Digital Circles and Balls:

Characterization, Properties, and Applications to Image Analysis

Sahadev Bera

Advisor
Professor Bhargab B. Bhattacharya

A thesis submitted for the degree of
Doctor of Philosophy

Indian Statistical Institute, Kolkata
India
February 2015

Dedication

To my parents

Acknowledgements

Completing my PhD degree is probably the most challenging activity of my student life. The best and worst moments of my doctoral journey have been shared with many people. It has been a great privilege to spend several years in Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata, India as a researcher. Here comes the opportunity to express my gratitude to those who have supported me in this doctoral journey.

My first debt of gratitude must go to my thesis advisor, Professor Bhargab B. Bhattacharya. I am fortunate enough to be his doctoral student. He patiently provided the vision, encouragement and advise necessary for me to proceed through the doctoral program and complete my dissertation. He has been a strong and supportive adviser to me throughout my doctoral career, but he has always given me great freedom to pursue independent work.

The other distinguished persons, whom I would like to acknowledge are Dr. Partha Bhowmick, Dr. Peer Stelldinger, Dr. Sarmishtha Ghoshal, Dr. Nil Ratan Bandyopadhyay, Dr. Arindam Biswas, and Dr. Shyamosree Pal, with whom I have co-authored several publications. Among them, Dr. Partha Bhowmick deserves a special mention for giving me enough time for discussions and constant guidance throughout this journey. His disciplined and methodical approach has made my work simple.

I also like to thank all the teachers and staff of my department for their help and support, which has gone a long way towards the successful completion of my work. Also a special thanks to my friends and colleagues in and outside the department whose silent and supportive presence and company throughout this journey. Finally, I thank all my family members for supporting me and having faith in me.

Sahadev Bera
`sahadevbera@gmail.com`
February 2015, India

Abstract

In this thesis, we have reported some new theoretical findings, empirical formulations, useful heuristics, and efficient algorithms related to *digital circle*, *digital disc*, and *digital sphere*, along with their practical applications to the analysis of geometric information embedded in a digital image. Detecting digital circles and circular arcs from a digital image is very important in shape recognition. Several image processing techniques were proposed over the years to extract circles and circular arc from a digital image and to interpret related issues. We have proposed a novel technique for the segmentation of a digital circle, which is based on a variant of well known *chord property* and *sagitta property* of an euclidean circle. We show that the radius of a digital circle estimated by this technique is more accurate and thus, this method is very effective in segmenting *circles and circular arcs* from digitized engineering drawings. Next, we use a set of consecutive and concentric digital circles to construct a digital disc. Such a construction raises a new problem of absentee-pixel characterization. We present a novel characterization of the absentee-pixels that appear in the cover of a digital disc with concentric digital circles. The characterization is based on several number-theoretic and geometric properties of a digital circle. The notion of infimum parabola and supremum parabola has been used to derive the count of these absentees. Using this parabolic characterization, we derive a necessary and sufficient condition for a pixel to be a disc absentee, and obtain the underlying geometric properties of the absentees. An algorithm for identifying the absentee-pixels is also presented. Later, we have generalized this idea to 3D and show that the construction of a digital sphere of revolution obtained by circularly sweeping a digital semicircle (generatrix) around its diameter, results in the appearance of some holes (absentee-voxels) in its spherical surface of revolution. We present a characterization of these absentee-voxels using certain properties of digital geometry and show that their count varies quadratically with the radius of the semicircular generatrix. Also, we design an algorithm to fill the holes of the absentee-voxels so as to generate a spherical surface of revolution, which is complete and realistic from the viewpoint of visual perception. We further show that covering a solid sphere by a set of complete spheres also results to an asymptotically larger count of absentee-voxels, which is cubic in the radius of the sphere. Necessary characterization and generation of a complete solid sphere have also been worked out in the final stage.

The segmentation of objects embedded in a digital image is an important task in image processing with numerous applications. In this thesis, we study two specific engineering problems: (i) characterization of micro-pores on a porous silicon (PS) chip by image analysis, and (ii) granular object segmentation for application to agriculture. Whilst regular structures like micro-test tubes and micro-beakers fabricated on a PS chip offer potential platforms for implementing various biosensors, controlling the uniformity of pores during electrochemical etching is a challenging problem. One important objective of such fabrication procedure is to ensure the circularity of pore boundaries. Thus, to tune up and standardize the etching process, a fast image analysis technique is needed to evaluate and quantify the geometry of these nano-scale PS structures. We present an automated approach to pore image analysis: given a top-view image of a PS chip captured by a scanning electron microscope (SEM), the porous regions are segmented and each of the pore boundaries is approximated by a circle. Granular object segmentation is another important task of image processing with several fields of applications including agriculture. A simple algorithm for automated analysis of granulometric images consisting of touching or overlapping convex objects such as coffee bean, food grain, is presented. The algorithm is based on certain underlying digital-geometric features embedded in their snapshots. Using the concept of an outer isothetic cover and geometric convexity, the separator of two overlapping objects is identified. The objects can then be isolated by removing the isothetic covers and the separator. The technique needs only integer computation and its termination time can be controlled by choosing a resolution parameter. The thesis ends with future research directions and a few interesting problems related with digital circle, digital disc, and digital sphere.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

This thesis presents a consolidated study on several interesting digital-geometric aspects of digital circle, digital disc, digital ball, and digital surface of revolution along with some related applications in image processing and computer graphics. This chapter summarizes the overall flow of the thesis. Section 1.1 puts forth the motivation and objective of the thesis. Section 1.2 narrates a brief review of previous work and our contributions. Section 1.3 outlines the organization of the thesis.

1.1 Motivation of the Work

During the last century, the usefulness and scope of digital imaging have expanded rapidly due to the availability of low-cost and reliable digital cameras, scanners, and memory devices. All photos, pictures, diagrams, and videos that we handle in our daily life are now represented in a digital form for the convenience of electronic storage, display, sharing, and transmission. The underlying geometric properties of a digital image, if properly understood and harnessed, are powerful enough to provide several interesting clues for characterizing the objects embedded in the image. Digital-geometric algorithms provide a strong basis and a convenient mechanism for the analysis of an image needed in shape extraction, and for the synthesis of objects needed in computer graphics. Digital-geometric techniques have been used in surface area estimation of digital objects [Wiederhold and Villafuerte (2011)]. Some results from discrete geometry have been successfully applied to medical imaging such as tomographic image reconstruction from the projections [Balázs (2013), Batenburg *et al.* (2013), Hantos and Balázs (2013), Varga *et al.* (2014)]. In this thesis, we focus our study particularly on digital circles, balls, circular arcs, and their various applications. In [Bera *et al.* (2010, 2014d)], a variant of the *chord property* [Weisstein (a)] and *Sagitta property* [Weisstein (b)] of a circle is presented that leads to a faster digital circle extraction algorithm. Such properties help to expedite the process of circular arc recognition and segmentation in engineering artwork and drawings. A new property of digital disc in $\mathbb{Z}^2$ namely "disc absentee" is also introduced [Bera *et al.* (2013), Bhowmick *et al.* (2009)]. Several interesting characteristics of disc absentees are reported. Further, these characteristics are studied for a 3D sphere of revolution in $\mathbb{Z}^3$ [Bera *et al.* (2014a,b)]. The characterization of such absentee-pixels in a 2D digital circle or in a 3D ball is a challenging problem to settle. These results will also aid the construction of a digital surface of revolution in 3D.

In computer graphics, various objects are represented by a set of pixels in $\mathbb{Z}^2$ or voxels in $\mathbb{Z}^3$. For a given curve or a surface, the set of pixels or voxels can be obtained by scanning the object or by capturing a photograph. On the other hand, for the synthesis or construction of real-life objects such as axis-symmetric potteries, digital circles may be used to generate the corresponding *digital surface of revolution*. Recently, Kumar *et al.* [Kumar *et al.* (2010)] presented an algorithm for automated generation and rendering of several potteries, based on the construction of a digital surface of revolution. However, these surfaces inherently consist of several blank (absentee) voxels. In order to fill in these blank holes on the object surface, the locations of these absentee-voxels are needed. The underlying open problem of absentee-voxels characterization has been addressed in this thesis [Bera *et al.* (2013, 2014a,b), Bhowmick *et al.* (2009)].

The segmentation of digital image based on various features is a well-researched topic in image analysis. Fast and accurate segmentation of an image into its constituent regions or objects [Gonzalez and Woods (2001)] is a challenging research problem with numerous practical relevance. Various image segmentation techniques have been extensively studied that achieve efficient and accurate results. Most of them are based on Hough Transform(HT) [Chen and Chung (2001), Chiu and Liaw (2005), Coeurjolly *et al.* (2004), Davies (1988b), Illingworth and Kittler (1988), Leavers (1993)], mathematical morphology [Iwanowski (2007)], level set methods [Malladi *et al.* (1995), Sethian (1996)], watershed transform [Chen *et al.* (2004), Vincent and Soille (1991)], or their variants. There are also modified HT methods [Davies (1988b), Kimme *et al.* (1975), Yip *et al.* (1992)], which are suitable for detecting circles in a digital image. Another class of algorithms utilizes certain geometric properties of a circle to improve their performance [Ho and Chen (1995)].

Several work on object recognition based on the analysis of line segments, ellipses, and appearance factors also appeared in the literature [Chia *et al.* (2011, 2012), Gopalakrishnan *et al.* (2010)]. Also, several well known image segmentation techniques are based on watershed transform [Chen *et al.* (2004), Vincent and Soille (1991)]. The segmentation of a digital curve has numerous engineering applications. For example, it can be used to automate an instrument that is designed to extract the geometric structure of nano-scale objects such as porous silicon by analyzing its captured microscopic image. In [Bera *et al.* (2012)] an algorithm based on digital geometry is presented for structural analysis of a porous silicon image. Quality evaluation of various agricultural products, e.g., corns, coffee-beans, is also an important problem in food processing industry. Recently a fast digital-geometric algorithm is presented in [Bera *et al.* (2014c)] for granulometric image analysis. The proposed curve identification and segmentation technique is very useful in designing an automated system for agricultural screening of food items.

1.1.1 Objective of the thesis

The major contributions of the thesis are summarized below:

A. Theory: Mathematical characterization and study of properties

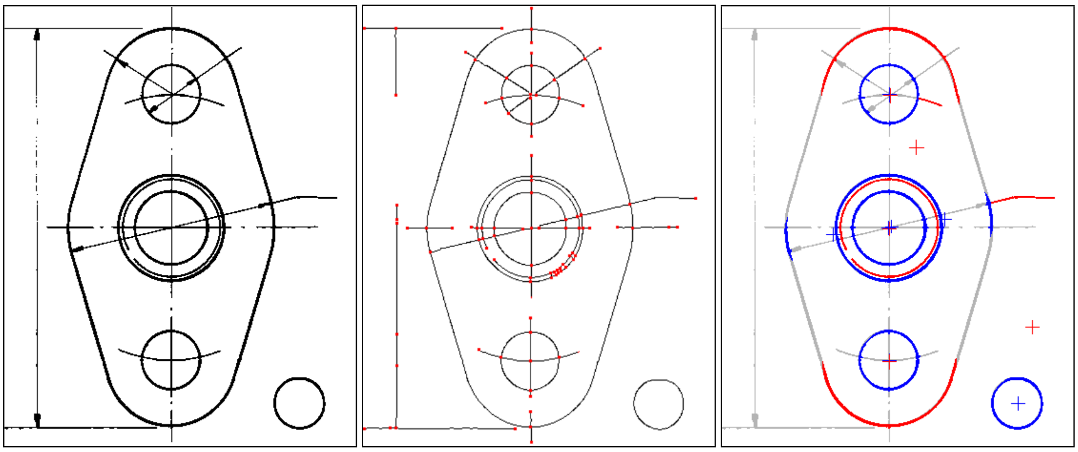
The following three problems related to digital circles, discs, and balls are studied:

- the characterization of a circular-arc image using digital-geometric properties;
- covering a digital disc with concentric circles in $\mathbb{Z}^2$;
- generation of surfaces of revolution in $\mathbb{Z}^3$.

B. Applications: Image analysis for various engineering purposes

We have explored a few application areas where the underlying digital-geometric properties will be helpful in image analysis. They include:

- the analysis of engineering artwork and drawings;
- circularity analysis of nano-scale structures in a porous-silicon image;
- digital geometry based segmentation technique for granulometric applications.



(a) (b) (c)

Figure 1.1: Step-wise snapshots of our experiment on `2007-1.tif`: (a) input image; (b) after finding the intersection points and end points (c) final result.

1.2 Proposed Work in the Background of Prior Art

In this section, we report an overview of the proposed work. Also, for each problem, we present the results of our investigation.

1.2.1 Circular arc detection using chord and Sagitta Properties

Fast and accurate recognition of circles or circular arcs in a digital image is a challenging problem with practical relevance in shape recognition [Davies (1997), Gonzalez and Woods (2001), Sonka *et al.* (1998)]. Circle detection is also important in computer graphics [Pratt (1987)], physics [Chernov and Ososkov (1984), Crawford (1983)], biology and medicine [Biggerstaff (1972), Paton (1970)], and in industry [Carter and Yan (2005), Kasa (1976), Thomas and Chan (1989)]. There exist several algorithms, most of which are based on Hough transform (HT) or its variants [Chiu and Liaw (2005), Coeurjolly *et al.* (2004), Davies (1988b), Illingworth and Kittler (1988), Kim and Kim (2001), Leavers (1993), Xu and Oja (1993)]. The Hough transform (HT) based techniques [Illingworth and Kittler (1988), Leavers (1993)] are widely used to extract digital primitives, such as straight lines, circles, and ellipses. Although HT is robust against noise, clutters, object defects, and shape distortions, its main drawbacks are large requirement of computation time and memory, both of which increase exponentially with the number of parameters. Many techniques were proposed to improve the performance of HT, namely the fast Hough transform (FHT) [Guil *et al.* (1995), Li *et al.* (1986)], circle Hough transform (CHT) [Duda and Hart (1972)], the randomized Hough transform (RHT) [Xu and Oja (1993), Xu *et al.* (1990)] and the adaptive Hough transform (AHT) [Illingworth and Kittler (1987)]. Some modifications have been proposed to improve the performance of CHT. In one such method, the parameter space is decomposed into several lower dimension parameter spaces [Yip *et al.* (1992)]. However, the methods of parameter estimation of a circle based on local geometric properties suffer from poor consistency and location inaccuracy because of quantization error. In order to overcome these disadvantages, Ho and Chen [Ho and Chen (1995)] used global geometrical symmetry of circle to reduce the dimension of the parameter space. On the other hand, several algorithms which do not use histograms in the parameter space have been proposed such as least-square fitting algorithms [Chernov and Lesort (2005), Thomas and Chan (1989)], algorithms based on geometric property of a circle [Chen and Chung (2001), Chung and Huang (2007), Ho and Chen (1995)], and those based on genetic algorithms [Nagao (1993)]. Note that the non-HT based algorithms extract a circle faster than a HT-based method. In addition, a hierarchical approach leads to a significant reduction of both computation time and storage requirement. Based on CHT, a size invariant circle detection algorithm was also proposed [Atherton and Kerbyson (1999)]. In order to improve the performance further, Chen and Chung [Chen and Chung (2001)] proposed an efficient randomized algorithm (RCD) for detecting circles that does not use HT as well as the accumulator needed for saving the information of related parameters. The gradient information of each edge pixel is used [Rad *et al.* (2003)] to reduce the time or memory complexity. Chiu *et al.* [Chiu and Liaw (2005)] also proposed an effective voting method for circle detection, which does not use HT. The UpWrite method [McLaughlin and Alder (1998)] deals with local models of the pixels within small neighborhoods, computed by a spot algorithm, to classify them by their geometric features, for circle detection. Recently, a fast randomized circle detection algorithm is presented in [Jia *et al.* (2011)] to determine the centers and radii of circular components. More recent work related to line and circle detection can be found in [Kolesnikov (2012), Kolesnikov and Kauranne (2014)]. In [Kolesnikov (2012)], a dynamic programming solution to multi-model curve approximation constrained by a given error threshold is proposed. Kolesnikov and Kauranne [Kolesnikov and Kauranne (2014)] have proposed a parameterized model of rate-distortion curve, which produces a multi-model approximation by minimizing the associated cost function. Overall, these methods minimize the requirement of memory while reducing the computational time compared to HT-based methods.

In order to further reduce the time complexity and memory requirement, we propose an improved algorithm [Bera *et al.* (2010, 2014d)] for recognizing digital circles as well as circular arcs, based on the *chord property* [Weisstein (a)] and the *Sagitta property* [Weisstein (b)] of an euclidean circle. Our method detects not only isolated digital circles or circular arcs, but also fragmented arc segments, which may be joined to form a full circle or a larger arc. First, all intersecting points among the input curves are detected, and then all arc segments lying between two consecutive intersection points are extracted. Next, using the *chord property*, the circularity of each arc segment is verified and the resulting circular segments are identified. The *Sagitta property* is then applied to determine the radii of the circular arc segments, and the corresponding centers. Finally, two arc segments with the closest radii and centers are merged iteratively to obtain a complete circle or a larger circular arc segment. For improving the accuracy of computation of radii and centers, a technique based on restricted Hough transform is then used. The attractiveness of this algorithm is that it acts on curve pixels only and extracts digital circles or circular arcs by discarding the straight line segments efficiently, and then merges the detected circular segments to form a complete digital circle or a larger arc. This technique offers a general technique os circular arc detection, and enhances its scope. In order to demonstrate the efficiency of the proposed method, we have considered several benchmark engineering drawings and artwork. We have performed tests on several datasets including the GREC datasets [GREC (2007, 2013)] and SMP dataset. In addition we have prepared a dataset SMP by scanning engineering drawing books, e.g., [Simmons *et al.* (2010)]. One set of experimental results is shown in Fig. 1.1.

1.2.2 On covering a digital disc with concentric circles in $\mathbb{Z}^2$

Following the analysis of a digital circle in an image, we have studied certain aspects of other digital objects like digital disc, sphere, and surface of revolution. There exist various work on the characterization and generation of digital circles, rings, discs, and circular arcs in 2D digital plane [Andres (1994), Andres and Jacob (1997), Chan and

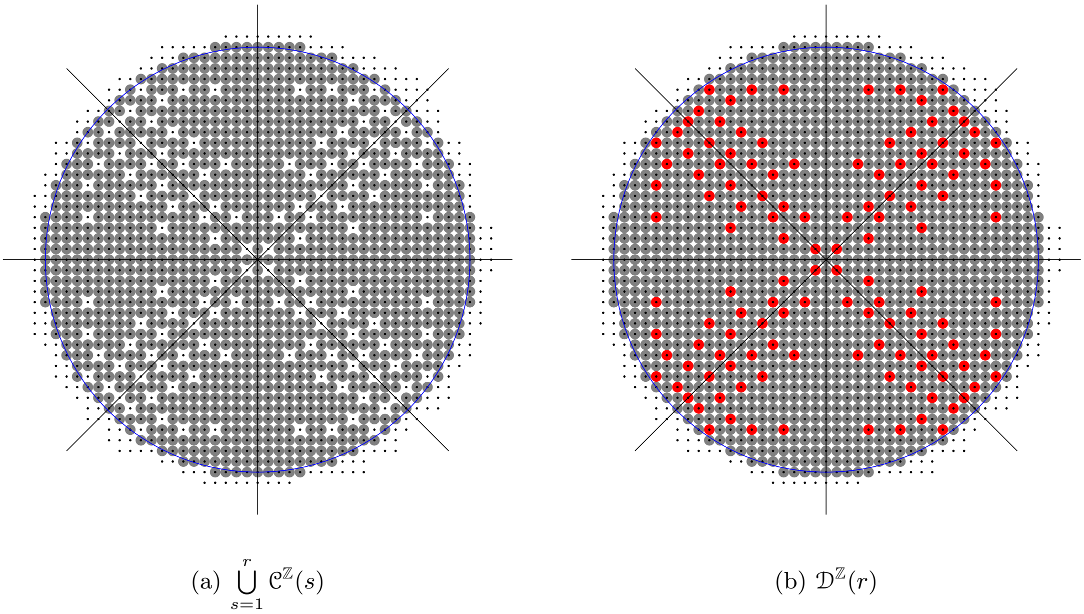

(a) $\bigcup_{s=1}^{r} \mathcal{C}^{\mathbb{Z}}(s)$ (b) $\mathcal{D}^{\mathbb{Z}}(r)$

Figure 1.2: Absentee-pixels (red) for $r = 20$ while covering a digital disc by the circle pixels (gray) in $\mathbb{Z}^2$.

Thomas (1995), Davies (1988a), Doros (1984), Haralick (1974), Nagy (2004), Pal and Bhowmick (2012), Thomas and Chan (1989), Worring and Smeulders (1995), Yuen and Feng (1996)]. Constructing a digital circle in a computationally efficient way is about half-a-century-old problem, originated during the earliest period of scan-conversion technique [Badler (1977), Bresenham (1977), Chung (1977), Danielsson (1978), Doros (1979), Horn (1976), Kulpa (1979), Pitteway (1974), Shimizu (1981)]. In later years, several approaches were suggested for improving the method of circle construction or circle approximation [Bhowmick and Bhattacharya (2008), Blinn (1987), Bresenham (1985), Hsu *et al.* (1993), Mcllroy (1983), Nagy and Strand (2011), Suenaga *et al.* (1979), Wright (1990), Wu and Rokne (1987), Yao and Rokne (1995)]. Added to this, is the more challenging problem of recognizing circular arcs/objects in a digital image, which have several solutions based on the characterization and parameterization of circular arcs [Chattopadhyay *et al.* (1994), Chen and Chung (2001), Chiu and Liaw (2005), Coeurjolly *et al.* (2004), Davies (1987), Ho and Chen (1995), Kim and Kim (2001), Kulpa and Kruse (1983), Nakamura and Aizawa (1984), Pal and Bhowmick (2012), Pla (1996), Rosin and West (1988)]. These geometric

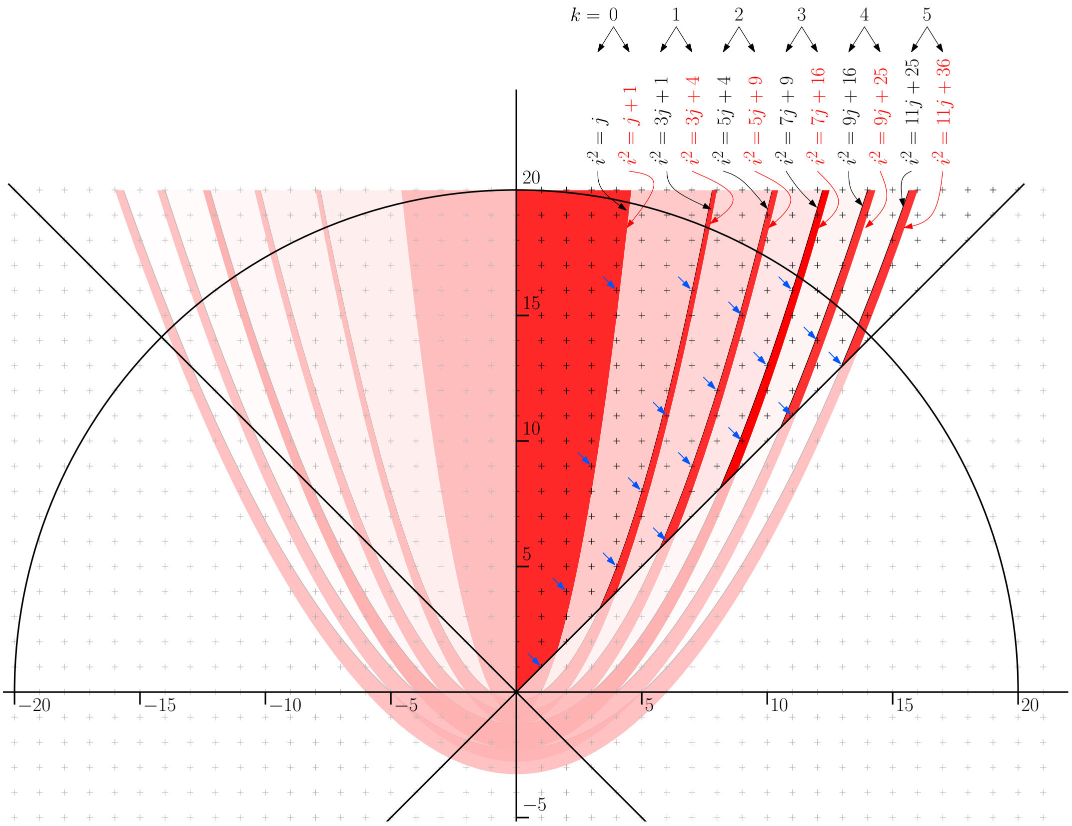


Figure 1.3: Parabolic characterization of the absentee-pixels (pointed by blue arrows). Octant 1 is made bright while other seven octants are dimmed.

characterizations are helpful for solving many scientific problems involving digital circles and circular shapes, since the properties of Euclidean/real circles are often found to be inadequate and inappropriate to deal the underlying problems. Hence, with the emergence of new digital paradigms, such as digital calculus [Nakamura and Aizawa (1984)], digital geometry [Klette and Rosenfeld (2004a)], theory of words and numbers [Klette and Rosenfeld (2004b), Mignosi (1991)], an appropriate characterization of geometric primitives in general, and in digital circles and discs in particular, is highly needed to enrich our understanding of the relevant digital paradigms.

In order to study a digital disc in $\mathbb{Z}^2$, we introduce a new characteristic namely, "disc absentee" [Bhowmick *et al.* (2009)]. When one attempts to cover a digital disc of radius r with concentric digital circles of integer radii $0, 1, 2, \cdots, r$, many uncovered

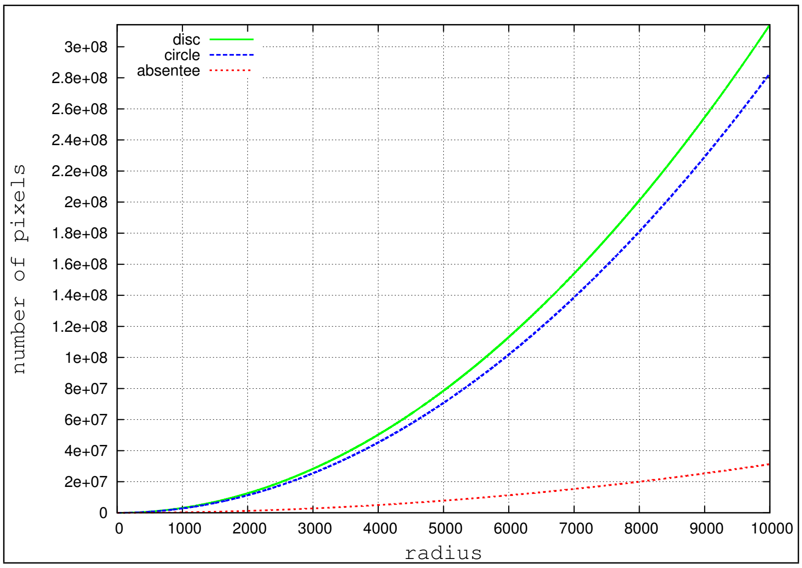


Figure 1.4: Count of circle pixels, absentee-pixels, and disc pixels versus radius.

pixels in the disc are observed. We call these missing pixels as *disc absentee* [Bera *et al.* (2013)]. The absentees occur in multitude—an observation that motivates us to properly characterize and enumerate them. An illustration of absentee-pixels is shown in Fig. 1.2. Our characterization is based on some underlying number-theoretic properties. We show that the count of absentee-pixels increases quadratically with the radius of the disc. The notion of infimum parabola and supremum parabola has been used to study and enumerate such absentees. The underlying parabolas (for $r = 20$) are shown in Fig. 1.3. Using this parabolic characterization, we derive a necessary and sufficient condition for a pixel to be a disc absentee, and derive several geometric properties of absentees. An algorithm to locate the absentees is also presented. Extensive test results have been furnished to substantiate our theoretical findings as well. We compute the count of circle pixels, $\sum\limits_{s=0}^{r} |\mathcal{C}^{\mathbb{Z}}(s)|$, the count of absentee-pixels, $|\mathcal{A}^{\mathbb{Z}^2}(r)|$, and the total count of pixels in a digital disc, i.e., $|\mathcal{D}^{\mathbb{Z}}(r)| = \sum\limits_{s=0}^{r} |\mathcal{C}^{\mathbb{Z}}(s)| + |\mathcal{A}^{\mathbb{Z}^2}(r)|$, with radius r varying from 0 to 10000. These values of pixel-count are plotted against radius r as shown in Fig. 1.4. We observe that as the radius increases, all the three parameters increase with a quadratic dependency on r.

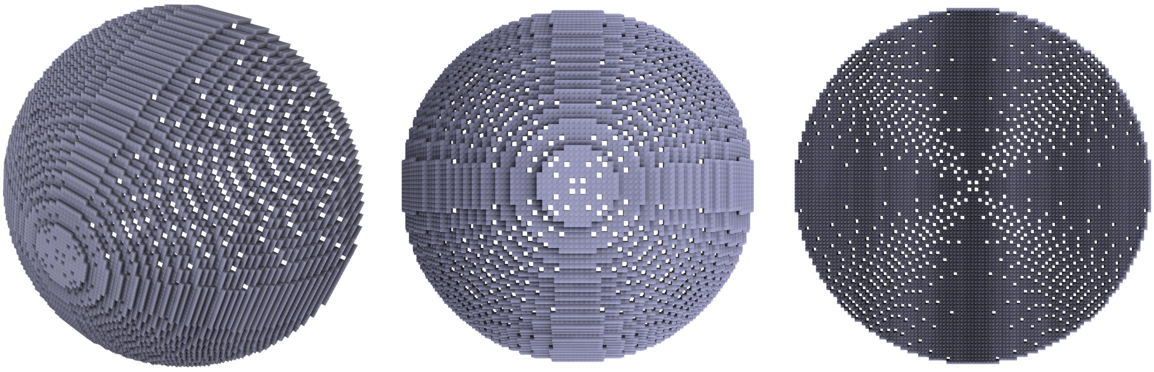


Figure 1.5: Absentee-voxels in the digital hollow hemispherical ball with $r = 50$. Left: oblique view; middle: top view; right: bottom view.

1.2.3 Covering absentees in a surface of revolution in $\mathbb{Z}^3$

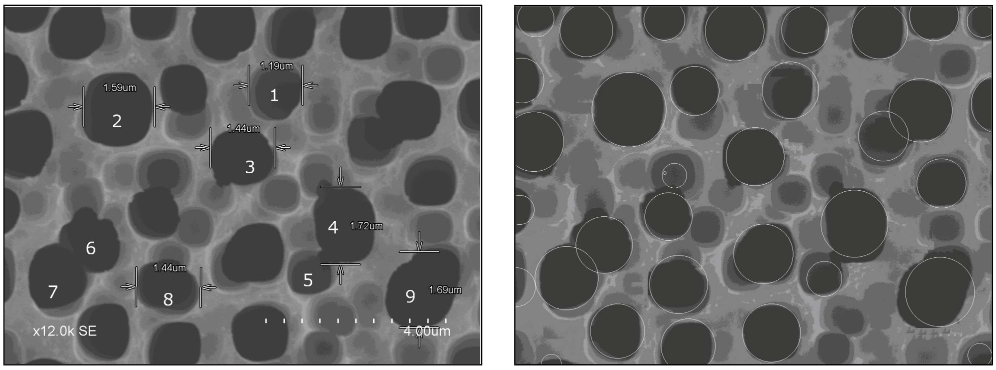
Following the characterization of disc absentees, we extend the idea from $\mathbb{Z}^2$ to $\mathbb{Z}^3$ to study the absentee-voxels in a digital sphere of revolution [Bera *et al.* (2014a,b)]. Over the last two decades, the studies on geometric primitives in 2D and 3D digital space have gained much momentum because of their numerous applications in computer graphics, image processing, and computer vision. Apart from the characterization of straight lines and planes [Brimkov and Barneva (2002), Brimkov *et al.* (2007, 2008), Christ *et al.* (2012), Chun *et al.* (2009), Feschet and Reveillès (2006), Fukshansky *et al.* (2012), Kenmochi *et al.* (2008), Woo *et al.* (2008)], several theoretical work, mostly on digital spheres and hyperspheres [Andres and Jacob (1997), Draine and Flatau (1994), Fiorio and Toutant (2006), Fiorio *et al.* (2006), Montani and Scopigno (1990), Toutant *et al.* (2013), Zubko *et al.* (2010)], have appeared in the literature. Some prior work discuss how to find the lattice points on or inside a real sphere of a given radius [Brimkov and Barneva (2008), Chamizo and Cristobal (2012), Chamizo *et al.* (2007), Ewell (2000), Fomenko (2002), Heath-Brown (1999), Magyar (2007), Tsang (2000)]. Some of them addresses the problem of finding a real sphere that passes through a given set of lattice points [Maehara (2010)]. They are closely related to the determination of lattice points on circles [Cappell and Shaneson (2007), Honsberger (1973)], ellipsoids [Chamizo *et al.* (2009), Kühleitner (2000)], or on several types of surfaces of revolution [Chamizo (1998)].

Very recently, the notion of *discrete spherical geodesic path* between two voxels lying on a discrete sphere has been introduced in [Biswas and Bhowmick (2014)], and a number-theoretic algorithm has been proposed for construction of such paths in optimal time. Based on the recent result of digital calculus [Nakamura and Aizawa (1984)], digital geometry [Klette and Rosenfeld (2004a)], theory of words and numbers [Klette and Rosenfeld (2004b), Mignosi (1991)], we aim to provide a new characterization of digital sphere in 3D discrete space.

Following a similar argument as in the case of a digital disc, we show that the construction of a digital sphere by circularly sweeping a digital semicircle (generatrix) around its diameter results in the appearance of some holes (absentee-voxels) in its spherical surface of revolution. This incompleteness calls for a proper characterization of the absentee-voxels whose restoration in the surface of revolution can ensure a complete construction. In this work, we present a characterization of the absentee-voxels using certain techniques of digital geometry and show that their count varies quadratically with the radius of the semicircular generatrix. A few examples of absentee-voxels in a digital surface are shown in Fig. 1.5. The characterization of such absentee-voxels is important in computer graphics as it helps in generating a complete surface. For creating real life objects, like potteries, the *digital-surface-of-revolution* technique can be applied [Kumar *et al.* (2010)]. Next, we design an algorithm to fill up the absentee-voxels so as to generate a spherical surface of revolution, which is complete and more realistic from the viewpoint of visual perception. We further show that covering a solid sphere by a set of complete spheres also results to an asymptotically larger count of absentees, which is cubic in the radius of the sphere. Necessary characterization and generation of complete solid spheres are also studied. Test results have been furnished to substantiate our theoretical findings.

1.2.4 Circularity analysis of nano-scale structures in porous silicon images

Porous Silicon (PS) based devices have recently emerged as a potential platform for exploring numerous applications to nano-biotechnology, e.g., medical diagnostics, in-vitro pathogen detection, gene identification, and DNA sequencing [Betty (2008), Ghoshal *et al.* (2010, 2011), Granitzer and Rumpf (2010), Stewart and Buriak (2000)]. Because of its non-toxic nature and biodegradability, it is also highly suitable for implementing in-vivo biosensors and drug delivery modules [Anglin *et al.* (2008), Salonen *et al.* (2008)]. The intriguing property of visible light emission from electrochemically etched PS was observed long ago [Cullis and Canham (1991)]. PS chips with an average pore diameter ≤ 2



(a) (b)

Figure 1.6: Snapshots of the experiment on a porous silicon image: (a) input image (b) superimposed fitted circles.

nm [Rouquerol *et al.* (1994)] are called microporous and they admit photo luminescence (PL) at room temperature whereas, those with pore diameters > 50 *nm* [Rouquerol *et al.* (1994)], are called macroporous and they have applications to photonics, sensor technology and biomedicine [Betty (2008), Lehmann (2003), Lin *et al.* (1997), Reddy *et al.* (2001), Saha *et al.* (2006)].

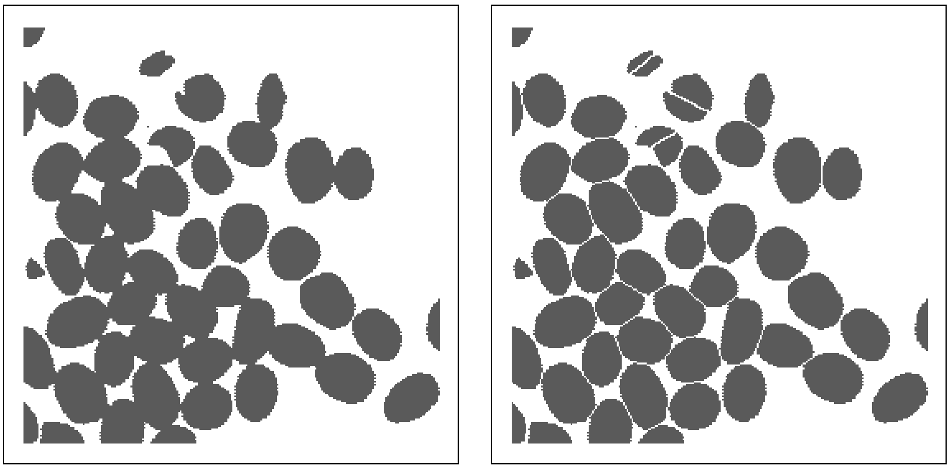
PS chips are fabricated by anodic electrochemical etching of monocrystalline silicon in HF solution. The porosity, which is defined as the fraction of void within the PS layer can be controlled by varying the anodization conditions. Many physical properties of PS, e.g., luminescence, refractive index, and heat conductivity are determined by the fraction of porosity. For biological applications, a uniform arrangement of porous structures called micro-test tubes or micro-beakers having diameters approximately $1-1.5\ \mu m$ are desirable for loading nanoparticles or drugs within the pores [Ghoshal *et al.* (2011)]. Various techniques of creating uniform macroporous structures by controlling formation parameters were reported in the literature [Harraz *et al.* (2005), Vyatkin *et al.* (2002)]. PS also provides a viable platform for observing surface-enhanced Raman scattering (SERS), which is useful in detecting the presence of chemical and biological molecules [Chan *et al.* (2003), Jiao *et al.* (2010)]. Microbeakers on PS (< 100 *nm* in width) with pore size ($> 1.5\ \mu m$ in diameter) can be used as a SERS substrate for various bio-sensing applications. Ideally, these structures should be produced on the PS chip as a regular array of circular pores. However, because of the process uncertainty, pores often appear with deformed boundaries as observed from the captured SEM images [Ghoshal *et al.* (2011)].

In order to formulate the problem, we first observe certain characteristic properties of a PS chip image taken by a scanning electron microscope (SEM). A typical top-view SEM image of a PS chip is shown in Fig. 1.6(a), where the pore boundaries appear as nearly circular objects. As the top surface of a PS chip resembles a 3D terrain, some pores are focused and some are defocused in the captured image. The deep black objects of the image are in the focused plane and thus the corresponding pores appear with sharp boundaries in the image; the lighter black objects represent those pores, which are in the defocused planes and blurred. Thus to analyze the detailed structure of the pores, we first need to perform automatic segmentation to isolate each object from the image.

In [Bera *et al.* (2012)], we address the problem of estimating the circularity of the pore structures based on an image processing technique. Whilst regular structures like micro-test tubes and micro-beakers fabricated on Porous Silicon (PS) offer potential platforms for implementing various biosensors, controlling the uniformity of pores during electrochemical etching is a challenging problem. One important objective of such fabrication procedure is to ensure the circularity of pore boundaries. Thus to tune up and standardize the etching process, a fast image analysis technique is needed to evaluate and quantify the geometry of these nano-scale PS structures. We present an automated approach to pore image analysis: given a top-view image of a PS chip captured by a Scanning Electron Microscope (SEM), the porous regions are segmented and each of the pore boundaries is approximated by a circle. We use a simple digital-geometric technique to determine a best-fitting circle for each pore and compute the Hausdorff distance [Rucklidge (1997)] between them to estimate the quality of pore formation. Experimental results on SEM images of PS structures are shown in Fig. 1.6.

1.2.5 Digital-geometric segmentation for granulometric applications

Fast and automated analysis of a granulometric image consisting of convex objects such as agricultural products, coffee-beans, nuts, cookies, and chocolates, has many practical applications in the field of digital image segmentation. Manual segmentation and counting large number of objects from an image is quite tedious. Several techniques have been proposed for agricultural product inspection [Keagy and Schatzki (1993), Keagy *et al.* (1996), Schatzki and Wong (1989), Schatzki *et al.* (1981, 1997)] using X-ray images. X-ray images



(a) (b)

Figure 1.7: Snapshots of experiment on a coffee-bean image (a) input image (b) result of segmented objects.

provide the internal product details, which allow the analysts to detect the presence of damage due to worms, or other defects by non-destructive (non-invasive) methods; worms contribute to conditions favoring mold growth and toxin production. There exist various prior work on the segmentation of agricultural product [Casasent *et al.* (1996), Talukder and Casasent (1998), Talukder *et al.* (1999a,b)]. Most of them are based on watershed transform [Chen *et al.* (2004), Talukder *et al.* (1999a), Vincent and Soille (1991)], mathematical morphology [Iwanowski (2007)], granulometric methods [Vincent (2000)], or their variants.

Among the various image segmentation techniques, the watershed algorithm is a popular segmentation method, which was originated from the concept of mathematical morphology [Vincent and Soille (1991)]. This technique has been successfully applied for gray-tone image segmentation in various fields including medicine [Cates *et al.* (2005), Cristoforetti *et al.* (2008), Pratikakis *et al.* (1999)], computer vision [Park *et al.* (2005)], biomedicine [Charles *et al.* (2008), Jalba *et al.* (2004)], signal processing [Leprettre and Martin (2002)], industry [Du and Sun (2006), Malcolm *et al.* (2007)], remote sensing [Hall and Hay (2003), Karantzalos and Argialas (2006)], computer-aided design [Razdan and Bae (2003)], and video coding [Wang (1998a)]. The watershed algorithm has also been applied for colored image segmentation [Jung (2007)]. Recently a automatic segmentation of granular objects using local density clustering and gradient-barrier watershed is presented in [Yang and Ahuja (2014)].

When two or more objects in a binary image overlap or touch each other, a single connected object is formed. In order to analyze different characteristics of the objects, it is necessary to segment them into individual components. In this regard, a well known technique is the morphological watershed algorithm, which uses distance transforms [Dougherty (1992), Orbert *et al.* (1993)]. The watershed algorithm segments an image into different regions by treating its inverse distance map as a landscape and the local minima as markers. Each of the segmented regions is labeled with a unique index. Different objects can be separated and identified using the indices of the segmented regions. The effective performance of watershed segmentation depends on the selection of local minima or markers. The spurious markers often lead to over-segmentation, which is a major drawback of a watershed-based algorithm. The performance becomes worse when the objects are irregular-shaped, overlapped or connected, as more spurious local minima tend to occur in the distance transform. Thus, a preprocessing of the markers is needed to improve the performance. Several modified algorithms have been proposed to overcome the over-segmentation issue [Lin *et al.* (2003), Long *et al.* (2007), Umesh Adiga and Chaudhuri (2001)].

The concavity-point-analysis based methods [Farhan *et al.* (2010, 2013), Fernandez *et al.* (1995), Kumar *et al.* (2006), Liang (1989), Wang *et al.* (2012), Wang and Hao (2007), Wang (1998b), Wen *et al.* (2009), Zhong *et al.* (2009)] of the ensemble of convex objects are also widely used. These methods first find the concavity points. Then they determine candidate split lines and finally select the best split lines [Farhan *et al.* (2010), Fernandez *et al.* (1995), Kumar *et al.* (2006), Liang (1989), Wang *et al.* (2012), Wang and Hao (2007), Wang (1998b), Wen *et al.* (2009), Zhong *et al.* (2009)]. Another approach uses concavity points to segment the object contour. Then ellipses are fitted to the segmented contours to determine the split boundary of the clumps [Bai *et al.* (2009), Cong and Parvin (2000), Kothari *et al.* (2009)].

In this thesis, we present a new and efficient algorithm for the segmentation of touching or overlapping convex [Klette and Rosenfeld (2004a)] objects based on digital geometry. We use the concept of outer isothetic cover (OIC) [Biswas *et al.* (2010)] to determine the joining points of the edges of two objects. Next, these joining points are partitioned into suitably matching pairs using the convexity property. The straight line segment that connects a matched pair indicates the separator of two touching or overlapping convex objects, which can be used to isolate them. The advantage of the proposed method lies in the fact that the over-segmentation error is significantly reduced and the required computation is limited to the integer domain only. Experiments have been performed on several images [Chen *et al.* (2004), Sun and Luo (2009)]. Results on a coffee-bean image [Chen *et al.* (2004)] are shown in Fig. 1.7.

1.3 Organization of the Thesis

Keeping in view of the inter-dependence of theory and applications of a digital circle, disc, and a sphere, this thesis has been organized as follows. In Chapter 2 we have reviewed existing techniques and have suggested an improved algorithm [Bera *et al.* (2010, 2014d)] for the segmentation of a circle or a circular arc. In Chapter 3, we present a characterization of the absentee-pixels that appear in the cover of a digital disc with concentric digital circles [Bera *et al.* (2013)]. In Chapter 4, we extend this idea to 3D space in order to construct a digital sphere of revolution by circularly sweeping a digital semicircle (generatrix) around its diameter [Bera *et al.* (2014b)]. The absentee-voxels that appear on the surface are then characterized. In the next two chapters (Chapter 5 and Chapter 6), we study two engineering applications of digital circles and curves. Chapter 5 reports a methodology for circularity analysis in nano-scale structures such as a porous-silicon image [Bera *et al.* (2012)]. An algorithm for automated granulometric image analysis [Bera *et al.* (2014a)] is presented in Chapter 6. Finally, in Chapter 7, we draw the concluding notes on this thesis and discuss possible future directions.

Chapter 2

Chord and Sagitta in $\mathbb{Z}^2$: An Analysis towards Fast and Robust Circular Arc Detection

2.1 Introduction

The properties of a digital circle are well-studied in discrete geometry and have found diverse real-life applications in various fields of science and engineering. Fast and accurate recognition of circles or circular arcs in a digital image is a challenging problem with practical relevance in computer vision [Davies (1997), Gonzalez and Woods (2001), Pratt (1987), Sonka *et al.* (1998)], physics [Carter and Yan (2005), Chernov and Ososkov (1984), Crawford (1983)], biology and medicine [Biggerstaff (1972), Paton (1970)], and industrial engineering [Kasa (1976), Thomas and Chan (1989)]. Most of the existing algorithms for detection of circles and circular arcs are based on the properties of circles on the real or Euclidean plane. However, the properties of a real circle cannot be readily used for analyzing a digital circle, since the latter essentially comprises a sequence of points on the integer plane. In this chapter, we study some of these real-geometric properties of the circle, which can be used to detect digital circles and circular arcs after considering their impact on $\mathbb{Z}^2$.

The chapter is organized as follows. A brief review of existing work related with digital (circular) arc segmentation is presented in Sec. 2.1.1. The chord-and-sagitta properties a real circle are introduced in Sec. 2.2. We derive some important results related to these properties for digital circles, in the respective subsections. Sec. 2.3 describes how these techniques are applied to recognize circular arcs and to estimate their centers and radii. Experimental results are reported in Sec. 2.4. Finally, in Sec. 2.5, we draw the concluding notes and discuss future research issues.

2.1.1 Related Work

There exist several algorithms in the literature for the detection of circular arcs in a digital image. Most of these algorithms are based on Hough transform (HT) or its variants [Chiu and Liaw (2005), Coeurjolly *et al.* (2004), Davies (1988b), Duda and Hart (1972), Illingworth and Kittler (1988), Kim and Kim (2001), Leavers (1993), Xu and Oja (1993)]. Several other techniques have been proposed later to improve the performance of HT, such as Fast Hough Transform (FHT) [Guil *et al.* (1995), Li *et al.* (1986)], Randomized Hough Transform (RHT) [Xu and Oja (1993), Xu *et al.* (1990)], and Adaptive Hough Transform (AHT) [Illingworth and Kittler (1987)]. The main objective of these HT-based methods is either to reduce the computation or to reduce the memory requirement. In one such method, the parameter space is decomposed into several lower dimension parameter spaces [Yip *et al.* (1992)]. It then estimates the parameters of the circles based on local geometrical properties; however, they suffer from poor consistency and location accuracy because of quantization error. To overcome these disadvantages, Ho and Chen [6] used the global geometrical symmetry of circle to reduce the dimension of the parameter space. Xu *et al.* [Xu and Oja (1993)] presented a randomized Hough transform, which reduces the storage requirement and computational time significantly compared to other methods based on the conventional HT.

Li *et al.* [Li *et al.* (1986)] have developed a fast algorithm for the Hough transform that can be incorporated into the solutions to many problems in computer vision such as line detection, plane detection, segmentation, and motion estimation. The fast Hough transform (FHT) algorithm assumes that image space features vote for sets of points lying on hyperplanes in the parameter space. It recursively divides the parameter space into hypercubes from low to high resolution and performs the Hough transform only on the hypercubes when their votes exceed a selected threshold. The decision on whether a hypercube receives a vote from a hyperplane depends on whether the hyperplane intersects the hypercube. This hierarchical approach leads to a significant reduction of both computation and storage. Based on CHT, a size-invariant circle-detection algorithm was proposed [Atherton and Kerbyson (1999)].

Some methods use randomized selection of edge points and geometrical properties of circle instead of using the information of edge pixels and evidence histograms in the parameter space. Kim *et al.* [Kim and Kim (2001)] have proposed a two-step circle detection algorithm, given a pair of intersecting chords, in which the first step is to compute the center of the circle using 2D-HT. In the second step, the 1D radius histogram is used to identify the circle and to compute its radius.

Although HT is robust against noise, clutter, object defect, and shape distortion, its main drawback is high computational time and space. So, several other techniques have also been proposed, which do not use histograms in the parameter space, such as least-squares fitting [Chernov and Lesort (2005), Thomas and Chan (1989)], randomization and geometry [Chen and Chung (2001), Chung and Huang (2007), Ho and Chen (1995)], and genetic algorithm [Nagao (1993)]. These non-HT-based algorithms can extract circles and arcs faster than the HT-based methods, as they do not use histograms in the parameter space. Some of these algorithms are discussed below.

Xu *et al.* [Xu *et al.* (1990)] presented an approach that randomly selects three pixels. The method selects three non-collinear edge pixels and votes for the circle parameters which are found by using the circle equation. In order to improve the performance, Chen and Chung [Chen and Chung (2001)] proposed an efficient randomized algorithm (RCD) for detecting circles that does not use HT as well as the accumulator needed for saving the information of related parameters. The underlying concept in RCD is to first select four edge pixels randomly in the image and then to use a distance criterion to determine whether there might exist a possible circle, and finally to collect further evidence for determining whether or not, it is indeed a circle.

Gradient information of each edge pixel is used in [Rad *et al.* (2003)] to reduce the computing time or the memory requirement. After computing the gradient of the whole image, the pair of anti-parallel vectors are searched to detect circle. Chiu *et al.* [Chiu and Liaw (2005)] proposed an effective voting method for circle detection, which also does not use HT. Rather, it reduces the data set by a sampling technique; once the first two points are chosen, the third one is chosen following certain criteria of circularity. The UpWrite method [McLaughlin and Alder (1998)] works with local models of the pixels within small neighborhoods, computed by a spot algorithm, to classify these local models by their geometric features for circle detection.

Recently Jia *et al.* [Jia *et al.* (2011)] present a fast randomized circle detection algorithm, which can be applied to determine the centers and radii of circular components. Firstly, the gradient of each pixel in the image is computed using Gaussian template.

Then, the edge map of the image, obtained by applying Canny edge detector [Gonzalez and Woods (2001)], is tackled to acquire the curves consisting of 8-connected edge points. Subsequently, for the detection of the center, edge points for each curve are picked up, and the point, passed through by most of the gradient lines of the edge points, corresponds to a center. The radius is estimated from the distances of the corresponding edge points from the center. The algorithm performs much better in terms of efficiency compared to randomized circle detection algorithm (RCD). More recent work related to line and circle detection can be seen in [Kolesnikov (2012), Kolesnikov and Kauranne (2014)]. In [Kolesnikov (2012)], Kolesnikov proposed dynamic programming solutions to multi-model curve approximation constrained by a given error threshold. In [Kolesnikov and Kauranne (2014)], Kolesnikov and Kauranne have proposed a parameterized model of rate-distortion curve, which produces a multi-model approximation by minimizing the associated cost function.

2.1.2 Our Contribution

All prior work use different properties of real circles while detecting circles and circular arcs in the digital plane. The *chord property* [Weisstein (a)] and the *sagitta property* [Weisstein (b)] are two most important properties of real circles, whose deviation in $\mathbb{Z}^2$ has to be analyzed so that they can be efficiently used in the detection of digital circles and arcs. In this work, we study, for the first time, some digital-geometric properties of chord and sagitta. Based on these properties, we propose a novel technique for recognizing digital circles and circular arcs. The chord property in $\mathbb{Z}^2$ is first used to identify the circular curve segments. The sagitta property is then used to determine the radii and the centers of the circular curve segments. The proposed technique not only detects isolated circles and circular arcs, but also joins the concentric arc segments with the same or nearly same radius to make out a full circle or a larger arc. For improving the accuracy of the radii and the centers, a *restricted Hough transform* (rHT) is applied.

2.2 Chord and Sagitta Properties in $\mathbb{Z}^2$

In order to study the properties of chord and sagitta in $\mathbb{Z}^2$, we start with some definitions from the literature [Gonzalez and Woods (2001), Klette and Rosenfeld (2004a)]. A pixel

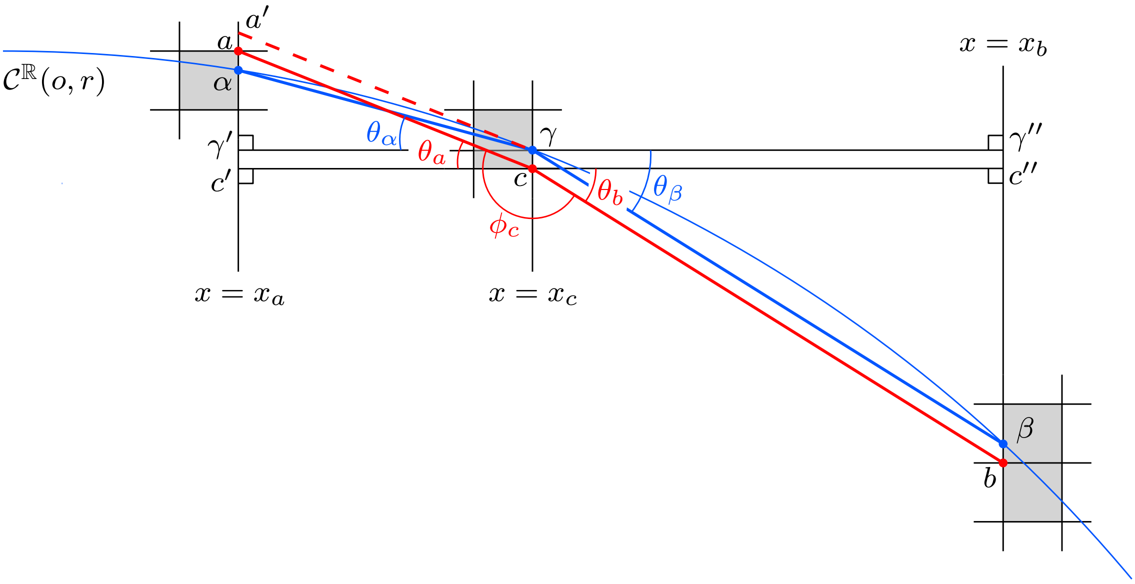


Figure 2.1: Deviation of the chord property (Case 1). Points in $\mathbb{R}^2$ $(\alpha, \beta, \ldots)$ or on the real circle $\mathcal{C}^{\mathbb{R}}(o,r)$ are shown in blue, and points shown in red (a, b, c) belong to the digital circle, $\mathcal{C}^{\mathbb{Z}}(o,r)$. $\mathcal{A}^{\mathbb{R}}(\alpha,\beta)$ is an arc of $\mathcal{C}^{\mathbb{R}}(o,r)$, which corresponds to the given digital (circular) arc $\mathcal{A}^{\mathbb{Z}}(a,b)$. As c changes its place along $\mathcal{A}^{\mathbb{Z}}(a,b)$ such that $|y_\gamma - y_c| < \frac{1}{2}$, the angle ϕ_c gets deviated by $\pm\delta_\phi$.

p is a point in $\mathbb{Z}^2$. Two pixels $p(x_p, y_p)$ and $q(x_q, y_q)$, $p \neq q$, are 8-neighbors of each other if and only if $\max\{|x_p - x_q|, |y_p - y_q|\} \leq 1$. A sequence of pixels defines a (8-connected) digital curve if and only if each pixel, excepting the first one and the last one, has exactly two neighbors in its 8-neighborhood. For an open digital curve, the first and the last pixel have one 8-neighbor each; for a closed digital curve, they have two 8-neighbors each.

