Skeleton based visual pattern recognition: applications to tabletop interaction

By

Andoni Beristain Iraola

http://www.ehu.es/ccwintco

Dissertation presented to the Department of Computer Science
and Artificial Intelligence in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

PhD Advisor:

Prof. Manuel Graña Romay

At
The University of the Basque Country
Donostia - San Sebastian
2009

Acknowledgments

I wish to thank Prof. Manuel Graña, my PhD supervisor, for his advice, help, support, and patience. This would not have been possible without him. My sincere thanks also go to the Innovae Vision staff, where I started working as Software Engineer at the research department, and who encouraged me to do my PhD. studies. And also to the Grupo de Inteligencia Computacional (GIC) research group from the Computer Science faculty of the University of the Basque Country, who help to each other contributing to the benefit of the whole group, and the improvement and growing of the individuals which constitute it. I want to specially thank the oportunity I have had to be part of this group. I wish there was a way to express my gratitude to my parents, who have always encouraged me and supported me in all my achievements, and also to my brother. And in the same way to my family and friends for their support and encourage. I want to thank to Prof. W. Hesselink for the support with his Medial Axis computation based in the Distance Transform, and in the same way to Bai and Latecki, for their help to understand some details of their research works, and permission to upload improved versions of their source code to our research group web page. And finally to the Industry department of the Basque Government for his support to this research work through an Ikertu grant for the achievent of the PhD degree in a Company, Innovae Vision.

– *Eskerrik asko bihotz-bihotzez denoi zuen laguntza eta animoengatik* –

Reconocimiento visual de patrones basado en esqueletos: aplicaciones a la interacción en mesas computacionales por

Andoni Beristain Iraola

Submitted to the Department of Computer Science and Artificial Intelligence on November 12, 2009, in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Abstract

El contexto de aplicación de esta tesis es la interacción en mesas computacionales (tabletops) mediante gestos manuales reconocidos visualmente. Se ha definido un lenguaje de interacción conveniente para este tipo de sistemas y aplicaciones. Los gestos de la mano son capturados con una cámara óptica, la imagen es procesada hasta obtener el esqueleto de la mano. El reconocimiento de los gestos se realiza mediante técnicas de reconocimiento de patrones especialmente diseñadas para trabajar sobre representaciones dadas por información extraída del esqueleto de la figura en la imagen. La tesis presenta una aportación teórica y computacional en el proceso de cálculo del esqueleto que consigue mayor estabilidad que otros algoritmos en la literatura. Dicha aportación parte del método de Voronoi de construcción de esqueletos y en base a resultados que permiten una poda eficiente y estable, obtiene esqueletos estables en tiempo real (30fps) que hacen posible la aplicación de esta técnica en el contexto de las mesas computacionales.

Skeleton based visual pattern recognition: applications to tabletop interaction by

Andoni Beristain Iraola

Submitted to the Department of Computer Science and Artificial Intelligence on November 12, 2009, in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Abstract

The practical context of this thesis work is natural interaction in Interactive Multimedia Tabletops using visually recognized hand gestures. An interaction language has been defined for the specific characteristics of this kind of systems and applications. Hand gestures are captured using an optical video camera, then the image is processed to obtain the hand shape skeleton. Gesture recognition is performed using pattern matching techniques specifically designed to work on the information provided by skeletal representations of the shapes in the image. This thesis work introduces a theoretical and computational contribution to the skeleton computation procedure which achieves better stability than other algorithms in the literature. This contribution starts from Voronoi skeleton computation method and based in results which permit an efficient and robust pruning, obtains stable skeletons in realtime (60fps), which make possible the application of this technique in the context of Interactive Multimedia Tabletops.

Contents

List of Figures

List of Tables

"Start by doing what's necessary, then do what's possible, and suddenly you are doing the impossible."

Saint Francis of Assisi

Chapter 1

Introduction

This chapter is a general introduction to the PhD Thesis Report. Section 1.1 introduces the context of this PhD Thesis Report, section 1.2 presents the objectives, section 1.3 describes the contributions, and section 1.4 presents the structure of this PhD Thesis Report.

1.1 Context and Motivation Background

This PhD research work has been carried out in a collaborative company-university context. The PhD candidate began working at the research department of the Innovae Vision company. And due to his personal interest and the encouragement of the company, started his PhD studies obtaining a special grant from the *Departamento de Industria* of the *Gobierno Vasco*, which allows to pursue PhD research sharing time at the industry and the University. Innovae Vision is a spin-off company created by members of the Grupo de Inteligencia Computacional / Computational Intelligence Group (GIC) from the Computer Science Faculty of the University of the Basque Country.

