In-plane object detection : detection algorithms and visibility problems

Citation for published version (APA):
Jovanovic, N. (2011). *In-plane object detection : detection algorithms and visibility problems*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mathematics and Computer Science]. Technische Universiteit Eindhoven. https://doi.org/10.6100/IR716187

DOI:
10.6100/IR716187

Document status and date:
Published: 01/01/2011

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 16. Nov. 2023

In-Plane Object Detection:
Detection Algorithms and Visibility Problems

Nataša Jovanović

The work described in this thesis has been carried out at the Philips Research Laboratories in Eindhoven, the Netherlands, as part of the Philips Research programme.

© Philips Electronics N.V. 2011
All rights are reserved. Reproduction in whole or in part is
prohibited without the written consent of the copyright owner.

A catalogue record is available from
the Eindhoven University of Technology Library
ISBN: 978-90-386-2542-3

Cover design by Davor Iliev

In-Plane Object Detection:
Detection Algorithms and Visibility Problems

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus, prof.dr.ir. C.J. van Duijn, voor een commissie aangewezen door het College voor Promoties in het openbaar te verdedigen op dinsdag 30 augustus 2011 om 16.00 uur

door

Nataša Jovanović

geboren te Lebane, Servië

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr. E.H.L. Aarts
en
prof.dr. J.J. Lukkien

Copromotor:
dr.ir. J.H.M. Korst

Preface

It was not always my ambition to pursue a PhD. It was during my Masters graduation internship at Philips Research that the idea crossed my mind, and thanks to some people, I was back in the Netherlands shortly after my graduation to work as a PhD student on one of my favorite topics. Four years later, I take the opportunity to write words of gratitude to everyone who helped me in many different ways along this exciting journey of getting a PhD.

First of all, I would like to thank Jan Korst, my daily supervisor at Philips Research and the only person who always supported me without questioning in all my decisions. Thank you for reading every single piece of my written work dozens of times, for correcting all my ”article mistakes” (I will learn eventually), for most amazing discussions and interesting meetings, for showing me the bright side in everything and most of all, thank you for your understanding. I cannot count the things I have learned from you. I was lucky to have you as my supervisor.

I would also like to thank my promoters Emile Aarts and Johan Lukkien for giving me the opportunity to do this PhD. I am grateful for your guidance, useful advice and for your valuable and always new insights. I have really enjoyed all our meetings and I am looking forward to many more in the future.

I wish to thank all the members of my reading committee, professors Jan van Leeuwen, Rolf Klein and Mark de Berg, for reviewing this thesis. I especially want to thank Mark de Berg for his elaborate comments that helped to substantially improve some parts of this thesis.

My PhD project was a collaboration project between Philips Research and the Eindhoven University of Technology, which means that I was lucky to work in both places and meet many people. First, I would like to thank all my colleagues from the Media Interaction group, the User Experiences group and the Human Interaction and Experiences group at Philips Research for making the working environment so pleasant. I am most grateful to the Entertaible project team for sharing their ideas and providing so many inspiring problems - this thesis would not have existed otherwise. I had the pleasure of working together with Ramon, Verus, Gijs, Greg, Marjolein, Kero, Robert, Gert-Jan, Dirk - thanks for your unselfish help in discussions, enjoyable coffee breaks, technical support and translating letters written in Dutch. Many thanks to my colleagues in the System Architecture and Networking group for having patience to listen to my geometry presentations and all the great vlaai after SAN meetings. I also want to thank my officemates Tanir, Mark, Waqar, Norbert, Hilal and Richard, and my fellow PhD students Mike, Melissa, Remi, Ugur and Martijn, simply for being such a great group of colleagues. Many thanks to the members of the Algorithms group at the TU/e for accepting me as an ”external” member of their group and all the great fun we had together visiting conferences.

There is a large group of people that I am proud to call my friends and I will here only mention their names. My friends here in the Netherlands: Caka, Vlado, Vesna, Goran, Sofija, Zvezdan, Sonja, Jasen, Meri, Dragan, Biba, Mira, Andjela, Miloš, Aleksandra, Nikola, Marija, Danijela and my friends and relatives in both of my home countries: Andriana, Deni, Ivana, Bobi, Goce, Irena, Ace, Sandra, Radmila, Faci, Kiki, Ana, Ljilja, Dani, Cece, Neša, Vanja and Toni - thank you! My sincere gratitude to Ilija, Nada, Milojka, Snežana, Angele, Pale and čika Djoka for believing in me since the day I was born.

For their endless love, support and sacrifice, I wish to thank my parents Anka and Radivoje and my sister Tamara. Thanks to your love, I never felt alone even though I was far from home. Thanks to you, I have become what I am now. I know that you are proud of me as much as I am proud of having you as my family.

Finally, I would like to thank Žarko, my dearest love and my best friend, for his love and patience, for comforting me when I was sad and for knowing how to bring a smile to my face. I know it wasn't easy to convince me to believe in myself. Thank you for showing me how great life can be.

Nataša Jovanović Eindhoven, July 2011

Contents

Preface iii

1 Introduction 1
1.1 In-plane object detection . . . 2
1.2 Research problems . . . 3
1.3 Related work . . . 9
1.4 Outline . . . 11

2 Detection Algorithms 15
2.1 Defining the detection problem . . . 16
2.2 Determining an n-wedge intersection in linear time . . . 21
2.3 Bounding the number of non-empty n-wedge intersections . . . 24
2.4 Cut-off algorithm . . . 25
2.5 Point-by-point algorithm . . . 28
2.6 Discussion . . . 32

3 Limitations of in-plane object detection 35
3.1 Introducing shadow regions . . . 38
3.2 Modelling Light Corridors . . . 44
3.3 Shadow regions algorithm . . . 45
3.4 Determining the number of shadow regions . . . 48
3.5 Discussion . . . 55

4 Visibility problems I:
Dark and shadow points among objects 59
4.1 Dark and shadow points and blocking sets of disks . . . 60
4.2 The hidden point problem . . . 63
4.3 The occluded point problem . . . 77

5 Visibility problems II:
Hiding objects 81
5.1 The occluded disk problem . . . 82

5.2 The hidden disk problem 93
5.3 The merging disks problem 104

6 Conclusion **111**

Bibliography **115**

Symbol Index **121**

Summary **124**

Curriculum Vitae **126**

1

Introduction

Detecting the presence of people or objects in a given environment recently became a necessity in many aspects of our modern way of living. Everyday, millions of people use devices such as touch screens, medical scanners, alarm and video surveillance systems, navigation systems, and various automated robots and vehicles in manufacturing processes. The functioning of all these devices relies on efficiently performing the task of *object detection*.

The main interest of any detection technology is to detect the presence of objects in a given environment, providing in addition, as much information about the objects as possible. The type of information that is useful, and certainly the precision and the speed with which such information is needed, strongly depends on the application using the detection technology. For some applications like medical imaging, the precision of the detection is of vital importance, for others like single-touch screens it is more important to enable high responsiveness.

Over the years, various object detection technologies have been developed, as a solution to the many different demands and purposes. Ultra-sound and magnetic resonance are heavily used for medical imaging, different types of cameras in computer vision, pressure sensors in traffic control systems, capacitive sensing for the purposes of highly responsive touch screens, etc. In this thesis, we focus on an innovative in-plane detection technique developed for table-top interaction [Hollemans et al., 2006b] that uses light rays to detect the presence of multiple objects simultaneously, to determine the objects' locations and to provide approximations in terms of size and shape. More precisely, we study the underlying geometric problems of in-plane object detection and propose algorithmic solutions, which positions the work presented in this thesis in the field of computational geometry.

Overview. The rest of this chapter is organized as follows. The in-plane detection technique and the device that uses it are described in Section 1.1. The research problems addressed in this thesis are presented in Section 1.2. In Section 1.3 we give an overview of the work related to the research presented in this thesis, before we provide the detailed outline of the thesis in Section 1.4.

1.1 In-plane object detection

Most of the recently developed interactive table-top devices, such as Microsoft Surface [Izadi et al., 2008; Wobbrock et al., 2009; Dietz and Eidelson, 2009], Diamond Touch [Dietz & Leigh, 2001], and Jeff Han's multi-touch panel [Han, 2005] detect objects using cameras and/or projectors positioned below or above the touch surface. The innovative technology of Entertaible [Hollemans et al., 2006a; Bergman and Hollemans, 2006; Hollemans et al., 2006b], the multi-touch screen developed by Philips Research, enables in-plane detection of multiple objects using infrared emitters (LEDs) and sensors positioned in a frame around the screen.

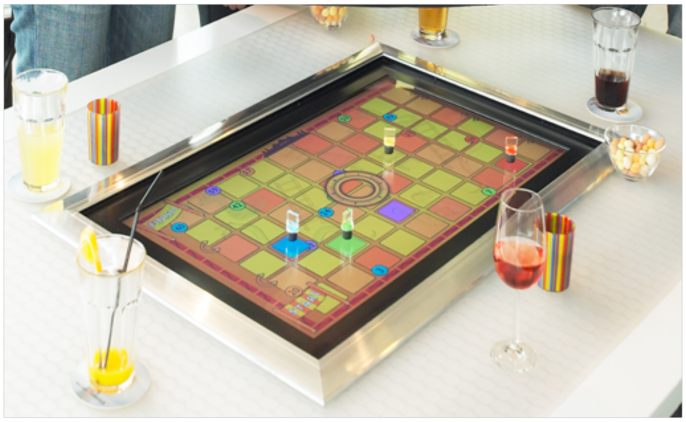

Figure 1.1. Entertaible.

A prototype of Entertaible was shown in the 2006 Consumer Electronics Show, where it was presented as a gaming platform that combines the interactivity of computer games with the social attractiveness of a board game [Frapolli, Hirsbrunner & Lalanne, 2007]. The device is equipped with 260 emitters and 260 sensors positioned in an alternating fashion around a 32-inch LCD screen; see Figure 1.1.

Given the physical limitations of the hardware, more precisely, the size of an LED and a sensor unit, the Entertaible can detect objects of diameter 1cm or larger. Depending on the type of LEDs and on the implemented detection algorithm, the entire process of detecting objects, also called a detection cycle, can be repeated 25 to 100 times per second, enabling in that way real-time tracking of moving objects on the table.

The technology uses light emitters and light sensors that are placed on some fixed positions around the space called the detection area, e.g., at the four edges of a rectangular table. In turn, emitters generate a short flash of infrared light and the objects placed in the detection area block that light for some of the sensors. In other words, some of the sensors are in the shadow of one or more objects; see Figure 1.3 for an illustration.

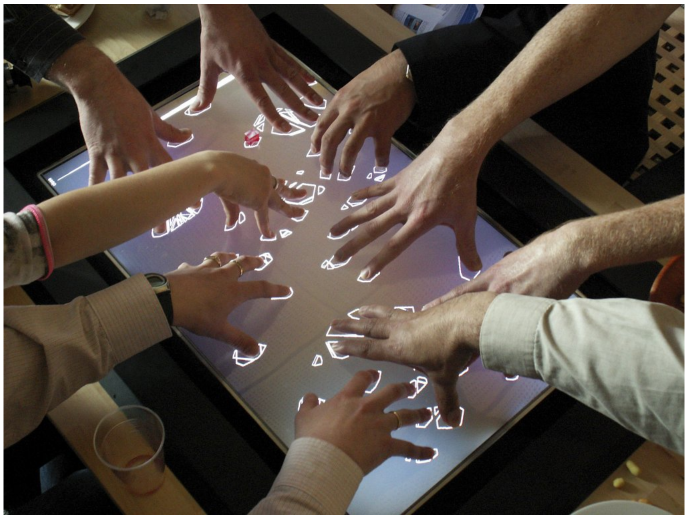

Figure 1.2. Entertaible detects 45 fingers.

This information can be used to determine the position and shape of objects [Jovanović, 2007], e.g., game pieces or fingers on the table; see Figure 1.2. If detection cycles are short enough, then moving objects can be tracked, for instance to recognize gestures made by fingers.

1.2 Research problems

The detection technique we described above gives rise to many interesting problems. For instance, there are problems of architectural nature that consider the network of hardware units that are used, and there are problems from the human-computer interaction domain that rise when the technology is implemented in some

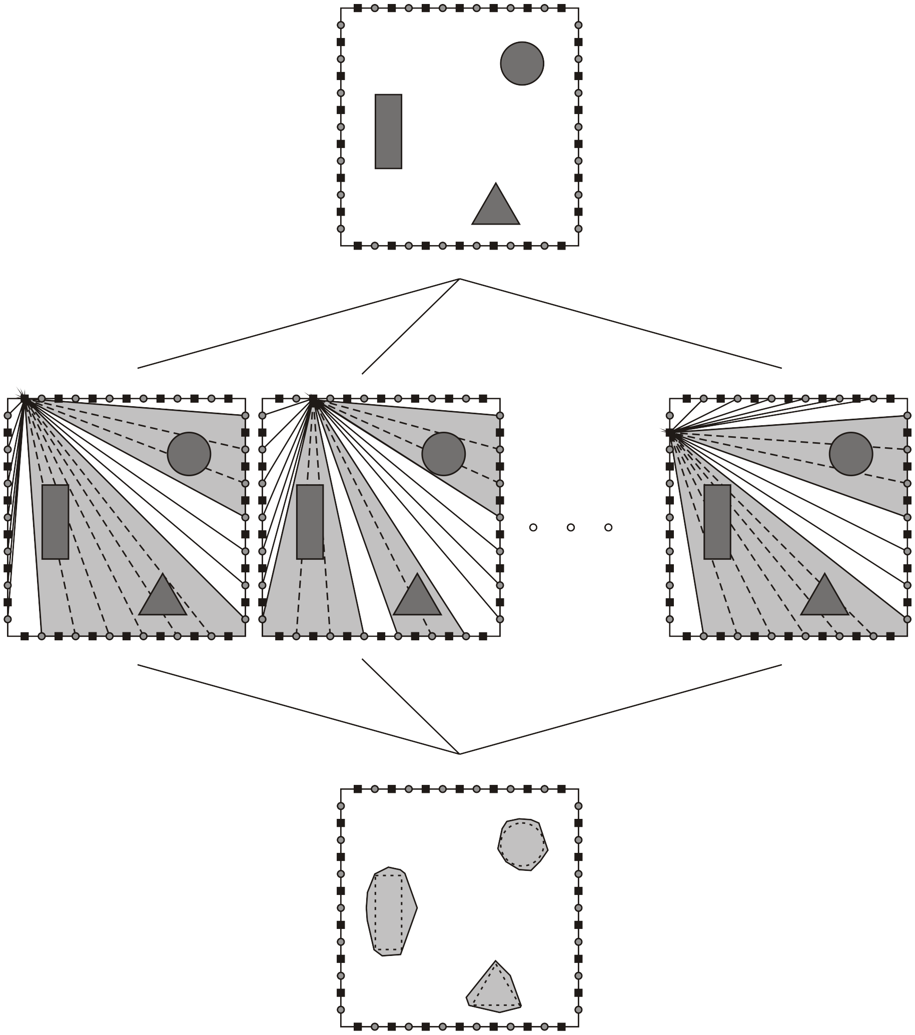

Figure 1.3. In-plane object detection: using the information on blocked light to provide an approximation of geometry of objects.

device. This thesis, however, considers exclusively the geometric problems that are at the core of the detection technique. These problems are in the realm of computer science, and more specifically, in the domain of computational geometry.

The detection problem

The essential problem in the process of object detection is to determine the location, size and shape of each of the objects, as accurately as possible. We refer to this problem as the *detection problem*. A solution to this problem is given by the design of an efficient algorithm that determines from the given sensor data a list of detected objects, with an approximation of their exact location and size.

For reason of simplicity, we will assume that emitters and sensors are points in the perimeter of a rectangular detection area. A light flash of an emitter e can be detected by a sensor s, provided that there is no object blocking the line of sight between e and s. After a short detection cycle, in which each emitter has flashed once, we know for each pair (e,s) whether or not the corresponding line of sight is blocked. This information is represented by a so-called *blocking matrix*, where each row corresponds to an emitter and each column corresponds to a sensor. The blocking matrix B is a binary matrix, where the value $b_{ij} = 1$ if the line of sight between emitter e_i and sensor s_j is not blocked by an object, and $b_{ij} = 0$ otherwise. Considering only the lines of sight of one emitter, which are either blocked or not, the detection area can be partitioned into a set of light and shadow areas. The shadow areas that are not separated by any non-blocked line of sight, are grouped into a *shadow wedge*. In this way, we can determine a set of shadow wedges corresponding to one emitter. Figure 1.3 shows the shadow wedges for three of the emitters. Note that each shadow wedge contains at least one object.

We consider an object to be *detectable* if it is in a shadow wedge for each emitter, more precisely, an object can be detected if it blocks at least one line of sight for each emitter. In that case, intersecting the shadow wedges of all emitters, where each of the wedges contains the given object, results in an approximation of the object's location, shape and size, given in the form of a convex polygon circumscribing the object. Detecting multiple objects placed in the detection area amounts to finding all wedge-intersections that contain the objects; see Figure 1.3. In other words, the detection problem corresponds to the so-called *wedge intersection problem*, considered in Chapter 2.

Limitations of in-plane object detection

In addition to the design of a detection algorithm, one can explore many different aspects related to the detection problem, such as determining the accuracy of the object detection. The accuracy of a single detected object can be defined as the ratio between the object's area and the area of its approximation provided by the detection method [Jovanović, 2007]. This accuracy depends on the number of emitters and sensors, and consequently, on the number of lines of sight defined by them. Clearly, the larger the number of emitters and sensors, the higher is the detection accuracy; see Figure 1.4. When detecting multiple objects, however, the accuracy level also depends on the objects' relative positions, hence, the combined occlusion that the objects define. In order to determine the maximum level of accuracy that can be achieved, we will assume an ”ideal” environment with infinitely many emitters and sensors in a frame surrounding the objects. In this way, we focus only on the intrinsic shortcoming of the detection technique implied by the objects’ positions. The problem of finding the best possible approximation assuming infinite number of emitters and sensors is called the *shadow regions problem*. This problem will be the subject of Chapter 3.

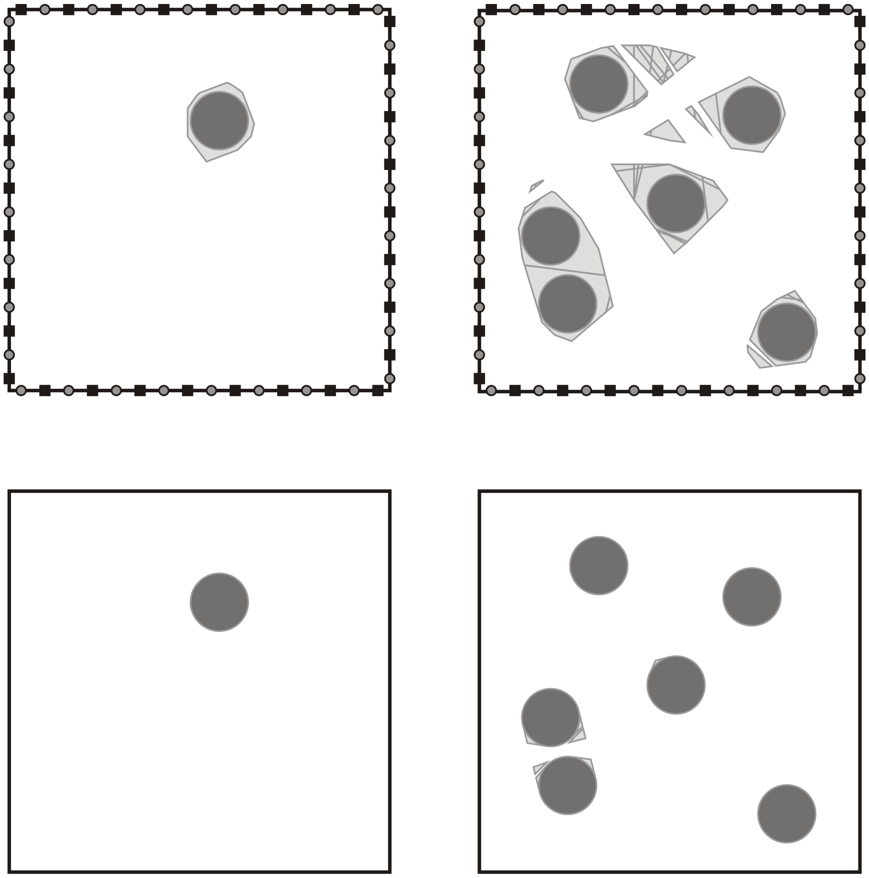

Figure 1.4. Detecting a single circular object (left) and six identical circular objects (right) with 32 emitters and 32 sensors (top) and with infinitely many emitters and sensors (bottom).

Visibility problems

Detecting objects using light rays is based on the fact that the objects partly block the emitted light. In this way, for each line of sight connecting one emitter and one sensor we are able to determine whether or not the line of sight is blocked, and use that information to detect the objects. However, this detection method at the same time, also provides grounds for many *visibility problems*. One example of such a visibility problem is the so-called *phantom object* problem [Jovanović, 2007], which occurs usually in the presence of many objects in the detection area. If all of the lines of sight intersecting some given area are blocked, it is assumed that the area contains an object that blocks all those lines. However, the same may occur in the presence of many objects, which ”accidentally” block the lines of sight crossing some area, and as a result, a non-existing object is reported as detected; see Figure 1.5. Such a detected non-existing object is called a phantom object. It can happen as well that we have a real object on the location of a phantom object, however, the detection technique is unable to distinguish between these two cases and hence, unable to provide the correct information. In the absence of the solution in the detection domain, it is of high importance to explore the occurrence of the phantom objects, so that they can be avoided in the application domain, for example, by designing a multi-touch screen game that does not allow the game pieces to be positioned in a manner that leads to a phantom object occurrence.

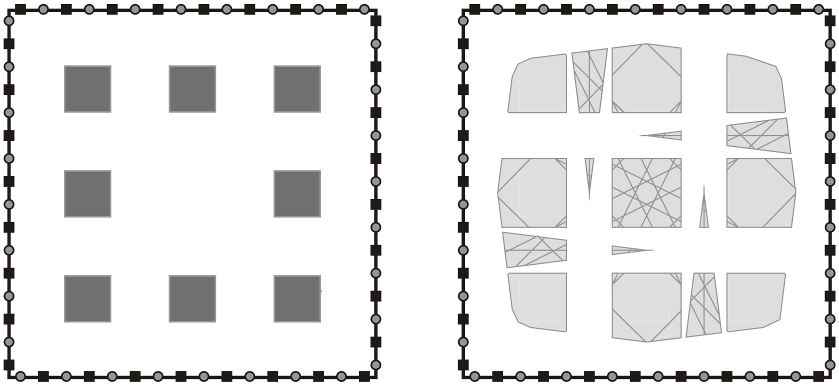

Figure 1.5. Eight square objects positioned inside the detection area create multiple phantom objects.

Another example of a visibility problem is the problem of two or more objects being detected as one object. In this case, the lines of sight that separate the objects are blocked by other objects. Unlike the phantom problem, this problem can usually be solved in the detection domain when additional information on objects' size is available, which is the case for many applications.

Based on the visibility problems identified in practice, we define and discuss five visibility problems in this thesis. The detection area is assumed to be the entire plane in which non-overlapping unit disks representing objects can be positioned. In addition, we assume the minimum mutual distance between any pair of disks to be at least some given distance d.

We consider the problems of determining the minimum number of non-overlapping unit disks needed to block a given set of rays or lines. The five visibility problems discussed in this thesis are:

- *The hidden point problem*, i.e., blocking the set of all rays emanating from a given point;
- *The occluded point problem*, i.e., blocking the set of all lines containing a given point;
- *The occluded disk problem*, i.e., blocking the set of all lines intersecting a given unit disk;
- *The hidden disk problem*, i.e., blocking the set of all rays emanating from a given unit disk;

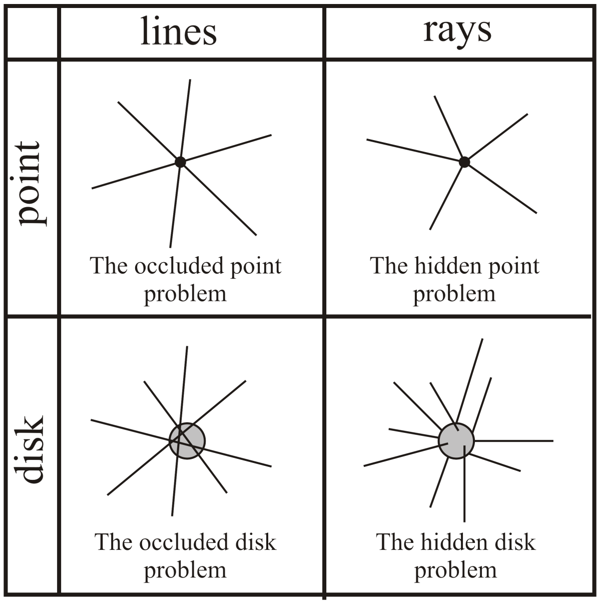

Figure 1.6. Four visibility problems defined for four sets of lines/rays.

- *The merging disks problem*, i.e., blocking the set of all lines passing between two given unit disks;

For each of the problems we focus on deriving asymptotic bounds on the minimum number of disks needed for blocking, when the minimum mutual distance d between a pair of disks goes to infinity.

Besides the above mentioned problems, there are also many geometric problems related to *shape recognition*, where the essence is in determining an object's real shape from the approximation provided by the detection method. Note that whatever the shape of an object is, its approximation will always be a convex polygon, as illustrated in Figure 1.7. In addition to the detection problems which are defined for a static environment, one can consider also *tracking problems*, i.e., detecting objects in a dynamic environment, which results in the recognition of movements. These problems, however, are not the subject of the research presented in this thesis.

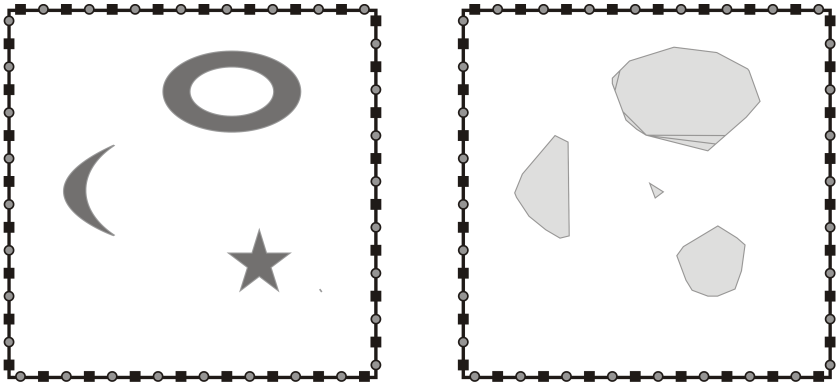

Figure 1.7. The detection method results in all objects being reported as convex polygons.

1.3 Related work

The in-plane object detection technique is related to some extent to the methods of image reconstruction known as tomography [Slaney & Kak, 1988; Jain, 1989], heavily used in medical imaging. Tomography is an imaging method based on sectioning or slicing, where projections of the same 2D or 3D image from different angles are used to reconstruct the original image. The projections (sections) of an image are obtained using any kind of penetrating wave, e.g., X-rays for CT scans or radio-frequency wave for MRI scans. The original image is reconstructed from a set of projections using a reconstructive algorithm, which is usually classified as either filtered back projection [Lauritsch & Haerer, 1998; Katsevich, 2004] or iterative reconstruction [Gordon & Herman, 1974; Gilbert, 1972; Chew & Wang, 1990; Mueller & Yagel, 2000]. These algorithms are based on the mathematics of the Radon Transform [Gindikin, 1994; Ramm & Katsevich, 1996; Deans, 1983], which allows to determine the density distribution of an unknown object from the data collected from cross-sectional scans. The collected data consists of detected signal strength in multiple end points, with known points of the signal origin. The tomographic reconstruction and our in-plane object detection are therefore, based on the same idea. However, instead of the complex integral transform used on a wide range of different signal strengths, we use only binary data gathered by the sensors (received or not received the light), which is then easily transformed into a geometric model of the objects.

As we briefly explained in the previous section, the detection method is based on determining all shadow wedge intersections. As illustrated in Figure 1.3, a shadow wedge has the shape of a convex polygon. Hence, a detection algorithm is based on the operation of determining the intersection of two given polygons. Finding the intersection of two polygons, which is essentially the problem of finding line segment intersections [De Berg et al., 2000], is a problem considered by many computer scientists since the foundation of the field of computational geometry. Some interesting results on this topic that are related to our detection method can be found in [Shamos & Hoey, 1976; Shamos, 1978; Bentley & Ottmann, 1979; Nievergelt & Preparata, 1982; Chazelle, 1986; Chazelle & Dobkin, 1987; Chazelle & Edelsbrunner, 1992; Balaban, 1995].

Agarwal & Sharir [2000] gave a survey on combinatorial and algorithmic properties of the arrangements of geometric objects, where an arrangement of a finite collection of geometric objects is defined as a decomposition of the space into connected cells induced by them. The problem of determining all shadow regions in the plane defined by a given set of unit disks, i.e., the shadow region problem considered in Chapter 3 of this thesis, is therefore related to the arrangements of disks and lines in the plane.

The geometric problems that are the subject of this thesis are closely related to the *illumination problems* [Soltan, 1979; Urrutia and Zaks, 1989; Czyzowicz et al., 1989; Schmitt, 1993; Szabó, 1998; Martini and Soltan, 1999], extensively studied in the field of combinatorial geometry. These problems deal with illumination of bodies by a finite or infinite number of point light sources, where a point x on the boundary of a given body is said to be illuminated by a point y outside that body if the line segment (x,y) does not intersect the body. The origins of the illumination problems lie in the problems of covering a convex body by identical or smaller homothetic copies of it [Levi, 1954; Boroczky & Soltan, 1996; Rogers & Zong, 1997]. The connections between the two classes of problems [Soltan, 1966] were established in [Boltjansky, 1960] and [Hadwiger, 1960] by introducing two different types of exterior illumination, namely, the illumination by a direction and the illumination by an exterior point. The notion of visibility between an exterior point and a boundary point of a convex body was introduced in [Valentine, 1970]. Among many interesting problems related to the illumination is the problem of finding the minimum number of exterior points that illuminate a given convex body. Considered in two-dimensional Euclidean space, a variant of this problem is known in computational geometry as the classical *art gallery problem*, i.e., the problem of determining the minimum number of guards sufficient to cover the interior of any n-wall art gallery. The guards are points that can survey 360° around their fixed positions and the art gallery is a polygon with n edges. The art gallery problem was originally posed by Klee in 1973. Since then, many variants of the problem [O'Rourke, 1987] in conjunction with the related illumination problems have been investigated [Urrutia, 2000] and there is a tremendous amount of results published on the topic.

The visibility problems considered in this thesis relate to determining the minimum number of disks needed to block a given set of lines or rays. Defined in that way, these problems are related to the visibility (illumination) problems of a family of non-overlapping convex bodies [Tóth, 1977]. A family of pairwise non-overlapping congruent balls in Euclidean space forms a cloud for a given ball, if each ray emanating from the center of the given ball intersects at least one of the balls of the family [Szabo & Ujvary-Menyhart, 2002]. Similarly, a family of non-overlapping congruent balls forms a dark cloud for a given ball, if each ray emanating from the given ball intersects at least one of the balls of the family [Soltan, 1995; Zong, 1997]. Although defined in a similar way, the problems of determining minimum (dark) clouds on one side and the problems of a hidden point or disk on the other, focus on different aspects. The constraint in both the hidden point and the hidden disk problem is that the distance between two disks is at least some given distance d. However, the problems of finding the minimum cloud and the minimum dark cloud pose no requirements on the minimum distance between the balls in the cloud and the solutions to both problems in the two-dimensional space are well-known [Tóth, 1959]. Therefore, these problems are only challenging in the three-dimensional space. More visibility problems related to the problems considered in Chapter 4 and 5 are the problem of intersecting convex sets by rays [Fulek, Holmsen & Pach, 2008], the floodlight illumination problem [Steiger and Streinu, 1998; Steiger and Streinu, 1994; O'Rourke et al., 1995; Estivill-Castro et al., 1995], many variants of the orchard visibility problem [Pólya, 1918; Allen, 1986; Kruskal, 2008; Dumitrescu and Jiang, 2010] and various visibility problems defined in [O'Rourke, 2004].

1.4 Outline

The rest of the thesis is organized as follows. In Chapter 2, which is based on [Jovanović, Korst & Pronk, 2009], we consider the following problem. Given a rectangle Γ with n emitters and n sensors on its perimeter, objects in Γ can be detected by determining which of the $n \times n$ line segments (e,s) between emitters and sensors are blocked by objects. The problem of object detection can be formulated as the problem of finding all non-empty n-wedge intersections, where a wedge is defined by a consecutive set of blocked line segments from the same emitter. We show that for a given set of n wedges, one emanating from each emitter, we can determine the intersection (i.e., the convex polygon) in $\mathcal{O}(n)$ time, assuming some given ordering of the wedges. We present two algorithms that determine all non-empty n-wedge intersections in $\mathcal{O}(n^5)$ and $\mathcal{O}(n^3)$ time, respectively, assuming that objects are sufficiently large.

In Chapter 3, we describe the limitations of the detection method by exploring the object detection in an "ideal" environment with infinitely many light emitters and light sensors. Formally, the problem of interest to us can be defined in the following way. Given a set of N non-overlapping unit disks in the plane, a line ℓ is called blocked if it intersects at least one of the disks and a point p is called a shadow point if all lines containing p are blocked. In addition, a maximal closed set of shadow points is called a shadow region. We derive properties of shadow regions, and present an $\mathcal{O}(N^4)$ algorithm that outputs all shadow regions. We prove that the number of shadow regions is $\Omega(N^4)$ for some instances, which implies that the worst-case time complexity of the presented algorithm is optimal.

We address two visibility problems related to in-plane object detection in Chapter 4. First, we consider the problem of determining the minimum number N_d of non-overlapping unit disks that is required to block all rays emanating from a point p, where each disk has at least a distance d to point p and to any other disk. We study the asymptotic behavior of N_d, as d tends to infinity. By deriving upper bounds and lower bounds, we prove that both upper and lower bounds on N_d are quadratic in d, where the upper bound is based on establishing an interesting link between unit disks positioned on a regular triangular grid and Farey sequences from number theory. By positioning point p as well as the centers of the disks on the grid points of such a triangular grid, we create hexagonal rings of disks around p. We prove that we need exactly $d-1$ of these hexagons to block all rays emanating from p. From these results, we straightforwardly derive quadratic upper and lower bounds on the minimum number of non-overlapping unit disks that is needed to block all lines containing p, where again the minimum mutual distance between any two disks as well as the distance between a disk to point p is at least d. The results presented in Chapter 4 are published in [Jovanović et al., 2009].

In Chapter 5, which is based on [Jovanović et al., 2008; Jovanović et al., 2010a; Jovanović et al., 2010b], we continue with a similar discussion where we explore

three visibility problems related to hiding objects. First, we consider the occluded disk problem that is formally defined as follows. A unit disk δ is occluded by a set of non-overlapping disks $\mathcal{D}$ if every line that intersects δ is blocked by at least one of the disks in $\mathcal{D}$. We focus on determining the minimum number of disks that occlude a given disk assuming that the minimum mutual distance between each pair of disks is d. We derive upper and lower bounds on this minimum number of disks for small values of distance d, more precisely, for $2 \leq d \leq 4$, before we present the asymptotic bounds that are quadratic in d as a result that directly follows from the results presented in Chapter 4.
Next, we focus on the hidden disk problem, where a unit disk δ is said to be hidden by a set of non-overlapping unit disks $\mathcal{D}$ if each ray emanating from δ is blocked by at least one of the disks in $\mathcal{D}$. We consider the problem of finding the minimum number of disks in $\mathcal{D}$ that hide a given disk, where the minimum distance d from a disk to any other disk is given. We study the asymptotic behavior of the minimum number of disks as the minimum mutual distance approaches infinity. Using a regular ordering of disks on concentric circular rings we derive an upper bound and prove that the minimum number of disks required for blocking is quadratic in the minimum distance between the disks.
The last problem considered in Chapter 5 is the merging disks problem. For a given minimum mutual distance d between the disks, we focus on determining the minimum number of non-overlapping unit disks that block the set of lines passing between two given unit disks. We first present the minimum blocking sets of disks for $2 \leq d \leq 3$, before we present upper and lower bounds for $d \to \infty$.