2.2.1 Chord Property

We show here the deviation of chord property for digital circular arcs. See Figure 2.1 for an illustration. Let $\mathcal{C}^{\mathbb{R}}(o,r)$ be the real circle having center at $o(0,0)$ and radius $r \in \mathbb{Z}^+$. Let $\mathcal{A}^{\mathbb{R}}(\alpha,\beta)$ be an arc of $\mathcal{C}^{\mathbb{R}}(o,r)$ having endpoints $\alpha(x_\alpha, y_\alpha)$ and $\beta(x_\beta, y_\beta)$. Let $\gamma(x_\gamma, y_\gamma)$ be another point on $\mathcal{A}^{\mathbb{R}}(\alpha,\beta)$. Then, by the *chord property* of real circle, $\alpha\beta$ subtends a fixed angle ϕ_γ at γ, irrespective of its position on $\mathcal{A}^{\mathbb{R}}(\alpha,\beta) \smallsetminus \{\alpha,\beta\}$ [Weisstein (a)].

However, this is not true for a digital (circular) arc, as shown below.

Let $\mathcal{C}^{\mathbb{Z}}(o,r)$ be the digital circle corresponding to $\mathcal{C}^{\mathbb{R}}(o,r)$, and $\mathcal{A}^{\mathbb{Z}}(a,b)$ be the digital arc corresponding to $\mathcal{A}^{\mathbb{R}}(\alpha,\beta)$, where $a(x_a,y_a)$, $b(x_b,y_b)$, $c(x_c,y_c)$ are the respective integer points corresponding to α,β,γ; ϕ_c be the angle subtended by the line segment ab at c; c' and γ' be the respective feet of the perpendiculars dropped from c and γ to the line $x = x_a$, and c'' and γ'' be those from c and γ to $x = x_b$. Let θ_a, θ_α, θ_b, and θ_β be the acute angles subtended at c and γ by the corresponding perpendiculars. Then, the angle ϕ_c deviates from ϕ_γ by an amount depending on the position of c in $\mathcal{A}^{\mathbb{Z}}(a,b)$. In particular, we have the following theorem.

Theorem 2.2.1 *If $\mathcal{A}^{\mathbb{Z}}(a,b)$ is a digital circular arc with a point c on it, then the circumferential angular deviation of c from its corresponding point γ on the real arc is less than* $\sin^{-1}\frac{1}{ac} + \sin^{-1}\frac{1}{cb}$.

PROOF. Let $x_{ac} = x_c - x_a$, $y_{ca} = y_a - y_c$, $x_{\alpha\gamma} = x_\gamma - x_\alpha$, $y_{\gamma\alpha} = y_\alpha - y_\gamma$, $x_{cb} = x_b - x_c$, $y_{bc} = y_c - y_b$, $x_{\gamma\beta} = x_\beta - x_\gamma$, $y_{\beta\gamma} = y_\gamma - y_\beta$.

As shown in [Bhowmick and Bhattacharya (2008)], a real circle with integer center and integer radius never intersects a grid line at the middle between two consecutive grid points. Hence, we get

$$|x_{ac} - x_{\alpha\gamma}| < 1,\ \ |y_{ac} - y_{\gamma\alpha}| < 1, |x_{cb} - x_{\gamma\beta}| < 1, |y_{bc} - y_{\beta\gamma}| < 1. \tag{2.1}$$

Now, we have two possible cases as follows, which are illustrated in Figure 2.1 and Figure 2.2.

Case 1 x-coordinates of both α and γ are integers. W.l.o.g., let α and γ be in Octant 1. Then, $x_a = x_\alpha$ and $x_c = x_\gamma$, whence $x_{ac} = x_{\alpha\gamma}$. So, for maximum deviation, $y_a > y_\alpha$ and $y_\gamma > y_c$. Now, consider the line $\gamma a'$ parallel to ac. The circumferential angular deviation at c is $\theta_{a\alpha} = \angle a'\gamma\alpha$. Using simple geometric property in $\Delta a'\gamma\alpha$, we get $a'\gamma \sin\theta_{a\alpha} \leq \alpha a'$. By Eqn. 2.1, we have $\alpha a' = \alpha a + aa' = \alpha a + c\gamma = |y_{ac} - y_{\gamma\alpha}| < 1$. As $a'\gamma = ac$, we get $\theta_{a\alpha} < \sin^{-1}\frac{1}{ac}$.

When y-coordinates of both α and γ are integers, we also get $\theta_{a\alpha} < \sin^{-1}\frac{1}{ac}$, the proof being similar as above. And when β and γ have integer x- or y-coordinates, the proof is again similar, and we get $\theta_{b\beta} < \sin^{-1}\frac{1}{bc}$.

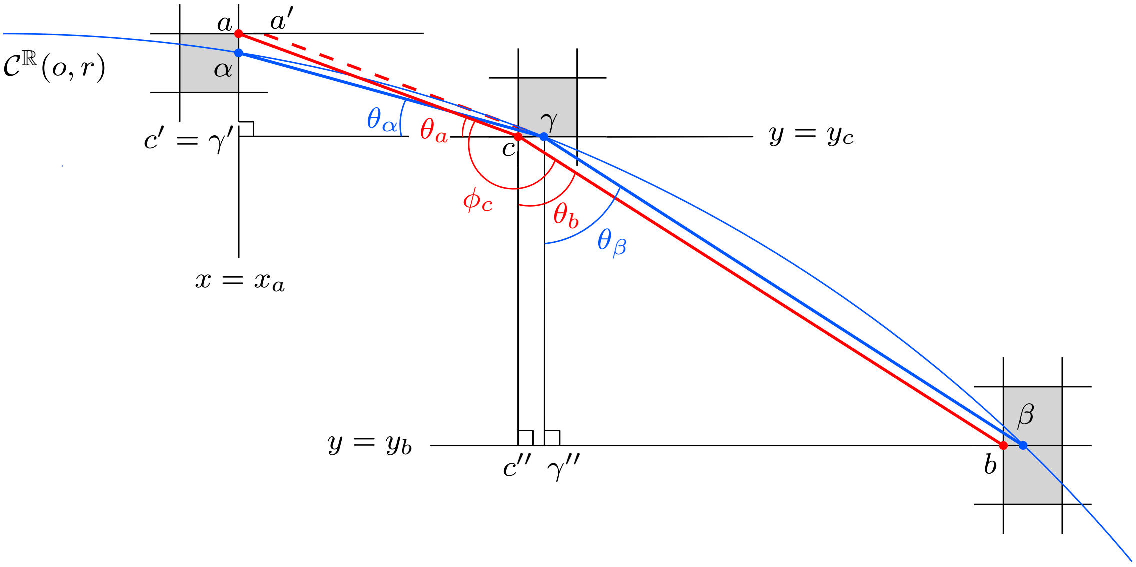


Figure 2.2: Deviation of the chord property (Case 2).

Case 2 x-coordinate of α and y-coordinate of γ are integers. W.l.o.g., let α be in Octant 1 and γ in Octant 2. Then, $x_a = x_\alpha$, $|y_a - y_\alpha| < \frac{1}{2}$, $y_c = y_\gamma$, $|x_c - x_\gamma| < \frac{1}{2}$. For maximum deviation, $y_a > y_\alpha$ and $x_\gamma > x_c$. Now, consider the line $\gamma a'$ parallel (and equal in length) to ac. The circumferential angular deviation at c is $\theta_{a\alpha} = \angle a'\gamma\alpha$. Similar to Case 1, in $\Delta a'\gamma\alpha$, we get $a'\gamma \sin\theta_{a\alpha} \leq \alpha a'$. As $aa' = c'c'' = c\gamma < \frac{1}{2}$ and $\alpha a' < aa' + a\alpha < \frac{1}{2} + \frac{1}{2} = 1$, we get $\theta_{a\alpha} < \sin^{-1}\frac{1}{ac}$.

For a different combination on integer values of x- and y-coordinates, the proof is similar. □

From the above theorem, it is clear that the circumferential angular deviation is dominated by the smaller value between ac and cb. We have the following corollary if they are equal.

Corollary 2.2.2 *If m is the middle pixel (one of the two if it is not unique) of $\mathcal{A}^{\mathbb{Z}}(a,b)$, then the maximum possible deviation of ϕ_m from ϕ_γ is given by*

$$\delta_\phi < 2\sin^{-1}\frac{1}{am} < 2\sin^{-1}\frac{2}{ab}, \quad \textit{since } am + mb > ab. \tag{2.2}$$

Theorem 2.2.1 also implies that if ac or cb is sufficiently small, then the total error would be unacceptably large. To measure this, we define *circumferential angle relative*

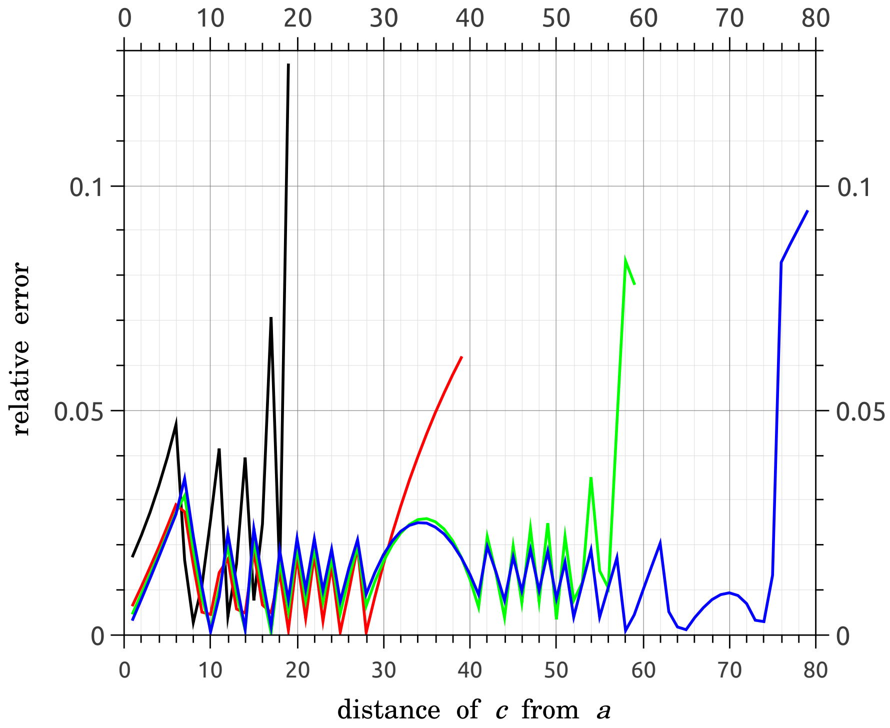


Figure 2.3: Circumferential angle relative error $\varepsilon^r_{c\in ab}$ versus distance of c from the endpoint a of the digital arc ab of circle having radius $r = 50$. Four different digital arcs of length 20 (black), 40 (red), 60 (green), and 80 (blue) are considered.

error at the pixel c lying on the digital circular arc ab of radius r as $\varepsilon^r_{c\in ab} = \frac{|\phi_\gamma - \phi_c|}{\phi_\gamma}$. For a given digital arc ab, we vary the position of c on ab and compute the corresponding value of $\varepsilon^r_{c\in ab}$. Figure 2.3 shows a set of four such plots of $\varepsilon^r_{c\in ab}$ for four different digital circular arcs of length 20, 40, 60, and 80, taken from a digital circle of radius $r = 50$. It is clear from these plots that the error is abnormally large at or near the two endpoints, whereas in the intermediate portion, it is much less. Hence, such errors have to be kept in consideration for obtaining proper results. In our algorithm, we have considered this, resulting to satisfactory performance in terms of both precision and robustness.

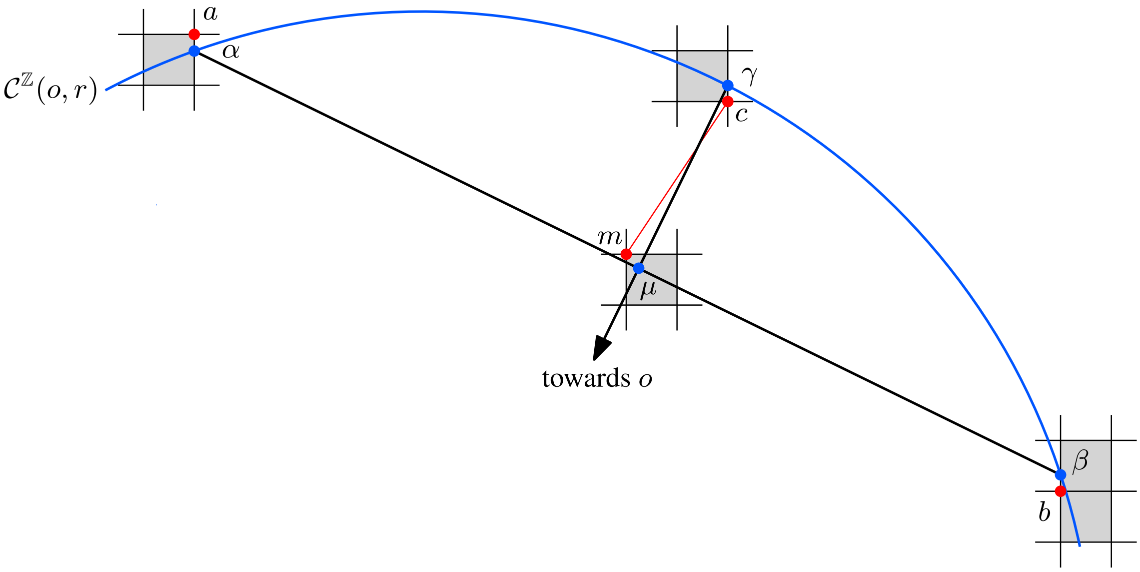


Figure 2.4: Deviation of the sagitta property. Points in $\mathbb{R}^2$ or on $\mathcal{C}^{\mathbb{R}}(o,r)$ are in blue, and those in $\mathbb{Z}^2$ or in $\mathcal{C}^{\mathbb{Z}}(o,r)$ are in red. The real sagitta here is $\mu\gamma$, and its discrete counterpart is mc.

2.2.2 Sagitta Property

We show here the deviation of sagitta property for digital circular arcs. See Figure 2.4 for an illustration. Let $\mathcal{C}^{\mathbb{R}}(o,r)$ be the real circle having center at $o(0,0)$ and radius $r \in \mathbb{Z}^+$. Let $\alpha(x_\alpha, y_\alpha)$ and $\beta(x_\beta, y_\beta)$ be two points on the boundary of $\mathcal{C}^{\mathbb{R}}(o,r)$ such that x_α and x_β are integers. Let $\mathcal{A}^{\mathbb{R}}(\alpha,\beta)$ be the arc of smaller length between the two arcs of $\mathcal{C}^{\mathbb{R}}(o,r)$ with endpoints α and β. Then the *sagitta* [Weisstein (b)] of the circular arc $\mathcal{A}^{\mathbb{R}}(\alpha,\beta)$ is the straight line segment drawn perpendicular to the chord $\alpha\beta$, which connects the midpoint $\mu(x_\mu, y_\mu)$ of $\alpha\beta$ with $\mathcal{A}^{\mathbb{R}}(\alpha,\beta)$, as shown in Figure 2.4. Let the sagitta intersect the arc $\mathcal{A}^{\mathbb{R}}(\alpha,\beta)$ at γ; then

$$\begin{aligned} r^2 &= (r - d(\mu,\gamma))^2 + d^2(\alpha,\mu) \\ \Rightarrow 2rd(\mu,\gamma) &= \left(\frac{d(\alpha,\beta)}{2}\right)^2 + d^2(\mu,\gamma) \\ \Rightarrow r &= \frac{d^2(\alpha,\beta)}{8d(\mu,\gamma)} + \frac{d(\mu,\gamma)}{2} \end{aligned} \tag{2.3}$$

The above equation is known as the *sagitta property* for real circle, where $d(\alpha,\beta)$ denotes the Euclidean distance between the points α and β, and $d(\mu,\gamma)$ is that between μ and γ. The sagitta property gives the radius of the arc, irrespective of its length in real domain. However, this is not true for a digital (circular) arc, as shown below.

Let $\mathcal{C}^{\mathbb{Z}}(o,r)$ be the digital circle corresponding to $\mathcal{C}^{\mathbb{R}}(o,r)$, and $\mathcal{A}^{\mathbb{Z}}(a,b)$ be the digital arc corresponding to $\mathcal{A}^{\mathbb{R}}(\alpha,\beta)$, where a, b, c, and m are the respective integer points corresponding to α,β,γ, and μ. The sagitta of the digital arc $\mathcal{A}^{\mathbb{Z}}(a,b)$ is $\tilde{s} = mc$ and hence using the sagitta property the estimated radius of the arc is $\tilde{r} = \frac{d^2(a,b)}{8d(m,c)} + \frac{d(m,c)}{2}$. Then, depending on the value of $d(a,b)$ and $d(m,c)$, the radius $\tilde{r}$ varies between $r-\delta$ and $r+\delta$, δ being the *absolute error* in radius estimation. In particular, we have the following theorem.

Theorem 2.2.3 *The relative error ε_r in radius estimation using sagitta property is given by*

$$\frac{|r-\tilde{r}|}{r} \leqslant \left|1 - \frac{\tilde{s}}{2r}\right|, \tag{2.4}$$

where $\tilde{r}$ is the estimated radius using sagitta property and $\tilde{s}$ is the length of the discrete sagitta.

Proof. Observe that $2\tilde{r} \geqslant d(m,c) = \tilde{s}$. So, relative error in radius estimation is given by $\varepsilon_r = \frac{|r-\tilde{r}|}{r} \leqslant \left|1 - \frac{\tilde{s}}{2r}\right|$. □

Theorem 2.2.3 implies that for a fixed value of r, as $\tilde{s}$ increases, the relative radius error decreases. But if $\tilde{s}$ is small relative to r, then the relative radius error would be large. Maximum possible value of $\tilde{s}$ is r, and then the relative radius error is 0. Figure 2.5 shows a set of four plots of relative radius error versus relative arc lengths for four different digital circular arcs of radius 20, 50, 100, and 200. The *relative arc length* is defined as the ratio of the length of the digital arc to its semi-perimeter, measured in number of pixels. Minimum relative error, maximum relative error, and average relative error are plotted with green, red, and blue colors respectively. It is clear from these plots that for a fixed radius, as the arc length increases, the corresponding sagitta $\tilde{s}$ increases in length, and so the relative error decreases, which we have proved in Theorem 2.2.3. This property is considered in our algorithm to obtain the desired result of circular arc detection.

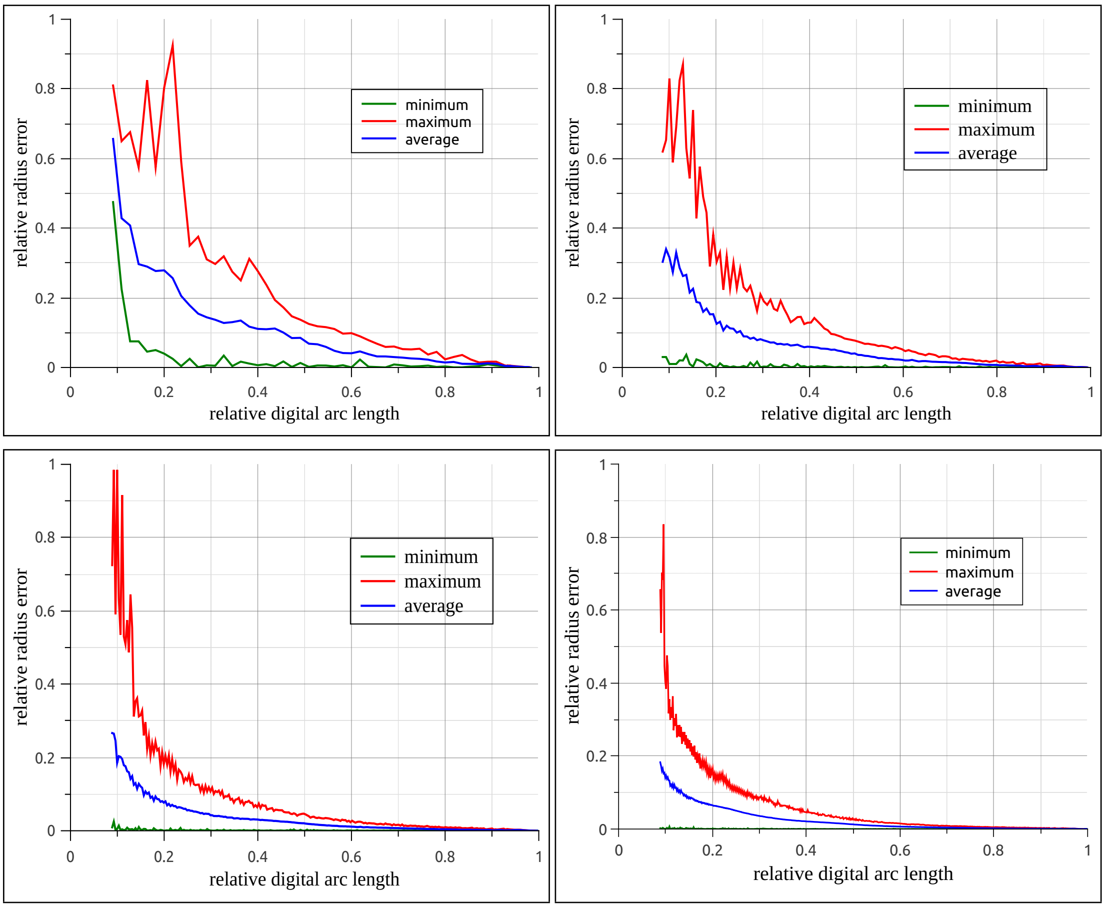


Figure 2.5: Relative radius error ε_r versus relative digital arc length for $r = 20, 50, 100, 200$, as estimated using sagitta property.

2.3 CSA: Proposed Algorithm by Chord and Sagitta Analysis

Let $\mathcal{I}$ be the input digital image containing various digital curves like straight line segments, circles, and circular arcs. Our algorithm checks each curve segment separately for its circularity. For this, it first extracts all the curve segments. As the image $\mathcal{I}$ may contain thick curves segments, we use thinning [Gonzalez and Woods (2001)] as preprocessing before applying the algorithm. The subsequent steps are as follows.

2.3.1 Removing the Straight Segments

The digital curve segments are first extracted from the thinned image and stored in a list of segments, $\mathcal{L}$. Each entry in $\mathcal{L}$ contains the coordinates of two endpoints defining the curve segment, and a pointer to the list of curve points. The center and the radius are also stored in it after their computation. To identify the circular arc segments, we first remove the digital straight line segments from the list $\mathcal{L}$. It may be mentioned that there are several techniques to determine digital straightness available in the literature [Bhowmick and Bhattacharya (2007), Klette and Rosenfeld (2004a,b), Rosin (1997)]. We have used the concept of area deviation [Wall and Danielsson (1984)], which is realizable in purely integer domain using a few primitive operations only. The method is as follows.

Let $S := \langle a = c_1, c_2, \ldots, c_k = b \rangle$ be a digital curve segment with endpoints a and b. Let c_i $(2 \leq i \leq k-1)$ be any point on the segment S other than a and b. Let h_i be the distance of the point c_i from the real straight line segment ab. Then S is considered to be a single digital straight line segment starting from a and ending at b, provided the following condition is satisfied.

$$\max_{2 \leqslant i \leqslant k-1} |\triangle(a, c_i, b)| \leq \tau_h d_\top(a, b) \tag{2.5}$$

Here, $|\triangle(a, c_i, b)|$ denotes twice the magnitude of area of the triangle with vertices $a := (x_1, y_1)$, $c_i := (x_i, y_i)$, and $b := (x_k, y_k)$, and $d_\top(a, b) := \max(|x_1 - x_k|, |y_1 - y_k|)$ is the maximum isothetic distance between the points a and b, and $\tau_h = 2$ in our experiments. Since all these points are in two-dimensional digital space, the above measures are computable in the integer domain as follows.

$$\triangle(a, c_i, b) = \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_i & x_k \\ y_1 & y_i & y_k \end{vmatrix} \tag{2.6}$$

As $\triangle(a, c_i, b)$ gives twice the signed area of the triangle with vertices a, c_i, and b, the digital curve segment S forms a single straight line segment, provided the (maximum) area of the triangle having ab as the base and the third vertex as the point of S farthest from ab, does not exceed the area of the triangle with base-length $d_\top(a, b)$ (isothetic length) and height τ_h.

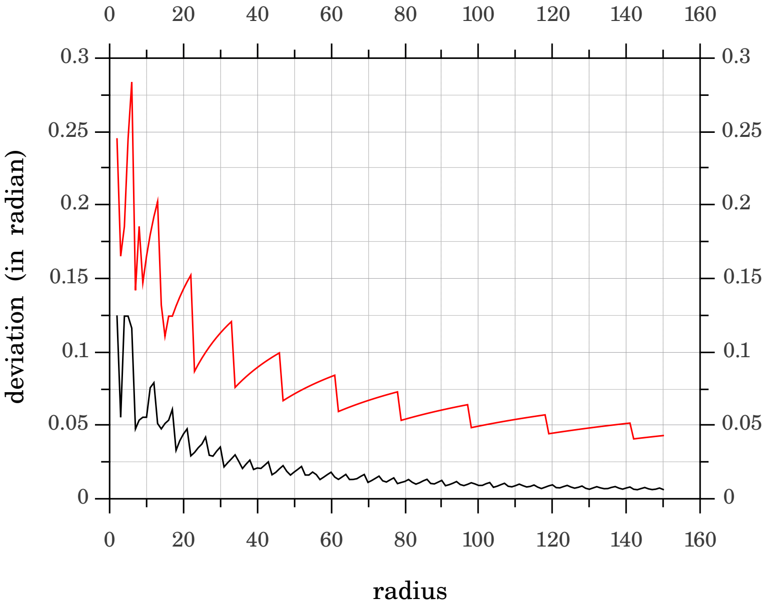


Figure 2.6: Plot of maximum deviation of circumferential angle (in radian) of a digital circle from its real counterpart. The black curve corresponds to the central region, and red to the remaining part of the semi-circle excepting its two endpoints.

2.3.2 Verifying the Circularity

As explained in Sec. 2.3.1, after removal of the digitally straight segments, the remaining segments present in the list $\mathcal{L}$ are not digitally straight. That is, each segment S in $\mathcal{L}$ is made of one or more circular segments with or without one or more intervening straight parts. So, for each segment S, we check its circularity using the chord property in $\mathbb{Z}^2$, as explained in Sec. 2.2.1. If the segment S consists of both circular and straight components, then we extract its circular part(s) only from S and discard its straight portion, store these circular segment(s) in the list $\mathcal{L}$ with necessary updates, and remove the original segment S from $\mathcal{L}$. As explained in Sec. 2.2.1, circumferential angular deviation is too high near the endpoints of an arc. So, we exclude some pixels from both ends for chord property checking. We define the *central region* of an arc as the sequence of pixels lying in its central one-third portion. We verify the circularity for the central region of an arc. The remaining points (i.e., one-third from either end) of S are disregarded from circularity test as they are prone to high deviation of the chord property. The count of pixels in a semicircle of radius 1 is 3, and hence it is trivially accepted. However, such occurrences are not found as valid circles or circular arcs in a digital drawing. The count of pixels in a semicircle of radius 2 is 7, and hence the chord property is checked for arcs having length $\tau_c = 7$ or more. After deleting the arcs of length less than τ_c from the list $\mathcal{L}$, the chord property is checked for each of the remaining arcs. Let $S := \langle a = c_1, c_2, \ldots, c_k = b \rangle$ be an arc in the list $\mathcal{L}$. We verify the circularity for the central region of S, namely $S' := \left\langle c_{\lfloor k/3 \rfloor}, c_{\lfloor k/3 \rfloor + 1}, \ldots, c_{m-1}, c_m, c_{m+1}, \ldots, c_{\lfloor 2k/3 \rfloor - 1}, c_{\lfloor 2k/3 \rfloor} \right\rangle$. Hence, if c_m ($m = \lfloor k/2 \rfloor$) is the midpoint of S and the angle subtended by the chord ab at m is estimated to be ϕ_m, then S is considered to be satisfying the chord property in $\mathbb{Z}^2$, provided the angle ϕ_c subtended by ab at each point $c \in S'$ satisfies the following equation.

$$\max_{c \in S'} \{ |\phi_c - \phi_m| \} \leq \delta_\phi \tag{2.7}$$

As shown in Figure 2.6, the deviation of a circumferential angle in the central region is less than $\pi/18 \approx 0.1745$ radian. Hence in our experiments, we have taken $\delta_\phi = \pi/18$ radian. If S is not found to be circular, then we divide S into two equal parts and recursively check for digital straightness and the chord property on each part. The process is continued until a part is smaller than τ_c in length or it satisfies digital straightness or satisfies chord property.

2.3.3 Parameter Estimation

The centers and radii of the detected circular arcs are computed using the sagitta property of the circle, as explained in Sec. 2.2.2. While combining the circular arcs, necessary care has to be taken for the inevitable error that creeps in owing to the usage of sagitta property, which is a property of real circles only. Since we deal with digital curve segments, the *cumulative error* of the effective radius computed for a combined/growing circular arc using the aforesaid sagitta property is very likely to increase with an increase in the number of segments constituting that arc. Hence, to enhance accuracy, we merge two digital circular segments S and S' into $S'' := S \cup S'$, if (i) S and S' have a common endpoint in $\mathcal{L}$ and (ii) S'' satisfies the *chord property*. Since the node corresponding to each segment in the list $\mathcal{L}$ contains endpoints, center, radius, and a pointer to the list of curve points, the attributes of the segment S are updated by those of S'', and the data structure $\mathcal{L}$ is updated accordingly.

2.3.4 Parameter Finalization

In spite of the treatments to reduce discretization errors while employing chord property to detect circular arcs and while employing sagitta property to combine two or more circular arcs and to compute the effective radius and center, some error may still be present in the estimated values of the radii and the center. To remove such errors, we apply a *restricted Hough transform* (rHT) on each circular arc $S \in \mathcal{L}$ with a small parameter space [Chen and Chung (2001)]. Let $q(x_q, y_q)$ and r be the respective center and radius of S estimated using the sagitta property. As explained in Sec. 2.2.2, the relative radius error can have a maximum value of 1. Hence, the restricted parameter space is considered as $[x_q - \delta, x_q + \delta] \times [y_q - \delta, y_q + \delta] \times [\tilde{r} - \delta, \tilde{r} + \delta]$, where $\delta = \tilde{r}$. A 3D integer array, H, is taken corresponding to this parameter space, each of whose entry is initialized to zero. For every three points c, c', and c'' from S, with c lying in its left region, c' lying in its central region, and c'' lying in its right region, we estimate the center $q'(x', y')$ and the radius r' of the (real) circle passing through c, c', and c''. The corresponding entry in H is incremented accordingly. Finally, the entry in H corresponding to the maximum frequency provides the final center and radius of S.

2.3.5 Demonstration of CSA

A demonstration of the proposed algorithm (CSA) on a sample image is shown in Figure 2.7. All the digital curve segments in the image are extracted and stored in the list $\mathcal{L}$ (Figure 2.7(b)). The straight line segments are removed from $\mathcal{L}$ using the straightness properties. For example, fo and oq are two of the straight line segments that are removed (Figure 2.7(c)). Then using the *chord property*, the circular segments are detected with necessary updates in the list $\mathcal{L}$. For example, the digital curve segment db consists of two circular segments. After their extraction, the segment dc is stored in the node of the original segment and the other one, i.e., cb, is stored in a newly created node in the list $\mathcal{L}$. Similarly, for the segment mr, the circular arc ml is extracted, inserted in the node of mr, and the straight part lr is removed.

Two or more adjacent arcs are combined if they jointly satisfy the *chord property* in order to get larger arcs for reducing the computational error in the next step while applying the *sagitta property* (Sec. 2.3.3). After this combining/merging, the number of circular segments gets significantly reduced, as reflected in Figure 2.7(d). The radius and the center of each arc in $\mathcal{L}$ are computed using the *sagitta property* and stored in the node of the corresponding arc (Figure 2.7(e)). Next, we apply rHT on these arcs (Sec. 2.3.4). The resultant image is shown in Figure 2.7(f). Finally, we consider the detected circular arcs, and for each pixel on a detected arc, the object pixels in its 8-neighborhood are iteratively marked as pixels of the corresponding circular arc. Figure 2.7(g) shows the final circular arcs detected by our algorithm, and the corresponding ground-truth is shown in Figure 2.7(h).

2.4 Experimental Results

We have implemented our algorithm CSA, and also some existing algorithms (Sec. 2.4.1), in C on the openSUSE™ OS Release 11.0 HP xw4600 Workstation with Intel® Core™2 Duo, 3 GHz processor. We have performed tests on several datasets including the GREC datasets [GREC (2007, 2013)]. The results of our algorithm on some of the images from these datasets are shown here.

Figure 2.8 shows the step-by-step output of our algorithm on the image `g07-tr1.tif` from GREC2007 dataset. The input image contains closely placed concentric circles and circular arcs, which have been accurately detected by our algorithm. Another set of results is shown in Figure 2.9. Its input image is `g07-tr4.tif` from GREC2007 dataset, which is a comparatively simpler image containing only full circles, for which our algorithm gives perfect output. Even for noisy or unclear input images, our algorithm detects the circles and the circular arcs, as evident from the results on two images shown in Figure 2.10. Figure 2.11 shows the result on another pair of images from GREC2007 dataset. Although the images contain annotation text, the arcs are correctly detected by our algorithm. Some more results on GREC2013 datasets and our own dataset (SMP) are given in Appendix 1. We have prepared the dataset SMP by scanning engineering drawing books, e.g., [Simmons *et al.* (2010)].

We have tested our algorithm to evaluate its robustness against rotation. Results on `g07-tr6.tif` for different angles of rotation are shown in Figure 2.12. These results

Table 2.1: Results for images from GREC2007 dataset for the proposed algorithm.

Image `g-07-`	#rows× #columns	N_c	N_g	N_p	N_{fa}	N_{fn}	E1	E2	AD	Time (sec.)
2	792 × 662	29072	7820	7243	72	649	0.921	8.299	0.975	0.121
3	924 × 1167	53899	6663	6045	282	900	4.232	13.507	0.978	0.180
4	638 × 2046	53156	16245	16398	155	2	0.954	0.012	0.997	0.217
5	590 × 977	8478	4326	4355	35	6	0.809	0.139	0.995	0.125
6	562 × 905	8321	2639	2884	247	2	9.360	0.076	0.970	0.120
7	779 × 907	14817	11435	11494	69	10	0.603	0.088	0.995	0.145
8	982 × 1064	25358	12026	12845	821	2	6.827	0.017	0.968	0.166
9	700 × 1400	44717	18650	18849	347	148	1.861	0.794	0.989	0.205
10	862 × 853	15189	9930	9923	53	60	0.534	0.604	0.993	0.152
11	1043 × 900	19182	12493	12369	14	138	0.112	1.104	0.992	0.231

indicate that the proposed algorithm has a reasonable robustness against rotation. In order to evaluate its performance on noisy images, we have also run our algorithm on some images after adding salt-and-pepper noise. The results on the image `g07-tr6.tif` for different noise levels are shown in Figure 2.13. From these results, we can notice that the algorithm gives an acceptable result up to 5% noise.

For a quantitative evaluation, we have analyzed its performance using some conventional empirical measures, namely *Type I error* (E1), *Type II error* (E2), and *accuracy of detection* (AD). These parameters are evaluated in terms of the following variables:

- N_c = the number of curve pixels in the original image.
- N_g = the number of pixels on circular arcs in the ground-truth image.
- N_p = the number of pixels on circular arcs detected by the proposed algorithm.
- N_{fa} = the number of false-acceptance pixels.
- N_{fr} = the number of false-rejection pixels.

Note that both N_{fa} and N_{fr} represent erroneous output of the proposed algorithm. The value of N_{fa} gives the number of pixels detected as part of circular arcs by the proposed algorithm; these pixels do not correspond to circular arcs in the ground-truth image. On the other hand, the value of N_{fr} gives the number of pixels that are incorrectly detected as non-circular arc pixels by the proposed algorithm when they actually lie on circular arcs in the ground-truth image. The error estimates are defined using these variables as follows.

- *Type I error*: $\mathrm{E1} = \frac{N_{fa}}{N_g} \times 100\%$.
- *Type II error*: $\mathrm{E2} = \frac{N_{fr}}{N_g} \times 100\%$.
- *Accuracy of detection*:

$$\begin{aligned}\mathrm{AD} &= \frac{\#\text{ pixels correctly detected on circular and non-circular arcs}}{\#\text{ curve pixels in the original image}}\\ &= \frac{N_c - (N_{fa} + N_{fr})}{N_c}.\end{aligned}$$

A comparative study of some of the images from the GREC2007 dataset is given in Table 2.1.

2.4.1 Comparison with Existing Methods

In this section, we compare our work with two existing methods, namely, *randomized Hough transform* (RHT) Xu and Oja (1993) and the HT-based circular arc detection using *effective voting method* (EVM) [Chiu and Liaw (2005)]. Both these methods are basically improvisations on HT, which is by far the most popular arc detection technique. We first provide short descriptions of these two methods, and then compare them with the proposed method (CSA).

Comparison with RHT [Xu and Oja (1993)] The improvement of HT in terms of computational time is effected in this method by randomly selecting three points from the image I at each step and mapping them into a point in the parameter space. The parameter space is implicitly represented by a set P, each of whose elements contains a real-valued vector (circle parameter values) and an integer score. At each step, the parameter space is updated by checking whether the mapped point already exists in P; if so, then its score is incremented by one; and if not, then a new element is added to P, with its score set to one. In the process of updating, if an element obtains a score equal to the threshold n_t (= 2 or 3), then it is considered as a candidate for a true circle. If

Table 2.2: Results of running EVM for different values of T_e on the image `g07-tr7.tif`. The best results are obtained for $T_e = 0.32$–0.43 when ten circles (circular arcs) are detected out of the fourteen circles/circular arcs present in the input image (Figure 2.16(b)).

T_e	#circles & arcs detected	#correct circles & arcs detected / 14	N_p	Error				AD	Time (sec.)
				N_{fa}	N_{fr}	E1 (%)	E2 (%)		
0.20	77	12	12368	1541	608	13.476	5.317	0.855	54.272
0.25	21	10	9787	399	2047	3.489	17.901	0.835	54.272
0.30	11	10	9478	138	2095	1.207	18.321	0.849	54.272
0.32–0.43	10	10	9467	127	2095	1.111	18.321	0.850	54.272
0.45	8	8	7981	106	3560	0.927	31.132	0.753	54.272
0.50–0.85	6	6	6457	79	5057	0.691	44.224	0.653	54.272
0.90	1	1	588	10	10857	0.088	94.945	0.267	54.272
0.95	0	0	0	0	11435	0.000	100.000	0.228	54.272

the number of points of I lying on this circle is greater than a predefined threshold T_r, then it is reported and all the concerned points are removed from I. The above process is repeated until the stopping criterion is satisfied.

Comparison with EVM [Chiu and Liaw (2005)] In this method, point triplets are selectively chosen by considering the point pairs from the sampled object points, M. While obtaining a triplet, all point pairs in M are considered; and for each pair $(p, q) \in M^2$, the triplet (p, q, r) is chosen such that r is an object point in the image I and $pq = qr$. For each selected triplet, the entire image I is searched for all the object points lying on the circle C represented by it, and the *existing rate* of C is determined. The existing rate of C is the ratio of the number of object points lying on C to the circumference of C. Based on the existing rate values, each object point in I votes for only that circle which has the highest existing rate among all the circles passing through it. Finally, all the circles that have existing rate values higher than the given threshold value T_e are reported.

Thus, we observe that the above methods require a predefined threshold value (T_r for RHT and T_e for EVM) for detecting the true circles in the image. If only complete circles are present in the image, then high threshold values suffice in both the cases. However, for the images, which also contain (partial) circular arcs, the thresholds have to be lowered sufficiently to get proper result. So, for best results, the optimum threshold value for each image should be set individually depending on the nature of the circular arcs present in the image. But obtaining an optimal threshold value is quite difficult, and hence this issue

Table 2.3: Results of different (randomized) runs of RHT for $T_r = 0.46$ on the image `g07-tr7.tif`.

T_r	Run #	#circles & arcs detected	#correct circles & arcs detected / 14	N_p	Error				AD	Time (sec.)
					N_{fa}	N_{fr}	E1 (%)	E2 (%)		
0.46	1	7	7	7021	97	4511	0.848	39.449	0.689	7.749
	2	7	7	7089	93	4439	0.813	38.819	0.694	9.709
	3	8	8	7875	98	3658	0.857	31.989	0.746	6.471
	4	7	7	7251	95	4279	0.831	37.420	0.705	1.273
	5	5	5	5995	91	5531	0.796	48.369	0.621	1.637
	6	10	10	9084	121	2472	1.058	21.618	0.825	1.638
	7	7	7	7267	100	4268	0.874	37.324	0.705	6.223
	8	6	6	5763	84	5756	0.735	50.337	0.606	6.992
	9	6	6	5876	85	5644	0.743	49.357	0.613	7.463
	10	10	10	9072	120	2483	1.049	21.714	0.824	11.123

becomes a major drawback of both EVM and RHT methods.

Our method, on the contrary, suffers from no threshold-related weakness, as it requires no threshold value and outputs an optimal or near-optimal result. This is evident from Figs. 2.14, 2.15, and 2.16, showing the output results for RHT, EVM, and our algorithm (CSA) respectively, for the image `g07-tr7.tif` that actually contains 6 circles and 8 semicircular arcs, as shown in its ground-truth image (Figure 2.16(b)). Hence, for each individual image, depending on the extent of noise and the distortion of circular arcs and circles, the threshold value has to be properly set for RHT and EVM. From Figure 2.15 and Table 2.2, we can see that the best possible result for the image `g07-tr7.tif` is obtained when T_e lies in the range $[0.32, 0.43]$; but even in this range, the result is not perfect, as four circular arcs still remain undetected. In case of RHT, the situation is even worse, as each run of the algorithm under the same condition does not always produce the same result (due to randomization) when the value of T_r is low. A summary of the statistical details for ten different (randomized) runs of the algorithm with $T_r = 0.46$ is given in Table 2.3, and a sample set of output images for five of these runs is given in Figure 2.14.

For a better understanding, three plots are given in Figure 2.17, Figure 2.18, and Figure 2.19, comparing the values of E2, AD, and CPU time for the three algorithms. In the plots of Figure 2.17 and Figure 2.18, the x-axis represents the image number in GREC2007 dataset. Against the image number, E2 is plotted using sticks of black, red, and blue (Figure 2.17), corresponding to RHT, EVM, and CSA, respectively. In Figure 2.18, the AD value is plotted for the three algorithms with the same color code as in Figure 2.17. A high value of E2 signifies that not all the circles and circular arcs have been detected, or even if they are detected, their centers and radii have not been estimated accurately. From these plots, it may be noticed that RHT and EVM algorithms are not efficient in detecting circular arcs. For our algorithm, in most of the cases, the E2 values are significantly lower (less than 9, except for the image `g07-tr3.tif`) compared to those for RHT and EVM. For statistical details, see the adjoining tables given with the respective plots. In the plots of Figure 2.18, barring a couple of cases, the AD values corresponding to our algorithm are either at par or better than those corresponding to RHT. Although the EVM algorithm is occasionally better than our algorithm in terms of AD, it has a poorer performance compared to our algorithm in terms of CPU time. This is clearly evident from the plots in Figure 2.19 where the CPU time (in seconds) for each algorithm is plotted against the number of object pixels (N) processed. The reason for such a better performance of our algorithm in terms of CPU time can be attributed to the fact that it works only on arc pixels and not on all pixels of the input image.

2.4.2 Result on Natural Images

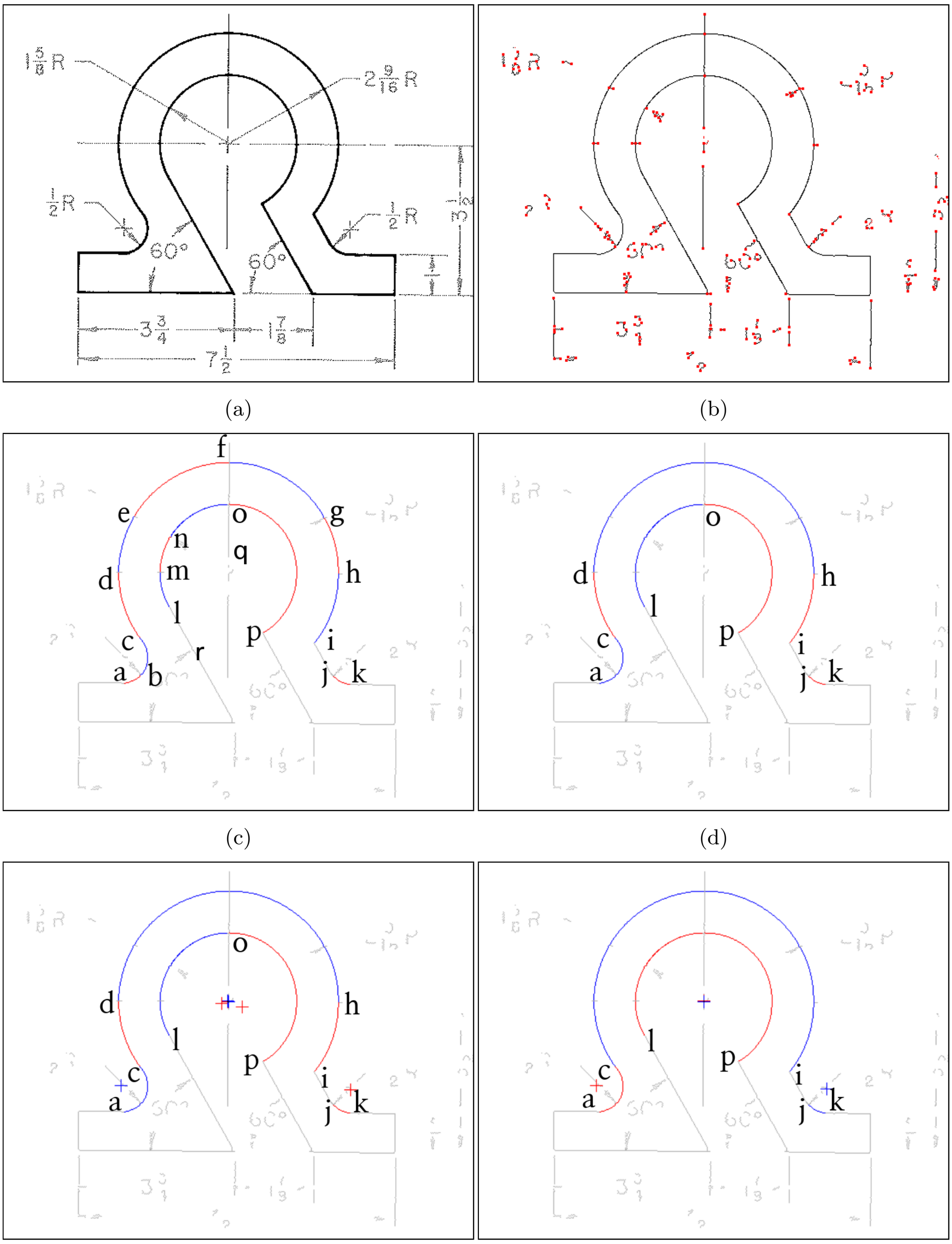
We have also tested our algorithm on natural images. For these images, we first extract the edge map using Canny edge detection algorithm [Gonzalez and Woods (2001)], and then run our algorithm on the edge map. A result obtained for the image `sunset` is shown in Figure 2.20. Notice that the algorithm correctly detects the semicircular arc that represents the boundary of the sun. Another set of results for two more natural images, namely `igloo` and `coins`, and a set on some real-world objects are shown in Figure 2.21 and Figure 2.22, respectively.

2.5 Conclusion

We have shown how the *chord-and-sagitta property*, with their appropriate adaptations on the digital plane, can be used for identifying digital circles and circular arcs in a binary image. A restricted Hough transform is finally used to further improve the accuracy of centers and radii. We have tested our algorithm on various datasets and the related results demonstrate the efficiency and robustness of the technique and the strength of the proposed approach in detecting circles and arcs on the digital plane.

Since for each circle or circular arc, we have computed the center and radius before applying the Hough transform, the size of the Hough space is very small. Hence, the requirement of accumulator memory and the computation time are reduced significantly. This is indeed reflected in the runtime comparison between the proposed algorithm and other existing algorithms.

The work discussed in this chapter is focused mainly on the detection of (digitally) circular arcs after removal of (digitally) straight pieces from a given binary image. It has a broader perspective in the sense that when we have to decompose a given digital curve into a minimum number of line segments and circular arcs, taken in totality, then the problem is quite challenging. Although there has been some work in recent time related to this problem, e.g., [Kolesnikov (2012), Kolesnikov and Kauranne (2014), Nguyen and Debled-Rennesson (2011)], none of these, however, has claimed or proved the minimality of the output. It may be mentioned in this context that decomposing a digital curve into a sequence of minimum number of straight segments can be done in linear time [Feschet and Tougne (2005)], although a similar problem of decomposing a 3D digital surface into a minimum number of digital plane segments is known to be NP-hard [Sivignon and Coeurjolly (2009)].

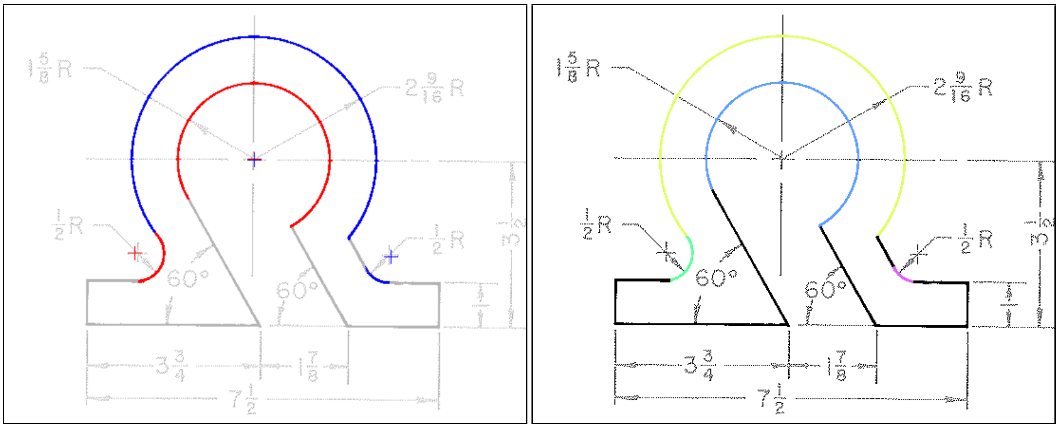


(a) (b)

(c) (d)

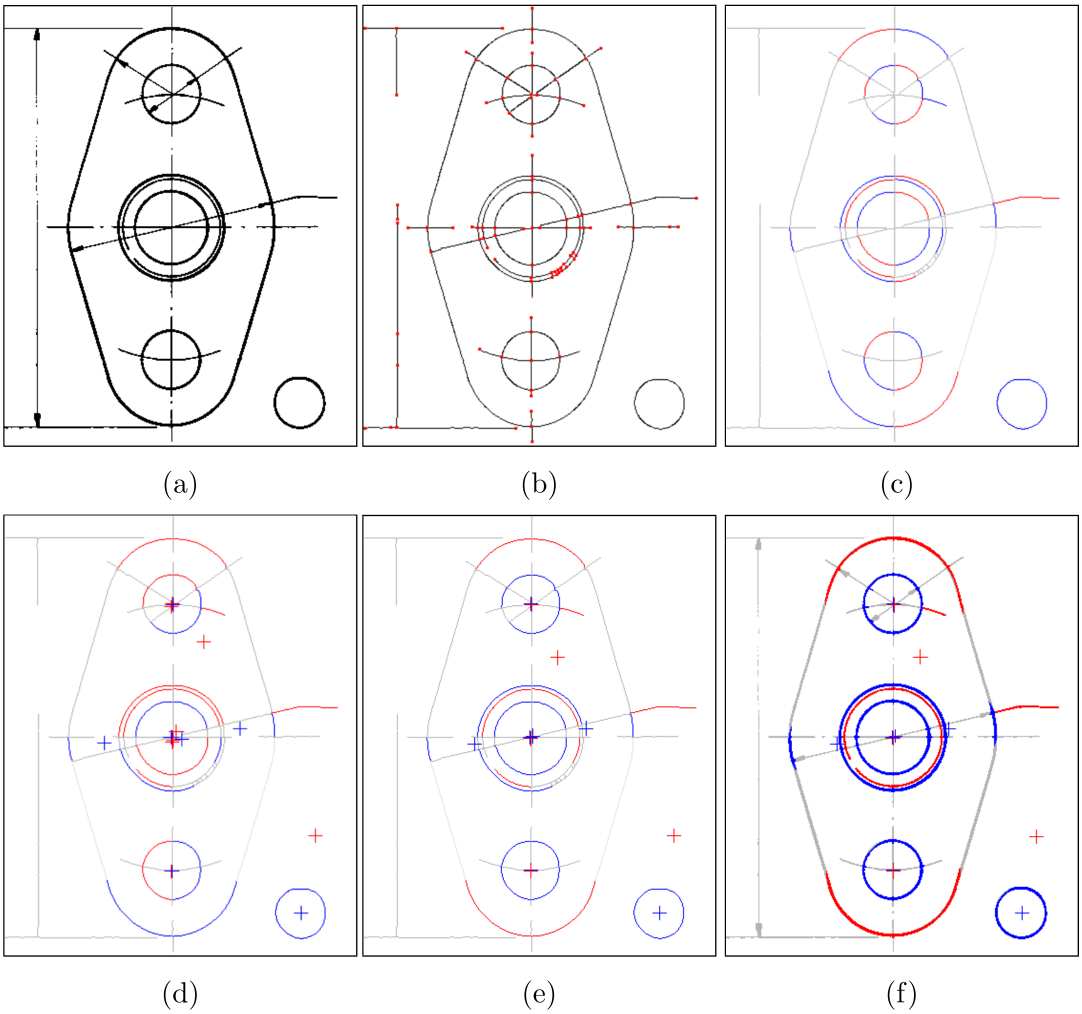
(e) (f)

(Continued to next page.)



