper wall contains $(X, Y + 1.5)$. So any point (x_1, y_1) such that $x_1 = X$ and $y_1 \in [y, Y + 1.5)$ has the same digital representation, i.e., f can pass through any of those points (x_1, y_1). So the upper envelope of width $(Y + 1.5) - y > 0$ exists (always happen as a consequence of the density property of real numbers which states that for any two distinct real numbers, say x and y, there exits a rational number, say r, such that $x < r < y$).

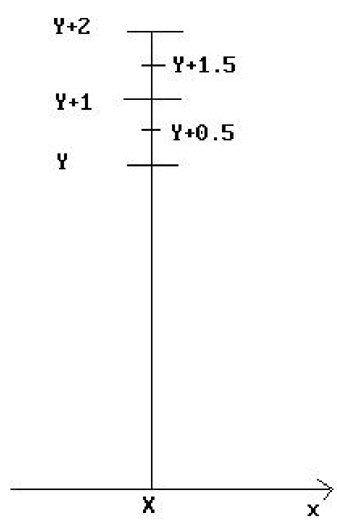

Fig. 5.2: Illustration of lower and upper walls.

So whether the lower envelope exists or not we always get upper envelope of positive width and we can fit more than one curves of same type in the envelope. □

Corollary 1: Corresponding to a straight line segment in digital space, there exists a region as shown in Figure 5.3, in the Euclidean space bounded by a polygonal curve with the left side (the vertical grid line on which the left-most grid point of the digital straight line segment lies) and right side (the vertical grid line on which other end points lies). Every straight line segment that lies within this region with the end points on the left and the right sides, has the same digital representation.

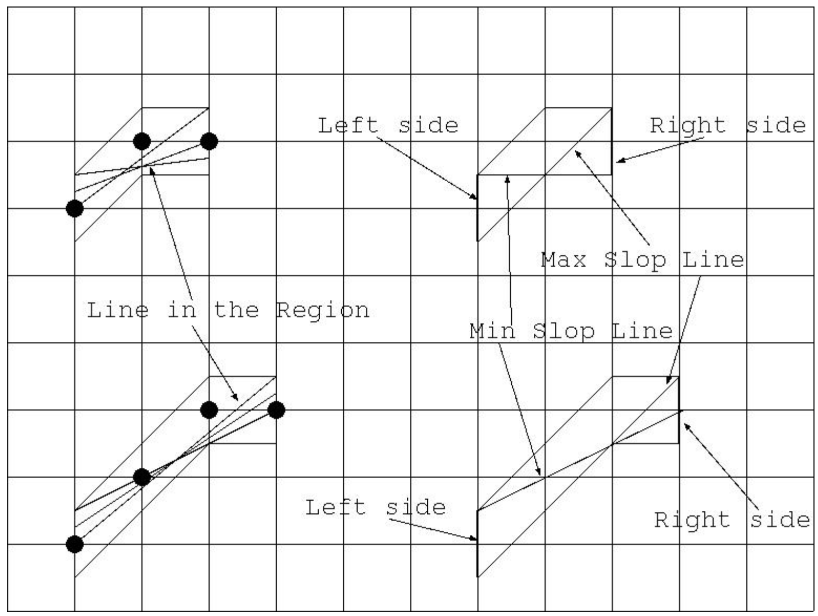

Fig. 5.3: Polygonal region with non-zero area correspond to a DSS or DSL.

For any straight line segment or a straight line this region has an area greater than zero, i.e., the region never collapses with on the straight line segment or the straight line, since it is impossible to have the straight line segment or straight line to touch the upper and lower walls.

For an infinite ray or a straight line this region is a pipe with a non-zero and finite diameter , but with infinite length. The sides of the pipe will be parallel to the ray or the straight line as shown in Figure 5.4. In other words all the infinite rays or lines whose digital representations are the same will be parallel to each other and lie within a pipe. This region may be called corridor of the straight line segment or the straight line.

Corollary 2: For a digital circle, the enveloping region will be an annular ring as shown in Figure 5.5.

5.2 Tangent to a Digital Circle

Definition 10 *A digital straight line segment (DSS) is said to be a tangent to a digital curve if there exists a (more than one is also possible) straight line segment and a curve of the same type in Euclidean space such that their digital representations are same as the given DSS and the digital curve, and the line segment is a tangent to the curve in the Euclidean space.*

This definition applies to any digital curve including a digital circle.

Example: Let l be a DSS and C be a digital circle as shown in Figure 5.6. Let l_1 be a line segment in Euclidean space in the corridor of l. If l_1 is tangent to a circle lying in the annular corridor of C, then the DSS is a tangent to the digital circle C.

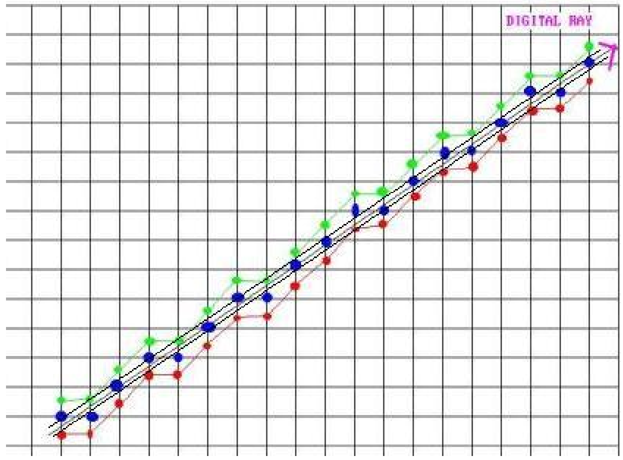

Fig. 5.4: Pipe corridor for a DSL.