Finally, we give concluding remarks in Chapter 6 by summarizing the results of the previous chapters.

2

Detection Algorithms

To enable simultaneous detection of multiple objects on table-top interactive devices, a detection technology that uses light emitters and light sensors has been developed. The emitters and the sensors are placed on some fixed positions in a frame around the table. In turn, emitters generate a short flash of infrared light. After one emitter has flashed, all sensors report whether or not they detect the emitted light. A light flash of an emitter e can be detected by a sensor s, provided that there is no object blocking the line segment between e and s. After a short detection cycle, in which each emitter has flashed once, we know for each pair (e,s) whether or not the corresponding line segment is blocked. This information can be used to solve a detection problem, more precisely, to determine the position and shape of objects, e.g., game pieces or fingers on the table [Hollemans et al., 2006].

In order to track moving objects on the table, for instance to recognize gestures made by fingers, the detection problem should be solved repeatedly in a short time, after each detection cycle. Given the physical limitations of the emitters (LEDs) used, the frequency of the detection cycles can be at high as 100 Hz, which leaves only 10 ms of time to solve the detection problem. Therefore, it is of utmost importance to develop a fast detection algorithm that can support real-time applications like multi-player board games, map navigation or web browsing.

Overview. The detection algorithms that make use of the described technology are the main topic of this chapter. We introduce notation and formally define the detection problem in Section 2.1. In Section 2.2 we present an algorithm for the detection of a single object with a run time that is linear in the number of emitters. The number of objects that can be detected is discussed in Section 2.3. In Section 2.4 and 2.5 we present two detection algorithms, the cut-off algorithm and the point-by-point algorithm, that run in $\mathcal{O}(n^5)$ and $\mathcal{O}(n^3)$ time, respectively, where n is the number of emitters. In addition, we give pointers to alternative algorithms. Implementation issues and other details of practical relevance are discussed in Section 2.6.

2.1 Defining the detection problem

Let Γ be a rectangular detection area and let $E=(e_0,\dots,e_{n-1})$ and $S=(s_0,\dots,s_{n-1})$ be two sets of n points on the perimeter of Γ, both ordered in clockwise fashion, where E is the set of emitters and S is the set of sensors. The points from E and S are positioned alternately on the border of Γ and the distance between each pair of neighboring points is r. In addition, for each emitter e_i on one side of Γ there is a sensor s_j on the opposite side of Γ, such that the line segment (e_i,s_j) is perpendicular to the edges of Γ containing e_i and s_j. A line segment connecting one emitter and one sensor is called a *line of sight*.

A closed shape placed in the interior of the detection area Γ is called an object. Objects do not intersect the boundaries of Γ and they do not overlap each other.

Let the *blocking matrix* B be an $n \times n$ binary matrix where $b_{i,j} = 1$ if the line of sight between emitter e_i and sensor s_j is not blocked by any object, and $b_{i,j} = 0$ otherwise. By definition, we have $b_{i,j} = 1$ when e_i and s_j are on the same side of Γ. For a given emitter e_i, a maximal sequence of consecutive sensors for which the light of e_i is blocked defines a shadow wedge, or wedge for short. Let $s_{j+1},\dots,s_{k-1}$ be such a sequence, where indices are taken modulo n. Hence, potentially a wedge is defined by a sequence $s_{j+1},\dots,s_{n-1},s_0,\dots,s_{k-1}$, where $k < j$. For an illustration see Figure 2.1.

The shadow wedge can be formally defined as the intersection of two half-planes as follows.

Definition 2.1 (shadow wedge). Let e_i be an emitter and $s_j,\dots,s_k$ be sensors such that $s_{j+1},\dots,s_{k-1}$ are blocked from e_i, and s_j and s_k are not blocked from e_i, and $k \neq j, j+1$. Let h_j^+ denote the half-plane bounded by the line through e_i, s_j and containing s_{j+1}, and let h_k^- denote the half-plane bounded by the line through e_i, s_k

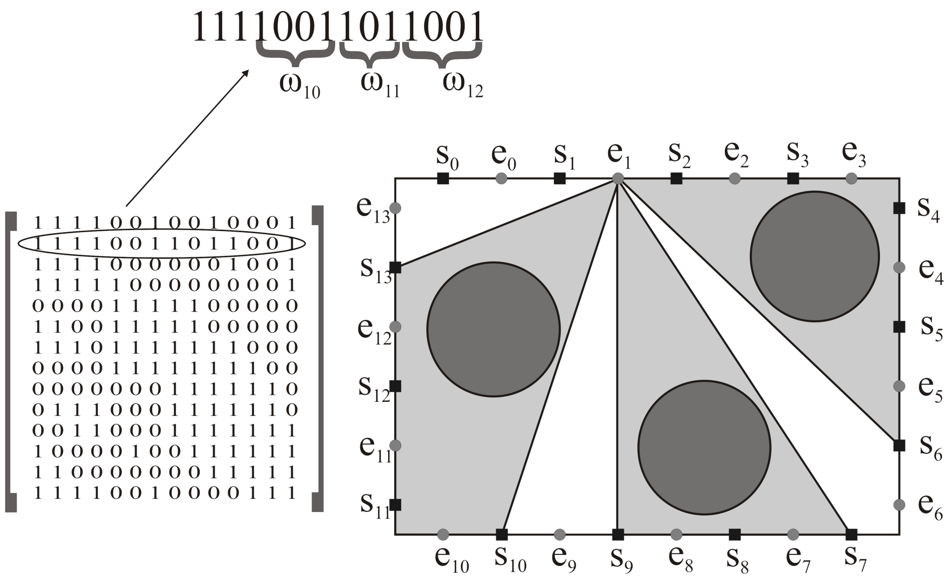

Figure 2.1. An example of the blocking matrix and the corresponding shadow wedges for emitter e_1.

and containing s_{j+1}. Then we say that $\omega = h_j^+ \cap h_k^-$ is the shadow wedge defined by a 3-tuple (e_i, s_j, s_k). □

Note that for a wedge (e_i, s_j, s_k), we have that $b_{i,j} = b_{i,k} = 1$ and $b_{i,l} = 0$, for $l = j+1, \ldots, k-1$, where all indices are taken modulo n. Hence, the light of e_i is not blocked for left and right bounding sensors s_j and s_k, but it is for all sensors in between s_j and s_k.

Since objects are completely inside the detection area Γ, a wedge can be bounded by intersecting it with rectangle Γ, resulting in a polygon of 3, 4 or 5 edges, depending on whether s_j and s_k are on the same, adjacent, or non-adjacent sides of Γ.

In the case of detection of only one object placed in Γ, there is at most one shadow wedge per emitter defined by the object. When multiple objects are placed in Γ, there may be multiple wedges emanating from one emitter. However, the number of wedges per emitter does not necessarily correspond to the number of objects. The number of wedges per emitter is at most the number of objects and often, multiple objects are contained in a single wedge. For example, emitter e_3 in Figure 2.1 has only one wedge. All wedges emanating from emitter e_i are given by the sequence $(\omega_{i0}, \omega_{i1}, \ldots, \omega_{im_i-1})$, $m_i < n/2$, ordered in clockwise fashion, when viewed from emitter e_i; see Figure 2.1. Hence, m_i gives the number of wedges emanating from e_i.

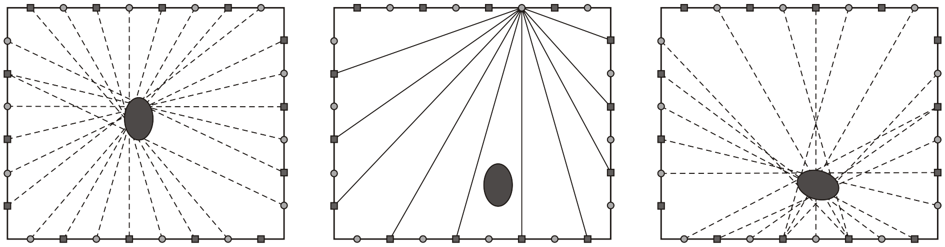

Figure 2.2. Detectibility of an object depends on its position and orientation: left and right - the object is detectable since it blocks at least one line of sight for each emitter; in the middle - the object is not detectable.

An object placed in Γ that is given by its position and orientation is *detectable* if it intersects at least one line segment (e_i, s_j) for each $e_i \in E$. An object that is detectable on one position in Γ may not be detectable on some other position in Γ, as illustrated in Figure 2.2. Also, a different orientation of an object can result in the object being detectable in one case, but not detectable in another; see Figure 2.2.

Definition 2.2 (surely detectable object). An object o placed in Γ is surely detectable if o intersects at least one line segment (e_i, s_j) for each $e_i \in E$, irrespective of its position and orientation. □

By definition, any detectable object (See Figure 2.2) is in a wedge for each emitter. As a consequence, we can detect such an object by determining the intersection of all n wedges in which it is positioned. The resulting convex polygon circumscribes the object(s) located in this intersection (See Figure 2.3).

Lemma 2.1. *Let r be the distance between a pair of neighboring emitters and sensors, then a circular object of radius at least r is surely detectable.*

Proof. A circle of radius at least r is too large to be inscribed in a triangle that has at least one edge of size less or equal to $2r$. Given that the distance between two neighboring sensors s and s' is $2r$ when s and s' are on the same side of Γ, the circle of radius at least r cannot be inscribed in a triangle with vertices in e, s and s', i.e., the circle intersects either (e, s) or (e, s'). Quadrilaterals in the corners of Γ determined by two neighboring sensors on different sides of Γ, one vertex of Γ and one emitter are also not large enough to contain the circle of radius at least r, by which we proved the lemma. □

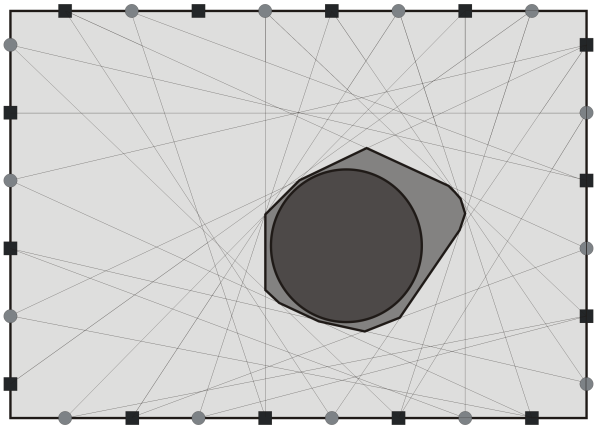

Figure 2.3. The polygon circumscribing the circular object is the result of intersection of n wedges containing the circular object.

For reasons of convenience, and as a consequence of Lemma 2.1, we only consider detecting objects that are each large enough to contain a circle of radius r.

Definition 2.3 (wedge vector). A wedge vector $(z_0, z_1, \ldots, z_{n-1})$, where z_i is an integer such that $0 \leq z_i < m_i$, specifies a combination of n wedges, one emanating from each emitter, given by $\omega_{0z_0}, \omega_{1z_1}, \ldots, \omega_{n-1z_{n-1}}$. □

To detect an object, we simply have to determine the convex polygon that is defined by the intersection of wedges, one from each emitter. This gives rise to the following definition.

Definition 2.4 (*n*-wedge intersection). An n-wedge intersection $\mathcal{C}$ specified by a wedge vector $(z_0, z_1, \ldots, z_{n-1})$, with $0 \leq z_i < m_i$, is given by

$$\mathcal{C} = \bigcap_{i=0}^{n-1} \omega_{iz_i}$$

□

Note that two n-wedge intersections specified by different wedge vectors do not overlap, since no two wedges of the same emitter overlap. Although they do not overlap, two n-wedge intersections can still have a non-empty intersection in the sense that they can have one edge in common. This situation can occur when the right bounding sensor of one wedge is identical to the left bounding sensor of a neighboring wedge of the same emitter.

After we defined an n-wedge intersection, we are ready to formulate the problem of interest to us, as follows.

Wedge Intersection Problem. *Given the n $\times$ n blocking matrix B, determine all non-empty n-wedge intersections.*

More specifically, we are interested in developing efficient algorithms to determine all non-empty n-wedge intersections for a given blocking matrix B. In other words, the wedge intersection problem represents a detection problem as defined in Section 1.2.

An n-wedge intersection usually corresponds to one object, as shown by experimental results in [Jovanović, 2007]. Occasionally, multiple objects can be detected as one, especially if their mutual distance is too small or if the view on the area between them is occluded by other objects; see Figure 2.4 - left. Furthermore, incidentally a small area that is in the ”shadow” of an object for each emitter results in a non-empty n-wedge intersection that can be identified as an object; see Figure 2.4 - right. In other words, an n-wedge intersection can circumscribe 0, 1 or more objects, which will be discussed in more detail in the following chapters.

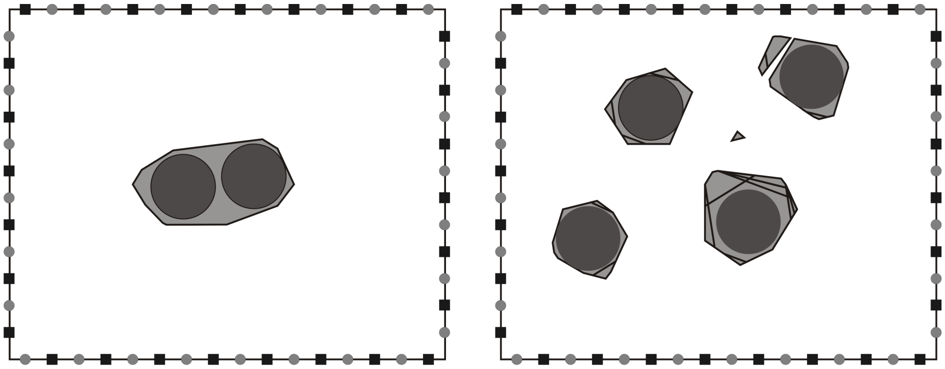

Figure 2.4. Left: an n-wedge intersection that contains two objects; right: four circular objects create small n-wedge intersections that do not contain any object.

As we described above, detecting objects using light emitters and light sensors requires determining all non-empty n-wedge intersections. Therefore, determining an intersection of n wedges efficiently is of utmost importance for real-time applications. In the following section, we show that a single n-wedge intersection can be determined in a time that is linear in the number of emitters. In Section 2.3 we discuss the number of non-empty n-wedge intersections, given that we focus only on detecting objects large enough to contain a circle of radius r.

2.2 Determining an n-wedge intersection in linear time

The $\mathcal{O}(n \log n)$ time algorithm to determine the intersection of n half-planes [Shamos & Hoey, 1976] can be used straightforwardly to determine the intersection of n wedges in $\mathcal{O}(n \log n)$ time. Furthermore, the problem of computing the intersection of a given set of half-planes is dual to the problem of computing the convex hull of a set of points [Edelsbrunner, 1987; De Berg et al., 2000]. It is known that the convex hull of a set of points can be computed in linear time if the points are already sorted by x-coordinate [De Berg et al., 2000]. Hence, the intersection of n half-planes can be determined in $\mathcal{O}(n)$ time if the half-planes are ordered by slope. In this section, however, we use a different approach to show that an n-wedge intersection can be determined in linear time. This approach will make it easier to understand Section 2.4. The algorithm we present here is using the ordering of the wedges that is implied by the clockwise order of the emitters.

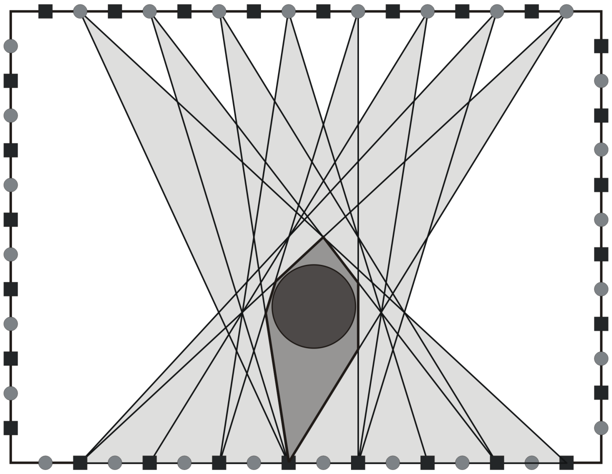

Figure 2.5. Intersecting an ordered set of wedges: the resulting polygon circumscribing the circular object after intersecting the first 8 wedges.

The algorithm starts with an initial polygon $P_0 = \Gamma$. We next iteratively intersect a polygon with a wedge. In iteration i we determine the intersection between polygon P_i and wedge ω_{iz_i}, resulting in polygon P_{i+1}, see Figure 2.5. Hence, after the n-th iteration, the resulting P_n will be the intersection of all n wedges; see the pseudocode below.

The vertices of a polygon are given by a cyclic sequence of points, ordered in clockwise direction. The counterclockwise neighbor (predecessor) of a vertex υ is denoted by pred(υ) and the clockwise neighbor (successor) of υ is denoted by succ(υ). The basic idea is that through all of the iterations we keep track of the

Algorithm 1 DetermineIntersectionOfWedges($(z_0, z_1, \dots, z_{n-1})$)

```
$P_0 \leftarrow \Gamma$;
for $i = 0$ to $n-1$ do
    update the leftmost vertex $\upsilon_{\text{leftmost}}$ and the rightmost vertex $\upsilon_{\text{rightmost}}$, viewed from emitter $e_i$;
    if wedge $\omega_{iz_i} = (e_i, s_j, s_k)$ and polygon $P_i$ do not intersect then
        return $\mathcal{C} = \emptyset$;
    else
        $P_{i+1} \leftarrow P_i$;
        if $\upsilon_{\text{leftmost}}$ is to the left of line $(e_i, s_j)$ then
            $P_{i+1} \leftarrow \text{LEFTCUT}(P_i, e_i, s_j)$;
        end if
        if $\upsilon_{\text{rightmost}}$ is to the right of line $(e_i, s_k)$ then
            $P_{i+1} \leftarrow \text{RIGHTCUT}(P_{i+1}, e_i, s_k)$;
        end if
    end if
end for
$\mathcal{C} \leftarrow P_n$;
```

leftmost and rightmost vertices of P_i, when viewed from the current wedge's emitter. Note that we consider the wedges in order of their emitters, which are ordered in clockwise fashion. In iteration i we intersect P_i with $\omega_{iz_i} = (e_i, s_j, s_k)$ for some j and k. At the start of iteration i the left and rightmost pointers are based on e_{i-1}'s viewpoint, from the previous iteration. We first update the pointers to the leftmost and rightmost vertices. Each of these pointers is updated by considering their clockwise neighbors (possibly repeatedly), as new candidates. Next, we first check whether or not the intersection is empty. It is empty if and only if the leftmost point is to the right of (e_i, s_k) or the rightmost point is to the left of (e_i, s_j). In that case we can stop. Otherwise, we carry out the intersection. This potentially results in cutting part of the polygon from its left side and/or from its right side, again when viewed from the wedge's emitter.

If there is a left cut (see Figure 2.6), then the leftmost vertex of P_i is to the left of line (e_i, s_j), and we start hopping to neighboring vertices until we cross this line. We do this both in clockwise and counterclockwise directions, in both cases starting at the leftmost vertex; see the routine LEFTCUT given in pseudo-code. Note that the total number of hops is at least two, and that each additional hop removes an additional vertex.

If the rightmost vertex of P_i is to the right of line (e_i, s_k), then a right cut is per-

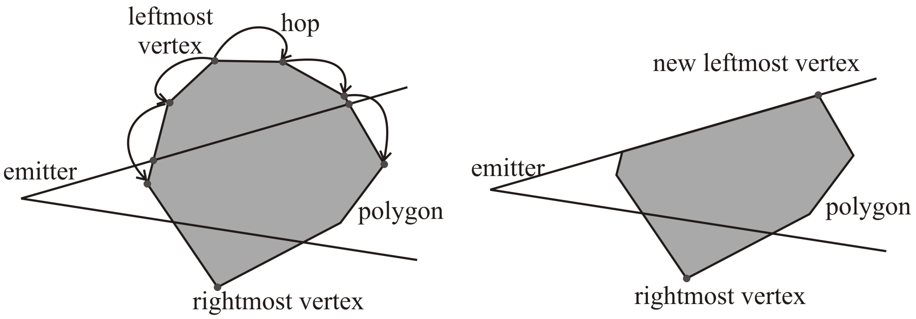

Figure 2.6. Intersecting a wedge and a polygon: the left cut.

formed in the same way as the left cut. Next, we show that the running time of Algorithm 1 is linear in the number of emitters.

Algorithm 2 LEFTCUT(P_i, e_i, s_j)

```
ccwHopVertex ← pred($\upsilon_{\text{leftmost}}$);
while ccwHopVertex is on the left of line $(e_i, s_j)$ do
    if succ(ccwHopVertex) is not $\upsilon_{\text{leftmost}}$ then
        Delete succ(ccwHopVertex) from the list of vertices of $P_i$;
    end if
    ccwHopVertex ← pred(ccwHopVertex);
end while
Insert the intersection point of line $(e_i, s_j)$ and edge
(ccwHopVertex, succ(ccwHopVertex)) as a new vertex of $P_i$,
after ccwHopVertex by replacing the previous succ(ccwHopVertex);
cwHopVertex ← succ($\upsilon_{\text{leftmost}}$);
while cwHopVertex is on the left of line $(e_i, s_j)$ do
    Delete pred(cwHopVertex) from the list of vertices of $P_i$;
    cwHopVertex ← pred(cwHopVertex);
end while
Insert the intersection point of line $(e_i, s_j)$ and edge
(cwHopVertex, pred(cwHopVertex)) as a new vertex of $P_i$,
before cwHopVertex by replacing the previous pred(cwHopVertex).
The new vertex is a potential leftmost vertex for iteration $i+1$;
```

Lemma 2.2. *Given a wedge vector* $(z_0, z_1, \ldots, z_{n-1})$*, the n-wedge intersection* $\mathcal{C} = \bigcap_{i=0}^{n-1} \omega_{iz_i}$ *can be determined in* $\mathcal{O}(n)$ *time.*

Proof. We consider the time complexity of Algorithm 1, by amortizing over all its iterations. Each cut removes *at least* one vertex and introduces two new ones.

Therefore, the net increase of the number of vertices per cut is *at most* one and the net increase of a complete intersection is *at most* two.

In each cut we determine at most two new vertices, each of them as an intersection of two lines, i.e., one bounding line of a wedge and one polygon edge. The intersection of two lines is a constant-time operation and at most 4 of these operations are performed in one wedge-polygon intersection. A hop is also a constant time operation containing the check whether a point, i.e., a vertex of the polygon is to the left or right of a line. The number of hops per intersection with both left and right cut is at least 4. Each additional hop that is performed removes an additional vertex. Summed over all n intersections, the total number of additional hops is bounded by $2n$. Given the clockwise ordering of the wedges, updating the leftmost and rightmost vertices, amortized over all n iterations, is also bounded by $\mathcal{O}(n)$ steps.

Hence, the intersection of an ordered set of n wedges can be determined in $\mathcal{O}(n)$ time. □

2.3 Bounding the number of non-empty n-wedge intersections

Given the result of Lemma 2.2, one approach to the wedge intersection problem would be to simply determine the wedges for all of the emitters from the input matrix V and then to determine for each possible wedge vector the corresponding n-wedge intersection. However, this would result in a non-polynomial algorithm. More precisely, there are arbitrarily large instances with $\Theta(n)$ wedges per emitter. The n emitters with $\Theta(n)$ wedges each, define $\Theta(n^n)$ wedge vectors that may result in a non-empty intersection, thus, $\Theta(n^n)$ potential n-wedge intersections.

A trivial upper bound on the number of non-empty n-wedge intersections can be determined using the solution of the well-known plane partitioning problem; see [Graham, Knuth & Patashnik, 1994]. The authors prove by mathematical induction that k lines in the plane can partition the plane into at most $k(k+1)/2+1$ regions. The number of line segments defined by n emitters and n sensors is quadratic in n. Hence, the number of convex regions that they define is $\mathcal{O}(n^4)$, i.e., there are $\mathcal{O}(n^4)$ non-empty n-wedge intersections.

This upper bound can be directly used to develop an $\mathcal{O}(n^4)$ algorithm for determining all n-wedge intersections. More precisely, the arrangement of $\mathcal{O}(n^2)$ lines that bound all shadow wedges can be constructed in $\mathcal{O}(n^4)$ time [Agarwal and Sharir, 2000; Edelsbrunner, 1987; De Berg et al., 2000]. If the depth of a cell is defined as the number of wedges containing the cell, then the set of n-wedge intersections is exactly the set of cells of depth n in the arrangement. Hence, to determine all

n-wedge intersections, we compute the depth of each cell, which can also be done in $\mathcal{O}(n^4)$ time by traversing the dual graph of the arrangement of lines [Edelsbrunner, 1987; Guibas and Sharir, 1993; De Berg et al., 2000], while maintaining the depth. Computing the size of an n-wedge intersection can be done in time linear in the number of edges [O'Rourke, 1998; Sack and Urrutia, 2000], hence, removing n-wedge intersections that are not sufficiently large also takes $\mathcal{O}(n^4)$ time.

We can further improve the upper bound on the number of non-empty n-wedge intersections since we only consider detecting objects that are large enough to contain a circle of radius at least r, where $2r$ is the distance between each pair of neighboring sensors on the same side of Γ. Since an object has an area of at least πr^2 and the total area of Γ is $\mathcal{O}(n^2 r^2)$, we can bound the number of objects that can be placed in Γ by $\mathcal{O}(n^2)$. Similarly, the number of sufficiently large n-wedge intersections is bounded by $\mathcal{O}(n^2)$. This upper bound on the number of sufficiently large n-wedge intersections plays an important role in the assessment of the computational complexity of the detection algorithms presented in Sections 2.4 and 2.5.

2.4 Cut-off algorithm

In this section, we present the cut-off algorithm, a detection algorithm that was implemented on the Entertaible. The cut-off algorithm is an incremental algorithm, and it represents a generalization of the algorithm presented in Section 2.2 when multiple objects are placed in the detection area. In other words, this algorithm builds upon a simple and intuitive idea of getting better approximations on geometry of multiple objects in each iteration, by cutting off parts of the shadow wedges that do not contain any objects.

The cut-off algorithm for determining all non-empty n-wedge intersections consists of repeatedly pair-wise intersecting the wedges of the current emitter e_i, with the polygons $\mathcal{P}_i$ that are the result of intersecting all wedges from the emitters $e_0, \ldots, e_{i-1}$; see pseudo-code of Algorithm 3. The algorithm starts with intersecting the wedges of the emitter e_0 with the rectangle Γ, yielding a first set of polygons $\mathcal{P}_1$ and ends with the intersection of the wedges of e_{n-1} with the set of polygons $\mathcal{P}_{n-1}$. The resulting set $\mathcal{P}_n$ of all n-wedge intersections represents the detected objects. Figure 2.7 illustrates the working of the algorithm.

Now let us take a closer look at the time complexity of the cut-off algorithm.

The list of wedges of one emitter can be determined in $\mathcal{O}(n)$ time, since we have to check the values for each of the n sensors. The number of wedges per emitter is

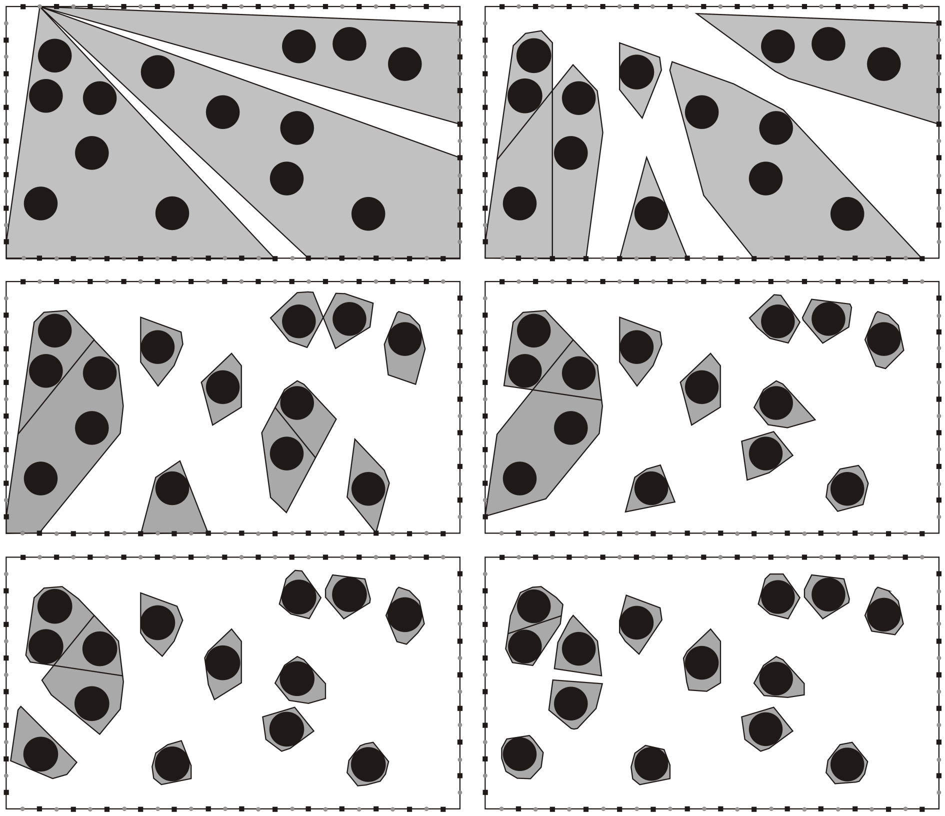

Figure 2.7. The cut-off algorithm in action: the set of polygons after 1, 6, 13, 20, 30 and all 40 iterations.

at most $\lfloor n/2 \rfloor$, by the definition of a wedge.

If a wedge ω_{ij} and a polygon P_k intersect and P_k is not entirely contained in ω_{ij}, then we "cut off" at most two parts of P_k. Each cut removes at least one vertex and introduces at most two new ones. Hence, the net increase per cut is at most one and the net increase of a complete intersection is at most two vertices. This gives an increase of at most $2n$ vertices after n intersections. Hence, all polygons P_k have $\mathcal{O}(n)$ vertices. In [De Berg et al., 2000; O'Rourke, 1998; Shamos, 1978] it is shown that the intersection between two convex polygons with n_1 and n_2 vertices can be determined in $\mathcal{O}(n_1 + n_2)$ time. Therefore, the intersection between a wedge and a polygon can be determined in $\mathcal{O}(n)$ time; see also Section 2.2.

As we explained in Section 2.3, the number of objects that can be placed in Γ is $\mathcal{O}(n^2)$. Hence, the list $\mathcal{P}_k$ of intermediate polygons is of size $\mathcal{O}(n^2)$ in each

Algorithm 3 CUTOFFALGORITHM(B)

```
$\mathcal{P} \leftarrow \{\Gamma\}$
for $i = 0$ to $n - 1$ do
  determine the list $\mathcal{W}_i$ of wedges of emitter $e_i$ in clockwise order
  for $j \leftarrow 0$ to $size(\mathcal{W}_i) - 1$ do
    $T \leftarrow \emptyset$
    for $k \leftarrow 0$ to $size(\mathcal{P}) - 1$ do
      if wedge $w_{ij}$ and polygon $P_k$ intersect then
        determine their intersection
        if the intersection is not too small then
          add the intersection to list $T$
        end if
      end if
    end for
    $\mathcal{P} \leftarrow T$
  end for
end for
```

iteration, provided that we remove the small ones at the end of each iteration.

In the second for-loop of the cut-off algorithm, we determine the intersection for each pair of $\mathcal{O}(n)$ wedges with $\mathcal{O}(n^2)$ polygons. Each of all $\mathcal{O}(n^3)$ combinations that may result in a non-empty intersection can be determined in $\mathcal{O}(n)$ time as we showed above. This results in $\mathcal{O}(n^4)$ time to determine the new set of (intermediate) polygons. It can be shown that a wedge-polygon combination that has an empty intersection can be verified as such in constant time. Hence, $\mathcal{O}(n^3)$ time is needed to discard all these combinations.

At the end of each iteration we determine the area of all non-empty $\mathcal{O}(n^3)$ polygons, to delete the ones that have an area smaller than πr^2. Each of the polygons has $\mathcal{O}(n)$ vertices. The area of such a polygon can be determined in $\mathcal{O}(n)$ time [O'Rourke, 1998], resulting in $\mathcal{O}(n^4)$ time for discarding the small polygons.

Finally, this has to be repeated for all n emitters, with which we proved the following theorem.

Theorem 2.1. *The overall time complexity of the cut-off algorithm is* $\mathcal{O}(n^5)$. □

Note that the size of the given blocking matrix is n^2, so the worst-case time complexity of the cut-off algorithm is $\mathcal{O}(\xi^2\sqrt{\xi})$, where ξ is the input size of the problem instance.

The worst case time complexity of the cut-off algorithm is significantly worse than that of the algorithms we present in the next section. However, it also has a big advantage over those algorithms in the sense that even if the detection cycle is cut short, it can still provide some approximation on the geometry of objects with guarantees that none of the objects is missed; see Figure 2.7.

2.5 Point-by-point algorithm

The problem of finding all large enough n-wedge intersections is related to the problem of computing many faces in an arrangement of lines, a problem that has been extensively studied in the field of computational geometry [Agarwal, Matousek & Schwarzkopf, 1998]. In more detail, the problem of computing many faces in an arrangement of lines consists in computing all cells in an arrangement of k lines, where each cell contains at least one point from a given set Q of m points. If each of the k lines in the arrangement corresponds to a bounding line of a shadow wedge and the set Q of points is chosen so that a large enough n-wedge intersection contains at least one point in Q, then each cell containing a point that is inside of n wedges is an n-wedge intersection. In this way, determining all large enough n-wedge intersections consists of (1) choosing a set of points accordingly, (2) computing all cells in the arrangement containing a point, (3) determining which of the computed cells are inside of n wedges and (4) checking the size of the computed n-wedge intersections.

It is proved that the total number of edges of the *marked cells*, i.e., the cells containing at least one of m points in an arrangement of k lines is $\Theta(k^{2/3}m^{2/3}+k+m)$ [Agarwal et al., 1998; Clarkson et al., 1990; Szemerédi and Trotter, 1983; Edelsbrunner and Welzl, 1986; Edelsbrunner et al., 1990]. There are many algorithms for computing marked cells presented in the literature. Most of the algorithms that are nearly worst-case optimal belong to the class of randomized algorithms, although there are also a few deterministic ones. The running time of these algorithms in our case is $\mathcal{O}(n^{4/3}m^{2/3}+n^2+m)$ modulo some logarithmic factors, since the number of lines is $\mathcal{O}(n^2)$. However, all these algorithms are rather involved and difficult to implement.

Next, we describe in detail an easy-to-implement deterministic algorithm for computing all large enough non-empty n-wedge intersections. The algorithm determines all the n-wedge intersections that contain one or more points from a given set $\mathcal{Q}$ of m points inside Γ. This set of m points is chosen such that each surely detectable object covers at least one of them. We show here that this algorithm runs in $\mathcal{O}(nm+n^2)$ time, hence, it has a slightly worse worst-case time complexity than the algorithms mentioned above for the cases of $m \geq n$.

First, we explain a preprocessing step of transforming the blocking matrix B such that the index number of the wedge of one emitter containing a given point q in the interior of Γ can be determined in constant time.

A shadow wedge $\omega = (e_i, s_j, s_k)$ is said to contain a point q in the interior of Γ if $q \in \omega \cap \Gamma$ and $q \notin (e_i, s_k)$. Hence, an interior point of Γ may be contained in at most one wedge of an emitter e_i. In this way, for each point $q \in \Gamma$, a wedge vector $z(q) = (z_0, z_1, \ldots, z_{n-1})$ is defined, where for each i either the wedge ω_{iz_i} contains q or $z_i = -1$. Note that if $z_i \neq -1$ for all $i \in \{0, 1, \ldots, n-1\}$, the n-wedge intersection corresponding to the wedge vector $z(q)$ contains point q.