The PhD candidate started his PhD research work on the field of Affective Computing, specifically on real-time facial expression recognition for emotional state inference in the company Innovae Vision [12], and his participation in the framework applied research project REKEMOZIO[13], funded by the *Departamento de Industria*. As a result of this research work, we published the paper [14] in the New Mathematics and Natural Computing journal. Unfortunately, after two years of work, the company also stopped the research on automatic visual emotion recognition, for strategical reasons.

The company become interested in an innovative natural interaction paradigm, called Interactive Multimedia Tabletops (IMT), or tabletops. An IMT is an interface to a underlying computer system, dressed up like a table-shaped furniture. The IMT displays a graphical user interface (GUI) on its horizontal surface, and permits interacting with its elements using natural interaction methods like voice or hand gestures. Natural interaction is a hot research topics for Innovae Vision, and all parties, Innovae Vision, the PhD supervisor and the PhD candidate agreed to shift the PhD research focus to the field of IMT. Among all the existing natural interaction methods, Innovae Vision was specially interested in both gestural interaction and multitouch interaction, which are performed using the user's hands, for its relationship with other business interests of the company. This combination of this interests took the shape of a research project in Innovae Vision aiming to develop a fully functional IMT prototype. The PhD candidate assumed the responsibility to design it and develop an efficient hand gesture recognition algorithm tailored for that IMT to recognize a set of gestures defined in the project description document. There is a glimpse of the IMT design in section 2.8. Therefore the content of this PhD dissertation report are the foundations and development of an efficient visual hand gesture recognition system based on an skeletonization approach.

Nowadays the use of multitouch surfaces is becoming a *de facto* standard for interfaces, including tabletops and mobile phones. From a scientific point of view, the recognition of patterns in multitouch devices does not pose an interesting challenge: Image segmentation is almost trivial under the appropriate settings, and the patterns of the gestures are not difficult to recognize and interpret. Manufactures of multitouch surfaces provide APIs including recognition of a well known set of static or dynamics patterns of touch. The visual recognition of had gestures can be combined with multitouch information, and has some advantages by itself. Being a non touching method can be better accepted in some cultural settings and can be more natural for some tasks.

Therefore the PhD candidate focused his research on hand gesture recognition in the context of IMT. Hand gestures are based in hand postures and motions, including posture changes in some cases, but there is not hardly any change in the appearance of the hand. Consequently, a shape based hand gesture recognition approach was chosen. The hand is a biological amorphous object, with an invariant interpersonal structure, but highly variable in its size and details, and it can adopt a large set of postures. This, joined to the problem of segmenting the hand from the user's arm for hand gesture recognition.

Skeletons have the appealing of providing an intuitive simplification and representation of the objects. It does agree with our intuitive approach to represent objects by sketches and strokes. Therefore they have received a lot of attention in the literature of pattern recognition, computational geometry and image processing. There have been attempts to produce efficient and stable skeletonization algorithms from many different approaches, rooted in diverse computational foundations. Therefore they were selected as the basic approach for object representation.

1.2 Objectives

The long term goal of the research project where the present PhD work is embedded is to develop a fully functional IMT combining natural hand and multitouch gestures. This long term effort is not covered extensively in this report. Here we will recall the objectives that were set at the beginning or during the evolution of the PhD work closely related to the contents of the dissertation report. Other works which fall well outside the scope of this memory are not mentioned.

- Produce an state of the art review of the IMT field, including the issues of interface definition and interaction. This review would be used in the IMT design at the company, as well as to identify business opportunities and desired target users. The interest of the company was quickly focused into interactions for casual users.

- Define an interaction language appropriate for some general tasks.

- Produce a review of hand gesture approaches, that would lead to the appropriate selection of the image data representation. The data representation chosen from the very early stages of the work was the skeleton, conditioning all the other aspects of the work.

- Produce a review of skeletonization algorithms, and the pattern recognition approaches based on the skeleton representations.

- Propose an efficient and stable skeletonization algorithm. Efficiency meaning the ability to perform in real time. Stability meaning that small shape changes produce small skeleton variations.

- Validate the skeletons obtained from this algorithm as appropriate representations for pattern recognition, compared with state of the art skeletonization algorithms, specially for hand gesture recognition.

- Produce a real time implementation of the algorithm, suitable for integration in the IMT software.

1.3 Contributions

The main contributions of this PhD Thesis work are:

- The creation of a hand gesture image database, publicly available through the GIC research group web page. This image database served us to validate the skeletonization algorithm under several views, from the measure of skeleton's complexity to the realization of classification experiments.