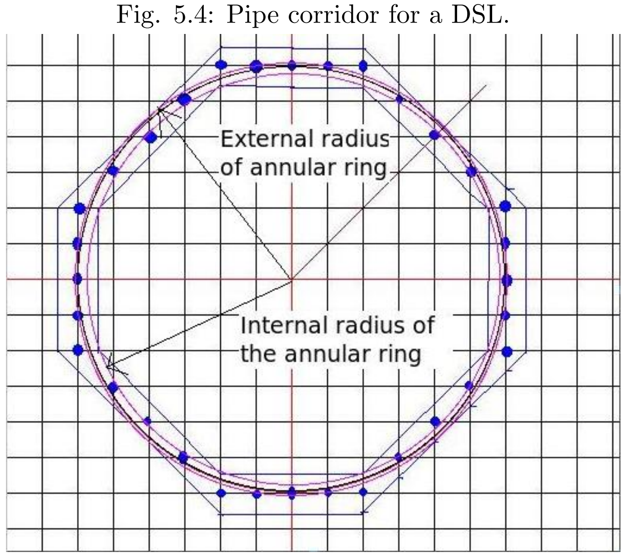

Fig. 5.5: Annular corridor for a digital circle.

Conversely, let a line segment l_1 be tangent to a circle in the Euclidean space. Then the digital representation of l_1 is also tangent to the corresponding digital circle.

5.3 Intersection of a Digital Circle with a Digital Straight Line

Definition 11 *Two curves in the digital space are said to intersect if each Euclidean curve in one corridor corresponding to one digital curve, intersects all the Euclidean curves in other corridor corresponding to other digital curve.*

Example: Let l be a DSS and C be a digital circle as shown in Figure 5.7. Let l_1 be a line segment in Euclidean space in the corridor of l. If l_1 intersects each circle lying in the annular corridor of C, then the DSS intersects the digital circle C.

Conversely, let a line segment l_1 intersect a circle C_1 in the Euclidean space. Then the digital representation of l_1 also intersects the corresponding digital circle.

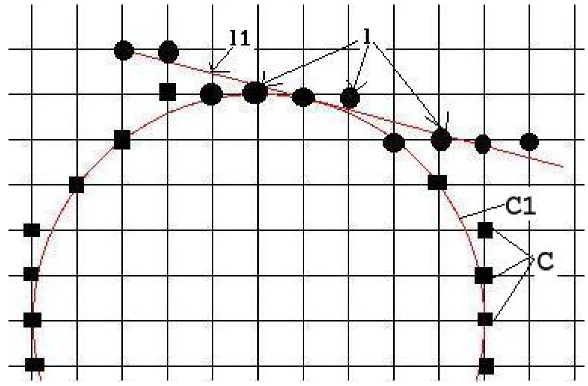

Fig. 5.6: Tangent of a digital curve.

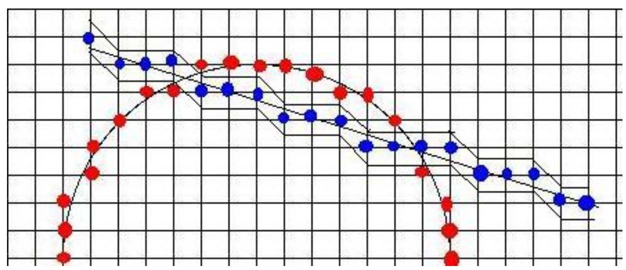

Fig. 5.7: Illustration of intersection.

From the above definitions, we can conclude that if the corridors of two digital curves are intersecting then,

(i) if there exists **at least** one pair of Euclidean curves one from each of the two corridors, such that they touch each other, then the two digital curves also touch each other.

(ii) if no such pair exists, then the two digital curves intersect each other.

In Figure 5.8 we show that for the digital circle C and the DSS l there exists two curves $C1$ and $l1$ in Euclidean space in the corridors of C and l respectively such that $l1$ touch $C1$. So l is a tangent to the digital circle C. But there also exists another circle in the Euclidean space in the corridor of C which intersects $l1$.

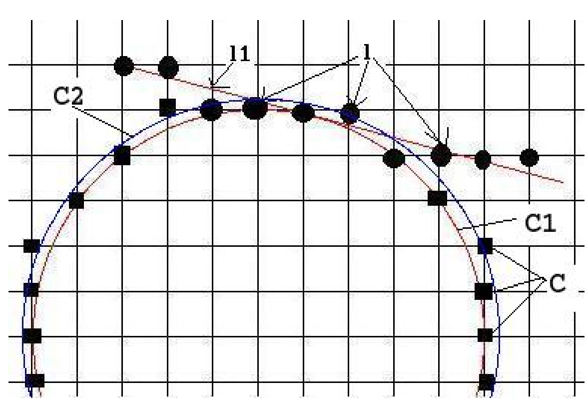

Fig. 5.8: Illustrating the distinction between touch and intersection.

Chapter 6

Properties of a Digital Circle

6.1 Recognition of a Digital Circle

In this section, we present a new method for detecting digital circles in a digital image. The method is based on a property of digital circle, which we call "symmetric property" of three points in digital space. By using this method, we extract the circle with its center and radius. The method consists of two steps: (i) edge detection and (ii) circle extraction. The edge detection procedure produces a binary image with the edges as foreground. Next, we apply circle extraction algorithm on the binary image to identify the circle, and if successful, determine the center and radius of circle. To demonstrate the capability and efficacy of the proposed method, one simple algorithm for detecting the center and radius of a digital circle, based on simple properties of a digital circle, has been reported. Some experimental results have been furnished to elucidate the analytical power and algorithmic efficiency of the proposed approach.

For many years, procedures for image segmentation have been a main research focus in the area of image analysis. Many different approaches have been developed. These may broadly be classified into boundary-based, region-based, and hybrid strategies [7]. There is also a method known for extracting circular wheels based on geometrical transformation [5].