The blocking matrix B is transformed by summing it with an $n \times n$ matrix B' which is given by

$$b'_{ij} = \begin{cases} -2, & \text{if } b_{ij} = 1 \\ z_i, & \text{if } b_{ij} = 0 \end{cases}$$

where z_i is the index number of the corresponding wedge of the emitter e_i. Therefore, the transformed blocking matrix B is given by

$$b_{ij} = \begin{cases} -1, & \text{if } (e_i, s_j) \text{ is visible} \\ z_i, & \text{if } (e_i, s_j) \text{ is inside the } z_i\text{-th wedge of emitter } e_i \end{cases}$$

Simply said, each value 1 in B is replaced by the value -1, while each value 0 in B is replaced by the index number z_i of the corresponding wedge. Clearly, the time complexity of the preprocessing step of transforming the blocking matrix B is $\mathcal{O}(n^2)$.

Lemma 2.3. *For an arbitrary point q in the interior of* Γ *the corresponding wedge vector* $z(q) = (z_0, z_1, \ldots, z_{n-1})$ *can be determined in* $\mathcal{O}(n)$ *time.*

Proof. Given a point q and an emitter e_i we can determine a sensor that is on or closest to the line (e_i, q) in constant time. For the resulting sensor, we can simply look up the index number of the wedge in the transformed matrix B in constant time. If the line (e_i, q) intersects the rectangle Γ between sensors s_j and s_{j+1}, of which one is blocked and the other is not, then the index of the wedge in which the blocked sensor is positioned, is used. Thus, to determine all n wedges, each from a different emitter and each containing the given point q, takes $\mathcal{O}(n)$ time, which concludes the proof of the lemma. □

From Lemma 2.2 and Lemma 2.3 we get the following result.

Corollary 2.1. *For an arbitrary point q in the interior of Γ the n-wedge intersection containing q can be determined in $\mathcal{O}(n)$ time.* □

Next, we show that we can choose a set of points in Γ such that each surely detectable object placed in Γ contains at least one of those points. From the discussion in Section 2.1, we consider only surely detectable objects, i.e., the objects that are large enough to contain a circle of radius r.

Lemma 2.4. *There is a set of $\mathcal{O}(n^2)$ points, such that each surely detectable object covers at least one of them.*

Proof. Let us define a grid G inside Γ with all horizontal and vertical line segments that have endpoints in the set of emitter and sensor points $E \cup S$. The grid is a regular square grid with nearest neighbors at a mutual distance r. Given that the area of Γ is $\mathcal{O}(n^2 r^2)$, the number of grid points in G is $\mathcal{O}(n^2)$. One can easily see that any object placed in Γ, into which a circle of radius at least r can be inscribed, will cover at least one of these points. The largest circle that does not contain a point from G in its interior has a radius of $\frac{1}{2}\sqrt{2}r < r$. □

From the theory of covering rectangles with a minimum number of circles [Melissen, 1997; Pach and Agarwal, 1995], there are sets of $\mathcal{O}(n^2)$ points in Γ that contain less points than the square grid G and for which at least one of those points is inside each object. For reasons of simplicity, let us just stick to the square grid G.

The point-by-point algorithm for determining all large enough n-wedge intersections consists of three main phases; see the pseudo-code below.

First, for each $q \in \mathcal{Q}$ we determine the wedge vector from the blocking matrix. Clearly, if a point is not in a wedge for every emitter (i.e., if the wedge vector contains one or more '-1' entries), then the resulting intersection will be empty. So, we can discard such a point. The points are being discarded with the first occurrence of the value -1 in their wedge vectors. At the end of this phase, all remaining points are inside non-empty wedge intersections, but a wedge vector can still occur multiple times (See Figure 2.8 - middle).

In the second phase of the algorithm, we further reduce the set $\mathcal{Q}$ of points by grouping points that have the same wedge vector (i.e., to keep only one representative point per group; See Figure 2.8 - right). The points of the same group are removed by using a radix sort algorithm.

In the final third phase, for each remaining point in $\mathcal{Q}$ and its corresponding wedge vector, we determine the n-wedge intersection. Then, we check whether or not the intersection is sufficiently large to be considered as an object, and if it is, we add it

Algorithm 4 POINTBYPOINTALGORITHM(B, $\mathcal{Q}$)

```
/* Phase I */
$\mathcal{P} \leftarrow \emptyset$
for each $q \in \mathcal{Q}$ do
    determine corresponding wedge vector $z(q)$;
    delete $q$ if value -1 occurs in $z(q)$
end for
/* Phase II */
group the points of $\mathcal{Q}$ that have the same wedge vector;
keep only one representative point per wedge vector and
delete other points that correspond to the same wedge vector
/* Phase III */
for each $q \in \mathcal{Q}$ and the corresponding $z(q)$ do
    determine the $n$-wedge intersection containing $q$ by calling
    DetermineIntersectionOfWedges($z(q)$)
    if the intersection is not too small then
        add the intersection to the list $\mathcal{P}$
    end if
end for
```

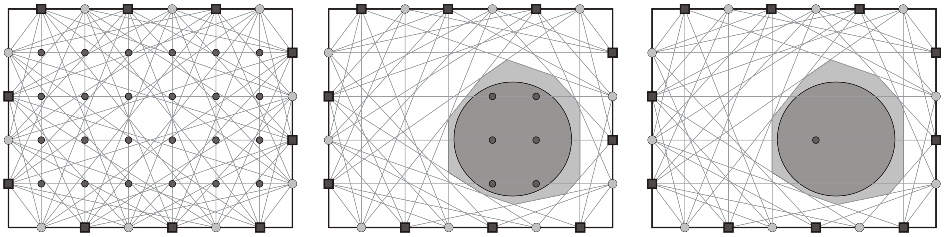

Figure 2.8. Reducing the chosen set $\mathcal{Q}$ of points: the initial set of points on the left, the points after discarding non-object points in the middle, and the point(s) after the removal of the duplicates on the right.

to the list $\mathcal{P}$ of large enough n-wedge intersections.

Next, we discuss the time complexity of the point-by-point algorithm.

In the first phase we determine the wedge vectors of at most m points. In other words, for each point q_i from the given set $\mathcal{Q}$ we determine the index numbers $z_0, \dots, z_{n-1}$ of the wedges of each of the emitters $e_0, \dots, e_{n-1}$ that contain that point. From Lemma 2.3 we know that such an operation can be done in $\mathcal{O}(n)$ time per point, resulting in $\mathcal{O}(mn)$ running time of the first phase of the algorithm.

In the second phase, we sort m points, i.e., m corresponding n-wedge vectors, using radix sort [Cormen et al., 2001]. Since the number of wedges of an emitter is at most $\lfloor n/2 \rfloor$, each value z_j of a wedge vector may be an integer in the range $[0, \lfloor \frac{n}{2} \rfloor]$, which results in the $\mathcal{O}(mn + n^2)$ running time of the second phase.

The third phase consists of determining the intersection of each of the $\mathcal{O}(m)$ ordered sets of n wedges containing a point from $\mathcal{Q}$. From Corollary 2.1, such an n-wedge intersection can be determined in $\mathcal{O}(n)$ time. Computing the area (size) of an n-wedge intersection can be also done in $\mathcal{O}(n)$ time, as we mentioned in Section 2.4, which results in the $\mathcal{O}(mn)$ running time of this third phase.

From Lemma 2.4, we have that $\mathcal{Q} = G$, i.e., $m = \mathcal{O}(n^2)$. Thus, we proved the following theorem.

Theorem 2.2. *The running time of the point-by-point algorithm is* $\mathcal{O}(n^3)$. □

Expressed in the input size ξ of a problem instance, the time complexity of the point-by-point algorithm is $\mathcal{O}(\xi\sqrt{\xi})$.

2.6 Discussion

Both cut-off and point-by-point algorithms output the set of all non-empty n-wedge intersections that are large enough to contain a circle of radius r, where n is the number of emitters, which are here assumed to be point sources of light. In practice, however, neither emitters nor sensors are points in the frame, but they have a width of approximately 5 mm. This results in having a light stripe connecting one emitter and one sensor, instead of a line segment, i.e., a ray of light. Furthermore, we assumed that each sensor detects the light from any emitter, when there is no object to block the emitted light. However, this assumption is not entirely true, given that the intensity of light emitted by one LED is not uniform over the entire angle π. More precisely, the light intensity detectable by the sensors is only within the angle of 30-40 degrees, which significantly reduces the number of lines of sight that can be used in the detection process [Jovanović, 2007].

Lines of sight defined by the emitters and sensors partition the rectangular detection area into a large number of small convex polygons. Based on this partition, one can determine the minimum size of detectable circular objects in the detection area, i.e., the mapping between each point in Γ and the minimum radius of a circular object with the center in that point that can be detected. This analysis gives rise to many optimization problems, such as the problem of finding the positioning of a fixed number of emitters and sensors that provides the optimal accuracy of object detection [Jovanović, 2007].

Ideally, each non-empty n-wedge intersection corresponds to one object placed in the detection area. However, one n-wedge intersection can correspond to 0, 1 or more objects, as we explained in Section 2.1. One can reason that this phenomenon is a direct consequence of a large number of blocked lines of sight. The number of blocked and non-blocked lines of sight is implied by two essentially different classes of parameters: the *object parameters*, regarding the objects that should be detected and the *environmental parameters*, regarding the whole setup of the emitters and sensors within some detection area. The variation of the object parameters can be limited, for example, when the detection device is intended to recognize game pieces only. To optimize the detection in this case, the game pieces can be, for instance, identical in the detection plane, or especially designed for the device allowing always the light from emitters to reach some of the sensors. For a general purpose detection device, however, it is very important to explore the environmental parameters. Clearly, the larger the number of emitters and sensors, the larger is the number of lines of sight defined by them, which consequently results in higher accuracy of detecting objects. The question that logically follows then is whether or not it is possible to detect the objects exactly as they are, and if not, what is the maximum level of accuracy that can be achieved. In the next chapter, we will discuss the output of the detection algorithms and some of its properties in the asymptotic case, assuming the detection area to be a two-dimensional plane and each line in that plane to be a line of sight between one emitter and one sensor.

3

Limitations of in-plane object detection

Object detection using light emitters and light sensors is based on the information which of the lines of sight connecting one emitter and one sensor are blocked by the objects. The detection algorithms described in Chapter 2 use this information to provide an approximation on the geometry of the objects placed in the detection area. In order to achieve a satisfactory level of accuracy of object detection, the detection area should be ”well covered” by the (intersecting) lines of sight [Jovanović, 2007]. Finding the positioning of the fixed number of emitters and sensors that ensures a high level of accuracy is one question that has been considered in [Jovanović, 2007]. Here, we focus on analyzing the limitations of the detection technique, by exploring the potential problems that can rise even in the conditions of an ideal environment.

In order to examine the limits of the detection technique, we assume here a less realistic environment setup than the one discussed in Chapter 2. We consider in this chapter the case where the distance r between a pair of neighboring emitters and sensors goes to 0. In other words, we assume that there are infinitely many emitters and sensors in a frame in the plane surrounding objects and that each line in the plane connects one emitter and one sensor. Furthermore, we restrict ourselves to detecting only circular objects of the same size. In this way, by setting up the ideal detection environment, we can concentrate on an intrinsic shortcoming of the detection technique: the objects cannot be ideally detected, i.e., they cannot be recognized exactly as they are. The objects themselves strongly influence the level of the detection accuracy. Even more, all the detection problems that have been identified in practice, such as reporting a non-existing object, still occur in the ideal environment. This gives rise to the shadow regions problem defined below, that is the main problem considered in this chapter.

Let $\mathcal{D}$ be a set of N closed and non-overlapping unit disks, i.e., disks with radius 1, in the two-dimensional plane. Each disk of $\mathcal{D}$ represents a circular object in the detection area.

Definition 3.1 (blocked line). A line ℓ is called blocked if it has a non-empty intersection with at least one of the disks in $\mathcal{D}$. $\square$

Definition 3.2 (shadow point). A point p is called a shadow point, if all lines containing p are blocked. $\square$

By definition, each point in a disk $\delta \in \mathcal{D}$ is a shadow point. A point that is not a shadow point, is called a *light point*.

Lemma 3.1. *A line ℓ that is not blocked only contains light points.*

Proof. The claim follows directly from the definition of a shadow point. $\square$

For a light point p it holds that there is at least one line in the plane that does not intersect any of the disks in $\mathcal{D}$. As a consequence, all the points outside the convex hull of disks are light. In other words, all shadow points defined by the disks in $\mathcal{D}$ are inside the convex hull of disks, denoted as $\mathcal{H}(\mathcal{D})$.

Definition 3.3 (shadow region). A closed shape ς in the plane is a shadow region, if each point in ς is a shadow point and if ς is connected and maximal in the sense that there is no shape ς' containing only shadow points for which $\varsigma \subset \varsigma'$. $\square$

One can observe that a shadow region ς is bounded by line segments and/or circular arcs, where the former are referred to as *edges* of ς.

From the definition of a shadow region, it follows that the collection of shadow regions partition the set of shadow points. By definition, each disk $\delta \in \mathcal{D}$ is contained in a shadow region.

Shadow Regions Problem. *Given the set $\mathcal{D}$ of disks, determine the set $\mathcal{S}$ of all shadow regions in the plane.*

In other words, we are interested in designing an efficient algorithm that outputs the set of all shadow regions, for a given set $\mathcal{D}$ of disks. Figure 3.1 illustrates a set of 281 shadow regions defined by 70 randomly positioned unit disks. Note that the set of all shadow regions in the plane defined by the disks in $\mathcal{D}$ represents the output

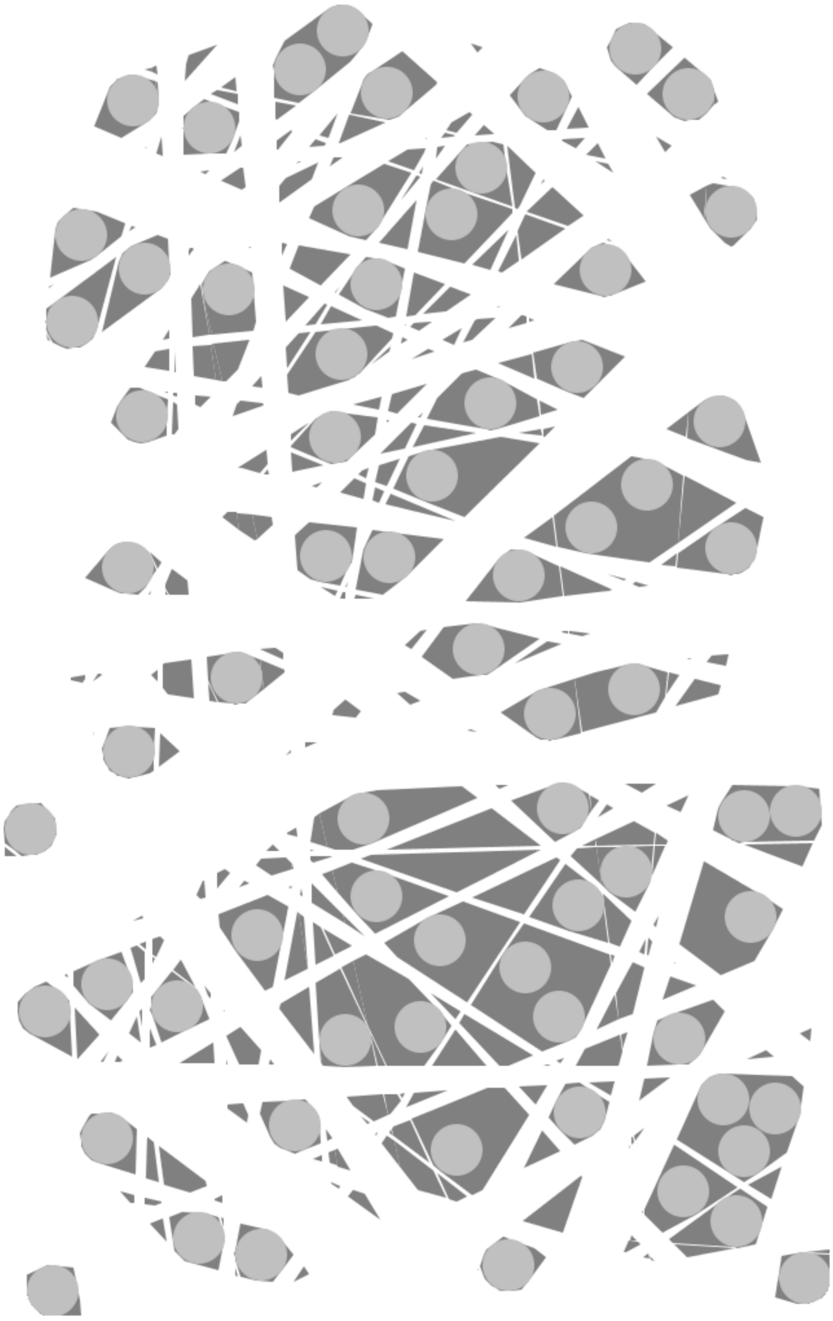

Figure 3.1. The shadow regions defined by 70 unit disks.

of the detection algorithms in the asymptotic case, i.e., in the ideal environment with infinitely many emitters and sensors in a frame surrounding the objects, thus, infinitely many lines of sight.

Overview. This chapter is organized as follows. In Section 3.1, we describe some basic properties of the shadow regions and show that they are determined by the tangent lines defined by the pairs of disks in $\mathcal{D}$. In Section 3.2 we define so-called light corridors, which are areas inside $\mathcal{H}(\mathcal{D})$ that contain only light points. We establish the relation between shadow regions and light corridors and prove that the number of light corridors is bounded by N^2, where N denotes the number of disks. We present an $\mathcal{O}(N^4)$ algorithm for computing all shadow regions in Section 3.3. In Section 3.4 we show that the number of shadow regions is $\Omega(N^4)$ for some instances. In this way, we prove the optimality of the worst-case time complexity of the presented algorithm. Finally, we discuss the relation between the shadow regions and the output of the detection algorithms from Chapter 2 in Section 3.5.

3.1 Introducing shadow regions

Let ℓ be a line in the plane such that it intersects the convex hull $\mathcal{H}(\mathcal{D})$ of disks.

Definition 3.4 (defining line). A line ℓ is called a defining line for a shadow region ς if it contains an edge of ς. □

Lemma 3.2. *Let ℓ be a defining line for a shadow region ς. Then the following holds:*

- *ℓ does not intersect any disk in $\mathcal{D}$ in more than one point*
- *ℓ is tangent to at least two disks in $\mathcal{D}$*
- *ℓ is not tangent to any three disks δ_1, δ_2 and δ_3, where δ_1 and δ_2 are on the same side of ℓ and δ_3 is such that its point of tangency with ℓ is between the points of tangency of δ_1 and δ_2 with ℓ.*

Proof. We prove the lemma by contradiction. Hence, suppose that ℓ is a defining line for a shadow region ς and suppose that (1) ℓ intersects a disk in $\mathcal{D}$ in more than one point, (2) ℓ does not intersect with any disk, (3) ℓ is tangent to exactly one disk or (4) ℓ is tangent to three disks δ_1, δ_2 and δ_3, where δ_3 is between δ_1 and δ_2 assuming that the disks are ordered by their points of tangency with ℓ. We now show that all these cases yield a contradiction.

1. Let ℓ intersect a disk $\delta \in \mathcal{D}$ in more than one point, and let ℓ be a defining line for a shadow region ς, i.e., ℓ contains an edge (q_1, q_2) of ς; see Figure 3.2. Let ℓ_1 and ℓ_3 be two lines that contain point q_1 and that are tangent to δ,

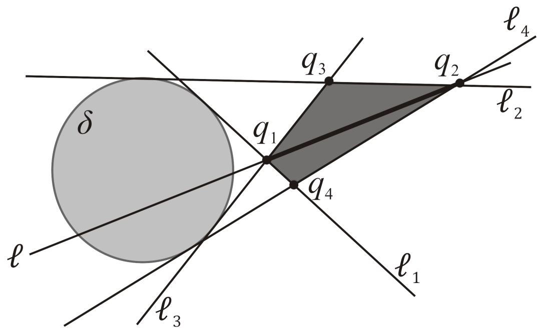

Figure 3.2. The line ℓ that intersects δ in more than one point cannot define a shadow region.

and let ℓ_2 and ℓ_4 be two lines that contain point q_2 and that are tangent to δ. We denote $q_3 = \ell_2 \cap \ell_3$ and $q_4 = \ell_4 \cap \ell_1$. If we chose an arbitrary point p inside the quadrilateral $q_1q_3q_2q_4$, then each line containing p is either blocked by the disk δ, or it intersects the edge (q_1, q_2). By assumption, (q_1, q_2) contains only shadow points, thus, each line intersecting (q_1, q_2) is blocked, as follows from Lemma 3.1. Hence, quadrilateral $q_1q_3q_2q_4$ is a shadow region, which is in contradiction with (q_1, q_2) being an edge of a shadow region.

2. If ℓ does not intersect any disk in $\mathcal{D}$, then ℓ contains only light points. Hence, ℓ does not define any shadow region.

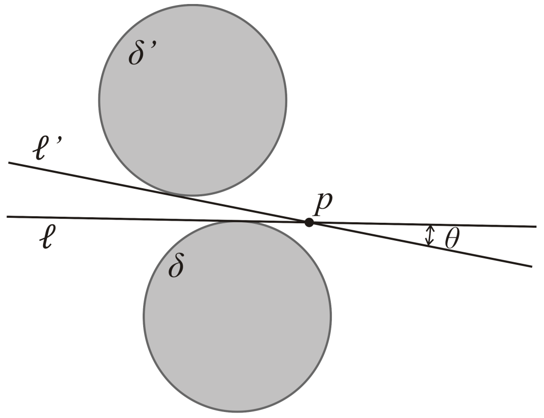

Figure 3.3. The line ℓ that is not tangent to more than one disk cannot contain the shadow point $p \notin \delta$.

3. Let ℓ be tangent to a disk $\delta \in \mathcal{D}$, such that it does not intersect any other disk

in $\mathcal{D}$. In order for ℓ to be a defining line it should contain a shadow point p outside δ. Let $p \in \ell$ with $p \notin \delta$ be a shadow point; see Figure 3.3. Since ℓ is not tangent to any other disk in $\mathcal{D}$, ℓ can be rotated around p over some angle θ, in the direction away from the disk δ, until it becomes tangent to some disk $\delta' \in \mathcal{D}$. We denote the rotated line as ℓ'. Then, any line containing p that is inside the angle θ between ℓ and ℓ' is not blocked, which is in contradiction with p being a shadow point.

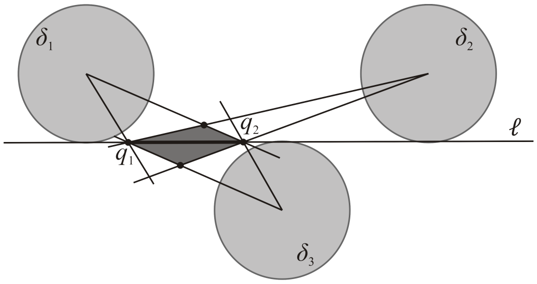

Figure 3.4. The line ℓ that is tangent to three disks ordered as illustrated cannot define a shadow region.

4. Let ℓ be a line tangent to three disks δ_1, δ_2 and δ_3, where δ_3 is between δ_1 and δ_2 when the disks are ordered by their points of tangency with ℓ; see Figure 3.4. Furthermore, let ℓ contain a shadow edge (q_1, q_2). There are several cases of different positioning of the shadow edge (q_1, q_2) that need to be considered. Here, we prove only the case when (q_1, q_2) is between the points of tangency of δ_1 and δ_3. The other cases can be proved in a similar fashion. Now, let a shadow edge (q_1, q_2) be between the points of tangency of δ_1 and δ_3. We connect the centers of the disks with the points q_1 and q_2, defining in that way six lines. Then, the smallest quadrilateral defined by these lines that has (q_1, q_2) as its diagonal is a shadow region, which is in contradiction with (q_1, q_2) being an edge of a shadow region.

□

Now, let us take a look at some small examples of $\mathcal{D}$, so that we can get a notion of the size, shape and the number of shadow regions defined by the disks and the corresponding number of edges.

From the definition of a shadow region in the previous section, the trivial case of $\mathcal{D}$ consisting of only one disk results in the disk being the only shadow region in the plane having no edges.

Let $N = 2$, i.e., the set $\mathcal{D}$ consists of two non-overlapping unit disks. Two non-tangent disks define four common tangent lines: a pair of external (parallel) tangent

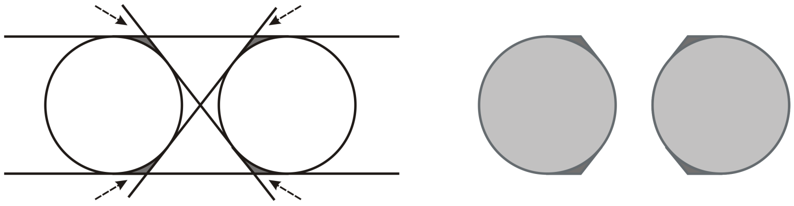

Figure 3.5. The shadow regions defined by 2 unit disks; the arrows point at the small attached shadow areas.

lines and a pair of internal (crossing) tangent lines. The four tangent lines define four shadow areas that are attached to the disks; see Figure 3.5. By definition, a disk and all its attached shadow areas represent one shadow region. Hence, two disks define at most two shadow regions. Note that the size of the shadow regions depends on the distance between the two disks: the closer the disks, the larger the shadow regions.

In the case of two disks being mutually tangent, there is only one shadow region, as depicted in Figure 3.6. This occurs due to degeneracy of two internal tangent lines to only one common tangent line.

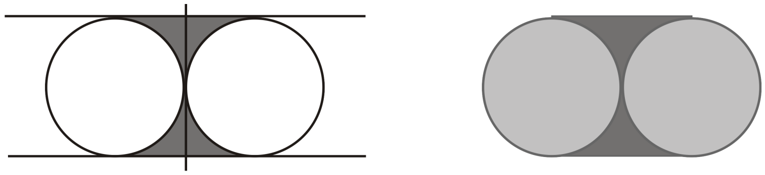

Figure 3.6. The shadow region defined by two tangent unit disks.

Now, let $N = 3$. Depending on the mutual distance, the three disks may define one or more *free shadow regions*, i.e., shadow areas that are not attached to any of the disks; see Figure 3.7. A free shadow region is bounded by line segments only, thus, it has the shape of a polygon. It can be shown that three disks can define at most four free shadow regions, which implies that they can define 1 to 7 shadow regions in total.

From the description above, a shadow region can be formally represented by a cyclic sequence of points $p_0, p_1, \dots, p_k$, ordered in clockwise order, such that p_{i+1} is the successor of p_i, for $i = 0, 1, \dots, k-1$, and p_0 is the successor of p_k. Each two neighboring points are connected by either a line segment or a circular arc of radius 1; see Figure 3.8. For example, the shadow region illustrated in Figure 3.8 is denoted as $\widehat{p_0 p_1} p_2 \widehat{p_3 p_4} p_5$. Note that two successive points that lie on the same disk are, by definition, connected by a circular arc, and two successive points not lying on the same disk are connected by a line segment. In addition, there is always

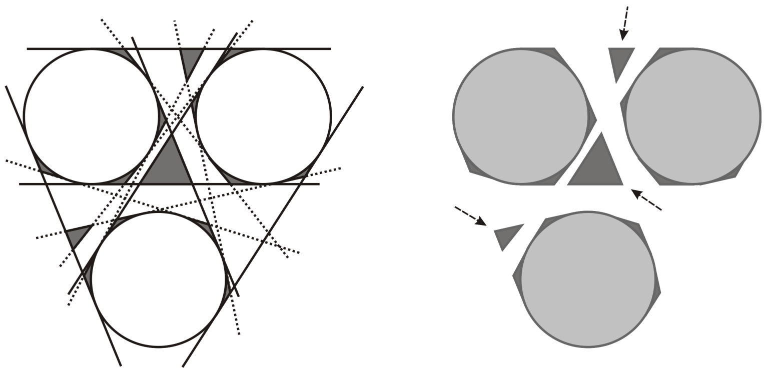

Figure 3.7. The shadow regions defined by 3 unit disks; the arrows point at the free shadow areas.

at least one line segment between two successive circular arcs.

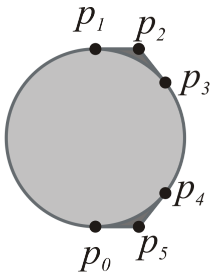

Figure 3.8. The shadow region $\widehat{p_0p_1}p_2\widehat{p_3p_4}p_5$ defined by a set of points connected by either a circular arc or a line segment.

We observe that N disks define at most $2N(N-1)$ common tangent lines, which can partition the plane into $\mathcal{O}(N^4)$ non-overlapping convex polygons that contain either shadow points only or light points only. In Section 3.4, we will prove that there are instances for which the number of shadow regions defined by N disks is $\Omega(N^4)$.

Let us first prove some lemmas that will turn out useful in proving our main result.

Lemma 3.3. *A shadow region is convex.*

Proof. We prove the lemma by contradiction. Hence, assume that a shadow region ς is not convex. Let p be a light point inside the convex hull $\mathcal{H}(\varsigma)$ of ς and outside ς. Each line containing p intersects the shadow region ς, which implies that it is blocked. This implies that p is a shadow point, which contradicts with the assumption of p being a light point. □

As a consequence of Lemma 3.2, in the process of determining the shadow regions, we consider only the set $\mathcal{T}$ of defining lines, i.e., from the set of all tangent lines defined by the disks, we exclude the lines for which at least one of the three conditions in Lemma 3.2 does not hold.

Let $t \in \mathcal{T}$ be a line tangent to two disks δ_1 and δ_2 in $\mathcal{D}$. The points of tangency between disks δ_1 and δ_2 and line t we denote as p_{δ_1} and p_{δ_2}. Furthermore, let δ_1 and δ_2 be such that there is no other disk δ_3 in $\mathcal{D}$ tangent to t such that its point of tangency p_{δ_3} with t is between p_{δ_1} and p_{δ_2}. The points of tangency p_{δ_1} and p_{δ_2} divide t into three parts: one line segment $(p_{\delta_1}, p_{\delta_2})$ connecting p_{δ_1} and p_{δ_2}, and two rays with apices in p_{δ_1} and p_{δ_2} denoted by ρ_{δ_1} and ρ_{δ_2}.

Lemma 3.4. *If disks δ_1 and δ_2 are not on the same side of t, line segment $(p_{\delta_1}, p_{\delta_2})$ does not define a shadow region. If disks δ_1 and δ_2 are on the same side of t, the rays ρ_{δ_1} and ρ_{δ_2} do not define a shadow region.*

Proof. We prove only the case of disks δ_1 and δ_2 being on different sides of t. The other case can be proved in the similar way.
Let t be an internal tangent line for disks δ_1 and δ_2 and let the line segment $(p_{\delta_1}, p_{\delta_2})$ contain a shadow point p, such that $p \notin \delta_1$ and $p \notin \delta_2$. If t is not a tangent line for any other disk except δ_1 and δ_2, then t can be rotated around point p over some angle θ, in the direction that it does not intersect the disks δ_1 and δ_2, until it becomes tangent to some disk $\delta_3 \in \mathcal{D}$ or eventually, to both δ_1 and δ_2. Then, any line containing p "inside" θ is not blocked, which is in contradiction with p being a shadow point. If t is also a tangent line to a disk δ_4, so that it is not possible to rotate t around p over any angle θ without intersecting at least one of the disks δ_1, δ_2 and δ_4, then from the third condition of Lemma 3.2, t is not a defining line for any shadow region. $\square$

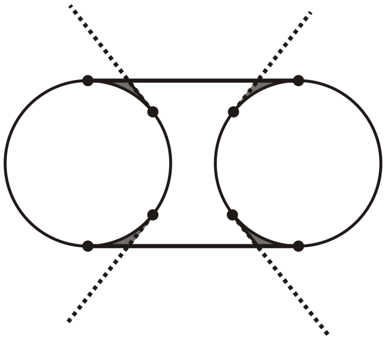

Figure 3.9. Parts of the tangent lines that define the shadow regions.

From Lemma 3.4, an internal tangent line may be involved in the definition of shadow regions through the pair of rays with apices in the points of tangency. The external tangent lines are involved in the definition of shadow regions through the line segments connecting the points of tangency; see Figure 3.9.

3.2 Modelling Light Corridors

Let $\mathcal{L}$ be the set of all lines in the plane that do not intersect any disk, hence, $\mathcal{L}$ is the set of lines that only contain light points. Set $\mathcal{L}$ can be partitioned into two subsets, called *dividing lines* and *non-dividing lines*. For a non-dividing line all disks are on the same side of that line. Each dividing line specifies a partition of the set of disks into two non-empty sets.

Definition 3.5 (light corridor). The collection of dividing lines specifying the same bipartition of the disks in $\mathcal{D}$ is called a light corridor. $\square$

An illustration of a light corridor is given in Figure 3.10. By definition, a light corridor contains only light points. Note that each light point inside $\mathcal{H}(\mathcal{D})$ is contained in one or more light corridors. This means that the collection of shadow regions is given by the difference between $\mathcal{H}(\mathcal{D})$ and the union of all light corridors. Clearly, if the set of dividing lines is empty, then there is only one shadow region given by $\mathcal{H}(\mathcal{D})$.

Let $\mathcal{T}$ be the set of all defining lines. A light corridor can be characterized by its so-called *supporting lines*, i.e., two internal tangent lines t and t' in $\mathcal{T}$ that are clockwise fixed and counterclockwise fixed, respectively; see Figure 3.10. For t this means that it cannot be rotated in clockwise direction around any point on the line over any angle θ such that it does not intersect at least one of the disks in more than one point. An analogous interpretation holds for t'. The supporting lines define the "in" and "out" of the corridor through $\mathcal{H}(\mathcal{D})$. Inside the convex hull $\mathcal{H}(\mathcal{D})$, each light corridor is an open non-convex area, bounded by a set of line segments and a set of circular arcs of radius 1.

A light corridor is denoted by $\delta_1\delta_2\ldots\delta_j|\delta_{j+1}\delta_{j+2}\ldots\delta_k$, where disks $\delta_1\delta_2\ldots\delta_j$ define one side of the corridor, the disks $\delta_{j+1}\delta_{j+2}\ldots\delta_k$ define the other side of the corridor, and disks δ_j and δ_{j+1} define the clockwise tangent t and disks δ_1 and δ_k define the counterclockwise tangent t'; see Figure 3.10.

Lemma 3.5. *The number of light corridors defined by N non-overlapping unit disks is at most $N(N-1)/2$.*

Proof. The N disks define at most $N(N-1)$ internal tangent lines in $\mathcal{T}$. Each internal tangent line in $\mathcal{T}$ defines one bipartition of disks, which corresponds to

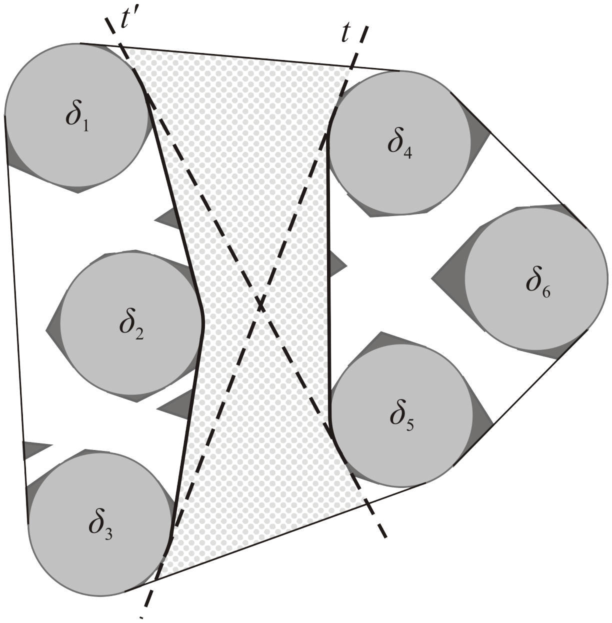

Figure 3.10. An example of a light corridor.

exactly one light corridor. Hence, the number of light corridors is not larger than the number of internal tangent lines in $\mathcal{T}$. Moreover, each light corridor is characterized by a pair of internal tangent lines, which implies that the number of light corridors is at most $N(N-1)/2$. □

The upper bound on the number of light corridors presented in Lemma 3.5 is tight. To prove this claim, we present an example of a linear number of disks creating a quadratic number of light corridors; see Figure 3.11. More precisely, N disks are placed on a large circular ring in such a way that their centers form a regular N-gon. In this way, each two pairs of neighboring disks in the ring define one light corridor, i.e., exactly $N-1$ light corridors "begin" between the same pair of neighboring disks, as illustrated in Figure 3.11. Hence, the disks define $N(N-1)/2$ light corridors in total.