- A review of the state of the art on tabletops. Being a very quickly growing field, we hope that some of the views reflected in this review will still be valid in the future, although the details will become obsolete quickly.

- A design proposal for an IMT using the combination of hand gesture and multitouch interaction for user interaction.

- The proposal, formal justification and implementation (real time) of an efficient and stable skeleton computation and pruning algorithm. This is the central contribution of this thesis, and it is developed in chapter 4.

- The development of several software prototypes for the validation of our algorithm, some of them also available at our research group web page.

- The realization of computational experiments showing the improvement of our algorithm over other state of the art approaches.

1.4 Structure of the PhD Thesis Report

The remaining of the PhD report is structured as follows.

1. Chapter 2 reviews Interactive Multimedia Tabletops, including some design insights for IMT oriented to casual users, and the description of a prototypical design of a tabletop which combines the use of hand gestures and multitouch interaction techniques.

2. Chapter 3 reviews binary image skeletons in 2D. This review includes skeleton computation techniques, skeleton regularization procedures and shape matching algorithms using skeletal representations of shapes.

3. Chapter 4 presents our skeleton computation and pruning algorithm, using the Voronoi Tessellation.

4. Chapter 5 validates our skeletonization procedure in terms of computational efficiency, robustness to noise in the shape boundary and shape recognition performance.

5. Chapter 6 summarizes the conclusions of the PhD research results.

6. Appendix A describes hardware and techniques used to develop multitouch displays.

7. Appendix B provides with the mathematical background required for the comprehension of some sections in the report.

Chapter 2

Interactive Multimedia Tabletop (IMT)

The Interactive Multimedia Tabletops (IMT) are the application and physical environment of this PhD Thesis works. The motivation for the ensuing chapters was the development of an innovative design for a tabletop to be built and distributed by the local industry. In this chapter we perform a review of the state of the art of tabletop devices at the time of working on this thesis. Section 2.1 gives some background motivation for tabletops. Section 2.2 reviews some previous works in the design of such systems. Section 2.3 describes cooperative and collective work in tabletops. Section 2.4 presents the interaction issues in IMT. Section 2.5 introduces specific user interfaces for IMT. Section 2.6 focuses in multitouch IMT. Section 2.7 introduces some of the issues in IMT oriented to casual users. After all this review work, a design proposal is presented in section 2.8. And the chapter finishes with some conclusions in section 2.9.

2.1 Introduction and motivation

Computing devices and digital communication means are becoming ubiquitous [15, 16, 17, 18] part of our daily life. With technological companies launching new products everyday, competitiveness among them is producing devices with increasing and innovative features, easier to use and more visually appealing. Despite the big increment in computational power, Human Computer Interaction (HCI) methods have remained almost unalterable. Keyboard and mouse are still the most common input devices. Other interaction devices are the optical pen, the Tablet PC, which is a combination of laptop and digitizing tablet (using a stylus), some digital version of a paper sheet and a pen goes a step further in the way to achieve a natural interaction with machines. Conventional configuration of CPU box, screen, keyboard and mouse has not changed in the last twenty years. The tabletop configuration is proposed as a step with much improved ergonomics for certain applications. Two main interaction paradigms are the promising killing applications for tabletops:

- The management of multimedia data, including Content Based Information Retrieval (CBIR) [19, 20, 21], organization and visualization and editing.
- The collective work by groups of people, sharing and manipulating multimedia information.

It seems that the tabletop metaphor may be much more adequate for the management of big amounts of heterogeneous information items, which can be displayed, edited and shared much better on a big horizontal surface than on vertical surfaces. Two main categories of users can be defined:

- Casual users: the system must offer simple functions for any kind of users. The systems are designed to be placed in public locations of great people flow, with the purpose of offering some kind of service, like the location of points of interest, advertising, or showing the available resources in a specific place. In this kind of scenarios robust and simple interaction is more important than sophistication. Target locations for these systems are art and science museums. Section 2.7 of this document is devoted to them.
- Expert user: for them, the IMT is conceived for a more intensive use which would take advantage of its full interaction potential. In this kind of systems it is common to have an adaptation and learning stage, though the interaction methods are purportedly natural, to be able to exploit all the possibilities offered by the IMT. Examples are systems to help in the design step of software projects, civil architecture or advertising campaigns.