Boundary-based approaches focus on delineating the interface between the object and the surrounding co-objects in the image. Region-based approaches concentrate on delineating the region occupied by the object in the image. Hybrid approaches attempt to combine the strengths of both boundary-based and region-based approaches. The geometrical transformation approach use some transformation e.g., Hough transform to create a distinction between the object and background.

All these methods described above, basically produce a segmented image containing the edges as foreground object. In this section, we introduce a hybrid image segmentation technique, which can automatically find the center and radius of a circle in the digital image. It is also noticed that our method is applicable to an image containing more than one overlapping circles, and once identified, can be used to find centres and radii of the circles.

For a given circle in the Euclidean space, one can easily find its center and radius by taking any three points on the circle, and drawing the perpendicular bisectors of the line segments obtained by joining two pairs of points. But in the digital space, this approach may not hold in general. If we draw two perpendicular bisectors of line segments joining two pairs of points on a digital circle, most of the time, they will meet at a point which is not the center of the digital circle. So we must have additional conditions to be fulfilled by the three points to find the center and radius of the digital circle. In Section 6.1.1, we present one such technique of finding the center and radius of a digital circle, using three special class of points on it.

6.1.1 Symmetric Property of Three Digital Points

Let p_1, p_2, p_3 be three points in digital space. Let the perpendicular bisectors of p_1p_2 and p_2p_3 intersect at c.

Let $r = round$(Euclidean distance between c and p_1), where $round(x)$ is round function defined as, $round(x) = \lfloor (x + 0.5) \rfloor$ [3].

Let $S = q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8$ be the set of points genrated by the reflection of point p_1 taking c as center and r as radius.

If p_2, p_3 are in S then the three points p_1, p_2, p_3 are said to satisfy the *symmetric property.*

Definition 12 *The radius of a digital circle is the Euclidean distance between center and a grid point on the vertical or the horizontal grid line passing through center (the centre of the digital circle lies on a grid point).*

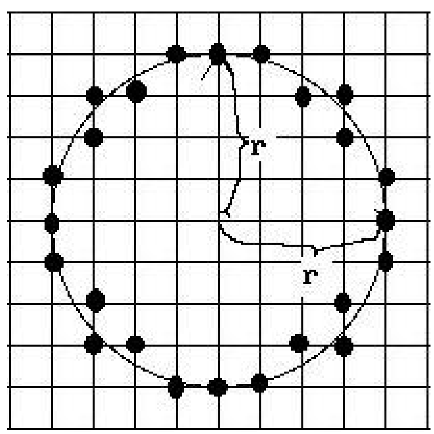

Fig. 6.1: Radius of a digital circle.

Theorem 1 *Any three points in the digital space satisfying the symmetric property can determine a Bresenhum circle passing through the three points.*

Proof. Let p_1 be any grid point defining the circle in first octant, and $p_2, p_3, p_4, p_5, p_6, p_7, p_8$ be the 7 symmetric points of p_1 in the other 7 octants as shown in Figure 6.2.

These 8 points lie on a Euclidean circle whose center can be determined by three points out of these 8 points and due to symmetry of the 8 points this center must be same as the center of the digital circle.

Now we have to find the radius from three grid points and the center. Assume that the radius of a digital circle is always an integer.

According to a lemma by Bhowmick et al. [1], an Euclidean circle of integer radius with the centre at integer coordinates, can never pass through the point $(x, y + 0.5)$ where x and y are integers ($x^2 + y^2 = r^2 \Rightarrow y^2 = r^2 - x^2$, which implies that a fraction is equal to an integer).

Let d be the distance between the point of intersection of Euclidean circle to the vertical grid line and the corresponding grid point (called isothetic distance). Then $d < 0.5$.

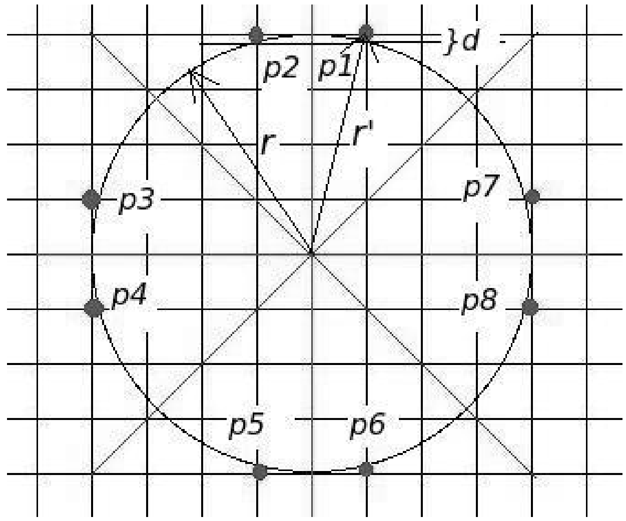

Fig. 6.2: Eight symmetric points.

Let r (integer) be the actual radius of the digital circle and r' be the distance between the center and one of the three grid points. Let e be the error. Then $e \leq d < 0.5$.

Therefore$r = r' \pm e \Rightarrow r' = r \mp e$

Now $\lfloor (r' + 0.5) \rfloor = \lfloor (r \mp e + 0.5) \rfloor = r$, because r is an integer and $e < 0.5$ and $(\mp e + 0.5) < 1$. $\square$

6.1.2 Algorithm for Recognizing a Digital Circle

Input : A digital image.
Output : Centers and radii of the circles present in the input image.

Step 1: Use the Canny edge detection algorithm [2] to find the edges of the given image.

Step 2: Store foreground, i.e., edge pixels in an array A (say).