(g) (h)

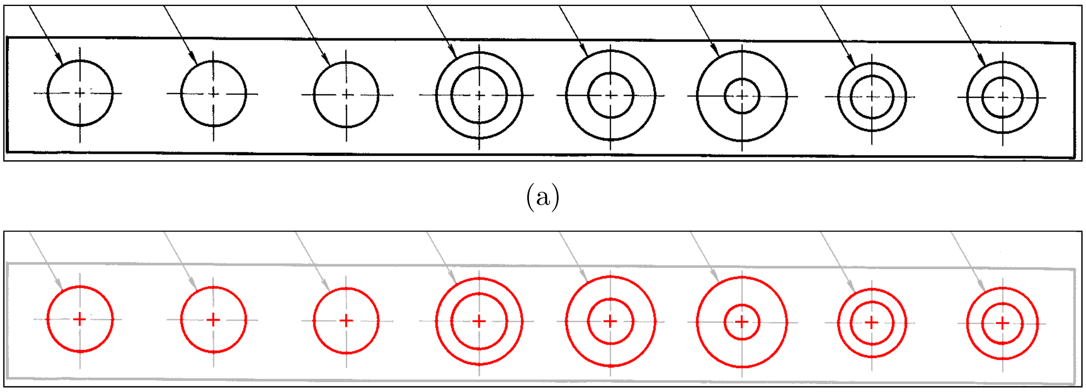
Figure 2.7: Step-wise snapshots of the algorithm on `g07-tr6.tif` from GREC2007 dataset [GREC (2007)]: (a) input image; (b) segments after thinning; (c) circular arcs by *chord property*; (d) after combining adjacent arcs; (e) centers by *sagitta property*; (f) after applying *restricted Hough transform*; (g) final result; (h) ground-truth;



(a) (b) (c)

(d) (e) (f)

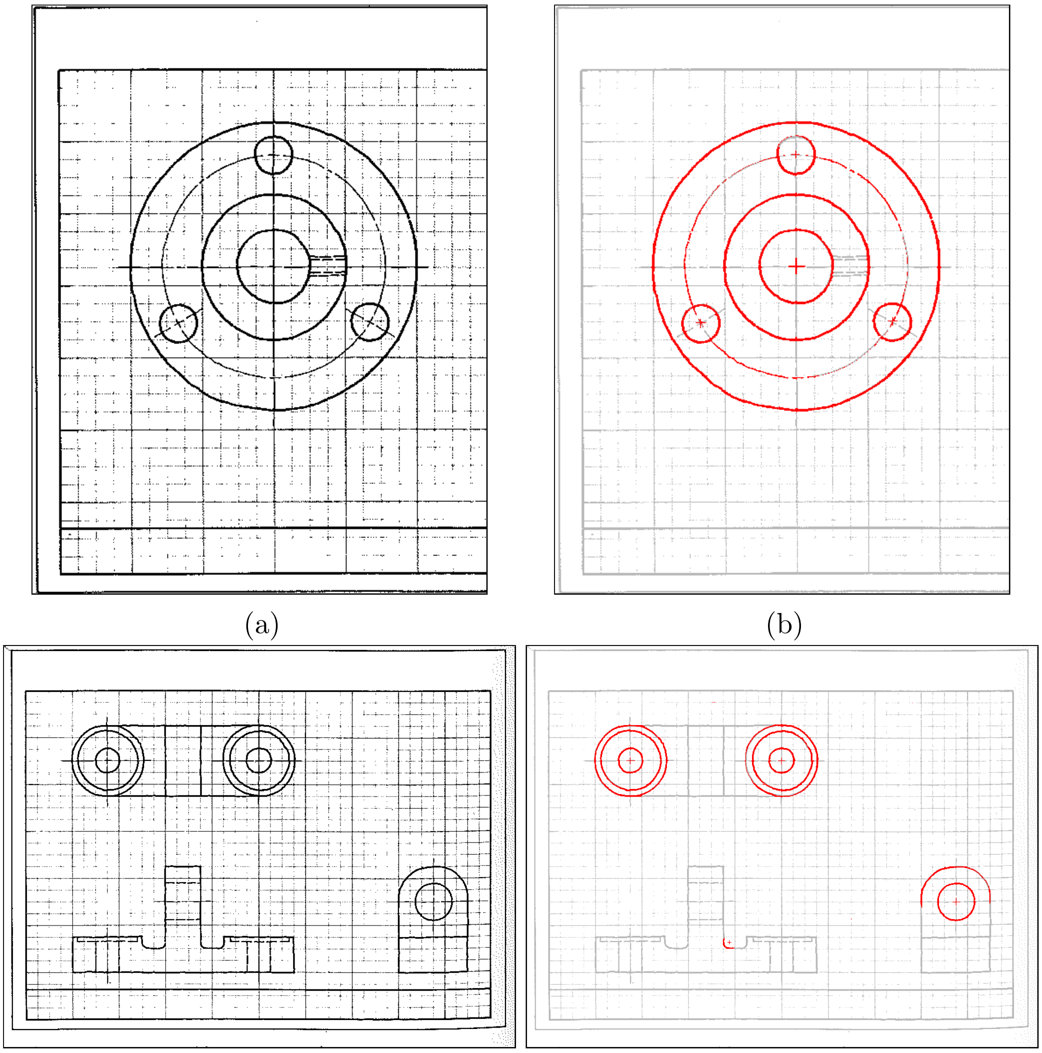
Figure 2.8: Step-wise snapshots of our experiment on `g07-tr1.tif` from GREC2007 dataset [GREC (2007)]: (a) input image; (b) segments after thinning; (c) circular arcs by *chord property* and after combining adjacent arcs; (d) centers by *sagitta property*; (e) after applying *restricted Hough transform*; (f) final result.



(a)

(b)

Figure 2.9: An example of perfect result by our algorithm on the image cropped from `g07-tr4.tif` of GREC2007 dataset, which contains only full circles.



(a) (b)

(c) (d)

Figure 2.10: Results by our algorithm on two difficult images from GREC2007 dataset. **Top:** `g07-tr2.tif`. **Bottom:** `g07-tr3.tif`. (a) and (c) are the respective inputs, and (b) and (d) are the respective outputs.

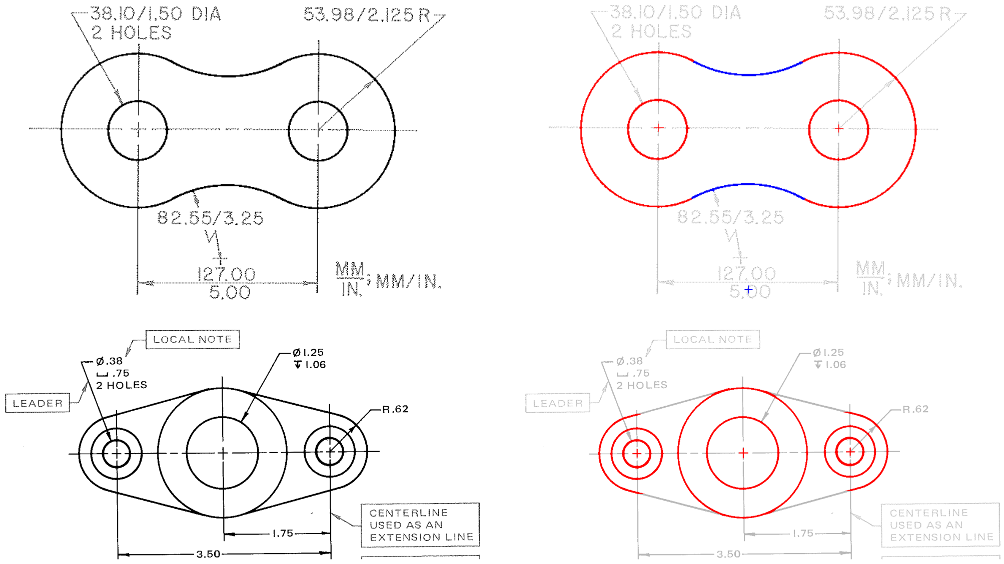


Figure 2.11: Results by our algorithm on two images from GREC2007 dataset. **Top:** `g07-tr5.tif`. **Bottom:** `g07-tr9.tif`.

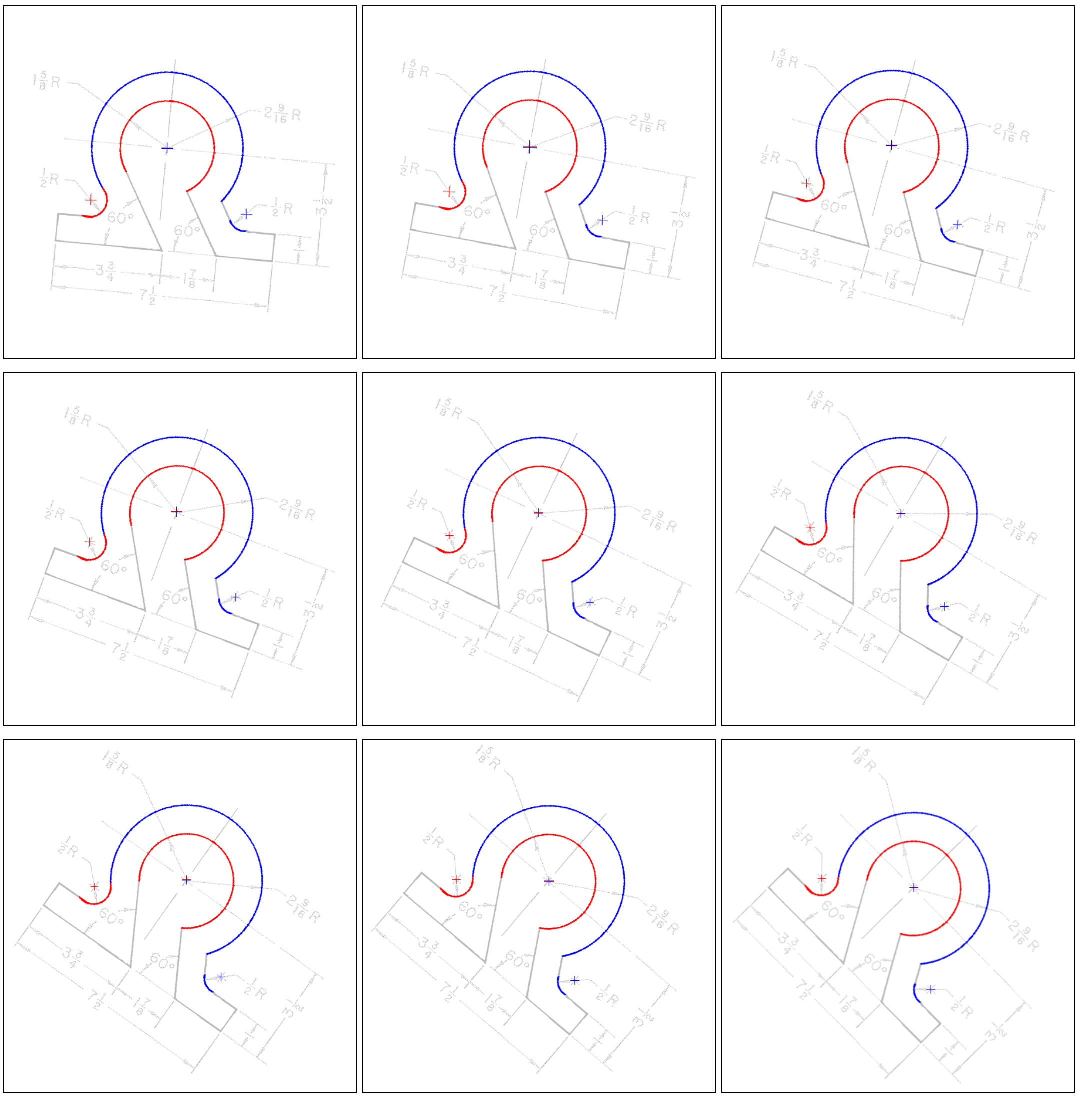


Figure 2.12: Results showing robustness of our algorithm against rotation. The output images are for `g07-tr6.tif`, rotated clockwise by 5^o, 10^o, $15^o, \ldots, 45^o$, shown here in row-major order.

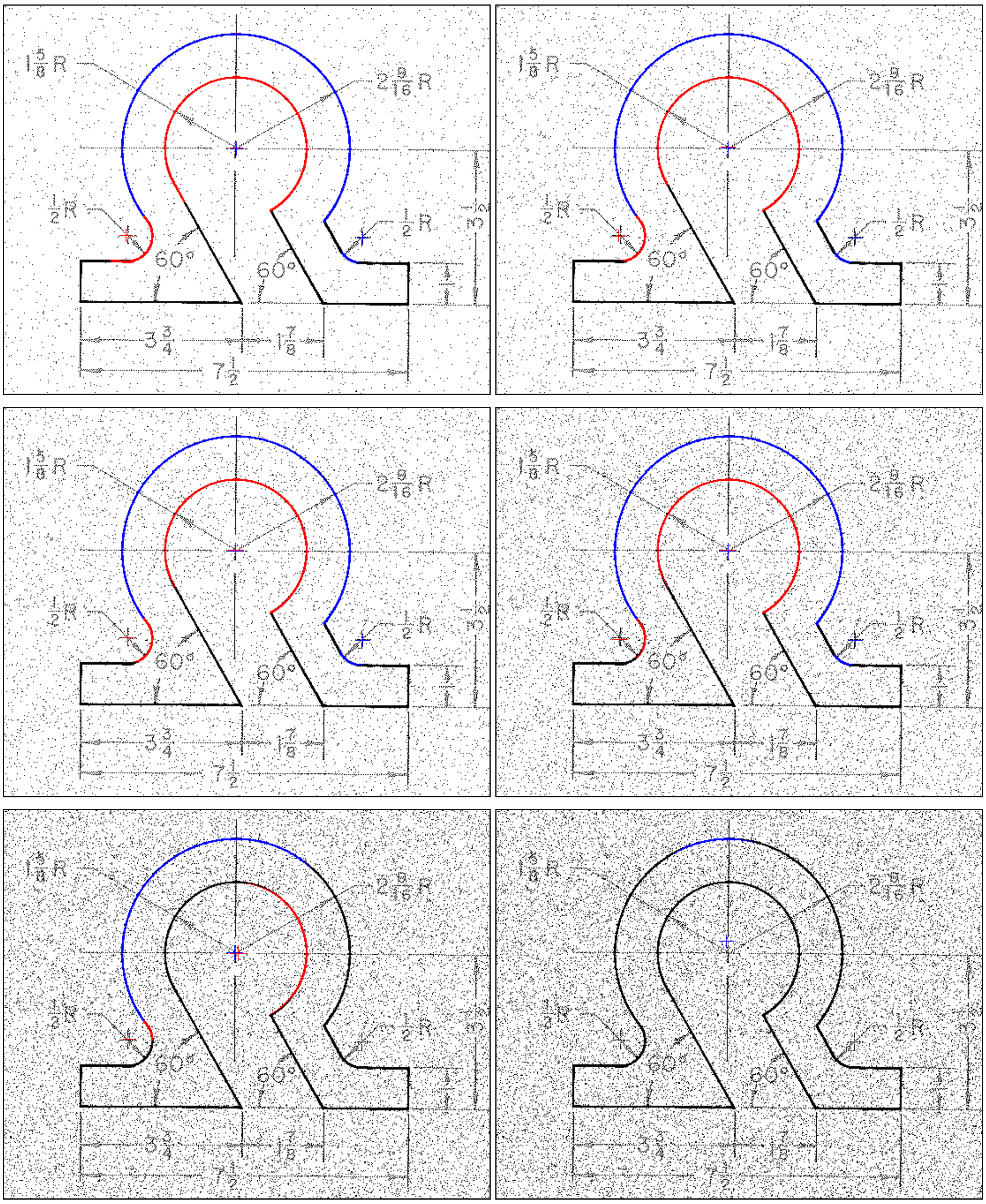
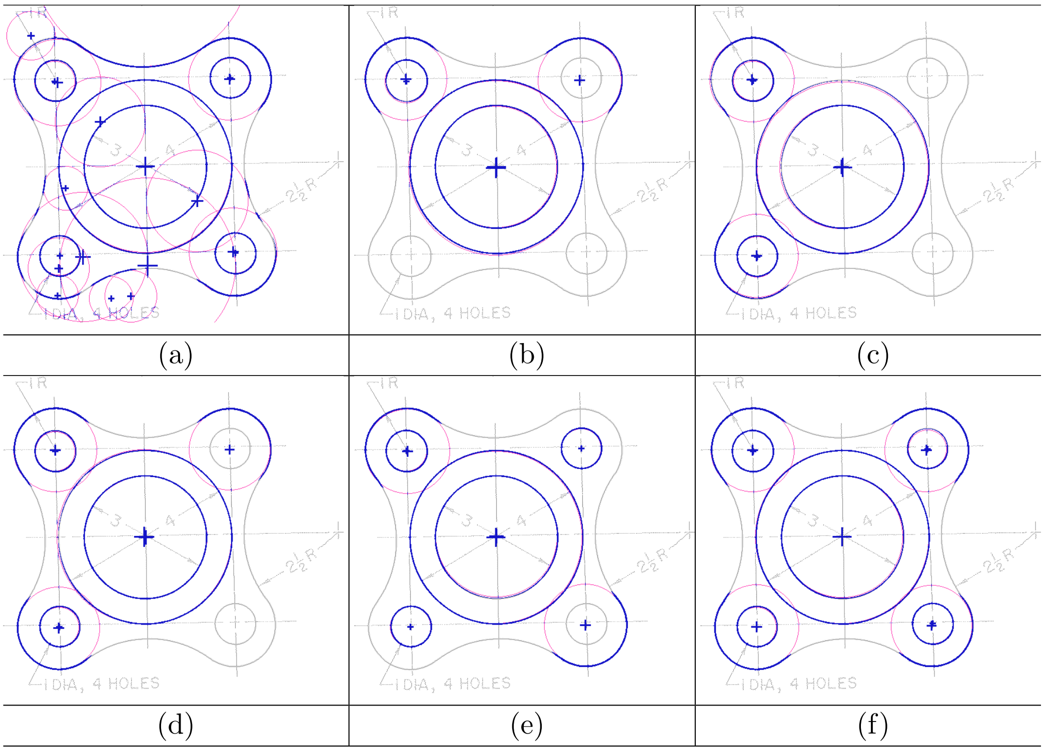


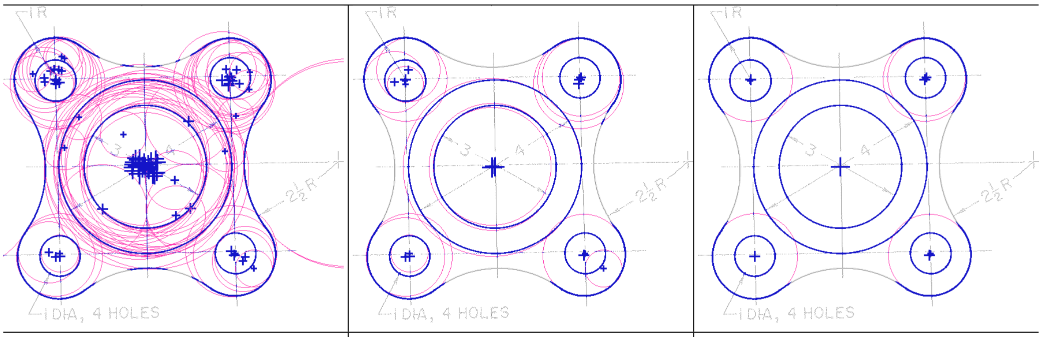
Figure 2.13: Results of our algorithm after adding salt-and-pepper noise in the image `g07-tr6.tif` to different levels (1%, 2%, 3%, 5%, 8%, 10%, shown in row-major order).



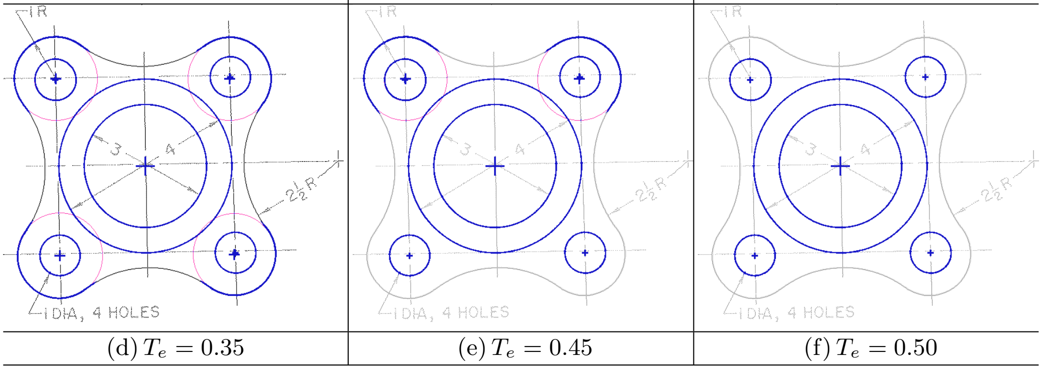
(a) (b) (c)

(d) (e) (f)

Figure 2.14: Results of different (randomized) runs of RHT for (a) $T_r = 0.23$ and (b–f) $T_r = 0.46$ on the image `g07-tr7.tif`. All the different runs for $T_r = 0.46$ produce different outputs due to randomization. See Table 2.3 for statistical details.

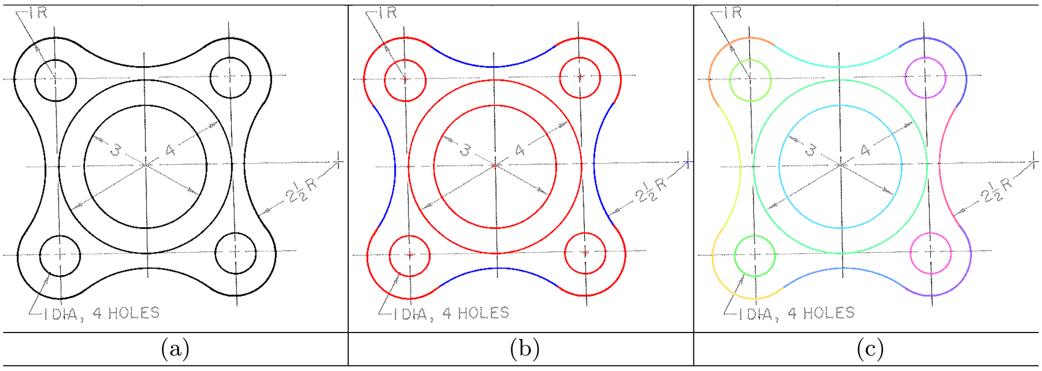


(a) $T_e = 0.20$ (b) $T_e = 0.25$ (c) $T_e = 0.30$



(d) $T_e = 0.35$ (e) $T_e = 0.45$ (f) $T_e = 0.50$

Figure 2.15: Results of running the algorithm EVM for different values of T_e on the image `g07-tr7.tif`. Notice that for low values of T_e, many extraneous arcs are reported along with the correct arcs; and for higher values of T_e, the output image cannot detect all the circular arcs accurately. See Table 2.2 for statistical details.



(a) (b) (c)

Figure 2.16: Result by our algorithm on (a) `g07-tr7.tif` from GREC2007 dataset. Unlike EVM and RHT, our algorithm does not require any parameter tuning, and (b) its result is close to (c) the benchmark result.

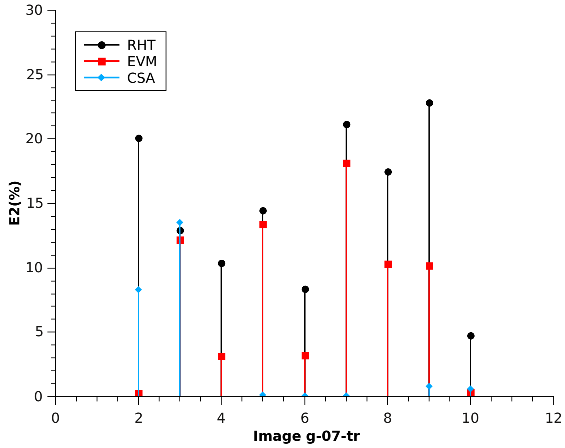


Image g-07 -tr	Error					
	E1 (%)			E2 (%)		
	EVM	RHT	CSA	EVM	RHT	CSA
2	3.171	3.453	0.921	0.294	20.115	8.299
3	3.542	7.969	4.232	12.202	12.937	13.507
4	0.721	0.936	0.954	3.158	10.372	0.012
5	1.179	1.271	0.809	13.385	14.494	0.139
6	7.579	4.775	9.360	3.221	8.375	0.076
7	0.848	0.892	0.603	18.120	21.189	0.874
8	3.318	4.798	6.827	10.294	17.446	0.017
9	2.461	12.000	1.861	10.182	22.804	0.794
10	2.236	2.064	0.534	0.303	4.743	0.604
11	0.240	0.472	0.112	21.556	24.918	1.105

Figure 2.17: Plots of E2 for RHT, EVM, and CSA (proposed algorithm) for images from GREC2007 dataset.

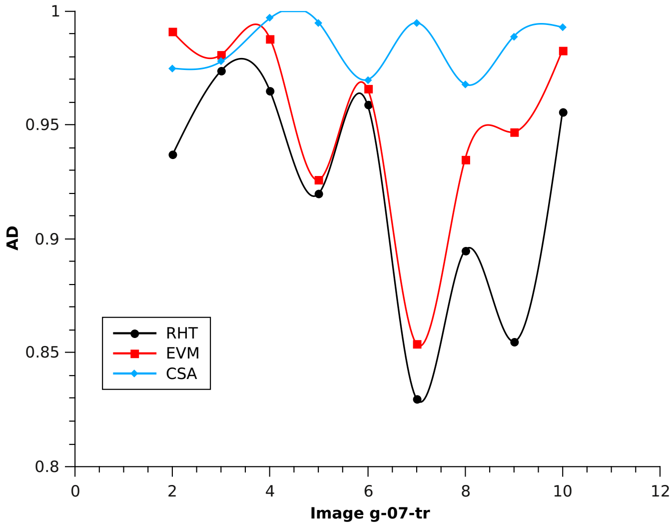
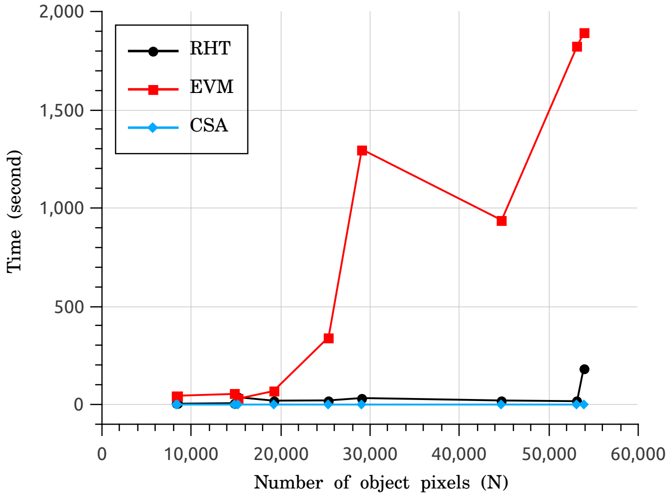


Image go7-tr	AD		
	EVM	RHT	CSA
2	0.991	0.937	0.975
3	0.981	0.974	0.978
4	0.988	0.965	0.997
5	0.926	0.920	0.995
6	0.966	0.959	0.970
7	0.854	0.830	0.995
8	0.935	0.895	0.968
9	0.947	0.855	0.989
10	0.983	0.956	0.993
11	0.858	0.835	0.992

Figure 2.18: Plots of AD for RHT, EVM, and CSA (proposed algorithm) for images from GREC2007 dataset.



N	Image	CPU time (sec.)		
	`go7-tr`	EVM	RHT	CSA
8,321	6	46.892	3.764	0.120
8,478	5	45.104	4.389	0.125
14,817	7	54.272	7.228	0.145
15,189	10	29.730	37.527	0.152
19,182	11	68.625	20.021	0.231
25,358	8	343.077	21.350	0.166
29,072	2	1,298.658	33.309	0.121
44,717	9	938.428	20.903	0.205
53,156	4	1,822.414	17.229	0.217
53,899	3	1,890.005	185.968	0.180

Figure 2.19: CPU time comparison of RHT, EVM, and CSA (proposed algorithm) for images from GREC2007 dataset.

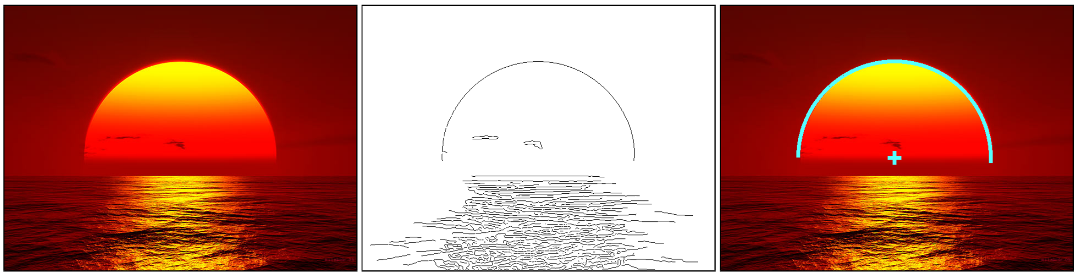

(a) (b) (c)

Figure 2.20: Result by our algorithm on the image `sunset`. (a) Input image. (b) Edge set obtained by Canny edge detection algorithm. (c) Output image.

Figure 2.21: Result by our algorithm on two more natural images.

Figure 2.22: Result by our algorithm on some real-world images.

Chapter 3

Covering a Digital Disc with Concentric Circles in $\mathbb{Z}^2$

3.1 Introduction

There exist various works on the characterization of digital circles, rings, discs, and circular arcs [Chan and Thomas (1995), Davies (1988a), Doros (1984), Haralick (1974), Nagy (2004), Pal and Bhowmick (2012), Thomas and Chan (1989), Worring and Smeulders (1995), Yuen and Feng (1996)]. Such a characterization is required for solving many scientific problems involving digital circles and circular shapes, since the properties of Euclidean/real circles are often inadequate and inappropriate to deal those problems. Hence, with the emergence of new digital paradigms, such as digital calculus [Nakamura and Aizawa (1984)], digital geometry [Asano *et al.* (2009), Klette and Rosenfeld (2004a)], theory of words and numbers [Klette and Rosenfeld (2004b), Mignosi (1991)], etc., proper characterization of geometric primitives in general, and digital circles and discs in particular, is required to enrich our understanding and to enhance these new-born paradigms of digital era.

Possibly the most significant problem in which the characterization of a digital circle helps, is its construction. Constructing a digital circle in a computationally efficient way is about half-a-century-old problem, originated during the earliest period of scan-conversion technique [Badler (1977), Bresenham (1977), Chung (1977), Danielsson (1978), Doros (1979), Horn (1976), Kulpa (1979), Pitteway (1974), Shimizu (1981)]. In later years, several approaches have been suggested for improving the method of circle construction or circle approximation [Bhowmick and Bhattacharya (2008), Blinn (1987), Bresenham (1985), Hsu *et al.* (1993), Mcllroy (1983), Nagy and Strand (2011), Suenaga *et al.* (1979), Wright (1990), Wu and Rokne (1987), Yao and Rokne (1995)]. Added to this, is the more challenging problem on recognizing circular arcs/objects in digital images, which have several solutions based on characterization and parameterization of circular arcs [Chattopadhyay *et al.* (1994), Chen and Chung (2001), Chiu and Liaw (2005), Coeurjolly *et al.* (2004), Davies (1987), Ho and Chen (1995), Kim and Kim (2001), Kulpa and Kruse (1983), Nakamura and Aizawa (1984), Pal and Bhowmick (2012), Pla (1996), Rosin and West (1988)].

This chapter is focused on finding and characterizing the absente-pixels (2D points with integer coordinates) in the cover of a digital disc with concentric digital circles. That absentees occur in such covers is possibly a lesser fact. The greater fact is, they occur in multitude—an observation that motivates the requirement of their proper characterization.

The chapter is organized as follows. We introduce few definitions and important properties in Sec. 3.2 relevant to digital circles and digital discs considered in our work. In Sec. 3.2.1, we derive the necessary and sufficient condition for a disc pixel to be an absentee, i.e., the condition satisfied by a pixel belonging to a digital disc of (integer) radius r but not belonging to any digital circle of radius $s \in \{0, 1, 2, \ldots, r\}$. Using these results, we characterize the set of disc absentees, which are not covered by the set of concentric digital circles, in a novel manner using the notion of *infimum parabola* and *supremum parabola*, as explained in Sec. 3.3. Based on all these, we derive the count of disc absentees. In Sec. 3.4, we prove that the number of absentee for a disc of radius r is $\Theta(r^2)$. The relative count of absentees w.r.t. total pixels in a disc is a rational number for a finite radius r and tends to an irrational limit as $r \to \infty$, which is shown in Sec. 3.4.1. Finally, in Sec. 3.5, we present our test results and draw the concluding notes with a few interesting problems related to this chapter.

3.2 Characterizing the Disc Absentees

There exist several definitions of digital circles and digital discs in the literature, depending on whether the radius and the center coordinates are real or integer values. Irrespective of the real- or integer-valued radius and center coordinates, a digital circle or a digital disc is defined as a set of points with integer coordinates, which are called *digital points* or *pixels* [Klette and Rosenfeld (2004a)]. In this chapter, we consider the *grid intersection digitization* [Klette and Rosenfeld (2004a), Stelldinger (2007)] of a real circle with integer radius and having center with integer coordinates. Such a digitization produces a digital circle, which can be generated by the well-known *midpoint circle algorithm* or the *Bresenham*

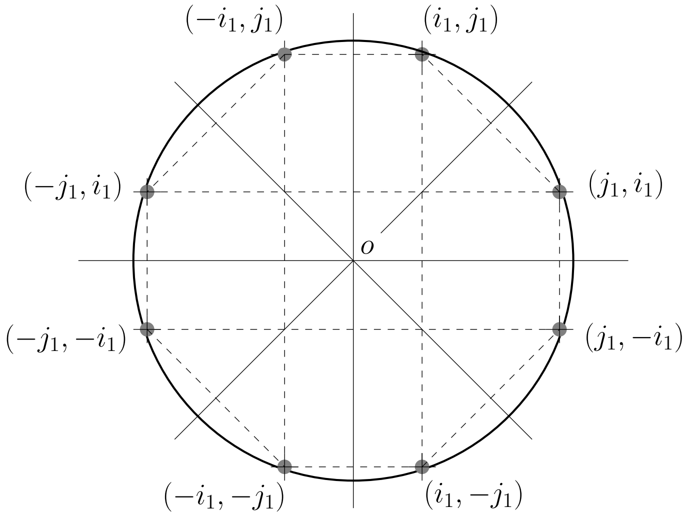


Figure 3.1: 8-symmetric points $\{(i,j) : \{|i|\} \cup \{|j|\} = \{i_1, j_1\}\}$ in a digital circle $\mathcal{C}^{\mathbb{Z}}(r)$.

circle algorithm [Foley *et al.* (1993), Pitteway (1974)], and its definition is as follows.

Definition 3.2.1 *A digital circle with radius* $r \in \mathbb{Z}^+$ *and center* $o(0,0)$ *is a set of digital points or pixels, given by*

$$\mathcal{C}^{\mathbb{Z}}(o,r) = \left\{(i,j) \in \mathbb{Z}^2 : \{|i|\} \cup \{|j|\} = \{i_1, j_1\} \wedge 0 \leq i_1 \leq j_1 \leq r \wedge |j_1 - \sqrt{r^2 - i_1^2}| < \frac{1}{2}\right\}.$$

All the results in this chapter are valid for any center with integer coordinates. So, for sake of simplicity, henceforth we consider the center as o and use the notation $\mathcal{C}^{\mathbb{Z}}(r)$ instead of $\mathcal{C}^{\mathbb{Z}}(o,r)$. We specify it explicitly when the center is not o.

A real point or a pixel (x,y) is said to be lying in Octant 1 if and only if $0 \leqslant x \leqslant y$ (Figure 3.1). We use the notation $\mathcal{C}_1^{\mathbb{Z}}(r)$ to denote Octant 1 of $\mathcal{C}^{\mathbb{Z}}(r)$, and $\mathbb{Z}_1^2$ to denote all points in Octant 1 of $\mathbb{Z}^2$. Note that in Def. 3.2.1, the digital point (i_1, j_1) belongs to $\mathcal{C}_1^{\mathbb{Z}}(r)$, and hence $\{(i,j) : \{|i|\} \cup \{|j|\} = \{i_1, j_1\}\}$ constitutes the eight symmetric points corresponding to (i_1, j_1), as shown in Figure 3.1. Rewriting the equation in symmetry-free form, we get the following definition of a digital circle.

Definition 3.2.2 $\mathcal{C}^{\mathbb{Z}}(r) = \left\{(i,j) \in \mathbb{Z}^2 : \left|\max(|i|,|j|) - \sqrt{r^2 - (\min(|i|,|j|))^2}\right| < \frac{1}{2}\right\}.$

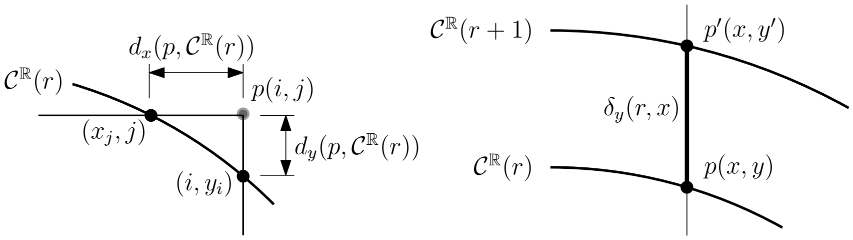


Figure 3.2: Vertical and horizontal distances.

Definition 3.2.3 *A digital disc* $\mathcal{D}^{\mathbb{Z}}(r)$*, having radius* $r \in \mathbb{Z}^+$*, consists of the points constituting* $\mathcal{C}^{\mathbb{Z}}(r)$ *and all digital points lying inside* $\mathcal{C}^{\mathbb{Z}}(r)$*, and is given by*

$$\mathcal{D}^{\mathbb{Z}}(r) = \left\{ (i, j_c) \in \mathbb{Z}^2 : \ 0 \leq i \cdot i_c \leq i_c^2 \ \wedge \ \left| \max(|i_c|, |j_c|) - \sqrt{r^2 - (\min(|i_c|, |j_c|))^2} \right| < \tfrac{1}{2} \ \right\}.$$

Note that in Def. 3.2.3, the condition $0 \leq i \cdot i_c \leq i_c^2$ relates a disc pixel (i, j_c) to a circle pixel (i_c, j_c), as $0 \leq i_c$ implies $0 \leq i \leq i_c$ and $i_c \leq 0$ implies $i_c \leq i \leq 0$.

Let $\mathcal{C}^{\mathbb{R}}(r)$ denote the real circle with integer radius r. We use the notation $p(x,y) \in \mathcal{C}^{\mathbb{R}}(r)$ if $p(x,y)$ lies on the circumference of $\mathcal{C}^{\mathbb{R}}(r)$, and $p(x,y) \in \mathcal{C}_1^{\mathbb{R}}(r)$ if p lies in Octant 1 and on the circumference of $\mathcal{C}^{\mathbb{R}}(r)$. If a pixel $p(i,j)$ lies in Octant 1 and outside $\mathcal{C}^{\mathbb{R}}(r)$, such that $j \leqslant r$, then its *x-distance* or *horizontal distance* from $\mathcal{C}^{\mathbb{R}}(r)$ is given by $d_x(p, \mathcal{C}^{\mathbb{R}}(r)) = i - x_j$, considering that the horizontal line through p intersects $\mathcal{C}^{\mathbb{R}}(r)$ in the first quadrant at (x_j, j). Similarly, if p lies in Octant 1 and outside $\mathcal{C}^{\mathbb{R}}(r)$, such that $i \leqslant r$, then its *y-distance* or *vertical distance* from $\mathcal{C}^{\mathbb{R}}(r)$ is $d_y(p, \mathcal{C}^{\mathbb{R}}(r)) = j - y_i$, considering (i, y_i) to be the point of intersection between the vertical line through p and first quadrant of $\mathcal{C}^{\mathbb{R}}(r)$. Thus, the vertical distance of a point $p(x,y) \in \mathcal{C}_1^{\mathbb{R}}(r)$ from $\mathcal{C}^{\mathbb{R}}(r+1)$ determines the *y-distance between* $\mathcal{C}^{\mathbb{R}}(r)$ *and* $\mathcal{C}^{\mathbb{R}}(r+1)$ *at abscissa* x, and is given by $\delta_y(r,x) = y' - y$, considering that $p'(x,y') \in \mathcal{C}_1^{\mathbb{R}}(r+1)$, as shown in Figure 3.2. Given below is Lemma 3.2.1 that states the monotonicity of $\delta_y(r,x)$ in Octant 1. Lemma 3.2.1 is easy to prove; it is used to find the supremum of $\delta_y(r,x)$, as stated in Lemma 3.2.2.

Lemma 3.2.1 *The value of* r *remaining unchanged,* $\delta_y(r,x)$ *increases monotonically with* x*.*

Lemma 3.2.2 $\delta_y(r, x) < \sqrt{2}$.

Proof. From Lemma 3.2.1, for a given value of r, $\delta_y(r, x)$ is maximum in Octant 1 when x is maximum, which occurs when $x = y$, or, $x^2 = y^2 = r^2/2$ and $y'^2 = (r+1)^2 - x^2 = (r+1)^2 - r^2/2$. Hence,

$$\delta_y(r, x) \quad < \sqrt{(r+1)^2 - \frac{r^2}{2}} - \frac{r}{\sqrt{2}} = \frac{\sqrt{(r+2)^2 - 2} - r}{\sqrt{2}}. \tag{3.1}$$

As $\sqrt{(r+2)^2 - 2} < r + 2$, we get $\delta_y(r, x) < \sqrt{2}$. $\square$

Definition 3.2.4 *A pixel p is an absente-pixel (or simply an absentee) if and only if $p \in \mathcal{D}^{\mathbb{Z}}(r)$ and $p \notin \bigcup\limits_{s=1}^{r} \mathcal{C}^{\mathbb{Z}}(s)$. Hence, the set of disc absentees is given by $\mathcal{A}^{\mathbb{Z}^2}(r) = \mathcal{D}^{\mathbb{Z}}(r) \setminus \bigcup\limits_{s=1}^{r} \mathcal{C}^{\mathbb{Z}}(s)$.*

Figure 3.3 shows the absente-pixels for $r = 20$ over all the eight octants. (Octants are numbered in clockwise order; for Octant 1, $0 \leq i \leq j$; for Octant 2, $0 \leq j \leq i$; $\ldots$; for Octant 8, $0 \leq -i \leq j$.) It is evident that an absentee is a *disc pixel* but not any *circle pixel*. Hence, an absentee lies between two digital circles of consecutive radii. In particular, for any absentee in Octant 1, we have the following lemma.

Lemma 3.2.3 *If $p(i, j)$ is an absentee in Octant 1, then $(i, j-1) \in \mathcal{C}^{\mathbb{Z}}(r)$ and $(i, j+1) \in \mathcal{C}^{\mathbb{Z}}(r+1)$ for some $r \in \mathbb{Z}^+$.*

Proof. As $p(i, j) \notin \mathcal{C}^{\mathbb{Z}}(r)$ and $p(i, j) \notin \mathcal{C}^{\mathbb{Z}}(r+1)$, we have

$$d_y(p, \mathcal{C}^{\mathbb{R}}(r)) > 1/2 \ \wedge \ d_y(p, \mathcal{C}^{\mathbb{R}}(r+1)) > 1/2.$$

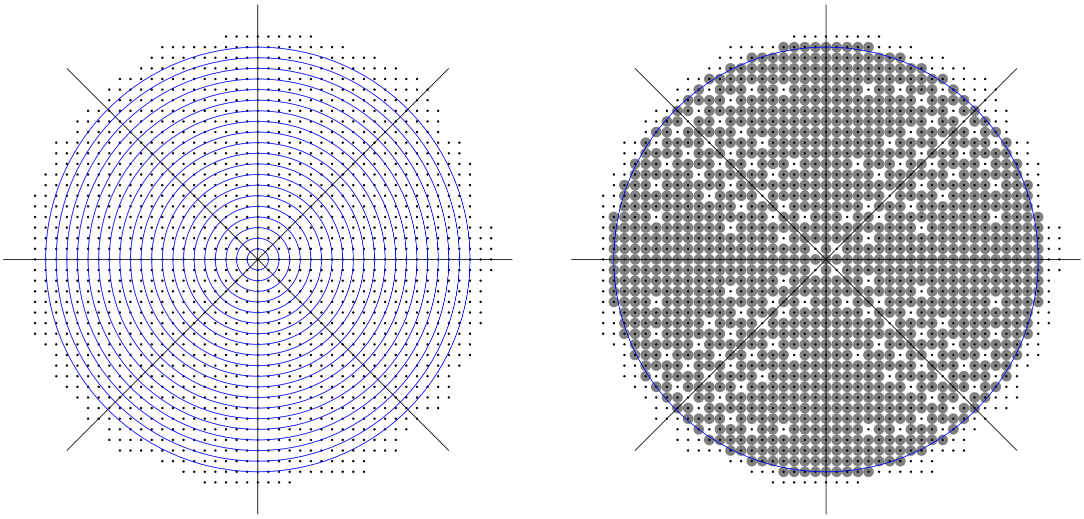
Hence, if $p'(i, j+1)$ is an absentee, then also

$$d_y(p', \mathcal{C}^{\mathbb{R}}(r)) > 1/2 \ \wedge \ d_y(p', \mathcal{C}^{\mathbb{R}}(r+1)) > 1/2.$$

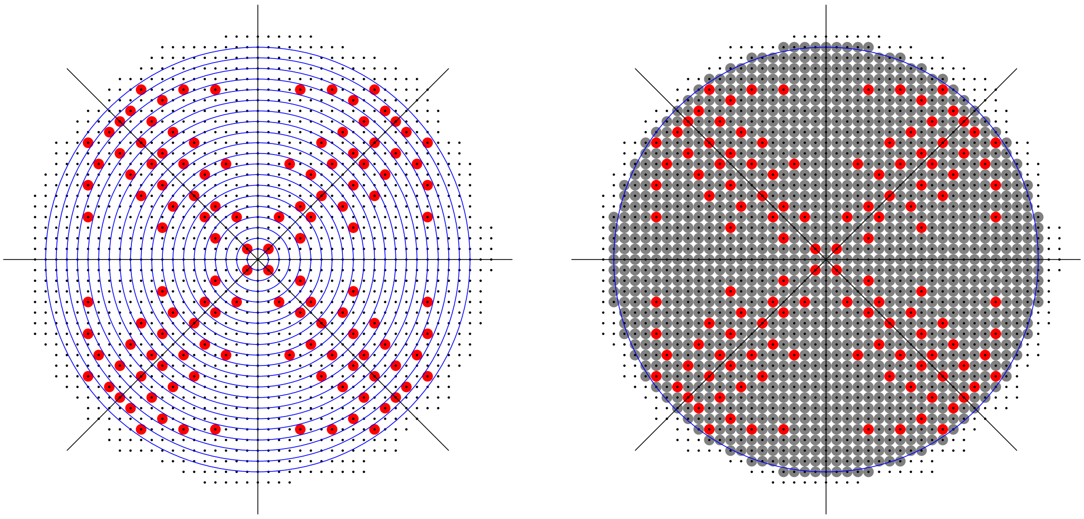
Thus,

$$\begin{aligned}\delta_y(r, i) &= d_y(p, \mathcal{C}^{\mathbb{R}}(r)) + d_y(p, p') + d_y(p', \mathcal{C}^{\mathbb{R}}(r+1)) \\ &> 1/2 + 1 + 1/2 \\ &= 2,\end{aligned}$$

which contradicts Lemma 3.2.2. Similar arguments can be given to show that $(i, j-1)$ is also not an absentee. In effect, $(i, j-1) \in \mathcal{C}^{\mathbb{Z}}(r)$ and $(i, j+1) \in \mathcal{C}^{\mathbb{Z}}(r+1)$ for some $r \in \mathbb{Z}^+$. $\square$



(a) $\left\{\mathcal{C}^{\mathbb{R}}(s) : 1 \leq s \leq r\right\}$ (b) $\bigcup_{s=1}^{r} \mathcal{C}^{\mathbb{Z}}(s)$



(c) $\mathcal{A}^{\mathbb{Z}^2}(r) = \mathcal{D}^{\mathbb{Z}}(r) \smallsetminus \bigcup_{s=1}^{r} \mathcal{C}^{\mathbb{Z}}(s)$ (d) $\mathcal{D}^{\mathbb{Z}}(r)$

Figure 3.3: Absentee-pixels (red) for $r = 20$ while covering a digital disc by circle pixels (gray) in $\mathbb{Z}^2$.

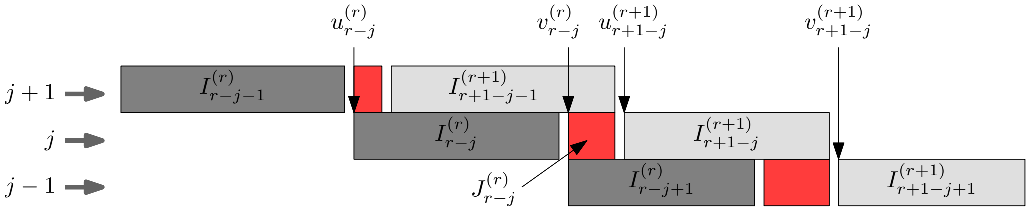


Figure 3.4: The interval $J^{(r)}_{r-j}$ in which an absentee lies. Note that the light-gray intervals correspond to radius $r+1$, and the deep-gray intervals to radius r.

From Lemma 3.2.3, the following corollaries can be easily made.

Corollary 3.2.4 *An absente-pixel always lies between two digital circles of consecutive radii.*

Corollary 3.2.5 *In Octant 1, no two equi-abscissa absentees exist whose ordinates differ by unity.*

Corollary 3.2.6 *For each absentee (i, j) in Octant 1, there exists an absentee (i', j') in each other octant such that $\{|i'|\} \cup \{|j'|\} = \{i, j\}$. In other words, (i, j) is an absentee in Octant 1 if and only if each other point of $\{(i', j') : \{|i'|\} \cup \{|j'|\} = \{i, j\}\}$ is also an absentee in some other octant.*

3.2.1 Necessary and Sufficient Conditions for a Disc Absentee

We use the following lemma from a previous work [Bhowmick and Bhattacharya (2008)] to obtain the necessary and sufficient conditions to test whether a given pixel is an absentee or not.

Lemma 3.2.7 *The squares of abscissae of the pixels in $\mathcal{C}^{\mathbb{Z}}_1(r)$ whose ordinates are j lie in the interval $I^{(r)}_{r-j} = \left[u^{(r)}_{r-j}, v^{(r)}_{r-j}\right)$, where*

$$\begin{aligned} u^{(r)}_{r-j} &= r^2 - j^2 - j, \\ v^{(r)}_{r-j} &= r^2 - j^2 + j. \end{aligned} \tag{3.2}$$

From Lemma 3.2.7, the interval containing the squares of abscissae of the pixels in $\mathcal{C}^{\mathbb{Z}}_1(r+1)$ with ordinate j is, therefore, given by

$$I^{(r+1)}_{r+1-j} := \left[u^{(r+1)}_{r+1-j}, v^{(r+1)}_{r+1-j}\right) = \left[(r+1)^2 - j^2 - j, (r+1)^2 - j^2 + j\right),$$

as shown in Figure 3.4. Thus, the difference between the lower limit of $I^{(r+1)}_{r+1-j}$ and the upper limit of $I^{(r)}_{r-j}$ is given by

$$u^{(r+1)}_{r+1-j} - v^{(r)}_{r-j} = ((r+1)^2 - j^2 - j) - (r^2 - j^2 + j) = 2(r-j) + 1. \tag{3.3}$$

As $j \leqslant r$ (Definition 3.2.1), we have the following lemma.

Lemma 3.2.8 *For $r > 0$, the intervals $I^{(r)}_{r-j}$ and $I^{(r+1)}_{r+1-j}$ are disjoint and $u^{(r+1)}_{r+1-j} > v^{(r)}_{r-j}$.*

Lemma 3.2.8, in fact, authenticates the uniqueness of the digital circle on which a given pixel p lies. For, if $p = (i, j)$ be such that the square number i^2 lies in $I^{(r)}_{r-j}$, then i^2 does not lie in $I^{(r+1)}_{r+1-j}$ in particular, and in no other interval $I^{(s)}_{s-j}$ in general, whence p lies only on the digital circle $\mathcal{C}^{\mathbb{Z}}(r+1)$. Furthermore, if i^2 lies somewhere in the interval lying right of $I^{(r)}_{r-j}$ and left of $I^{(r+1)}_{r+1-j}$, then p becomes an absentee. Hence, we have the following lemma (see Figure 3.4).

Lemma 3.2.9 *A pixel $p(i, j)$ is an absentee if and only if i^2 lies in $J^{(r)}_{r-j} := \left[v^{(r)}_{r-j}, u^{(r+1)}_{r+1-j}\right)$ for some $r \in \mathbb{Z}^+$.*

An example of absentee is $(2, 4)$, since for $j = 4$, we have $r = 4$ for which $v^{(r)}_{r-j} = r^2 - j^2 + j = 16 - 16 + 4 = 4$ and $u^{(r+1)}_{r+1-j} = (r+1)^2 - j^2 - j = 25 - 16 - 4 = 5$, thus giving $J^{(4)}_0 = [4, 5) = [4, 4]$ in which lies the square number 4. On the contrary, $(3, 4)$ is not an absentee, as for $j = 4$, there is no such r for which $J^{(r)}_{r-4}$ contains 3^2; in fact, for $j = 4$, we get the interval $I^{(5)}_{5-4} = [5^2 - 4^2 - 4, 5^2 - 4^4 + 4) = [5, 12]$ with $r = 5$, which contains 3^2, thereby making $(3, 4)$ a point on $\mathcal{C}^{\mathbb{Z}}(o, 5)$.

Although Lemma 3.2.9 provides a way to decide whether or not a given pixel is an absentee, it requires to find for which value(s) of r the existence of square numbers in $J^{(r)}_{r-j}$ has to be checked. The following theorem tells exactly the value of r for which $J^{(r)}_{r-j}$ has to be tested for a given pixel $p(i, j)$.