2.1.1 Definition

We comment on the meaning of each of the terms in the expression *Interactive Multimedia Tabletop* (IMT):

- Tabletop: An IMT is a table shaped furniture, a horizontal surface elevated to a certain height by means of some kind of stand. The horizontal surface of the IMT displays a Graphical User Interface (GUI) or Tangible User Interface (TUI) (described in section 2.5.2). Since it is table shaped it can be accessed all around it according to some ergonomic design criteria. A table offers a natural means to share information items in group interaction, allowing parallel multiuser interaction.

- Interactive: it must permit a rich, fluent and natural interaction between the users and the IMT, through hand gestures, physical objects and voice commands minimizing the use of additional interaction devices. Some systems employ only one interaction method while others use a combination of them.

- Multimedia: the users will be naturally interested in visualizing and manipulating multimedia information, like videos, pictures, 3D, animations and sound. A priority goal is to take advantage of the natural interaction methods to make the management of these kind of data easy and appealing to the user.

This kind of interfaces/systems are still under research and development phase, but interest on them has increased recently with big companies like Microsoft or Apple working on systems with multitouch interface, described later on. Several examples of IMT systems are show in figure 2.1.

2.1.2 Advantages of Using Tabletops

An IMT allows users to interact face to face, acting as a mediator for communication and sharing data items, promoting a cooperative working environment for the completion of a common task. There are examples of tabletops [22, 23] where collective work is carried out on the tabletop itself, even when work teams are geographically distant through network communication using other IMT and/or computers [24, 25, 26], thus, IMT are not restricted to small groups nor isolated environments.

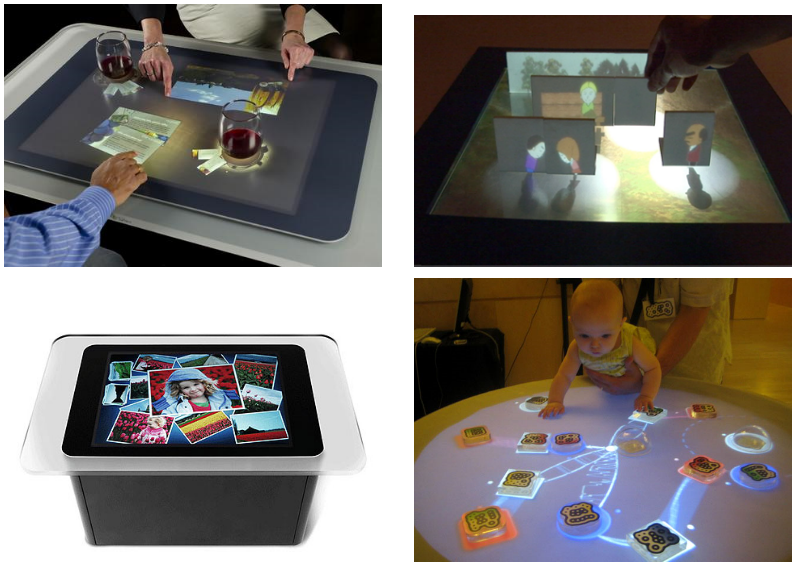


Figure 2.1: Some tabletop systems

The use of natural interaction methods allows complex operations to be carried out by means of actions which are easy to learn and execute for common people, because they find them alike to other gestures of their everyday life, though the meaning of these actions could be very complex and difficult to specify in other ways. An example of this kind of interaction is a multi-touch drawing application, where a user can use his fingers or hands to draw or paint. Voice interaction using natural language is another paradigm.

The external appearance of an IMT is quite different from the aspect of an ordinary computer. This design makes ordinary people forget their prejudices against computers, since the IMT is not perceived as a foreign object but as an object already culturally integrated. Under this view, an IMT goes an step further to achieve a more friendly human-computer interaction. Other characteristics that also help to this purpose are the aesthetic design of the IMT, the graphical interface and the interaction methods used.

When the interaction surface is vertical, the systems are called Interactive Multimedia Walls or Interactive Multimedia Boards [27, 28]. In contrast to tabletops, group interaction can not be performed face to face. Moreover, it produces more fatigue on the user than the IMT. On the other hand, a wall permits to step back to acquire a more global view, which is not possible with tabletops.

2.2 Review

Research on Interactive Multimedia Tabletops is quite recent, for several reasons: (a) because multimedia data management was not generalized until the last half of the 90s[1] decade, (b) increased computational complexity implied by the need of natural interaction, and (c) the recent appearance of key technologies to support it, e.g. multitouch surfaces.