Step 3: Take three points p_1, p_2, p_3 from A and check whether they satisfy the symmetric property.

Step 4: If yes, generate the corresponding Bresenham circle and match with the image pixels stored in A.

In general a real image may not contain an exact digital circle. So we have to add some tolerance in Step 4, depending on the image.

6.1.3 Reduction in Calculation

The above algorithm take a significant amount of time as it checks all combinations of 3 pixels out of all foreground pixels. But we can reduce the time by the following technique:

(i) Let p_1, p_2, p_3 be three points. Consider the line segments p_1p_2 or p_2p_3 or p_3p_1. Let any one of them be horizontal or vertical, i.e., it coincides with the grid line. Let p_ip_j be such a line segment. If the Euclidean distance between p_i and p_j is even, i.e, if there is an odd number of pixels between p_i and p_j then they may satisfy the symmetric property otherwise not. In other words the above condition is a necessary condition for a digital circle to exists.

(ii) First, we find different components of the foreground pixels. Then we apply the algorithm on each component.

6.1.4 Experiment and Results

The proposed algorithm has been implemented on various synthetic images. Some such images in which the proposed method can recognize circle are shown below.

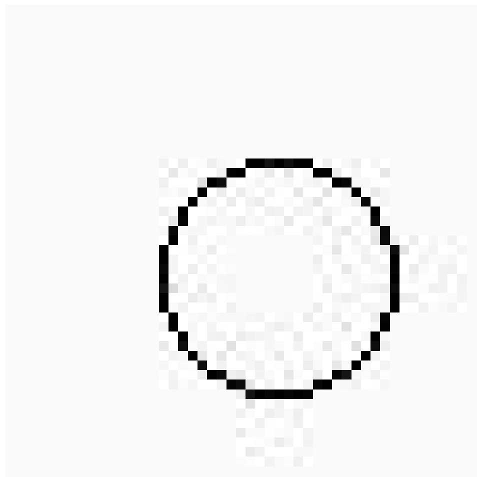

Fig. 6.3: Synthetic circle of radius 12.

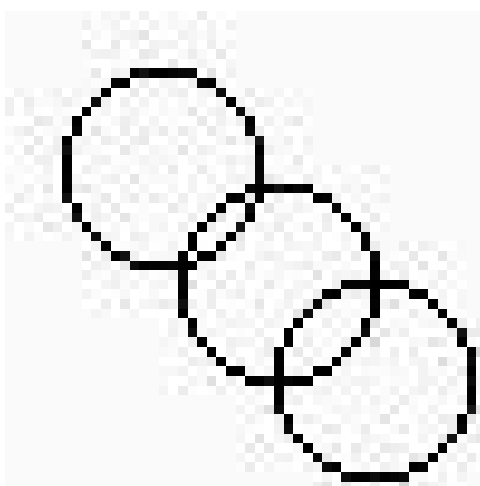

Fig. 6.4: Synthetic circles of radius 10.

6.2 Problem of Tangent Recognition

Given a digital circle C in Digital space and a DSS, consider the problem of ascertaining whether or not the DSS is a tangent to C.

The following two algorithms may be used for this purpose.

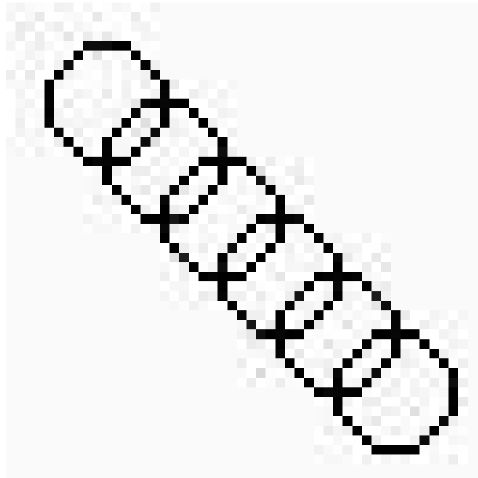

Fig. 6.5: Synthetic circles of radii 6.

Algorithm 1

(1) Find the annular region of the digital circle i.e. the internal radius and external radius of the region.

(2) Find the polygonal corridor corresponding to the DSS.

(3) Then check whether there exits an Euclidean circle in the annular region and a Euclidean line segment in the corridor of DSS such that line segment is a tangent to the circle.

Algorithm 2

(1) Find the center and radius of the digital circle.

(2) Find the polygonal corridor corresponding to the DSS.

(3) Check whether there exits two Euclidean line segments such that the radius of the digital circle lies in the close interval of the distances of two line segments from the center of the digital circle.

6.3 Some Experimental Result on Digital Circle and its Tangent

Some interesting properties have been reported here.

1. The number of common points between a digital circle of radius 1 and its tangent is only one.

2. The number of common points between a digital circle and its tangent is greater than equal to one.

3. The set of common points between a digital circle of radius greater than 1 and its tangent may be a single or more 8-connected components.

Examples:

(i) The set of common points of a digital circle of $Radius = 3$, and its tangent at $(0, 3)$ is an 8-connected components.

(ii) The set of common points of a digital circle of $Radius = 4$, and its tangent at $(1, \sqrt(4^2 - 1^2))$ contains two 8-connected components.

4. It is possible that there are common points on one side of the tangent point, but no common point exists on the other side.

Example: $Radius = 6$; the tangent at $(3, \sqrt{(6^2 - 3^2)})$

The above observations leads to the following open problem.

6.4 Open Problem

Given a digital circle C, with radius r and a point p on C. The set of common grid points between C and its tangent at p may be
(i) a single point, or
(ii) a single connected component, or
(iii) more than one connected components.
Determine a mathematical relation that characterizes the above three cases.