3.3 Shadow regions algorithm

In this section, we present an algorithm for determining the set of all shadow regions defined by N non-overlapping unit disks. We give the algorithm in a step-by-step manner and discuss its overall time complexity.

The algorithm for determining all shadow regions defined by N non-overlapping

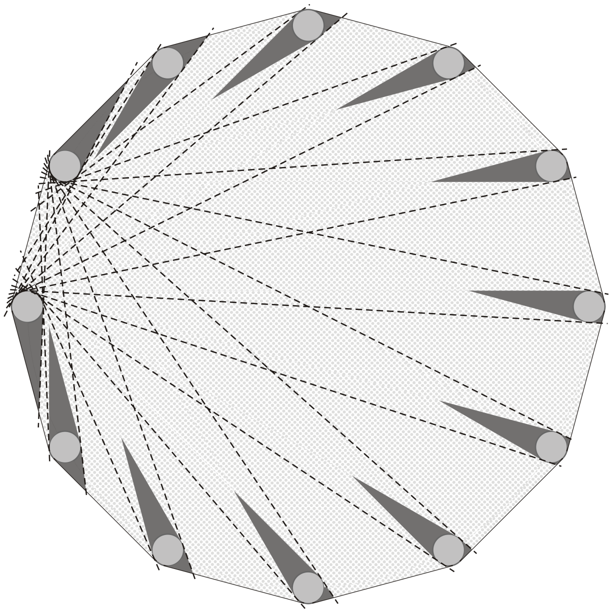

Figure 3.11. A linear number of disks creating a quadratic number of light corridors: twelve disks define eleven light corridors per one pair of neighboring disks.

unit disks consists of the following four main steps:

1. Determine the convex hull $\mathcal{H}(\mathcal{D})$;
2. Determine the set $\mathcal{T}$ of all defining tangent lines;
3. Determine all light corridors inside $\mathcal{H}(\mathcal{D})$;
4. Determine the union $\mathcal{U}$ of all light corridors and next, the set of all shadow regions, by finding the set difference between $\mathcal{H}(\mathcal{D})$ and $\mathcal{U}$.

Let us now take a closer look at each step of the algorithm and its worst-case time complexity.

The problem of determining the convex hull $\mathcal{H}(\mathcal{D})$ spanned by the disks in $\mathcal{D}$ consists of (1) finding the convex hull of a set of points, in our case, the disks' centers and (2) offsetting the resulting polygon from the first step, i.e., determining the Minkowski sum of a polygon and a unit disk. Both problems are well-known in the field of computational geometry [De Berg et al., 2000], and without giving any details, the first step of determining the convex hull $\mathcal{H}(\mathcal{D})$ can be done in $\mathcal{O}(N \log N)$ time.

The set $\mathcal{T}$ of all defining lines, as defined in Section 3.1, can be determined in $\mathcal{O}(N^2 \log N)$ time, as follows. For each disk in $\mathcal{D}$, we sort radially the other $N-1$

disks, which takes $\mathcal{O}(N \log N)$ time. This structure allows us to find in linear time all tangent lines of one disk that do not intersect any of the disks in $\mathcal{D}$ in more than one point. From the first two conditions of Lemma 3.2, these lines are potentially the defining lines. In addition to each potentially defining line determined, we keep the information on tangent disks and the tangency points, and the part(s) of the line which are involved in the definition of the shadow regions, i.e., the rays or the line segment, as explained in Lemma 3.4. If a line is tangent to more than two disks, then there are multiple defining parts associated with the line. A line that cannot be a defining line according to the third condition of Lemma 3.2 is associated with at least one pair of non-identical rays with a common apex. Using this information, we can look for and remove the lines that do not satisfy the third condition of Lemma 3.2. Hence, it takes $\mathcal{O}(N \log N)$ time to determine all defining lines of one disk and all the additional properties. Therefore, finding the set $\mathcal{T}$ of defining tangent lines for all N disks takes $\mathcal{O}(N^2 \log N)$ time.

We can determine the set of all light corridors in $\mathcal{O}(N^2)$ time if, for each of the disks, we have a list of all its points of tangency sorted in cyclic order. Such lists can be determined in $\mathcal{O}(N^2 \log N)$ time since all the defining lines are determined, hence, all the points of tangency for each of the disks. As a result, the third step of the algorithm can be implemented to run in $\mathcal{O}(N^2 \log N)$ time.

As mentioned in the proof of Lemma 3.5, an internal tangent line in $\mathcal{T}$ characterizes one light corridor. Starting with an internal line from $\mathcal{T}$, we determine the corresponding light corridor as follows. We start by including one ray of the chosen internal line. Then, we simply look up the corresponding point of tangency on the tangent disk and take the successor point of tangency from the sorted list of points for that disk. That point is a starting point for either a line segment, or a ray of some other tangent line. In the case of a starting point of a line segment, we look up the ending point on the next disk, etc. The computation of one side of the corridor is finished when a ray occurs in the sequence. In other words, we determine the sequence ray-arc-segment-arc-segment-arc-$\cdots$-arc-ray, which uniquely represents one side of the corridor. The other side is determined in the same way, starting with the other ray of the originally chosen internal tangent line.

From Lemma 3.5, the number of light corridors is $\mathcal{O}(N^2)$. In addition, the number of all defining tangent lines is also quadratic in the number of disks, which implies that the total number of all rays (2 rays per internal tangent) and line segments (1 line segment per external tangent) together is also $\mathcal{O}(N^2)$. In this way, amortized over all iterations, the light corridors can be determined in $\mathcal{O}(N^2)$ time. Hence, the total time complexity of the third step of the algorithm is dominated by the complexity of the sorting algorithm used to sort the points of tangency for each of the disks. In other words, the third step of the algorithm runs in $\mathcal{O}(N^2 \log N)$ time.

The problem of determining the union $\mathcal{U}$ of all light corridors comes down to the problem of finding the intersections of a set of line segments and circular arcs. This is a well-known and extensively studied problem and there are many solutions proposed [De Berg et al., 2000]. Using the deterministic algorithm presented in [Balaban, 1995], the intersections of n_s line or curve segments can be determined in $\mathcal{O}(n_s \log n_s + k_s)$ time, where k_s is the number of intersecting pairs. Given that we have $\mathcal{O}(N^2)$ line segments and circular arcs, the number k_s of intersecting pairs is $\mathcal{O}(N^4)$. Therefore, using this algorithm, the union $\mathcal{U}$ of all light corridors can be determined in $\mathcal{O}(N^4)$ time. The set of all shadow regions is then simply determined as a complement set of $\mathcal{U}$ within the convex hull $\mathcal{H}(\mathcal{D})$.

With the discussion above, we get to the following result.

Theorem 3.1. *The set of all shadow regions defined by N non-overlapping unit disks can be determined in* $\mathcal{O}(N^4)$ *time.* □

3.4 Determining the number of shadow regions

In this section, we discuss the number of shadow regions that can be created by the disks in $\mathcal{D}$. More precisely, we prove that the number of shadow regions is $\mathcal{O}(N^4)$ and then show that this bound is tight, by presenting an example of linear number of disks defining $\Omega(N^4)$ shadow regions. This implies that the $\mathcal{O}(N^4)$ algorithm presented in Section 3.3 determines the set of all shadow regions in optimal time.

Lemma 3.6. *The number of shadow regions defined by N non-overlapping unit disks is* $\mathcal{O}(N^4)$.

Proof. As mentioned in Section 3.1, N unit disks define $\mathcal{O}(N^2)$ tangent lines that partition the plane into $\mathcal{O}(N^4)$ convex polygons. A shadow region is a subset of one or more of these polygons, where by definition a polygon is in at most one shadow region. Hence, the number of shadow regions is $\mathcal{O}(N^4)$. □

Next, we prove that the bound presented in Lemma 3.6 is tight. The proof is based on a specific construction containing two ”columns” of disks, each column containing N equidistant disks, such that each disk of one column is directly opposite to a disk of the other column. The idea behind the construction is to obtain a quadratic number of thin light corridors that pass between the disks of the two columns, i.e., in the left to right direction. If these corridors do not intersect within some finite area of width w, then adding another two ”rows” of N disks each, in the same way, creates a quadratic number of light corridors in the top-bottom direction, resulting in $\Theta(N^4)$ shadow regions in an area of $w \times w$; see an illustration in Figure 3.12.

If we need to add only linear number of mutually tangent disks to block the light corridors that come from other (e.g., diagonal) directions, we then have a linear number of disks creating $\Theta(N^4)$ shadow regions.

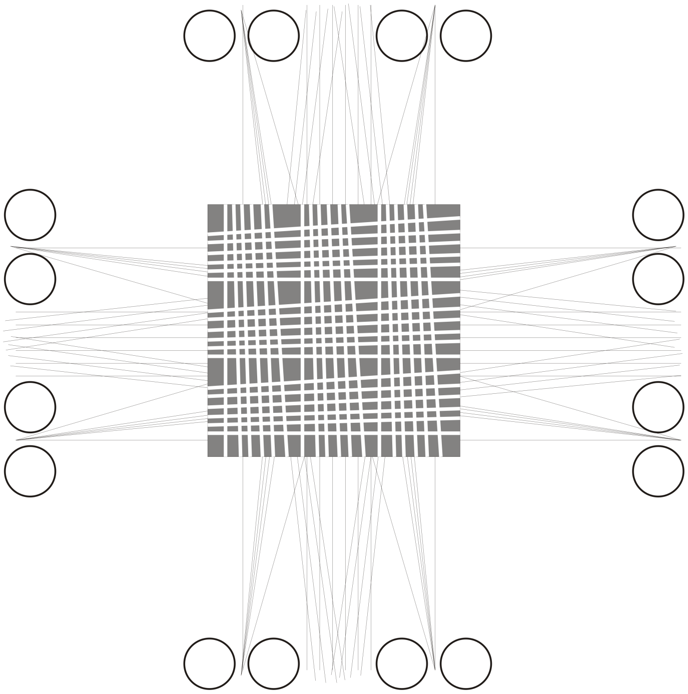

Figure 3.12. Constructing $\Theta(N^4)$ shadow regions with a linear number of disks.

Let ℓ_{left} be the line connecting the centers $O_1, O_2, \ldots, O_N$ of the disks $\delta_1, \delta_2, \ldots, \delta_N$ in the left column and in the same fashion, let ℓ_{right} be the line connecting the centers $O'_1, O'_2, \ldots, O'_N$ of the disks $\delta'_1, \delta'_2, \ldots, \delta'_N$ in the right column; see Figure 3.13. Furthermore, let h denote the distance between the columns, i.e., the distance between ℓ_{left} and ℓ_{right}, and let d denote the distance between two neighboring disks in one column, measured from center to center. Given h, the distance d is chosen so that the top two disks of one column and the bottom two disks of the other column are all tangent to the same line. In this way, there is no light corridor defined by these four disks, however, there is exactly one light corridor between any other two pairs of neighboring disks in different columns. From the congruence of the grey triangles in Figure 3.13, we can find the relation between the distances d and h, i.e.,

$$\frac{1}{d/2} = \frac{h}{\sqrt{h^2 + (N-2)^2 d^2}}$$

from where we can express the distance d as a function of h

$$d = \frac{2h}{\sqrt{h^2 - 4(N-2)^2}} \tag{3.1}$$

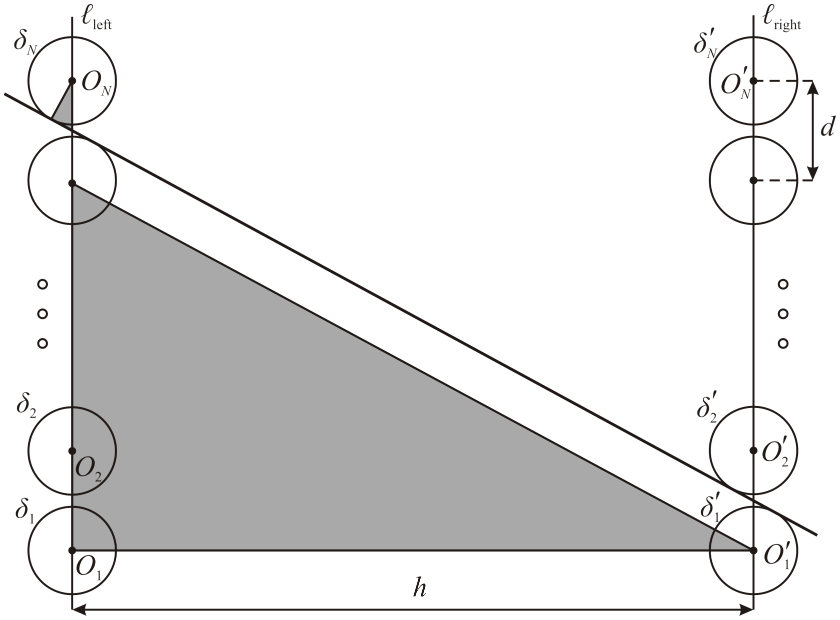

Figure 3.13. The columns of disks, each column containing N disks.

For the time being, we consider only the light corridors between pairs (δ_i, δ_{i+1}) of neighboring disks from the left column and pairs $(\delta'_j, \delta'_{j+1})$ of neighboring disks from the right column, where $i, j \in \{1, \ldots, N-1\}$.

The distance d between the neighboring disks is determined by the distance h between the columns. From Equation (3.1), we get that if $h \to \infty$, then $d \to 2$. Using elementary calculus, it can be shown that increasing the distance h between the columns results in decreasing the width of the corridors. Note that the corridors are not all of the same width, i.e., the long corridors are thinner than the short corridors.

It remains to be shown that there is an area between the columns where no two corridors intersect. Furthermore, we want to show that for some h, the width w of that area can be at least $N \cdot d$. In this way, overlapping (or intersecting) this area containing the left to right non-intersecting corridors with the area containing the top to bottom non-intersecting corridors, results in creating $\Theta(N^4)$ shadow regions.

In Section 3.2 we showed that each light corridor is characterized by a pair of two internal tangent lines. In this special case of disks being placed in two columns, one can easily notice that, between the columns, each corridor is bounded by a pair of parallel line segments. Without loss of generality, we consider the left column as the beginning and the right column as the end of the corridors. Among the intersection points of the corridors' bounding line segments, we can distinguish two subsets of points: the *splitting points* and the *meeting points*, as illustrated in Figure 3.14. The splitting point of two light corridors that begin between the same pair of disks is the common (intersection) point of these corridors furthest from ℓ_{left}. In a similar way, the meeting point of two corridors that do not begin between the same pair of disks is the intersection point of these two corridors closest to ℓ_{left}.

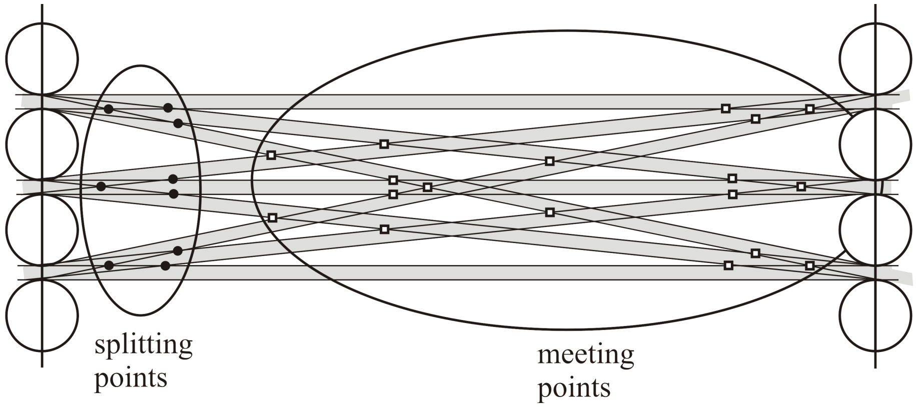

Figure 3.14. The splitting points and the meeting points of nine light corridors passing between eight disks in the columns.

Let ℓ_s denote the vertical line containing the splitting point(s) furthest from ℓ_{left} and let ℓ_m denote the vertical line containing the meeting point(s) closest to ℓ_{left}. Clearly, if the distance $\bar{h}_s$ between ℓ_s and ℓ_{left} is smaller than the distance $\bar{h}_m$ between ℓ_m and ℓ_{left}, the area between two vertical lines ℓ_s and ℓ_m, is the area inside which no two corridors intersect. In addition, the width w of the area is given by

$$w = \bar{h}_m - \bar{h}_s \tag{3.2}$$

Next, we express the distance $\bar{h}_s$ as a function of the distance h between the columns of disks. Let us consider only the $N-1$ light corridors that all begin between one pair of neighboring disks in the left column. One can show that among the splitting points of these corridors, the splitting point furthest from ℓ_{left}, is the splitting point of two neighboring light corridors, i.e., the corridors that end between the neighboring pairs of disks in the right column. Let h_s be the distance from the splitting point Q of an arbitrary pair of neighboring corridors to the line ℓ_{left}. Clearly, $\bar{h}_s$ is the maximum of all distances h_s of the splitting points of all neighboring corridors.

Lemma 3.7. *For an arbitrary pair of neighboring light corridors C_j and C_{j+1}, it holds that*

$$\lim_{h \to \infty} h_s = 0.$$

Proof. Let C_j and C_{j+1} be two neighboring corridors that begin between the disks δ_j and δ_{j+1} with centers O_j and O_{j+1} on ℓ_{left}, respectively; see Figure 3.15. Furthermore, let C_j end between the disks δ'_{j+k-1} and δ'_{j+k} and let C_{j+1} end between the disks δ'_{j+k} and δ'_{j+k+1}.

The splitting point Q of the corridors C_j and C_{j+1} is the intersection point of the tangent lines t_j and t_{j+1}.The point of tangency between the disk δ_j and the line t_j is denoted as T_j, while the point of tangency between the disk δ_{j+1} and the line t_{j+1} is denoted as T_{j+1}. In addition, let $Q_j = \ell_{\text{left}} \cap t_j$ and $Q_{j+1} = \ell_{\text{left}} \cap t_{j+1}$.
Since $\angle O_j O'_{j+k} O_{j+k} = \angle Q_j O_j T_j$ and $\angle O_{j+1} O'_{j+k} O_{j+k} = \angle Q_{j+1} O_{j+1} T_{j+1}$, and by denoting

$$\alpha = \angle Q_j O_j T_j, \quad \beta = \angle Q_{j+1} O_{j+1} T_{j+1},$$
$$u = Q_j Q_{j+1}, \;\; u_j = O_j Q_j, \;\; u_{j+1} = O_{j+1} Q_{j+1},$$

we can express the relations

$$\begin{aligned} u_j &= \frac{1}{\cos\alpha}, \quad u_{j+1} = \frac{1}{\cos\beta}, \\ u &= d - (u_j + u_{j+1}) = d - (\frac{1}{\cos\alpha} + \frac{1}{\cos\beta}) \end{aligned} \tag{3.3}$$

From (3.3) and

$$\triangle Q Q_j Q_{j+1} \sim \triangle O'_{j+k} O_j O_{j+1} \tag{3.4}$$

we derive the distance h_s as follows.

$$\begin{aligned} \frac{u}{h_s} &= \frac{d}{h} \\ \Leftrightarrow \quad h_s &= \frac{u \cdot h}{d}. \end{aligned} \tag{3.5}$$

Since $O_j O_{j+k} = k \cdot d$ and

$$\cos\alpha = \frac{h}{\sqrt{h^2 + k^2 d^2}}, \quad \cos\beta = \frac{h}{\sqrt{h^2 + (k-1)^2 d^2}},$$

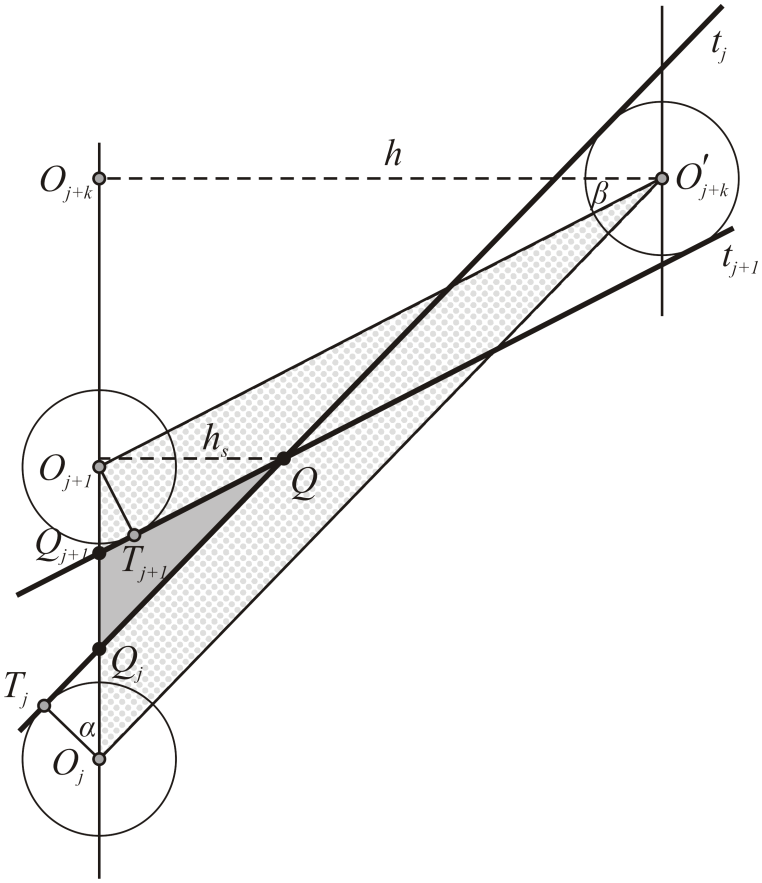

Figure 3.15. Deriving the distance h_s from the splitting point Q to the line ℓ_{left}.

we have

$$h_s = h - \frac{\sqrt{h^2 + k^2 d^2} + \sqrt{h^2 + (k-1)^2 d^2}}{d} \tag{3.6}$$

From (3.1) and (3.6) we have

$$h_s = h - \frac{1}{2}\left(\sqrt{h^2 - 4(N-2)^2 + 4k^2} + \sqrt{h^2 - 4(N-2)^2 + 4(k-1)^2}\right) \tag{3.7}$$

Since $k \le N-1$, expressing the limit for h_s in (3.7) when $h \to \infty$ implies the final result, i.e.,

$$\lim_{h\to\infty} h_s = 0.$$

□

In other words, for h large enough, all light corridors split on distance ε from the line ℓ_{left} and "enter" the area in which they do not intersect.

From Equation (3.2), to determine the width w of the area where the light corridors do not intersect, besides the distance $\bar{h}_s$, we also need to determine the distance $\bar{h}_m$, i.e. the distance from the closest meeting point(s) to the line ℓ_{left}. We first determine the light corridors that define the closest meeting point(s).

One can show that the light corridors that define the closest meeting point(s) begin between two neighboring pairs of disks; see Figure 3.14. More precisely, the bottom-most corridor C_b of all corridors beginning between the pair of disks (δ_{j+1}, δ_j) and the top-most corridor C_t of all corridors beginning between the pair of disks (δ_j, δ_{j-1}) define (one of) the closest meeting point(s) to the line ℓ_{left}. Let h_m be the distance from the splitting point Q' of the corridors C_b and C_t to the line ℓ_{left}. Note that C_b ends between the bottom pair of disks (δ'_1, δ'_2) and C_t ends between the top pair of disks $(\delta'_N, \delta'_{N-1})$ in the right column. In a similar way as Lemma 3.7, using elementary calculus, one can prove the following lemma.

Lemma 3.8. *For the distance h_m, it holds that $h_m \to \infty$, when $h \to \infty$.* □

From Lemma 3.7 and Lemma 3.8 and Equation (3.2), we can conclude that for h large enough, the width w of the area where corridors do not intersect can be of size $N \cdot d$. Note that the area is not in the middle between the columns. Instead, we have two such areas of non-intersecting corridors adjacent to the left and to the right column, respectively.

In the next step of the construction, we add $2N$ disks organized in two rows that are on the top and the bottom side, as we mentioned earlier in this section, and such that the areas of non-intersecting corridors completely overlap. Each of the $\mathcal{O}(N^2)$ light corridors in the left to right direction intersects each of the $\mathcal{O}(N^2)$ light corridors in the top to bottom direction. Hence, they partition the square area of size w^2 into $\Theta(N^4)$ regions. In order for these regions to be the *shadow* regions, the light coming from directions different than left, right, top or bottom must be blocked. Therefore, in addition to the $4N$ disks used in this construction, we "close the gaps" by extending, for example, the top row and the left column by $\lceil \frac{N}{2} \rceil$ tangent disks each and the right column and the bottom row with $2N$ tangent disks each; see Figure 3.16. These blocking disks ensure that there are no additional corridors intersecting the area partitioned into shadow regions by the constructed light corridors. Hence, we proved that the number of shadow regions defined by N unit disks is $\mathcal{O}(N^4)$ and that this bound is tight.

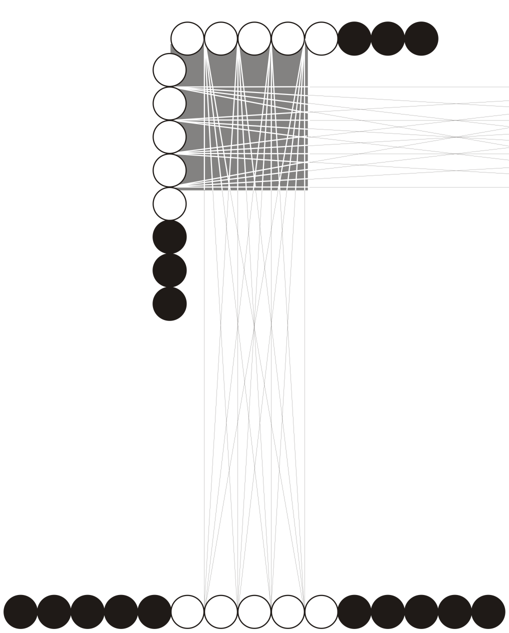

Figure 3.16. A linear number of unit disks defining $\Theta(N^4)$ shadow regions - the thin light corridors pass between the white disks; the black disks are mutually tangent, hence, representing the blocking disks.

3.5 Discussion

In Chapter 2 we described two algorithms for detecting objects placed in a rectangular detection area. Both algorithms use as an input the blocking matrix, which contains the information on blocked and non-blocked lines of sight for all emitter-sensor pairs. The output is a set of convex polygons that contains all of the objects. As we explained in Section 2.1 and 2.3, the algorithms only detect objects that are large enough to contain a circle of radius r, where r is the distance between an emitter and its neighboring sensor. This restriction on the minimum size of the objects follows from the definition of a surely detectable object (see Section 2.1). In a sense, there is a trade-off between enabling detection of objects of all sizes with no guarantees on detection of small objects, and enabling a guaranteed detection of objects larger than some minimum size given. However, we emphasize that this restriction results in running times of the detection algorithms being significantly improved. The cut-off algorithm discards the shadow wedge intersections that are not sufficiently large to contain one object. This procedure is a very important step in the early stages, when most of the non-object intersections are discarded. Keeping these intersections as potential objects implies usually a large number of further intersecting operations that costs extra computation time. The actual time complexity of the point-by-point algorithm strongly depends on the number of objects that need to be detected, and finding a tight upper bound on that number would not have been possible without a restriction on the minimum size of the objects. Hence, both algorithms enable detection of all objects larger than a circle of radius r, irrespective of their position and shape, by providing an approximation on their geometry.

The measure of the accuracy of object detection can be simply defined as a ratio between the total area occupied by the objects and the total area reported as a result of detection. In a broader sense, the accuracy also includes the error in the number of objects detected, as it is defined in [Jovanović, 2007]. More precisely, the occurrences of the visibility problems, such as reporting multiple objects as one and reporting non-existing objects, are considered as detection failures, and each detection failure is penalized by decreasing the measure of the detection accuracy.

The shadow regions algorithm presented in this chapter provides the output of a detection algorithm with the maximum accuracy of detecting objects that can be achieved. In other words, the output of the shadow region algorithm is identical to the output of a detection algorithm, when detecting circular objects of equal size using infinitely many emitters and infinitely many sensors positioned in a frame surrounding the objects.

Figures 3.17 and 3.18 illustrate the difference in detection accuracy for 14 circular objects with 40 emitters and 40 sensors in one case, and with infinitely many emitters and sensors in the other case. The latter represents the output of the shadow regions algorithm (see Section 3.3). Clearly, the precision of the detection with a small number of emitters and sensors is much lower than the precision of the detection in the ideal environment. Note that the three free shadow regions near the bottom right corner in Figure 3.18 are included into three separate detected objects in Figure 3.18. These free shadow regions are smaller than the objects, hence, using any of the detection algorithms, they would have been discarded. Consequently, the accuracy of the detection would have been increased. However, that is possible only if the additional shadow area is detached from the shadow regions, i.e., the n-wedge intersections, that contain the objects. The only way to achieve that is by increasing the density of the emitters and sensors in the rectangular frame.

Based on experimental results [Jovanović, 2007], the objects may be positioned in

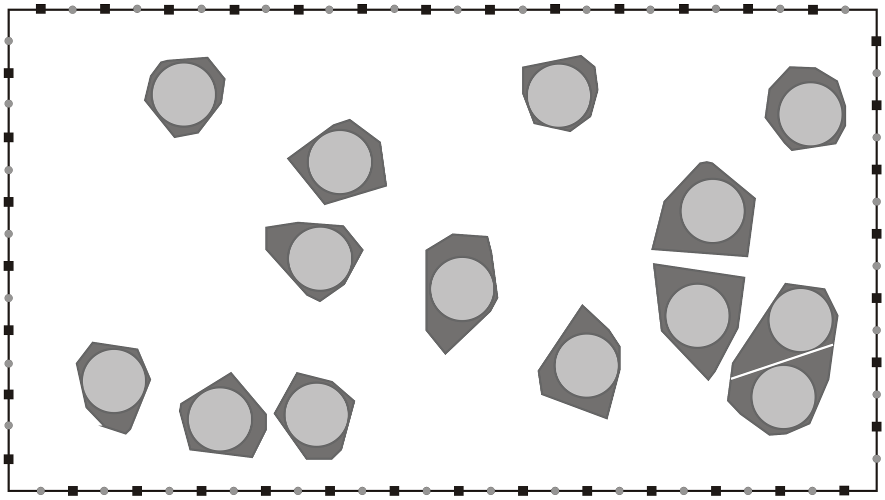

Figure 3.17. Output of the cut-off algorithm and the point-by-point algorithm: detecting 14 circular objects using 40 emitters and 40 sensors placed in a rectangular frame.

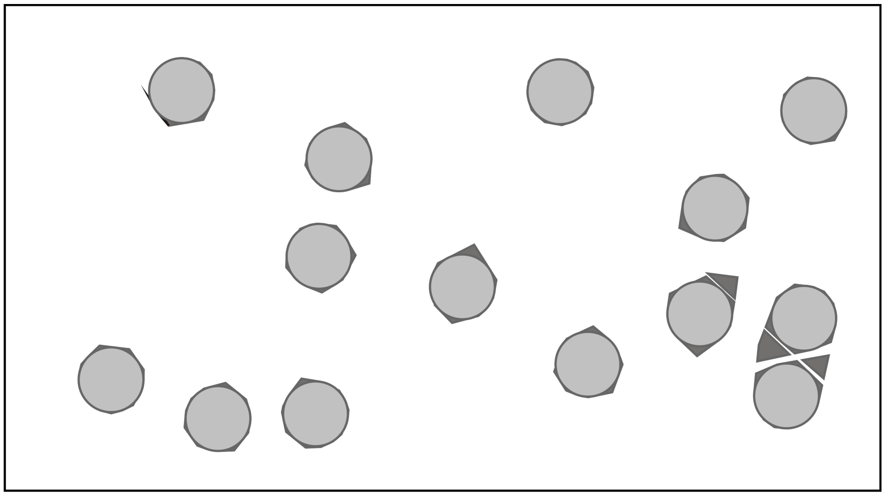

Figure 3.18. Output of the shadow regions algorithm: detecting 14 circular objects using infinitely many emitters and infinitely many sensors placed in a rectangular frame.

such a way that they block all the light crossing an area larger than the minimum object size. Such an area is by definition an n-wedge intersection, and because of its size, it is reported in the output of a detection algorithm. This corresponds to the situation where the disks define a free shadow region larger than a disk, in the ideal environment defined in this chapter. As we indicated in Section 3.1, the size of a shadow region depends on the distance between the disks defining it. In Chapter 5 we will discuss the relation between the number of disks that define such a free shadow region and the distance between the disks, and we will show that a set of disks can ”hide” one disk even if the distance between each two disks tends to infinity.

4

Visibility problems I: Dark and shadow points among objects

The accuracy of object detection based on the technology that uses emitters and sensors, depends on many parameters. In Chapter 3, we considered detecting circular objects assuming infinitely many emitters and sensors in a frame surrounding the objects. As a consequence of such an assumption, each line in the plane can be considered as a line of sight between one emitter and one sensor, which results in the maximum level of accuracy that can be achieved for a given placement of objects in the detection area. However, the detection is also heavily influenced by the objects' relative positions. One observation is that the size of the shadow regions is closely related to the distance between the objects. More precisely, the objects that are relatively close to each other define larger shadow regions than the objects that are relatively far from each other. Besides decreasing the accuracy of the detection, the objects that are close to each other can cause more serious problems in the process of object detection. A large free shadow region, i.e., a shadow region that does not contain any object, may be interpreted by the detection device as an object in the detection area. Furthermore, two or more objects may be detected as one, and generally, this issue cannot be resolved easily when there is no additional information on the number and/or the size of the objects.

Ideally, the detection process results in reporting the geometry of the objects, exactly as they are. However, the objects define additional shadow points that decrease the accuracy of detection. In this chapter, we explore the relation between the occurrence of ”free” shadow points and the distance between the objects.

As in the previous chapter, we restrict ourself to circular objects having identical size. We also restrict the minimum distance between each pair of objects to be some given distance d. The reason for these restrictions comes from the application domain. Many board games use pawns that are circular on the bottom (e.g., chess) and both the pawns and the board with the playing fields can be designed so that the pawns cannot be on a distance less than some predefined distance. In order to avoid the visibility problems caused by the occurrences of free shadow regions, one needs to find the relation between the number and the positions of the objects and the minimum mutual distance given. However, the problem of finding this relation is not a simple or a precisely defined problem, but rather a complex collection of many related challenging problems. We start the investigation by considering the problem of determining the minimum number of objects that can define one free shadow point, when the distance between any two objects is at least d.

Overview. The rest of this chapter is organized as follows. In Section 4.1 we define two types of blocking sets of disks, namely, sets of disks that block all the rays emanating from a given point, in which case the point is called dark, or sets of disks that block all the lines passing through a point, in which case the point is a shadow point, as defined in Chapter 3. Then we formally define two separate problems of determining minimum blocking sets depending on a given minimum mutual distance d between the disks. We refer to the problems as *the hidden point problem* and *the occluded point problem*. In Section 4.2 and Section 4.3 we propose the solutions to the hidden point problem and the occluded point problem, respectively, by presenting the asymptotic bounds on the minimum cardinality of blocking sets of disks when d tends to infinity.

4.1 Dark and shadow points and blocking sets of disks

In the previous chapter we defined a shadow point as a point for which all lines containing it are blocked by one or more non-overlapping unit disks. Here, we introduce another class of points that may occur within a set of disks, i.e., the dark points.