2.2.1 Early systems

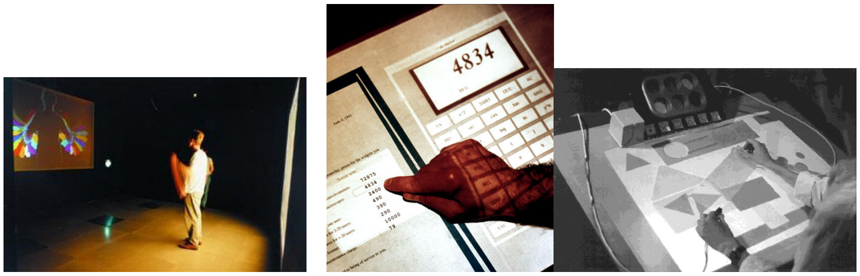
An example of systems designed in the 1980s decade is the VIDEOPLACE [29] illustrated in figure 2.2a, which consists of a vertical screen where objects were projected, and the silhouette of the users was captured using a camera. The horizontal version of VIDEOPLACE was called VIDEODESK.

Developed in the early 90s, DigitalDesk [30] shown in figure 2.2b consists of a physical desk with a camera and projector over it. The system projects images on the desk and was able to detect real objects, perform OCR, and recognize some hand gestures realized over the desk. Its design included the possibility to perform network interactions. The VideoDraw and Video Whiteboard [31, 32] allowed collaborative drawing among different users using virtual blackboards. The TeamWorkStation [33, 34] combined the concepts of real and virtual desk. The DOLPHIN system [35] was composed of a vertical blackboard and an optical pencil to interact with it, which is able to interpret the strokes made to interact with the virtual objects in the blackboard graphical interface. Another system that made use of the stylus as interaction method was described in [36]. Early use of physical actuators for the interaction was demonstrated in the Active Desk [37], where "bricks" were used in a painting application.

2.2.2 Current Trends

Most of the recent IMT designs involve multitouch interaction. We discuss some of them in section 2.6, and appendix A presents some of the designs for completeness of this report. The most common multitouch techniques are based on Frustrated Total Internal Reflection (FTIR) and Capacitive Effects. FTIR are usually used in combination with one or more video projectors for

[1] http://en.wikipedia.org/wiki/Computer_graphics



(a) VIDEOPLACE in use. (b) DigitalDesk. (c) GraspDraw application on the Active Desk tabletop.

Figure 2.2: Early gestural interaction systems

display, while the Capacitive Effect is mainly used in systems whose displays are LCD screens.

Hand gesture recognition, which one of the subjects of this thesis, is scarcely proposed as a mean of interaction in the latest systems. On the contrary, physical actuators are used by many systems. Moreover, in some systems the tabletop itself can even move the actuator [38, 39].

Most of the latest research works about IMT focus their efforts on GUI improvement and the adaptation of current conventional software to the IMT paradigm, aiming towards simultaneous interaction between multiple users, locally and/or remotely [40, 41, 42, 24]. These works are not innovative on the technical aspects of the interaction method chosen.

Voice interaction is limited to a handful of research works. An interesting research work, [43] is an example of combination of voice and multitouch on an IMT, as well as an additional vertical screen that can be used to see remote users working in another connected IMT, by means of a camera integrated in the vertical screen.

2.2.3 Future avenues for research

Tabletop interaction was been shown in science fiction films such as Minority Report (2002), Iron Man (2008) or The Island (2005). Though some of the interactions shown lack scientific and technical rigor, they have the quality of proof-of-concept illustrations. The following are some key technologies that may help the advance of IMT to the point of science fiction performance:

- 3D cameras that provide 3D information about the scene. The classical approach is the stereo vision rigs. Nowadays there are affordable Time Of Flight (TOF) cameras based on modulated infrared light. Both could be used for IMT design [44]. The main inconvenience of stereo rigs is that they need to factor out the images projected on the screen, and their sensitivity to illumination conditions. The TOF cameras are robust against these conditions, but their current realizations are very noisy and low-resolution.

- 3D displays for the realistic rendering of 3D data [45]. Some current technical solutions are quickly spinning LCD screen, infrared lasers that excite points in a gas, or transform air in light emitting plasma, and controlled falling water droplets to scatter projector light in a 3D volume. These display techniques do not allow interaction with the user/observer. Some of them are rather experimental at the moment.

- Large scale multitouch surfaces. The problem to obtain these kind of devices is that their detection resolution decreases proportionally to the surface size, because the underlying recognition devices are limited in resolution. The Frustrated Total Internal Reflection (FTIR) resolution is limited by the resolution of the camera used. The Capacitive Effect is limited by the surface's internal wire density. Large LCD displays have the same resolution problem, which can be reduced by the composition of mosaics.