Chapter 7

Computation of Area of Circle and hence π

Computation of the area of a circle is a difficult task and quite impossible to find a good approximations. There are different methods for calculating the area of a circle, some of which are described in this chapter. These methods generally calculate approximate area of one quarter of the circle by dividing the portion into several parts and then one calculates the area of each part separately and adds them. In these methods, if larger number of divisions are considered, a better approximation is obtained and the computed value of π also becomes more accurate. The method proposed here is is based on certain digital geometric properties of digital circle. This yields much better accuracy and speed of convergence.

7.1 History of π [11]

π, which is denoted by the Greek letter , is the most famous ratio in mathematics, and is one of the most ancient numbers known to humanity. π is approximately 3.14 - the number of times that a circle's diameter will fit around the circle. π goes on forever, and can't be calculated to perfect precision: 3.1415926535897932384626433832795028841971693993751.... This is known as the decimal expansion of π. No apparent pattern emerges in the succession of digits - a predestined yet unfathomable code. They do not repeat periodically, seemingly to pop up by blind chance, lacking any perceivable order, rule, reason, or design - "random" integers, infinitely.

In 1991, the Chudnovsky brothers in New York, using their computer, m zero, calculated π to two billion two hundred sixty million three hundred twenty one thousand three hundred sixty three digits (2,260,321,363). They halted the program that summer.

π has had various names through the ages, and all of them are either words or abstract symbols, since π is a number that can't be shown completely and exactly in any finite form of representation. π is a transcendental number. A transcendental number is a number but can't be expressed in any finite series of either arithmetical or algebraic operations. π slips away from all rational methods to locate it. It is indescribable and can't be found. Ferdinand Lindemann, a German mathematician, proved the transcendence of π in 1882.

π possibly first entered human consciousness in Egypt. The earliest known reference to π occurs in a Middle Kingdom papyrus scroll, written around 1650 BCE by a scribe named Ahmes. He began scroll with the words: "The Entrance Into the Knowledge of All Existing Things" and remarks in passing that he composed the scroll "in likeness to writings made of old." Towards the end of the scroll, which is composed of various mathematical problems and their solutions, the area of a circle is found using a rough sort of π.

Around 200 BCE, Archimedes of Syracuse found that π is somewhere about 3.14 (in fractions, Greeks did not have decimals). Knowledge of π then bogged down until the 17th century. π was then called the Ludolphian number, after Ludolph van Ceulen, a German mathematician. The first person to use the Greek letter for the number was William Jones, a Welsh mathematician, who coined it in 1706.

Physicists have noted the ubiquity of π in nature. π is obvious in the disks of the moon and the sun. The double helix of DNA revolves around π. π hides in the rainbow, and sits in the pupil of the eye, and when a raindrop falls into water π emerges in the spreading rings. π can be found in waves and ripples and spectra of all kinds, and therefore π occurs in colours and music. π has lately turned up in superstrings.

π occurs naturally in tables of death, in what is known as a Gaussian distribution of deaths in a population; that is, when a person dies, the event "feels" π. It is one of the great mysteries why nature seems to know mathematics.

7.2 Methods for Computation of Circle Area and hence π

Now we describe some methods in detail for computation of area of a circle and hence the value of π.

7.2.1 Archimedes Method [10]

Archimedes (approximately 285—212 B.C.) was the most famous ancient Greek mathematician and inventor. He invented the Screw of Archimedes, a device to lift water, and played a major role in the defense of Syracuse against a Roman Siege, inventing many war machines that were so effective that they long delayed the final sacking of the city.

Archimedes' mathematical work exhibits great boldness and originality in thought, as well as extreme rigor. Among his mathematical accomplishments is the computation of pi, which is the ratio of the circumference of a circle to its diameter. His approach consisted of inscribing and circumscribing regular polygons with many sides in and around the circle, and computing the perimeter of these polygons as shown in Figure 7.1. This provided him with an upper and a lower bound for π.

In this method as the number of sides of the polygons increases we get better approximation for pi.

7.2.2 Mid Point Method [9]

In this method, the circle divide into no. of parts selected and height of the rectangle is considered as the mid point of the both end of the rectangle. This method gives better approximation than other methods.

7.2.3 Left End Point Method [9]

In Left end point method, the height of the rectangle is calculated at the left point of the rectangle. It gives less approximation than Mid point method but gives better approximation than Right end point method.

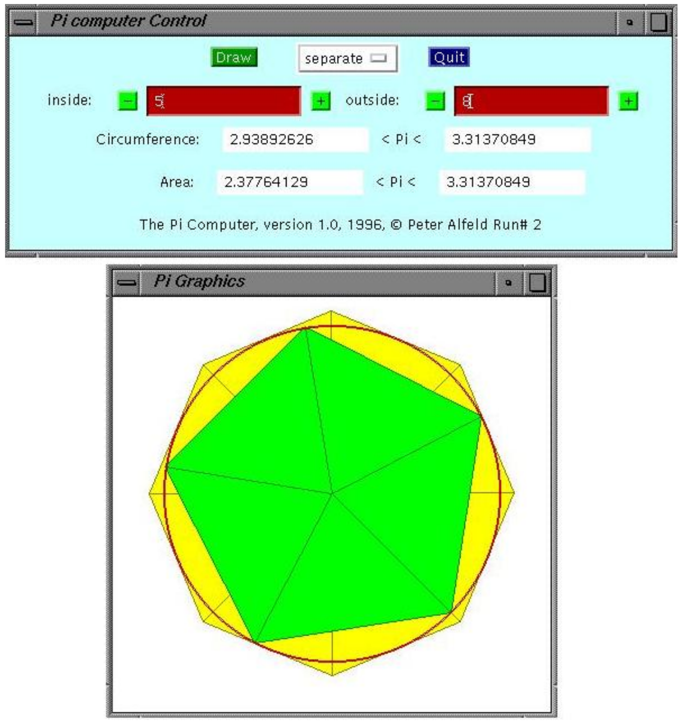

Fig. 7.1: Archimedes' method.