Theorem 3.2.10 *A pixel $p(i, j)$ is an absentee if and only if $i^2 \in J^{(r)}_{r-j}$, where $r = \max\left\{s \in \mathbb{Z} : s^2 < i^2 + j^2\right\}$.*

Proof. If $p(i, j)$ is an absentee, then by Lemma 3.2.3, the real circles $\mathcal{C}^{\mathbb{R}}(r)$ and $\mathcal{C}^{\mathbb{R}}(r+1)$ intersect the vertical line $x = i$ respectively below and above p (Figure 3.4). If these respective points of intersection be $q(i, y_i)$ and $q'(i, y'_i)$, then $y_i < j < y'_i$, which yields

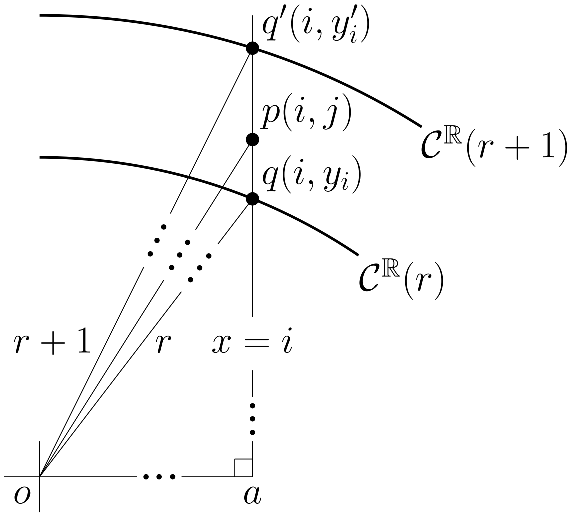


Figure 3.5: Illustration for proof of Theorem 3.2.10.

$i^2 + y_i^2 < i^2 + j^2 < i^2 + {y'}_i^2$, or, $r^2 < i^2 + j^2 < (r+1)^2$, or, $i^2 + j^2 \in \left(r^2, (r+1)^2\right)$, which implies $r = \max\left\{s \in \mathbb{Z} : s^2 < i^2 + j^2\right\}$.

The reverse proof is straightforward. Since a value of r exists, namely, $\max\left\{s \in \mathbb{Z} : s^2 < i^2 + j^2\right\}$ for which $i^2 \in J_{r-j}^{(r)}$, by Lemma 3.2.9, (i, j) becomes an absentee. □

3.3 Characterizing the Family

To characterize the absentees as a whole, we use Eqn. 3.2 for the expanded form of (the lower and the upper limits of) $J_{r-j}^{(r)}$. We replace r by $j + k$, where k is the index of a run of pixels in $\mathcal{C}_1^{\mathbb{Z}}(r)$ with ordinate j, i.e., $j = r - k$. (For $k = 0$, we have the topmost run with $j = r$; for $k = 1$, we have the second run from the top with $j = r - 1$; and so on). Similarly, we replace $r + 1$ by $j + (k + 1)$, where $k + 1$ is the index of a run of pixels in $\mathcal{C}_1^{\mathbb{Z}}(r)$ with ordinate j, i.e., $j = (r + 1) - (k + 1) = r - k$. Thus,

$$\begin{aligned} v_{r-j}^{(r)} &= (2k+1)j + k^2, \\ u_{r+1-j}^{(r+1)} &= (2k+1)j + (k+1)^2. \end{aligned} \tag{3.4}$$

Hence, if $p(i, j)$ is a point in Octant 1 and lies on k^{th} run of $\mathcal{C}_1^{\mathbb{Z}}(r)$, then

$$i^2 < (2k+1)j + k^2; \tag{3.5}$$

and if $p(i, j)$ is a point in Octant 1 and lies left of the $(k+1)^{th}$ run of $\mathcal{C}_1^{\mathbb{Z}}(r+1)$, then

$$i^2 < (2k+1)j + (k+1)^2. \tag{3.6}$$

Equations 3.5 and 3.6 correspond to two parabolic regions in the real (xy-)plane on replacing i and j by x and y, respectively, k being considered as a constant. These *open parabolic regions* are given by

$$\begin{aligned} \underline{P}_k &: x^2 < (2k+1)y + k^2, \\ \overline{P}_k &: x^2 < (2k+1)y + (k+1)^2. \end{aligned} \tag{3.7}$$

Evidently, the pixels or integer points lying in the region given by $\overline{P}_k \setminus \underline{P}_k$ in Octant 1 for a given pair of j and k—and hence for a given (r, j)-pair—are absentees in Octant 1. Hence, we have the following lemma.

Lemma 3.3.1 *The number of square numbers in*

$$J_{r-j}^{(r)} = \left|\left\{(i, j) : \ (i, j) \in \left(\overline{P}_k \setminus \underline{P}_k\right) \cap \ \mathbb{Z}_1^2\right\}\right|.$$

From Lemma 3.3.1, we can derive the region of all absentees for a given value of k by considering all possible values of j for $r \geqslant 0$ so that $r - j = k$. Thus, all the integer points of Octant 1, which are contained in the following *half-open parabolic strip*, are absentee points.

$$P_k := \overline{P}_k \setminus \underline{P}_k = (2k+1)y + k^2 \leq x^2 < (2k+1)y + (k+1)^2. \tag{3.8}$$

In essence, we have the following lemma.

Lemma 3.3.2 *All pixels in* $F_k := P_k \cap \mathbb{Z}_1^2$ *are absentees.*

The above characterization provides a way to obtain the absentees for different values of k. For $k = 0$, we get the absentees corresponding to P_0 as those satisfying $j \leqslant i^2 < j+1$. For $j = 1$, we get $i = 1$, whence $(1, 1)$ is an absentee lying between the digital circles of radius $r = j = 1$ (as $r - j = k = 0$) and radius $r + 1 = 2$. For $j = 2$ (or $j = 3$), there is no such i satisfying $2 \leqslant i^2 < 3$ (or $3 \leqslant i^2 < 4$), which implies there is no absentee having ordinate 2 (or 3) and lying between the digital circles of radii 2 and 3 (or, 3 and 4). For $j = 4$, we have the square $i^2 = 2^2$ satisfying $4 \leqslant i^2 < 5$, which implies $(2, 4)$ is an absentee. Similar absentees occur for $j = 9, 16, 25, \ldots$ corresponding to $k = 0$, as shown in Figure 3.3 and Figure 3.6.

For $k = 1$, the absentees correspond to all the integer points in P_1. Hence, the integer points obtained on solving $3j+1 \leqslant i^2 < 3j+4$, namely $(4,5),(5,8),(6,11),(7,16),\ldots$, are all the absentees corresponding to $k = 1$. Similarly, $k = 2$ correspond to P_2, and on solving $5j + 4 \leqslant i^2 < 5j + 9$, we get the successive absentees as $(6,6),(7,9),(8,12),(9,15),\ldots$. In general, the absentees corresponding to P_k are, therefore, given by the integer pairs satisfying $(2k + 1)j + k^2 \leqslant i^2 < (2k + 1)j + (k + 1)^2$. The family of all the half-open parabolic strips, $P_0, P_1, P_2, \ldots$, thus contains all the absentees in Octant 1, as stated in the following theorem.

Theorem 3.3.3 *Only and all the absentees of Octant 1 lie in*

$$\mathfrak{F} := \left\{P_k \cap \mathbb{Z}_1^2 : k = 0, 1, 2, \ldots\right\}.$$

Proof. Follows from Lemma 3.3.2. □

Figures 3.6 and 3.7 show the parabolic characterization of absentees. From Eqn. 3.7, the equation of the *infimum parabola* defining the inner boundary of the parabolic region $\underline{P}_k$ is $x^2 = (2k + 1)y + k^2$, or, $x^2 = (2k + 1)(y + \frac{k^2}{2k+1})$, whose axis is $x = 0$ and vertex is $(0, -\frac{k^2}{2k+1})$. Similarly, the *supremum parabola* defining the outer boundary of $\overline{P}_k$ is $x^2 = (2k + 1)(y + \frac{(k+1)^2}{2k+1})$, whose axis is $x = 0$ and vertex is $(0, -\frac{(k+1)^2}{2k+1})$. Clearly, as k increases, the vertices move downward along the y-axis.

Lemma 3.3.4 *For a given k, $P_k \cap \mathbb{Z}_1^2$ contains exactly one absentee on each vertical grid line.*

Proof. From Lemma 3.2.9 and Eqn. 3.4, a pixel (i, j) is an absentee and lies in P_k if and only if

$$\begin{aligned}
&(2k + 1)j + k^2 \leqslant i^2 < (2k + 1)j + (k + 1)^2 \\
\text{or,}\quad &k^2 - i^2 \leqslant -(2k + 1)j < (k + 1)^2 - i^2 \\
\text{or,}\quad &i^2 - (k + 1)^2 < (2k + 1)j \leqslant i^2 - k^2. \qquad (3.9)
\end{aligned}$$

Since the interval $\left(i^2 - (k + 1)^2, i^2 - k^2\right]$ contains $2k + 1$ consecutive integers, there occurs exactly one element j' in this interval such that $j' \bmod (2k + 1) = 0$. Thus, on dividing j' by $2k + 1$, we get j as the quotient, whence the proof. □

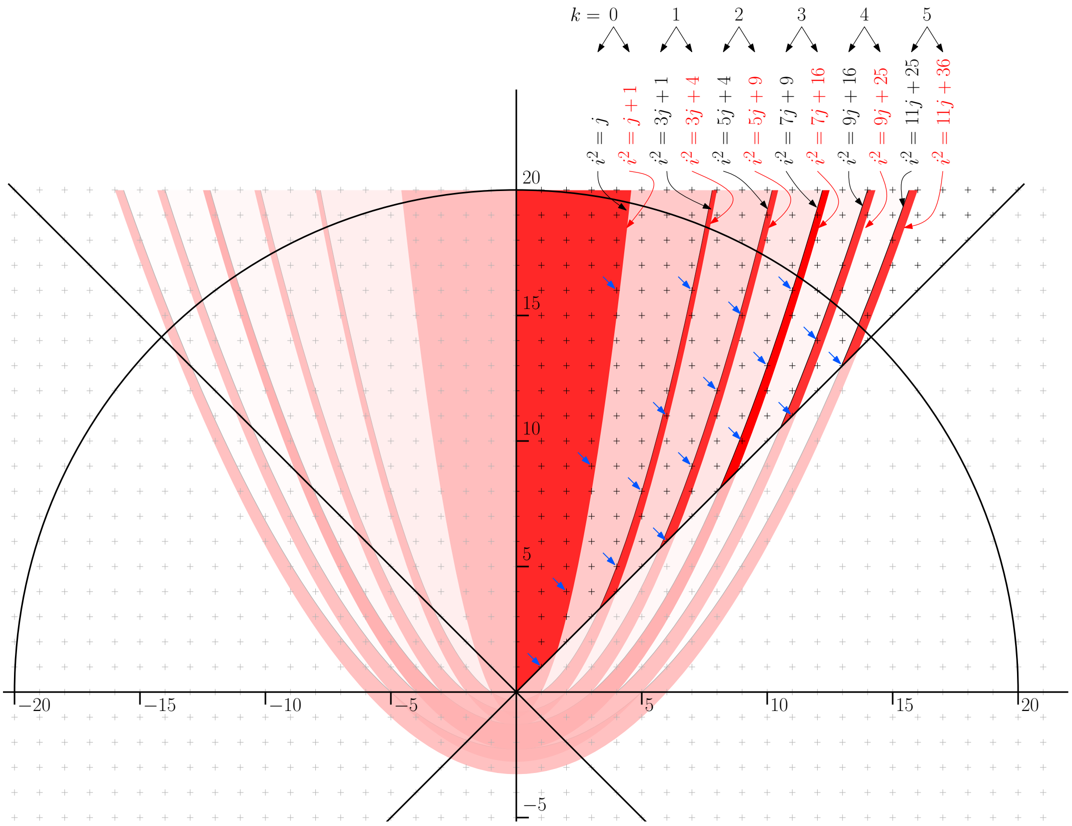


Figure 3.6: Parabolic characterization of the absente-pixels. The absentees corresponding to circles up to radius 20 are pointed by blue arrows. Octant 1 is made bright while other seven octants are dimmed.

Lemma 3.3.5 *The count of absentees contained by the parabolic strip* P_k *in* $\mathcal{D}_1^{\mathbb{Z}}(r)$ *is given by*

$$|\mathcal{A}_k^{\mathbb{Z}^2}(r)| = \left\lceil \sqrt{(2k+1)r - k(k+1)} \right\rceil - \left\lceil \left((2k+1) + \sqrt{8k^2+4k+1}\right)/2 \right\rceil.$$

PROOF. An illustration of the proof is given in Figure 3.8. If $p(x_p, y_p)$ be the point of intersection of $y = x$ with the parabolic boundary of $\underline{P}_k$, i.e., $x^2 = (2k+1)y + k^2$ (Eqn. 3.7), then

$$x_p = \frac{(2k+1) + \sqrt{8k^2+4k+1}}{2}. \tag{3.10}$$

Similarly, if $q(x_q, y_q)$ be the point of intersection of $y = r - k - 1$ with the parabolic

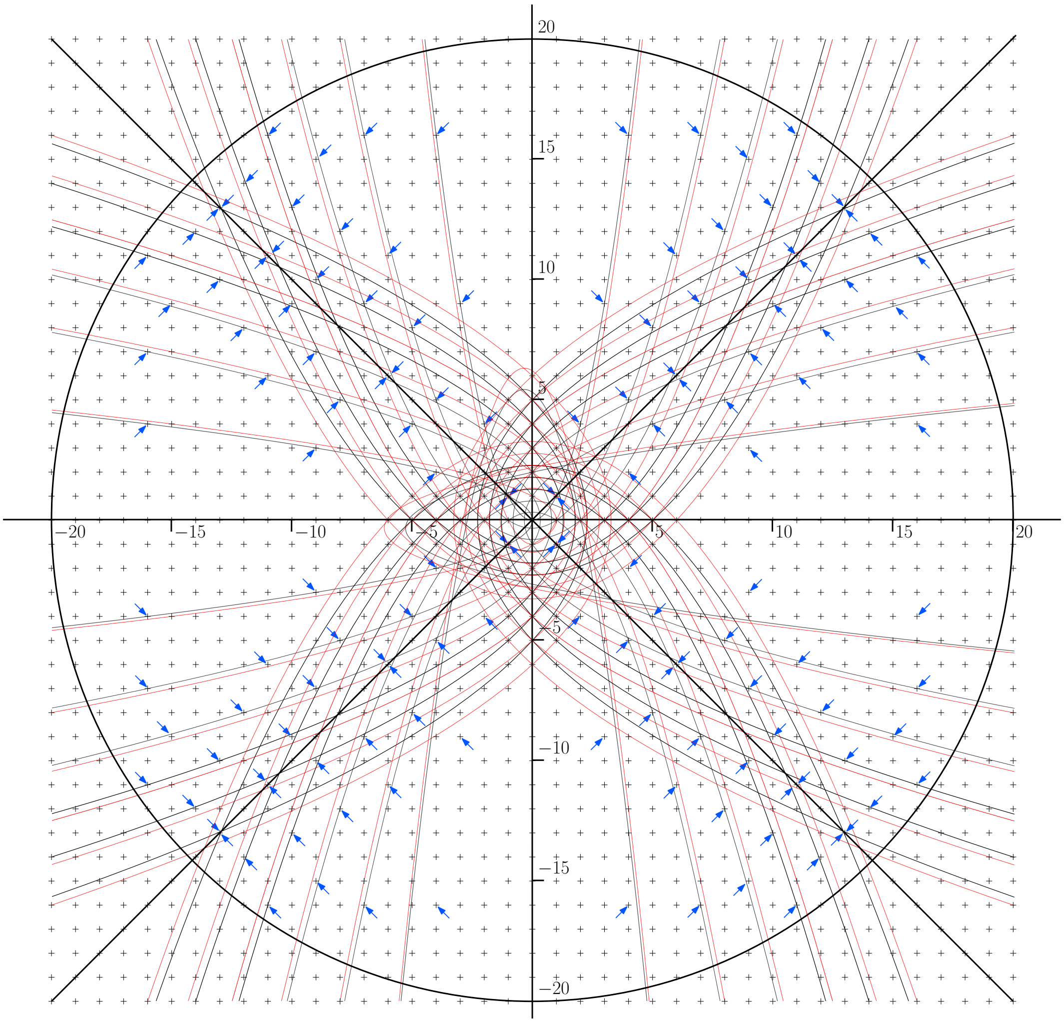


Figure 3.7: Parabolic characterization of the absente-pixels in all eight octants. The absentees up to radius 20 are shown by blue arrows. Infimum parabolas are shown in black and supremum parabolas in red.

boundary of $\overline{P}_k$, i.e., $x^2 = (2k+1)y + (k+1)^2$, then

$$x_q = \sqrt{(2k+1)r - k(k+1)}. \tag{3.11}$$

Hence, the first and the last ordinate lines passing through P_k (in Octant 1) are given by $x = \lceil x_p \rceil$ and $x = \lceil x_q \rceil - 1$, respectively. Since each of these vertical grid line contains exactly one absentee in P_k (Lemma 3.3.4), the total count of absentees in Octant 1 becomes $|\mathcal{A}^{\mathbb{Z}^2}(r)| = \lceil x_q \rceil - 1 - \lceil x_p \rceil + 1 = \lceil x_q \rceil - \lceil x_p \rceil$, and the proof follows. $\square$

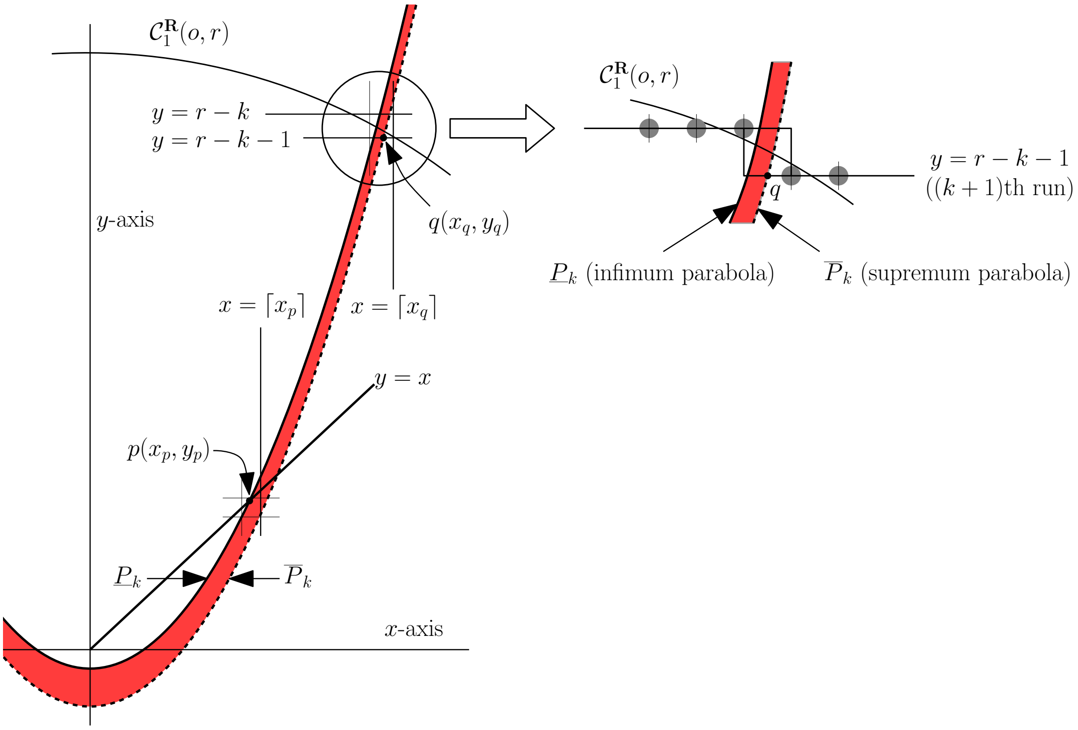


Figure 3.8: Illustration of Lemma 3.3.5.

Lemma 3.3.6 *For a given r, the number of half-open parabolic strips intersecting $\mathcal{C}_1^{\mathbb{Z}}(r)$ is given by $m_r = r - \lceil r/\sqrt{2} \rceil + 1$.*

Proof. Each parabolic strip P_k $(k \geq 0)$ corresponds to kth run of $\mathcal{C}_1^{\mathbb{Z}}(r)$ (Eqn. 3.7). The point of intersection of the line $y = x$ with the real circle $\mathcal{C}^{\mathbb{R}}(r)$ is $(r/\sqrt{2}, r/\sqrt{2})$. Hence, the number of horizontal grid lines intersecting Octant 1 of $\mathcal{C}^{\mathbb{R}}(r)$ is $r - \lceil r/\sqrt{2} \rceil + 1$. Each of these grid lines contains a run of $\mathcal{C}_1^{\mathbb{Z}}(r)$, whence the proof. $\square$

Using the above lemmas, we get the total count of absentees in $\mathcal{D}_1^{\mathbb{Z}}(r)$ as follows.

Theorem 3.3.7 *The total count of absentees lying in $\mathcal{D}^{\mathbb{Z}}(r)$ is given by*

$$|\mathcal{A}^{\mathbb{Z}^2}(r)| = 8 \sum_{k=0}^{m_r - 1} |\mathcal{A}_k^{\mathbb{Z}^2}(r)|,$$

where $|\mathcal{A}_k^{\mathbb{Z}^2}(r)| = \left\lceil \sqrt{(2k+1)r - k(k+1)} \right\rceil - \left\lceil 2k + 1 + \frac{1}{2}\sqrt{(8k^2 + 4k + 1)} \right\rceil$ *and* $m_r = r - \lceil r/\sqrt{2} \rceil + 1$.

PROOF. From Lemma 3.3.5 and Lemma 3.3.6, we get the count of absentees in $\mathcal{D}_1^{\mathbb{Z}}(r)$ by summing up $|\mathcal{A}_k^{\mathbb{Z}^2}(r)|$ for all possible values of k from 0 to $m_r - 1$. This count, multiplied by 8, produces the total count of absentees for all the eight octants, i.e., $\mathcal{D}^{\mathbb{Z}}(r)$. $\square$

Observe that the absentees in Octant 1 having their abscissa and ordinate values same (i.e., lie on $x = y$) also belong to Octant 2. Hence, the total count of absentees lying inside $\mathcal{C}^{\mathbb{Z}}(r)$ is given by $|\mathcal{A}^{\mathbb{Z}^2}(r)|$ minus the count of absentees lying on $x + y = 0$ or $x - y = 0$.

3.3.1 Algorithm to Find Absentees

Using Lemma 3.3.2 and the number-theoretic algorithm for construction of digital circles [Bhowmick and Bhattacharya (2008)], the algorithm to find the set of absentees is designed, as shown in Algorithm 1. The set of all absentees corresponding to a disc of radius r is denoted by $\mathcal{A}^{\mathbb{Z}^2}(r)$. This set can be constructed by checking each digital point of the disc using Lemma 3.2.9. But only those pixels that lie in the parabolic strips P_k for $k = 0, 1, \ldots, m_r$ are checked by our algorithm. This is shown in Steps 9–18. In Step 15, the absentee set $\mathcal{A}^{\mathbb{Z}^2}(r)$ is updated by including eight symmetric absentee points (or four, if $i = j$).

3.4 Order of Absentee Count

As mentioned in Sec. 3.2, a circle pixel belongs to the digital disc $\mathcal{D}^{\mathbb{Z}}(r)$ as well as some digital circle $\mathcal{C}^{\mathbb{Z}}(s)$ for $s \leq r$. An absente-pixel, on the contrary, belongs to $\mathcal{D}^{\mathbb{Z}}(r)$ but not to any $\mathcal{C}^{\mathbb{Z}}(s)$ for $s \leq r$ (see Figure 3.3). In order to count the absente-pixels inside $\mathcal{D}^{\mathbb{Z}}(r)$, the number of disc pixels and the number of circle pixels inside $\mathcal{D}^{\mathbb{Z}}(r)$ have to be computed. The number of circle pixels inside $\mathcal{D}^{\mathbb{Z}}(r)$ is given by summing up the number of pixels constituting the digital circles with radius s in $[0, r]$, as follows.

$$n_c = \sum_{s=0}^{r} |\mathcal{C}^{\mathbb{Z}}(s)|.$$

An algorithm for construction of a digital circle, e.g., [Bhowmick and Bhattacharya (2008)], can be used for computing $|\mathcal{C}^{\mathbb{Z}}(s)|$, which denotes the perimeter of a digital circle with radius s. However, here we first derive a closed form formula for perimeter of a digital

Algorithm 1: Disc-Absentee

```
Input: Radius $r$ of a disc
Output: Set of the absentees
$\mathcal{A}^{\mathbb{Z}^2}(r) \leftarrow \emptyset$
int $i \leftarrow 0, j \leftarrow r, s \leftarrow 0, w \leftarrow r-1, k \leftarrow 0, i_a, j_a$
int $l \leftarrow 2w$
while $j \geq i$ do
    repeat
        $s \leftarrow s + 2i + 1$
        $i \leftarrow i + 1$
    until $s \leq w$
    $i_a \leftarrow i - 1, j_a \leftarrow j$
    while $j_a \geq i_a$ do
        if $i_a^2 < (2k+1)j_a + k^2$ then
            $j_a \leftarrow j_a - 1$
        else
            if $i_a^2 < (2k+1)j_a + (k+1)^2$ then
                $\mathcal{A}^{\mathbb{Z}^2}(r) \leftarrow \mathcal{A}^{\mathbb{Z}^2}(r) \cup \{(i,j) : \{|i|\} \cup \{|j|\} = \{i_a, j_a\}\}$
            $i_a \leftarrow i_a - 1$
    $w \leftarrow w + l$
    $l \leftarrow l - 2$, $j \leftarrow j - 1$, $k \leftarrow k + 1$
return $\mathcal{A}^{\mathbb{Z}^2}(r)$
```

circle. Next, an approximate result for the number of disc pixels is shown. Using these, a tighter bound on the order of disc absentees is obtained.

It is easy to observe that no two circle pixels of $\mathcal{C}^{\mathbb{Z}}(r)$ in Octant 1 lie on the same vertical grid line. Hence, the circle pixels in Octant 1 can be counted by noting the integer distance of the point of intersection of the real straight line $y = x$ with the real circle $\mathcal{C}^{\mathbb{R}}(r)$ from the y-axis. The exact count depends on whether or not a circle pixel lies on $y = x$, as shown shortly (Theorem 3.4.2). Before that, we show the condition that determines whether or not a circle pixel lies on $y = x$. Note that Octant 1 and Octant 2 are symmetric with respect to $y = x$. Two cases may arise: (i) a circle pixel lies on $y = x$; (ii) $y = x$ passes through the middle of two circle pixels (Figure 3.9). Let $p'(x', y') \in \mathbb{R}^2$ be the point of intersection of $\mathcal{C}^{\mathbb{R}}(r)$ and $y = x$. Observe that $x' = y'$, and hence the pixel $p(i,j)$ corresponding to the point p' lies on $y = x$, as i and j are rounded values of $x'(= y')$. We have to check whether p belongs to $\mathcal{C}^{\mathbb{Z}}(r)$. Clearly, $x = r\sin(\frac{\pi}{4}) = \frac{r}{\sqrt{2}}$, and hence $i = j = \lfloor(x + \frac{1}{2})\rfloor$. If $p \in \mathcal{C}^{\mathbb{Z}}(r)$, then $i^2 + (j+e)^2 = r^2$, where $-\frac{1}{2} < e < \frac{1}{2}$. As $i = j$, we have $e = \sqrt{r^2 - i^2} - i$. Hence, we have the following lemma.

Lemma 3.4.1 *A pixel* $p(i,i) \in \mathcal{C}^{\mathbb{Z}}(r)$ *if* $-\frac{1}{2} < \sqrt{r^2 - i^2} - i < \frac{1}{2}$, *where* $i = \lfloor(\frac{r}{\sqrt{2}} + \frac{1}{2})\rfloor$.

If the pixel $p(i,i) \in \mathcal{C}^{\mathbb{Z}}(r)$, then the straight lines $y = 0$, $x = 0$, $y = x$ and $y = -x$ hit 2 pixels each, in total of 8 pixels of $\mathcal{C}^{\mathbb{Z}}(r)$. Also, the count of pixel in Octant 1 and in the area bounded by the lines $x = 0$ and $y = x$ is $i - 1$; hence in 8 octants, the count is $8(i-1)$. Therefore, the total count of pixels in $\mathcal{C}^{\mathbb{Z}}(r)$ is $8i$.

If the pixel $p(i,i) \notin \mathcal{C}^{\mathbb{Z}}(r)$, then the straight lines $y = 0$ and $x = 0$ hits 2 pixels each, in total 4 of pixels of $\mathcal{C}^{\mathbb{Z}}(r)$, but the straight lines $y = x$ and $y = -x$ do not pass through any pixel of $\mathcal{C}^{\mathbb{Z}}(r)$. The pixel $p(i,i)$ may lie inside or outside of the circle $\mathcal{C}^{\mathbb{R}}(r)$. If $p(i,i)$ lies inside, i.e., $\sqrt{i^2 + i^2} = \sqrt{2}i < r$, then the count of pixels in Octant 1 between the lines $x = 0$ and $y = x$ is i; hence in 8 octants, the count is $8i$. Therefore, the total count of pixels in $\mathcal{C}^{\mathbb{Z}}(r)$ is $8i + 4$. On the other hand, if $p(i,i)$ lies outside, i.e., $\sqrt{i^2 + i^2} = \sqrt{2}i > r$, then the total in 8 octants will be $8(i-1)$. Therefore, the total count of pixels in $\mathcal{C}^{\mathbb{Z}}(r)$ is $8i - 4$. To summarize, we have the following theorem.

Theorem 3.4.2 *For a digital circle* $\mathcal{C}^{\mathbb{Z}}(r)$, *its perimeter is* $|\mathcal{C}^{\mathbb{Z}}(r)| \in \{8i-4, 8i, 8i+4\}$, *where* $i = \left\lfloor\left(\frac{r}{\sqrt{2}} + \frac{1}{2}\right)\right\rfloor$. *In particular,*

$$|\mathcal{C}^{\mathbb{Z}}(r)| = \begin{cases} 8i & \textit{if } p(i,i) \in \mathcal{C}^{\mathbb{Z}}(r) \\ 8i+4 & \textit{if } p(i,i) \notin \mathcal{C}^{\mathbb{Z}}(r) \textit{ and } \sqrt{2}i < r \\ 8i-4 & \textit{if } p(i,i) \notin \mathcal{C}^{\mathbb{Z}}(r) \textit{ and } \sqrt{2}i > r \end{cases} \quad \textit{where } i = \left\lfloor\left(\frac{r}{\sqrt{2}} + \frac{1}{2}\right)\right\rfloor. \tag{3.12}$$

Examples Consider the three digital circles with $r = 2, 10, 15$, as shown in Figure 3.9. For $r = 10$, $i = \lfloor(\frac{10}{\sqrt{2}} + \frac{1}{2})\rfloor = 7$ and $e = \sqrt{r^2 - i^2} - i = 0.141 < \frac{1}{2}$; so the perimeter is $8i = 56$. For $r = 2$, $i = \lfloor(\frac{2}{\sqrt{2}} + \frac{1}{2})\rfloor = 1$ and $e = \sqrt{r^2 - i^2} - i = 0.732 > \frac{1}{2}$. Hence, no pixel of $\mathcal{C}^{\mathbb{Z}}(2)$ lies on the line $y = x$. Again, $\sqrt{2}i = 1.414 < r$; so the perimeter is $8i + 4 = 12$. For $r = 15$, $i = \lfloor(\frac{15}{\sqrt{2}} + \frac{1}{2})\rfloor = 11$ and $e = \sqrt{r^2 - i^2} - i = -0.802 < -\frac{1}{2}$. Hence, no pixel of $\mathcal{C}^{\mathbb{Z}}(15)$ lies on the line $y = x$. As $\sqrt{2}i = 15.556 > r$, the perimeter is $8i - 4 = 84$.

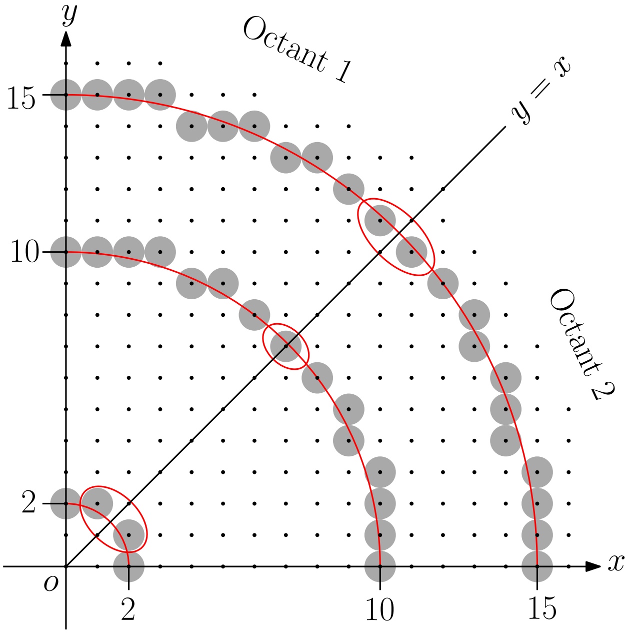


Figure 3.9: Octant 1 and Octant 2 of three representative digital circles. $r = 2(i = 1)$: perimeter $= 8i + 4 = 12$; $r = 10(i = 7)$: perimeter $= 8i = 56$; $r = 15(i = 11)$: perimeter $= 8i - 4 = 84$.

3.4.1 Absentee Count

Consider a digital disc $\mathcal{D}^{\mathbb{Z}}(r)$; see Definition 3.2.3. A closed-form solution in terms of r for the count of total pixels in a disc is still an open problem. Cappell *et al.* [Cappell and Shaneson (2007)] have reported a result on the number of digital points enclosed in the euclidean disc $\mathcal{D}^{\mathbb{R}}(r)$, which is $\pi r^2 + O(r^{\frac{1}{2}} + \epsilon)$, for some $\epsilon > 0$. In our work, we consider the digital disc, which contains some additional points apart from the points lying inside $\mathcal{D}^{\mathbb{R}}(r)$. Hence, the total number of points constituting $\mathcal{D}^{\mathbb{Z}}(r)$ is $|\mathcal{D}^{\mathbb{Z}}(r)| > \pi r^2$. This result is used by us to derive an asymptotically tight bound on the count of disc absentees, as stated in the following theorem.

Theorem 3.4.3 $|\mathcal{A}^{\mathbb{Z}^2}(r)| = \Theta(r^2)$.

Proof. From Theorem 3.4.2, the total count of circle pixels in $\mathcal{D}^{\mathbb{Z}}(r)$ is given by

$$\sum_{s=0}^{r} |\mathcal{C}^{\mathbb{Z}}(s)| < \sum_{s=0}^{r} \left(8\frac{s}{\sqrt{2}} + 8\right) = 2\sqrt{2}(r^2 + r) + 8(r + 1)$$

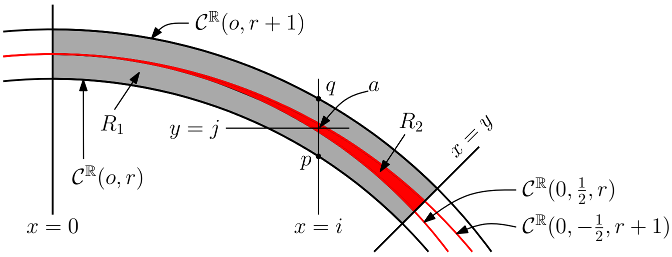


Figure 3.10: Absentee region shown in red.

Hence,

$$\begin{aligned} |\mathcal{A}^{\mathbb{Z}^2}(r)| &= |\mathcal{D}^{\mathbb{Z}}(r)| - \sum_{s=0}^{r} |\mathcal{C}^{\mathbb{Z}}(s)| \\ &> \pi r^2 - 2\sqrt{2}(r^2 + r) - 8(r+1) \\ &= (\pi - 2\sqrt{2})r^2 - (20 + 2\sqrt{2})r + 4 \\ &= \Omega(r^2). \end{aligned} \tag{3.13}$$

Again, as $\mathcal{A}^{\mathbb{Z}^2}(r) \subset \mathcal{D}^{\mathbb{Z}}(r)$, we have $|\mathcal{A}^{\mathbb{Z}^2}(r)| = O(r^2)$, which concludes the proof. □

Both the total count of disc pixels and the count of disc absentees are of the order $\Theta(r^2)$. We show here that their ratio is a constant, which tends to an irrational number as r tends to infinity. The result is stated in the following theorem.

Theorem 3.4.4 *The relative count of absente-pixels w.r.t. disc pixels, for the limiting case as $r \to \infty$, is given by $\alpha_\infty = 1 - \frac{2\sqrt{2}}{\pi}$.*

PROOF. Consider the real region R_1 in Octant 1 between the real circles $\mathcal{C}^{\mathbb{R}}(r)$ and $\mathcal{C}^{\mathbb{R}}(r+1)$, as shown in Figure 3.10. Let $a(i, j)$ be a pixel lying in the region R_1. Let the straight line $x = i$ intersect the circles at $p(i, y_p)$ and $q(i, y_q)$. The pixel a is an absentee if $j - y_p > \frac{1}{2}$ and $y_q - j > \frac{1}{2}$. Consequently, any pixel that lies in the interval $(y_p + \frac{1}{2}, y_q - \frac{1}{2})$ is an absentee. We call this interval as *absentee interval* for the straight line $x = i$ in Octant 1 lying between the circles $\mathcal{C}^{\mathbb{R}}(r)$ and $\mathcal{C}^{\mathbb{R}}(r+1)$. These absentee intervals lie in a region R_2, called the *absentee region* (Figure 3.10), which is defined as the interior of the real disc $\mathcal{D}^{\mathbb{R}}(0, -\frac{1}{2}, r+1) : y = \sqrt{(r+1)^2 - x^2} - \frac{1}{2}$ minus the real disc $\mathcal{D}^{\mathbb{R}}(0, \frac{1}{2}, r) : y = \sqrt{r^2 - x^2} + \frac{1}{2}$ in Octant 1.

To find the ratio of disc absentees to disc pixels, the areas of the euclidean regions are computed instead of counting the pixels in these regions, since the pixels are uniformly distributed in the real region. Let $\alpha_{(r,1)}$ and $\alpha_{(r,2)}$ be the areas of the regions R_1 and R_2 respectively. Then,

$$\alpha_{(r,1)} = \frac{\pi(r+1)^2 - \pi r^2}{8} = \frac{\pi(2r+1)}{8}$$

and

$$\alpha_{(r,2)} = \frac{\pi(2r+1)}{8} - \frac{\frac{\sqrt{4(r+1)^2-\frac{1}{2}}}{\sqrt{2}} + 4(r+1)^2\sin^{-1}\left(\frac{1}{2\sqrt{2}(r+1)}\right) + \frac{\sqrt{4r^2-\frac{1}{2}}}{\sqrt{2}} + 4r^2\sin^{-1}\left(\frac{1}{2\sqrt{2}r}\right)}{8}.$$

A sketch of the derivation is given in the Appendix 2. Thus, the relative count of absentee-pixels w.r.t. disc pixels is given by

$$\begin{aligned} &\alpha_r \\ = \;& \frac{\alpha_{(r,2)}}{\alpha_{(r,1)}} \\ = \;& 1 - \frac{\frac{\sqrt{4(r+1)^2-\frac{1}{2}}}{\sqrt{2}} + 4(r+1)^2\sin^{-1}\left(\frac{1}{2\sqrt{2}(r+1)}\right) + \frac{\sqrt{4r^2-\frac{1}{2}}}{\sqrt{2}} + 4r^2\sin^{-1}\left(\frac{1}{2\sqrt{2}r}\right)}{\pi(2r+1)} \end{aligned} \tag{3.14}$$

Taking limit as $r \to \infty$, we get

$$\alpha_\infty = 1 - \left(\frac{\sqrt{2}}{2\pi} + \frac{\sqrt{2}}{2\pi} + \frac{\sqrt{2}}{2\pi} + \frac{\sqrt{2}}{2\pi}\right) = 1 - \frac{2\sqrt{2}}{\pi}.$$

□

An approximate value of α_∞ is 0.09968368. Note that the limiting value of α_r as $r \to \infty$ may be a rational or an irrational number, as the numerator and the denominator are both irrational. But α_r being a ratio of disc absentees to disc pixels, it is actually a rational number when r is finite. Its irrational value may arise in the limiting case, as a sequence of rational numbers may converge to an irrational number [Erdos (1988)]. Another interesting revelation of Theorem 3.4.4 is the relation of π with α_∞. It should be mentioned here that, like all irrational numbers, π cannot be represented as a simple fraction; instead, it can be represented as an infinite series or as an infinite continued fraction [Borwein and Borwein (1988), Lange (1999)]. Hence, α_∞ may also possess a similar representation in terms of an infinite series or an infinite continued fraction, which can be explored in future to obtain further theoretical results.

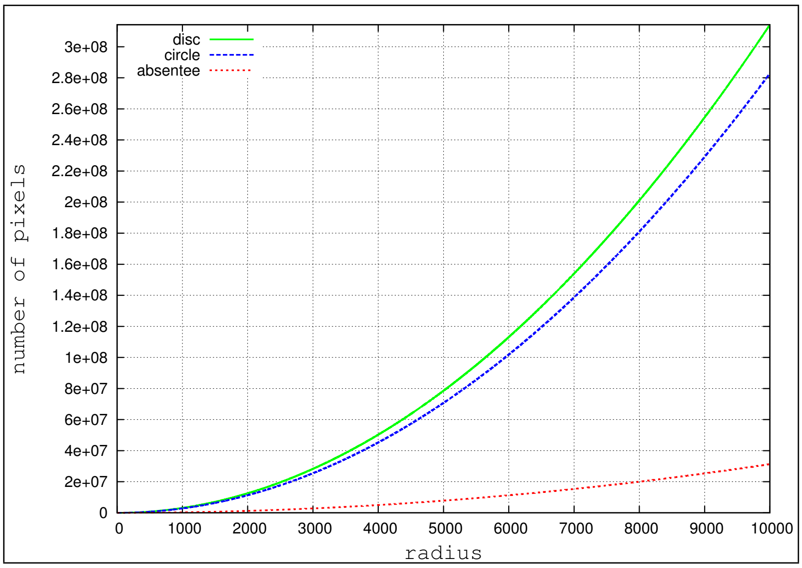


Figure 3.11: Count of circle pixels, absente-pixels, and disc pixels versus radius.

3.5 Results and Conclusion

We have performed an experiment to compute the count of absente-pixels and circle pixels for increasing radius of the digital disc. Table 3.1 shows the count of circle pixels, $\sum_{s=0}^{r} |\mathcal{C}^{\mathbb{Z}}(s)|$, the count of absente-pixels, $|\mathcal{A}^{\mathbb{Z}^2}(r)|$, and the total count of pixels, i.e., $|\mathcal{D}^{\mathbb{Z}}(r)| = \sum_{s=0}^{r} |\mathcal{C}^{\mathbb{Z}}(s)| + |\mathcal{A}^{\mathbb{Z}^2}(r)|$, corresponding to digital discs with radius r varying from 0 to 10000. As derived analytically, we observe that as the radius increases, all the three values increase with a quadratic dependency on r. We plot these pixel counts against radius r, as shown in Figure 3.11. Table 3.2 shows the relative count of absente-pixels (Eqn. 3.14) corresponding to digital discs with radius r varying from 0 to 10000 (no absentee for $r = 0, 1$). As derived analytically, we observe that as the radius increases, the approximate value of α_r tends to the analytically found limit, 0.09968368. The corresponding plot for r varying from 0 to 10000 is shown in Figure 3.12.

We have shown how the absente-pixels can be identified algorithmically in a cover of a digital disc with concentric digital circles. We have also presented several important results for characterizing such absente-pixels, based on their underlying mathematical structures. Our study indicates that the ratio of the absente-pixels to the total number of

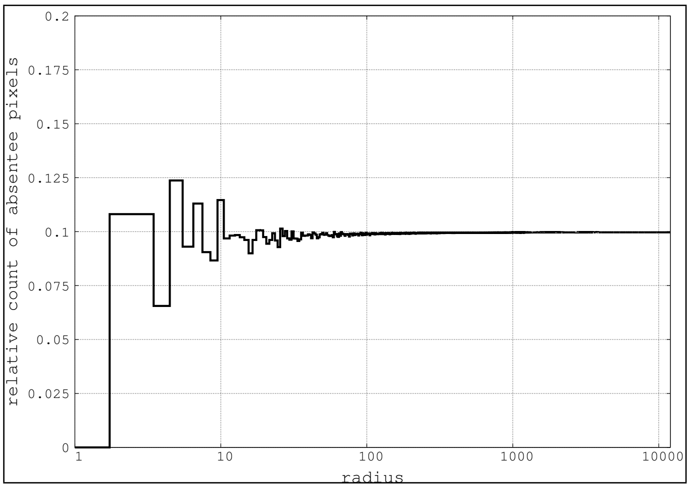


Figure 3.12: Relative count of absente-pixels versus radius (in $\log_{10}$ scale).

disc pixels tends to a constant for large radius. Analysis of these absentees calls for further in-depth study, especially if we want to cover a digital sphere with coaxial digital circles. An asymptotic tight bound for the count of absentees is given, but finding a closed-form solution on the exact count of absentees for a given radius still remains an open problem.

Table 3.1: Number of absente-pixels for radius $r = 0, 1, 2, \ldots, 10000$.

r	$\sum_{s=0}^{r} \lvert\mathcal{C}^{\mathbb{Z}}(s)\rvert$	$\lvert\mathcal{A}^{\mathbb{Z}^2}(r)\rvert$	$\lvert\mathcal{D}^{\mathbb{Z}}(r)\rvert$
0	1	0	1
1	5	0	5
2	17	4	21
3	33	4	37
4	57	4	61
5	85	12	97
6	117	12	129
7	157	20	177
8	201	20	221
9	253	24	277
10	309	40	349
20	1185	128	1313
30	2629	280	2909
40	4637	508	5145
50	7209	796	8005
60	10349	1148	11497
70	14057	1540	15597
80	18325	1996	20321
90	23161	2540	25701
100	28565	3124	31689
200	113701	12552	126253
300	255401	28160	283561
400	453677	50152	503829
500	708517	78304	786821
600	1019929	112728	1132657
700	1387905	153532	1541437
800	1812453	200384	2012837
900	2293569	253696	2547265
1000	2831253	313152	3144405
1100	3425505	378944	3804449
1200	4076325	451028	4527353
1300	4783713	529204	5312917
1400	5547673	613832	6161505
1500	6368197	704572	7072769
1600	7245293	801712	8047005
1700	8178957	905108	9084065
1800	9169193	1014644	10183837
1900	10215993	1130596	11346589
2000	11319361	1252664	12572025
2100	12479297	1381164	13860461
2200	13695805	1515720	15211525
2300	14968881	1656672	16625553
2400	16298525	1803800	18102325
2500	17684737	1957304	19642041
2600	19127517	2117004	21244521
3000	25464325	2818560	28282885
3500	34658129	3836308	38494437
4000	45266141	5010740	50276881
4500	57288373	6341644	63630017
5000	70724817	7829252	78554069
5500	85575473	9473324	95048797
6000	101840341	11274004	113114345
6500	119519425	13231280	132750705
7000	138612721	15345068	153957789
7500	159120233	17615628	176735861
8000	181041957	20042600	201084557
8500	204377897	22626352	227004249
9000	229128049	25366328	254494377
9500	255292413	28263472	283555885
10000	282870989	31316576	314187565

Table 3.2: Relative count of absente-pixels versus radius.

r	α_r	r	α_r	r	α_r	r	α_r
2	0.1904762	17	0.0961338	180	0.0992366	1700	0.0996369
3	0.1081081	18	0.1006524	200	0.0994194	1800	0.0996328
4	0.0655738	19	0.1005868	300	0.0993084	1900	0.0996419
5	0.1237113	20	0.0974867	400	0.0995417	2000	0.0996390
6	0.0930233	30	0.0962530	500	0.0995195	2500	0.0996487
7	0.1129944	40	0.0987366	600	0.0995253	3000	0.0996560
8	0.0904977	50	0.0994379	700	0.0996032	3500	0.0996588
9	0.0866426	60	0.0998521	800	0.0995530	4000	0.0996629
10	0.1146132	70	0.0987369	900	0.0995954	4500	0.0996643
11	0.0968523	80	0.0982235	1000	0.0995902	5000	0.0996670
12	0.0981595	90	0.0988288	1100	0.0996055	6000	0.0996691
13	0.0984183	100	0.0985831	1200	0.0996229	7000	0.0996706
14	0.0974125	120	0.0990764	1300	0.0996071	8000	0.0996725
15	0.0961282	140	0.0993079	1500	0.0996176	9000	0.0996734
16	0.0899408	160	0.0991629	1600	0.0996286	10000	0.0996748

Chapter 4

Covering a Solid Sphere with Consecutive Concentric Spheres in $\mathbb{Z}^3$

4.1 Introduction

Over the last two decades, a multitude of work related to the study of geometric primitives in 2D and 3D digital space have gained momentum because of their numerous applications in computer graphics, image processing, and computer vision. Apart from straight lines and planes Brimkov and Barneva (2002), Brimkov *et al.* (2007, 2008, 2011), Christ *et al.* (2012), Chun *et al.* (2009), Feschet and Reveillès (2006), Fukshansky *et al.* (2012), Kenmochi *et al.* (2008), Woo *et al.* (2008), several theoretical studies on the characterization of digital spheres and hyperspheres have appeared recently. A majority of prior work in 3D digital space focus on the characterization and generation of circles, rings, discs, and circular arcs in the digital plane Andres (1994), Andres and Jacob (1997), Chan and Thomas (1995), Davies (1988a), Doros (1984), Haralick (1974), Nagy (2004), Pal and Bhowmick (2012), Thomas and Chan (1989), Yuen and Feng (1996). For various problems in science and engineering, discrete spheres are often required for simulation of experiments. For example, in [Draine and Flatau (1994), Zubko *et al.* (2010)], discrete spheres are used to test the accuracy of the *discrete dipole approximation* (DDA) for computing scattering and absorption by isolated, homogeneous spheres, as well as by targets consisting of two contiguous spheres. Hence, with the emergence of new paradigms, such as digital calculus [Nakamura and Aizawa (1984)], digital geometry [Klette and Rosenfeld (2004a)], and the theory of words and numbers [Klette and Rosenfeld (2004b), Mignosi (1991)], a proper characterization of a digital sphere is highly needed for better understanding of the effects of discretization in the digital space.

Our work here is concerned with a *closed digital surface* in $\mathbb{Z}^3$ constructed by revolving a 2D-generatrix around an axis such that it approximates a real spherical surface. A set of such spheres with consecutive radii can then be used to construct a solid digital sphere. Note that a solid sphere thus obtained in $\mathbb{Z}^3$ will show a 16-symmetry of rotations in contrast to those constructed in [Andres (1994)], which possess 48-symmetry.

There are some related prior work on finding the lattice points on or inside a real sphere of a given radius Brimkov and Barneva (2008), Chamizo and Cristobal (2012), Chamizo *et al.* (2007), Ewell (2000), Fomenko (2002), Heath-Brown (1999), Magyar (2007), Tsang (2000), and on finding a real sphere that passes through a given set of lattice points [Maehara (2010)]. Most of them are closely related to the determination of lattice points in in circles [Honsberger (1973)], ellipsoids [Chamizo *et al.* (2009), Kühleitner (2000)], or surfaces of revolution [Chamizo (1998)].

The characterization of a discrete analytical hypersphere is reported in [Andres and Jacob (1997)], which leads to an algorithm for generating a discrete hypersphere with arbitrarily located center, radius, and thickness in the real space. It is an extension of the algorithm for generating discrete analytical circles. A further extension is studied in [Fiorio *et al.* (2006)] based on a non-constant thickness function Fiorio and Toutant (2006), but no algorithm for generation of a digital sphere or hypersphere has been proposed. Recently, analytical descriptions of various classes of digital circles, spheres, and some cases of hyperspheres in a morphological framework have been proposed in [Toutant *et al.* (2013)]. Very recently, the notion of *discrete spherical geodesic path* between two voxels lying on a discrete sphere has been introduced in [Biswas and Bhowmick (2014)], and a number-theoretic algorithm has been proposed for construction of such paths in optimal time.

In [Montani and Scopigno (1990)], an algorithm for digitization of a real sphere with integer radius has been proposed. It constructs the sphere as a sequence of contiguous digital circles by using Bresenham's circle drawing algorithm. Such an approach fails to ensure the completeness of the generated digital sphere, since it gives rise to absentee (missing) voxels, as shown in this chapter. The digital sphere generated by our algorithm, on the contrary, does not have any absentee-voxels, since it fixes these absentees based on a digital-geometric characterization.

4.1.1 Our Contribution

The work proposed in this chapter is focused on finding and fixing the absentee-voxels (3D points with integer coordinates) in the digital spherical surface of revolution. Covering such a spherical surface of revolution by coaxial digital circles (with integer radius and integer center) in $\mathbb{Z}^3$ cannot produce the desired completeness of the surface owing to absentee-voxels. Interestingly, the occurrence of absentees in such a cover is possibly a lesser fact. The greater fact is that the absentees occur in multitude—an observation that motivates the requirement of their proper characterization. In this chapter, we present, for the first time, a characterization of these absentees, which subsequently aids in designing an algorithm for generating a complete spherical surface in $\mathbb{Z}^3$. The properties of absentee-voxels were neither studied earlier, nor they follow from any related work Andres and Jacob (1997), Biswas and Bhowmick (2014), Fiorio and Toutant (2006), Montani and Scopigno (1990), Toutant *et al.* (2013). Further, as the proposed technique is based on the revolution of a digital generatrix, it can be used as a generic strategy for the generation of any digital surface of revolution.

The rest of the chapter is organized as follows. In Sec. 4.2, we introduce few definitions and important properties related with digital circles, digital discs, and digital spheres considered in our work. In Sec. 4.3, we derive the necessary and sufficient condition for a voxel to be an absentee in a sphere of revolution. We also prove that the absentee count while covering a digital sphere of radius r by coaxial digital circles—generated by the circular sweep of a digitally circular arc of radius r (digital generatrix)—varies quadratically with r. In Sec. 4.3.1, we characterize the absentee family, and use it in Sec. 4.3.2 for fixing the absentees in our proposed algorithm for generating a complete (i.e., absentee-free) sphere of revolution. In Sec. 4.4, we discuss further about the absentees in covering a solid sphere by union of complete spheres. We show here that these absentees are of two kinds: *absentee line segments* and *absentee circles*. We derive their characterization in Sec. 4.4.1. We use this characterization in Sec. 4.4.2 to show that the absentee counts corresponding to absentee line segments and absentee circles are $\Theta(r^{5/2})$ and $\Theta(r^3)$ respectively. The algorithm for fixing these absentees while generating a complete solid sphere is given in Sec. 4.4.3. Finally, in Sec. 4.5, we present some test results to substantiate our theoretical findings.

4.2 Preliminaries

There exist several definitions of digital circles (and discs, spheres, etc.) in the literature, depending on whether the radius and the center coordinates are real or integer values. Irrespective of these definitions, a digital circle (sphere) is essentially a set of points with integer coordinates, which are called *digital points* or *pixels* (*voxels*) [Klette and Rosenfeld (2004a)]. In this chapter, we consider the *grid intersection digitization* [Klette and Rosenfeld (2004a), Stelldinger (2007)] of a real circle with integer radius and having center with integer coordinates. Such a digitization produces a digital circle, which can be generated by the well-known *midpoint circle algorithm* or the *Bresenham circle algorithm* [Foley *et al.* (1993)], and its definition is as follows.