- Enhanced gesture recognition. The interaction through and body gestures is a natural communication channel for humans. Real time robust gesture recognition is still an open problem despite advances in computer vision.

- Pressure sensing surfaces. A limitation of current multitouch devices is that the information provided by the touch action is binary. Touch pressure intensity may be an important information source, for emulating some actions like drawing with a pencil, modeling with clay, or using the accelerator and brake pedals in a car. It could also be used as a way to provide interaction with a depth dimension. A new kind of devices called IMPAD (Inexpensive Multi-Touch Pressure Acquisition Devices) may become the next standard for multitouch surfaces.

2.3 Collective cooperative work and IMT

The IMT provide an enhanced support for collective work and this kind of utility is a driven force for their development. Some requirements on IMT design directed towards collective work [23] are the following

- Provide support for interpersonal interaction.
- Allow seamless transition between different activities: type of activity (i.e. writing, drawing), span of the activity (i.e. personal versus collective).
- Seamless transition between the the table and its environment: Importing and exporting data, recognition of foreign devices (i.e. mobile phone) and their integration in the computing environment.
- Shared access to physical and virtual objects should be available.
- Cultural restrictions regarding interpersonal relations.
- Remote collaboration should be available, including audiovisual information.
- Adaptation of current conventional applications to the IMT.

Some of the issues raised by the collective work interaction over the IMT are discussed in [24, 46].

Sharing The issue at hand is the user coordination inside the workspace, the management of the access rights to shared resources and their transference among users. The coupling between users refers to the degree of dependence (mutual or asymmetric) between users for the realization of their tasks. Tightly coupled tasks imply that user actions are coordinated and fluent, because each of the users knows the purposes and intentions of the other user, reducing and avoiding collisions and interferences. Loosely coupled tasks imply that users behave more like isolated individuals, though there are still some kind of collective actions. Sharing [47] is the ability to change the accessibility of a digital object dynamically, switching between personal and public access. On the other hand, the abuse of private spaces is an antisocial behavior, causing contempt, and hampering collective work .

Territoriality The issue at hand is how the the workspace (IMT surface) is distributed among users to coordinate their activities. Workspace segmentation occurs with almost zero communication between users. Usually, it is a fluid and quasi automatic process. The works in [48, 46] identify three kinds of territory:

Personal territory: It is a patch of IMT surface close to each user, and it is used for the manipulation and storage of personal data and to carry out individual tasks.

Group territory: Patches of IMT surface reachable for every user, used for sharing, exchanging or storage of shared objects. It can be defined as the remaining space after the claiming of all personal spaces.

Storage territory: They can be either temporary or static, e.g., a pile of pictures. Storage territories can be owned by an individual or a group, depending on their location.

Collisions A collision happens when two or more users interfere with each other while interacting with the IMT. Direct interaction methods are prone to physical collisions while indirect interaction methods are not [49]. When the design emphasis of the IMT is on natural interaction, collisions are unavoidable. Thus, a design goal must be reducing the number of collision and their impact on working dynamics. However, there is not any published research work looking for solutions to this issue.

Privacy Privacy goes beyond sharing. The main goal is not only to restrict access to another users' private objects, but also to prevent users to see any personal data displayed on them [50].

2.4 Interaction in IMT

The interaction methods available for an IMT are a key design element, which will determine usability aspects of the IMT such as the level of expertise required of the user. Naturalness is a central design idea. Natural interaction close to the natural social interaction is one goal of IMT designs.

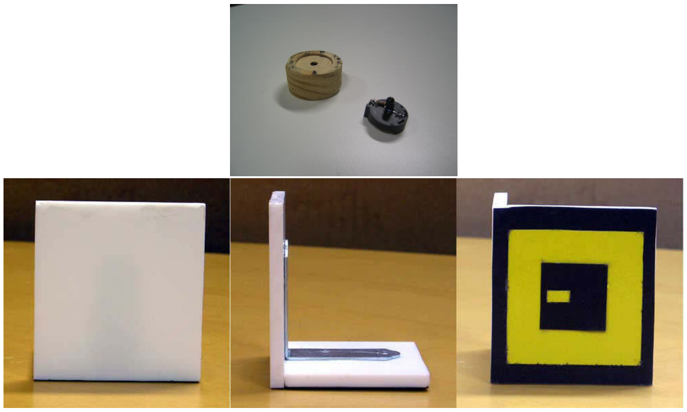


Figure 2.3: Tabletop actuator examples.