Definition 4.1 (blocked ray). A ray ρ in the plane is said to be blocked by a disk δ, if ρ and δ have a non-empty intersection. □

Definition 4.2 (dark point). A point q is a dark point if all rays emanating from q are blocked by a set of disks. □

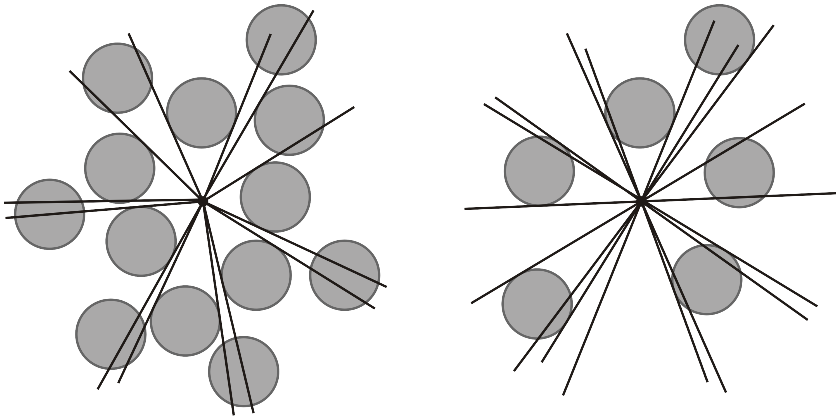

Figure 4.1. Left: an example of a dark point and the corresponding blocking set of disks; right: an example of a shadow point and the corresponding blocking set of disks.

For convenience, we here restate the definition of a shadow point.

Definition 4.3 (shadow point). A point q is called a shadow point if all lines containing q are blocked by a set of disks. □

By definition, any dark point is a shadow point. Hence, the set of all shadow points, i.e., the set $\mathcal{S}$ of all shadow regions defined in Chapter 3, is a superset for the set of all dark points defined by a set of disks in the plane. As in the previous chapter, we assume closed and non-overlapping disks of unit radius.

Let p be a point in the two-dimensional plane. Let $\mathcal{R}_p$ denote the set of all rays that emanate from p and let $\mathcal{L}_p$ denote the set of all lines that contain p.

Definition 4.4 (blocking set). A set $\mathcal{D}$ of non-overlapping unit disks is called a blocking set for the set of rays $\mathcal{R}$ if every ray $\rho \in \mathcal{R}$ is blocked by a disk in $\mathcal{D}$.
A set $\mathcal{D}$ of non-overlapping unit disks is called a blocking set for the set of lines $\mathcal{L}$ if every line $\ell \in \mathcal{L}$ is blocked by a disk in $\mathcal{D}$. □

Hence, if for a point p there exists a blocking set $\mathcal{D}$ for all rays in $\mathcal{R}_p$, then p is a dark point; see Figure 4.1-left. In the same way, if $\mathcal{D}$ is a blocking set for the set of lines $\mathcal{L}_p$, then point p is a shadow point; see Figure 4.1-right.

Definition 4.5 (d-apart blocking set). A blocking set $\mathcal{D}$ of disks is called d-apart if the distance between each pair of disks in $\mathcal{D}$ as well as the distance between each disk and the given point p is at least d. □

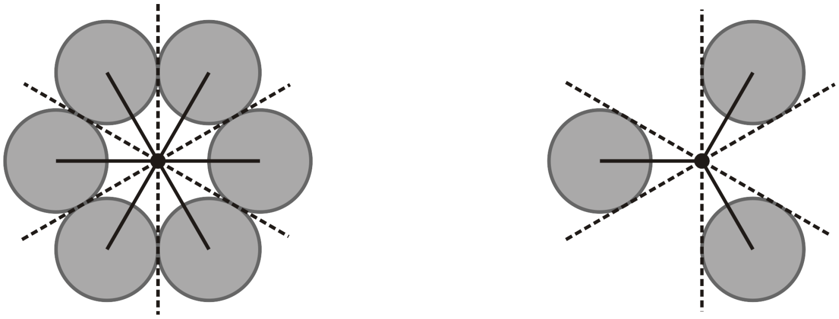

Figure 4.2. Minimum blocking sets for the set of rays - left, and the set of lines - right, for $d = 2$.

Distances are measured from center to center. Hence, the minimum distance between two disks is 2, in the case of two mutually tangent disks.

In the rest of the chapter, we focus our attention on two problems: (*i*) the hidden point problem and (*ii*) the occluded point problem. The essence of both problems is determining d-apart blocking sets of minimum cardinality, for some given distance d, where in the hidden point problem we consider blocking sets of disks for the set of rays $\mathcal{R}_p$ and in the occluded point problem we consider blocking sets for the set of lines $\mathcal{L}_p$.

Figure 4.2 illustrates the solutions of these problems for $d = 2$. The disks of the blocking sets form a (partial) hexagonal lattice. These solutions follow directly from the well-known results in the theory of plane and circle packing [Melissen, 1997; Conway et al., 1999], where it is proved that the hexagonal lattice is the densest of all possible plane packings [Tóth, 1954].

The question of the existence of dark points among disks was posed in [Mitchell, 2007]. In [Dumitrescu & Jiang, 2010], the authors show that for large enough radius of a circle packed by unit disks, there are dark points on the boundary of the disks. Furthermore, they propose an algorithmic solution to the problem of finding a boundary illumination map, i.e., the problem of finding all dark points that are on the boundary of disks in a given set of disks. Hence, these authors only consider dark points on the boundary of the disks.

4.2 The hidden point problem

In this section, we discuss the problem of blocking all rays $\mathcal{R}_p$ emanating from a given point p. Formally, we define the problem as follows.

The hidden point problem. *Given a minimum mutual distance d, what is the minimum cardinality N_d of a d-apart blocking set for the set $\mathcal{R}_p$ of rays?*

More specifically, we are interested in the asymptotic behavior of N_d, as d tends to infinity. For reasons of convenience, we will assume d to be a positive integer at least 2.

As the main result, we present the upper and lower bounds on the minimum cardinality N_d of a blocking set for $\mathcal{R}_p$, for $d \to \infty$. The upper bounds are derived by considering blocking sets for which point p as well as the disks in $\mathcal{D}$ are positioned on grid points of a regular triangular grid. In this way, the disks are grouped into successive regular hexagons with sides of length d, $2d$, etc., all centered at p. Using a slightly simplified model of the disks, we prove that we need *exactly* $d-1$ hexagons to block all rays, by establishing an interesting link between blocking disks and Farey sequences. Farey sequences have been studied extensively in number theory, already in the 19-th century; see e.g., [Hardy & Wright, 1979]. They have some amazing properties that we use to prove our results.

The lower bounds on N_d are derived by counting the number of circles that can be packed in regular hexagonal rings, such that the distance between them is at least d. In this way, we establish the lower bound that is quadratic in d.

Combining the results on upper and lower bounds, we prove that the minimum cardinality N_d of a d-apart blocking set for the set of rays $\mathcal{R}_p$ is quadratic in d. More precisely, we show that

$$\frac{\pi^2}{16} \leq \lim_{d\to\infty} \frac{N_d}{d^2} \leq \frac{18}{\pi^2}. \tag{4.1}$$

4.2.1 Determining Upper Bounds

In this section we show that $N_d = \mathcal{O}(d^2)$. We present blocking sets for which point p as well as the centers of the disks are positioned on the grid points of a regular triangular grid; see Figure 4.3. The centers of the 6 disks that are at a distance d from p form a regular hexagon H_1, with sides of length d. The centers of the disks that are at a distance d from the disks in H_1 (and outside the hexagon) form a regular hexagon H_2 with sides of length $2d$, etc. First, we show that $d-1$ hexagons are sufficient to block all rays. Then, we give a tight upper bound on the number of disks that are positioned in these $d-1$ hexagons.

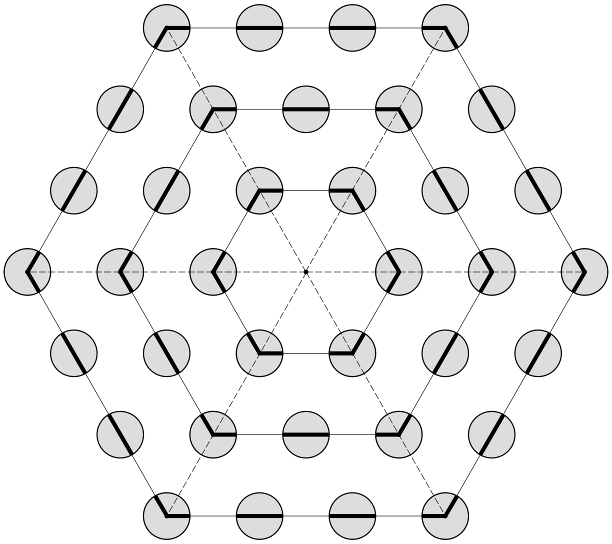

Figure 4.3. Intersection of disks and hexagons.

Bounding the number of hexagons

To prove that we need *at most* $d-1$ hexagons, we make the following simplification. Instead of determining which rays are blocked by the disks, we determine which rays are blocked by the line segments that are defined as the intersection of the disks with the hexagons; see bold lines in Figure 4.3. Note that a ray that is blocked by such a line segment is also blocked by the corresponding disk. Hence, the number of hexagons that we need to block all rays by the line segments is an upper bound on the number of hexagons that we need to block all rays by the disks.

We next prove that the number of hexagons that are needed to block all rays by the line segments is *exactly* $d-1$. To do so, we project all line segments in hexagons $H_1, H_2, \ldots, H_{d-1}$ from center point p onto the sides of hexagon H_d. To easily distinguish line segments before and after projection, we will call the latter intervals. For symmetry reasons, we can concentrate on one of the six sections of hexagon H_d; say the upper middle section in Figure 4.3. This section is an equilateral triangle, with sides of length d^2 and with p as one of its vertices. Before projecting the line segments onto the upper side of this triangle (i.e., the side opposite to p), we normalize the triangle by dividing all lengths by d^2. Hence, we reduce the sides of the triangle to 1. We associate the interval $[0,1]$ with the upper side of the triangle. This interval must be completely covered by the projected intervals. As illustrated in Figure 4.4, the i-th line segment on hexagon H_j, having a length $2/d^2$,

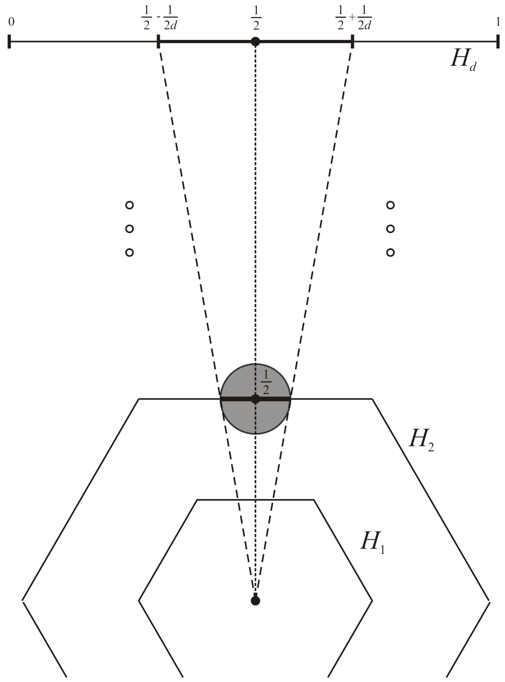

Figure 4.4. Projecting a line segment of length $\frac{2}{d^2}$ from the hexagon H_2 onto the side of the hexagon H_d to an interval of length $\frac{1}{d}$.

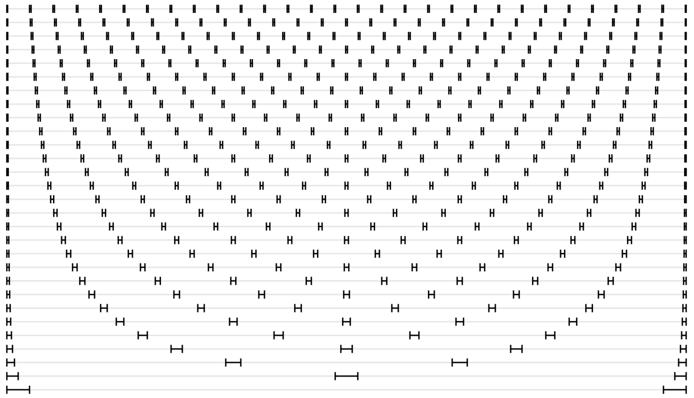

Figure 4.5. The intervals associated with the successive hexagons, shown as part of the interval $[0,1]$. The bottom row gives the two intervals related to 0/1 and 1/1. The second row gives the intervals related to 0/2, 1/2, and 2/2, etc.

is projected onto the upper side of the triangle to an interval of length

$$\frac{d}{j} \cdot \frac{2}{d^2} = \frac{2}{jd}.$$

The midpoint of the original line segment (i.e., the center of the disk) will be projected on point i/j in the interval $[0,1]$; see Figure 4.6. Hence, the resulting interval, after projection, can be expressed as

$$\left[\frac{i}{j} - \frac{1}{jd}, \frac{i}{j} + \frac{1}{jd}\right], \tag{4.2}$$

potentially truncated when it falls outside the interval $[0,1]$. Figure 4.5 shows the intervals associated with the successive hexagons for $d = 30$.

We next show that an interval is completely contained in another if the corresponding fraction i/j is reducible. For example, the intervals related to $2/4, 3/6, \ldots$ are completely contained in the interval related to $1/2$.

Lemma 4.1. *If a given fraction i/j can be reduced to a fraction i'/j' then any ray blocked by the* $\left[\frac{i}{j} - \frac{1}{jd}, \frac{i}{j} + \frac{1}{jd}\right]$ *will also be blocked by* $\left[\frac{i'}{j'} - \frac{1}{j'd}, \frac{i'}{j'} + \frac{1}{j'd}\right]$.

Proof. Using that $i'/j' = i/j$ and $0 < j' < j$, it is easily shown that

$$\frac{i'}{j'} - \frac{1}{j'd} < \frac{i}{j} - \frac{1}{jd} < \frac{i}{j} + \frac{1}{jd} < \frac{i'}{j'} + \frac{1}{j'd}.$$

which implies the result. □

In other words, a line segment associated with the reducible fraction is obsolete,

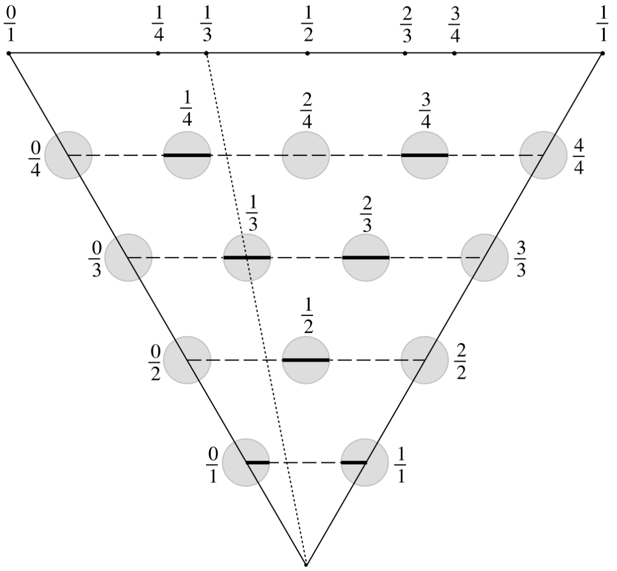

Figure 4.6. With each disk we can associate a fraction i/j where i and j are integers satisfying $0 \leq i \leq j$ and $1 \leq j$. The center of the disk is projected to point i/j in the interval $[0,1]$.

because it only blocks the rays that are already blocked by the line segment associated with the corresponding irreducible fraction. Hence, we can restrict ourselves to intervals for which the midpoints i/j are irreducible. To prove that all the remaining intervals cover the complete interval $[0,1]$, we use some well-known results from number theory. Let us first introduce the concept of Farey sequence from this domain.

Definition 4.6 (Farey sequence). The Farey sequence $\mathcal{F}_n$ is the ordered sequence of all non-negative irreducible fractions $\frac{a}{b}$ for which $1 \leq b \leq n$ and $0 \leq a \leq b$. □

For example, the Farey sequence $\mathcal{F}_5$ is given by

$$\frac{0}{1}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \frac{1}{1}.$$

These Farey sequences have been extensively studied, already in the 19-th century. We need the following two well-known properties.

Lemma 4.2. *Let $\frac{a}{b}$ and $\frac{a'}{b'}$ be two successive terms of a Farey sequence $\mathcal{F}_n$, with $\frac{a}{b} < \frac{a'}{b'}$. Then the following properties hold*

$$a'b - ab' = 1 \tag{4.3}$$

and

$$b + b' > n. \tag{4.4}$$

□

Proofs of these properties can be found in e.g. [Hardy & Wright, 1979]. For the original sources, we refer to pointers in [Dickson, 2005]. In addition, for each $n \geq 2$, it can be shown that $\mathcal{F}_{n+1}$ can be constructed from $\mathcal{F}_n$ by simply inserting the term

$$\frac{a+a'}{b+b'}$$

between two successive terms a/b and a'/b' for which $b + b' = n + 1$. It is quite easy to prove that

$$\frac{a}{b} < \frac{a+a'}{b+b'} < \frac{a'}{b'}$$

and that this new term is irreducible. Hence, this results in a very efficient way to generate Farey sequences, since we only add terms that are irreducible and these are directly added at their correct position in the sequence.

Let $\mathcal{I}_{d-1}$ denote the set of intervals related to $H_1, H_2, \ldots, H_{d-1}$, i.e.,

$$\mathcal{I}_{d-1} = \left\{ \left[\frac{a}{b} - \frac{1}{bd}, \frac{a}{b} + \frac{1}{bd} \right] \right\}$$

where $1 \leq b \leq d-1$, $0 \leq a \leq b$, and $\frac{a}{b}$ is irreducible, and let $\mathcal{I}$ be the sequence of all intervals from $\mathcal{I}_{d-1}$ ordered by increasing midpoint (i.e., by $\frac{a}{b}$). Furthermore, let the midpoint of an interval $I \in \mathcal{I}_{d-1}$ be denoted by $m(I)$. By definition of a Farey sequence, the midpoints of the intervals in $\mathcal{I}$ are given by the terms in the Farey sequence $\mathcal{F}_{d-1}$. Using the above properties of Farey sequences, we can now prove the following results.

Theorem 4.1. *Two successive intervals I, I' in $\mathcal{I}$ always overlap, i.e.,*

$$I \cap I' \neq \emptyset,$$

but they never cover each others midpoint, i.e.,

$$m(I') \notin I \quad \text{and} \quad m(I) \notin I'.$$

Proof. Let I and I' be two successive intervals in $\mathcal{I}$, such that

$$I = \left[\frac{a}{b} - \frac{1}{bd}, \frac{a}{b} + \frac{1}{bd} \right] \text{ and } I' = \left[\frac{a'}{b'} - \frac{1}{b'd}, \frac{a'}{b'} + \frac{1}{b'd} \right].$$

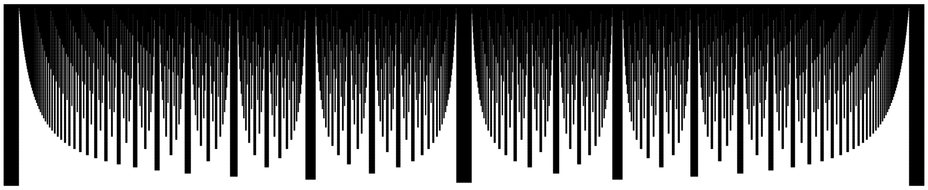

Figure 4.7. Illustration of the merged intervals: each higher row is obtained by including the intervals of the next hexagon.

The distance between the midpoints of I and I' is given by

$$m(I') - m(I) = \frac{a'}{b'} - \frac{a}{b} = \frac{a'b - ab'}{bb'} = \frac{1}{bb'},$$

using (4.3) to derive the last equality. The sum of the right half of I and the left half of I' is given by

$$\frac{1}{bd} + \frac{1}{b'd} = \frac{b'+b}{bb'd} \geq \frac{d}{bb'd} = \frac{1}{bb'},$$

using (4.4) and the fact that b and b' are integer. Hence, I and I' intersect in at least one point.

We next prove that I and I' never cover each others midpoint. We already proved that the distance between the midpoints of I and I' is given by

$$m(I') - m(I) = \frac{1}{bb'}.$$

The size of the right half of I, i.e. the part of I larger than $m(I)$, is given by

$$\frac{1}{bd} < \frac{1}{bb'},$$

since $b' < d$. Likewise, the size of the left half of I' is given by

$$\frac{1}{b'd} < \frac{1}{bb'},$$

since $b < d$. Hence, I and I' do not cover each others midpoint. □

Corollary 4.1. *The union of all intervals in* $\mathcal{I}_{d-1}$ *covers the complete interval* $[0,1]$ *and for each* $I \in \mathcal{I}_{d-1}$ *the union of all intervals in* $\mathcal{I}_{d-1} \backslash I$ *does not cover* $[0,1]$ *completely.*

Proof. Since the leftmost interval covers point 0 and the rightmost interval covers point 1 and since each pair of successive intervals overlap each other, necessarily the complete interval $[0,1]$ is covered. Furthermore, if interval $I \in \mathcal{I}_{d-1}$ is removed, then point $m(I)$ will not be covered by the other intervals. □

In other words, exactly $d-1$ hexagons are required to block all rays. We end this section with the observation that intervals relating to the first $\lfloor d/2 \rfloor$ hexagons never overlap.

Let $\mathcal{I}_{\lfloor d/2 \rfloor} \subseteq \mathcal{I}_{d-1}$ be the subset of intervals relating to the first $\lfloor d/2 \rfloor$ hexagons, more precisely, the hexagons $H_1, H_2, \ldots, H_{\lfloor d/2 \rfloor}$.

Theorem 4.2. *For each pair of intervals $I, I' \in \mathcal{I}_{\lfloor d/2 \rfloor}$ we have that $I \cap I' = \emptyset$.*

Proof. Let the intervals in $\mathcal{I}_{\lfloor d/2 \rfloor}$ be ordered by increasing midpoint. Let a/b and a'/b' be the midpoints of two successive intervals I and I' in this ordering. The fractions a/b and a'/b' are irreducible by definition and they correspond to two successive terms in the Farey sequence $\mathcal{F}_{\lfloor d/2 \rfloor}$. By definition of a Farey sequence,

$$b \leq \lfloor d/2 \rfloor \quad \text{and} \quad b' \leq \lfloor d/2 \rfloor,$$

and furthermore, $b \neq b'$, which implies that

$$b + b' < d. \tag{4.5}$$

To prove that intervals I and I' do not overlap, we need to show that the left end-point of the interval I', i.e., the fraction $\frac{a'}{b'} - \frac{1}{b'd}$ is larger than fraction $\frac{a}{b} + \frac{1}{bd}$ that is the right end-point of the interval I. The difference between the two fractions is given by

$$\begin{aligned} \frac{a'}{b'} - \frac{1}{b'd} - \left(\frac{a}{b} + \frac{1}{bd}\right) &= \frac{a'b - ab'}{bb'} - \frac{b+b'}{bb'd} \\ &= \frac{1}{bb'} - \frac{b+b'}{bb'd} \\ &= \frac{d-(b+b')}{bb'd} \end{aligned} \tag{4.6}$$

using property (4.3) of the Farey sequences to derive the equality in the middle. From (4.5) we have that $d-(b+b') > 0$, which implies that $I \cap I' = \emptyset$. □

The union of the intervals that are obtained by repeatedly adding the intervals of the successive hexagons gives rise to nice self-similar images; see Figure 4.7 for an example with $d = 60$. The figure consists of 59 rows, numbered from bottom to top, where row i gives the intervals in I_i as part of the interval $[0, 1]$.

Bounding the number of disks

To determine the total number of disks that we use in the hexagons, we could make the following straightforward observation. Hexagon H_j contains at most a total of

$6j$ disks. Hence, the total number of disks can be bounded by

$$\sum_{j=1}^{d-1} 6j = 3d^2 - 3d.$$

However, in that case we would also count the disks that correspond to reducible fractions. Leaving out the disks that correspond to reducible fractions, as it is shown for $d = 5$ in Figure 4.8, gives in the following result.

Theorem 4.3. *For the minimum number* N_d *of disks to block all rays in* $\mathcal{R}_p$ *we have*

$$\lim_{d\to\infty} \frac{N_d}{d^2} \leq \frac{18}{\pi^2}.$$

Proof. Theorem 4.1 implies that the midpoint of each interval in $\mathcal{I}_{d-1}$ is not covered by any other interval. This implies that *all* intervals that correspond to irreducible fractions are needed to block all rays. The number of irreducible fractions with a given denominator n is given by Euler's totient function, which is usually denoted by $\varphi(n)$; see e.g. [Conway & Guy, 1995]. Using this result, we can bound the total number of disks that are required to block all rays by

$$6\sum_{i=1}^{d-1} \varphi(i).$$

The sum $\sum_{i=1}^{n} \varphi(i)$ can be bounded by

$$\frac{3}{\pi^2} \cdot n^2 + \mathcal{O}(n \log n),$$

see e.g. [Graham, Knuth & Patashnik, 1994]. This implies the above result. □

Note that we used the simplification to consider only the blocking by line segments instead of the blocking by disks. One can show that the error that we make by this simplification is smallest in the middle, i.e. for disks with i/j close to $1/2$ and increases to both ends to a factor of $2/3\sqrt{3} \approx 1.15$. We also implemented an algorithm where the actual disks are projected on the side of hexagon H_d. Experiments indicate that still exactly $d-1$ hexagons are needed to block all rays. However, in addition to the disks that correspond to reducible fractions we can now discard additional disks in the hexagons H_k with k close to d. Numerical results suggest that, instead of discarding approximately a fraction of $(1 - 6/\pi^2) \approx 0.39$ of the disks, we can now discard approximately a fraction of 0.45. Hence, by avoiding this simplification we cannot gain in number of required hexagons, but we can slightly improve the constant given by Theorem 4.3.

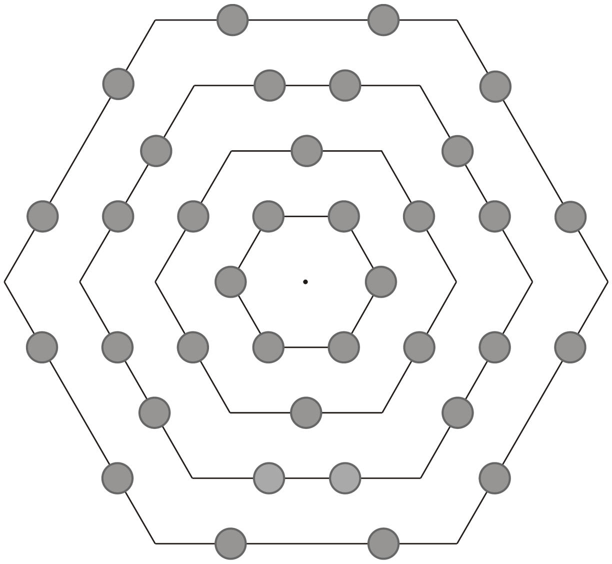

Figure 4.8. A blocking set for the hidden point problem for $d = 5$.

4.2.2 Determining lower bounds

In this section we derive a lower bound on the minimum number N_d of disks required to block all rays in $\mathcal{R}_p$. The primary aim of this section is to derive a lower bound that is also quadratic in d. In addition, we want to derive a tight lower bound on $\lim_{d\to\infty} N_d/d^2$.

For this analysis we partition the plane into hexagonal rings $R_1, R_2, \ldots$, where R_i is the area between hexagons H_i and H_{i+1}, excluding inner hexagon H_i and including outer hexagon H_{i+1} for $i > 1$, and including both inner hexagon H_i and outer hexagon H_{i+1} for $i = 1$. The width of each of the 6 stripes of these rings is $\frac{d}{2}\sqrt{3}$. Ring R_i has inner sides of length id and outer sides of length $(i+1)d$. Figure 4.9 shows ring R_2.

In order to determine a tight lower bound on N_d, we need to determine the maximum number j of hexagonal rings that is necessary to block all rays in $\mathcal{R}_p$. For this reason, we place the maximum number of d-apart disks inside each of the hexagonal rings. A ring R_j is said to contain a disk δ, if the disk's center $O \in R_j$. With the maximum number of disks placed, we determine the number j of rings by assuming that each ray is blocked by exactly one disk and that each of the disks blocks as many rays as one disk in a given ring can block.

We first determine the maximum number of disks that can be placed within the union of j hexagonal rings, such that the disks form a d-apart blocking set.

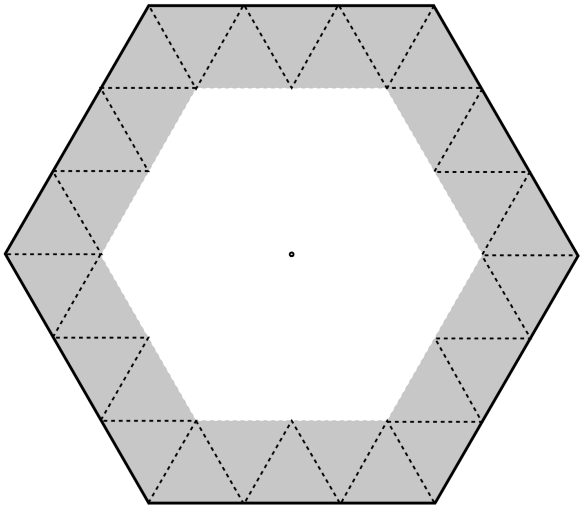

Figure 4.9. Ring R_2 excluding the inner sides of length $2d$, including the outer sides of length $3d$.

Lemma 4.3. *For $j \geq 1$, the number of d-apart disks within the union $\bigcup_{i=1}^{j} R_i$ is at most* $3(j+1)(j+2)$.

Proof. The total number of d-apart disks within $\bigcup_{i=1}^{j} R_i$ is obtained by using an arrangement of the disks as shown in Figure 4.3. Hence we have hexagons $H_1, H_2, \ldots, H_{j+1}$, where H_i contains $6i$ disks, resulting in

$$\sum_{i=1}^{j+1} 6i = 3(j+1)(j+2)$$

disks. The problem is equivalent to packing circles of radius $d/2$ in a regular hexagon of side-length $(j+3/2)d$. Using the result by L. Fejes Toth that can be found in e.g. [Pach & Agarwal, 1995], we conclude that this arrangement is the best possible: in the interior of union $\bigcup_{i=1}^{j} R_i$ we have an optimal packing on a regular triangular grid. In addition, the area of the disks that falls outside hexagon H_{j+1} is maximal. For further details we refer to e.g. [Melissen, 1997], where the closely related circle packing in equilateral triangles is considered. □

Next, we define a collection of rays that one disk blocks.

Definition 4.7 (blocking wedge). A blocking wedge $\beta(\delta)$ of a disk δ is the wedge that contains all rays in $\mathcal{R}_p$ that intersect δ. □

The size of a blocking wedge is given by the angle that the wedge covers. Hence,

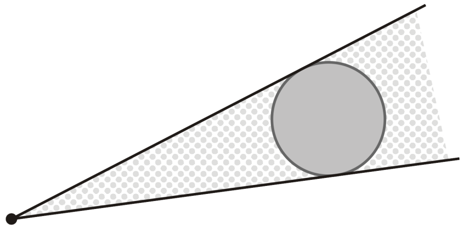

Figure 4.10. An illustration of a blocking wedge.

the size of the blocking wedge of one disk that is on distance d from the point p is given by $2\arcsin\frac{1}{d}$. This leads to the proof of the following lemma (see Figure 4.11-left).

Lemma 4.4. *Let* δ *and* δ' *be two disks on distance* x *and* x' *from the point* p*, respectively, where* $x < x'$*. Then the blocking wedge* $\beta(\delta)$ *is larger than the blocking wedge* $\beta(\delta')$*.* □

From Lemma 4.4 one can easily show the following corollary (see Figure 4.11-middle).

Corollary 4.2. *Let* δ *and* δ' *be two disks such that their centers and the point* p *are collinear, and* δ *is between* p *and* δ'*. Then the blocking wedge* $\beta(\delta)$ *contains the blocking wedge* $\beta(\delta')$*.* □

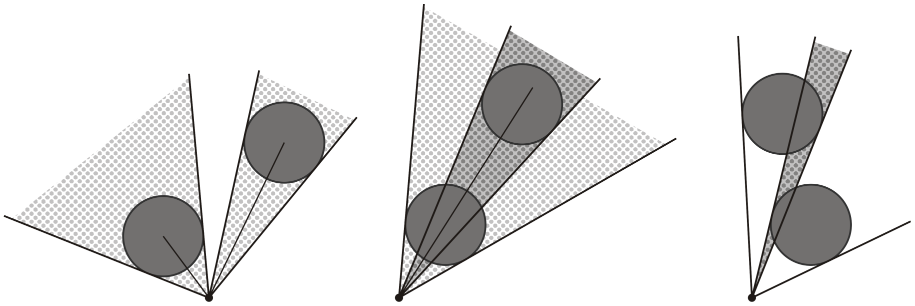

Figure 4.11. An illustration of the size of two blocking wedges depending on the distance of the disks to the central point - left; a blocking wedge of one disk contains an entire blocking wedge of another disk that is further from the central point - middle; the blocking wedges of two disks partly overlap - right.

Note that the blocking wedges of the disks may also partly overlap (see Figure 4.11-right). If the union of the blocking wedges of all of the disks in a given set covers the entire angle 2π, then the disks form a blocking set for the set of rays $\mathcal{R}_p$.

Lemma 4.5. *The blocking wedge* $\beta(\delta)$ *of a disk* δ *in ring* R_j *is of size at most*

$$\frac{4\sqrt{3}}{3jd}+\mathcal{O}\left(\frac{1}{j^3d^3}\right).$$

Proof. From Lemma 4.4, to determine an upper bound on the size of a blocking wedge, we need to determine the minimum distance to the point p on which a disk in the ring R_j can be placed. The inner side of the ring R_j is of size jd. The rings are by definition positioned so that the point p is the center for each of them. Therefore, the minimum distance between the point p and a point in R_j is given by $jd\sqrt{3}/2$. The blocking wedge of a disk positioned on distance x from the point p is given by $2\arcsin\frac{1}{x}$. To bound this expression, we use that

$$\arcsin\frac{1}{x}<\frac{1}{x}+\frac{1}{x^3},$$

for $x>1$. Substituting the distance $jd\sqrt{3}/2$ for x in the inequality above and by multiplying the whole inequality by 2, we get the required result. □

For ease of notation, let

$$\alpha_j=4\frac{\sqrt{3}}{3jd},$$

i.e. α_j denotes the size of the maximum blocking wedge of a disk in ring R_j. In order not to complicate the argumentation, we neglect the $\mathcal{O}(1/(j^3d^3))$ term below. A more precise analysis yields the same final result.

For a set $\mathcal{D}$ of disks, let $\beta(\mathcal{D})$ denote the size of the union of the blocking wedges of all disks in $\mathcal{D}$. Furthermore, let

$$\mu_j=\begin{cases}0, & j=0\\ 3(j+1)(j+2), & \text{if } j\geq 1\end{cases}$$

In other words, μ_j denotes the maximum number of disks that can be placed within the union of j hexagonal rings $R_1,R_2,\dots,R_j$, as it is shown in Lemma 4.3. The following lemma gives an upperbound on $\beta(\mathcal{D})$ for a set $\mathcal{D}$ of d-apart disks.

Lemma 4.6. *Let* $\mathcal{D}$ *be a set of* N d*-apart disks. Then for all* $j\geq 1$ *such that* $N\geq\mu_j$ *it holds that*

$$\beta(\mathcal{D})\leq\sum_{i=1}^{j}(\mu_i-\mu_{i-1})\alpha_i+(N-\mu_j)\alpha_{j+1}.$$

Proof. Let $j\geq 1$. We denote the disks in $\mathcal{D}$ by $\delta_1,\dots,\delta_N$, and assume that they are ordered with increasing ring number. Note that α_k is decreasing in k, as it follows from Lemma 4.4. The size of the blocking wedge of any disk is thus at most α_1.

As a consequence, $\beta(\{\delta_1,\ldots,\delta_{\mu_1}\}) \leq \mu_1\alpha_1$.