7.2.4 Right End Point Method [9]

In Right end point method, the height of the rectangle is calculated as the right end point of the circle. This methods gives less approximate method.

7.2.5 Simpson Method [9]

This method is given by Simpson. In this method, we have to draw arc instead of rectangle, which require at least three points. So after dividing the circle we will draw the arc and calculate the area using Simpson method. Simpson's Rule for calculating area is given as :

$\int_a^b f(x)dx = h/3[y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + \ldots\ldots\ldots\ldots + 2y_{n-2} + 4y_{n-1} + y_n]$, where $h = (a - b)/n$.

7.2.6 Trapezoidal Method [9]

In this Trapezoidal method, we join the top of the rectangle by straight lines instead of curves and calculate the area using Trapezoidal method. Trapezoidal method is given below :

$\int_a^b f(x)dx = h/2[y_0 + 2y_1 + 2y_2 + 2y_3 + 2y_4 + \ldots\ldots\ldots\ldots + 2y_{n-1} + y_n]$, where $h = (b - a)/2n$.

7.2.7 A New Approach to Computation of area of a Digital Circle and the value of π based Digital Geometry

In this subsection, we introduce a new method for finding the area of a digital circle based on its number-theoretic interpretation [1]. This method is much faster than those based on the simple counting of the number of pixels in the digital circle. The proposed algorithm for computation of π is described in Figure 7.7.

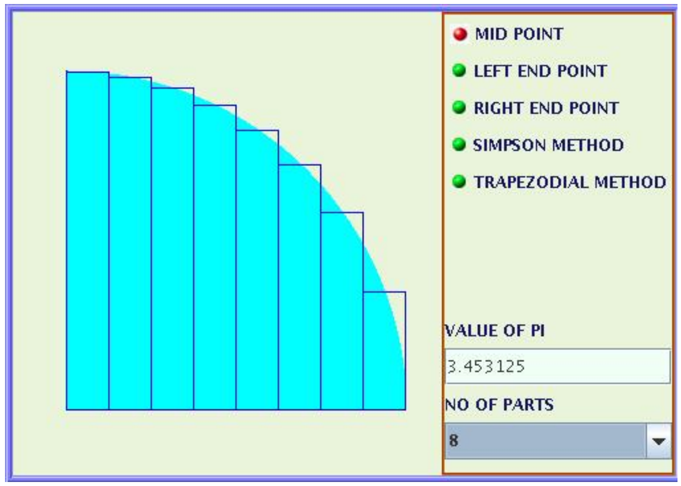

Fig. 7.2: Mid point method.

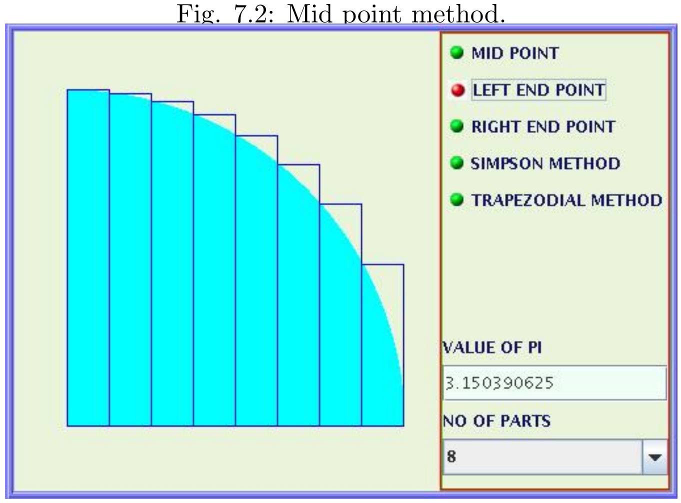

Fig. 7.3: Left end point method.

In our algorithm, we use a novel idea to get more accurate approximation of the actual area of the circle. In the first octant, the digital circle consists of pixels of grids points arranged along consecutive horizontal grid line. For each grid point on the digital circle, we count the number of full squares (cells) and half squares and add an extra half square when we go down to the next grid line by one grid point. Following this concept, the bounded region shown in Figure 7.9 is ab approximation of the area in the first octant. The overall approach is shown in Figure 7.8.

Experimental Results

Experimental results shows that as the radius increases, the relative error in calculating area of the circle decreases significantly, and consequently, we get better approximate value of π. This shown in Figure 7.10.

The proposed method can also be used for calculating area of a digital circle.

7.3 Comparison

All the methods described above calculate area by dividing the region into several parts. As the number of parts increases, calculated area tends to be more close to the actual area. But in digital

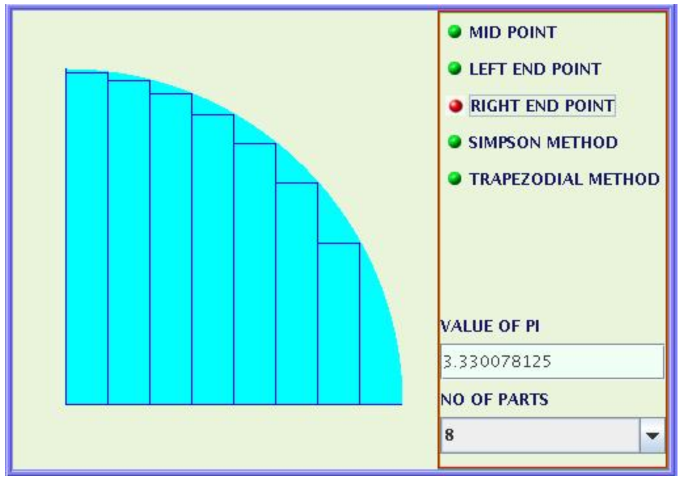

Fig. 7.4: Right end point method.