Definition 4.2.1 (Digital circle) *A digital circle with radius* $r \in \mathbb{Z}^+$ *and center* $o = (0,0)$ *is given by* $\mathcal{C}^{\mathbb{Z}}(r) = \Big\{(i,j) \in \mathbb{Z}^2 : \ \big|\max(|i|,|j|) - \sqrt{r^2 - (\min(|i|,|j|))^2}\big| < \frac{1}{2} \ \Big\}$.

The points in $\mathcal{C}^{\mathbb{Z}}(r)$ are connected in 8-neighborhood. The points defining its *interior* are connected in 4-neighborhood, and hence separated by $\mathcal{C}^{\mathbb{Z}}(r)$ from its *exterior* points, which are also connected in 4-neighborhood [Klette and Rosenfeld (2004a)].

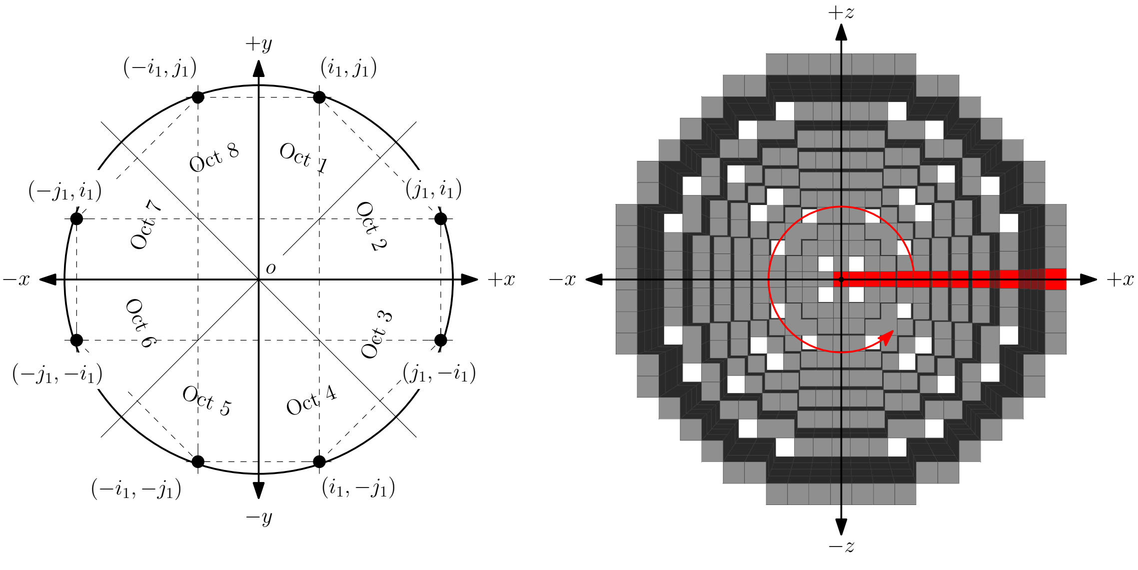
All the results in this chapter are valid for any non-negative integer radius and any center with integer coordinates. So, for sake of simplicity, henceforth we consider the center as o and use the notation $\mathcal{C}^{\mathbb{Z}}(r)$ instead of $\mathcal{C}^{\mathbb{Z}}(o,r)$, where $r \in \mathbb{Z}^+ \cup \{0\}$. We specify it explicitly when the center is not o.

A real point or a pixel (x, y) is said to be lying in Octant 1 if and only if $0 \leqslant x \leqslant y$ (Figure 4.1(a)). We use the notation $\mathcal{C}_1^{\mathbb{Z}}(r)$ to denote Octant 1 of $\mathcal{C}^{\mathbb{Z}}(r)$, and $\mathbb{Z}_1^2$ to denote all points in Octant 1 of $\mathbb{Z}^2$.

Definition 4.2.2 (Digital disc) *A digital disc of radius* r *consists of all digital points in* $\mathcal{C}^{\mathbb{Z}}(r)$ *and its interior, and is given by*

$$\mathcal{D}^{\mathbb{Z}}(r) = \Big\{(i, j_c) \in \mathbb{Z}^2 : \ 0 \le i \cdot i_c \le i_c^2 \ \wedge \Big|\max(|i_c|,|j_c|) - \sqrt{r^2 - (\min(|i_c|,|j_c|))^2}\Big| < \frac{1}{2}\Big\}.$$

Note that in Def. 4.2.2, the condition $0 \le i \cdot i_c \le i_c^2$ relates a disc pixel (i, j_c) to a circle pixel (i_c, j_c), as $0 \le i_c$ implies $0 \le i \le i_c$ and $i_c \le 0$ implies $i_c \le i \le 0$. If we consider the union of all digital circles centered at o and radius in $\{0, 1, 2, \ldots, r\}$, then the resultant



(a) (b)

Figure 4.1: (a) 8-symmetric points $\{(i, j) : \{|i|\} \cup \{|j|\} = \{i_1, j_1\}\}$ in eight respective octants of a digital circle $\mathcal{C}^{\mathbb{Z}}(r)$. (b) $\mathcal{H}^{\mathbb{Z}}_{\cup}(r)$ for $r = 10$, with the $+y$ axis pointing inwards w.r.t. the plane of the paper.

set $\mathcal{D}^{\mathbb{Z}}_{\cup}(r) := \bigcup\limits_{s=1}^{r} \mathcal{C}^{\mathbb{Z}}(s)$ is not identical with the digital disc of radius r. The set $\mathcal{D}^{\mathbb{Z}}_{\cup}(r)$ contains absentee-pixels, as defined below.

Definition 4.2.3 (Disc absentee) *A pixel p is a disc absentee if and only if there exists some $r' \in \{1, 2, \ldots, r\}$ such that p is a point in the interior of $\mathcal{C}^{\mathbb{Z}}(r')$ and in the exterior of $\mathcal{C}^{\mathbb{Z}}(r' - 1)$.*

The above definition implies that if p is any disc absentee, then p does not belong to any digital circle, i.e., $p \in \mathcal{D}^{\mathbb{Z}}(r)$ and $p \notin \mathcal{D}^{\mathbb{Z}}_{\cup}(r)$. Hence, the set of disc absentees is given by $\mathcal{A}^{\mathbb{Z}^2}(r) = \mathcal{D}^{\mathbb{Z}}(r) \smallsetminus \mathcal{D}^{\mathbb{Z}}_{\cup}(r)$. The above definition of disc absentee is used in the following definitions related to *spherical surfaces of revolution* in $\mathbb{Z}^3$. However, henceforth we do not use the term "of revolution" for sake of simplicity. We also drop the term "digital" from "digital circle" or "digital surface".

Let $\mathcal{C}^{\mathbb{Z}}_{1,2}(r)$ denote the first quadrant (comprising the first and the second octants) of $\mathcal{C}^{\mathbb{Z}}(r)$, which is used as the *generatrix*. When we rotate $\mathcal{C}^{\mathbb{Z}}_{1,2}(r)$ in $\mathbb{Z}^3$ about y-axis through

360^o, we get a stack (sequence) of circles representing a *hemisphere*, given by

$$\mathcal{H}_{\cup}^{\mathbb{Z}}(r) = \bigcup_{(i,j)\in\mathcal{C}_{1,2}^{\mathbb{Z}}(r)} \mathcal{C}^{\mathbb{Z}}(c,i),$$

where $c = (0,j,0)$ denotes the center of $\mathcal{C}^{\mathbb{Z}}(c,i)$, as shown in Figure 4.1(b). Each circle $\mathcal{C}^{\mathbb{Z}}(c,i)$ in this stack is generated by rotating a pixel $(i,j) \in \mathcal{C}_{1,2}^{\mathbb{Z}}(r)$ in $\mathbb{Z}^3$ about y-axis. The previous circle in the stack is either $\mathcal{C}^{\mathbb{Z}}(c',i-1)$ or $\mathcal{C}^{\mathbb{Z}}(c'',i)$, where $c' = (0,j',0)$ with $j' \in \{j, j+1\}$, and $c'' = (0, j+1, 0)$. There is no absentee between $\mathcal{C}^{\mathbb{Z}}(c,i)$ and $\mathcal{C}^{\mathbb{Z}}(c'',i)$, as they have the same radius. But as the radii of $\mathcal{C}^{\mathbb{Z}}(c,i)$ and $\mathcal{C}^{\mathbb{Z}}(c',i-1)$ differ by unity, there would be absentees (Definition 4.2.4) between them in $\mathcal{H}_{\cup}^{\mathbb{Z}}(r)$. Each such absentee p would lie on the plane of $\mathcal{C}^{\mathbb{Z}}(c,i)$ in the exterior of $\mathcal{C}^{\mathbb{Z}}(c,i-1)$, since p did not appear in the part of $\mathcal{H}_{\cup}^{\mathbb{Z}}(r)$ constructed up to $\mathcal{C}^{\mathbb{Z}}(c',i-1)$ and appeared only after constructing $\mathcal{C}^{\mathbb{Z}}(c,i)$. Hence, we have the following definition.

Definition 4.2.4 (Sphere absentee) *A voxel p is a sphere absentee lying on the plane $y = j$ if and only if there exist two consecutive points (i,j) and $(i-1,j')$ in $\mathcal{C}_{1,2}^{\mathbb{Z}}(r)$, $j' \in \{j, j+1\}$, such that p lies in the interior of $\mathcal{C}^{\mathbb{Z}}(c,i)$ and in the exterior of $\mathcal{C}^{\mathbb{Z}}(c,i-1)$, where $c = (0,j,0)$.*

On inclusion of the sphere absentees (lying above zx-plane) with $\mathcal{H}_{\cup}^{\mathbb{Z}}(r)$, we get the *complete hemisphere*, namely $\mathcal{H}^{\mathbb{Z}}(r)$. On taking $\mathcal{H}_{\cup}^{\mathbb{Z}}(r)$ and its reflection on zx-plane, we get the *sphere*, namely $\mathcal{S}_{\cup}^{\mathbb{Z}}(r)$. Similarly, the union of $\mathcal{H}^{\mathbb{Z}}(r)$ and its reflection on zx-plane gives the *complete sphere* in $\mathbb{Z}^3$. Let $\mathcal{A}^{\mathbb{Z}^3}(r)$ be the set of sphere absentees. The number of points in $\mathcal{A}^{\mathbb{Z}^3}(r)$ is double the absentee count in $\mathcal{H}_{\cup}^{\mathbb{Z}}(r)$. We have the following definitions on spheres and their absentees.

Definition 4.2.5 (Complete sphere) *A complete (hollow) sphere of radius r is given by $\mathcal{S}^{\mathbb{Z}}(r) = \mathcal{S}_{\cup}^{\mathbb{Z}}(r) \cup \mathcal{A}^{\mathbb{Z}^3}(r)$.*

Definition 4.2.6 (Complete solid sphere) *A complete solid sphere $\mathbf{S}^{\mathbb{Z}}(r)$ of radius r is given by the union of $\mathcal{S}^{\mathbb{Z}}(r)$ and voxels lying inside $\mathcal{S}^{\mathbb{Z}}(r)$.*

Definition 4.2.7 (Solid sphere absentee) *A voxel p is a solid sphere absentee if and only if $p \in \mathbf{S}^{\mathbb{Z}}(r) \smallsetminus \mathbf{S}_{\cup}^{\mathbb{Z}}(r)$, where $\mathbf{S}_{\cup}^{\mathbb{Z}}(r) = \bigcup\limits_{r'=0}^{r} \mathcal{S}^{\mathbb{Z}}(r')$.*

4.2.1 Previous Results

We need the following results from [Bera *et al.* (2013)] to count and fix the absentees in the surface of revolution.

Theorem 4.2.1 *The total count of disc absentees lying in* $\mathcal{D}^{\mathbb{Z}}_{\cup}(r)$ *is given by*

$$|\mathcal{A}^{\mathbb{Z}^2}(r)| = 8 \sum_{k=0}^{m_r-1} |\mathcal{A}^{\mathbb{Z}^2}_k(r)|,$$

where $|\mathcal{A}^{\mathbb{Z}^2}_k(r)| = \left\lceil \sqrt{(2k+1)r - k(k+1)} \right\rceil - \left\lceil 2k+1+\frac{1}{2}\sqrt{(8k^2+4k+1)} \right\rceil$
and $m_r = r - \lceil r/\sqrt{2} \rceil + 1$.

Theorem 4.2.2 $|\mathcal{A}^{\mathbb{Z}^2}(r)| = \Theta(r^2)$.

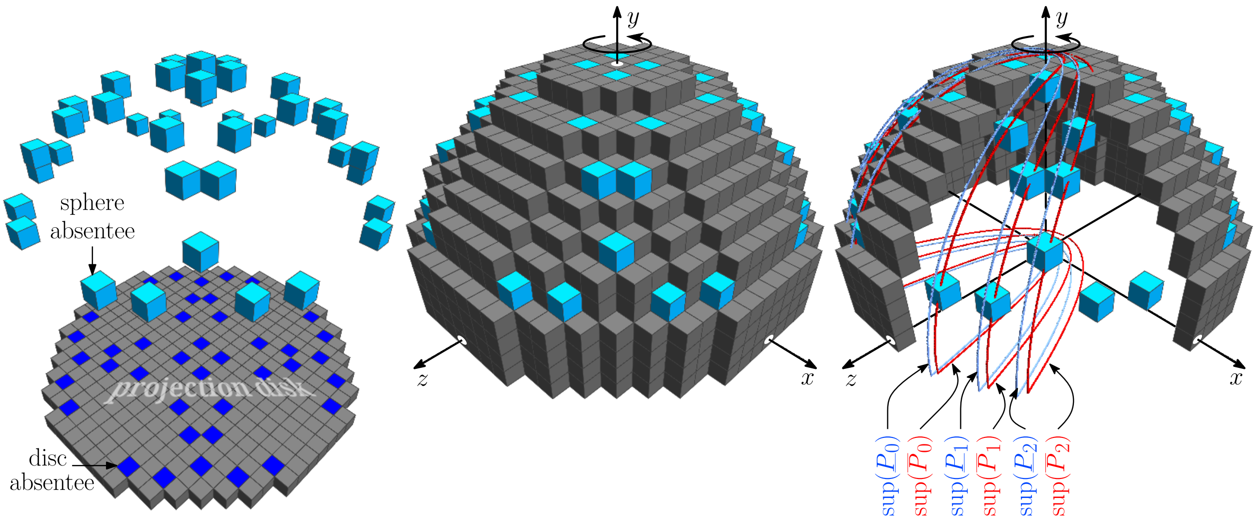
4.3 Absentees in a Digital Sphere

As mentioned earlier in Sec. 4.2, the hemisphere and the sphere have absentee-voxels, which can be characterized based on their unique correspondence with the absentee-pixels of $\mathcal{D}^{\mathbb{Z}}_{\cup}(r)$. To establish this correspondence, we consider two consecutive pixels $p_i(x_i, y_i)$ and $p_{i+1}(x_{i+1}, y_{i+1})$ of the generating curve $\mathcal{C}^{\mathbb{Z}}_{1,2}(r)$ corresponding to $\mathcal{H}^{\mathbb{Z}}_{\cup}(r)$. We have three possible cases as follows.

1. $(x_{i+1}, y_{i+1}) = (x_i + 1, y_i)$ (Octant 1)
2. $(x_{i+1}, y_{i+1}) = (x_i + 1, y_i - 1)$ (Octant 1 or Octant 2)
3. $(x_{i+1}, y_{i+1}) = (x_i, y_i - 1)$ (Octant 2)

For Case 1, we get two concentric circles of radii differing by unity and lying on the same plane; the radii of the circles corresponding to p_i and p_{i+1} are x_i and $(x_{i+1} =)x_i + 1$. Hence, for Case 1, the absentee-voxels between two consecutive circles easily correspond to the absentee-pixels between $\mathcal{C}^{\mathbb{Z}}(o, x_i)$ and $\mathcal{C}^{\mathbb{Z}}(o, x_i + 1)$.

For Case 2, the circle generated by p_{i+1} has radius $x_{i+1} = x_i + 1$ and its plane lies one voxel apart w.r.t. the plane of the circle generated by p_i. Hence, if these two circles are projected on zx-plane, then the absentee-pixels lying between the projected circles have a correspondence with the absentee-voxels between the original circles.



(a) (b) (c)

Figure 4.2: (a) One-to-one correspondence for $r = 10$ between absentee-voxels (shown in red) in $\mathcal{H}^{\mathbb{Z}}_{\cup}(r)$ and absentee-pixels (shown in blue) in $\mathcal{D}^{\mathbb{Z}}_{\cup}(r)$. (b) Hemisphere of $r = 10$ after fixing absentees. (c) Parabolic surfaces of translation, produced by translating parabolas given in Eqn. 4.5 and Eqn. 4.6, $h = 0, 1, 2$, along y-axis.

For Case 3, we do not have an absentee, as the circles generated by p_i and p_{i+1} have the same radius ($x_i = x_{i+1}$).

Hence, the count of absentee-voxels in $\mathcal{H}^{\mathbb{Z}}_{\cup}(r)$ is same as the count of absentee-pixels in $\mathcal{D}^{\mathbb{Z}}_{\cup}(r)$. However, it may be noted that the count of voxels present in $\mathcal{H}^{\mathbb{Z}}_{\cup}(r)$ would be greater than the count of pixels present in $\mathcal{D}^{\mathbb{Z}}_{\cup}(r)$, since for each circle of a particular radius $r' \in \{0, 1, 2, \ldots, r\}$ in $\mathcal{D}^{\mathbb{Z}}_{\cup}(r)$, there would be one or more circles of radius r' (in succession) in $\mathcal{H}^{\mathbb{Z}}_{\cup}(r)$. We have the following lemma on the correspondence of absentee set in $\mathcal{H}^{\mathbb{Z}}_{\cup}(r)$ with that in $\mathcal{D}^{\mathbb{Z}}_{\cup}(r)$.

Lemma 4.3.1 *If (i, j, k) is an absentee-voxels in $\mathcal{H}^{\mathbb{Z}}_{\cup}(r)$, then (i, k) is an absentee-pixel in $\mathcal{D}^{\mathbb{Z}}_{\cup}(r)$. Conversely, if (i, k) is an absentee-pixel in $\mathcal{D}^{\mathbb{Z}}_{\cup}(r)$, then there exists a unique positive integer j such that (i, j, k) is an absentee-voxels in $\mathcal{H}^{\mathbb{Z}}_{\cup}(r)$.*

The above one-to-one correspondence between the absentees in the hemispherical surface for radius $r = 10$ and the absentees related to the disc of radius $r = 10$ is shown in Figure 4.2. This one-to-one correspondence between the absentee set in $\mathcal{H}^{\mathbb{Z}}_{\cup}(r)$ and that in $\mathcal{D}^{\mathbb{Z}}_{\cup}(r)$ leads to the following theorem.

Theorem 4.3.2 *The total count of absentee-voxels in* $\mathcal{H}^{\mathbb{Z}}_{\mathsf{U}}(r)$ *is* $|\mathcal{A}^{\mathbb{Z}^2}(r)| = \Theta(r^2)$*.*

Proof. Follows from Lemma 4.3.1 and Theorem 4.2.2. □

On taking the reflection of $\mathcal{H}^{\mathbb{Z}}(r)$ about the zx-plane, we get the complementary hemisphere, namely $\mathcal{H}'^{\mathbb{Z}}(r)$. The set $\mathcal{H}^{\mathbb{Z}}(r) \cup \mathcal{H}'^{\mathbb{Z}}(r)$ is the sphere, $\mathcal{S}^{\mathbb{Z}}(r)$, corresponding to which we get double the count of absentee-voxels compared to that in $\mathcal{H}^{\mathbb{Z}}(r)$. Hence, we have the following theorem.

Theorem 4.3.3 *The total count of absentee-voxels lying on* $\mathcal{S}^{\mathbb{Z}}_{\mathsf{U}}(r)$ *is given by*

$$|\mathcal{A}^{\mathbb{Z}^3}(r)| = 2|\mathcal{A}^{\mathbb{Z}^2}(r)| = 16 \sum_{k=0}^{m_r-1} |\mathcal{A}^{\mathbb{Z}^2}_k(r)| = \Theta(r^2).$$

Proof. Follows from Theorem 4.2.1 and Theorem 4.3.2. □

4.3.1 Characterizing the Absentee Family

We use the following lemmas from [Bera *et al.* (2013), Bhowmick and Bhattacharya (2008)] for deriving the necessary and sufficient conditions to decide whether a given voxel is an absentee or not. We use $\mathcal{C}^{\mathbb{Z}}_{1}(r)$ to denote Octant 1 of $\mathcal{C}^{\mathbb{Z}}(r)$.

Lemma 4.3.4 (circle pixel [Bhowmick and Bhattacharya (2008)]) *The squares of abscissae of the pixels with* $z = k$ *in* $\mathcal{C}^{\mathbb{Z}}_{1}(r')$ *drawn on zx-plane lie in the interval* $I^{(r')}_{r'-k} := \left[u^{(r')}_{r'-k}, v^{(r')}_{r'-k}\right)$*, where* $u^{(r')}_{r'-k} = r'^2 - k^2 - k$ *and* $v^{(r')}_{r'-k} = r'^2 - k^2 + k$*.*

Lemma 4.3.5 (absentee [Bera *et al.* (2013)]) *A point* $(i, 0, k) \in \mathbb{Z}^3$ *is an absentee on the zx-plane if and only if* i^2 *lies in the integer interval* $J^{(r')}_{r'-k} := \left[v^{(r')}_{r'-k}, u^{(r'+1)}_{r'+1-k}\right)$ *for some* $r' \in \mathbb{Z}^+$*.*

We have now the following theorem on the necessity and sufficiency for an absentee-voxels in $\mathcal{S}^{\mathbb{Z}}_{\mathsf{U}}(r)$.

Theorem 4.3.6 *A voxel* $p(i, j, k)$ *is an absentee if and only if* $i^2 \in J^{(r')}_{r'-k}$ *for some* $r' \in \mathbb{Z}^+$ *and* $r'^2 \in I^{(r)}_{r-j}$*.*

Proof. Lemma 4.3.1 implies that when $p(i, j, k)$ is an absentee-voxels in $\mathcal{H}^{\mathbb{Z}}_{\mathsf{U}}(r)$, then its projection pixel $p'(i, k)$ on zx-plane is absentee-pixel in $\mathcal{D}^{\mathbb{Z}}_{\mathsf{U}}(r)$. Hence, by Lemma 4.3.5, i^2

lies in $J^{(r')}_{r'-k}$ for some $r' \in \mathbb{Z}^+$. What now remains to check is the condition for y-coordinate of p. Observe that there exists a circle $\mathcal{C}^{\mathbb{Z}}(c, r')$ centered at $c = (0, j, 0)$ on the hemisphere such that the the projection of $\mathcal{C}^{\mathbb{Z}}(c, r')$ on zx-plane is the circle of radius r' in $\mathcal{D}^{\mathbb{Z}}_{\cup}(r)$. Again $p(i, j, k)$ and $\mathcal{C}^{\mathbb{Z}}(c, r')$ lie on the same plane, i.e., $y = j$. Hence, the pixel (r', j) must lie on the generating circular arc, $\mathcal{C}^{\mathbb{Z}}_{1,2}(r)$, and so by Lemma 4.3.4, we have $r'^2 \in I^{(r)}_{r-j}$.

Conversely, if $i^2 \in J^{(r')}_{r'-k}$, then $p \notin \mathcal{H}^{\mathbb{Z}}_{\cup}(r)$; and if $r'^2 \in I^{(r)}_{r-j}$ for some $r' \in \{0, 1, 2, \ldots, r\}$, then $p \in \mathcal{H}^{\mathbb{Z}}(r)$, wherefore p is an absentee. □

An example of absentee-voxels is $(2, 9, 4)$ in hemisphere of radius $r = 10$ (Figure 4.2), since for $k = 4$, we have $r' = 4$ for which $v^{(r')}_{r'-k} = r'^2 - k^2 + k = 16 - 16 + 4 = 4$, $u^{(r'+1)}_{r'+1-k} = (r' + 1)^2 - k^2 - k = 25 - 16 - 4 = 5$, thus giving $J^{(4)}_0 = [4, 5) = [4, 4]$ in which lies the square number $4 = i^2$ and $u^{(r)}_{r-j} = r^2 - j^2 - j = 10^2 - 9^2 - 9 = 10$, $v^{(r)}_{r-j} = r^2 - j^2 + j = 10^2 - 9^2 + 9 = 28$, thus giving $I^{(r)}_{r-j} = [10, 28)$ which contains $r'^2 = 16$.

On the contrary, $(3, 9, 4)$ is not an absentee-voxels, as for $k = 4$, there is no such r' for which $J^{(r')}_{r'-4}$ contains 3^2; in fact, for $k = 4$, we get the interval $I^{(5)}_{5-4} = [5^2 - 4^2 - 4, 5^2 - 4^4 + 4) = [5, 12]$ with $r' = 5$, which contains 3^2, thereby making $(3, 9, 4)$ a point on hemisphere of radius $r = 10$ at the plane $y = 9$.

To characterize the absentees as a whole, we use Lemma 4.3.5 for the expanded form of (the lower and the upper limits of) $J^{(r')}_{r'-k}$. We replace r' by $k+h$ and $r'+1$ by $k+(h+1)$, where the $h(\geq 0)$*th run of pixels* in $\mathcal{C}^{\mathbb{Z}}_1(r')$ drawn on zx-plane has $z = k$ [Bhowmick and Bhattacharya (2008)]. Thus,

$$\begin{aligned} v^{(r')}_{r'-k} &= (2h + 1)k + h^2, \\ u^{(r'+1)}_{r'+1-k} &= (2h + 1)k + (h + 1)^2. \end{aligned} \tag{4.1}$$

Hence, if $p(i, 0, k)$ is a point in Octant 1 and lies on h^{th} run of $\mathcal{C}^{\mathbb{Z}}_1(r')$, then

$$i^2 < (2h + 1)k + h^2; \tag{4.2}$$

and if $p(i, 0, k)$ is a point in Octant 1 and lies left of the $(h + 1)^{th}$ run of $\mathcal{C}^{\mathbb{Z}}_1(r' + 1)$, then

$$i^2 < (2h + 1)k + (h + 1)^2. \tag{4.3}$$

Equations 4.2 and 4.3 correspond to two parabolic regions in the real (zx-)plane on replacing i and k by x and z, respectively, h being considered as a constant. These *open* *parabolic regions* are given by

$$\begin{aligned} \underline{P}_{h,1} &: x^2 < (2h+1)z + h^2, \\ \overline{P}_{h,1} &: x^2 < (2h+1)z + (h+1)^2. \end{aligned} \tag{4.4}$$

The respective suprema of these two regions are given by two parabolas, namely

$$\sup(\underline{P}_{h,1}) : x^2 = (2h+1)z + h^2 \tag{4.5}$$

and

$$\sup(\overline{P}_{h,1}) : x^2 = (2h+1)z + (h+1)^2. \tag{4.6}$$

In 3D space, these two suprema correspond to two *parabolic surfaces of translation*, produced by translating $\sup(\underline{P}_{h,1})$ and $\sup(\overline{P}_{h,1})$ along y-axis, as shown in Figure 4.2c. Evidently, the absentees of $\mathcal{H}^{\mathbb{Z}}_{\cup}(r)$ in Octant 1 and Octant 8 lie in the *half-open 3D parabolic region* given by $P_h := \overline{P}_{h,1} \smallsetminus \underline{P}_{h,1}$ for a given pair of k and h, i.e., for a given (r', k)-pair. The family of all the half-open 3D parabolic regions, $P_0, P_1, P_2, \ldots$, thus contains all the absentees $\mathcal{H}^{\mathbb{Z}}_{\cup}(r)$ in Octant 1 and Octant 8, as stated in the following theorem.

Theorem 4.3.7 *All the absentees of* $\mathcal{H}^{\mathbb{Z}}_{\cup}(r)$ *in Octant 1 and Octant 8 lie in*

$$\mathcal{F} := \left\{ P_h \cap \mathbb{Z}^3_1 : h = 0, 1, 2, \ldots \right\}.$$

PROOF. Follows from Theorem 4.3.6 and Eqn. 4.4. □

4.3.2 Fixing the Absentee-voxels

Algorithm 2 (AVH) shows the steps for fixing the absentee-voxels corresponding to the hemisphere $\mathcal{H}^{\mathbb{Z}}_{\cup}(r)$ having radius r. The generating curve, which is an input to this algorithm, is the circular arc, $\mathcal{C}^{\mathbb{Z}}_{1,2}(r)$. This circular arc is a (ordered) sequence of points, $\{p_t(i_t, j_t, 0) \in \mathbb{Z}^3 : t = 1, 2, \ldots, n_r\}$, whose first point is $p_1(0, r, 0)$ and last point is $p_{n_r}(r, 0, 0)$. The point p_{t+1} can have i_{t+1} either same as i_t of the previous point p_t or greater than i_t by unity. For the former case, there is no absentee between the two circles generated by p_t and p_{t+1}. For the latter, the absentees are computed by invoking the procedure ACC, as shown in Step 4 of Algorithm 2.

The procedure ACC finds the absentee-voxels between two concentric circles, $\mathcal{C}^{\mathbb{Z}}_{y=j}(c, i)$ and $\mathcal{C}^{\mathbb{Z}}_{y=j}(c, i+1)$ of radii i and $i+1$, each centered at $(0, j, 0)$ on $y = j$ plane. The set

Algorithm 2: (AVH) Fixing absentee-voxels in the hemisphere

```
   Input: Generating circular arc, $\mathcal{C}^{\mathbb{Z}}_{1,2}(r) := \{p_1, p_2, \ldots, p_{n_r}\}$
   Output: Absentee-voxels in $\mathcal{H}^{\mathbb{Z}}_{\cup}(r)$
1  $\mathcal{A}^{\mathbb{Z}^3}(r) \leftarrow \emptyset$
2  for $t = 1, 2, \ldots, n_r - 1$ do
3      if $i_{t+1} > i_t$ then
4          $\mathcal{A}^{\mathbb{Z}^3}(r) \leftarrow \mathcal{A}^{\mathbb{Z}^3}(r) \cup \mathrm{ACC}(i_t, j_t)$
5  return $\mathcal{A}^{\mathbb{Z}^3}(r)$
```

Procedure ACC(i_t, j_t)

```
1  $A \leftarrow \emptyset, r \leftarrow i_t$
2  int $i \leftarrow 0, k \leftarrow r, s \leftarrow 0, w \leftarrow r - 1$
3  int $l \leftarrow 2w$
4  while $k \geq i$ do
5      repeat
6          $s \leftarrow s + 2i + 1$
7          $i \leftarrow i + 1$
8      until $s \leq w$;
9      if $i^2 \in J^{(r)}_{r-k}$ and $k \geq i$ then
10         $A \leftarrow A \cup \{(i', j_t, k') : \{|i'|\} \cup \{|k'|\} = \{i, k\}\}$
11     $w \leftarrow w + l,\ l \leftarrow l - 2,\ k \leftarrow k - 1$
12 return $A$
```

of all absentees between these two circles is denoted by A. As an absentee lies just after the end of a *voxel-run* corresponding to the interval $I^{(r)}_{r-j}$ (Lemma 4.3.4), the procedure ACC first computes the voxel-run in the plane $y = j$ (Steps 5–8). Then, in Step 9, it determines whether the next voxel is an absentee in Octant 1, using Lemma 4.3.5. For each absentee-voxels in Octant 1, the absentees in all other octants are included in A, as shown in Step 10. Figure 4.2(a) shows the hemisphere for $r = 10$, whose absentees (shown in red) have been fixed by Algorithm 2.

4.4 Absentees in a Solid Sphere

As mentioned earlier in Sec. 4.2, the absentee-voxels in a solid sphere $\mathbf{S}^{\mathbb{Z}}_{\cup}(r)$ can also be characterized using the set of disc absentees, $\mathcal{A}^{\mathbb{Z}^2}(r)$. The set of voxels defining $\mathbf{S}^{\mathbb{Z}}_{\cup}(r)$ is given by the union of the voxel sets corresponding to the complete spheres of radii $0, 1, 2, \ldots, r$ (Definition 4.2.7).

To find the absentees in $\mathbf{S}^{\mathbb{Z}}_{\cup}(r)$, we consider its lower (or upper) hemisphere, $\mathbf{H}^{\mathbb{Z}}_{\cup}(r)$. Three-fourth of the upper hemisphere and the entire lower hemisphere of $\mathbf{S}^{\mathbb{Z}}_{\cup}(r)$ are shown in Figure 4.3a. Observe that the set of voxels of $\mathbf{H}^{\mathbb{Z}}_{\cup}(r)$ lying in the 1st quadrant of xy-, yz-, or zx-plane is given by the union of voxels comprising those arcs of the complete spheres which lie in the 1st quadrant of the concerned plane. Hence, the above set of voxels is same as the subset of $\mathcal{D}^{\mathbb{Z}}_{\cup}(r)$ lying in this quadrant, or, the absentee-voxels in this quadrant are in one-to-one correspondence with the disc absentees in $\mathcal{D}^{\mathbb{Z}}_{\cup}(r)$ (Figure 4.2a). The absentees in this quadrant are, however, characterized depending on the coordinate plane, as follows.

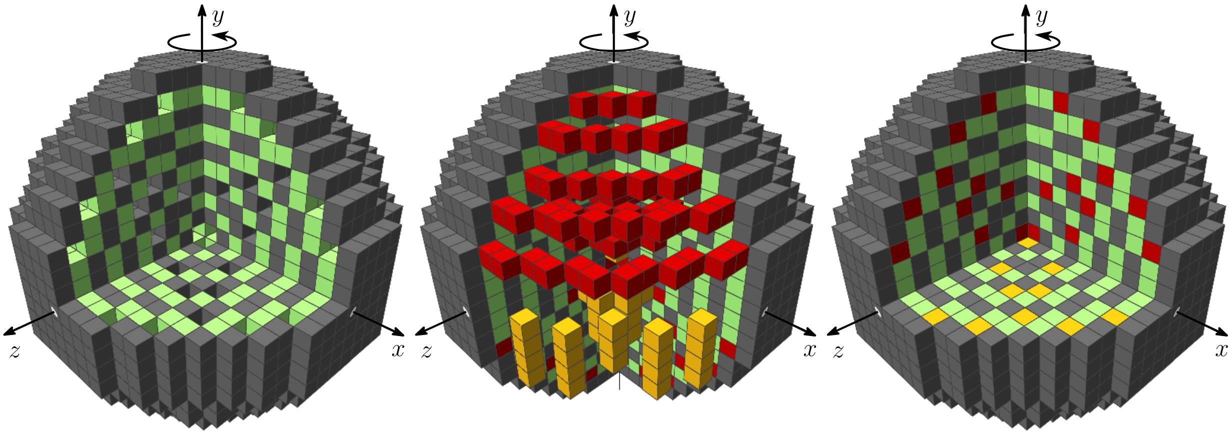
(AL) For each absentee $p(i, 0, k)$ in the 1st quadrant of zx-plane, there are absentees in $\mathbf{H}^{\mathbb{Z}}_{\cup}(r)$, which comprise an *absentee line segment*, given by $\mathbf{L}^{\mathbb{Z}^3}_{(i,k)} = \{(i, j, k) : j' < j \leq 0 \wedge (i, j', k) \in \mathbf{H}^{\mathbb{Z}}_{\cup}(r)\}$. These absentee line segments are shown in yellow in the lower hemisphere in Figure 4.3b.

(AC) For each absentee $p(i, j, 0)$ in the 1st quadrant of xy-plane, there are absentees in $\mathbf{H}^{\mathbb{Z}}_{\cup}(r)$, which comprise an *absentee circle*, given by $\mathbf{C}^{\mathbb{Z}^3}_{(i,j)} = \{(i', j, k') : (i', j, k') \in \mathcal{C}^{\mathbb{Z}}_{y=j}(c, i) \wedge c = (0, j, 0)\}$. These absentee circles, shown in red in the upper hemisphere in Figure 4.3b, pass through the absentees in the 1st quadrant of yz-plane.

4.4.1 Characterizing the Absentee Family

We characterize here the family of absentee-voxels comprising the absentee line segments that comprising the absentee circles. For this, we need the following theorems on the necessity and sufficiency for an absentee belonging to an absentee line segment or an absentee circle in $\mathbf{H}^{\mathbb{Z}}_{\cup}(r)$.

Theorem 4.4.1 *A voxel* $p(i, j, k)$ *belongs to an absentee line segment if and only if* $i^2 \in J^{(r')}_{r'-k}$ *for some* $r' \in \mathbb{Z}^+$ *and* $0 \leq j \leq \lfloor \sqrt{r'} \rfloor + 1$.



(a) (b) (c)

Figure 4.3: Covering a solid sphere ($r = 10$) by concentric complete spheres. (a) Concentric complete spheres for $r = 0, 1, \ldots, 10$. (b) Absentee line segments (yellow) in the lower hemisphere and absentee circles (red) in the upper hemisphere. (c) Complete solid sphere after fixing the absentee line segments and absentee circles.

Proof. From (AL), if $p(i,j,k)$ belongs to an absentee line segment $\mathbf{L}^{\mathbb{Z}^3}_{(i,k)}$, then $p(i,0,k)$ is a disc absentee on zx-plane. Hence, by Lemma 4.3.5, $i^2 \in J^{(r')}_{r'-k}$ for some $r' \in \mathbb{Z}^+$. Further, $\mathbf{L}^{\mathbb{Z}^3}_{(i,k)}$ lies between two complete hemispheres, namely $\mathbf{H}^{\mathbb{Z}}_{\cup}(r')$ and $\mathbf{H}^{\mathbb{Z}}_{\cup}(r'+1)$, where $r' = \lfloor i^2 + k^2 - k \rfloor$. In particular, $\mathbf{L}^{\mathbb{Z}^3}_{(i,k)}$ lies between the surface of revolution generated by the topmost run of (generatrix) $\mathcal{C}^{\mathbb{Z}}_{34}(r'+1)$ and that generated by the topmost run of $\mathcal{C}^{\mathbb{Z}}_{34}(r')$. Hence, the value of j can be at most the length of the topmost run of $\mathcal{C}^{\mathbb{Z}}_{34}(r'+1)$, which is $\lfloor \sqrt{r'} \rfloor + 1$ by Lemma 4.3.4.

Conversely, if $i^2 \in J^{(r')}_{r'-k}$ for some $r' \in \mathbb{Z}^+$, then by Lemma 4.3.5, $p(i,0,k)$ is a disc absentee on zx-plane. Hence, if $0 \leq j \leq \lfloor \sqrt{r'} \rfloor + 1$, then from (AL), (i,j,k) belongs to $\mathbf{L}^{\mathbb{Z}^3}_{(i,k)}$ that lies between the surfaces of revolution generated by $\mathcal{C}^{\mathbb{Z}}_{34}(r'+1)$ and $\mathcal{C}^{\mathbb{Z}}_{34}(r')$. □

Theorem 4.4.2 *A voxel $p(i,j,k)$ belongs to an absentee circle if and only if $i^2 \in I^{(r')}_{r'-k}$ for some $r' \in \mathbb{Z}^+$ such that $r'^2 \in J^{(r'')}_{r''-j}$ for some $r'' \in \mathbb{Z}^+$.*

Proof. From (AC), if $p(i,j,k)$ belongs to an absentee circle, then that absentee circle $\mathcal{C}^{\mathbb{Z}}(r')$ lies on the plane $y = j$, where (r', j) is a disc absentee in $\mathcal{D}^{\mathbb{Z}}_{\cup}(r)$ in xy-plane. Hence, by Lemma 4.3.5, $r'^2 \in J^{(r'')}_{r''-j}$ for some $r'' \in \mathbb{Z}^+$. Further, since $p \in \mathcal{C}^{\mathbb{Z}}(r')$, we get $i^2 \in I^{(r')}_{r'-k}$ by Lemma 4.3.4.

Conversely, if $r'^2 \in J^{(r'')}_{r''-j}$ for some $r'' \in \mathbb{Z}^+$, then by Lemma 4.3.5, (r', j) is a disc absentee on xy-plane, and so $\mathcal{C}^{\mathbb{Z}}(r')$ is an absentee circle. Hence, by Lemma 4.3.4, if $i^2 \in I^{(r')}_{r'-k}$, then $p \in \mathcal{C}^{\mathbb{Z}}(r')$, or, p belongs to an absentee circle. □

The absentee circles are characterized based on locations of their corresponding disc absentees on (real) xy-plane. The digital disc has eight octants. All the absentee circles corresponding to the disc absentees in Octant 1 include the disc absentees in Octant 8. Reflection of these absentee circles simply gives all the absentee circles corresponding to the disc absentees in Octant 4 (and 5), and hence the characterization of absentee circles corresponding to Octant 4 is very much similar to that corresponding to Octant 1. But the characterization of absentee circles corresponding to Octant 2 (and 7) is different and it can be used to obtain the characterization corresponding to Octant 3 (and 6) also. Hence, following are two theorems on characterization of absentee circles—one for Octant 1 and another for Octant 2.

Theorem 4.4.3 *All the absentee circles of $\mathbf{S}^{\mathbb{Z}}_{\cup}(r)$ corresponding to Octant 1 lie in*

$$\mathcal{F}_1 := \left\{\mathbf{P}_{h,1} \cap \mathbb{Z}^3_1 : h = 0, 1, 2, \ldots\right\} \tag{4.7}$$

where,

$$\mathbf{P}_{h,1} = \overline{\mathbf{P}}_{h,1} \smallsetminus \underline{\mathbf{P}}_{h,1} \tag{4.8}$$

such that

$$\begin{aligned} \underline{\mathbf{P}}_{h,1} &: x^2 + z^2 < (2h+1)y + h^2, \\ \overline{\mathbf{P}}_{h,1} &: x^2 + z^2 < (2h+1)y + (h+1)^2. \end{aligned} \tag{4.9}$$

Proof. We use Theorem 4.4.2 for the expanded forms of the lower and the upper limits of $J^{(r'')}_{r''-j}$. Using Lemma 4.3.5 and replacing $r'' - j$ by h, we get

$$\begin{aligned} v^{(r'')}_{r''-j} &= (2h+1)j + h^2, \\ u^{(r''+1)}_{r''+1-j} &= (2h+1)j + (h+1)^2. \end{aligned} \tag{4.10}$$

Hence, if $p(i, j = r'' - h, 0)$ is a point on or inside $\mathcal{C}^{\mathbb{Z}}_1(r'')$ but strictly inside $\mathcal{C}^{\mathbb{Z}}_1(r''+1)$, then

$$\begin{aligned} i^2 &< (2h+1)j + h^2, \\ i^2 &< (2h+1)j + (h+1)^2. \end{aligned} \tag{4.11}$$

Equation 4.11 corresponds to two *open parabolic regions* on the xy-plane, on replacing i and j by x and y respectively, h being considered as a constant. These open parabolic regions are given by

$$\begin{aligned} \underline{P}_{h,1} &: x^2 < (2h+1)y + h^2, \\ \overline{P}_{h,1} &: x^2 < (2h+1)y + (h+1)^2. \end{aligned} \tag{4.12}$$

The respective suprema of these two regions are given by two parabolas, namely $\sup(\underline{P}_{h,1}) : x^2 = (2h+1)y + h^2$ and $\sup(\overline{P}_{h,1}) : x^2 = (2h+1)y + (h+1)^2$. On rotating $\sup(\underline{P}_{h,1})$ and $\sup(\overline{P}_{h,1})$ about y-axis, we get two paraboloidal surfaces that enclose two *open paraboloidal spaces* given by Eqn. 4.9. Evidently, the absentee circles corresponding to the disc absentees in Octant 1 and Octant 8 lie in the *half-open paraboloidal space* $\mathbf{P}_{h,1}$, given by Eqn. 4.8, for a given value of $h(= r'' - j)$. Hence, the family $\mathcal{F}_1$ of all these half-open paraboloidal spaces is given by Eqn. 4.7, which contains all the aforesaid absentee circles. An illustration is shown in Figure 4.4 for $r = 10$. □

It may be observed that the parabolic regions in Octant 1 containing the disc absentees have all their axes coinciding with y-axis. In Octant 2, all the axes of the parabolic regions containing the disc absentees coincide with x-axis. With this dimensional symmetry, we interchange y and $\sqrt{x^2+z^2}$ in Theorem 4.4.3 to obtain the following corollary.

Corollary 4.4.4 *All the absentee circles of* $\mathbf{S}^{\mathbb{Z}}_{\mathsf{U}}(r)$ *corresponding to Octant 2 lie in*

$$\mathcal{F}_2 := \left\{ \mathbf{P}_{h,2} \cap \mathbb{Z}^3_1 : h = 0, 1, 2, \ldots \right\} \tag{4.13}$$

where,

$$\mathbf{P}_{h,2} = \overline{\mathbf{P}}_{h,2} \smallsetminus \underline{\mathbf{P}}_{h,2} \tag{4.14}$$

such that

$$\begin{aligned} \underline{\mathbf{P}}_{h,2} &: y^2 < (2h+1)\sqrt{x^2+z^2} + h^2, \\ \overline{\mathbf{P}}_{h,2} &: y^2 < (2h+1)\sqrt{x^2+z^2} + (h+1)^2. \end{aligned} \tag{4.15}$$

4.4.2 Absentee Count

As stated in Lemma 4.3.1, there is a one-to-one correspondence between the absentee sets in $\mathbf{H}^{\mathbb{Z}}_{\mathsf{U}}(r)$ and in $\mathcal{D}^{\mathbb{Z}}_{\mathsf{U}}(r)$. Each absentee gives rise to an absentee line segment in $\mathbf{H}^{\mathbb{Z}}_{\mathsf{U}}(r)$ by Property AL (Section 4.4). Similarly, by Property AC, we get the absentee circles in $\mathbf{H}^{\mathbb{Z}}_{\mathsf{U}}(r)$. As a result, we have the following lemma.

Lemma 4.4.5 *The respective counts of absentee line segments and absentee circles in* $\mathbf{H}^{\mathbb{Z}}_{\mathrm{U}}(r)$ *are* $|\mathcal{A}^{\mathbb{Z}^2}(r)|$ *and* $\frac{1}{4}|\mathcal{A}^{\mathbb{Z}^2}(r)|$.

Lemma 4.4.5 provides the exact count of absentee line segments and that of absentee circles. The asymptotic count of absentee-voxels in $\mathbf{S}^{\mathbb{Z}}_{\mathrm{U}}(r)$ is stated shortly in Theorem 4.4.7. For this, we introduce the following lemma.

Lemma 4.4.6 *The number of disc absentees lying between* $\mathcal{C}^{\mathbb{Z}}(r)$ *and* $\mathcal{C}^{\mathbb{Z}}(r+\rho)$ *is* $\Theta(r)$ *for any constant* $\rho \geq 55$*. In particular, the count is bounded below by* $21r+6$ *if* $\rho = 55$*.*

PROOF. Let $a_{r,\rho} = |\mathcal{A}^{\mathbb{Z}^2}(r+\rho)| - |\mathcal{A}^{\mathbb{Z}^2}(r)|$ for $\rho \geq 1$. Clearly,

$$a_{r,\rho} = |\mathcal{D}^{\mathbb{Z}}(r+\rho)| - |\mathcal{D}^{\mathbb{Z}}(r)| - \sum_{r'=r+1}^{r+\rho} |\mathcal{C}^{\mathbb{Z}}(r')|. \tag{4.16}$$

We find a lower bound of $a_{r,\rho}$ using a lower bound of $|\mathcal{D}^{\mathbb{Z}}(r+\rho)|$ and upper bounds of $|\mathcal{D}^{\mathbb{Z}}(r)|$ and $\sum_{r'=r+1}^{r+\rho} |\mathcal{C}^{\mathbb{Z}}(r')|$.

Lower bound of $|\mathcal{D}^{\mathbb{Z}}(r+\rho)|$ The union of the unit squares centered at the points in $\mathcal{D}^{\mathbb{Z}}(r+\rho)$ forms an axis-parallel polygon whose interior would contain the real circle centered at o and having radius $r+\rho-1$. This yields $|\mathcal{D}^{\mathbb{Z}}(r+\rho)| > \pi(r+\rho-1)^2$.

Upper bound of $|\mathcal{D}^{\mathbb{Z}}(r)|$ With a similar analysis as above, we get $|\mathcal{D}^{\mathbb{Z}}(r)| < \pi(r+1)^2$.

Lower bound of $\sum_{r'=r+1}^{r+\rho} |\mathcal{C}^{\mathbb{Z}}(r')|$ Using $|\mathcal{C}^{\mathbb{Z}}(r')| \leq 4\sqrt{2}r' + 8$ (Theorem 4 in [Bera *et al.* (2013)]), we get

$$\sum_{r'=r+1}^{r+\rho} |\mathcal{C}^{\mathbb{Z}}(r')| \leq 4\sqrt{2} \sum_{r'=r+1}^{r+\rho} r' + 8\rho = 2\sqrt{2}\rho(2r+\rho+2\sqrt{2}+1).$$

Hence, from Equation 4.16,

$$\begin{aligned} a_{r,\rho} &> \pi(r+\rho-1)^2 - \pi(r+1)^2 - 2\sqrt{2}\rho(2r+\rho+2\sqrt{2}+1) \\ &= ((2\pi-4\sqrt{2})\rho-4\pi)r + \rho(\pi(\rho-2) - 2\sqrt{2}(\rho+2\sqrt{2}+1)). \end{aligned} \tag{4.17}$$

We now choose the value of ρ so that $((2\pi - 4\sqrt{2})\rho - 4\pi) > 0$ and $\rho(\pi(\rho-2) - 2\sqrt{2}(\rho + 2\sqrt{2}+1)) > 0$. For the former, $\rho > \frac{2\pi}{\pi - 2\sqrt{2}}$; and for the latter, $\rho > \frac{2\pi+8+2\sqrt{2}}{\pi-2\sqrt{2}}$. Consequently, we get

$$\rho > \frac{2\pi + 8 + 2\sqrt{2}}{\pi - 2\sqrt{2}} = 54.641 \text{ (approx.)}, \text{ or, } \rho \geq 55.$$

So, with $\rho(\geq 55)$ as a constant, $a_{r,\rho} = \Omega(r)$. In particular, for $\rho = 55$, Equation 4.17 yields

$$\begin{aligned} a_{r,\rho} &> (106\pi - 220\sqrt{2})r + 2915\pi - 110\sqrt{2}(56 + 2\sqrt{2}) = 21.882r + 6.187 \text{ (approx.)} \\ &> 21r + 6. \end{aligned}$$

As the value of $a_{r,\rho}$ cannot exceed $O(r)$ for any positive integer constant ρ, we get $a_{r,\rho} = \Theta(r)$. $\square$

Theorem 4.4.7 *The count of all absentee-voxels in* $\mathbf{S}^{\mathbb{Z}}_{\cup}(r)$ *is* $\Theta(r^3)$.

PROOF. We first count the absentee-voxels comprising the absentee line segments in $\mathbf{H}^{\mathbb{Z}}_{\cup}(r)$. From the radius interval $[1, r]$, we consider the sub-intervals $\{[(t-1)\rho+1, t\rho+1] : 1 \leq t \leq \lfloor r/\rho \rfloor\}$, so that the length of each sub-interval is constant $(= \rho + 1)$. Corresponding to the t-th sub-interval, the count of absentees between $\mathcal{C}^{\mathbb{Z}}((t-1)\rho+1)$ and $\mathcal{C}^{\mathbb{Z}}(t\rho+1)$ is $\Theta((t-1)\rho+1)$ by Lemma 4.4.6, which simplifies to $\Theta(t)$ on setting $\rho = 55$.

From Theorem 4.4.1, the count of voxels in $\mathbf{L}^{\mathbb{Z}^3}_{(i,\rho)}$ is given by the length of the topmost run of $\mathcal{C}^{\mathbb{Z}}_{34}(r'+1)$, which is $\lfloor \sqrt{r'} \rfloor + 1$ by Lemma 4.3.4. So, for each absentee-pixel in $\mathcal{A}^{\mathbb{Z}^2}(r'+\rho) \smallsetminus \mathcal{A}^{\mathbb{Z}^2}(r')$, where $r' = (t-1)\rho + 1$, the corresponding absentee line segment consists of $\Theta(\sqrt{(t-1)\rho+1})$ to $\Theta(\sqrt{t\rho+1})$ voxels, which simplifies to $\Theta(\sqrt{t})$, as ρ is a constant. Hence, the count of absentee-voxels comprising all the absentee line segments in $\mathbf{H}^{\mathbb{Z}}_{\cup}(r)$ is

$$\sum_{t=1}^{\lfloor r/\rho \rfloor} \Theta(t)\Theta(\sqrt{t}) = \Theta\left(\sum_{t=1}^{\lfloor r/\rho \rfloor} t^{3/2}\right) = \Theta\left(\sum_{t=1}^{\Theta(r)} t^{3/2}\right) = \Theta\left(r^{5/2}\right). \tag{4.18}$$

Now we count the absentee-voxels comprising the absentee circles in $\mathbf{H}^{\mathbb{Z}}_{\cup}(r)$. From (AC), each absentee $p(i,j,0)$ corresponds to an absentee circle $\mathbf{C}^{\mathbb{Z}^3}_{(i,j)}$, which has radius i and lies on the plane $y = j$. Its symmetric absentee $p'(j,i,0)$ corresponds to another absentee circle $\mathbf{C}^{\mathbb{Z}^3}_{(j,i)}$, which has radius j and lies on the plane $y = i$. Voxel count of these two absentee circles is $\Theta(i) + \Theta(j) = \Theta(i+j)$, which is asymptotically same as the voxel count of $\mathbf{C}^{\mathbb{Z}^3}_{(r',r',0)}$, where $r' = i + j$. By Lemma 4.4.6, for $r' = (t-1)\rho + 1$, the count of absentees between $\mathcal{C}^{\mathbb{Z}}(r')$ and $\mathcal{C}^{\mathbb{Z}}(r' + \rho)$ is $\Theta(r')$, and for each of these disc absentees, $i + j = \Theta(r') = \Theta(t)$. Hence, the count of absentee-voxels comprising all the absentee circles in $\mathbf{H}^{\mathbb{Z}}_{\mathrm{U}}(r)$ is

$$\sum_{t=1}^{\lfloor r/\rho \rfloor} \Theta(t)\Theta(t) = \Theta\left(r^3\right). \tag{4.19}$$

On doubling the absentee count as obtained above for $\mathbf{H}^{\mathbb{Z}}_{\mathrm{U}}(r)$, we get the count of all absentees in $\mathbf{S}^{\mathbb{Z}}_{\mathrm{U}}(r)$ as $\Theta(r^3)$. □

4.4.3 Fixing the Absentee Line Segments and Circles

Algorithm 3 (AVS) shows the steps for fixing the absentee line segments and absentee circles corresponding to the solid sphere $\mathbf{S}^{\mathbb{Z}}_{\mathrm{U}}(r)$ having radius r. For each disc absentee in Octant 1 on zx-plane, there are four or eight absentee line segments, which are computed by invoking the procedure AbLine, as shown in Step 14 of Algorithm 3. Again, for each disc absentee in Octant 1 and Octant 2 on xy-plane, there are two absentee circles—one for the upper hemisphere and another for the lower. These absentee circles are computed by invoking the procedure AbCircle, as shown in Step 15 of Algorithm 3.

The procedure AbLine takes the coordinates (i_a, k_a) of a disc absentee as input. It also takes a radius r as the third argument, such that (i_a, k_a) lies between $\mathcal{C}^{\mathbb{Z}}(r)$ and $\mathcal{C}^{\mathbb{Z}}(r+1)$. Based on these, it computes the voxels comprising the absentee line segments $\left\{\mathbf{L}^{\mathbb{Z}^3}_{(i'_a,k'_a)} : \{|i'_a|\} \cup \{|k'_a|\} = \{i_a, k_a\}\right\}$.