Now, let $2 \leq i \leq j$. A disk δ_k with $k > \mu_{i-1}$ cannot reside in one of the rings $R_1, R_2, \ldots, R_{i-1}$, and hence, its blocking wedge is of size at most α_i. The number of disks residing in ring R_i is by definition $\mu_i - \mu_{i-1}$. The number of disks residing outside of the union of j hexagonal rings $R_1, R_2, \ldots, R_j$ is $N - \mu_j$, as a consequence of Lemma 4.3. Hence, the blocking wedges of the disks δ_k with $\mu_{i-1} < k \leq \mu_i$ are jointly of size at most equal to $(\mu_i - \mu_{i-1})\alpha_i$, and the blocking wedges of disks δ_k with $\mu_j < k \leq N$ are jointly of size at most equal to $(N - \mu_j)\alpha_{j+1}$. □

Using this lemma, we can now prove the theorem that we use to determine the number j of hexagonal rings that is necessary to block all rays in $\mathcal{R}_p$.

Theorem 4.4. *Let $\mathcal{D}$ be a d-apart blocking set containing N disks. Furthermore, let the number j of hexagonal rings be such that $\mu_j \leq N < \mu_{j+1}$. Then it holds that*

$$3 + \sum_{i=2}^{j+1} \left(1 + \frac{1}{i}\right) \geq \frac{\pi}{4\sqrt{3}} d.$$

Proof. As $\mathcal{D}$ is a d-apart blocking set, $\beta(\mathcal{D}) = 2\pi$. Using Lemma 4.6, we obtain

$$2\pi = \beta(\mathcal{D}) \leq \sum_{i=1}^{j} (\mu_i - \mu_{i-1})\alpha_i + (N - \mu_j)\alpha_{j+1}. \tag{4.7}$$

From $N < \mu_{j+1}$ and $\alpha_{j+1} > 0$, we have that

$$(N - \mu_j)\alpha_{j+1} < (\mu_{j+1} - \mu_j)\alpha_{j+1}. \tag{4.8}$$

From (4.7) and (4.8) we then have

$$2\pi \leq \sum_{i=1}^{j+1} (\mu_i - \mu_{i-1})\alpha_i. \tag{4.9}$$

Substituting the explicit expressions for μ_i and α_i in (4.9) and splitting off the $i = 1$ term, we obtain

$$\begin{array}{rcl}
2\pi & \leq & (\mu_1 - \mu_0)\alpha_1 + \sum_{i=2}^{j+1} (\mu_i - \mu_{i-1})\alpha_i \\
\Leftrightarrow \; 2\pi & \leq & 18 \cdot \frac{4\sqrt{3}}{3d} + \sum_{i=2}^{j+1} (3(i+1)(i+2) - 3i(i+1)) \frac{4\sqrt{3}}{3id} \\
\Leftrightarrow \; \frac{\pi}{2\sqrt{3}} & \leq & \frac{6}{d} + \sum_{i=2}^{j+1} (i+1)(i+2-i) \frac{1}{id} \\
\Leftrightarrow \; \frac{\pi}{4\sqrt{3}} d & \leq & 3 + \sum_{i=2}^{j+1} (1 + \frac{1}{i})
\end{array}$$

□

The size of a d-apart blocking set thus satisfies

$$N_d \geq \mu_j = 3(j+1)(j+2),$$

where j is such that

$$3 + \sum_{i=2}^{j+1} (1 + \frac{1}{i}) \geq \frac{\pi}{4\sqrt{3}} d.$$

Using

$$3 + \sum_{i=2}^{j+1} (1 + \frac{1}{i}) = j + \mathcal{O}(\log j)$$

and $\mu_j = 3j^2 + \mathcal{O}(j)$, we finally determined a lower bound on the minimum number N_d of disks needed to block all rays in $\mathcal{R}_p$.

Theorem 4.5. *For the minimum number N_d of disks needed to block all rays in $\mathcal{R}_p$ we have that*

$$\lim_{d \to \infty} \frac{N_d}{d^2} \geq \frac{\pi^2}{16}.$$

□

4.3 The occluded point problem

In this section, we discuss the problem of determining the minimum number N_d of disks that form a d-apart blocking set for the set $\mathcal{L}_p$ of lines containing a given point p. As in the previous section, we focus on the asymptotic behavior of N_d, as d tends to infinity. Formally, the problem is defined as follows.

The occluded point problem. *Given a minimum mutual distance d, what is the minimum cardinality N_d of a d-apart blocking set for the set $\mathcal{L}_p$ of lines?*

Upper and lower bounds on the minimum number N_d of disks needed to block all the lines from $\mathcal{L}_p$ can be straightforwardly derived from the bounds of the hidden point problem. In more detail, each line that contains point p corresponds to two different rays emanating from p. A pair of rays in $\mathcal{R}_p$ that corresponds to one line in $\mathcal{L}_p$ is blocked by two diametrically opposite disks in the solution of the hidden point problem. Clearly, to block the corresponding line in $\mathcal{L}_p$ only one of the diametrically opposite disks is needed. Hence, the number of disks needed to block all the lines in $\mathcal{L}_p$ is one half of the number of disks needed to block all the rays in $\mathcal{R}_p$; see Figure 4.12 in comparison with Figure 4.8. Note that in this way, for each blocking set for $\mathcal{R}_p$ that has λ pairs of diametrically opposite disks we have 2^λ different blocking sets for $\mathcal{L}_p$ since for each of the λ pairs we can choose any of the two disks.

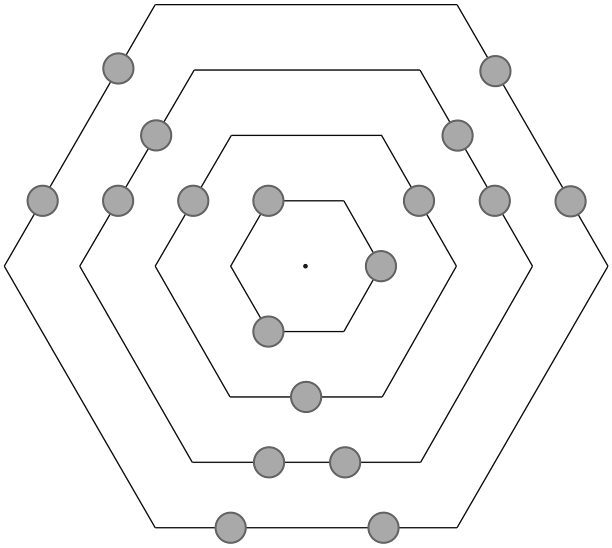

Figure 4.12. An example of a blocking set for the occluded point problem for $d = 5$.

The lower bound for the occluded point problem is derived in a similar way as the lower bound for the hidden point problem. We place the maximum number of disks in one half-plane defined by a line containing point p. By maximizing the size of the blocking wedge of each disk, we determine the number of half-rings needed such that the blocking wedges of all of the disks sum up to $\pi/2$. We assume then that it is possible to place again the maximum number of disks in the other half-plane so that these disks block the complementary wedges that are joined also equal to $\pi/2$. In this way, the blocking wedges of the disks in both half-planes completely cover the angle π. Note that for the occluded point problem, we do not need the blocking wedges to jointly cover the entire angle 2π, which is again a consequence of blocking lines instead of rays.

The upper and lower bound for the occluded point problem are given by the following corollary.

Corollary 4.3. *For the minimum number N_d of disks needed to block all lines in $\mathcal{L}_p$ we have that*

$$\frac{\pi^2}{64} \leq \lim_{d\to\infty} \frac{N_d}{d^2} \leq \frac{9}{\pi^2}.$$

Proof. The upper bound is straightforwardly derived from the result of Theorem 4.3. The lower bound is derived as explained above, by substituting the values

$$\begin{aligned} \beta(\mathcal{D}) &= \frac{\pi}{2}, \\ \mu_i &= \frac{3}{2}(i+1)(i+2), \\ \alpha_i &= \frac{4\sqrt{3}}{3id} \end{aligned}$$

in Lemma 4.6. □

We expect that the tightness of the bounds given by Corollary 4.3 can be improved. The following discussion provides details to support this expectation.

The lower bounds given by Theorem 4.5 and Corollary 4.3 for the hidden point problem and the occluded point problem, respectively, can be improved by expressing the exact values of the size of the blocking wedges of each of the disks. In other words, including the size of each of the blocking wedges separately instead of using the upper bound on that size given by Lemma 4.5 results in tighter lower bounds.

Let $\mathcal{D}$ be a d-apart blocking set for the set of lines $\mathcal{L}_p$ that contains N disks and that is constructed by taking one half of the disks of a blocking set for $\mathcal{R}_p$, as we described above. Furthermore, let $\mathcal{D}$ be such that the centers of all its disks are in one half-plane defined by a line containing p. With each disk $\delta \in \mathcal{D}$ we associate two centrally symmetric blocking wedges $\beta(\delta)$ and $\beta'(\delta)$ with the center in point p, where $\beta(\delta)$ contains δ. By definition, the union of these pairs of blocking wedges of all of the disks in $\mathcal{D}$ covers the entire angle 2π.

The size of a blocking wedge of one disk depends on the distance between the disk and the point p, as it is shown in Lemma 4.4. Simply said, the disks closer to the point p have larger blocking wedges. Now, let us consider only the blocking wedges $\beta(\delta_1), \beta(\delta_2), \ldots, \beta(\delta_N)$ that contain the disks of $\mathcal{D}$. Among these wedges we can identify the groups of small and partly overlapping blocking wedges that correspond to the disks that are relatively far from the point p. These groups of small overlapping wedges can be covered also by *only one* disk placed closer to the point p in the half-plane that initially does not contain any disk of $\mathcal{D}$. In other words, by using both half-planes when constructing blocking sets, it is possible to cover larger wedges by a smaller number of disks. In this way, the number of disks needed to block all the lines in $\mathcal{L}_p$ can generally be decreased.

Deriving a tighter upper bound using this observation is still a very challenging problem, given the minimum mutual distance between the disks that should be taken care of when placing the disks in the ”empty” half-plane. An illustration

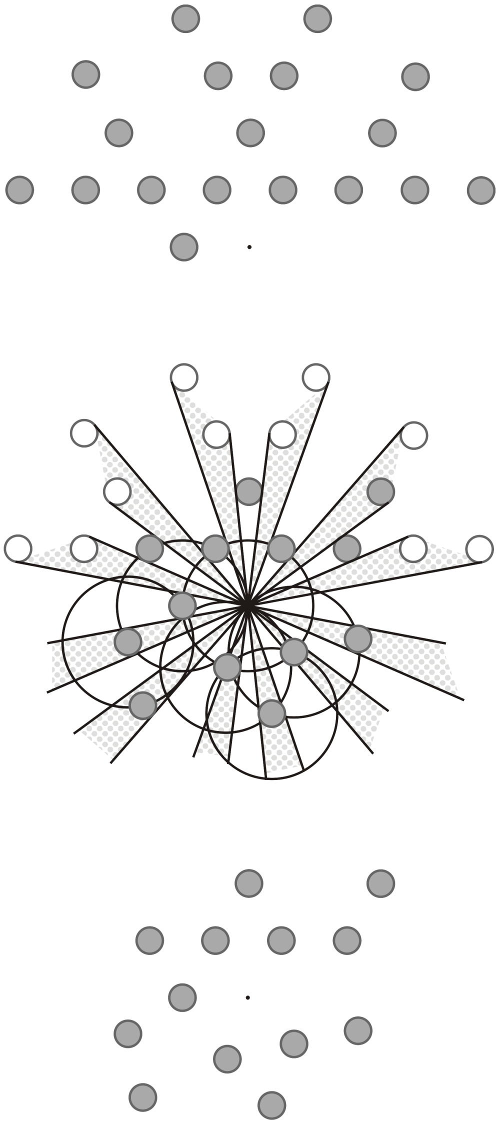

Figure 4.13. A blocking set for $\mathcal{L}_p$ for $d = 5$, straightforwardly derived from the solution of the hidden point problem contains 18 disks. Placing some of the disks closer to the center results in a blocking set containing 13 disks.

of the described procedure of decreasing the number of disks of a blocking set is shown in Figure 4.13. Finding an algorithm that generates the blocking sets of disks where the number of disks is minimized is considered a topic for future research.

5

Visibility problems II: Hiding objects

Detecting objects using light emitters and sensors placed in a frame surrounding the detection area may pose visibility problems occurring during the detection process. As we already explained in Chapter 4, the visibility problems occur for two reasons. First, the problems may be the result of a small number of emitters and sensors in the frame, which consequently leads to a small number of visible lines of sight, i.e., the lines of sight not blocked by the objects. However, we have shown in Chapter 3 that even with infinitely many emitters and sensors, visibility problems still may occur. In other words, the relative positions of the objects in the detection area and especially their mutual distance play a large role in creating visibility problems. Therefore, we once more investigate the occurrence of these problems in a less realistic environment, assuming that each line in the plane is a line of sight connecting one emitter and one sensor. As in Chapter 4, we assume all objects to be unit disks.

Overview. In this chapter we discuss three problems of finding the minimum number of disks needed to block (1) the set of lines intersecting a given disk, (2) the set of rays emanating from a given disk, and (3) the set of lines passing between two given disks. In Section 5.1 we define *the occluded disk problem*, i.e., the problem of falsely detecting a non-existing object. *The hidden disk problem* is the topic of Section 5.2. This problem corresponds to the problem of hiding an object that emits light, where a set of objects blocks all the light emitted by one object. We present solutions by deriving upper and lower bounds on the minimum cardinality of blocking sets. In the same manner, *the merging disks problem*, i.e., the problem of detecting multiple objects as one, is discussed in Section 5.3.

5.1 The occluded disk problem

Let U be a closed unit disk with center p in the two-dimensional plane and let $\mathcal{L}_U$ denote the set of all lines that have a non-empty intersection with U.

Definition 5.1 (occluded disk). A disk U is said to be occluded by a set $\mathcal{D}$ of non-overlapping unit disks if $\mathcal{D}$ is a blocking set for the set of lines $\mathcal{L}_U$. □

A blocking set $\mathcal{D}$ is called d-apart if the distance between each pair of disks in $\mathcal{D} \cup \{U\}$ is at least d, where distances are measured from center to center. We can now define the occluded disk problem, as follows.

The Occluded Disk Problem. Given a unit disk U and a minimum mutual distance d, what is the minimum cardinality N_d of a d-apart blocking set for the set $\mathcal{L}_U$ of lines?

The occluded disk problem corresponds to the problem of falsely detecting a non-existing object. In more detail, this type of problem occurs when multiple objects block all of the lines of sight that intersect some convex area, e.g., a disk. This ”occluded” area is then, by definition, an n-wedge intersection; see Section 2.1. Hence, the occluded area is necessarily reported as an object because a detection algorithm is unable to distinguish it from the ”real” objects.

Let δ and δ' be two disks such that their centers and the center p of the given disk U are collinear, and such that δ is between U and δ'. Using elementary calculus, one can easily prove the following lemma (see Figure 5.1).

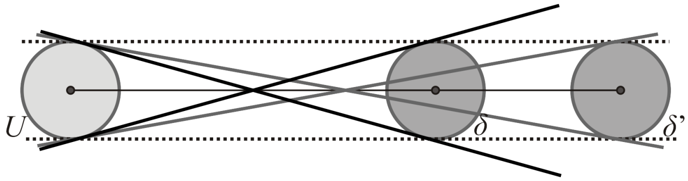

Figure 5.1. Any ray/line blocked by disk δ' is also blocked by disk δ.

Lemma 5.1. *Given three unit disks U, δ and δ' with their centers collinear and such that δ is between U and δ', any line $\ell \in \mathcal{L}_U$ blocked by disk δ' is also blocked by disk δ.* □

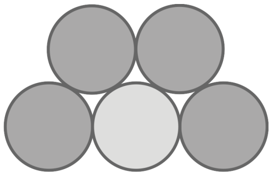

Figure 5.2. Minimum blocking set for $d = 2$.

The occluded disk problem has a simple solution [Stephenson, 2005] of cardinality 4 for $d = 2$; see Figure 5.2. Each line that intersects disk U in the middle, also intersects at least one of the four disks surrounding it.

Another example of a blocking set is shown in Figure 5.3. For $d = 3$, the 9 dark shaded disks positioned at points of a regular triangular grid, as illustrated below, block all the lines that intersect the light disk in the middle.

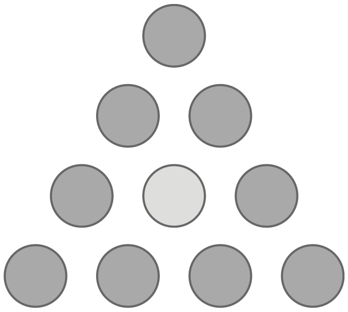

Figure 5.3. Example of a blocking set: 9 disks block all lines that intersect the light disk in the middle.

However, the problem of determining N_d for an arbitrary distance $d > 2$ is difficult, even for small values of d. Therefore, we first derive upper and lower bounds on N_d for $2 \leq d \leq 4$ using two different approaches that we explain in detail in Section 5.1.1 and Section 5.1.2, respectively, and present the results in Section 5.1.3. Then, in Section 5.1.4 we present bounds for $d \to \infty$, which is the result that directly follows from the solution of the occluded point problem.

5.1.1 Deriving upper bounds for small distances between disks

In this section, we construct a special class of blocking sets providing upper bounds on N_d for $2 \leq d \leq 4$. We focus on blocking sets that have an even number k of disks at a distance d from the center of the given disk U, such that they form a regular polygon, with either $k = 4$ or $k = 6$.

Each of these first k disks blocks some lines from the given set $\mathcal{L}_U$. The remaining set of lines, can be divided into disjunct *bundles of lines*. For $k = 4$ and $k = 6$, we obtain 2 and 3 bundles, respectively (see Figure 5.4).

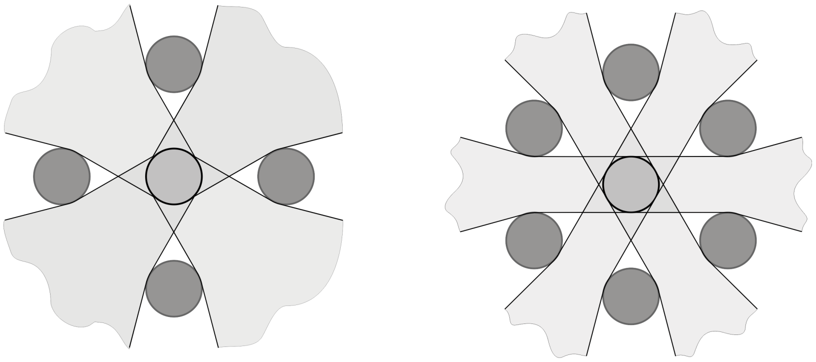

Figure 5.4. The remaining sets of lines grouped as disjunct bundles of lines.

Let $\mathcal{D}'$ be an arbitrary set of non-overlapping unit disks such that $\mathcal{D}' \cup \{U\}$ is d-apart and let $\mathcal{L}'_U \subset \mathcal{L}_U$ denote the set of all lines in $\mathcal{L}_U$ that are not blocked by the disks of $\mathcal{D}'$.

Definition 5.2 (bundle of lines). A collection of all lines in $\mathcal{L}'_U$ defining the same bipartition of disks in $\mathcal{D}'$ is called a bundle of lines. □

Note the similarity between a bundle of lines and a light corridor defined in Chapter 3. The only difference between the two is that the lines of a bundle contain shadow points by definition, since they intersect disk U, and a light corridor contains only light points.

In the same way as a light corridor, a bundle of lines is characterized by its supporting lines, the clockwise fixed tangent t and the counterclockwise fixed tangent t'. The supporting lines "bound" the bundle of lines, denoted as $\mathcal{L}(t,t')$. The angle between the supporting lines is denoted as θ.

We can now define a subproblem of the occluded disk problem as follows.

Bundle blocking problem. Given a bundle of lines $\mathcal{L}(t,t') \subset \mathcal{L}'_U$, find a blocking set $\mathcal{D}(\mathcal{L}(t,t'))$ of minimum cardinality, such that the blocking set $\mathcal{D}(\mathcal{L}(t,t')) \cup \mathcal{D}' \cup \{U\}$ is d-apart.

Given the restriction on the mutual distance, for each of the disks we define a *boundary circle* that determines the region in which it is not possible to place any additional disks. Therefore, a blocking set for a bundle of lines can be chosen to consist of the disks positioned between the supporting lines and on or outside the boundary circles (see the shaded area in Figure 5.5 as an example).

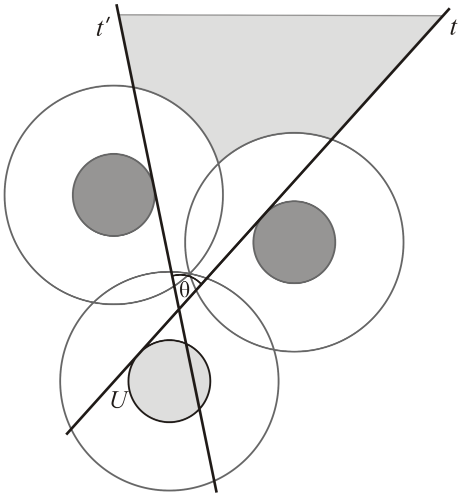

Figure 5.5. An example of a bundle of lines.

Every additional disk that we place in the shaded area reduces the set of lines of the bundle. However, depending on the position of the added disk, the non-blocked lines can all be in one bundle or can be separated into two disjunct bundles. In both cases, the angle(s) between the supporting lines of the new bundle(s) is/are strictly smaller than the angle between the supporting lines before placing the additional disk.

Next we propose a heuristic algorithm that tries to block a given bundle of lines $\mathcal{L}(t,t')$ by 1, 2, 3, 4, or 5 disks. We discuss each of the cases separately.

Blocking a bundle by 1 disk. To test whether or not one disk can block all the lines, we use a simple procedure. Let t and t' be the two supporting lines of $\mathcal{L}(t,t')$ and let θ be the angle between them. Let $\bar{p}$ be the intersection point of the bisector of the angle θ and a boundary circle, such that $\bar{p}$ is not in the interior of any other boundary circle. If the distance between $\bar{p}$ and t is less or equal to 1, it is possible to block the bundle with one disk (see Figure 5.6 - left).

Blocking a bundle by 2 disks. The essential part of the test whether or not two disks can block a given bundle is the observation that the first added disk can be chosen to be tangent to one of the supporting lines. Otherwise, it would separate the bundle into two disjunct bundles, requiring at least two additional disks for the blocking. Therefore, we add one disk such that it is tangent to one of the supporting lines and it is not possible to push the disk closer towards the center without violating the minimum distance requirement. Then, we test whether or not

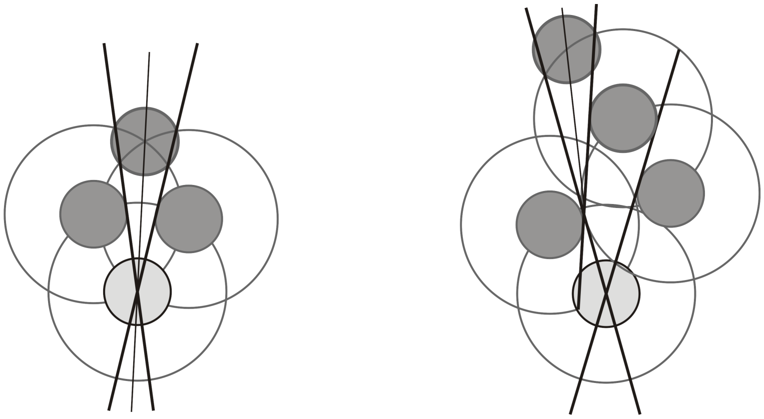

Figure 5.6. Blocking bundles of lines by one disk (left) and two disks (right).

the rest of the lines (the new bundle) can be blocked by one disk (see Figure 5.6 - right).

To test whether a bundle can be blocked by 3 or more disks, we need an additional construction method: find the position of one disk that is closest to U such that one of the new bundles of lines defined by that disk can be blocked by *exactly* one disk. The specific positions of the two disks can be found using analytic geometry and considering different cases.

Different positions of the first added disk result in different bundles of non-blocked lines. Therefore, we consider some cases of that positioning for testing whether the given bundle can be blocked by 3 or more disks.

Blocking a bundle by 3 disks. The analysis is by considering two cases. In the first case, we add the first disk such that it is tangent to one of the supporting lines. Then, the non-blocked lines are in one bundle, and we test whether or not these can be blocked by 2 disks.

In the second case, we first find the position to place the first disk such that one of the two new bundles of lines can be blocked by *exactly* one disk. Then, we check whether the non-blocked bundle of lines can be blocked by 1 disk.

Blocking a bundle by 4 disks. This test consists of checking two cases, as in the test with 3 disks. In the first case, we add one disk such that it is tangent to one of the supporting lines and check whether the new bundle can be blocked by 3 disks.

In the second case, we place the first disk such that one of the two new bundles of lines can be blocked by exactly one disk and check whether the remaining bundle can be blocked by 2 disks.

Blocking a bundle by 5 disks. Besides the two cases similar to those in tests with 3 and 4 disks, we have an additional one in which we place the first disk at the intersection point of the angle bisector and a boundary circle and check whether both new bundles of lines can be blocked by 2 disks.

Blocking the lines by 6 or more disks has also been considered. However, experimental results show that the bundles of lines defined by the first 4 or 6 disks on regular polygon positions can be blocked by at most 5 disks for $2 \leq d \leq 4$.

5.1.2 Deriving lower bounds for small distances between disks

In this section, we explain the approach we use to obtain lower bounds on N_d. For this, we consider the set $\mathcal{L}_p \subset \mathcal{L}_U$ that consists of all the lines from $\mathcal{L}_U$ that pass through the center p of the given disk U. The cardinality of a minimum blocking set $\mathcal{D}$ for the set of lines $\mathcal{L}_p$ represents a lower bound on N_d since $\mathcal{L}_p \subset \mathcal{L}_U$. A minimum blocking set $\mathcal{D}$ can be constructed, since one can prove that in the set of minimum blocking sets for $\mathcal{L}_p$, there are always non-overlapping ones, i.e., blocking sets for which the intersection of lines blocked by any two disks consists of at most one (tangent) line. The number of non-overlapping blocking sets for $\mathcal{L}_p$ of cardinality N can be reduced to a few cases, where for each case, we can determine the largest value of d possible for that case. We will first consider the case where all lines in $\mathcal{L}_p$ can be blocked by four disks.

Let d be a distance for which the cardinality of a minimum blocking set $\mathcal{D}$ is $N = 4$. The subset of lines from $\mathcal{L}_p$ blocked by a disk δ_i is given by *blocking angle* α_i, defined by the two lines in $\mathcal{L}_p$ that are tangent to δ_i. The blocking angle α_i is given by

$$\alpha_i = 2 \arcsin \frac{1}{d_i} \tag{5.1}$$

where d_i is the distance between p and the center of disk δ_i (See Figure 5.7).

Let $\delta_1, \delta_2, \delta_3$ and δ_4 be the four disks of a blocking set $\mathcal{D}$, and let d_1, d_2, d_3 and d_4 be the distances from p to their centers, respectively.

Let $\alpha_1, \alpha_2, \alpha_3$ and α_4 be the blocking angles of the disks $\delta_1, \delta_2, \delta_3$ and δ_4, respectively. We consider non-overlapping blocking sets, thus

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = \pi \tag{5.2}$$

Without loss of generality we assume

$$\alpha_1 \geq \alpha_2 \geq \alpha_3 \geq \alpha_4 \tag{5.3}$$

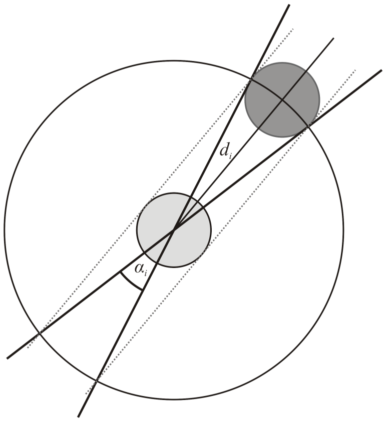

Figure 5.7. The blocking angle of a disk.

which implies

$$d_1 \leq d_2 \leq d_3 \leq d_4 \tag{5.4}$$

From (5.4) we have that $d = d_1$. Note that $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4$ is not feasible for $d > 2$, since four disks cannot be positioned in a way that each of them has a blocking angle $\pi/4$ and they do not violate a minimum distance condition.

For $\alpha_1 = \alpha_2 = \alpha_3 > \alpha_4$ we have a blocking set $\mathcal{D}$ as shown in Figure 5.8.

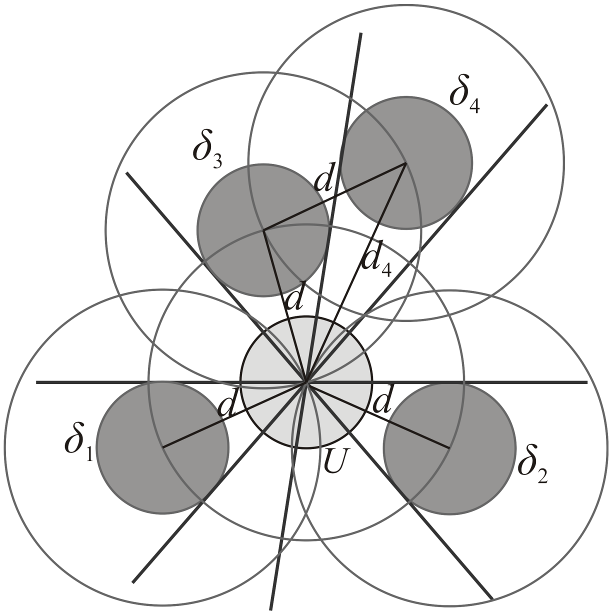

Figure 5.8. A non-overlapping blocking set of cardinality 4.

From the system of equations

$$\begin{aligned} 3\alpha_1 + \alpha_4 &= \pi \\ \alpha_1 &= 2\arcsin\frac{1}{d} \\ \alpha_4 &= 2\arcsin\frac{1}{d_4} \\ d_4 &= 2d\cos\frac{\alpha_1+\alpha_4}{2} \end{aligned} \tag{5.5}$$

we have $d = \frac{\sqrt{8}}{\sqrt{5-\sqrt{13}}}$.

We prove now that the derived distance $d = \frac{\sqrt{8}}{\sqrt{5-\sqrt{13}}}$ represents the maximum distance d for which $N = 4$. We consider an arbitrary non-overlapping blocking set $\mathcal{D}'$ of cardinality 4 such that for the blocking angles of its disks it holds that

$$\alpha'_1 > \alpha'_2 > \alpha'_3 > \alpha'_4 \tag{5.6}$$

The minimum mutual distance between the disks of $\mathcal{D}'$ is denoted as d'. From (5.6) we have that the disk δ'_1 is on the distance d' from the disk U. We have two cases:

1. If $\alpha_1 \leq \alpha'_1$ then $d \geq d'$, i.e. d is maximum distance.
2. Let $\alpha_1 > \alpha'_1$. From (5.6) and

   $$3\alpha_1 + \alpha_4 = \pi \tag{5.7}$$

   $$\alpha'_1 + \alpha'_2 + \alpha'_3 + \alpha'_4 = \pi \tag{5.8}$$

   we have

   $$\alpha_4 < \alpha'_4 \tag{5.9}$$

   At least two of four disks are adjacent having one common tangent line (in the same way as disks δ_1 and δ_4 shown in Figure 5.8). Then, the furthest disk δ'_4 is on the distance d'_4 from the center of the disk U:

   $${d'_4}^2 = x^2 + y^2 - 4 + 2\sqrt{y^2-4}\cdot\sqrt{x^2-1} \tag{5.10}$$

   where $x > d$ and $y > d$.

   The distance d_4 of the furthest disk in the blocking set $\mathcal{D}$ is

   $${d_4}^2 = d^2 + d^2 - 4 + 2\sqrt{d^2-4}\cdot\sqrt{d^2-1} \tag{5.11}$$

   From (5.10) and (5.11) we conclude that

   $$d'_4 > d_4 \tag{5.12}$$

   which implies

   $$\alpha'_4 < \alpha_4 \tag{5.13}$$

which contradicts (5.9).

Hence, we conclude that $d > d'$. In the same way it can be shown that different relations between the angles $\alpha'_1, \alpha'_2, \alpha'_3$ and α'_4 result in blocking sets with a smaller d. Therefore, the maximum distance d for which the blocking set $\mathcal{D}$ is of cardinality 4 is $d = \frac{\sqrt{8}}{\sqrt{5-\sqrt{13}}}$. The maximum distance d for which the blocking set $\mathcal{D}$ is of cardinality $N > 4$ can be derived in a similar fashion.

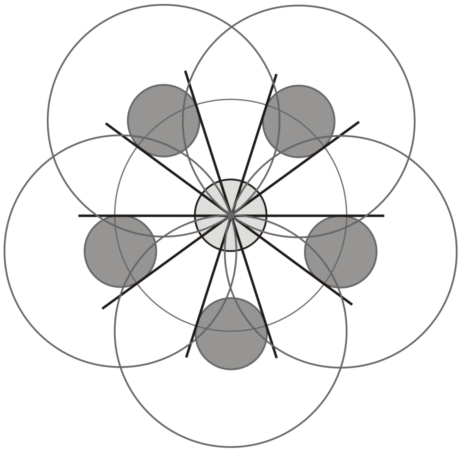

Figure 5.9. Every line passing through the center of the middle disk is blocked by at least one of the 5 disks around it.

The example in Figure 5.9 shows the optimal non-overlapping blocking set for the set of lines $\mathcal{L}_p$ for $k = 5$. The maximum distance d for which 5 disks can block the lines from $\mathcal{L}_p$ is simply derived as $d = 1/\sin\frac{\pi}{10}$.

5.1.3 Upper and lower bounds for small distances between disks: results

In this section we present the upper and lower bounds on minimum blocking sets that we obtained for $2 \leq d \leq 4$ by the methods explained in Sections 5.1.1 and 5.1.2. Furthermore, we present the asymptotic bounds on minimum blocking sets for the set of lines $\mathcal{L}_U$.

In Figure 5.10, d is given on the horizontal axis. The number N of disks is given on the vertical axis. For example, for $d = 3$, we have $5 \leq N_d \leq 9$.
Obviously, N_d is a monotonic function of d with positive integer values. Table 5.1 gives the d-values of the points where the bounds on that function change value.

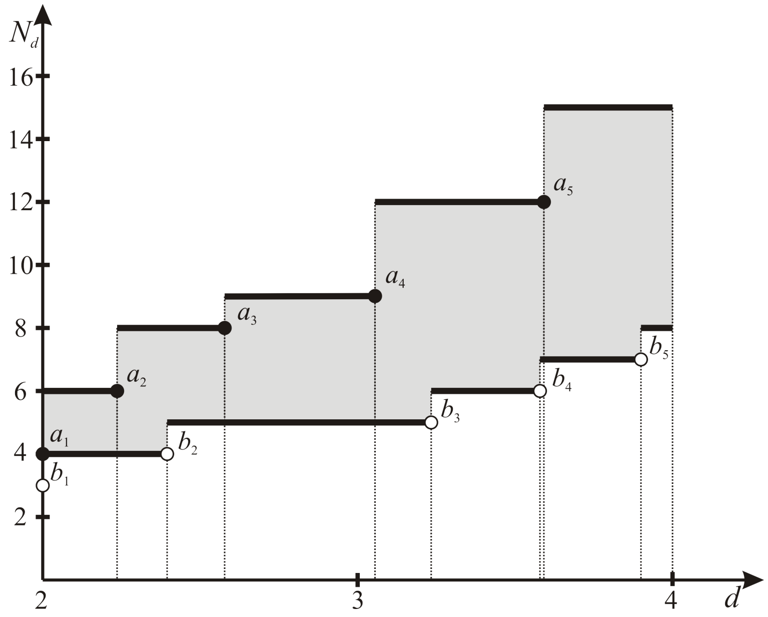

Figure 5.10. Upper and lower bounds on the cardinality of minimum blocking sets for the occluded disk problem.

The d-value of the lower bound point b_4 is

$$d_4 = 1/\sqrt{\frac{9}{16} - \frac{1}{16}y - \frac{1}{2}\sqrt{\frac{9}{32} + \frac{1}{16\sqrt[3]{18}}x + \frac{1}{8x\sqrt[3]{12}} + \frac{3}{32y}}},$$

where $x = \sqrt[3]{81 - \sqrt{6549}}$ and $y = \sqrt{9 - \frac{4}{x}\sqrt[3]{\frac{2}{3}} - 2x\sqrt[3]{\frac{4}{9}}}$.

i	1	2	3	4	5
a_i	2	2.2361	2.5776	3.0551	3.5914
b_i	$1/\sin\frac{\pi}{6}$	$\frac{\sqrt{8}}{\sqrt{5-\sqrt{13}}}$	$1/\sin\frac{\pi}{10}$	d_4	$4\cos\frac{\pi}{14}$

Table 5.1. Values of d where upper bounds a_i and lower bounds b_i change value.