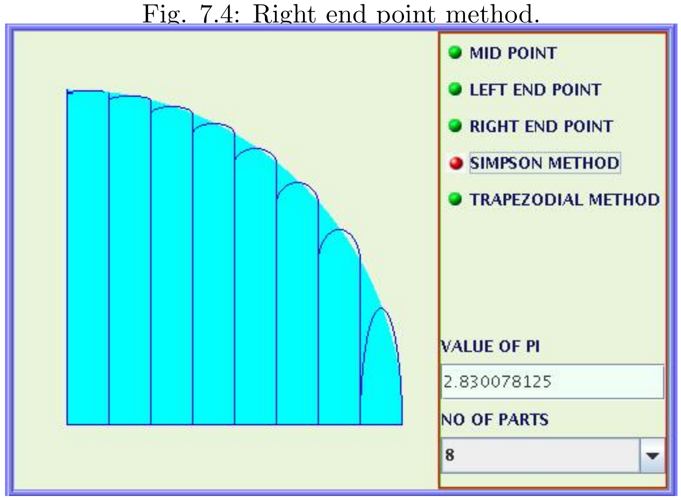

Fig. 7.5: Simpson method.

space, the unit of length is pixel width. Our method calculates the area dividing the region in pixel-width rectangles. Thus in the digital space, our method compares favorably in calculating the nearly accurate value among the methods described above. The accuracy comparison between proposed method and the method of Archimedes is shown in Figure 7.11.

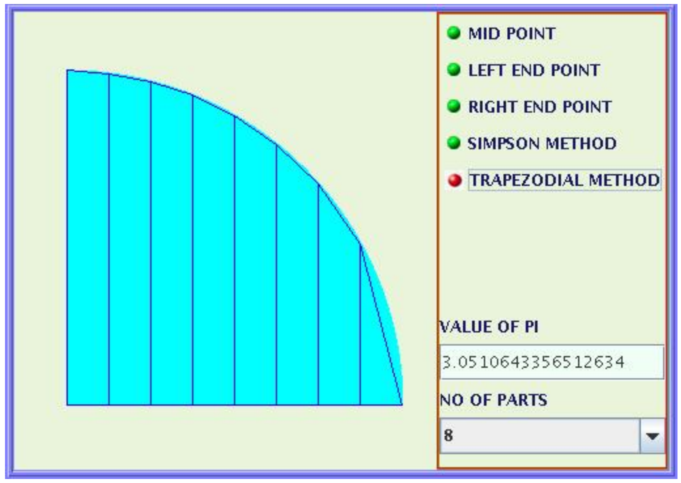

Fig. 7.6: Trapezoidal method.

```
int  AreaOfCircle(int radius){
        i = 0; s = 0;
        fullSquare = 0; halfSquare = 0;
        j = radius;
        w = radius- 1;
        l = w << 1;

        k = 1;
        while( j >= i ){

                do{
                        if( i != j ){
                                fullSquare = fullSquare + (j-k);
                                halfSquare++;
                        }
                        s = s + i;
                        i++;
                        k++;
                        s = s + i;
                }while( s <= w );

                if( (i-1)!=j ){
                        fullSquare--;
                        halfSquare++;
                }

                w = w + l;
                l = l - 2;
                j--;
        }

        area = 8 *  fullSquare + 4 * halfSquare ;
        pi = area / ( radius * radius );
 }
```

Fig. 7.7: Proposed algorithm for computation of area of digital circle.

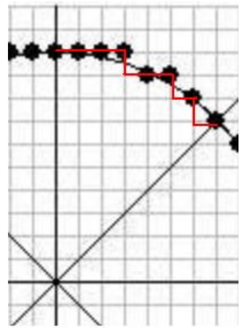

Fig. 7.8: Simple approach counting unit square inside the digital circle.

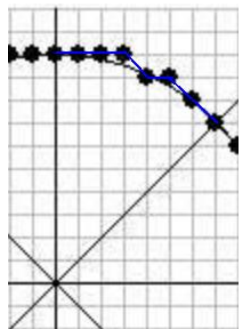

Fig. 7.9: Better approximation according to our approach.