The procedure AbCircle requires only the coordinates (i_a, j_a) of a disc absentee as input. If the disc absentee has $i_a = j_a$, then there arises one absentee circle with radius i_a and center $(0, j_a, 0)$; otherwise, there are two absentee circles with radius i_a and j_a, and centered at $(0, j_a, 0)$ and $(0, i_a, 0)$, respectively.

4.5 Test Results and Conclusion

We have implemented Algorithm 2 and Algorithm 3 to generate absentee-free spheres and solid spheres. Figure 4.5 shows (absentee-free) instances of a sphere and a solid sphere generated by these two algorithms for radius 20. Note that as the sphere and the solid sphere are generated by the proposed algorithms as surface of revolution using a

Algorithm 3: (AVS) Fixing absentee-voxels in solid sphere

```
Input: Radius $r$ of a solid sphere
Output: Set of the absentees
$\mathbf{A}^{\mathbb{Z}^3}(r) \leftarrow \emptyset$
int $i \leftarrow 0, j \leftarrow r, s \leftarrow 0, w \leftarrow r-1, h \leftarrow 0, i_a, j_a$
int $l \leftarrow 2w$
while $j \geq i$ do
    repeat
        $s \leftarrow s + 2i + 1,\ i \leftarrow i + 1$
    until $s \leq w$;
    $i_a \leftarrow i - 1, j_a \leftarrow j$
    while $j_a \geq i_a$ do
        if $i_a^2 < (2h+1)j_a + h^2$ then
            $j_a \leftarrow j_a - 1$
        else
            if $i_a^2 < (2h+1)j_a + (h+1)^2$ then
                $\mathbf{A}^{\mathbb{Z}^3}(r) \leftarrow \mathbf{A}^{\mathbb{Z}^3}(r) \cup \mathrm{AbLine}(i_a, j_a, j_a + h)$
                if $i_a = j_a$ then
                    $\mathbf{A}^{\mathbb{Z}^3}(r) \leftarrow \mathbf{A}^{\mathbb{Z}^3}(r) \cup \mathrm{AbCircle}(i_a, j_a)$
                else
                    $\mathbf{A}^{\mathbb{Z}^3}(r) \leftarrow \mathbf{A}^{\mathbb{Z}^3}(r) \cup \mathrm{AbCircle}(i_a, j_a)$
                    $\mathbf{A}^{\mathbb{Z}^3}(r) \leftarrow \mathbf{A}^{\mathbb{Z}^3}(r) \cup \mathrm{AbCircle}(j_a, i_a)$
            $i_a \leftarrow i_a - 1$
    $w \leftarrow w + l,\ l \leftarrow l - 2,\ j \leftarrow j - 1,\ h \leftarrow h + 1$
return $\mathbf{A}^{\mathbb{Z}^3}(r)$
```

digital circular arc as the generatrix, both of them portray 16 symmetric segments—eight above the zx-plane, and eight below it. This is in contrast with the 48-symmetric spheres generated by the algorithm in [Andres (1994)]. See, for example, two instances of a sphere of radius 7 in Figure 4.6. The first one is generated by Algorithm 2; out of the three portions, namely A, B, and C, marked on it, A and B are symmetric with each other but not with C. On the contrary, these three corresponding portions are all pairwise symmetric for the second one, which is generated by the algorithm in [Andres (1994)].

Table 4.1: Exact counts of voxels in $\mathcal{S}^{\mathbb{Z}}_{\cup}(r)$, $\mathcal{A}^{\mathbb{Z}^3}(r)$, and $\mathcal{S}^{\mathbb{Z}}(r)$.

r	$\lvert\mathcal{S}^{\mathbb{Z}}_{\cup}(r)\rvert$	$2\lvert\mathcal{A}^{\mathbb{Z}^3}(r)\rvert$	$\lvert\mathcal{S}^{\mathbb{Z}}(r)\rvert$	r	$\lvert\mathcal{S}^{\mathbb{Z}}_{\cup}(r)\rvert$	$2\lvert\mathcal{A}^{\mathbb{Z}^3}(r)\rvert$	$\lvert\mathcal{S}^{\mathbb{Z}}(r)\rvert$
0	1	0	1	1200	14543190	902056	15445246
1	6	0	6	1300	17063386	1058408	18121794
2	46	8	54	1400	19796562	1227664	21024226
3	82	8	90	1500	22720358	1409144	24129502
4	170	8	178	1600	25858590	1603424	27462014
5	254	24	278	1700	29186106	1810216	30996322
6	330	24	354	1800	32729258	2029288	34758546
7	498	40	538	1900	36460174	2261192	38721366
8	614	40	654	2000	40391978	2505328	42897306
9	830	48	878	2100	44542482	2762328	47304810
10	1002	80	1082	2200	48877878	3031440	51909318
20	3978	256	4234	2300	53433334	3313344	56746678
30	8962	560	9522	2400	58172210	3607600	61779810
40	16310	1016	17326	2500	63132842	3914608	67047450
50	25374	1592	26966	2600	68275238	4234008	72509246
60	36438	2296	38734	3000	90906366	5637120	96543486
70	49510	3080	52590	3500	123729002	7672616	131401618
80	64526	3992	68518	4000	161600518	10021480	171621998
90	81582	5080	86662	4500	204521258	12683288	217204546
100	100622	6248	106870	5000	252490950	15658504	268149454
200	404262	25104	429366	5500	305509450	18946648	324456098
300	908250	56320	964570	6000	363576838	22548008	386124846
400	1617026	100304	1717330	6500	426693594	26462560	453156154
500	2524486	156608	2681094	7000	494859006	30690136	525549142
600	3638230	225456	3863686	7500	568134414	35231256	603365670
700	4949282	307064	5256346	8000	646401914	40085200	686487114
800	6461350	400768	6862118	8500	729718814	45252704	774971518
900	8182310	507392	8689702	9000	818084450	50732656	868817106
1000	10097978	626304	10724282	9500	911499582	56526944	968026526
1100	12223938	757888	12981826	10000	1009962778	62620784	1072583562

We have performed experiments to compute the exact counts of absentee-voxels and sphere voxels for increasing radius of spheres of revolution. Table 4.1 shows the counts of voxels in $\mathcal{S}^{\mathbb{Z}}_{\cup}(r)$, $\mathcal{A}^{\mathbb{Z}^3}(r)$, and $\mathcal{S}^{\mathbb{Z}}(r)$, for r up to 10000. We have also plotted these counts against radius r in Figure 4.7. These experimental results reinforce our analytical findings that all the three counts have a quadratic dependency on r. The relative counts of absentee-voxels corresponding to digital spheres of revolution for radius up to 10000 are tabulated in Table 4.2 and plotted in Figure 4.8. In Table 4.2, the relative count $\mathcal{A}^{\mathbb{Z}^3}(r)/\mathcal{S}^{\mathbb{Z}}(r)$ is denoted by α_r. We observe from these data that with the increasing radius, the value of relative count for solid sphere tends to 0.058 approximately.

We have also generated through our experiments the exact counts of absentee-voxels

Table 4.2: Relative count of absentees versus radius in spheres of revolution.

r	α_r	r	α_r	r	α_r	r	α_r
2	0.148148	17	0.059393	180	0.058313	1700	0.058401
3	0.088889	18	0.063269	200	0.058468	1800	0.058382
4	0.044944	19	0.061507	300	0.058389	1900	0.058397
5	0.086331	20	0.060463	400	0.058407	2000	0.058403
6	0.067797	30	0.058811	500	0.058412	2500	0.058386
7	0.074349	40	0.058640	600	0.058353	3000	0.058389
8	0.061162	50	0.059037	700	0.058418	3500	0.058391
9	0.054670	60	0.059276	800	0.058403	4000	0.058393
10	0.073937	70	0.058566	900	0.058390	4500	0.058393
11	0.059435	80	0.058262	1000	0.058401	5000	0.058395
12	0.061617	90	0.058618	1100	0.058381	6000	0.058396
13	0.063277	100	0.058464	1200	0.058404	7000	0.058396
14	0.060094	120	0.058367	1300	0.058405	8000	0.058392
15	0.060453	140	0.058532	1500	0.058399	9000	0.058393
16	0.054637	160	0.058495	1600	0.058387	10000	0.058383

and sphere voxels corresponding to solid spheres of revolution. The counts of voxels in $\mathbf{S}^{\mathbb{Z}}_{\cup}(r)$, $\mathbf{A}^{\mathbb{Z}^3}(r)$, and $\mathbf{S}^{\mathbb{Z}}(r)$, for r up to 800, are shown in Table 4.3 and plotted in Figure 4.9. Similar to the previous set of results, these experimental results also reinforce our analytical findings that all the three counts corresponding to the solid sphere have a cubic dependency on r. The relative counts of absentee-voxels corresponding to solid spheres of revolution for radius up to 800 are tabulated in Table 4.4 and plotted in Figure 4.10. We observe from these data that with the increasing radius, the value of relative count tends to 0.101 approximately.

The above test results and their theoretical analysis indicate that the ratio of the absentee-voxels to the total number of voxels tends to a constant for large radius. As evident from the algorithms and related discussions, the knowledge of geometric distributions of absentee-voxels is found to be useful for algorithmic generation of a digital sphere. An asymptotic tight bound for the count of absentees is given, but finding a closed-form solution on the exact count of absentees for a given radius still remains an open problem. Characterization of these absentees requires further in-depth analysis, especially if we want to generate a solid digital sphere with concentric digital spheres. Apart from spheres, generation of various other types of surfaces, which should be free of any absentee-voxels, has also many applications in 3D imaging and graphics, such as creation of interesting pottery

Table 4.3: Exact counts of voxels in $\mathbf{S}^{\mathbb{Z}}_{\cup}(r)$, $\mathbf{A}^{\mathbb{Z}^3}(r)$, and $\mathbf{S}^{\mathbb{Z}}(r)$.

r	$\lvert\mathbf{S}^{\mathbb{Z}}_{\cup}(r)\rvert$	$2\lvert\mathbf{A}^{\mathbb{Z}^3}(r)\rvert$	$\lvert\mathbf{S}^{\mathbb{Z}}(r)\rvert$	r	$\lvert\mathbf{S}^{\mathbb{Z}}_{\cup}(r)\rvert$	$2\lvert\mathbf{A}^{\mathbb{Z}^3}(r)\rvert$	$\lvert\mathbf{S}^{\mathbb{Z}}(r)\rvert$
0	1	0	1	80	1941629	233828	2175457
1	7	0	7	90	2761237	333428	3094665
2	53	20	73	100	3785733	452052	4237785
3	143	20	163	150	12749489	1508868	14258357
4	321	20	341	200	30196125	3528744	33724869
5	591	132	723	250	58952525	6810356	65762881
6	945	132	1077	300	101848409	11688640	113537049
7	1483	276	1759	350	161726089	18514264	180240353
8	2153	276	2429	400	241406453	27530128	268936581
9	3039	360	3399	450	343714485	39030584	382745069
10	4121	752	4873	500	471497269	53389448	524886717
20	31377	4192	35569	550	627583253	70890036	698473289
30	104321	13144	117465	600	814799465	91803032	906602497
40	245349	31412	276761	650	1035980249	116498872	1152479121
50	477061	60436	537497	700	1293980265	145396532	1439376797
60	821805	103604	925409	750	1591598569	178467668	1770066237
70	1303165	159636	1462801	800	1931678709	216171360	2147850069

designs, as shown recently in [Kumar *et al.* (2011)].

Table 4.4: Relative counts of absentee-voxels versus radius in solid spheres of revolution.

r	α_r	r	α_r	r	α_r	r	α_r
2	0.27397	16	0.10654	120	0.10654	320	0.10300
3	0.12270	17	0.11685	130	0.10620	340	0.10283
4	0.05865	18	0.12414	140	0.10606	360	0.10250
5	0.18257	19	0.12345	150	0.10582	380	0.10233
6	0.12256	20	0.11786	160	0.10523	400	0.10237
7	0.15691	30	0.11190	170	0.10482	420	0.10219
8	0.11363	40	0.11350	180	0.10482	450	0.10198
9	0.10591	50	0.11244	190	0.10463	500	0.10172
10	0.15432	60	0.11195	200	0.10463	550	0.10149
11	0.12030	70	0.10913	220	0.10403	600	0.10126
12	0.12264	80	0.10748	240	0.10368	650	0.10109
13	0.12249	90	0.10774	260	0.10361	700	0.10101
14	0.12018	100	0.10667	280	0.10328	750	0.10083
15	0.11719	110	0.10661	300	0.10295	800	0.10065

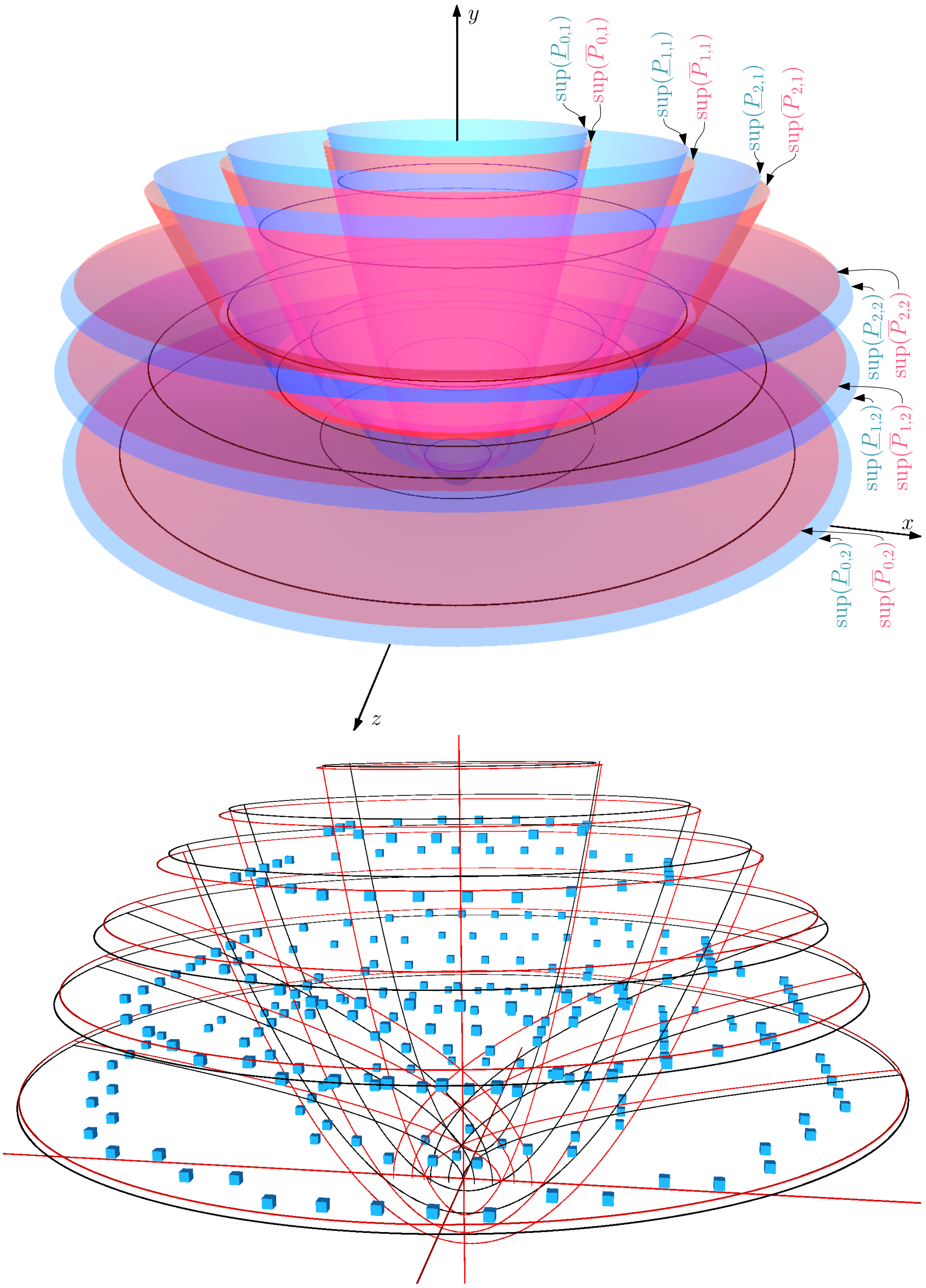


Figure 4.4: Illustration of Theorem 4.4.3 and Corollary 4.4.4 for $r = 10$. For clarity, the absentee circles lying in the open paraboloidal spaces are shown as real circles. Bottom: The absentee-voxels, shown reduced in size for clarity.

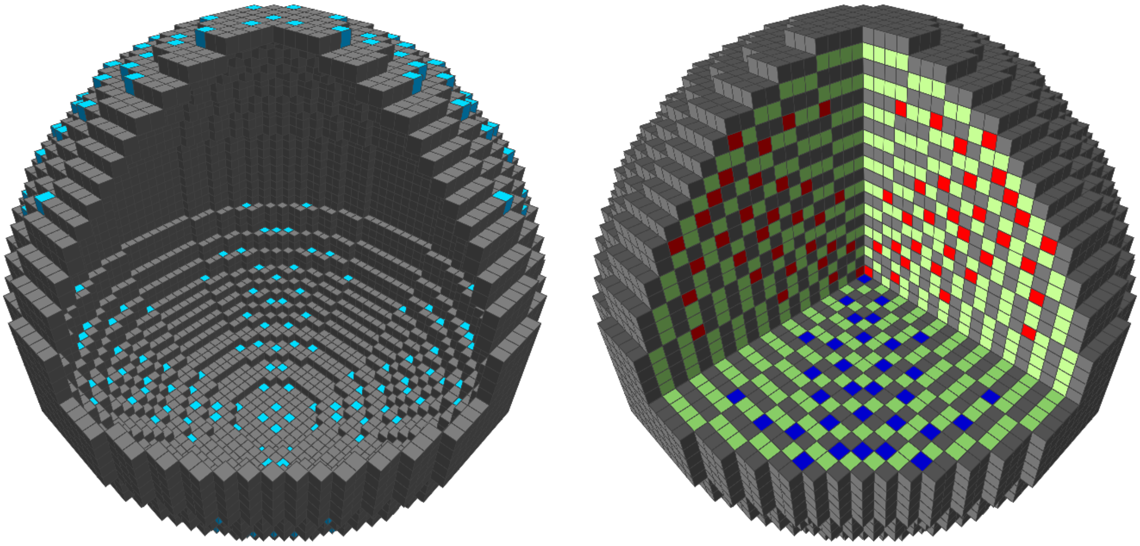
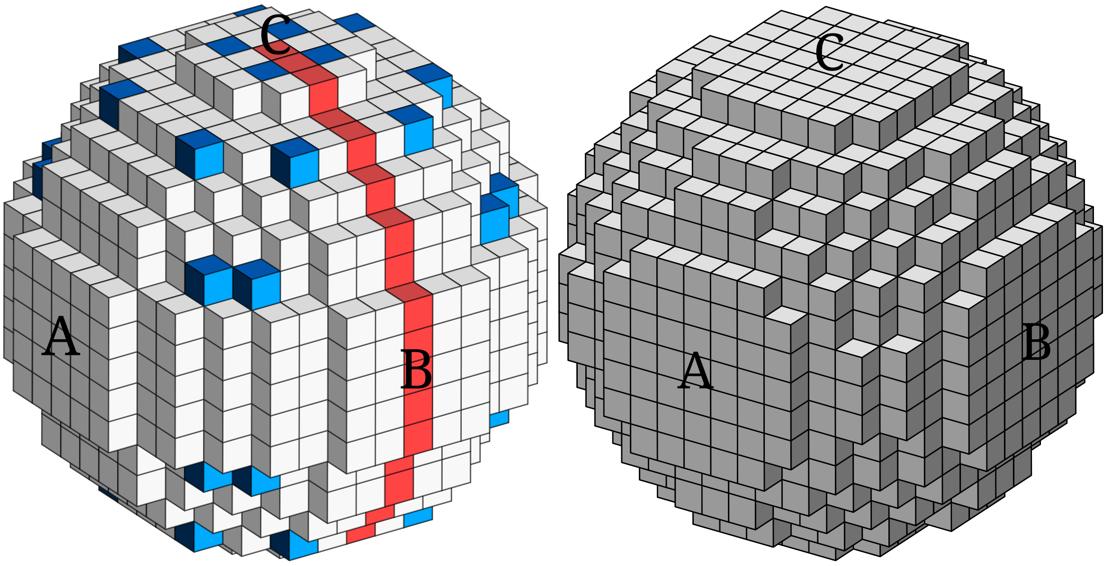


Figure 4.5: Sphere and solid sphere of radius 20 generated by the proposed algorithm.

```
Procedure AbLine($i_a, k_a, r$)
$A \leftarrow \emptyset$
int $j_a \leftarrow 0$
while $j_a \leq \lfloor\sqrt{r}\rfloor + 1$ do
    $A \leftarrow A \cup \{(i'_a, j'_a, k'_a) : \{|i'_a|\} \cup \{|k'_a|\} = \{i_a, k_a\} \wedge |j'_a| = j_a\}$
    $j_a \leftarrow j_a + 1$
return $A$
```

```
Procedure AbCircle($i_a, j_a$)
$A \leftarrow \emptyset, r \leftarrow i_a$
int $i_a \leftarrow 0, k_a \leftarrow r, s \leftarrow 0, w \leftarrow r - 1$
int $l \leftarrow 2w$
while $k_a \geq i_a$ do
    repeat
        $A \leftarrow A \cup \{(i'_a, j'_a, k'_a) : \{|i'_a|\} \cup \{|k'_a|\} = \{i_a, k_a\} \wedge |j'_a| = j_a\}$
        $s \leftarrow s + 2i_a + 1$ $i_a \leftarrow i_a + 1$
    until $s \leq w$;
    $w \leftarrow w + l$, $l \leftarrow l - 2$, $k_a \leftarrow k_a - 1$
return $A$
```



(a) (b)

Figure 4.6: Spheres of radius 7 generated by (a) Algorithm 2 and (b) algorithm in [Andres (1994)]. The blue voxels in (a) denote the absentees, and the red ones comprise the digital generatrix.

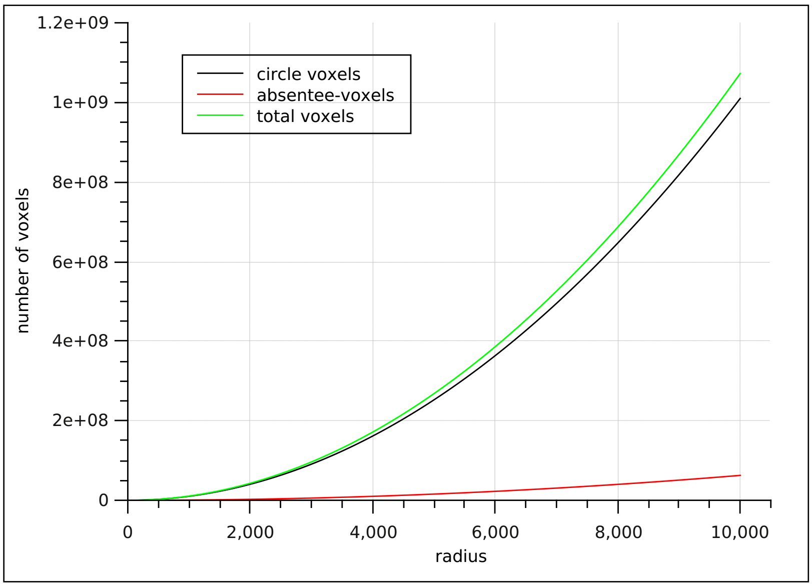


Figure 4.7: Exact counts of voxels in $\mathcal{S}^{\mathbb{Z}}_{\cup}(r)$, $\mathcal{A}^{\mathbb{Z}^3}(r)$, and $\mathcal{S}^{\mathbb{Z}}(r)$.

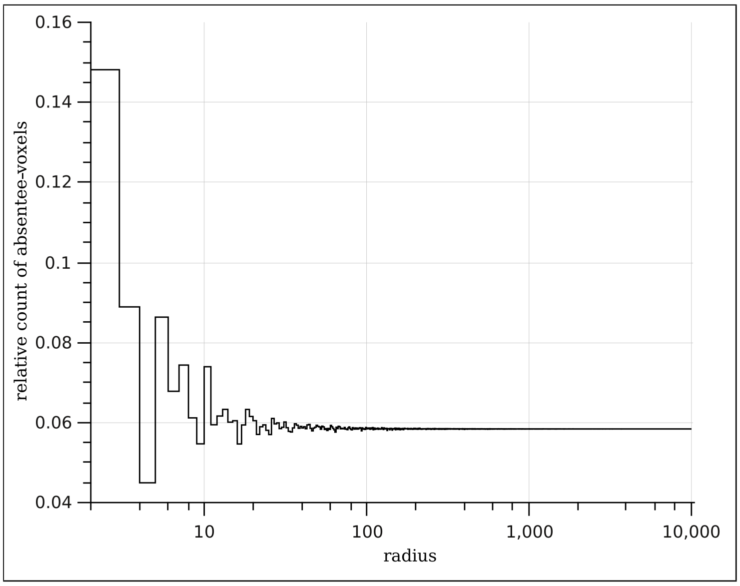


Figure 4.8: Relative count of absentees versus radius in spheres of revolution.

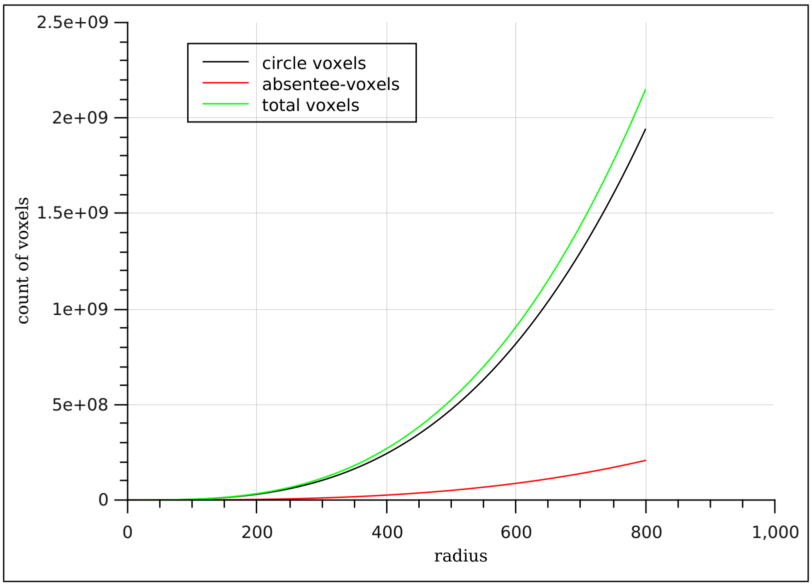


Figure 4.9: Exact counts of voxels in $\mathbf{S}^{\mathbb{Z}}_{\cup}(r)$ (circle voxels), $\mathbf{A}^{\mathbb{Z}^3}(r)$ (absentee-voxels), and $\mathbf{S}^{\mathbb{Z}}(r)$ (total voxels).

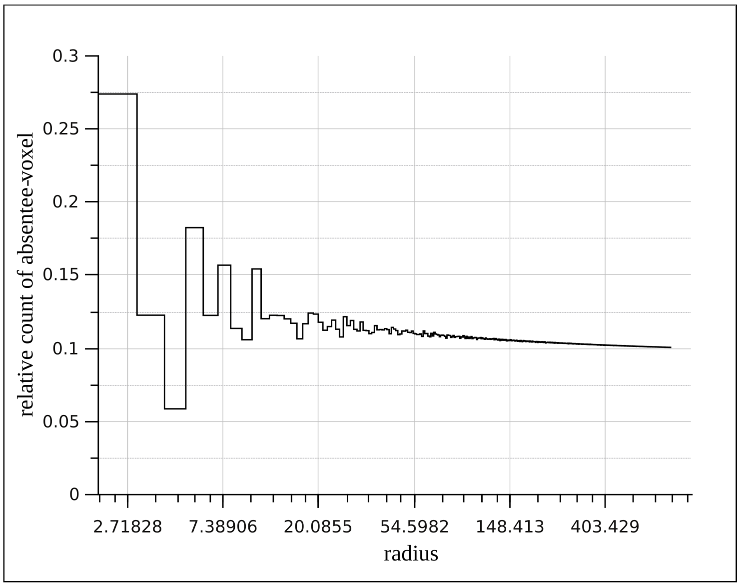


Figure 4.10: Relative counts of absentee-voxels versus radius in solid spheres of revolution.

Chapter 5

Circularity Analysis of Nano-scale Structures

5.1 Introduction

Porous Silicon (PS) based devices have recently emerged as a potential platform for exploring numerous applications to nano-biotechnology, e.g., medical diagnostics, in-vitro pathogen detection, gene identification, and DNA sequencing [Betty (2008), Ghoshal *et al.* (2010, 2011), Granitzer and Rumpf (2010), Stewart and Buriak (2000)]. Because of its non-toxic nature and biodegradability, it is also highly suitable for implementing in-vivo biosensors and drug delivery modules [Anglin *et al.* (2008), Salonen *et al.* (2008)]. The intriguing property of visible light emission from electrochemically etched PS was observed long ago [Cullis and Canham (1991)]. PS chips with an average pore diameter ≤ 2 nm [Rouquerol *et al.* (1994)] are called microporous and they admit photoluminescence (PL) at room temperature whereas, those with pore diameters > 50 nm [Rouquerol *et al.* (1994)], are called macroporous and they have applications to photonics, sensor technology and biomedicine [Betty (2008), Lehmann (2003), Lin *et al.* (1997), Reddy *et al.* (2001), Saha *et al.* (2006)].

Many physical properties of PS, e.g., luminescence, refractive index, and heat conductivity are determined by the fraction of porosity. For biological applications, a uniform arrangement of porous structures called micro-test tubes or micro-beakers having diameters approximately $1 - 1.5$ μm are desirable for loading nanoparticles or drugs within the pores [Ghoshal *et al.* (2011)]. Techniques of creating uniform macroporous structures by controlling formation parameters have been reported in the literature [Harraz *et al.* (2005), Vyatkin *et al.* (2002)]. PS also provides a viable platform for observing surface-enhanced Raman scattering (SERS), which is useful in detecting the presence of chemical and biological molecules [Chan *et al.* (2003), Jiao *et al.* (2010)]. Microbeakers on PS (< 100 nm in

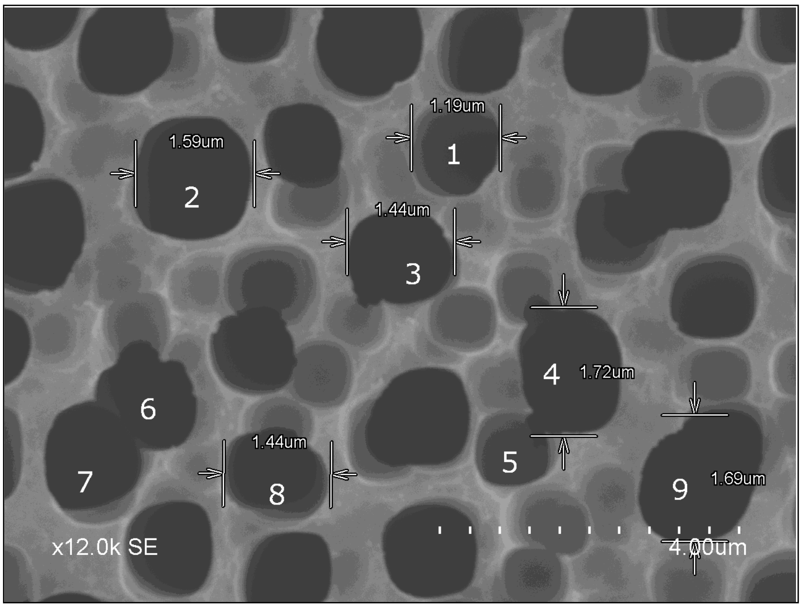


Figure 5.1: SEM image (1280 × 960) of microporous Si formed on p-type substrate with magnification 12000× (top-view).

width) with pore size (> 1.5 μm in diameter) can be used as a SERS substrate for various bio-sensing applications. Ideally, these structures should be produced on the PS chip as a regular array of circular pores. However, because of the process uncertainty, pores often appear with deformed boundaries as observed from the captured SEM images [Ghoshal *et al.* (2011)].

In this chapter, we address the problem of estimating the circularity of the pore structures based on an image processing technique. In a PS chip, adjacent pores may merge to form a connected pattern of complex shape. Based on some geometric properties of a digital circle, we propose a segmentation technique to isolate each of the pores from the given SEM image. Once the boundary of a pore is extracted, a circle-fitting algorithm is deployed to determine a best fit followed by estimating the deviation from circularity by computing the Hausdorff distance [Rucklidge (1997)] between the actual pore boundary and the fitted circle.

5.2 Formulation of the Problem

In order to formulate the problem, we first observe certain characteristic properties of a PS chip image taken by a scanning electron microscope (SEM). A typical top-view SEM image of a PS chip is shown in Fig. 5.1, where the pore boundaries appear as nearly circular objects. As the top surface of a PS chip resembles a 3D terrain, some pores are focused and some are defocused in the captured image. The deep black objects of the image are in the focused plane and thus the corresponding pores appear with sharp boundaries in the image; the lighter black objects represent those pores, which are in the defocused planes and blurred. Thus to analyze the detailed structure of the pores, we first need to perform automatic segmentation to isolate each object from the image.

An ideal microporous structure should consist of an array of pores each having a circular contour. However, in real life, the contours seldom become perfectly circular. Further, during electrochemical etching process, two or more neighboring pores may often coalesce to form a composite porous structure. In the image, the contour of such a pore appears as a closed curve consisting of several nearly-circular arcs. For example, in Fig. 5.1, the objects labeled as 1, 2, 3 and 8 represent a single porous structure each, whereas, the objects labeled as $4-5$, $6-7$ and 9 indicate connected structures formed by overlapping of two or more circular pores. Sometimes, the overlap is peripheral as in the objects labeled as $4-5$, $6-7$ (Fig. 5.1), where the constituent seed pores are visibly perceptible. In such cases, the challenge lies in segmenting out each individual circular contour from the image of the connected object. This is necessary, because otherwise, an attempt to fit a single digital circle [Klette and Rosenfeld (2004a)] to the original contour of the connected pore would incur a huge amount of error. In some other cases when the overlap between neighboring pores are significantly large as in the object labeled 9 in Fig. 5.1, then segmentation may not be useful as the overall contour appears to be that of a nearly-circular pore. In such a case, however, fitting a single circle to the entire contour is likely to yield a fair approximation.

5.3 Proposed Work

Given a SEM image, our algorithm consists of four steps: (i) segmentation of porous objects and extraction of their contours, (ii) isolation of individual circular components from the connected structures, if any, (iii) fitting a digital circle to the contour of each of the segmented pore images, and (iv) computation of Hausdorff distance between the actual contour and the fitted circle to estimate the goodness of circularity.

The various steps of the algorithm are now described below.

5.3.1 Segmenting the porous objects from the input image

The input SEM image $\mathcal{A}$ of the PS chip is supplied as a gray-tone image, in which it is easy to identify three types of objects depending on the intensity value of the pixels. The deep black objects represent the focused pores, the medium black objects represent the defocused pores or small pores, and the gray background, which corresponds to the PS substrate. This is evident from the histogram [Gonzalez and Woods (2001)] of the image (see Fig. 5.5(a)). Using these gray levels, we segment $\mathcal{A}$ and create two binary images one consisting of only the focused pores (deep black) $\mathcal{A}_1$ and another with only the defocused pores (medium black) $\mathcal{A}_2$ so that pores appear in black on a white background. Further analysis is performed on $\mathcal{A}_1$ and $\mathcal{A}_2$ separately.

Contour extraction: Once a binary image of the porous structure $\mathcal{A}_1$ is given, the next step is to extract a one-pixel thick contour for each of the nano-structures, which appear as a black object in $\mathcal{A}_1$. As discussed earlier, such a nano-structure may represent a single pore or a connected pattern of two or more pores. Let $\mathcal{A}_1$ contain m objects b_1, b_2, ... b_m and n pores a_1, a_2, ... a_n, $m(\leq n)$.

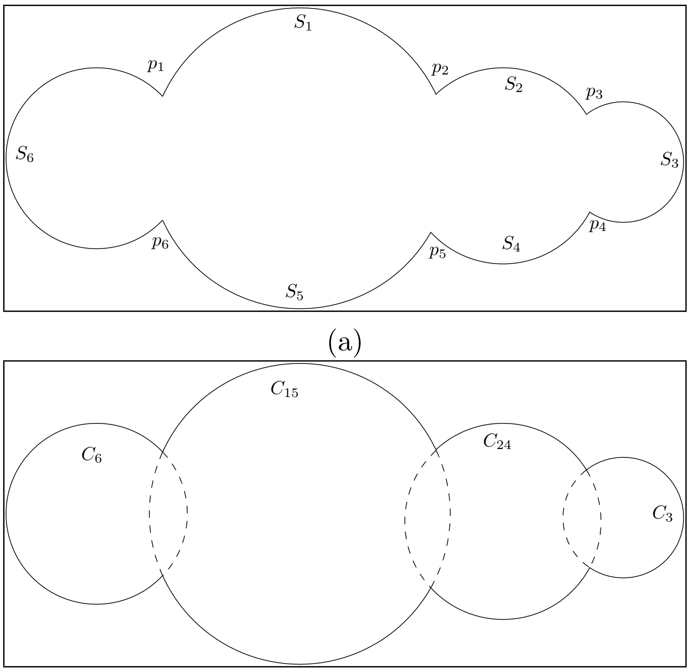
Although there exist several algorithms for detecting boundary/edge of an object, for instance, those based on Canny edge detection [Chanda and Majumder (2009)], Laplacian operator [Chanda and Majumder (2009)], Sobel operator [Chanda and Majumder (2009)], we here use a simple scan-type algorithm to output a single-pixel thick contour.

We first make a copy $\mathcal{B}_1$ of the image $\mathcal{A}_1$. Then the image $\mathcal{A}_1$ is scanned from left-to-right and top-to-bottom to check the *four neighbors* of each black pixel. If all the *four neighbors* of a black pixel are black in $\mathcal{A}_1$, then the corresponding pixel in $\mathcal{B}_1$ is replaced by a white pixel. Clearly, at the end of the scan, for each object b_i, $i = 1...m$ in $\mathcal{A}_1$, a one-pixel thick 8 *connected* closed curve c_i will be stored in $\mathcal{B}_1$. Let $C = c_1 \cup c_2 \cup ... \cup c_n$.

Isolating connected pores: Since some porous structure may appear as a connected pattern of overlapping pores, e.g., objects $4-5$, $6-7$ in Fig. 5.1, it is necessary to identify the underlying circular pores that form the pattern. Thus, after the one-pixel thick boundary of an object is extracted, we examine the curvature of the contour as follows. We assume that the closed contour consists of one or more circular arcs. In the former case, the structure will be approximated by a single fitting circle. Otherwise, the component arcs are identified and subsequently the constituent circular shapes are determined. In order to accomplish this, we check the chain code of the contour and make use of a property of a digital circle [Klette and Rosenfeld (2004a)].

Property of chain code of a digital circle: It is known that the chain code of a digital circle satisfies a simple differential property: for any two consecutive pixels along the circumference, the difference in chain code is either 0 or 1 or 7. Figure 5.3(b) shows the chain code of a digital circle. In other words, the angle between two consecutive pixel directions in the sense of 8-neighborhood is either 0 or $\frac{\pi}{4}$. However, if the boundary curve consists of two or more digital circles then at the turning points, the above differential property of the chain code will not hold. Fig. 5.2 shows an example of a closed curve consisting of six circular segments of four digital circles. Let $p_1, p_2, ..., p_6$ denote the six turning points and let $S_1, S_2, ..., S_6$ be the six segments. Since at a turning point (or joining point), two consecutive circular segments meet, there will be a sharp change in directional angle, which invalidates the differential property of the chain code.

Thus by checking the chain code along the boundary of the curve, we can easily identify the location of turning points and the arc segments. Ideally, the number of turning points should be an even; however, some of the turning points may disappear because of uneven electrochemical etching. If the number of turning points is 0, then it consists of a single pore. Otherwise, we need to identify the underlying pores in the connected pattern. For example, in Fig. 5.2, the total number of segments is six, but they belong to four circles, as segments S_1 and S_5 belong to the same circle. Similarly, S_2 and S_4 belong to the same circle. In order to identify these underlying circles, we use a technique of circular arc recognition based on a digital geometric method [Bera *et al.* (2010)]. For each of the circular segments $S_1, S_2, ..., S_6$, let $C_1, C_2, ..., C_6$ be the corresponding fitted digital circle. The digital circles C_1 and C_5 will have nearly equal radii, and their centres are very close to each other; hence they can be treated as components of a single digital circle C_{15}. Similarly C_2 and C_4 are recognized as arcs of a single digital circle C_{24}. So, in this case, the connected structure can finally be decomposed into four digital circles C_6, C_{15}, C_{24}, C_3 (see Fig. 5.2). In the next subsection, we describe the procedure of fitting a circle to a



(a)

(b)

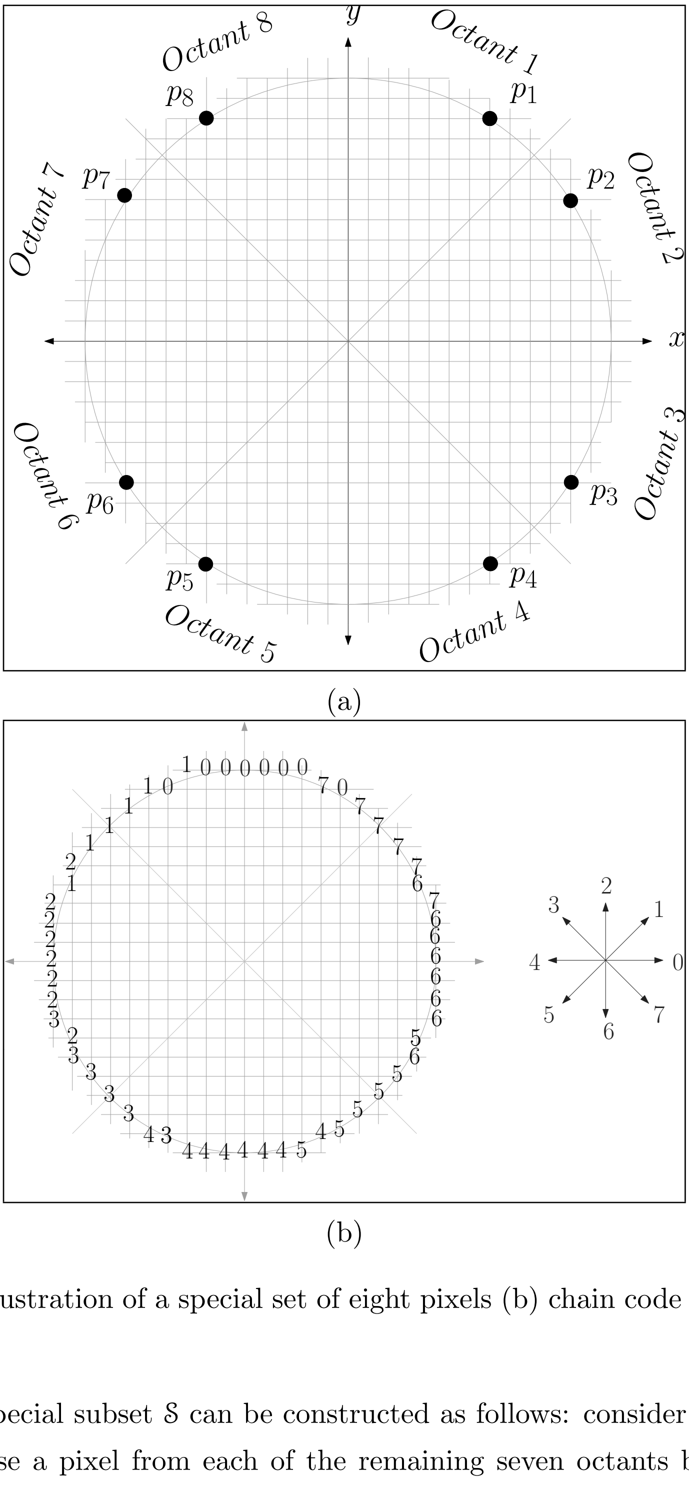
Figure 5.2: (a) A connected structure of 4 overlapping pores (b) isolation of pores.

digital arc in detail.

5.3.2 Fitting a circle to a closed digital curve

For each closed curve in $\mathcal{C}$, we first identify the digital circular segments (arcs), and then compute a fitting circle for each of these segments. If there is no turning point, only one circle is computed, otherwise multiple fitting circles are computed and merged, whenever possible, as discussed earlier.

A geometric property of a digital circle: In real space $\mathbb{R}^2$, there exists a unique circle that passes through three fixed non-collinear points. Also, given any three points on a circle, the construction procedure based on perpendicular bisectors will produce the same circle. However, any three points chosen from the boundary of a circle in digital space $\mathbb{Z}^2$, may not yield the same circle in the euclidean space. Only in the special cases when the three points are chosen from a special subset $\mathbb{S}$ of grid points $\mathcal{G}$ of a digital circle, the reconstructed euclidean circle determined by the two perpendicular bisectors become



(a)

(b)

Figure 5.3: (a) Illustration of a special set of eight pixels (b) chain code of a digital circle.

unique. Such a special subset $\mathcal{S}$ can be constructed as follows: consider a pixel in the 1_{st} octant, and choose a pixel from each of the remaining seven octants by reflection. For example, let p_1 be a pixel in 1_{st} Octant and $p_2, p_3, p_4, p_5, p_6, p_7, p_8$ be the seven pixels obtained by reflection as shown in Fig. 5.3(a). Let $\mathcal{S} = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8\}$. Then $\mathcal{S}$ is a special subset of $\mathcal{G}$.

In our problem, since the boundary of the pore structures are seldom perfectly circular, the above method will not work. Hence, for a closed curve c_i in C, we extract the segments, and then from each segment $\mathcal{H}$ we choose three pixels p_1, p_2, p_3 at random. Using the coordinate values of the pixels p_1, p_2, p_3 the center and radius of the digital circle passing through these three pixels are computed from the intersection of perpendicular bisectors and by applying rounding-off techniques. The center and radius are stored in the two arrays $\mathcal{C}n$ and $\mathcal{R}$ respectively. We repeat this procedure for a large number of triplet pixels chosen from the segment $\mathcal{H}$. Finally, the center and radius of the best fitting circle for the segment $\mathcal{H}$ are computed by taking the medians of the respective values stored in $\mathcal{C}n$ and $\mathcal{R}$. If for two or more segments, the computed values of their radii and the locations of their centres are very close, we merge these segments to fit a single digital circle.

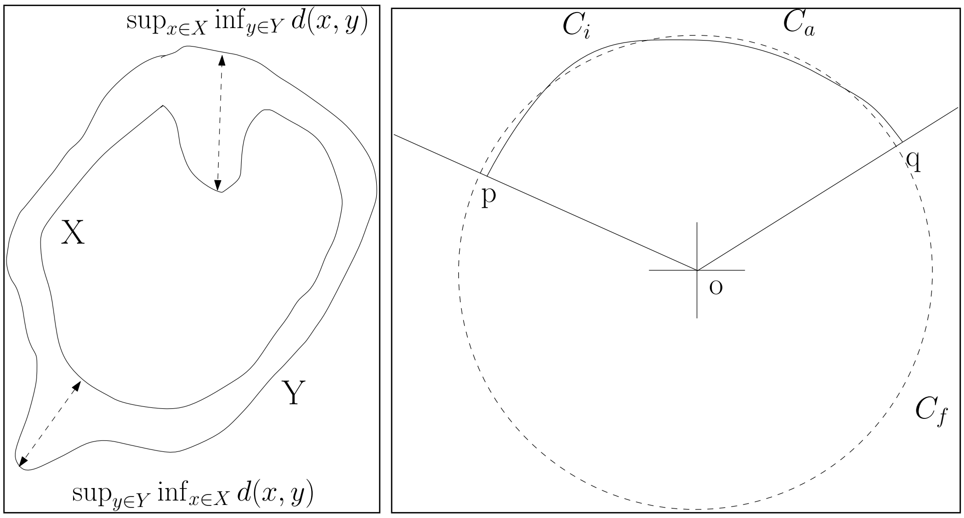
5.3.3 Computing Hausdorff distance

To estimate the deviation from circularity of the porous objects, we use Hausdorff distance [Rucklidge (1997)] as a measure of error. The Hausdorff distance (HD) or Hausdorff metric measures how far two subsets of a metric space are from each other. Informally, two sets are close in the sense of Hausdorff distance if every point of either set is close to some point of the other set. Let X and Y be two non-empty subsets of a metric space (M, d). We define their Hausdorff distance $d_H(X, Y)$ by

$$d_H(X,Y) = max\{\sup_{x \in X} \inf_{y \in Y} d(x,y), \sup_{y \in Y} \inf_{x \in X} d(x,y)\} \tag{5.1}$$

where sup represents the supremum and inf the infimum. Figure 5.4(a) shows the Hausdorff distance between two sets. In our experiment, we also attempt to fit a circle to an open segment, as shown in Fig. 5.4(b), where C_i is a contour segment and C_f is the fitted digital circle. So, if we compute the Hausdorff distance between C_i and C_f, we will observe unexpected result as the input curve C_i is not closed. To tackle this problem, we consider the arc C_a between op and oq and compute the Hausdorff distance only between C_i and C_a to estimate the error of fitting.

The HD of the input digital curve and the fitted digital circle is computed to estimate



(a) (b)

Figure 5.4: (a) Illustration of Hausdorff distance (b) computed segment in our experiment

the deviation from circularity. From the empirical evidence we conclude that if HD is within 30% of the radius of the fitted circle then the pore is fairly circular in shape.

5.3.4 Interpreting the structure of PS

The PS chip may consist of a large number of pores. We can classify them into three groups (best, good, bad) depending on their goodness of circularity and the requirement. For each pore we calculate the percentage error $(PE) = \frac{HD\times 100}{r}$, where r is the radius of fitted digital circle. Depending on the value of PE we classify the pores into three groups. Since our method also computes a median radius value of each pore, we can also estimate the volume of the pore, if the information about the height is known.

5.4 Experimental Results

We have implemented the algorithm in $\mathcal{C}$ on the openSUSE™ OS Release 11.0, HP xw4600 Workstation with Intel® Core™2 Duo, 3 GHz processor. We have performed the test on several SEM images of PS chips, computed the average diameter of each pore, and estimated their circularity from image analysis.

A step-wise demonstration of the proposed method on a sample gray-tone image is shown in Fig. 5.1. In this image, the diameters of some pores have been measured by the SEM along certain directions, which are marked in units of μm.

We first perform histogram thresholding to isolate the objects of the focused plane and produce a binary image containing nearly circular discs, each corresponding to a black porous structure of the input image. For each disc, a one-pixel contour is computed and a simple closed curve is obtained as in Fig. 5.5(a). Next, we analyze each curve by checking its chain code, and decompose it into segments, wherever applicable. A best fit digital circle is determined by computing the medians of the centres and radii arrays obtained by a repetitive experiment of randomly selecting triplet pixels from the segment for a large number of times. Figure 5.5(b) shows the image with fitted digital circles as obtained by our algorithm and Fig. 5.5(c) shows the superimposed circles on the actual images of the pores. We also report the location of the centre and diameter of the fitted circle and compute the PE for each fitted circle relative the actual pore. The measured and computed values of diameters are listed in the Table 5.1. It may be noted that the supplied values of diameter correspond to a measurement along certain directions as shown in Fig. 5.1, whereas the computed one reflects the median value. This accounts for the observed differences in the values of diameter. As the pore labeled as 4 is elliptical in shape, the computed diameter differs significantly from the measured one. This is also reflected in the PE value. In our experiments, 121 pixel length is equal to 1 μm as per SEM characteristics.

This objects in the defocused plane can also be analyzed similarly. Fig. 5.6 shows the results of the experiment on another SEM image of a PS chip. In this image, we have analyzed a total 181 pores, and the fitting digital circles are computed. From the viewpoint of circularity, we classify them as "best" (if $PE \leq 30$) or "good" (if $30 < PE \leq 50$) or "bad" (if $PE > 50$). Thus out of the 181 pores, 112 pores are classified as best, 38 as good and 31 as bad.

5.5 Conclusion

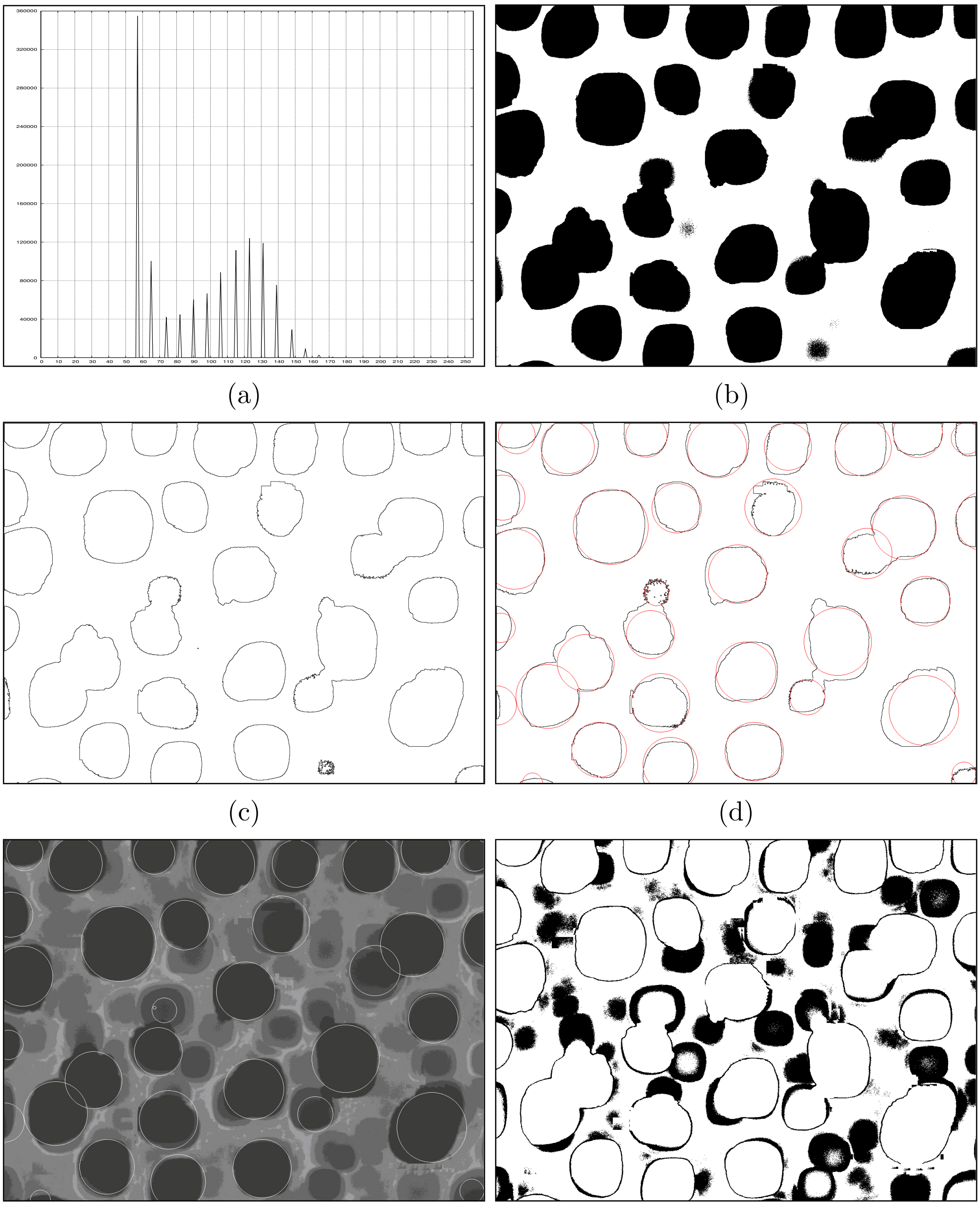
In this chapter, we have presented an automated technique for estimating the circularity of pores in PS chips, purely based on image analysis. Our procedure is simple, fast, and relies on some basic principles of digital circles. We have run our experiments on several SEM images of PS chips, results of some of which are presented in this chapter. The

Table 5.1: Comparison of observed and computed features of the pores for the image shown in Fig. 5.1.