By randomly generating blocking sets, we did not obtain sets with less disks than the corresponding upper bounds. This indicates that probably the lower bounds can be improved.

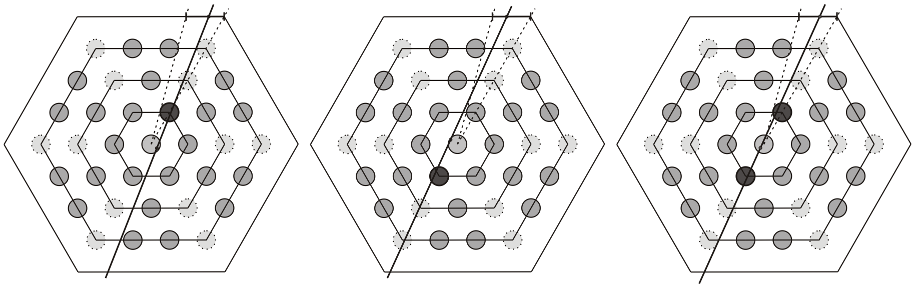

Figure 5.11. The line defined by an arbitrary point in U and an arbitrary point of the projection interval of a disk δ is blocked by δ - left, blocked by δ' that is symmetric to δ - middle, and blocked by both δ and δ' - right.

5.1.4 Asymptotic bounds

In this section, we derive upper and lower bounds on N_d, for $d \to \infty$. We first show that the blocking sets for $\mathcal{L}_p$ that we constructed in Section 4.3 with additional three disks are also blocking sets for $\mathcal{L}_U$. In more detail, let $\bar{\mathcal{D}}$ be a blocking set for $\mathcal{R}_p$, constructed as explained in Section 4.2.1, for a given distance d. Let ℓ be the line containing an arbitrary point on one edge of the hexagon H_d and an arbitrary point in disk U. The chosen point on the hexagon edge belongs to at least one interval that is the projection of a disk δ in $\bar{\mathcal{D}}$ from p onto the hexagon edge. One can prove that line ℓ is then blocked by δ or blocked by disk δ' that is symmetric to δ with respect to p, or it is blocked by both δ and δ'; see Figure 5.11. More precisely, if the line ℓ is blocked by a disk δ that is positioned in one of the hexagons $H_2,\ldots,H_{d-1}$, then ℓ is also blocked by the disk δ' that is diametrically opposite to δ. Hence, besides the six disks of the first hexagon H_1, we need only half of the disks in hexagons $H_2,\ldots,H_{d-1}$ to block all lines in $\mathcal{L}_U$. In other words, the blocking set for the set of lines $\mathcal{L}_p$ with additional three disks placed in the first hexagon is a blocking set for the set of lines $\mathcal{L}_U$; see Figure 5.12 in comparison to Figure 4.12.

Consequently, an upper bound on N_d for the occluded disk problem is given by

$$\lim_{d\to\infty}\frac{N_d}{d^2}\le\frac{9}{\pi^2}.$$

Since $\mathcal{L}_p \subset \mathcal{L}_U$, we need at least as many disks to block all the lines intersecting disk U as we need to block all the lines containing its center p. Therefore, a lower bound for the occluded disk problem is given by

$$\lim_{d\to\infty}\frac{N_d}{d^2}\ge\frac{\pi^2}{64}.$$

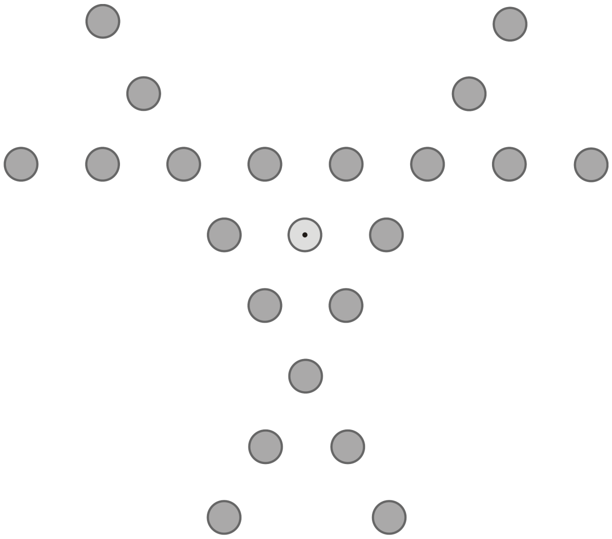

Figure 5.12. An example of a blocking set of disks for the occluded disk problem for $d = 5$.

Combining the results on upper and lower bounds for the occluded disk problem, we get to the following result.

Corollary 5.1. *For the minimum number N_d of disks needed to block all lines in $\mathcal{L}_U$, it holds that*

$$\frac{\pi^2}{64} \leq \lim_{d\to\infty} \frac{N_d}{d^2} \leq \frac{9}{\pi^2}.$$

□

5.2 The hidden disk problem

The results presented so far show that the detection algorithms presented in Chapter 2 cannot distinguish real objects from the occlusion reported as one or more objects. This limitation initiated a new approach to the object detection problem: the game pieces can also be equipped with emitters like LEDs and/or sensors, which would potentially lead to an improved detection. In this section, we discuss the problem of "hiding" objects that emit light.

Let $\mathcal{R}_U$ denote the set of all rays that emanate from a given disk U.

Definition 5.3 (hidden disk). A disk U is said to be hidden by a set $\mathcal{D}$ of non-overlapping unit disks if $\mathcal{D}$ is a blocking set for the set of rays $\mathcal{R}_U$. □

Hence, the given disk U here corresponds to an object that emits light and the set

$\mathcal{R}_U$ of rays corresponds to the set of light rays emitted by that object, assuming that the object emits light in all directions from each point on its boundary. The distance between the objects, i.e., the disks, is restricted to be not less than some given distance d, as it was the case with the previously discussed visibility problems. Formally, we can define the problem of interest to us as follows.

The Hidden Disk Problem. Given a minimum mutual distance d, what is the minimum cardinality N_d of a d-apart blocking set for the set $\mathcal{R}_U$ of rays?

As for the visibility problems we discussed so far, we propose a solution by deriving upper and lower bounds on N_d. We show that both upper and lower bounds on N_d are quadratic in d, i.e., we prove that $N_d = \Theta(d^2)$. In more detail, we first show that $N \geq 6$ disks can be positioned such that they form a 2-apart blocking set. The disks of that blocking set are placed on a circle concentric to U with neighboring disks being mutually tangent. We present a simple algorithm of pushing the disks towards the center of U such that the blocking of rays is preserved. The algorithm provides a regular ordering of disks on concentric circular rings such that the disks form a d-apart blocking set, where $d > 2$. This is used to show that

$$\frac{\pi^2}{16} \leq \lim_{d\to\infty} \frac{N_d}{d^2} \leq \frac{\pi^2}{2},$$

where the lower bound is derived as an immediate consequence of the lower bound in the hidden point problem, presented in the previous chapter.

5.2.1 Blocking rays emanating from a disk

An upper bound on N_d for blocking all rays $\mathcal{R}_U$ emanating from a given disk U is not possible to determine straightforwardly from the upper bound(s) we established in Sections 4.2.1, 4.3 and 5.1.3 for the three previous blocking set problems. More precisely, disks positioned on grid points of a regular triangular grid do not form a blocking set for $\mathcal{R}_U$, regardless of the number of the hexagonal rings used. The gaps between the neighboring disks in the corners of the hexagons allow some rays to “escape”; see Figure 5.13. Therefore, we take a different approach by considering a problem equivalent to the minimum blocking set problem. We show that a 2-apart blocking set with an arbitrary number of disks can always be constructed. Such a blocking set can then be transformed into a d-apart blocking, with $d > 2$, using a simple algorithm for which the blocking of rays is preserved. Finally, by maximizing the distance d for which a 2-apart blocking set can be transformed into a d-apart blocking set with the same number of disks, we derive an upper bound on N_d.

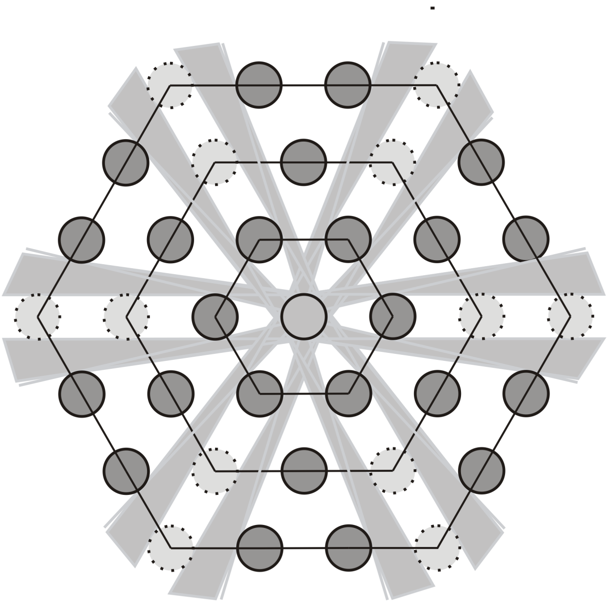

Figure 5.13. The rays emanating from U that are not blocked by any of the disks that block all the rays emanating from P.

We first define a problem that is equivalent to the hidden disk problem.

Maximum Distance Blocking Set Problem. *Given N unit disks, what is the maximum distance d for which the disks form a d-apart blocking set for the set* $\mathcal{R}_U$ *of rays?*

Next, we present an ordering of disks that enables blocking all rays from $\mathcal{R}_U$ for a given number N of disks. We assume for convenience that $N = 6\eta$. The N disks are placed on a circle c concentric to the given disk U, such that the centers of the disks are on the circle c and there is no gap between neighboring disks; see Figure 5.14. More precisely, two neighboring disks positioned on c are mutually tangent. The radius r_c of circle c is easily derived from $r_c = 1/\sin\frac{\pi}{6\eta}$. Given the mutual tangency of each pair of neighboring disks, one can easily see that any ray $\rho \in \mathcal{R}_U$ is blocked by at least one and at most two disks of the given set of 6η disks. Hence, these disks form a blocking set. The distance between two neighboring disks on c is 2, while the distance between U and a disk from the blocking set is at least 2 for any $\eta \geq 1$. Therefore, the constructed blocking set is 2-apart. Let this blocking set be denoted by $\mathcal{D}_2$.

For the maximum distance blocking set problem, we are interested in the maximum distance d for which the 6η disks form a d-apart blocking set for $\mathcal{R}_U$. As such, the

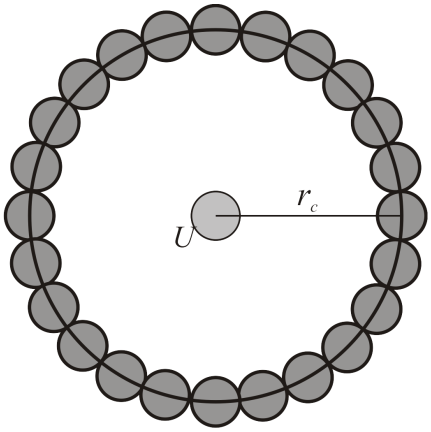

Figure 5.14. 24 disks positioned on the circle of radius r_c concentric to U.

problem appears to be hard: constructing a d-apart blocking set for an arbitrary d is certainly challenging, because it requires proving that a set of N disks is a blocking set. Therefore, we focus on *transforming* the constructed 2-apart blocking set into a d-apart blocking set.

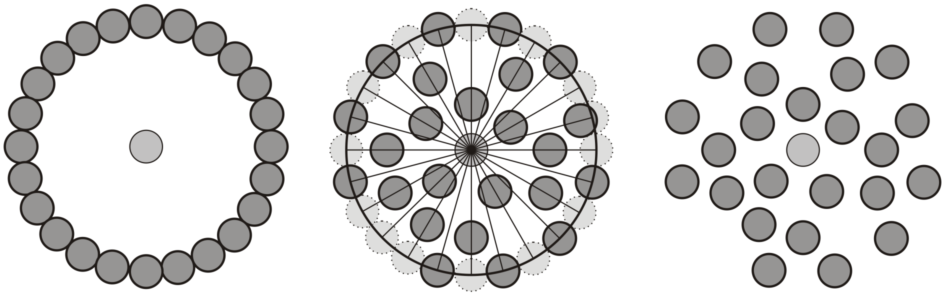

Figure 5.15. Transformation of $\mathcal{D}_2$ into a d-apart blocking set.

In order to transform $\mathcal{D}_2$ into a d-apart blocking set, with $d > 2$, the disks of $\mathcal{D}_2$ should be separated from each other, while the blocking of all rays should be preserved. To define a step of the proposed transformation we use Lemma 5.1, where it is shown that the rays blocked by a given disk δ are still blocked by δ after the disk is moved towards the center of U, i.e., along the line segment that connects the two positions of δ. Consequently, a transformation of the blocking set $\mathcal{D}_2$ where some disks of $\mathcal{D}_2$ are shifted from their original position on circle c towards the center of U represents a transformation into a d-apart blocking set, where d is the minimum of all pair-wise distances between the disks; see Figure 5.15. The problem of interest to us now is to determine the maximum d for which we can transform $\mathcal{D}_2$ into a d-apart blocking set.

5.2.2 Ordering disks on circular rings

In Section 5.2.1 we proved that we can construct blocking sets by pushing the disks of $\mathcal{D}_2$ into the interior of the circle c, given that the disks are moved in the direction of the center of c. In this section we propose a regular ordering of disks forming a blocking set that can be obtained as follows.

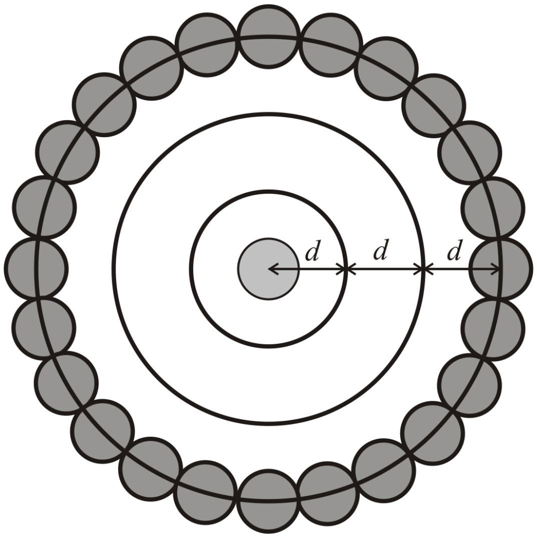

Figure 5.16. The definition of three circular rings with radii d, $2d$ and $3d$.

Let $\mathcal{D}_2$ be the 2-apart blocking set constructed as in Section 5.2.1, consisting of 6η disks. In the interior of the circle c we can define a number of circles called *rings* and denoted as $c_1, c_2, \ldots, c_k$, where the radius of the ring c_1 is d, the radius of c_2 is $2d$, etc. The last ring c_k with the radius kd is assumed to be the given circle, which has radius $r_c = 1/\sin\frac{\pi}{6\eta}$; see Figure 5.16. In the process of shifting the disks of $\mathcal{D}_2$ towards the center, we place the center of each of them exactly on one of the rings.

The line segment that connects the center of a disk in $\mathcal{D}_2$ and the center of U is called a *thread*. Thus, the disks of $\mathcal{D}_2$ define 6η threads. Since we chose to place the disks on the rings and the disks can be moved only along their threads, each disk can be placed in one of the k intersection points of its thread and the k rings. Note that the d-apart rings ensure that the distance between any two disks positioned on *different rings* is at least d. However, choosing an arbitrary ring for each disk may result in two disks of the *same ring* being less than distance d apart;

see Figure 5.17.

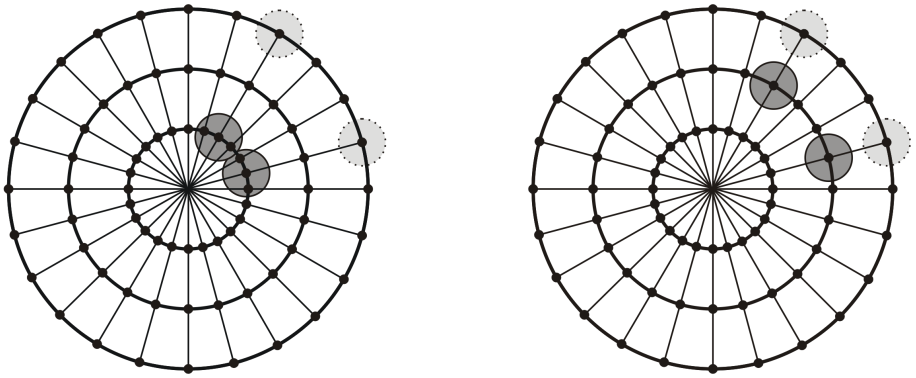

Figure 5.17. Shifting two disks onto inner rings: on the left, the disks are not d-apart, and on the right, the disks are d-apart.

The number k of rings determines the distance d for given η. Given that the radius of the largest ring is $r_c = 1/\sin\frac{\pi}{6\eta}$ and as we mentioned above $r_c = kd$, we have that

$$d = \frac{1}{k\sin\frac{\pi}{6\eta}}. \tag{5.14}$$

Hence, in order to maximize the distance d, we need to minimize the number k of rings needed, for 6η disks to form a d-apart blocking set. We assume the rings to be numbered from inside to outside.

For a ring of given radius, it is easy to determine the maximum number of disks that can be positioned equally spaced, such that the distance between two neighboring disks on this ring is at least d. For example, at most 6 disks can be placed on the first ring, at most 12 disks on the second ring, at most 18 disks on the third ring, etc. In this way, we can easily derive a lower bound on the minimum number k of rings needed, for a given η. However, the minimum number of rings that suffices for disks to form a d-apart blocking set is often larger than this lower bound. This is because of the restriction of fixed positions for placing the disks, which does not always allow placing the maximum number of disks on the rings. In the construction we propose, we place less than the maximum number of disks on some of the rings or even keep some of the rings empty.

In more detail, we choose to place $6\eta_j$ disks on the j-th ring, where

$$\eta_j = 2^{\lfloor \log_2 j \rfloor}, \tag{5.15}$$

such that the disks form a regular polygon. Note that $6\eta_j$ is *equal* to the maximum number of disks that can be placed, only for the rings $j = 2^l$, for some $l \geq 0$, however, it is *less* than maximum for all other rings; see the comparison given in Table 5.2.

Ring j	1	2	3	4	5	6	7	8
Max disks	6	12	18	24	30	36	42	48
$6\eta_j$	6	12	12	24	24	24	24	48

Table 5.2. The maximum number of disks and the chosen number of disks for rings 1 to 8.

For symmetry reasons, we focus on one of the six sections of $\mathcal{D}_2$ with η disks. We show that any set of η disks can be split into k subsets, where the j-th subset contains either $2^{\lfloor \log_2 j \rfloor}$ or 0 disks. The j-th subset is then placed on the j-th ring such that the distance between each two disks is at least d. More precisely, we show that the given number η can be represented as

$$\eta = \bar{\eta}_1 + \bar{\eta}_2 + \cdots + \bar{\eta}_k, \tag{5.16}$$

where $\bar{\eta}_j \in \{0, \eta_j\}$, or simplified, any natural number η can be represented as

$$\eta = b_0 + \underbrace{2+2}_{max\ 2} + \underbrace{4+4+4+4}_{max\ 4} + \cdots + \underbrace{2^y + 2^y + \cdots + 2^y}_{max\ 2^y}, \tag{5.17}$$

for some $y \geq 0$ and $b_0 \in \{0,1\}$. Note that the total number of addends in (5.16) is k, i.e., each addend corresponds to a ring, more precisely, to the number of disks placed on each of the six sections of the ring. This results in including the zero-addends in counting, since they indicate the presence of empty rings. More precisely, we include the zero-addends in counting when we have less than the maximum number of equal addends, for all addends except for the largest ones. For example, $\eta = 15$ can be represented as $15 = 1+2+0+4+4+4$ and the number of rings needed is $k = 6$, with the third ring being empty.

Formally, we prove the existence of a representation of η given by (5.17), using the following lemma.

Lemma 5.2. *For any positive integer* η *a sequence* $\mathcal{A}_\eta = (a_0, a_1, \ldots, a_y)$ *exists such that*

$$\eta = \sum_{i=0}^{y} a_i \cdot 2^i \tag{5.18}$$

where $0 \leq a_i \leq 2^i$ *and* $a_y > 0$.

Proof. The proof of the lemma follows from the binary scale representation of η. □

For a given η, there are generally multiple sequences $\mathcal{A}_\eta$. From Equation (5.14), to construct a d-apart blocking set, where distance d is as large as possible, we need to minimize the number k of rings. The number of rings we define is equal to the number of addends in (5.17). Hence, the number k of rings is given by

$$k = (1+2+4+8+\cdots+2^{y-1}) + a_y = 2^y - 1 + a_y \tag{5.19}$$

where a_y is the number of addends of size 2^y in (5.17). Hence, our interest is in the sequences $\mathcal{A}^*_\eta$ for which $2^y + a_y$ is minimal.

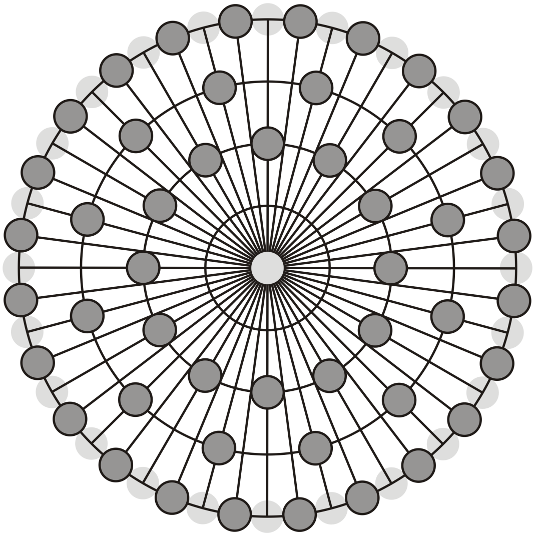

Figure 5.18. An example of a d-apart blocking set for $n = 8$, where $d \approx 4$.

Disk ordering algorithm

In the previous section we showed how to determine the number of rings and the number of disks on each of them, using Lemma 5.2 and choosing the sequence $\mathcal{A}^*_n$ for which the number of rings is minimal. In this section, we present an algorithm that given the sequence $\mathcal{A}^*_n$, for each disk of $\mathcal{D}_2$ determines the ring on which it should be placed, which results in the disks forming a d-apart blocking set; see Figure 5.18.

We restrict ourselves to finding the solutions for all η that are divisible by their largest addend 2^y in the representation (5.17). Note that $2^y|\eta$ implies that $\eta_j|\eta$, for all j.

Let us define a table τ with k rows and η columns, such that each thread corresponds to one column of τ and each ring corresponds to one row of τ, with the outermost ring corresponding to the top row. Each cell of the table τ then represents a position on which the corresponding disk can be placed, i.e., it is the intersection of its thread and a ring. When one disk is moved to a certain position, the value in the corresponding cell of τ is set to 1 or "full", while the other cells of the same column have values 0 or "empty"; see Figure 5.19. The defined table represents one of the six identical sections of the blocking set, thus, we consider the table as if its columns are cyclic (its first and its last column are connected).

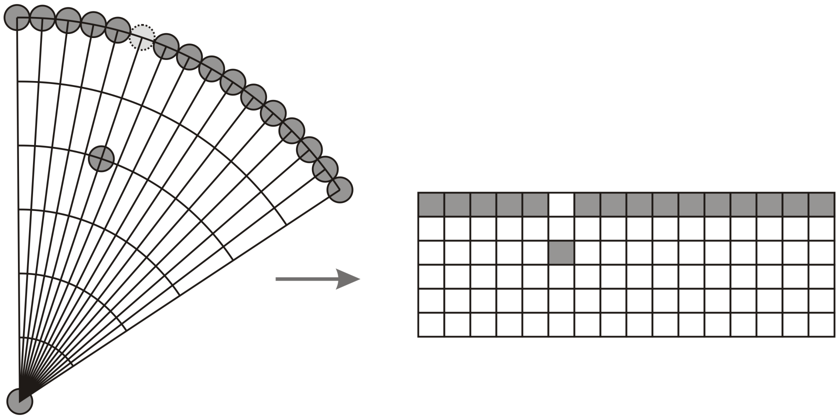

Figure 5.19. A set of 16 disks with 6 rings and the corresponding 6 x 16 table.

An ordering of full cells in a table τ is called *valid* if and only if the following conditions hold:

- There is exactly one full cell in each column;
- The j-th row is either empty or it contains exactly η_j full cells;
- The number of empty cells between any two successive full cells in the j-th row is exactly $\frac{\eta}{\eta_j} - 1$.

Lemma 5.3. *A valid table τ exists for any positive integer τ represented by (5.17) for which $2^y | \eta$.*

Proof. The proof of the lemma is given by a method for constructing a valid table, which follows from the equation $2^x = 2^{x-1} + 2^{x-1}$. In more detail, a complete row of full cells can be split into $\eta/2^y$ rows, where each row contains 2^y full cells, as illustrated in Figure 5.20. Each of the resulting rows can again be split into two rows, by pushing every second full cell to a new row. After a finite number of "splitting" steps, each row corresponds to a non-zero addend in representation (5.17). The rows can be swapped then if necessary, such that each row τ_{i+1} that is directly above a row τ_i contains at least the same number of full cells as τ_i. The process is completed by inserting empty rows where needed. □

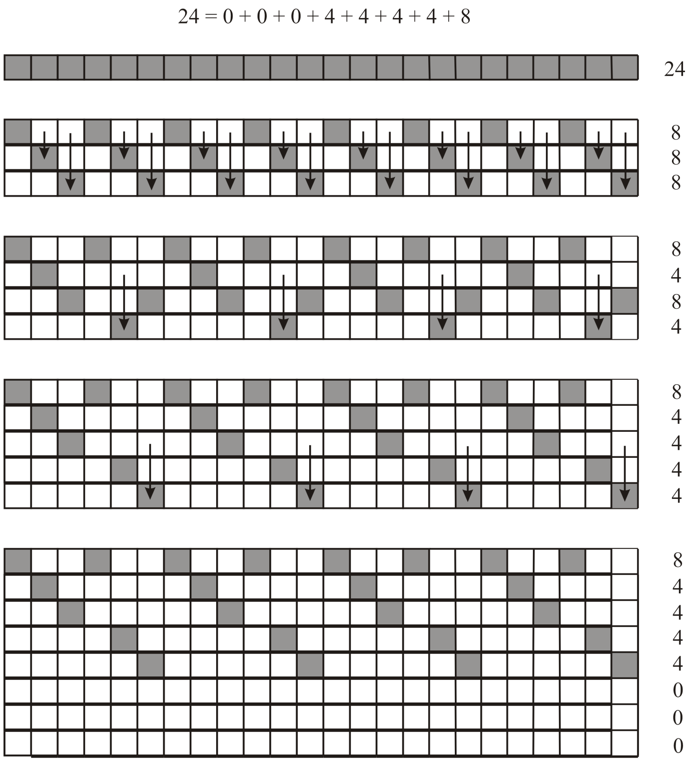

Figure 5.20. Constructing a valid table for $n = 24$.

Note that the proof of Lemma 5.3 represents a *disk ordering algorithm* that for each of the n disks determines the ring on which it should be placed, such that the disks form a d-apart blocking set.

5.2.3 Upper and lower bounds

In Sections 5.2.1 and 5.2.2, we showed that we can construct a d-apart blocking set for each η that is divisible by its largest addend in representation (5.17). In this section, we present upper and lower bounds on the cardinality N_d of such a blocking set, as a function of the minimum distance d. We start by deriving an upper bound.

One can easily show that the ordering of disks presented in Section 5.2.2 implies that the minimum of all pair-wise distances between the disks is d. The relation between the distance d, the given number η and the corresponding number k of rings is given by

$$d = \frac{1}{k \sin \frac{\pi}{6\eta}} \tag{5.20}$$

From the choice of sequence $\mathcal{A}^*_\eta$ in Lemma 5.2, for which $a_y + 2^y$ is minimal, we have that

$$\sum_{j=0}^{y-1} 2^{2j} + (a_y - 1) \cdot 2^y \leq \eta \tag{5.21}$$

where a_y is the number of largest addends 2^y in representation (5.17). From (5.21) and

$$\sum_{j=0}^{y-1} 2^{2j} = \frac{1}{3}(4^y - 1) \tag{5.22}$$

it follows that

$$4^y + 3(a_y - 1)2^y \leq 3\eta + 1 \tag{5.23}$$

With further transformations of inequality (5.23) we have

$$\begin{aligned} ((2^y)^2 + 2(a_y - 1)2^y) + (a_y - 1)2^y &\leq 3\eta + 1 \\ \Leftrightarrow k^2 + (a_y - 1)(2^y - a_y + 1) &\leq 3\eta + 1 \end{aligned} \tag{5.24}$$

Since $1 \leq a_y \leq 2^y$, we have that

$$(a_y - 1)(2^y - a_y + 1) \geq 0 \tag{5.25}$$

Finally, from (5.24) and (5.25), we bound the number k of rings by a function in η as follows.

$$k \leq \sqrt{3\eta + 1} \tag{5.26}$$

We transform (5.20) into

$$\frac{1}{kd} \leq \sin\frac{\pi}{6\eta} \tag{5.27}$$

and multiply (5.26) by $\sqrt{\eta}$

$$k\sqrt{\eta} \leq \sqrt{3\eta^2 + \eta} \tag{5.28}$$

Multiplication of (5.27) and (5.28) and expressing the limit for $d \to \infty$, results in

$$\lim_{d\to\infty} \frac{\eta}{d^2} \leq \frac{\pi^2}{12} \tag{5.29}$$

and since $N = 6\eta$, we derived an upper bound on N_d, i.e.,

$$\lim_{d\to\infty} \frac{N_d}{d^2} \leq \frac{\pi^2}{2} \tag{5.30}$$

The lower bound on the minimum number of disks which form a d-apart blocking set for the set of all rays emanating from a single point is $\frac{\pi^2}{16}d^2$, where d tends to infinity, as given by Theorem 4.5 in Section 4.2.2. To block the rays emanating from a given unit disk we need at least as many as to block the rays emanating from its center. Hence, the lower bound on the minimum number N_d of disks is given by

$$\lim_{d\to\infty} \frac{N_d}{d^2} \geq \frac{\pi^2}{16}. \tag{5.31}$$

Combining the results of (5.30) and (5.31), we proved the following theorem.

Theorem 5.1. *For the minimum cardinality N_d of a d-apart blocking set to block all rays emanating from a unit disk we have*

$$\frac{\pi^2}{16} \leq \lim_{d\to\infty} \frac{N_d}{d^2} \leq \frac{\pi^2}{2}.$$

□

5.2.4 Discussion

We expect that both bounds, especially the upper bound, can be further improved. The following discussion provides some directions for potential improvements.

Constructing a d-apart blocking set from $\mathcal{D}_2$ through a sequence of transformation steps where a number of disks is pushed towards the center results in the rather large constant $\pi^2/2$. The disks pushed inside circle c block much larger sets of rays than the sets of rays they block from their original positions on c. Consequently, the sets of rays blocked by two disks on different rings may not be disjoint. This implies that constructing blocking sets for which the overlap of sets of blocked rays is minimized may potentially provide a better upper bound. In addition, the number of disks on one ring is less than the maximum possible number for the majority of rings. Placing the maximum number of disks on each of the rings may further improve the upper bound. The combination of the last two conjectures may be used to define an optimization problem, similar to the problem of opening a combination lock with k rings, i.e. to find the rotation angle for each of the k rings that are d-apart and contain the maximum number of d-apart disks, such that the disks form a blocking set and the total overlap of blocked rays is minimized. We expect that the solution of this problem provides a better upper bound. The main challenge here is still the problem of proving that a set of disks, positioned following some constraints, is a blocking set for the set $\mathcal{R}_U$ of all rays.

5.3 The merging disks problem

Let U_1 and U_2 be two non-overlapping unit disks in the plane such that the distance between them is d. Furthermore, let $\mathcal{L}(U_1,U_2)$ be the set of all lines in the plane that do not intersect U_1 and U_2 and such that for each line $\ell \in \mathcal{L}(U_1,U_2)$, the disks U_1 and U_2 are on different sides of ℓ. Hence, the set $\mathcal{L}(U_1,U_2)$ of lines is bounded by the two internal tangent lines defined by U_1 and U_2, as illustrated in Figure 5.21.

Note that the set $\mathcal{L}(U_1,U_2)$ of lines represents a light corridor between the disks U_1 and U_2.

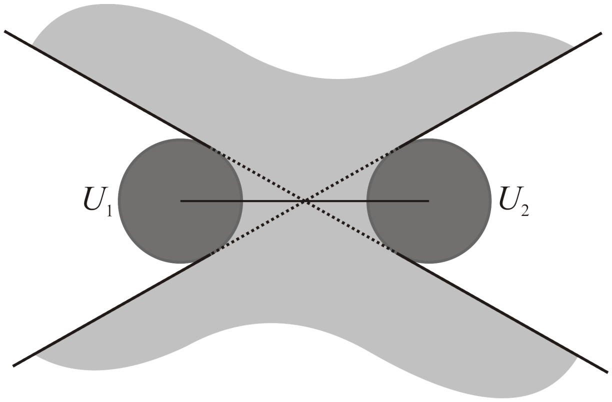

Figure 5.21. The bundle of lines between the disks U_1 and U_2.

For ease of notation, let $\mathcal{M} = \mathcal{L}(U_1,U_2)$. For some given distance d, a set $\mathcal{D}$ of non-overlapping unit disks is a d-apart blocking set for the set of lines $\mathcal{M}$ if the distance between any two disks in $\mathcal{D} \cup \{U_1,U_2\}$ is at least d. In the same way as we defined the four visibility problems considered so far, we formally define the merging disks problem as follows.

The Merging Disks Problem. *Given a minimum mutual distance d between two disks U_1 and U_2, what is the minimum cardinality N_d of a d-apart blocking set for the set $\mathcal{M} = \mathcal{L}(U_1,U_2)$ of lines?*

Note that the merging disks problem is in essence the bundle blocking problem, as defined in Section 5.1.1, where for $d \to \infty$, the set of lines to be blocked in the merging disks problem is significantly larger than the corresponding set of lines in the occluded disk problem. For small values of d, however, we use the methods and results presented in Section 5.1.1 to determine the minimum blocking sets for the merging disks problem.

5.3.1 Minimum blocking sets for small distances

Let us now take a closer look at examples of minimum blocking sets for the set of lines $\mathcal{M}$, for small values of d, more precisely, for $2 \leq d \leq 3$. If $d = 2$, then the disks U_1 and U_2 are mutually tangent, which implies that $\mathcal{M} = \emptyset$. The two disks then define one shadow region, as illustrated in Figure 3.6.

We next determine the maximum distance d for which the set $\mathcal{M}$ of lines can be blocked by one, two and three disks.
The problem of blocking a bundle of lines by one disk is already discussed in

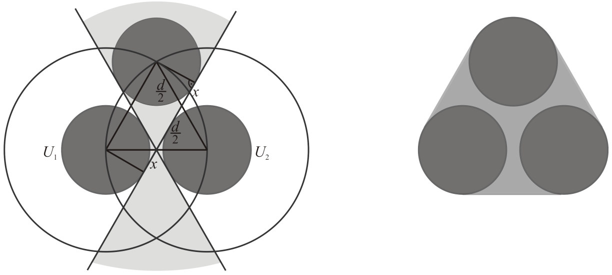

Figure 5.22. Left: determining the maximum distance d for which one disk can block the set $\mathcal{M}$ of lines; right: the shadow region defined by the three disks.

Section 5.1.1. In order to determine the maximum distance d for which $\mathcal{M}$ can be blocked by one disk, this "blocking" disk is positioned such that it is on distance d from disks U_1 and U_2 and it is tangent to both bounding lines of the set $\mathcal{M}$ of lines, as it is shown in Figure 5.22. From the system of equations

$$\begin{aligned} x^2 &= \frac{d^2}{4} - 1 \qquad (5.32) \\ d &= \frac{d}{2} + 2x \qquad (5.33) \end{aligned}$$

we have that $d = \frac{4}{3}\sqrt{3}$, or $d \approx 2.3094$.