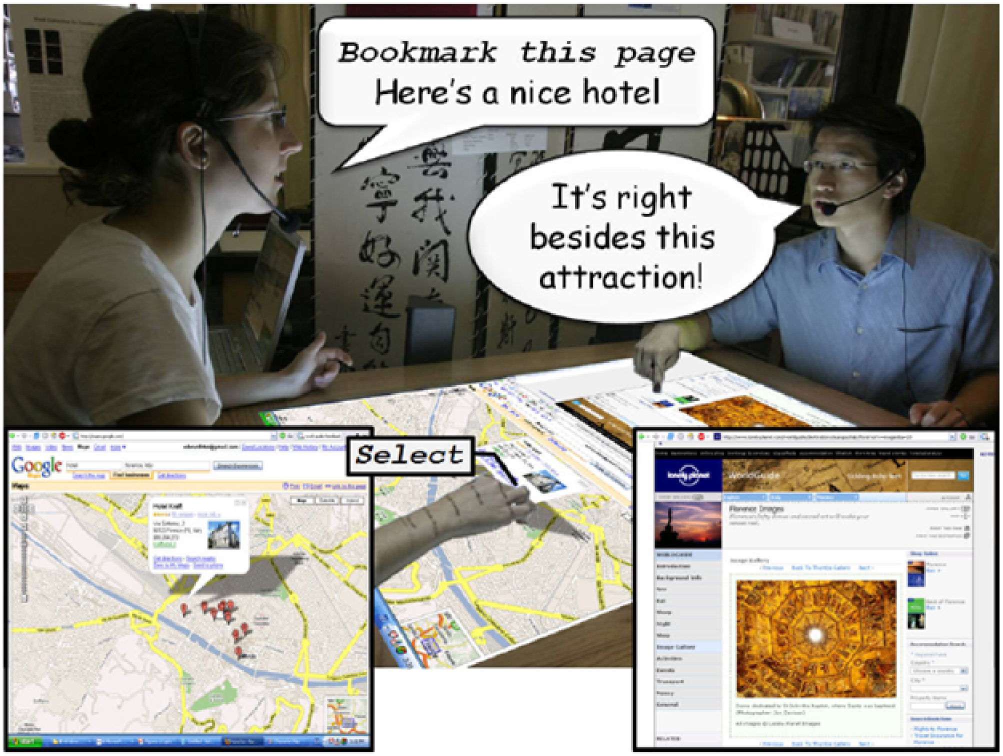


Figure 2.4: IMT using voice and multitouch interaction based in the DiamondTouch multitouch surface by MERL.

1. Hand Gestures: This interaction method uses hands for direct interaction with objects in the IMT workspace. Two different hand gestures are distinguished, conventional hand gestures an multitouch on the tabletop surface.

   (a) Multitouch: In this kind of interaction method, the user touches the horizontal surface of the tabletop to interact with it. The main difference with conventional touch systems is that the system is able to recognize several simultaneous touches in different location on the tabletop. And some systems are even able to recognize which user touched the surface and where.

   (b) Natural 2D gestures: A special case of gesture recognition [51, 52, 53], for IMT interaction, gestures are limited to the space over the tabletop and also to the tabletop height, user height and possible recognition restrictions. Although gestures are performed over the tabletop, depth information is not considered, therefore they are performed on a bi-dimensional plane for the IMT.

2. Physical actuators: Interaction by means of physical objects also called physical actuators, consists on the use of artificial objects which position, relative motion and/or acceleration can be obtained automatically [54, 39, 55, 56]. This interaction method is based on the rotation and translation of the actuators on the IMT surface [57]. Figure 2.3 shows several examples of actuators.

3. Voice: There is not any known IMT which uses voice interaction only. Voice is always combined with hand gesture or multitouch interaction [43, 58, 59]. In this kind of multimodal interfaces the voice indicates the action to perform and the hand gesture indicates its context, the objects and place where the action takes place. Figure 2.4 illustrates this interaction method over IMT. Voice interaction can be based on natural language or *ad-hoc* for the IMT context. In collective work environments conflicts may appear trying to distinguish when users are talking between them or to the IMT. In practice, users wear some kind of microphones to avoid the cocktail effect and reduce the effect of environmental acoustic noise. In this way, system's robustness is improved, but the naturalness of the interaction is reduced, forcing the user to wear a device. And this device is also a limiting factor for the

maximum number of simultaneous users on the IMT. A nice tutorial on voice recognition is [60] and [61] gives a list of available programming resources.

4. Direct versus Indirect Interaction: We say that interaction is direct when the object is manipulated acting directly on it [62]. Each level of mediating tool required for the object manipulation adds an abstraction level and makes the interaction less direct, more indirect. We consider that interaction is more natural when it is more direct. In [49] some differences between direct and indirect input devices are described.