CN	NCP	Center	r	HD	PE	CD	OD
1	700	(734, 227)	70	21.10	30.14	1.16	1.19
2	776	(302, 275)	99	10.00	10.10	1.67	1.59
3	686	(638, 403)	84	14.56	17.33	1.39	1.44
4	787	(909, 581)	93	26.65	28.66	1.53	1.72
5	458	(829, 727)	46	14.14	30.74	0.76	-
6	545	(232, 637)	81	18.97	23.42	1.34	-
7	532	(141, 721)	86	13.00	15.12	1.42	-
8	810	(436, 751)	80	13.00	16.25	1.32	1.44
9	852	(765,1125)	104	25.94	24.94	1.72	1.69

121 pixels = 1 μm, '-' Not observed
CN - Circle Number, NCP - Number of Contour pixels
r - Radius (in no. of pixel)
HD - Hausdorff distance (in no. of pixels)
PE - Percentage Error= $\frac{HD\times100}{r}$
CD - Computed diameter (μm)
OD - Observed diameter along a given direction by SEM (μm)

estimated diameter values are sufficiently accurate and conform nicely to the observed values. As the current practice of conducting manual evaluation of circularity in PS chips is a tedious and time-consuming process, the proposed automated procedure may be used as a powerful analysis tool. Improving the robustness and accuracy of the method may be further investigated as future research issues.

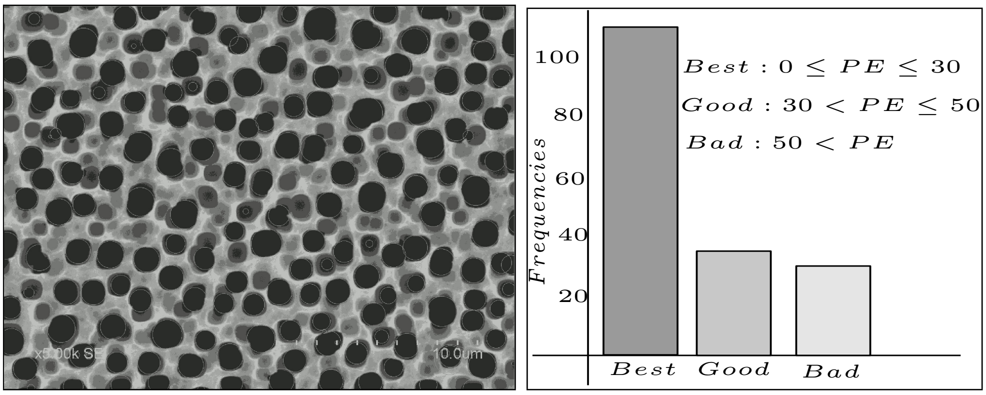


(a) (b)

(c) (d)

(e) (f)

Figure 5.5: Snapshots of the experiment on the image shown in Fig. 5.1 (a) histogram of the image (b) after binarization (c) single-pixel contour extraction for each object (d) fitting a digital circle (e) superimposed fitted circles(f) image of the defocused layer



(a) (b)

Figure 5.6: (a) Result for a PS image (b) circularity classification vs. frequency.

Chapter 6

Granulometric Image Analysis Based on Digital Geometry

6.1 Introduction

The properties of a digital object are well-studied in discrete geometry and they have found several real-life applications in various fields of science and engineering. For agricultural inspection, an automated analysis of granulometric images consisting of convex objects [Klette and Rosenfeld (2004a)] such as rice, wheat, cereals, coffee beans, nuts, cookies, blood cell, or chocolates, is highly desirable for the purpose of segmentation and quality evaluation. When two or more objects in a binary image touch each other or overlap on each other, a single connected object is formed. In order to analyze different characteristics of these objects, it is necessary to segment them into individual components. However, given a snapshot, manual segmentation and counting in a large ensemble of objects is quite tedious. Most of the existing techniques are based on different image segmentation techniques such as watershed transform Casasent *et al.* (1996), Chen *et al.* (2004), Keagy *et al.* (1996), Schatzki *et al.* (1997), Talukder *et al.* (1999b), mathematical morphology Iwanowski (2007), granulometric methods Vincent (2000), use of concavity information Farhan *et al.* (2010, 2013), Kumar *et al.* (2006), Wang *et al.* (2012), Wen *et al.* (2009), Zhong *et al.* (2009) or their variants. However, these techniques often produce under- or over-segmented results, run slowly, or need a large memory space during processing. In this chapter, we study the properties of an ensemble of convex objects from a digital-geometric viewpoint, and use them for concavity analysis, which, in turn, yields an accurate and fast segmentation of individual objects.

The rest of the chapter is organized as follows. A brief review of prior work on convex object segmentation is presented in Sec. 6.1.1. Sec. 6.2 presents the analysis of concavity in a digital object. Sec. 6.3 describes the proposed segmentation method based on digital-geometric properties. Comparative results with the watershed segmentation technique are reported in Sec. 6.4. Finally, conclusions are drawn in Sec. 6.5.

6.1.1 Related Work

Several techniques have been proposed for agricultural product inspection [Keagy and Schatzki (1993), Keagy *et al.* (1996), Schatzki and Wong (1989), Schatzki *et al.* (1981, 1997)] based on the analysis of an X-ray image. An X-ray image provides the internal details of product information, which allow the analysts to detect the presence of any damage due to worms, or other defects caused by non-destructive (non-invasive) methods. Such analysis is highly mandated in agricultural and food industry as worms often contribute to several conditions favoring mold growth or toxin production. Many prior work exist in the literature on the segmentation of agricultural product [Casasent *et al.* (1996), Talukder and Casasent (1998), Talukder *et al.* (1999a,b)]. Most of them are based on watershed transform [Chen *et al.* (2004), Talukder *et al.* (1999a), Vincent and Soille (1991)], mathematical morphology [Iwanowski (2007)], granulometric methods [Vincent (2000)], or their variants.

Among the various image segmentation techniques, the watershed algorithm is a popular segmentation method, which originates from the concept of mathematical morphology [Vincent and Soille (1991)]. This technique has been successfully applied for gray-tone image segmentation in various fields including medicine [Cates *et al.* (2005), Cristoforetti *et al.* (2008), Pratikakis *et al.* (1999)], computer vision [Park *et al.* (2005)], biomedicine [Charles *et al.* (2008), Jalba *et al.* (2004)], signal processing [Leprettre and Martin (2002)], industry [Du and Sun (2006), Malcolm *et al.* (2007)], remote sensing [Hall and Hay (2003), Karantzalos and Argialas (2006)], computer-aided design [Razdan and Bae (2003)], and video coding [Wang (1998a)]. The watershed algorithm has also been applied for colored image segmentation [Jung (2007)]. Recently, an automatic segmentation technique for granular objects using local density clustering and gradient-barrier watershed has been presented by Yang and Ahuja [Yang and Ahuja (2014)].

Another well known technique is the morphological watershed algorithm, which uses distance transforms [Dougherty (1992), Orbert *et al.* (1993)]. The watershed algorithm segments an image into different regions by treating its inverse distance map as a landscape and the local minima as markers. Each of the segmented regions is labeled with a unique index. Different objects can be separated and identified using the indices of the segmented regions. The effective performance of watershed segmentation depends on the selection of local minima or markers. The spurious markers lead to over-segmentation, which is a major drawback of the watershed algorithm. The performance becomes worse when the objects are irregular-shaped, overlapped, or connected, as more spurious local minima tend to occur in the distance transform. Thus, a preprocessing of the markers is needed to improve the performance. Other improved algorithms have been proposed to overcome the over-segmentation issue [Lin *et al.* (2003), Long *et al.* (2007), Umesh Adiga and Chaudhuri (2001)].

Several methods based on concavity analysis in an ensemble of convex objects are also widely used [Farhan *et al.* (2010, 2013), Fernandez *et al.* (1995), Kumar *et al.* (2006), Liang (1989), Wang *et al.* (2012), Wang and Hao (2007), Wang (1998b), Wen *et al.* (2009), Zhong *et al.* (2009)]. These methods aim to locate the concavity points in order to determine the possible split-lines. In other approaches, concavity points are first used to segment the object contour, and then the boundaries of the clumps needed for segmentation are determined by an ellipse-fitting method [Bai *et al.* (2009), Cong and Parvin (2000), Kothari *et al.* (2009)]. However, some of these existing work suffer from various drawbacks as discussed below.

In the method proposed by Kumar et al. [Kumar *et al.* (2006)], all the concavity points in a region consisting multiple concavity may not be detectable. It may also produce long invalid split-lines for a weak concavity point. A directional vector is normally used to determine the orientation of a concavity. However, in many cases, it may not indicate the correct orientation of the concavity. Later, Farhan et al. [Farhan *et al.* (2010)] reported an improved algorithm, but its performance is dependent on the choice of some parameters and it may produce over-splitting results.

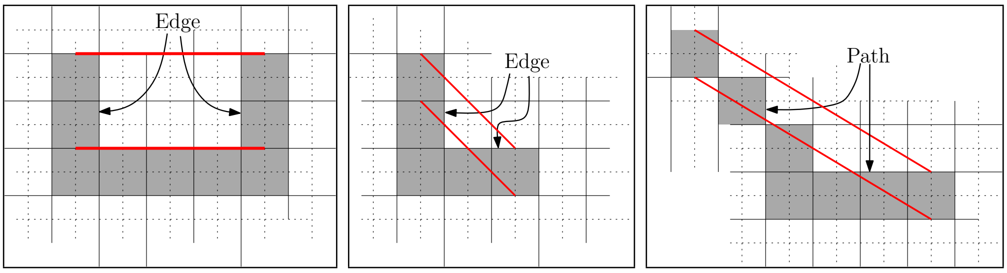
Wang [Wang (1998b)] used a polygonal-approximation technique to smooth the object contour. This may change the geometrical shape of the objects, and hence, may cause the loss of concavity points which have small concavity depths. As a result, some actual split-lines may fail to show up. In order to correct this, it uses the principle that the second concavity point should lie within the extended cone formed by the vertices of the first concavity point. However, this technique may also be misleading in the cases where the angle between the vertices is small or when two or more concavity points are present in the cone. The method applies a morphological operation and a minimum-distance path technique to determine the split-lines (separators). In many instances, it may fail to determine all split-lines correctly.

In Liang [Liang (1989)], the concavity points are detected by thresholding the angle near them and it uses a shortest-path method. This method may produce invalid concavity points when thresholding is applied without considering their depth, and may identify false split-lines.

The method proposed by Wang et al. [Wang *et al.* (2012)] detects the concavity points on the basis of the distance between potential concavity-point-pairs, from inside and along the contour, without considering the concaveness of the region. However, the nearest point may not always qualify as the best pair for a concavity point. Then the method determines a split-path in the intensity patch formed by a rectangular window between the concavity points. Unfortunately, in many instances, such an intensity-based split-path may appear as a curve which tends to go outside the window, thereby producing over- or under-segmentation results.

Recently Farhan et al. [Farhan *et al.* (2013)] proposed a non-parametric concavity-point analysis method for splitting the clumps of convex objects in a binary image. The method finds the concave-pairs by using a variable-size rectangular window along the direction vector. However, this may split the clump into a concave object and requires checking of concavity on the separator line segment. The concave pair is connected with a straight line segment or with a curve that follows a path of minimum or maximum intensity in its gray-scale image. As it uses gray-scale image processing, its time-complexity increases.

The methods based on ellipse-fitting [Bai *et al.* (2009), Cong and Parvin (2000), Kothari *et al.* (2009)] first compute a polygonal approximation of the contour, and then find the concavity points by using the angle between the vertices or the changing angle of tangents to the contour. The contour is segmented using the obtained concavity points. These methods are also computationally complex and parameter-dependent [Kumar *et al.* (2006)]. Moreover, an ellipse-fitting approach may not be applicable to complex clump objects as all the contour segments may not be present therein.



(a) (b) (c)

Figure 6.1: Different type concavity (a) ‘U’-type (b) ‘L’-type (c) ‘V’-type. Any line segment lies in between two red line is not contained in the object.

6.1.2 Our Contribution

This chapter presents a new and efficient algorithm for the segmentation of touching or overlapping convex [Klette and Rosenfeld (2004a)] objects based on digital geometry. We use the concept of outer isothetic cover (OIC) [Biswas *et al.* (2010)] to determine the joining points of the edges of two objects. Next, these joining points are partitioned into suitably matching pairs using the convexity property. The straight line segment that connects a matched pair indicates the separator of two touching or overlapping convex objects, which can be used to isolate them. The advantage of the proposed method lies in the fact that the under- or over-segmentation error is significantly reduced and the required computation is limited to the integer domain only.

6.2 Concavity in $\mathbb{Z}^2$

In order to study the concavity property in $\mathbb{Z}^2$, we start with some definitions from the literature [Gonzalez and Woods (2001), Klette and Rosenfeld (2004a)]. A pixel p is a point in $\mathbb{Z}^2$. Two pixels $p(x_p, y_p)$ and $q(x_q, y_q)$, $p \neq q$, are 4-neighbors of each other if and only if $|x_p - x_q| + |y_p - y_q| = 1$ and 8-neighbors of each other if and only if $\max\{|x_p - x_q|, |y_p - y_q|\} \leq 1$. A sequence of pixels defines a 4-connected (or 8-connected) digital object if between each pair of pixels there exists a 4-connected (or 8-connected) path in the object.

A finite or infinite 8-connected arc is called irreducible iff its set of grid points does not remain 8-connected if a non-endpoint is removed from it. A finite irreducible 8-connected arc is a digital straight segment (DSS) iff it satisfies the chord property. A set M of grid points satisfies the chord property iff, for any two distinct points p and q in M and any point $r(x_r, y_r)$ on the (real) line segment pq, there exists a grid point $t(x_t, y_t) \in M$ such that $\max(|x_r x_t|, |y_r y_t|) < 1$.

A finite set $M \in \mathbb{Z}^2$ is digitally convex if and only if either of the following is true:

1. For all $p, q \in M$, there exists at least one digital straight segment (DSS) that has p and q as end pixels is contained in M.

2. For all $p, q, r \in M$, all of the grid points in the $\triangle pqr$ are in M.

Using above definitions we find certain relationships among the pixels in the concave region of a digital object. Depending on the shape of the concave region we divide them into three classes namely ‘U’-type, ‘L’-type, and ‘V’-type as shown in Fig. 6.1. In an ‘U’-type region, there are three DSS components, two edges and one connector as in Fig. 6.1(a). By definition, it is a concave region. In an ‘L’-type region, there are two DSS which are perpendicular to each other and meet at a corner. By definition, it is also a concave region. In a ‘V’-type concave region, there are two paths consisting of several small DSS. These paths start and end at the corners of the concave region. In Fig. 6.1, the DSS corresponding to each real line segment that lies in the corridor (shown by red lines), is not contained in the corresponding object. Note that a ‘V’-type concavity is more general in nature, which includes the other two as special cases. These special cases often arise among the images of touching or overlapping objects, and because of their inherent property, they can be processed faster than the ‘V’-type. As a result, the performance of the proposed algorithm for locating concave regions is improved significantly.

6.3 Granulometric segmentation based on digital geometry

The proposed segmentation method operates on a binary image, which is either given as input or obtained by processing a gray-tone or a colored image. The input thus consists of a collection of digital images which represent an ensemble of touching or overlapping objects. An example of touching or slightly overlapping objects is shown in Fig. 6.2(a); in such an image, the entire geometric boundary of the individual object is still perceptible and hence the components are easy to segment by an automated method. However, the image of an ensemble as shown in Fig. 6.2(b), consists of fully overlapping objects in

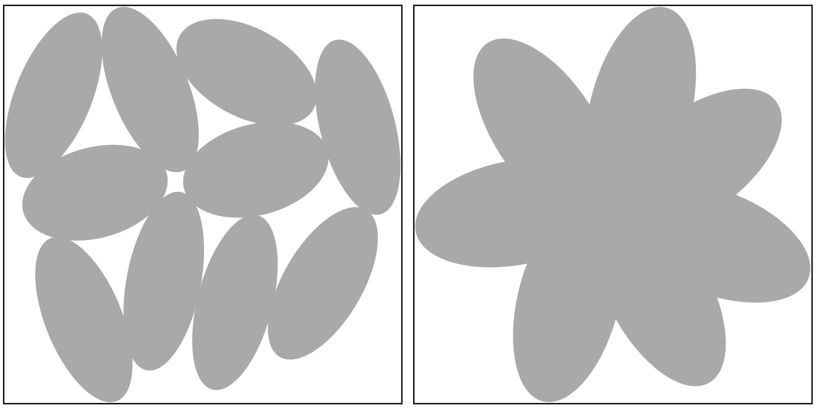

(a) (b)

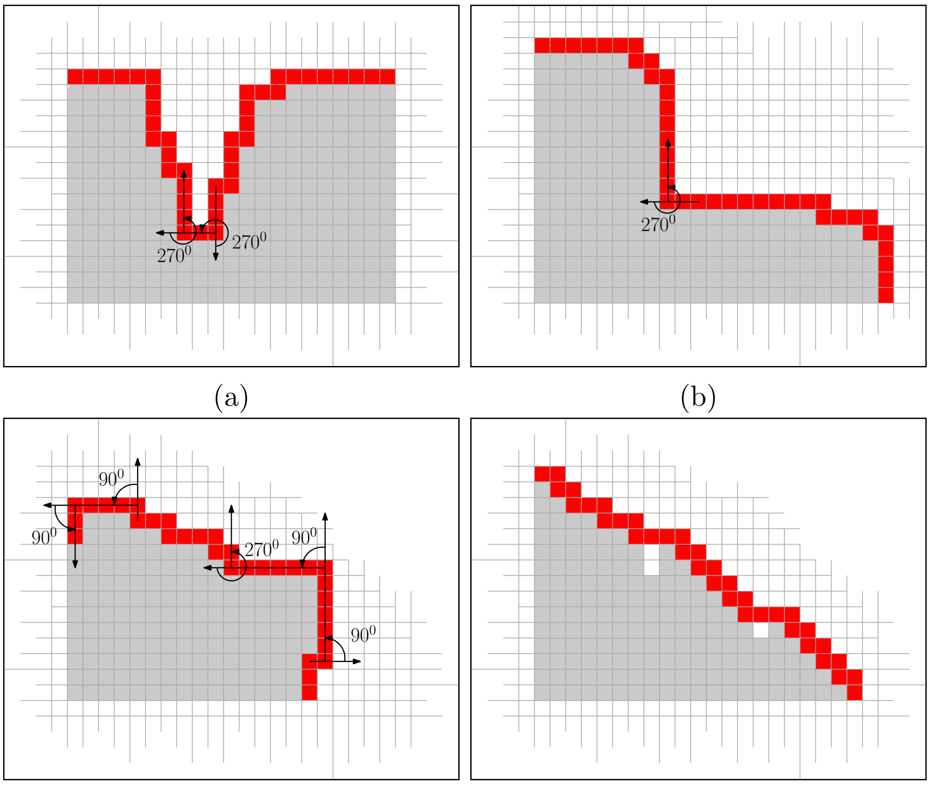
Figure 6.2: Different types of clumped objects (a) touching (b) overlapping.

which the individual boundaries are so occluded that it may be difficult to segment them without any knowledge of the ground truth. Our segmentation method can handle both these two classes of overlapping (i.e., clumped) objects as long as they are convex in shape. It consists of the following steps.

6.3.1 Deriving the Outer Isothetic Cover (OIC)

Let an input binary image $\mathcal{I}$, which consists of several touched and overlapped convex objects, be given. The outer isothetic cover (OIC) [Biswas *et al.* (2010)], which is the minimum-area isothetic polygon perceived on a background grid enclosing $\mathcal{I}$, is first determined. It may be noted that the OIC can be computed for different grid sizes, $g = 1$, 2, 3, $\cdots$, where g (resolution parameter) indicates the horizontal and vertical spacings of the background grid. When $g = 1$, the most accurate approximation of the geometrical shape of the object is captured (highest resolution). Hence, in the proposed algorithm the OIC for $g = 1$ is used for obtaining the best segmentation results. Note that the OIC provides the tightest outer approximation of the object that does not introduce any concavity; rather it removes the smaller concavities from the boundary of the object.

We have classified earlier the concave regions that may appear on the boundary of an object into three classes 'U', 'L', and 'V'. The structural information of the OIC can easily be utilized to identify them. The OIC will have two consecutive 270^0 vertices in an 'U'-type concave region, as shown in Fig. 6.3(a). Such a turn at this type concavity



(a) (b)

(c) (d)

Figure 6.3: Outer Isothetic Cover of concave objects (a) ‘U’-type having consecutive 270^0 vertices (b) ‘L’-type having one 270^0 vertex with edges greater than 2 (c) ‘V’-type having one 270^0 vertex and two paths (d) illustration of a concavity where OIC cannot enter.

is called a ‘U’-turn if it is encountered while traversing the boundary of the OIC [Biswas *et al.* (2010)]. In an ‘L’-type concave region, the boundary of OIC will show one 270^0 vertex with an incident edge of length greater than or equal to 2 grid units as shown in Fig. 6.3(b). The turn at this type of concavity is called an ‘L’-turn. When the OIC encounters one 270^0 vertex with incident edges of length equal to 1, then there will be a ‘V’-type concavity or no concavity. Such concavities can be identified when we observe two consecutive 90^0 vertices or two consecutive 270^0 vertices in both the (forward and backward) directions during the traversal as shown in Fig. 6.3(c). This type of turns is called a ‘V’-turn.

Since the convexity property of an object is migrated to its OIC, we have the following

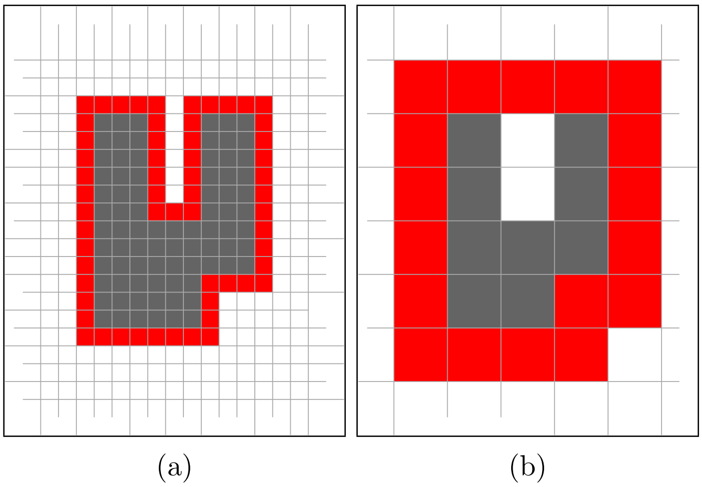


Figure 6.4: OIC of concave objects (a) for $g = 1$ (b) for $g = 3$.

lemma.

Lemma 6.3.1 *The OIC of a digitally convex object is also digitally convex.*

Proof. Assume that the OIC, S, of a convex digital object, $\mathcal{C}$, is not convex. Then S must contain at least one instance of 'U'-type or 'L'-type or 'V'-type region. Thus, the four object occupying cells should be arranged in the form of a 'U'-turn or 'L'-turn or 'V'-turn. Therefore, the object is no longer digitally convex. Hence, the proof follows by contradiction. □

The OIC of a concave object may not be necessarily concave; it depends on the size and the location of the concavity and the grid size g of the OIC. The small concavities of an object may not be reflected in its OIC at higher grid sizes. However, at lower grid size, the concavities of the object may also be present in the OIC. In Fig. 6.4(a), the OIC of a concave object appears as concave, whereas in Fig. 6.4(b), the OIC of the same object becomes convex at larger grid size. Thus, we have the following lemmas.

Lemma 6.3.2 *The OIC of a concave object may or may not be concave.*

Lemma 6.3.3 *If the OIC is concave then the object is concave. However, if the OIC is convex then the object can either be convex or concave.*

Note that if two convex objects overlap, then the resultant object may be either convex or concave. But when two or more circular/elliptic-shaped objects overlap then the resultant object is concave. The above lemma tells that all geometric concavities of two or more overlapped objects may or may not detected by OIC, it depends on the width of the region counted in pixels. We define the term concavity width as follows.

Definition 6.3.1 *The concavity width of a concave region is the minimum chessboard distance between any two pixels on two opposite walls of the region.*

Theorem 6.3.4 *A concavity is detected by OIC if and only if the concavity width of a concave region is greater than or equal to two.*

PROOF. A concavity is detected by OIC if and only if there is an instance of 'U'-type or 'L'-type or 'V'-type region. From the definitions of 'U'-, 'L'- or 'V'-turn, the proof easily follows. □

From the above theorem it is clear that if there is a concavity of width one, OIC cannot detect it. During the construction of an OIC, the traversal procedure does not enter a concave region whose width of the opening is one grid-unit [Biswas *et al.* (2010)]. An illustration of such concave regions is shown in Fig. 6.3(d). Note that this type concavity may appear in an image of convex objects, because of digitization error or some other inaccuracies. However, the proposed OIC-based technique is insensitive to such errors and hence can remove such false concave points during processing. Another point to be noted that the holes are generally concave and hence their covers are either concave or convex.

6.3.2 Identification and management of joining points

A granulometric image $\mathcal{I}$ often contains circular/elliptic-shaped (convex) overlapping objects. When two or more such convex objects touch or overlap, they give rise one or two concavities at the resultant boundary. Since by Lemma 6.3.1 the OIC of a convex object is also digitally convex, it will not contain any of the three concave regions. Let the points, where one of them occludes the other, be termed as *joining points*. Clearly, in the resultant OIC of overlapping convex objects, the concavities, i.e., the joining points will be indicated by a 'U'-turn. Thus, the midpoint of its two consecutive 270^0 vertices will coincide with the joining point. For 'L'-type and 'V'-type turns, the 270^0 vertex itself may be treated as the joining point (as shown in Figs. 6.3(b), (c)). Note that for a touching object with one touch, its OIC will consist of exactly two joining points. Hence, for the collection of objects having n touching-points, the OIC will have exactly $2n$ joining points as shown in Fig. 6.2(a). However, for overlapped objects, there will be a different scenario as some concave regions may be occluded by other objects as shown in Fig. 6.2(b).

The joining points, thus obtained, are stored in a linked list $\mathcal{L}_i$ as the OIC is traversed along its boundary. Each node contains the coordinates of the joining point, (x, y), the pointers to previous and next joining points, and a flag d. The value of the flag, $d = 1$, indicates that the joining point occurs on the OIC, that describes the outer border (called *border-OIC*), and $d = 0$ indicates that the joining point belongs to the OIC forming a hole (called *hole-OIC*) inside the overlapping objects. For each separate OIC, such a linked-list is formed, and finally, the roots and the number of nodes of these lists are stored in another list $\mathcal{L}$.

6.3.3 Determining the matching pairs and the separators

In order to identify the lines of separation (called *separators*) for two overlapping objects, the joining points are to be matched to form suitable matching pairs. If an OIC contains N joining points, there are $\binom{N}{2}$ possible separators. For accurate segmentation, we have to find the real separators among them by discarding the false pairs. Also for overlapping objects, since some joining points are occluded, an inappropriately matched-pair may lead to under- or over-segmentation. The ultimate purpose of identifying the separators is to obtain a correct segmentation. In this context, we have the following lemma.

Lemma 6.3.5 *A line segment connecting two joining points is a real separator of the objects if the separated components are devoid of all concavities on the line segment.*

Depending on the number of joining points on an OIC, the following cases may occur as described below.

No joining point: This indicates that the source is an isolated convex object as shown in Fig. 6.5(a).

One joining point: If an OIC contains only one joining point (shown as a blue dot), it indicates two overlapping convex objects with another occluded joining point as shown in Fig. 6.5(b). In order to construct the separator, we have to insert a dummy joining point (shown in red dot) in the middle of the closed contour as observed from the joining point. Also this dummy point should be at the local maxima or minima, and thus can be obtained by taking the average of two consecutive 90^0 vertices on the contour surrounding

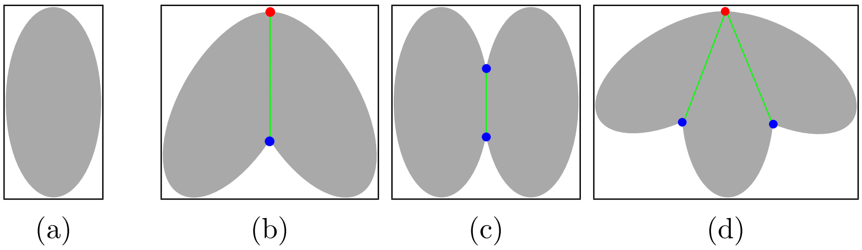


Figure 6.5: Possible overlapping of objects when the OIC contains (a) no joining point, (b) one joining point, (c) and (d) two joining points.

it. The segment that joins these two points will be an actual separator as it divides the object into two convex components.

Two joining points: If an OIC contains two joining points (shown in blue dots), then there are two possibilities: (i) two convex objects touching each other (as in Fig. 6.5(c)), or (ii) three overlapping convex objects (as in Fig. 6.5(d)). In the first case, the line that joins these two points provides the real separator as it divides the object into two convex components. However, in the second case, the connecting line segment does not indicate a separator as it divides the boundary into a convex object and a concave object. In this case, we need to insert a dummy point (shown in red) in the middle of the OIC as before, and draw the two separators as shown in Fig. 6.5(d).

Three joining points: In another case, three joining points may be identified in an OIC (shown a blue dots), which may consist of a hole (Fig. 6.6(a)) or three touching objects as in Fig. 6.6(b). In both the cases, three joining points may appear on the boundary. However, no matching pair among them will indicate a real separator. In this context, we have the following lemma.

Lemma 6.3.6 *No two joining points on the hole-OIC (as defined in Sec. 6.3.2) will form a matching pair.*

PROOF. Let p and q be the two joining points lying on the same hole-OIC, which form a matching pair. Then the line segment that joins them divides the inner hole, and hence, it has no significance as a separator between two overlapping objects. Hence the proof. □

The actual joining points on hole-OIC are replaced by dummy points (as in Fig. 6.6(a)) or a dummy point is inserted at the center of the actual joining points (as in Fig. 6.6(b)).

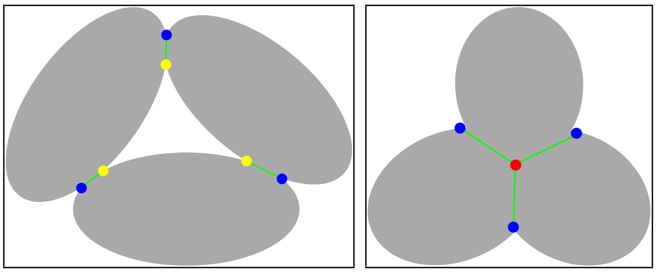

(a) (b)

Figure 6.6: Possible overlapping of objects when an OIC contains three joining points.

The separating line segments can then be obtained by joining the actual joining points to its nearest dummy point in the first case or to the central dummy point in the second case. When more than three objects touch or overlap, a similar technique is used to insert and match dummy vertices with the observed joining points.

6.3.4 Analysis of the segmented objects

The collection of the separators partitions the input image into several regions, where each region corresponds to a convex object. The total count of pixels in each object gives the approximate area of the object. Also, the shape of these regions may be used to analyze the granulometric attributes of the objects.

6.3.5 Demonstration of the proposed method

A step-wise demonstration of the proposed method on a coffee bean image is shown in Fig. 6.7. The input coffee bean image shown in Fig. 6.7(a) contains a numbers of beans colored as black. The outer isothetic covers (both hole-OIC and border-OIC indicated in red) are shown in Fig. 6.7(b). The joining points are marked (shown with blue dots) when an ‘U’- or ‘L’- or ‘V’-turn is observed while deriving the OICs (Fig. 6.7(c)). Each of the joining points of the hole-OIC is paired with its nearest joining point on the border-OIC; a separator is drawn for each matching pair (Fig. 6.7(d)). The remaining joining points are paired with the nearest unpaired joining point and the separators are drawn as shown in Fig. 6.7(e). Figure 6.7(f) shows the original image segmented into its components obtained after removing the isothetic cover and the separators.

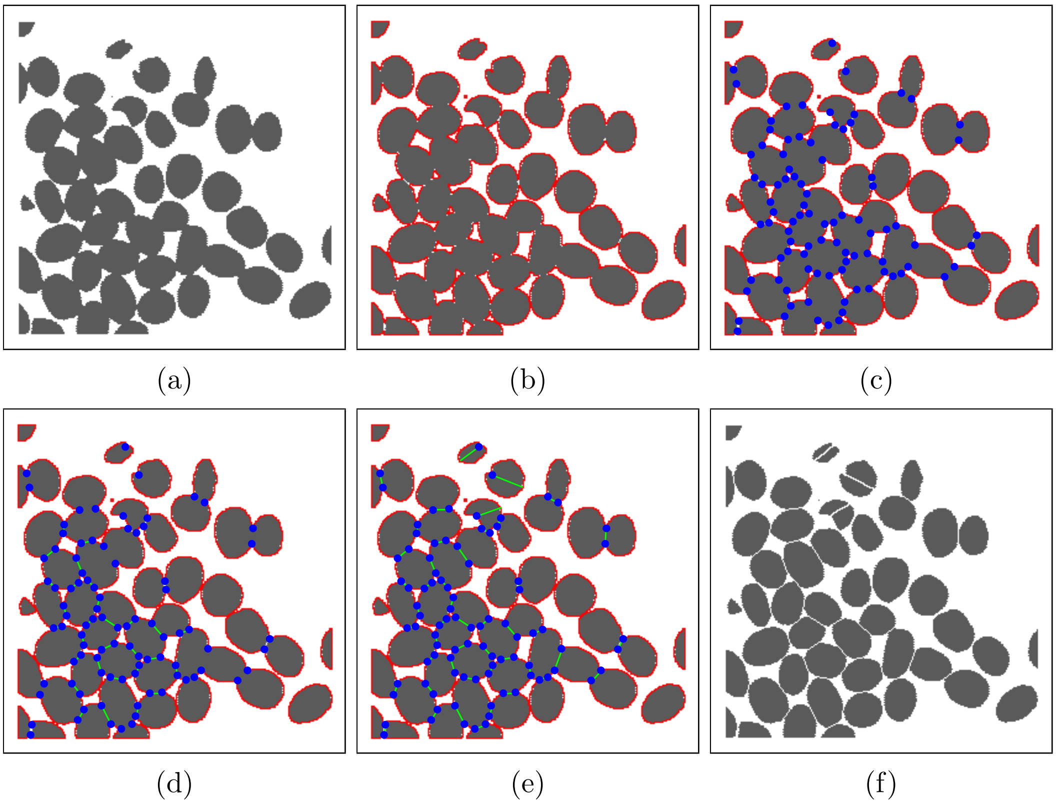
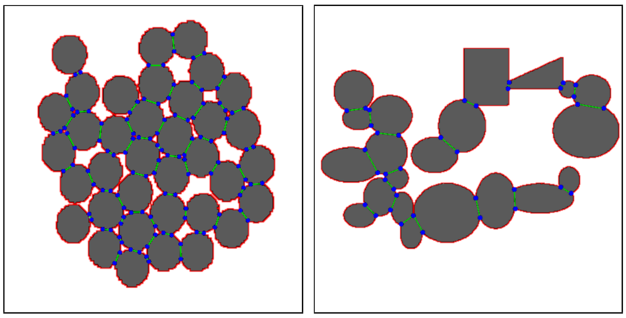


Figure 6.7: Step-wise snapshots of the experiment on a coffee bean image: (a) input image (b) after drawing outer isothetic cover (c) after finding the joining points (d) after joining the matching pair for points on OIC with $d = 0$, (e) drawing other separating lines (f) final result of segmentation

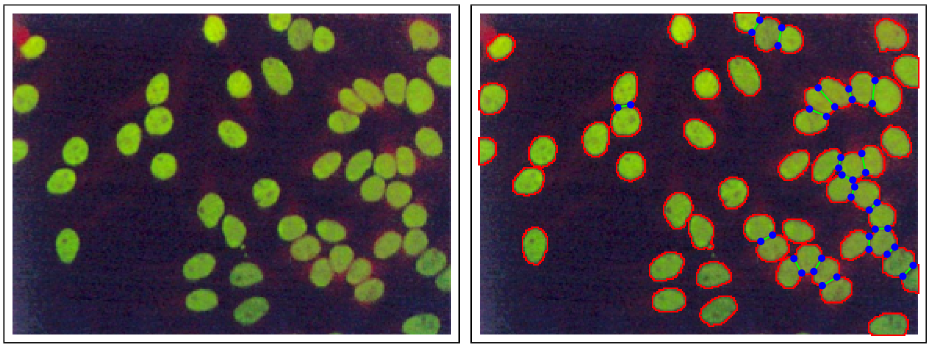
6.4 Experimental Results

The proposed algorithm has been implemented in Matlab version R2011b on the openSUSE™ OS Release 12.3 HP Compaq with Intel® Pentium Dual-Core, 1.7 GHz processor. The algorithm has been tested on several images. The image [Chen *et al.* (2004)] shown in Fig. 6.7(a) consists of 47 coffee beans, in which some of them are broken. The segmented image obtained by the proposed algorithm is shown in Fig. 6.7(f) with 50 separated coffee beans as the three of them have concavities and hence each separated into two. The proposed algorithm is also applied on several other images [Chen *et al.* (2004), Sun and Luo (2009)]. Fig. 6.8(a) shows an image consisting of 36 touching chocolates; Fig. 6.8(b)



(a) (b)

Figure 6.8: Experimental results on chocolate image and synthetic image.



(a) (b)

Figure 6.9: (a) input blood cell image (b) after drawing outer isothetic cover and separation line

shows 21 touching geometrical objects. Experimental results demonstrate that all of them are segmented successfully by the proposed algorithm.

Fig. 6.9(a) shows a blood cell image [Yang and Ahuja (2014)], where the foreground is contrasted against the background using a simple intensity thresholding technique. The proposed algorithm, when run on the foreground image, successfully segments all the 51 blood cells (see Fig. 6.9(b)). An image consisting of eleven coins is shown in Fig. 6.10(a), on which our algorithm is run for $g = 1$ and $g = 2$. In both the cases, correct segmentation is observed as shown in Fig. 6.10(b) and (c). The algorithm also performs very well on high-resolution images, e.g., biscuits and sweets (Fig. 6.11(a) and Fig. 6.11(b)).

We have also run a watershed-based algorithm on these examples. Figure 6.12 shows the segmentation results with different distance transforms (DT) used in the watershed

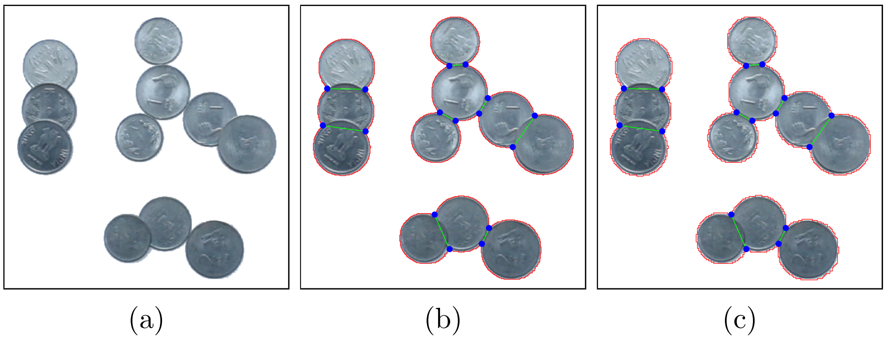


Figure 6.10: (a) input coin image (b) after drawing outer isothetic cover and separation line using $g = 1$ (c) after drawing outer isothetic cover and separation line using $g = 2$

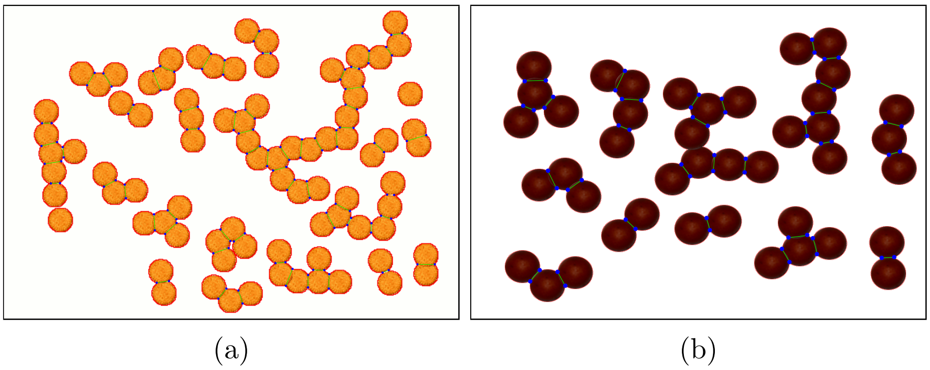


Figure 6.11: Experiment done on images (a) biscuits (b) sweets

segmentation method. We have implemented three distance transforms, namely Euclidean, cityblock, and chessboard [Chen *et al.* (2004)] for the watershed algorithm (see Table 6.1). In most of the cases, we observe that they produce over-segmentation. The proposed algorithm is also run with two values of the resolution parameter ($g = 1, 2$). For $g = 1$, our algorithm produces an exact segmentation. For $g = 2$, it may produce slight under-segmentation in a few cases; however, the execution time is reduced significantly compared to that of the watershed-based algorithms.

6.5 Conclusion

Granular object segmentation plays an important role in crop image analysis and in other diverse areas, where the recognition, counting, or classification of the embedded objects-of-interest is required. In this chapter, we have described an automated technique for

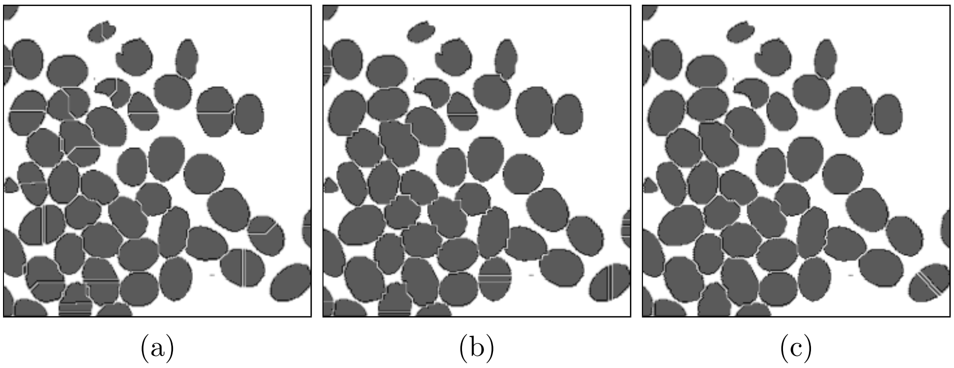

(a) (b) (c)

Figure 6.12: Results of watershed segmentation using three distance transforms (DT) on Fig. 6.7(a): the coffee-bean image (a) Euclidean (b) City block (c) Chessboard.

granulometric object segmentation based on digital-geometric features of the underlying binary image. It is simple and needs only integer-domain computation. The proposed algorithm outperforms the classical watershed-based algorithms in performance as evident from the experimental results.

Table 6.1: Comparative results for images

Image	Size	NOP	NO	Watershed algorithm using						Proposed algorithm			
				Euclidean DT		Cityblock DT		Chessboard DT		OIC (g=1)		OIC (g=2)	
				NDO	T	NDO	T	NDO	T	NDO	T	NDO	T
Fig. 6.7(a)	278×276	29573	47	65	0.0895	48	0.1041	48	0.1129	50	0.1482	48	0.0437
Fig. 6.8(a)	423×411	70397	36	41	0.2054	40	0.2423	41	0.2689	36	0.3329	35	0.1015
Fig. 6.8(b)	390×570	87375	21	45	0.2612	23	0.3029	18	0.3302	21	0.4692	20	0.1443
Fig. 6.9(a)	290×215	12121	51	59	0.2047	50	0.1664	50	0.1491	51	0.3159	51	0.1310
Fig. 6.10(a)	450×447	61141	11	17	0.2445	10	0.2718	10	0.3070	11	0.8015	11	0.2618
Fig. 6.11(a)	1352×1960	630764	83	171	3.0463	85	3.5599	86	3.7713	83	5.6303	83	1.9316
Fig. 6.11(b)	1352×1960	667830	42	123	3.0978	42	3.5666	42	3.8108	42	5.5587	42	1.9133

NOP : Number of object pixels; NO : Number of objects; NDO : Number of detected objects;
T : CPU time in seconds.

Chapter 7

Conclusion

Several new theoretical perspectives and related applications of digital circle, digital disc and digital sphere of revolution have been studied in this thesis. Comparative performances of existing approaches to solving a variety of problems have been investigated, and certain new techniques have been proposed for solving them in the digital space. Applications of digital-geometric techniques to circular arc segmentation, granulometric segmentation, and to shape estimation of nano-scale objects have been demonstrated with theoretical and supporting experimental results.

Combining interdisciplinary paradigms like digital imaging, number theory, and digital geometry, we have formulated and strengthened the proposed methods reported in this thesis. We demonstrate that digital geometry has immense potential in solving various geometric problems in the digital domain, as evident from many interesting applications to image processing and image analysis such as digital arc segmentation, object shape analysis, and convex object segmentation. We speculate that these results will find applications to several other frontiers such as medical imaging and diagnostics, and industrial imaging.

Several open problems have come up following the work undertaken in this thesis. One is finding a closed-form solution of the exact count of absentee-pixels for a digital disc of a given radius as mentioned in Chapter 3. The problem can stated as follows. Given an integer radius r of a digital disc, compute the exact count of absentees in the disc without using a recursion. An extension of this counting problem to a 3D sphere is also open. The problem is to determine a closed-form solution of the exact count of absentee-voxels in a digital sphere of revolution. Also, one can study the implementation challenges of the proposed segmentation and digital ball construction algorithms when a small work space is available as in the case of other digital-image analysis problems with memory constraints such as connected component labeling and erasing [Asano and Bereg (2012), Asano and Kumar (2013)].

Glossary

$\emptyset$	empty set,
$\mathcal{A}$	input SEM image, 5.3
AD	accuracy of detection, 2.4
AHT	adaptive Hough transform, 1.2
α_∞	limiting value of relative count of absentee-pixels w.r.t. disc pixels, as $r \to \infty$, 3.4
α_r	relative count of absentee-pixels w.r.t. disc pixels, 3.4
$\mathcal{A}^{\mathbb{R}}(\alpha, \beta)$	be an arc of $\mathcal{C}^{\mathbb{R}}(o, r)$ having endpoints α and β, 2.2
$\mathcal{A}^{\mathbb{Z}^2}(r)$	set of disc absentees, 3.2
$\mathcal{A}^{\mathbb{Z}^3}(r)$	set of sphere absentees, 4.2
$\mathcal{C}$	a convex digital object, 6.3
CHT	circle Hough transform, 1.2, 2.1
$\mathcal{C}^{\mathbb{R}}$	real circle, 3.2
$\mathcal{C}^{\mathbb{R}}(o, r)$	real circle having center at $o(0,0)$ and radius $r \in \mathbb{Z}^+$, 2.2
$\mathcal{C}^{\mathbb{Z}}$	digital circle, 3.2
$\mathcal{C}^{\mathbb{Z}}(r)$	digital circle of radius r, 1.2
$\mathcal{C}^{\mathbb{Z}}(o, r)$	digital circle of radius r with center at $o(0,0)$, 1.2
$\mathcal{C}^{\mathbb{Z}}_1(r)$	Octant 1 of $\mathcal{C}^{\mathbb{Z}}(r)$, 3.2
$\mathcal{C}^{\mathbb{Z}}_{1,2}(r)$	the first quadrant of $\mathcal{C}^{\mathbb{Z}}(r)$, 4.2
$d(\alpha, \beta)$	the Euclidean distance between the points α and β, 2.2
$\delta_y(r, x)$	y-distance between $\mathcal{C}^{\mathbb{R}}(r)$ and $\mathcal{C}^{\mathbb{R}}(r+1)$ at abscissa x, 3.2
δ_ϕ	angular deviation of a digital arc from its real arc, 2.2
d	a flag variable, 6.3

$\mathcal{D}^{\mathbb{Z}}$	digital disc, 3.2
$\mathcal{D}^{\mathbb{Z}}(r)$	A digital disc having radius $r \in \mathbb{Z}^+$, 1.2, 3.2
DSS	digital straight segment, 6.2
DT	distance transforms, 6.4
$\mathcal{D}^{\mathbb{Z}}_{\cup}(r)$	set of all digital circles centered at o and radius in $\{0, 1, 2, \ldots, r\}$, 4.2
E1	Type I error, 2.4
E2	Type II error, 2.4
$\varepsilon^{r}_{c \in ab}$	circumferential angle relative error at the pixel c lying on the digital circular arc ab of radius r, 2.2
$\mathcal{F}$	set of all the absentees of Octant 1, 3.3
FHT	fast Hough transform, 1.2
F_k	set of all absentees in P_k, 3.3
g	grid resolution parameter, 6.3
HT	Hough Transform, 1.1, 2.1
$\mathcal{H}^{\mathbb{Z}}_{\cup}(r)$	set of all $\mathcal{C}^{\mathbb{Z}}(c, i)$, where $c = (0, j, 0)$ and $(i, j) \in \mathcal{C}^{\mathbb{Z}}_{1,2}(r)$, 4.2
$\mathcal{I}$	input image, 6.3
$I^{(r)}_{r-j}$	interval in which circle pixel lies, 3.2
$J^{(r)}_{r-j}$	interval in which an absentee lies, 3.2
$\mathcal{L}$	a linked list, 6.3
M	finite set, 6.2
m_r	number of half-open parabolic strips intersecting $\mathcal{C}^{\mathbb{Z}}_{1}(r)$, 3.3

n_c	number of circle pixels inside $\mathcal{D}^{\mathbb{Z}}(r)$, 3.4
$N_8(q)$	8-neighbors of pixel q, 3.2
N_c	the number of curve pixels in the original image, 2.4
N_{fa}	the number of false-acceptance pixels, 2.4
N_{fr}	the number of false-rejection pixels, 2.4
N_g	the number of pixels on circular arcs in the ground-truth image, 2.4
N_p	the umber of pixels on circular arcs detected by the proposed algorithm, 2.4
OIC	Outer Isothetic Cover, 1.2, 6.3
ϕ_γ	an angle, 2.2
P_k	half-open parabolic strip, 3.3
$\underline{P}_k$	inner parabolic region, 3.3
$\overline{P}_k$	outer parabolic region, 3.3
PS	porous silicon, 1.2, 5.1
r	radius of a circle, 3.1
$\mathbb{R}^2$	real space, 5.3
RHT	randomized Hough transform, 1.2
S	digital curve segment, 2.3
$\mathcal{S}^{\mathbb{Z}}_{\mathbb{U}}(r)$	sphere (hollow) with absentee, 4.2
$\mathcal{S}^{\mathbb{Z}}(r)$	complete (hollow) sphere of radius r, 4.2
$\mathbf{S}^{\mathbb{Z}}(r)$	complete solid sphere of radius r, 4.2
SEM	scanning electron microscope, 1.2, 5.1
SERS	surface-enhanced Raman scattering, 1.2, 5.1
$\tilde{r}$	radius digital curve, 2.2

$\triangle(a,b,c)$	twice the area of the triangle with vertices a, b, and c, 2.3
$\mathbb{Z}^2$	2D digital space, 1.1, 6.2
$\mathbb{Z}^3$	3D digital space, 1.1
$\mathbb{Z}^2_1$	denote all points in Octant 1 of $\mathbb{Z}^2$, 3.2, 4.2

Bibliography[*]

ANDRES, E. (1994). Discrete circles, rings and spheres. *Computers & Graphics*, **18**, 695–706.

ANDRES, E. AND JACOB, M. A. (1997). The discrete analytical hyperspheres. *IEEE Transactions on Visualization and Computer Graphics*, **3**, 75–86.

ANGLIN, E. J., CHENG, L., FREEMAN, W. R. AND SAILOR, M. J. (2008). Porous silicon in drug delivery devices and materials. *Advanced Drug Delivery Reviews*, **60**, 1266–1277.

ASANO, T. AND BEREG, S. (2012). A new framework for connected components labeling of binary images. In *Combinatorial Image Analysis - 15th International Workshop, IWCIA 2012, Austin, TX, USA, November 28-30, 2012. Proceedings*, 90–102.

ASANO, T. AND KUMAR, R. (2013). A small-space algorithm for removing small connected components from a binary image. *IEICE Transactions*, **96-A**, 1044–1050.

ASANO, T., BRIMKOV, V. E. AND BARNEVA, R. P. (2009). Some theoretical challenges in digital geometry: A perspective. *Discrete Applied Mathematics*, **157**, 3362–3371.

ATHERTON, T. J. AND KERBYSON, D. J. (1999). Size invariant circle detection. *Image and Vision Computing*, **17**, 795–803.

BADLER, N. I. (1977). Disk generators for a raster display device. *Computer Graphics and Image Processing*, **6**, 589–593.

BAI, X., SUN, C. AND ZHOU, F. (2009). Splitting touching cells based on concave points and ellipse fitting. *Pattern Recognition*, **42**, 2434–2446.

BALÁZS, P. (2013). Complexity results for reconstructing binary images with disjoint components from horizontal and vertical projections. *Discrete Applied Mathematics*, **161**, 2224–2235.

BATENBURG, K. J., PALENSTIJN, W. J., BALÁZS, P. AND SIJBERS, J. (2013). Dynamic angle selection in binary tomography. *Computer Vision and Image Understanding*, **117**,

---

[*] **Reference Style:** AUTHORS (year). Title. *Journal/Proceedings name*, volume, pages.

306–318.

Bera, S., Bhowmick, P. and Bhattacharya, B. B. (2010). Detection of circular arcs in a digital image using chord and sagitta properties. *Proc. Eighth International Workshop on Graphics Recognition (GREC 2009), Lecture Notes in Computer Science (LNCS), Springer*, **6020**, 69–80.

Bera, S., Bhattacharya, B. B., Ghoshal, S. and Bandyopadhyay, N. R. (2012). Circularity analysis of nano-scale structures in porous silicon images. *Computational Modelling of Objects Represented in Images III: Fundamentals, Methods and Applications*, 403–408.

Bera, S., Bhowmick, P., Stelldinger, P. and Bhattacharya, B. B. (2013). On covering a digital disc with concentric circles in $\mathbb{Z}^2$. *Theoretical Computer Science*, **506**, 1–16.