Note that, for symmetry reasons, U_1 blocks the bundle of lines between U_2 and the blocking disk, and U_2 blocks the bundle of lines between U_1 and the blocking disk. Hence, the three disks are contained in one shadow region, i.e., the three corresponding circular objects will be detected as one large object; see Figure 5.22.

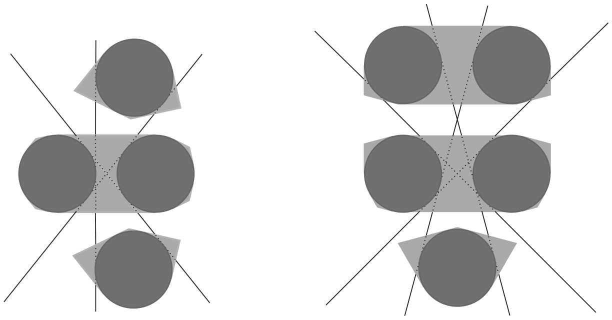

Figure 5.23. Blocking a bundle of lines between the given two disks by two and three disks and the corresponding sets of shadow regions.

The maximum distance d for which two and three disks block the set $\mathcal{M}$ of lines can be derived in a similar fashion, using elementary calculus. The positioning of the blocking disks and the corresponding shadow regions defined by the disks is illustrated in Figure 5.23. The corresponding values of the derived maximum distance are $\sqrt{\frac{9+\sqrt{17}}{2}}$ and $\sqrt{8}$, for the blocking sets of two and three disks, respectively. Note that there are three shadow regions defined in both cases.

Figure 5.24 illustrates a blocking set for $\mathcal{M}$ consisting of four disks for $d = 3$. The blocking set is constructed using a heuristic algorithm similar to the algorithm we used to determine the upper bounds for small values of d in the occluded disk problem. Since the maximum distance for which three disks can block $\mathcal{M}$ is $d < 3$ (see the results presented below), we conclude that the constructed blocking set is of minimum cardinality.

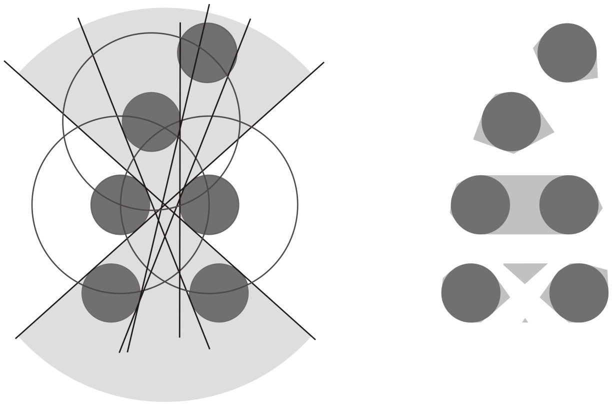

Figure 5.24. Left: blocking a bundle of lines by four disks for $d = 3$; right: the set of shadow regions defined by the six disks as an illustration of the merging disks problem.

The minimum cardinality N_d of a blocking set for distances $2 \leq d \leq 3$ is given in Figure 5.25. The distance d is given on the horizontal axis, and the minimum cardinality N_d is given on the vertical axis.

5.3.2 Asymptotic bounds

In this section, we derive asymptotic bounds on the minimum number N_d of disks forming a d-apart blocking set for the set $\mathcal{M}$ of lines, when $d \to \infty$. An upper bound is derived using the same construction method as for the hidden disk problem presented in Section 5.2. A lower bound for the merging disks problem is a result that follows directly from the lower bound of the occluded point problem presented in Section 4.3. Combining the results on upper and lower bounds, we show that the

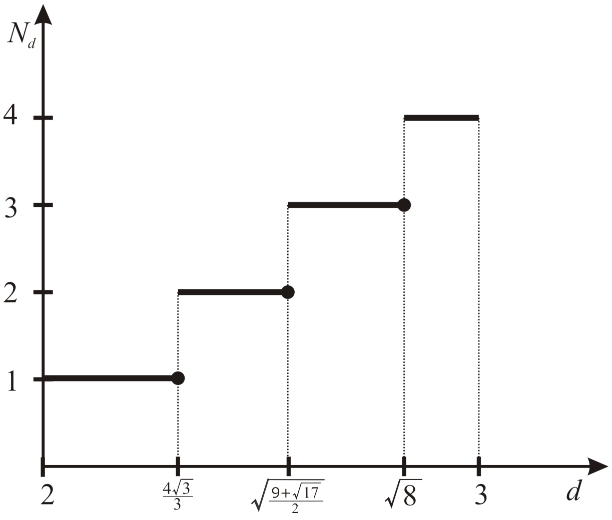

Figure 5.25. The minimum cardinality N_d of a blocking set for the merging disks problem as a function of distance d, where $2 \leq d \leq 3$.

minimum number N_d of disks for the merging disks problem is quadratic in d, i.e., we show that

$$\frac{\pi^2}{64} \leq \lim_{d\to\infty} \frac{N_d}{d^2} \leq \frac{\pi^2}{4}. \tag{5.34}$$

Let p be a midpoint of the line segment connecting the centers of the disks U_1 and U_2, and let c be a circle with center in p. In the similar way as in the previous section, we construct a 2-apart blocking set for $\mathcal{M}$ by placing N unit disks in one half-plane defined by the line connecting the centers of U_1 and U_2, such that the disks' centers are positioned on c and each two neighboring disks are mutually tangent; see Figure 5.26. This 2-apart blocking set is then transformed into a d-apart blocking set, where $d > 2$, using the disk ordering algorithm presented in Section 5.2.2. The k circular rings $c_1, c_2, \ldots, c_k$ defined in this case have radii $\frac{3d}{2}, \frac{5d}{2}, \ldots, \frac{(2k+1)d}{2}$, since a disk in the first ring must be at distance at least d to the disks U_1 and U_2, however, this has no influence on the asymptotic result. Hence, an upper bound on N_d for the merging disks problem is given by

$$\lim_{d\to\infty} \frac{N_d}{d^2} \leq \frac{\pi^2}{4} \tag{5.35}$$

Note that the upper bound can be improved by placing the disks in both half-planes, as discussed in Section 4.3 and illustrated in Figure 4.13.

To determine a lower bound on N_d, we make the following observation. For $d \to \infty$,

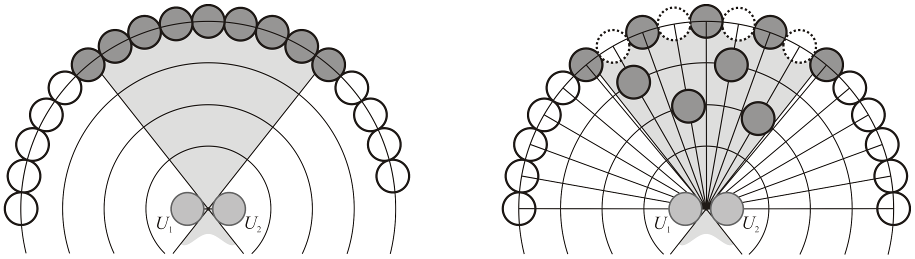

Figure 5.26. An illustration of the constructive method for deriving upper bounds for the merging disks problem in the asymptotic case.

the angle between the ”bounding” lines of the set $\mathcal{M}$ of lines, i.e., the internal tangent lines defined by U_1 and U_2, approaches π. In this case, the set $\mathcal{M}$ of lines is a superset of the set $\mathcal{L}_p$ of lines containing the point p. Therefore, we need at least as many disks to block the set of lines $\mathcal{M}$ as to block the set of lines $\mathcal{L}_p$. In other words, a lower bound for the occluded point problem is also a lower bound for the merging disks problem, i.e.,

$$\lim_{d \to \infty} \frac{N_d}{d^2} \geq \frac{\pi^2}{64} \tag{5.36}$$

In this way, we have determined upper and lower bounds on N_d for the merging disks problem that are both quadratic in d, for $d \to \infty$.

6

Conclusion

To enable interaction on multi-user table-top devices, an in-plane object detection technology has been developed by Philips Research. Objects place in the detection area, e.g., on a rectangular LCD screen, are detected using the information on blocked and non-blocked lines of sight between light emitters and light sensors positioned in a frame surrounding the detection area. By repeatedly performing the object detection process, it is possible to track moving objects, for instance, to recognize a hand gesture.

In Chapter 1 we defined the detection problem as the problem of determining approximations on the geometry of objects from the information on blocked and non-blocked lines of sight between the emitters and the sensors. In Chapter 2 we described two algorithmic solutions to the detection problem, assuming that emitters and sensors are points on the perimeter of a rectangular detection area such that the distance between each pair of neighboring points is r. Both algorithms use as an input the blocking matrix, which contains the information on blocked and non-blocked lines of sight for all emitter-sensor pairs, and provide the identical output: a set of convex polygons representing the objects. In order to ensure the detection of each of the objects, we restrict the minimum size of an object. More precisely, the algorithms are designed to only detect objects that are larger than a circle of radius r. This restriction is in favor of a guaranteed detection of objects larger than some minimum size given, over enabling detection of objects of all sizes with no guarantees on detection of small objects. In addition, the running times of both detection algorithms is significantly improved as a result of the restriction on the minimum object size. Both cut-off and point-by-point algorithms determine the set of all non-empty shadow wedge intersections that represent the detected objects. The cut-off algorithm discards the shadow wedge intersections that are not sufficiently large to contain one object. This procedure is a very important step in the early stages of the execution of the algorithm. Most of the non-object intersections are discarded then, while keeping them as potential objects results usually in a large number of further intersecting operations that cost extra computation time. The running time of the point-by-point algorithm strongly depends on the number of objects that need to be detected, and determining a tight upper bound on that number would not have been possible without a restriction on the minimum size of the objects. Hence, both algorithms enable detection of all objects larger than a circle of radius r, irrespective of their position and shape, by providing an approximation on their geometry.

With emitters and sensors being the points on the perimeter of the rectangular detection area, the set of all lines of sight defined between the emitters and the sensors partition the detection area into a large number of small convex polygons. From a given partitioning, one can find the mapping between each point in the detection area and the minimum radius of a circular object with the center in that point that is surely detectable. This gives rise to many optimization problems, such as the problem of finding the positioning of a fixed number of emitters and sensors that provides the optimal detectability and the problem of finding the minimum number and positioning of emitters and sensors that provides some predefined detectability. In addition to the considered environment setup, one can address similar optimization problems in a more realistic environment where neither emitters nor sensors are points in the frame, but instead, they have a width of approximately 5 mm. In this way, the minimum size of a surely detectable object is implied by the hardware limitations, i.e., the detection of objects smaller than a circle of radius 5mm is not guaranteed. Furthermore, different types of emitters have different light intensity distributions over the emission angle, which ranges from 0 to π in our case. However, there is no emitter with uniform light intensity distribution over the entire range. The light intensity detectable by the sensors is usually within the angle of 30-40 degrees, which reduces the number of lines of sight that can be used in the detection process to approximately 20% of all of the lines of sight.

Ideally, the detection process results in the reporting of all non-empty n-wedge intersections, where each n-wedge intersection corresponds to one object and moreover, each of these intersections has the same shape and size as the corresponding object. However, one n-wedge intersection can correspond to 0, 1 or more objects.

Even in the case where each reported n-wedge intersection corresponds to exactly one object, the size of the two is never the same. This gives rise to the definition of object detection accuracy. The measure of accuracy can simply be defined as the ratio between the total area occupied by the objects and the total area of reported n-wedge intersections. In a broader sense, each detection failure in the form of a visibility problem, such as reporting multiple objects as one or reporting non-existing objects, implies further decreasing of the measure of detection accuracy.

Regardless of the definition of the accuracy measure, one can reason that the level of the detection accuracy is a direct consequence of the number of blocked lines of sight. The number of blocked and non-blocked lines of sight depends on two essentially different factors: (1) the number and the density of emitters and sensors and (2) the positions and size of objects placed in the detection area. Clearly, the larger the number of emitters and sensors, the larger is the number of lines of sight defined by them, which consequently results in higher accuracy of detecting objects. Consequently, the maximum level of detection accuracy that can be achieved is when infinitely many emitters and sensors are positioned in a frame surrounding the objects, so that the distance r between an emitter and its neighboring sensor goes to zero.

In Chapter 3, we consider detecting identical circular objects assuming the above mentioned ideal environmental parameters. As a consequence of such an assumption, each line in the plane can be considered as a line of sight between one emitter and one sensor, which results in the maximum level of accuracy that can be achieved for a given placement of objects in the detection area. An algorithm that provides the same output as a detection algorithm would provide in the ideal detection environment is the shadow regions algorithm presented in Section 3.3. The algorithm determines the set of all shadow regions, which is a direct consequence of the objects' relative positions.

One can observe that the size of the shadow regions is closely related to the distance between the objects. More precisely, the closer the objects are to each other the larger are the shadow regions that they define. The objects that are close to each other can also cause many visibility problems in the process of object detection. For instance, a large free shadow region, i.e., a shadow region that does not contain any object, may be interpreted by the detection device as an object in the detection area. The light passing between two objects that are relatively close to each other can be easily blocked by other objects, causing in that way the two objects to be detected as one, and generally, this type of problem cannot be resolved easily when there is no additional information on the number and/or the size of the objects and their mutual distance.

In order to avoid these visibility problems, we explored the relation between the occurrence of ”free” shadow points/regions and the mutual distance between the objects in Chapters 4 and 5. Two visibility problems discussed in Chapter 4, namely, the hidden point problem and the occluded point problem, are the starting point of the investigation. The methods and solutions of these problems represent the essence in solving the more challenging problems in Chapter 5, namely, the occluded disk problem, the hidden disk problem and the merging disks problem.

For each of the five visibility problems considered, the distance between two objects is restricted to be at least some given distance d. We also restrict ourself to circular objects of identical size. The reason for these restrictions comes from the application domain. Many board games use pawns that are circular on the bottom. In addition, the pawns as well as the playing fields on the board can be designed so that the pawns cannot be on a distance less than some predefined distance.

The visibility problems considered are in essence the problems of finding the minimum number N_d of disks that blocks a given set of lines or rays, where the minimum mutual distance d between the disks is given. The five problems discussed are defined for five different sets of lines/rays (1) the set of all rays emanating from a point, (2) the set of all lines containing a point, (3) the set of all lines intersecting a disk, (4) the set of all rays emanating from a disk, and (5) the set of all lines passing between two disks. We showed that for each of the problems, the minimum number N_d of disks needed to block the given set of lines/rays is quadratic in the minimum mutual distance d between two disks. This result follows from deriving upper and lower bounds on N_d that are both quadratic in d.

Bibliography

AGARWAL, P.K., J. MATOUSEK, AND O. SCHWARZKOPF [1998], Computing many faces in arrangements of lines and segments, *SIAM J. Comput.* **27**, 491–505.

AGARWAL, P.K., AND M. SHARIR [2000], Arrangements and their applications, *Handbook of Computational Geometry*, 49–119.

ALLEN, T.T. [1986], Polya's orchard problem, *The American Mathematical Monthly* **93**, 98–104.

BALABAN, I. [1995], An optimal algorithm for finding segment intersections, *Proceedings of the 11th ACM Symposium on Computational Geometry*, Vancouver, Canada, 211–219.

BENTLEY, J.L., AND T.A. OTTMANN [1979], Algorithms for reporting and counting geometric intersections, *Computers, IEEE Transactions on* **100**, 643–647.

BERG, M. DE, M. VAN KREVELD, M. OVERMARS, AND O. SCHWARZKOPF [2000], *Computational Geometry: Algorithms and Applications*, Springer-Verlag.

BERGMAN, T., AND G. HOLLEMANS [2006], Making electronic games more sociable, *Password, Philips Research technology magazine*, 24–27.

BOLTJANSKY, VG [1960], The problem of the illumination of the boundary of a convex body, *Russian, Izv. Mold. Fil. Akad. Nauk SSSR (76)* **10**, 77–84.

BOROCZKY, K., AND V. SOLTAN [1996], Translational and homothetic clouds for a convex body, *Studia Scientiarum Mathematicarum Hungarica* **32**, 93–102.

CHAZELLE, B. [1986], Reporting and counting segment intersections, *Journal of Computer and System Sciences* **32**, 156–182.

CHAZELLE, B., AND DP DOBKIN [1987], Intersection of convex objects in two and three dimensions, *Journal of the ACM (JACM)* **34**, 1–27.

CHAZELLE, B., AND H. EDELSBRUNNER [1992], An optimal algorithm for intersecting line segments in the plane, *Journal of the ACM (JACM)* **39**, 1–54.

CHEW, W.C., AND Y.M. WANG [1990], Reconstruction of two-dimensional permittivity distribution using the distorted born iterative method, *Medical Imaging, IEEE Transactions on* **9**, 218–225.

CLARKSON, K., H. EDELSBRUNNER, L. GUIBAS, M. SHARIR, AND E. WELZL

[1990], Combinatorial complexity bounds for arrangements of curves and spheres, *Discrete and Computational Geometry* **5**, 99–160.

CONWAY, J.H., AND R.K. GUY [1995], *The Book of Numbers*, Springer.

CONWAY, J.H., N.J.A. SLOANE, AND E. BANNAI [1999], *Sphere packings, lattices, and groups*, Springer Verlag.

CORMEN, T., C.E. LEISERSON, R.L. RIVEST, AND C. STEIN [2001], *Introduction to algorithms*, MIT press.

CZYZOWICZ, J., E. RIVERA-CAMPO, J. URRUTIA, AND J. ZAKS [1989], Illuminating lines and circles on the plane, *University of Ottawa Computer Science TR-89-49*.

DEANS, S.R. [1983], 283, Wiley New York:.

DICKSON, L.E. [2005], Volume I: Divisibility and Primality, Dover Publications.

DIETZ, P.H., AND B.D. EIDELSON [2009], Surfaceware: dynamic tagging for microsoft surface, *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, TEI '09, New York, NY, USA, ACM, 249–254.

DIETZ, P., AND D. LEIGH [2001], Diamondtouch: a multi-user touch technology, *Proceedings of the 14th annual ACM symposium on User interface software and technology*, ACM, 219–226.

DUMITRESCU, A., AND M. JIANG [2010], The forest hiding problem, *Discrete and Computational Geometry*, 1–24.

EDELSBRUNNER, H. [1987], 10, Springer Verlag.

EDELSBRUNNER, H., L.J. GUIBAS, AND M. SHARIR [1990], The complexity and construction of many faces in arrangements of lines and of segments, *Discrete and Computational Geometry* **5**, 161–196.

EDELSBRUNNER, H., AND E. WELZL [1986], On the maximal number of edges of many faces in an arrangement, *Journal of Combinatorial Theory, Series A* **41**, 159–166.

ESTIVILL-CASTRO, V., J. O'ROURKE, J. URRUTIA, AND D. XU [1995], Illumination of polygons with vertex lights, *Information Processing Letters* **56**, 9–13.

FRAPOLLI, F., B. HIRSBRUNNER, AND D. LALANNE [2007], Dynamic rules: Towards interactive games intelligence, 7, Citeseer, 29–32.

FULEK, R., A.F. HOLMSEN, AND J. PACH [2008], Intersecting convex sets by rays, *SCG '08: Proceedings of the twenty-fourth annual symposium on Computational geometry*, New York, NY, USA, ACM, 385–391.

GILBERT, P. [1972], Iterative methods for the three-dimensional reconstruction of an object from projections, *Journal of Theoretical Biology* **36**, 105–117.

GINDIKIN, S.G. [1994], *Applied problems of Radon transform*, American Mathematical Society.

GORDON, R., AND G.T. HERMAN [1974], Three-dimensional reconstruction from projections: a review of algorithms, *Int. Rev. Cytol* **38**.

GRAHAM, R.L., D.E. KNUTH, AND O. PATASHNIK [1994], *Concrete Mathematics: A Foundation for Computer Science*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

GUIBAS, L., AND M. SHARIR [1993], Combinatorics and algorithms of arrangements, *New trends in discrete and computational geometry* **10**, 9 – 36.

HADWIGER, H. [1960], Ungeloste probleme no. 38, *Elem. Math* **15**, 130–131.

HAN, J.Y. [2005], Low-cost multi-touch sensing through frustrated total internal reflection, *Proceedings of the 18th annual ACM symposium on User interface software and technology*, ACM, 115–118.

HARDY, G. H., AND E. M. WRIGHT [1979], *An Introduction to the Theory of Numbers*, Oxford University Press, London, UK, Fifth Edition.

HOLLEMANS, G., T. BERGMAN, V. BUIL, K. VAN GELDER, M. GROTEN, J. HOONHOUT, T. LASHINA, E. VAN LOENEN, AND S. VAN DE WIJDEVEN [2006], Entertaible: Multi-user multi-object concurrent input, *Adjunct Proceedings of UIST* **6**, 55–56.

HOLLEMANS, G., S. VAN DE WIJDEVEN, T. BERGMAN, AND E. VAN LOENEN [2006a], *Ambient Lyfestile: From Concept to Experience*, Chapter Entertaible, 80–83. BIS Publishers.

HOLLEMANS, G., S. VAN DE WIJDEVEN, T. BERGMAN, AND E. VAN LOENEN [2006b], Entertaible: The best of two gaming worlds, *MST News International newsletter on micro-nano integration, special issue on Fun and Recreation with Microsystems*, 9–12.

IZADI, S., S. HODGES, S. TAYLOR, D. ROSENFELD, N. VILLAR, A. BUTLER, AND J. WESTHUES [2008], Going beyond the display: a surface technology with an electronically switchable diffuser, *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST '08, New York, NY, USA, ACM, 269–278.

JAIN, A.K. [1989], *Fundamentals of digital image processing*, Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

JOVANOVIĆ, N. [2007], *Accuracy and Reliability of Object Detection on Entertaibles*, Technical report, Philips Research Laboratories.

JOVANOVIĆ, N., J. KORST, Z. ALEKSOVSKI, AND R. JOVANOVIĆ [2010a], Asymptotic bounds on minimum number of disks required to hide a disk, *Proceedings of the 4th International Conference on Advanced Engineering Computing and Applications in Sciences*.

JOVANOVIĆ, N., J. KORST, Z. ALEKSOVSKI, AND R. JOVANOVIĆ [2010b], Hiding in the crowd: Asymptotic bounds on minimum blocking sets, *Proceedings of the 26th European Workshop on Computational Geometry*, Dortmund, Germany, 197–200.

JOVANOVIĆ, N., J. KORST, R. CLOUT, V. PRONK, AND L. TOLHUIZEN [2009], Candle in the woods: Asymptotic bounds on minimum blocking sets, Aarhus, Denmark, 148–152, Proceedings of ACM Symposium on Computational Geometry.

JOVANOVIĆ, N., J. KORST, AND A.J.E.M. JANSSEN [2008], Minimum blocking sets of circles for a set of lines in the plane, *Proceedings of the 20th Canadian Conference on Computational Geometry*, Montreal, Canada, 91–94.

JOVANOVIĆ, N., J. KORST, AND V. PRONK [2009], Object detection in flatland, *Proceedings of the 3rd International Conference on Advanced Engineering Computing and Applications in Sciences*, Sliema, Malta.

KATSEVICH, A. [2004], An improved exact filtered backprojection algorithm for spiral computed tomography* 1, *Advances in Applied Mathematics* **32**, 681–697.

KRUSKAL, C.P. [2008], The orchard visibility problem and some variants, *Journal of Computer and System Sciences* **74**, 587–597.

LAURITSCH, G., AND W.H. HAERER [1998], Theoretical framework for filtered back projection in tomosynthesis, 3338, 1127.

LEVI, F.W. [1954], Ein geometrisches uberdeckungsproblem, *Archiv der Mathematik* **5**, 476–478.

MARTINI, H., AND V. SOLTAN [1999], Combinatorial problems on the illumination of convex bodies, *Aequationes Mathematicae*, 121–152.

MELISSEN, H. [1997], *Packing and Covering with Circles*, Ph.D. thesis, Utrecht University, Utrecht, The Netherlands.

MITCHELL, J. [2007], Dark points among disks, *Open Problems from the 2007 Fall Workshop in Computational Geometry, http://www. research. ibm. com/people/l/lenchner/fwcg2007/fwcg_open_problems. pdf.*

MUELLER, K., AND R. YAGEL [2000], Rapid 3-d cone-beam reconstruction with the simultaneous algebraic reconstruction technique (sart) using 2-d texture mapping hardware, *Medical Imaging, IEEE Transactions on* **19**, 1227–1237.

NIEVERGELT, J., AND F.P. PREPARATA [1982], Plane-sweep algorithms for intersecting geometric figures, *Communications of the ACM* **25**, 739–747.

O'ROURKE, J. [1987], *Art gallery theorems and algorithms*, Oxford University Press, USA.

O'ROURKE, J. [1998], *Computational Geometry in C*, Cambridge University Press.

O'ROURKE, J. [2004], *Handbook of discrete and computational geometry*, Chapter Visibility. Chapman & Hall/CRC.

O'ROURKE, J., T. SHERMER, AND I. STREINU [1995], Illuminating convex polygons with vertex floodlights, *Proceedings of the Seventh Canadian Con-*

ference on Computational Geometry, 151–156.

PACH, J., AND P.K. AGARWAL [1995], *Combinatorial Geometry*, Wiley-Interscience, New York.

PÓLYA, G. [1918], Zahlentheoretisches und wahrscheinlichkeitstheoretisches uber die sichtweite im walde, *Arch. Math. Phys* **27**, 135–142.

RAMM, A.G., AND A.I. KATSEVICH [1996], *The Radon transform and local tomography*, CRC.

ROGERS, C.A., AND C. ZONG [1997], Covering convex bodies by translates of convex bodies, *Mathematika* **44**, 215–218.

SACK, J.R., AND J. URRUTIA [2000], *Handbook of computational geometry*, North-Holland Publishing Co.

SCHMITT, P. [1993], Problems in discrete and combinatorial geometry, *Handbook of Convex Geometry*, 449–483.

SHAMOS, M.I., AND D. HOEY [1976], Geometric intersection problems, *Foundations of Computer Science, 1976., 17th Annual Symposium on*, 208–215.

SHAMOS, M. I. [1978], *Computational Geometry*, Ph.D. thesis, Yale University, New Haven, CT, USA.

SLANEY, M., AND A. KAK [1988], Principles of computerized tomographic imaging, *Philadelphia: SIAM*, 327.

SOLTAN, P. [1966], Connections between the problems of covering and illumination of convex bodies, **4**, 91–93.

SOLTAN, V. [1979], Illumination of convex figures in the plane, **2**, 29–32.

SOLTAN, V. [1995], Problem 76: Dark cloud problem, *Math. Semesterber* **42**, 81–82.

STEIGER, W., AND I. STREINU [1994], Positive and negative results on the floodlight problem, *Sixth Canadian Conference on Computational Geometry, Saskatoon*, 2–6.

STEIGER, W., AND I. STREINU [1998], Illumination by floodlights, *Computational Geometry* **10**, 57–70.

STEPHENSON, K. [2005], *Introduction to Circle Packing: The Theory of Discrete Analytic Functions*, Cambridge University Press.

SZABÓ, L. [1998], Recent results on illumination problems.

SZABO, L., AND Z. UJVARY-MENYHART [2002], Clouds of planar convex bodies, *Aequationes Mathematicae*, 292–302.

SZEMERÉDI, E., AND W.T. TROTTER [1983], Extremal problems in discrete geometry, *Combinatorica* **3**, 381–392.

TÓTH, L.F. [1954], Lagerungen in der ebene, auf der kugel und im raum, *Bull. Amer. Math. Soc. 60 (1954), 202-206. DOI: 10.1090/S0002-9904-1954-09805-1 PII: S* **2**, 09805–1.

TÓTH, L.F. [1959], Verdeckung einer kugel durch kugeln, *Publ. Math. Debre-*

cen **6**, 234–240.

TÓTH, L.F. [1977], Illumination of convex discs, *Acta Mathematica Hungarica* **29**, 355–360.

URRUTIA, J. [2000], Art gallery and illumination problems, *Handbook of computational geometry*, 973–1027.

URRUTIA, J., AND J. ZAKS [1989], Illuminating convex sets, *University of Ottawa Computer Science TR-89-31*.

VALENTINE, FA [1970], Visible shorelines, *The American Mathematical Monthly* **77**, 146–152.

WOBBROCK, J.O., M.R. MORRIS, AND A.D. WILSON [2009], User-defined gestures for surface computing, *Proceedings of the 27th international conference on Human factors in computing systems*, CHI ’09, New York, NY, USA, ACM, 1083–1092.

ZONG, C. [1997], A problem of blocking light rays, *Geometriae Dedicata* **67**, 117–128.

Symbol Index

Detection Algorithms

e	emitter	15
s	sensor	15
n	number of emitters (sensors)	16
Γ	detection area	16
E	set of emitters	16
S	set of sensors	16
r	distance between an emitter and its neighboring sensor	16
(e_i, s_j)	line of sight	16
$B = [b_{i,j}]$	n x n visibility matrix	16
ω	shadow wedge	17
m_i	number of wedges of emitter e_i	17
o	object	18
$(z_0, \dots, z_{n-1})$	wedge vector	19
$\mathcal{C}$	n-wedge intersection	19
P_i	polygon	21
υ	vertex of a polygon	21
$\mathcal{Q}$	set of points	28
G	grid	30

Limitations of in-plane object detection

$\mathcal{D}$	a set of non-overlapping unit disks	36
N	the number of disks	36
ℓ, ℓ', ℓ_1	lines	36
p, q, p', q'	points	36
δ	a unit disk	36
ς	a shadow region	36
$\mathcal{H}(\mathcal{D})$	the convex hull of the disks in $\mathcal{D}$	36
$\mathcal{S}$	the set of all shadow regions	36
$\mathcal{T}$	the set of defining (tangent) lines	43
t, t', t_1	tangent lines	43

$(p_{\delta_1}, p_{\delta_2})$	a line segment	43
$\rho_{\delta_1}, \rho_{\delta_2}$	rays	43
$\mathcal{L}$	the set of all lines containing only light points	44
$\mathcal{U}$	the union of all light corridors	46
$\delta_1, \ldots, \delta_N$	the disks of the left column	49
$\delta'_1, \ldots, \delta'_N$	the disks of the right column	49
$O_1, \ldots, O_N$	the centers of disks in the left column	49
$O'_1, \ldots, O'_N$	the centers of disks in the right column	49
ℓ_{left}	line connecting the centers of the disks in the left column	49
ℓ_{right}	line connecting the centers of the disks in the left column	49
h	the distance between the columns	49
d	the distance between two neighboring disks in one column	49
w	the width of the area where no two corridors intersect	50
ℓ_s	the vertical line containing splitting points furthest from ℓ_{left}	51
ℓ_m	the vertical line containing meeting points closest to ℓ_{right}	51
$\bar{h}_s$	the distance between ℓ_s and ℓ_{left}	51
$\bar{h}_m$	the distance between ℓ_m and ℓ_{left}	51
C_j	the light corridor that begins between disks δ_j and δ_{j+1}	52

Visibility Problems I

d	the minimum distance between two disks	60
ρ	a ray	60
p	a given point	61
$\mathcal{R}_p$	the set of all rays emanating from the given point p	61
$\mathcal{L}_p$	the set of all lines containing the given point p	61
$\mathcal{D}$	a (blocking) set of disks	61
N_d	the minimum cardinality of a d-apart blocking set $\mathcal{D}$	63
$H_1, \ldots, H_d$	hexagons of side length $d, 2d, \ldots, d^2$ containing the disks	63
$\mathcal{F}_n$	the Farey sequence of order n	67
$\mathcal{I}_{d-1}$	the set of intervals related to $H_1, H_2, \ldots, H_{d-1}$	68
$\mathcal{I}$	all intervals from $\mathcal{I}_{d-1}$ ordered by increasing midpoint	68
I, I'	intervals in $\mathcal{I}_{d-1}$	68
$\varphi(n)$	Euler's totient function	71
$R_1, R_2, \ldots$	hexagonal rings	72
$\beta(\delta)$	the blocking wedge of the disk δ	73
α_j	the maximum blocking wedge of a disk in ring R_j	75
μ_j	the maximum number of disks within the union of $R_1, \ldots, R_j$	75

Visibility Problems II

U	the given disk	82
$\mathcal{L}_U$	the set of all lines that intersect U	82
$\mathcal{R}_U$	the set of all rays that emanate from U	93
$\mathcal{L}(t,t')$	a bundle of lines between tangent lines t and t'	84
c	a circle of large radius concentric to U	95
r_c	the radius of the circle c	95
η	the number of disks such that $N = 6\eta$	95
$\mathcal{D}_2$	a 2-apart blocking set	95
$c_1, \ldots, c_k$	concentric circular rings of radii $d, 2d, \ldots, kd$	97
k	the number of circular rings	97
η_j	the number of disks in one sixth of the j-th ring	98
$\mathcal{A}_\eta$	the sequence of integers corresponding to the number of disks in the rings	100
$\mathcal{A}_\eta^*$	the sequence of integers where k is minimal	100
τ	the table corresponding to the ordering of disks	100

Summary

In-plane Object Detection:
Detection Algorithms and Visibility Problems

A large number of devices today incorporate some form of detection of objects and people in a given environment. Various detection technologies have been developed over the years, as a response to many different demands. The devices such as video surveillance systems, scanners, touch screens and various systems for tracking people and objects in space, detect objects using camera videos and/or measurements gathered by sensors.

To enable simultaneous detection of multiple objects on table-top interactive devices designed to support games that combine the social attractiveness of traditional board games with the interactivity of computer games, an in-plane detection technology that uses LEDs and sensors was developed by Philips. The presence of objects on the table results in blocking light emitted by the LEDs for some of the sensors. This information can be used to determine the position and shape of objects such as game pieces or fingers on the table. If the detection process is performed fast enough, then moving objects can be tracked, for instance, to recognize gestures made by fingers. This detection technique gives rise to many interesting geometric problems, such as developing efficient detection algorithms. In addition, due to occlusion created by the multiple objects placed on the table, some visibility problems can occur in the process of detecting objects.

We present detection algorithms that use the sensor data as an input and provide an approximation on the geometry of objects as an output. We discuss the advantages and disadvantages of the presented algorithms and analyze their worst-case time complexity as a function of the number of LEDs and sensors.

In addition, the maximum level of the accuracy of detecting circular objects that can be achieved has been investigated. To investigate this maximum level of accuracy we assume infinitely many LEDs and sensors in a frame surrounding the

objects. We present and discuss a worst-case optimal algorithm that determines the output that a detection algorithm would provide in this case.

Several visibility problems have been explored that relate to occlusion, an intrinsic shortcoming of the detection technique. Among many visibility problems that can be identified, the focus was on five problems related to either falsely detecting a non-existing circular object or detecting multiple objects as one. These problems occur when multiple objects positioned in the detection area block all of the lines of sight between LEDs and sensors that cross some area that is not occupied by an object. In this thesis, we focused on exploring the worst-case scenarios, in other words, finding the minimum number of identical circular objects that can cause one such visibility problem to occur in relation to the distance between the objects. We have proved that this number is quadratic in the minimum mutual distance between the objects. This result can be used in practice, for example, to adapt the layout of game boards such that these visibility problems can be avoided.

Curriculum Vitae

Nataša Jovanović was born on 01-12-1980 in Lebane, Serbia. After finishing high school in 1999 at the Gimnazija ”Stojan Ljubic” in Lebane, she studied Applied Informatics at the Ss. Cyril and Methodius University in Skopje, Macedonia. In 2006 she joined the Media Interaction group of Philips Research in Eindhoven for a nine-month internship. In 2007 she graduated at the Institute of Informatics, Faculty of Natural Sciences and Mathematics, on the topic *Accuracy and Reliability of Object Detection on Entertaibles*. From 2007 until 2011 she started a PhD project at the Eindhoven University of Technology of which the results are presented in this dissertation. The work has been carried out in the User Experiences group and the Human Interaction & Experiences group at the Philips Research Laboratories in Eindhoven. In April 2011 she joined the System Architecture and Networking group at the Eindhoven University of Technology as a post-doctoral researcher.