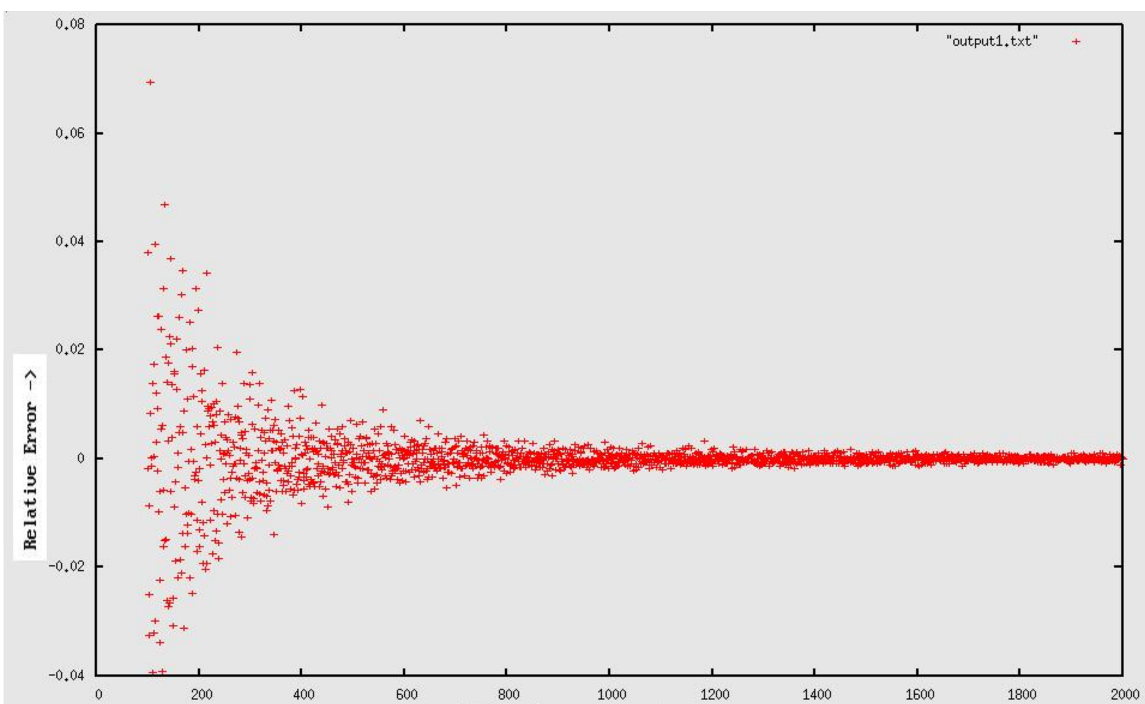

Fig. 7.10: Radius vs. relative error graph.

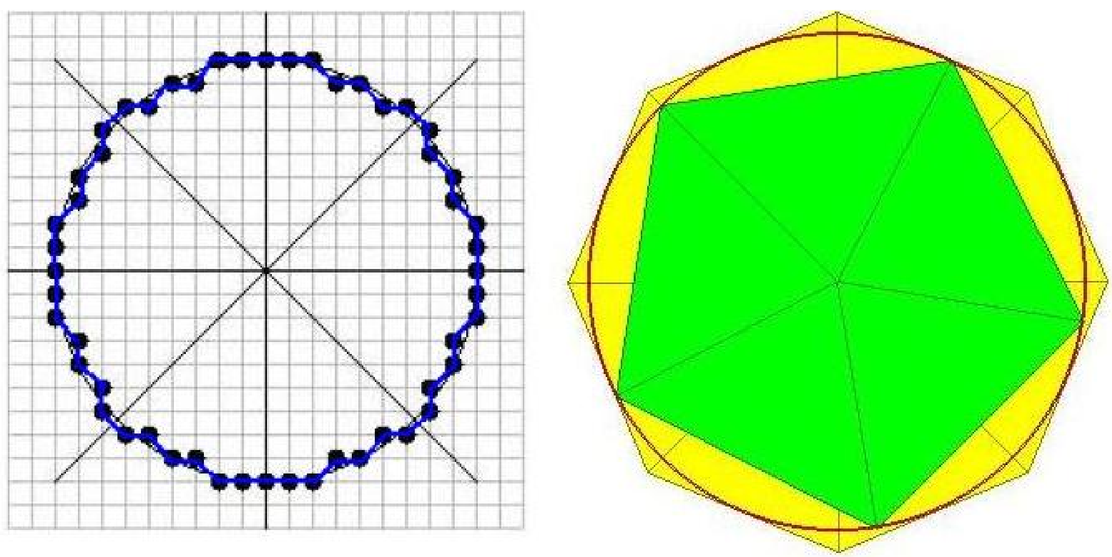

Fig. 7.10: Distinction between the proposed method and the method of Archimedes.

Chapter 8

Conclusion

Several new theoretical interpretations and related applications of digital curves, such as digital line segments and digital circles, have been studied in this report.

In Chapter 6, we have proposed an automatic circle recognition algorithm. Our approach integrates boundary-based image segmentation techniques and certain properties of digital circle. It applies the symmetric property of three points on the edge pixels to find the centers and radii of the existing circles. Our method concentrates on digital geometric features of a digital circle. This approach will be useful in real situations like medical imaging, for example iris contour analysis, wheel detection in a vehicle convoy, and in various nano-science imaging.

In Chapter 7 we have proposed a new and efficient technique for computing the area of digital circle and hence for computing value of π. This method can be useful for digital image analysis in many practical applications such as determination of pore sizes of porous silicon and monitoring prognosis of drugs on biological samples

Bibliography

[1] P. Bhowmick and B. B. Bhattacharya, Number-Theoretic Interpretation and Construction of a Digital Circle, *Discrete Applied Mathematics* (2007), doi:10.1016/j.dam.2007.10.022.

[2] J. Canny, A Computation Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8 (6) (1986) 679-698.

[3] R. Klette and A. Rosenfeld, *Digital Geometry: Geometric Method for Digital Picture Analysis*, Morgan Kaufmann Publishers 2004.

[4] R. Klette and A. Rosenfeld, *Digital Straightness - A Review*, Discrete applied mathematics 139:1-31-3, (2004) 197-230.

[5] M. K. Leung and T. S. Huang, Detecting Wheels of Vehicle in Stereo Images, *Proc. 10th Int. Conf. Pattern Recognition*, 263-267, 1990.

[6] D. P. Mukherjee, *Fundamentals of Computer Graphics and Multimedia*, Prentice Hall 2002.

[7] Y. Zhuge, J. K. Udupa and P. K. Saha, Vector Scale-based Fuzzy-connected Image Segmentation, *Computer Vision and Image Understanding* 101 (2006) 177-193.

[8] D.H. Bailey, J.M. Borwein, P.B. Borwein and S. Plouffe, The Quest for Pi, *Mathematical Intelligence* 19(1) (1997) 50-57.

[9] http://www.mathresource.iitb.ac.in

[10] http://www.math.utah.edu/ alfeld/Archimedes/Archimedes.html

[11] http://witcombe.sbc.edu/earthmysteries/EMPi.html