Bera, S., Bhowmick, P. and Bhattacharya, B. B. (2014a). A digital-geometric algorithm for generating a complete spherical surface in $\mathbb{Z}^3$. In *Proc. International Conference on Applied Algorithms (ICAA'14)*, vol. 8321 of *LNCS*, 49–61.

Bera, S., Bhowmick, P. and Bhattacharya, B. B. (2014b). On covering a solid sphere with concentric spheres in $\mathbb{Z}^3$. `http://arxiv.org/abs/1411.1395`.

Bera, S., Biswas, A. and Bhattacharya, B. B. (2014c). A fast digital-geometric approach for granulometric image analysis. In *Recent Advances in Information Technology*, vol. 266 of *Advances in Intelligent Systems and Computing*, Springer, 37–47.

Bera, S., Pal, S., Bhowmick, P. and Bhattacharya, B. B. (2014d). On Chord and Sagitta in $\mathbb{Z}^2$: An analysis towards fast and robust circular arc detection. `http://arxiv.org/abs/1411.1668`.

Betty, C. A. (2008). Porous silicon: a resourceful material for nanotechnology. *Recent Pat. Nanotechnol.*, **2**, 128–36.

Bhowmick, P. and Bhattacharya, B. B. (2007). Fast polygonal approximation of digital curves using relaxed straightness properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**, 1590–1602.

Bhowmick, P. and Bhattacharya, B. B. (2008). Number-theoretic interpretation and construction of a digital circle. *Discrete Applied Mathematics*, **156**, 2381–2399.

Bhowmick, P., Bera, S. and Bhattacharya, B. B. (2009). Digital circularity and its applications. In *Proc. Combinatorial Image Analysis, 13th International Workshop, IWCIA 2009,*, vol. 5852 of *LNCS*, Springer, 1–15.

Biggerstaff, R. H. (1972). Three variations in dental arch form estimated by a quadratic equation. *Journal of Dental Research*, **51**, 1509–1509.

Biswas, A., Bhowmick, P. and Bhattacharya, B. B. (2010). Construction of isothetic covers of a digital object: A combinatorial approach. *Journal of Visual Communication and Image Representation*, **21**, 295–310.

Biswas, R. and Bhowmick, P. (2014). On finding spherical geodesic paths and circles in $\mathbb{Z}^3$. In *Proc. Discrete Geometry for Computer Imagery*, vol. 8668 of *LNCS*, Springer, 396–409.

Blinn, J. F. (1987). How many ways can you draw a circle? *IEEE Computer Graphics and Applications*, **7**, 39–44.

Borwein, J. M. and Borwein, P. B. (1988). Ramanujan and pi. *Scientific American*, **258**, 112–117.

Bresenham, J. E. (1977). A linear algorithm for incremental digital display of circular arcs. *Communications of the ACM*, **20**, 100–106.

Bresenham, J. E. (1985). Run length slice algorithm for incremental lines. In R.A. Earnshaw, ed., *Fundamental Algorithms for Computer Graphics*, vol. F17 of *NATO ASI Series*, Springer-Verlag, New York, 59–104.

Brimkov, V. E. and Barneva, R. P. (2002). Graceful planes and lines. *Theoretical Computer Science*, **283**, 151–170.

Brimkov, V. E. and Barneva, R. P. (2008). On the polyhedral complexity of the integer points in a hyperball. *Theoretical Computer Science*, **406**, 24–30.

Brimkov, V. E., Coeurjolly, D. and Klette, R. (2007). Digital planarity — A review. *Discrete Applied Mathematics*, **155**, 468–495.

Brimkov, V. E., Barneva, R. P., Brimkov, B. and Vieilleville, F. (2008). Offset approach to defining 3D digital lines. In *Proceedings of the 4th International Symposium on Advances in Visual Computing*, vol. 5358, Springer-Verlag, 678–687.

Brimkov, V. E., Barneva, R. P. and Brimkov, B. (2011). Connected distance-based rasterization of objects in arbitrary dimension. *Graphical Models*, **73**, 323–334.

Cappell, S. E. and Shaneson, J. L. (2007). Some Problems in Number Theory I: The Circle Problem. `http://arxiv.org/abs/math.NT/0702613`.

Carter, R. M. and Yan, Y. (2005). Measurement of particle shape using digital imaging techniques. In *Journal of Physics: Conference Series*, vol. 15, 177–182.

Casasent, D., Talukdar, A., Cox, W., Chang, H. and Weber, D. (1996). Detection segmentation and pose estimation of multiple touching product inspection items. *SPIE - Photonics East'96, Optics in Agriculture, Forestry, and Biological Processing II*, **2907**, 205–216.

Cates, J. E., Whitaker, R. T. and Jones, G. M. (2005). Case study: an evaluation of user-assisted hierarchical watershed segmentation. *Medical Image Analysis*, **9**, 566–578.

Chamizo, F. (1998). Lattice points in bodies of revolution. *Acta Arithmetica*, **85**, 265–277.

Chamizo, F. and Cristobal, E. (2012). The sphere problem and the L-functions. *Acta Mathematica Hungarica*, **135**, 97–115.

Chamizo, F., Cristóbal, E. and Ubis, A. (2007). Visible lattice points in the sphere. *Journal of Number Theory*, **126**, 200–211.

Chamizo, F., Cristóbal, E. and Ubis, A. (2009). Lattice points in rational ellipsoids. *Journal of Mathematical Analysis and Applications*, **350**, 283–289.

Chan, S., Kwon, S., Koo, T. W., Lee, L. P. and Berlin, A. A. (2003). Surface-enhanced raman scattering of small molecules from silver coated silicon nanopores. *Advanced Materials*, **15**, 1595–1598.

Chan, Y. T. and Thomas, S. M. (1995). Cramer-Rao lower bounds for estimation of a circular arc center and its radius. *Graphical Models and Image Processing*, **57**, 527–532.

Chanda, B. and Majumder, D. D. (2009). *Digital Image Processing and Analysis*. PHI.

Charles, J. J., Kuncheva, L. I., Wells, B. and Lim, I. S. (2008). Object segmentation within microscope images of palynofacies. *Computers & Geosciences*, **34**, 688–698.

Chattopadhyay, S., Das, P. P. and Ghosh-Dastidar, D. (1994). Reconstruction of a digital circle. *Pattern Recognition*, **27**, 1663–1676.

Chen, Q., Yang, X. and Petriu, E. M. (2004). Watershed segmentation for binary images with different distance transforms. In *Proceedings of the 3rd IEEE International Workshop on Haptic, Audio and Visual Environments and Their Applications. HAVE 2004*, 111–116.

Chen, T. C. and Chung, K. L. (2001). An efficient randomized algorithm for detecting circles. *Computer Vision and Image Understanding*, **83**, 172–191.

Chernov, N. and Lesort, C. (2005). Least squares fitting of circles. *Journal of Mathematical Imaging and Vision*, **23**, 239–252.

Chernov, N. I. and Ososkov, G. A. (1984). Effective algorithms for circle fitting. *Computer Physics Communications*, **33**, 329–333.

Chia, A. Y. S., Rahardja, S., Rajan, D. and Leung, M. K. (2011). A split and merge based ellipse detector with self-correcting capability. *IEEE Transactions on Image Processing*, **20**, 1991–2006.

Chia, A. Y. S., Rajan, D., Leung, M. K. and Rahardja, S. (2012). Object recognition by discriminative combinations of line segments, ellipses, and appearance features. *IEEE Trans. Pattern Anal. Mach. Intell.*, **34**, 1758–1772.

Chiu, S. H. and Liaw, J. J. (2005). An effective voting method for circle detection. *Pattern Recognition Letters*, **26**, 121–133.

Christ, T., Pálvölgyi, D. and Stojakovic, M. (2012). Consistent digital line segments. *Discrete & Computational Geometry*, **47**, 691–710.

Chun, J., Korman, M., Nöllenburg, M. and Tokuyama, T. (2009). Consistent digital rays. *Discrete & Computational Geometry*, **42**, 359–378.

Chung, K. L. and Huang, Y. H. (2007). Speed up the computation of randomized algorithms for detecting lines, circles, and ellipses using novel tuning-and LUT-based voting platform. *Applied Mathematics and Computation*, **190**, 132–149.

Chung, W. L. (1977). On circle generation algorithms. *Computer Graphics and Image Processing*, **6**, 196–198.

Coeurjolly, D., Gérard, Y., Reveilles, J. P. and Tougne, L. (2004). An elementary algorithm for digital arc segmentation. *Discrete Applied Mathematics*, **139**, 31–50.

Cong, G. and Parvin, B. (2000). Model-based segmentation of nuclei. *Pattern Recog-*

nition, **33**, 1383–1393.

Crawford, J. F. (1983). A non-iterative method for fitting circular arcs to measured points. *Nuclear Instruments and Methods in Physics Research*, **211**, 223–225.

Cristoforetti, A., Faes, L., Ravelli, F., Centonze, M., Del Greco, M., Antolini, R. and Nollo, G. (2008). Isolation of the left atrial surface from cardiac multi-detector CT images based on marker controlled watershed segmentation. *Medical Engineering & Physics*, **30**, 48–58.

Cullis, A. G. and Canham, L. T. (1991). Visible light emission due to quantum size effects in highly porous crystalline silicon. *Nature*, **353**, 335–338.

Danielsson, P. E. (1978). Comments on circle generator for display devices. *Computer Graphics and Image Processing*, **7**, 300–301.

Davies, E. R. (1987). A high speed algorithm for circular object detection. *Pattern Recognition Letters*, **6**, 323–333.

Davies, E. R. (1988a). A hybrid sequential-parallel approach to accurate circle centre location. *Pattern Recognition Letters*, **7**, 279–290.

Davies, E. R. (1988b). A modified Hough scheme for general circle location. *Pattern Recognition Letters*, **7**, 37–43.

Davies, E. R. (1997). *Machine Vision: Theory, Algorithms and Practicalities*. Academic Press, London.

Doros, M. (1979). Algorithms for generation of discrete circles, rings, and disks. *Computer Graphics and Image Processing*, **10**, 366–371.

Doros, M. (1984). On some properties of the generation of discrete circular arcs on a square grid. *Computer Vision, Graphics, and Image Processing*, **28**, 377–383.

Dougherty, E. R. (1992). An introduction to morphological image processing. *Tutorial texts in Optical Engineering*.

Draine, B. T. and Flatau, P. J. (1994). Discrete dipole approximation for scattering calculations. *Journal of the Optical Society of America A*, **11**, 1491–1499.

Du, C. J. and Sun, D. W. (2006). Automatic measurement of pores and porosity in pork ham and their correlations with processing time, water content and texture. *Meat Science*, **72**, 294–302.

DUDA, R. O. AND HART, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, **15**, 11–15.

ERDOS, P. (1988). On the irrationality of certain series: problems and results. *New advances in Transcendence theory edited by A. Baker, Cambridge University Press*.

EWELL, J. A. (2000). Counting lattice points on spheres. *The Mathematical Intelligencer*, **22**, 51–53.

FARHAN, M., YLI-HARJA, O. AND NIEMISTÖ, A. (2010). An improved clump splitting method for convex objects. In *Proceedings of the 7th International Workshop on Computational Systems Biology*, 35–38.

FARHAN, M., YLI-HARJA, O. AND NIEMISTÖ, A. (2013). A novel method for splitting clumps of convex objects incorporating image intensity and using rectangular window-based concavity point-pair search. *Pattern Recognition*, **46**, 741–751.

FERNANDEZ, G., KUNT, M. AND ZRŸD, J. P. (1995). A new plant cell image segmentation algorithm. In *Proc. Image Analysis and Processing*, vol. 974 of *LNCS*, Springer, 229–234.

FESCHET, F. AND REVEILLÈS, J. P. (2006). A generic approach for n-dimensional digital lines. In *Proceedings of the 13th international conference on Discrete Geometry for Computer Imagery*, DGCI'06, Springer-Verlag, Berlin, Heidelberg, 29–40.

FESCHET, F. AND TOUGNE, L. (2005). On the min DSS problem of closed discrete curves. *Discrete Applied Mathematics*, **151**, 138–153.

FIORIO, C. AND TOUTANT, J. L. (2006). Arithmetic discrete hyperspheres and separatingness. In *Proceedings of the 13th international conference on Discrete Geometry for Computer Imagery*, DGCI'06, Springer-Verlag, Berlin, Heidelberg, 425–436.

FIORIO, C., JAMET, D. AND TOUTANT, J. L. (2006). Discrete circles: an arithmetical approach with non-constant thickness. In A.Y.W. Longin Jean Latecki David M. Mount, ed., *Vision Geometry XIV, Electronic Imaging, SPIE*, vol. 6066, San Jose (CA), USA, 60660C–1.60660C–12.

FOLEY, J. D., DAM, A. v., FEINER, S. K. AND HUGHES, J. F. (1993). *Computer Graphics — Principles and Practice*. Addison-Wesley, Reading (Mass.).

FOMENKO, O. M. (2002). Distribution of lattice points over the four-dimensional sphere.

Journal of Mathematical Sciences, **110**, 3164–3170.

Fukshansky, L., Henshaw, G., Liao, P., Prince, M., Sun, X. and Whitehead, S. (2012). On integral well-rounded lattices in the plane. *Discrete & Computational Geometry*, **48**, 735–748.

Ghoshal, S., Mitra, D., Roy, S. and Majumder, D. D. (2010). Biosensors and biochips for nanomedical applications: a review. *Sensors and Transducers*, **113**, 1–17.

Ghoshal, S., Ansar, A. A., Raja, S. O., Jana, A., Bandyopadhyay, N. R., Dasgupta, A. K. and Ray, M. (2011). Superparamagnetic iron oxide nanoparticle attachment on array of micro test tubes and microbeakers formed on p-type silicon substrate for biosensor applications. *Nanoscale Research Letters*, **6**, 540.

Gonzalez, R. C. and Woods, R. E. (2001). *Digital Image Processing*. Prentice Hall, second edition.

Gopalakrishnan, V., Hu, Y. and Rajan, D. (2010). Random walks on graphs for salient object detection in images. *IEEE Transactions on Image Processing*, **19**, 3232–3242.

Granitzer, P. and Rumpf, K. (2010). Porous silicona versatile host material. *Materials*, **3**, 943–998.

GREC (2007). Seventh IAPR international workshop on Graphics RECognition (GREC 2007). `http://www.iupr.org/arcseg2007` (not available now).

GREC (2013). Tenth IAPR international workshop on Graphics RECognition (GREC 2013). `http://grec2013.loria.fr/GREC2013/node/22`.

Guil, N., Villalba, J. and Zapata, E. L. (1995). A fast hough transform for segment detection. *IEEE Transactions on Image Processing*, **4**, 1541–1548.

Hall, O. and Hay, G. J. (2003). A multiscale object-specific approach to digital change detection. *International Journal of Applied Earth Observation and Geoinformation*, **4**, 311–327.

Hantos, N. and Balázs, P. (2013). The reconstruction of polyominoes from horizontal and vertical projections and morphological skeleton is np-complete. *Fundam. Inform.*, **125**, 343–359.

Haralick, R. M. (1974). A measure for circularity of digital figures. *IEEE Trans. Sys.,* *Man & Cybern.*, **4**, 394–396.

Harraz, F. A., Kamada, K., Kobayashi, K., Sakka, T. and Ogata, Y. H. (2005). Random macropore formation in p-type silicon in HF-containing organic solutions. *J. Electrochem. Soc.*, **152**, C213–C220.

Heath-Brown, D. R. (1999). Lattice points in the sphere. In *Proc. Intern. Conf. on Number Theory, Number Theory in Progress*, vol. 2, 883–892.

Ho, C. T. and Chen, L. H. (1995). A fast ellipse/circle detector using geometric symmetry. *Pattern Recognition*, **28**, 117–124.

Honsberger, R. (1973). Circles, Squares, and Lattice Points. *Mathematical Gems*, **I**, 117–127.

Horn, B. K. P. (1976). Circle generators for display devices. *Computer Graphics and Image Processing*, **5**, 280–288.

Hsu, S. Y., Chow, L. R. and Liu, C. H. (1993). A new approach for the generation of circles. *Computer Graphics Forum 12*, **2**, 105–109.

Illingworth, J. and Kittler, J. (1987). The adaptive Hough transform. *IEEE Trans. PAMI*, **9**, 690–698.

Illingworth, J. and Kittler, J. (1988). A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing*, **44**, 87–116.

Iwanowski, M. (2007). Morphological boundary pixel classification. In *Proceedings of the International Conference on "Computer as a Tool", EUROCON, 2007*, IEEE, 146–150.

Jalba, A. C., Wilkinson, M. H. and Roerdink, J. B. (2004). Automatic segmentation of diatom images for classification. *Microscopy Research and Technique*, **65**, 72–85.

Jia, L., Peng, C., Liu, H. M. and Wang, Z. H. (2011). A fast randomized circle detection algorithm. In *Proceedings of the 4th International Congress on Image and Signal Processing (CISP), 2011*, vol. 2, IEEE, 820–823.

Jiao, Y., Koktysh, D. S., Phambu, N. and Weiss, S. M. (2010). Dual-mode sensing platform based on colloidal gold functionalized porous silicon. *Appl. Phys. Lett.*, **97**, 153125–153127.

Jung, C. R. (2007). Unsupervised multiscale segmentation of color images. *Pattern Recognition Letters*, **28**, 523–533.

KARANTZALOS, K. AND ARGIALAS, D. (2006). Improving edge detection and watershed segmentation with anisotropic diffusion and morphological levellings. *International Journal of Remote Sensing*, **27**, 5427–5434.

KASA, I. (1976). A circle fitting procedure and its error analysis. *IEEE Transactions on Instrumentation and Measurement*, **25**, 8–14.

KEAGY, P. M. AND SCHATZKI, T. F. (1993). Machine recognition of weevil damage in wheat radiographs. In *Proc. SPIE*, vol. 1836, 108–119.

KEAGY, P. M., PARVIN, B. AND SCHATZKI, T. F. (1996). Machine recognition of navel orange worm damage in X-ray images of pistachio nuts. *LWT-Food Science and Technology*, **29**, 140–145.

KENMOCHI, Y., BUZER, L., SUGIMOTO, A. AND SHIMIZU, I. (2008). Digital planar surface segmentation using local geometric patterns. In *Proceedings of the 14th IAPR international conference on Discrete Geometry for Computer Imagery*, DGCI'08, Springer-Verlag, 322–333.

KIM, H. S. AND KIM, J. H. (2001). A two-step circle detection algorithm from the intersecting chords. *Pattern Recognition Letters*, **22**, 787–798.

KIMME, C., BALLARD, D. AND SKLANSKY, J. (1975). Finding circles by an array of accumulators. *Communications of the ACM*, **18**, 120–122.

KLETTE, R. AND ROSENFELD, A. (2004a). *Digital Geometry: Geometric Methods for Digital Picture Analysis*. Morgan Kaufmann Series in Computer Graphics and Geometric Modeling, Morgan Kaufmann, San Francisco.

KLETTE, R. AND ROSENFELD, A. (2004b). Digital straightness–a review. *Discrete Applied Mathematics*, **139**, 197–230.

KOLESNIKOV, A. (2012). Segmentation and multi-model approximation of digital curves. *Pattern Recognition Letters*, **33**, 1171–1179.

KOLESNIKOV, A. AND KAURANNE, T. (2014). Unsupervised segmentation and approximation of digital curves with rate-distortion curve modeling. *Pattern Recognition*, **47**, 623–633.

KOTHARI, S., CHAUDRY, Q. AND WANG, M. D. (2009). Automated cell counting and cluster segmentation using concavity detection and ellipse fitting techniques. In *Pro-*

ceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2009. ISBI'09, IEEE, 795–798.

Kühleitner, M. (2000). On lattice points in rational ellipsoids: An omega estimate for the error term. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, **70**, 105–111.

Kulpa, Z. (1979). A note on "circle generator for display devices". *Computer Graphics and Image Processing*, **9**, 102–103.

Kulpa, Z. and Kruse, B. (1983). Algorithms for circular propagation in discrete images. *Computer Vision, Graphics, and Image Processing*, **24**, 305–328.

Kumar, G., Sharma, N. K. and Bhowmick, P. (2010). Creating wheel-thrown potteries in digital space. In *Proc. International Conference on Arts and Technology (ArtsIT2009)*, vol. 30 of *Lecture Notes of ICST (LNICST) series*, Springer, 181–189.

Kumar, G., Sharma, N. and Bhowmick, P. (2011). Wheel-throwing in digital space using number-theoretic approach. *International Journal of Arts and Technology*, **4**, 196–215.

Kumar, S., Ong, S. H., Ranganath, S., Ong, T. C. and Chew, F. T. (2006). A rule-based approach for robust clump splitting. *Pattern Recognition*, **39**, 1088–1098.

Lange, L. J. (1999). An elegant continued fraction for π. *The American Mathematical Monthly*, **106**, 456–458.

Leavers, V. (1993). Survey: Which Hough transform? *CVGIP: Image Understanding*, **58**, 250–264.

Lehmann, V. (2003). Trends in fabrication and applications of macroporous silicon. *Physica Status Solidi (a)*, **197**, 13–15.

Leprettre, B. and Martin, N. (2002). Extraction of pertinent subsets from time–frequency representations for detection and recognition purposes. *Signal Processing*, **82**, 229–238.

Li, H., Lavin, M. A. and Le Master, R. J. (1986). Fast hough transform: A hierarchical approach. *Computer Vision, Graphics, and Image Processing*, **36**, 139–161.

Liang, J. (1989). Intelligent splitting in the chromosome domain. *Pattern Recognition*, **22**, 519–532.

Lin, G., Adiga, U., Olson, K., Guzowski, J. F., Barnes, C. A. and Roysam, B. (2003). A hybrid 3D watershed algorithm incorporating gradient cues and object models for automatic segmentation of nuclei in confocal image stacks. *Cytometry Part A*, **56**, 23–36.

Lin, V. S. Y., Motesharei, K., Dancil, K. P. S., Sailor, M. J. and Ghadiri, M. R. (1997). A porous silicon-based optical interferometric biosensor. *Science*, **278**, 840–843.

Long, F., Peng, H. and Myers, E. (2007). Automatic segmentation of nuclei in 3D microscopy images of c. elegans. In *Proceedings of the 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007*, IEEE, 536–539.

Maehara, H. (2010). On a sphere that passes through n lattice points. *European Journal of Combinatorics*, **31**, 617–621.

Magyar, A. (2007). On the distribution of lattice points on spheres and level surfaces of polynomials. *Journal of Number Theory*, **122**, 69–83.

Malcolm, A. A., Leong, H. Y., Spowage, A. C. and Shacklock, A. P. (2007). Image segmentation and analysis for porosity measurement. *Journal of Materials Processing Technology*, **192–193**, 391–396.

Malladi, R., Sethian, J. A. and Vemuri, B. C. (1995). Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**, 158–175.

McLaughlin, R. A. and Alder, M. D. (1998). The Hough transform versus the Up-Write. *IEEE Trans. PAMI*, **20**, 396–400.

Mcllroy, M. D. (1983). Best approximate circles on integer grids. *ACM Trans. Graphics*, **2**, 237–263.

Mignosi, F. (1991). On the number of factors of Sturmian words. *Theoretical Computer Science*, **82**, 71–84.

Montani, C. and Scopigno, R. (1990). Graphics gems (Chapter: *Spheres-to-voxels conversion*), A. S. Glassner (Ed.). Academic Press Professional, Inc., San Diego, CA, USA, 327–334.

Nagao, T. (1993). Pattern matching of binary shapes using a genetic method. *The Trans. of the IEICE*, **76**, 557565.

NAGY, B. (2004). Characterization of digital circles in triangular grid. *Pattern Recognition Letters*, **25**, 1231–1242.

NAGY, B. AND STRAND, R. (2011). Approximating euclidean circles by neighbourhood sequences in a hexagonal grid. *Theoretical Computer Science*, **412**, 1364–1377.

NAKAMURA, A. AND AIZAWA, K. (1984). Digital circles. *Computer Vision, Graphics, and Image Processing*, **26**, 242–255.

NGUYEN, T. AND DEBLED-RENNESSON, I. (2011). Decomposition of a curve into arcs and line segments based on dominant point detection. In *Proc. 17th Scandinavian Conference on Image Analysis (SCIA 2011)*, vol. 6688 of *LNCS*, Springer Berlin Heidelberg, 794–805.

ORBERT, C. L., BENGTSSON, E. W. AND NORDIN, B. G. (1993). Watershed segmentation of binary images using distance transformations. In *Proc. SPIE*, vol. 1902, 159–170.

PAL, S. AND BHOWMICK, P. (2012). Determining digital circularity using integer intervals. *Journal of Mathematical Imaging and Vision*, **42**, 1–24.

PARK, S. C., LIM, S. H., SIN, B. K. AND LEE, S. W. (2005). Tracking non-rigid objects using probabilistic Hausdorff distance matching. *Pattern Recognition*, **38**, 2373–2384.

PATON, K. (1970). Conic sections in chromosome analysis. *Pattern Recognition*, **2**, 39–51.

PITTEWAY, M. L. V. (1974). Integer circles, etc. — Some further thoughts. *Computer Graphics and Image Processing*, **3**, 262–265.

PLA, F. (1996). Recognition of partial circular shapes from segmented contours. *Computer Vision and Image Understanding*, **63**, 334–343.

PRATIKAKIS, I. E., SAHLI, H. AND CORNELIS, J. (1999). Low level image partitioning guided by the gradient watershed hierarchy. *Signal Processing*, **75**, 173–195.

PRATT, V. (1987). Direct least-squares fitting of algebraic surfaces. In *ACM SIGGRAPH*, vol. 21, ACM, 145–152.

RAD, A. A., FAEZ, K. AND QARAGOZLOU, N. (2003). Fast circle detection using gradient pair vectors. In *Proc. DICTA*, 879–887.

RAZDAN, A. AND BAE, M. (2003). A hybrid approach to feature segmentation of triangle meshes. *Computer-Aided Design*, **35**, 783–789.

REDDY, R., CHADHA, A. AND BHATTACHARYA, E. (2001). Porous silicon based potentiometric triglyceride biosensor. *Biosensors and Bioelectronics*, **16**, 313–317.

Rosin, P. L. (1997). Techniques for assessing polygonal approximations of curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**, 659–666.

Rosin, P. L. and West, G. A. W. (1988). Detection of circular arcs in images. In *Proc. 4th. Alvey Vision Conf., Manchester*, 259–263.

Rouquerol, J., Avnir, D., Fairbridge, C. W., Everett, D. H., Haynes, J. M., Pernicore, N., Ramsey, J. D. F., Sing, K. S. W. and Unger, K. K. (1994). Recommendations for the characterization of porous solids. *Pure and Applied Chemistry*, **66**, 1739–1758.

Rucklidge, W. (1997). Efficiently locating objects using the Hausdorff distance. *International Journal of Computer Vision*, **24**, 251–270.

Saha, H., Dey, S., Pramanik, C., Das, J. and Islam, T. (2006). Porous silicon-based smart sensors. *Encyclopedia of Sensors, American Scientific Publishers*, **8**, 163–196.

Salonen, J., Kaukonen, A. M., Hirvonen, J. and Lehto, V. P. (2008). Mesoporous silicon in drug delivery applications. *J. Pharmaceutical Sci.*, **97**, 632–653.

Schatzki, T. F. and Wong, R. Y. (1989). Detection of submilligram inclusions of heavy metals in processed foods. *Food Technology (USA)*, 72–76.

Schatzki, T. F., Witt, S. C., Wilkins, D. E. and Lenker, D. H. (1981). Characterization of growing lettuce from density contours–I. Head selection. *Pattern Recognition*, **13**, 333–340.

Schatzki, T. F., Haff, R. P., Young, R., Can, I., Le, L. C. and Toyofuku, N. (1997). Defect detection in apples by means of X-ray imaging. *Transactions of the ASABE*, **40**, 1407–1415.

Sethian, J. A. (1996). *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Material Science*. Cambridge University Press, UK.

Shimizu, K. (1981). Algorithm for generating a digital circle on a triangular grid. *Computer Graphics and Image Processing*, **15**, 401–402.

Simmons, C., Maguire, D. and Phelps, N. (2010). *Manual of Engineering Drawing (Technical Product Specification and Documentation to British and International Standards, 3rd Edition)*. Elsevier.

Sivignon, I. and Coeurjolly, D. (2009). Minimum decomposition of a digital surface into digital plane segments is NP-hard. *Discrete Applied Mathematics*, **157**, 558–570.

Sonka, M., Hlavac, V. and Boyle, R. (1998). *Image processing, analysis, and machine vision*. PWS, New York.

Stelldinger, P. (2007). *Image Digitization and its Influence on Shape Properties in Finite Dimensions*. IOS Press.

Stewart, M. P. and Buriak, J. M. (2000). Chemical and biological applications of porous silicon technology. *Advanced Materials*, **12**, 859–869.

Suenaga, Y., Kamae, T. and Kobayashi, T. (1979). A high speed algorithm for the generation of straight lines and circular arcs. *IEEE Trans. Comput.*, **28**, 728–736.

Sun, H. Q. and Luo, Y. J. (2009). Adaptive watershed segmentation of binary particle image. *Journal of Microscopy*, **233**, 326–330.

Talukder, A. and Casasent, D. P. (1998). Classification and pose estimation of objects using nonlinear features. In *Proceedings, SPIE*, vol. 3390, 12–23.

Talukder, A., Casasent, D., Lee, H. W., Keagy, P. M. and Schatzki, T. F. (1999a). Modified binary watershed algorithm for segmentation of X-ray agricultural products. In *Proceedings, SPIE, 3543, Precision Agriculture and Biological Quality, 53*.

Talukder, A., Casasent, D., Lee, H. W., Keagy, P. M. and Schatzki, T. F. (1999b). New feature extraction method for classification of agricultural products from X-ray images. In *Proceedings, SPIE*, vol. 3543 of *Precision Agriculture and Biological Quality*, 119–130.

Thomas, S. M. and Chan, Y. T. (1989). A simple approach for the estimation of circular arc center and its radius. *Computer Vision, Graphics, and Image Processing*, **45**, 362–370.

Toutant, J. L., Andres, E. and Roussillon, T. (2013). Digital circles, spheres and hyperspheres: From morphological models to analytical characterizations and topological properties. *Discrete Applied Mathematics*, **161**, 2662–2677.

Tsang, K. M. (2000). Counting lattice points in the sphere. *Bulletin of the London Mathematical Society*, **32**, 679–688.

Umesh Adiga, P. S. and Chaudhuri, B. B. (2001). An efficient method based on watershed and rule-based merging for segmentation of 3-D histo-pathological images. *Pattern Recognition*, **34**, 1449–1458.

Varga, L., Nyúl, L. G., Nagy, A. and Balázs, P. (2014). Local and global uncertainty in binary tomographic reconstruction. *Computer Vision and Image Understanding*, **129**, 52–62.

Vincent, L. (2000). Fast granulometric methods for the extraction of global image information. In *Proceedings, 11th Annual Symposium of the South African Pattern Recognition Association*, 119–140.

Vincent, L. and Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**, 583–598.

Vyatkin, A., Starkov, V., Tzeitlin, V., Presting, H., Konle, J. and Konig, U. (2002). Random and ordered macropore formation in p-type silicon. *J. Electrochem. Soc.*, **149**, G70–G76.

Wall, K. and Danielsson, P. E. (1984). A fast sequential method for polygonal approximation of digitized curves. *Computer Vision, Graphics, and Image Processing*, **28**, 220–227.

Wang, D. (1998a). Unsupervised video segmentation based on watersheds and temporal tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, **8**, 539–546.

Wang, H., Zhang, H. and Ray, N. (2012). Clump splitting via bottleneck detection and shape classification. *Pattern Recognition*, **45**, 2780–2787.

Wang, W. and Hao, S. (2007). Cell cluster image segmentation on form analysis. In *Proceedings of the third International Conference on Natural Computation, 2007. ICNC 2007*, vol. 4, IEEE, 833–836.

Wang, W. X. (1998b). Binary image segmentation of aggregates based on polygonal approximation and classification of concavities. *Pattern Recognition*, **31**, 1503–1524.

Weisstein, E. W. (a). “Chord”. From MathWorld–A Wolfram web resource. `http://mathworld.wolfram.com/Chord.html`.

Weisstein, E. W. (b). “Sagitta”. From MathWorld—A Wolfram web resource. `http://mathworld.wolfram.com/Sagitta.html`.

WEN, Q., CHANG, H. AND PARVIN, B. (2009). A delaunay triangulation approach for segmenting clumps of nuclei. In *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2009. ISBI'09*, IEEE, 9–12.

WIEDERHOLD, P. AND VILLAFUERTE, M. (2011). A new algorithm for triangulation from cross sections and its application to surface area estimation. *Int. J. Imaging Systems and Technology*, **21**, 58–66.

WOO, D. M., HAN, S. S., PARK, D. C. AND NGUYEN, Q. D. (2008). Extraction of 3D Line Segment Using Digital Elevation Data. In *Proceedings of the 2008 Congress on Image and Signal Processing*, vol. 2 of *CISP'08*, IEEE Computer Society, 734–738.

WORRING, M. AND SMEULDERS, A. W. M. (1995). Digitized circular arcs: Characterization and parameter estimation. *IEEE Trans. PAMI*, **17**, 587–598.

WRIGHT, W. E. (1990). Parallelization of Bresenham's line and circle algorithms. *IEEE Computer Graphics and Applications*, **10**, 60–67.

WU, X. AND ROKNE, J. G. (1987). Double-step incremental generation of lines and circles. *Computer Vision, Graphics, and Image Processing*, **37**, 331–344.

XU, L. AND OJA, E. (1993). Randomized Hough transform (RHT): Basic mechanisms, algorithms, and computational complexities. *CVGIP*, **57**, 131–154.

XU, L., OJA, E. AND KULTANEN, P. (1990). A new curve detection method: randomized Hough transform (RHT). *Pattern Recognition Letters*, **11**, 331–338.

YANG, H. AND AHUJA, N. (2014). Automatic segmentation of granular objects in images: Combining local density clustering and gradient-barrier watershed. *Pattern Recognition*, **47**, 2266–2279.

YAO, C. AND ROKNE, J. G. (1995). Hybrid scan-conversion of circles. *IEEE Trans. Visualization and Computer Graphics*, **1**, 311–318.

YIP, R., TAM, P. AND LEUNG, D. (1992). Modification of Hough transform for circles and ellipses detection using a 2-dimensional array. *Pattern Recognition*, **25**, 1007–1022.

YUEN, P. C. AND FENG, G. C. (1996). A novel method for parameter estimation of digital arc. *Pattern Recognition Letters*, **17**, 929–938.

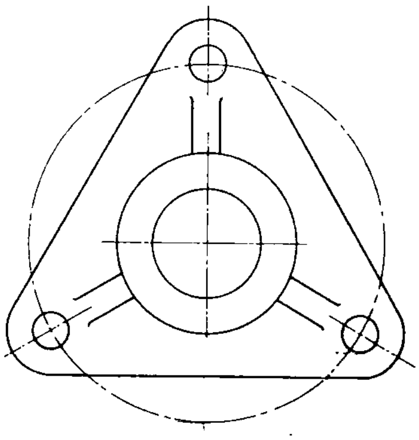
ZHONG, Q., ZHOU, P., YAO, Q. AND MAO, K. (2009). A novel segmentation algorithm for clustered slender-particles. *Computers and Electronics in Agriculture*, **69**, 118–127.

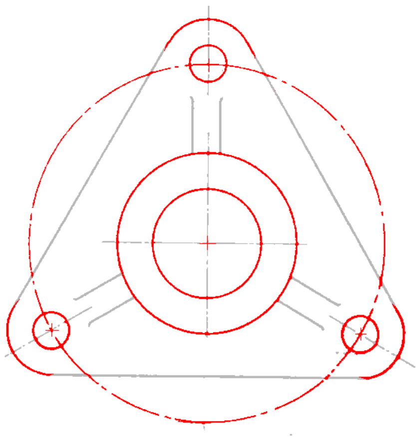
Zubko, E., Petrov, D., Grynko, Y., Shkuratov, Y., Okamoto, H., Muinonen, K., Nousiainen, T., Kimura, H., Yamamoto, T. and Videen, G. (2010). Validity criteria of the discrete dipole approximation. *Applied Optics*, **49**, 1267–1279.

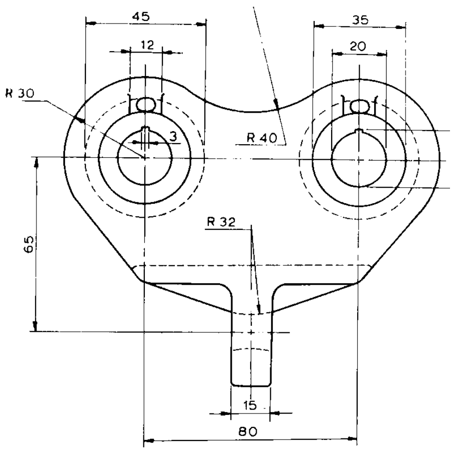
Appendix 1: Results of arc segmentation by CSA

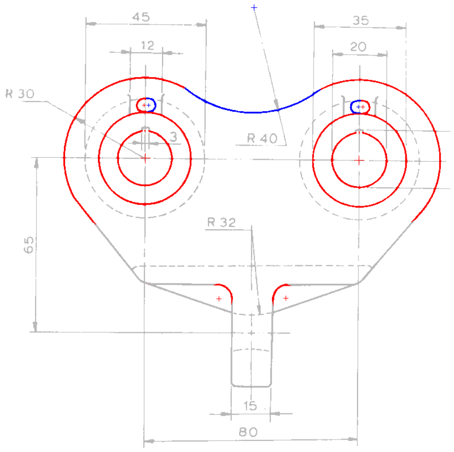
Results (input in black, output colored) of arc segmentation by CSA (Chapter 2) on some images from different dataset.

Results on GREC2013

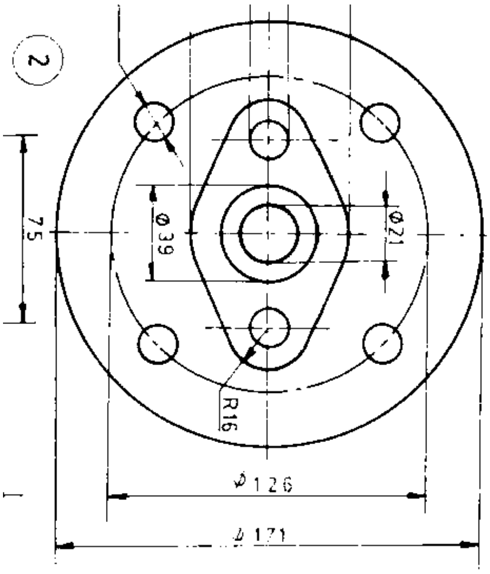


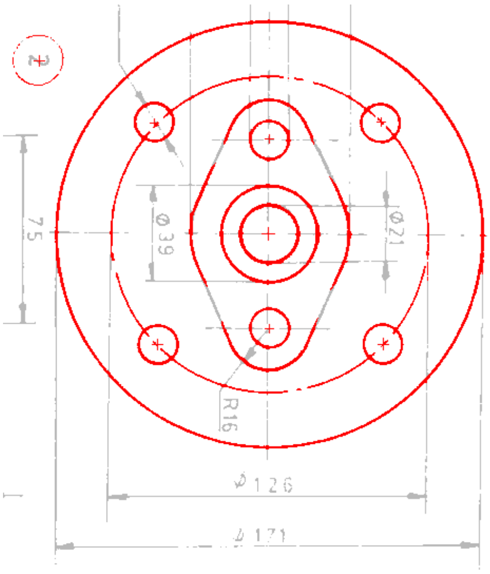


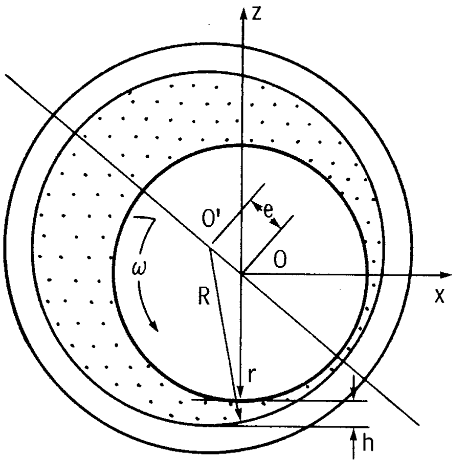


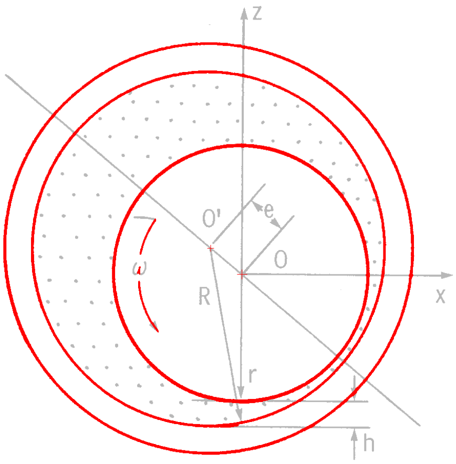


Results on GREC2013 (continued)

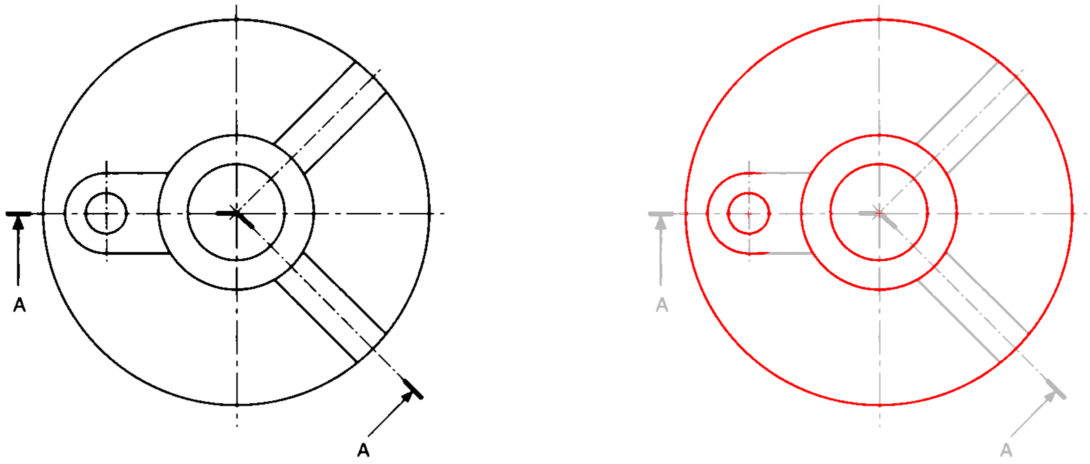


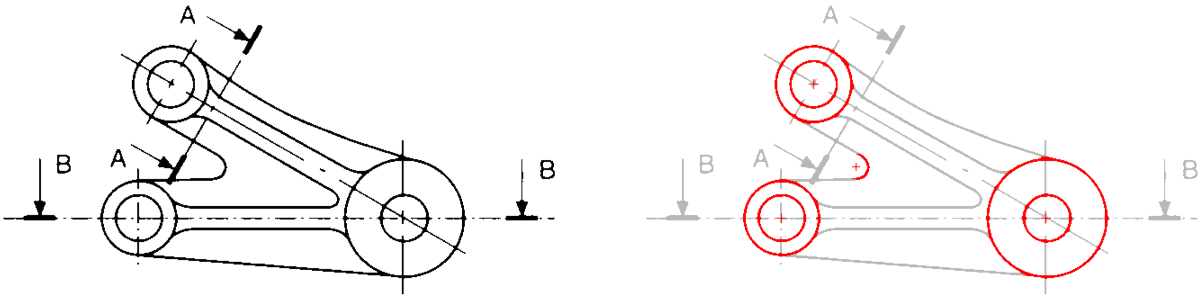


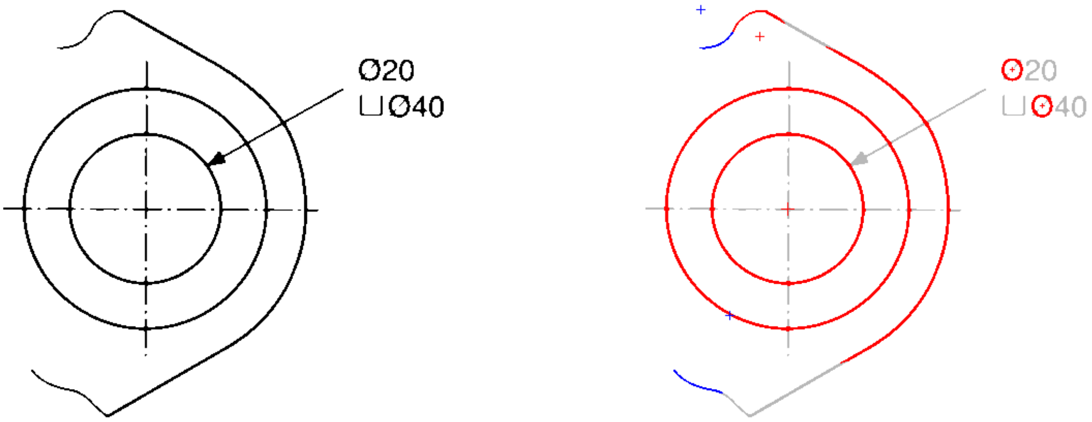




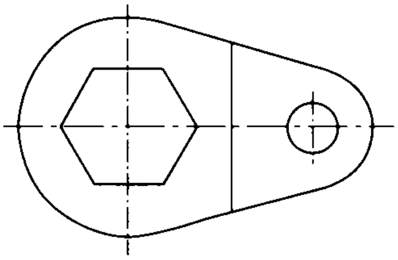
Results on SMP

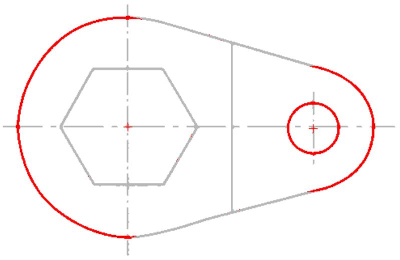


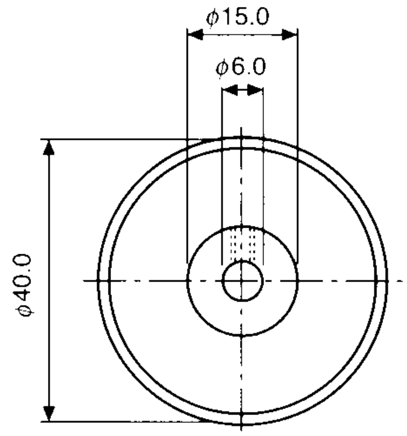


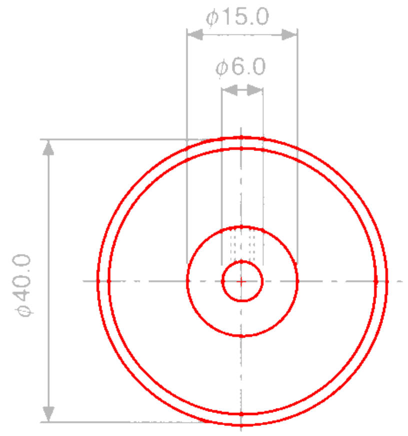


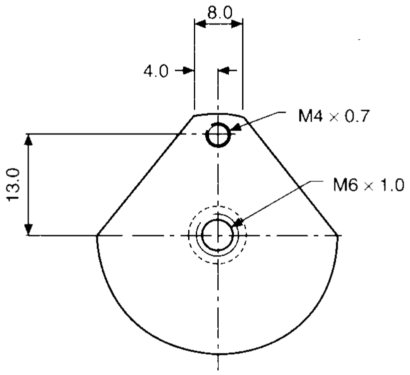
Results on SMP (continued)

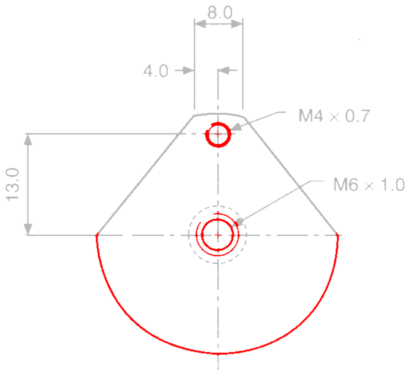


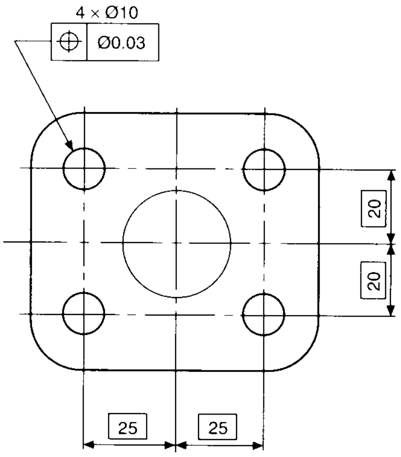


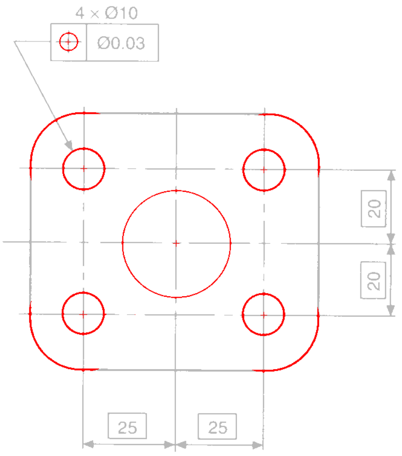












Appendix 2: Derivation of the ratio of disc-absentees to disc pixels

Details of the derivation to find the ratio of disc absentees to disc pixels (Chapter 3).

The equation of the real circle $\mathcal{C}^{\mathbb{R}}(0,-\frac{1}{2},r+1)$ is transformed to polar coordinates (ρ,θ) as follows.

$$\begin{aligned}
& \rho\sin\theta = \sqrt{(r+1)^2-\rho^2\cos^2\theta}-\frac{1}{2} \\
\text{or,}\quad & \rho^2+\rho\sin\theta+\frac{1}{4}-(r+1)^2=0 \\
\text{or,}\quad & \rho = \frac{-\sin\theta+\sqrt{\sin^2\theta-1+4(r+1)^2}}{2},\quad \text{as } \rho>0 \\
\text{or,}\quad & \rho^2 = \frac{4(r+1)^2-\cos 2\theta-2\sin\theta\sqrt{4(r+1)^2-\cos^2\theta}}{4}. \qquad (7.1)
\end{aligned}$$

Similarly, for the real circle $\mathcal{C}^{\mathbb{R}}(0,\frac{1}{2},r)$,

$$\rho^2 = \frac{4r^2-\cos 2\theta+2sin\theta\sqrt{4r^2-\cos^2\theta}}{4}. \qquad (7.2)$$

Hence, $\alpha_{(r,2)}$ is given by the area of $\mathcal{D}^{\mathbb{R}}(0,-\frac{1}{2},r+1)$ minus that of $\mathcal{D}^{\mathbb{R}}(0,\frac{1}{2},r)$ in Octant 1. Mathematically,

$$\begin{aligned}
\alpha_{(r,2)} &= \int_{\mathcal{C}^{\mathbb{R}}(0,-\frac{1}{2},r+1):\frac{\pi}{4}}^{\frac{\pi}{2}}\rho^2 d\theta-\int_{\mathcal{C}^{\mathbb{R}}(0,\frac{1}{2},r):\frac{\pi}{4}}^{\frac{\pi}{2}}\rho^2 d\theta \\
&= \int_{\frac{\pi}{4}}^{\frac{\pi}{2}}\frac{4(2r+1)-2\sin\theta\sqrt{4(r+1)^2-\cos^2\theta}-2\sin\theta\sqrt{4r^2-\cos^2\theta}}{8}d\theta \\
&= \frac{\pi(2r+1)}{8}-\frac{\frac{\sqrt{4(r+1)^2-\frac{1}{2}}}{\sqrt{2}}+4(r+1)^2\sin^{-1}\left(\frac{1}{2\sqrt{2}(r+1)}\right)+\frac{\sqrt{4r^2-\frac{1}{2}}}{\sqrt{2}}+4r^2\sin^{-1}\left(\frac{1}{2\sqrt{2}r}\right)}{8}
\end{aligned}$$

(7.3)

List of Publications of the Author

The thesis is based on original research work by Sahadev Bera. The relevant published/communicated papers are listed below[†].

Papers Published in Journals and as Book Chapters

1. BERA, S., BHATTACHARYA, B. B., GHOSHAL, S. AND BANDYOPADHYAY, N. R. (2012). Circularity analysis of nano-scale structures in porous silicon images. In *Computational Modelling of Object Presented in Images: Fundamentals, Methods and Applications*, edited by Paolo Di Giamberardino, Daniela Iacoviello, João Manuel R.S. Tavares, R.M. Natal Jorge, CRC Press, 403–408.
2. BERA, S., BHOWMICK, P., STELLDINGER, P. AND BHATTACHARYA, B. B. (2013). On covering a digital disc with concentric circles in $\mathbb{Z}^2$, *Theoretical Computer Science*, vol. 506, 1–16.
3. BERA, S., BISWAS, A. AND BHATTACHARYA, B. B. (2015). A Fast and Automated Granulometric Image Analysis Based on Digital Geometry, *Fundamenta Informaticae*, vol. 138, 1–18, to appear shortly.

Papers Published in Conference Proceedings

1. BERA, S., BHOWMICK, P. AND BHATTACHARYA, B. B. (2009). Detection of circular arcs in a digital image using Chord and Sagitta properties. In *Proc. 8th International Workshop Graphics Recognition (GREC 2009)*, vol. 6020 of *LNCS*, Springer-Verlag, Berlin, Heidelberg, 69–80.
2. BHOWMICK, P., BERA, S. AND BHATTACHARYA, B. B. (2009). Digital circularity and its applications. In *Proc. 13th International Workshop on Combinatorial Image Analysis (IWCIA 2009)*, vol. 5852 of *LNCS*, Springer-Verlag, Berlin, Heidelberg, 1–15; invited paper.

[†] **Reference Style:** AUTHORS (year). Title. *Journal/Proceedings name*, volume (number), pages.

3. Bera, S., Bhowmick, P. and Bhattacharya, B. B. (2014). A Digital-Geometric Algorithm for Generating a Complete Spherical Surface in $\mathbb{Z}^3$. In *Proc.* 1^{st} *International Conference on Applied Algorithms (ICAA 2014)*, *LNCS* vol. 8321, 49-61.
4. Bera, S., Biswas, A. and Bhattacharya, B. B. (2014). A Fast Digital-Geometric Approach for Granulometric Image Analysis. In *Proc.* 2^{nd} *International Conference on Recent Advances in Information Technology (RAIT 2014)*, Advances in Intelligent Systems and Computing, vol. 266, Springer, 37-47.

Papers submitted to Journals

1. Bera, S., Bhowmick, P. and Bhattacharya, B. B. (2015). On Covering a Solid Sphere with Concentric Spheres in $\mathbb{Z}^3$ (submitted); also in arXiv, `http://arxiv.org/abs/1411.1395`, 1-24.
2. Bera, S., Pal, S., Bhowmick, P. and Bhattacharya, B. B. (2015). On Chord and Sagitta in $\mathbb{Z}^2$: An Analysis towards Fast and Robust Circular Arc Detection (submitted); also in arXiv, `http://arxiv.org/abs/1411.1668`, 1-34.