   (a) Direct Input Devices. They promote a fluid and natural language. Since this is the way in which ordinary objects are usually manipulated, it is a natural procedure. They support group improving understanding of the intention and actions of the user manipulating the object by the rest of the group. Also, anticipating the result to an action is immediate. On the negative side, this kind of devices induce physical tiredness on the user. Direct interaction limits the possibility to reach an object by the device's constraints. Commonplace gestures can be distracting. The input device can produce occlusions on the display. Users can collide physically in the workspace, when trying to access or manipulate it simultaneously. Finally, direct input devices can be seen as intrusive in the territory of another user.

   (b) Indirect Input Devices. Distant objects are easily accessible. The effort required to use them is low, compared to the effects produced on the objects. Occlusion in the workspace is limited or even discarded. The drawbacks are the limitation of the gestures used, the interaction is less intuitive and the user needs training, there is a loss of spontaneous collaboration, multiple representations of the same thing could also produce distraction and confusion.

5. Interaction Languages: Interaction language types define the association between words or gestures and their meaning. They are defined by a lexicon, a grammar and a semantic specification.

   (a) Natural interaction languages try to emulate real life interaction with virtual objects, that is, virtual objects are displayed and manipulated much as their physical counterpart. For example, in hand gesture interaction, virtual objects can be picked up, moved or even thrown. In voice interaction the emphasis would be in the use of unconstrained natural language. The goal is that the user needs no training to start interacting with the IMT. They are limited by the sensorization of the IMT and the problems of dealing with ambiguity.

(b) Symbolic Languages: define completely new languages based on some particular criteria, like efficiency, without any intended correspondence with the physical world. They require a learning process undertaken by the user. But learning cost pays off in terms of efficiency and lack of ambiguities.

(c) Actuator Based languages make use of physical objects called actuators to operate on virtual objects [55], they can be translated and rotated on the tabletop to interact with objects. Thus, the IMT becomes a Tangible User Interface (TUI). (see section 2.5.2). Virtual actuators are TUI controls used to operate on other virtual objects on the TUI [63].

2.5 User Interfaces for IMT

In the HCI context, the Graphical User Interface (GUI) is the technological artifact of an interactive system which permits, through a visual language, a friendly interaction with a computer system. The GUI uses a set of pictures and graphic objects to represent the information and actions available to the user. Actions are usually performed using direct interaction. We can distinguish several categories in graphical interfaces for IMT, namely monouser, local multiuser and remote multiuser:

1. Monouser: Designed for its use by only one person at a time [30, 64, 41], they do not promote group work. They are developed to perform complex object manipulation actions more easily than using the conventional computer interface. The interface can be tailored to the actual user. In some cases, the size of the IMT can be an issue depending on the interaction design.

2. Locally Multiuser: Designed for their use by several users simultaneously, adding the design complexities and technological issues raised

by multiuser interaction. For instance, the management of objects gets complicated by their growing number as the number of simultaneous users grows, cluttering the interface and creating interferences [65]. Another issue is the access to shared objects, their creation or cloning, or even the physical distribution on the surface allowing easy access to them. In [63] a set of virtual controls called iDwidgets, conceived for multiuser interaction, are presented. Each control can be personalized for each user. Moreover, the same person can play different roles on the IMT, therefore having different profiles. The available modifications of the controls consist on changes in the behavior, content, appearance and interaction method.

3. Remote Multiuser: Remote multiuser interfaces have two additional problems: (a) the complexity of maintaining interface consistency and coherency across different and remote IMTs, and (b) the necessity of perceiving remote users' presence on the IMT and their actions. From the point of view of the GUI, consistency is limited to update the appearance of the workspace including the effects of the actions performed by distant users. Local users must have real time perception of distant users to make collective work plausible. An example is the system described in [26], as illustrated in 2.5. Local users can see distant users on the screen, and can also talk to them. Moreover, a virtual shadow of the distant user interaction is shown on the tabletop, so distant users' movements can be seen to better understand their intentions and actions.

2.5.1 Object Manipulation

The main issues and solutions for the handling of virtual objects in the IMT workspace are the following ones

- Size and Physical Reach. Most IMT systems are collective work oriented, therefore tabletop surface and resolution must be big enough to make personal and collective work comfortable [66]. The increase in the tabletop's physical surface implies the problem of reaching distant objects when the interaction is direct, due to anthropometric limitations. There are different solutions to this issue, from the determination of the *Zone of Comfortable Reach (ZCR)* [67], to the definition of throw

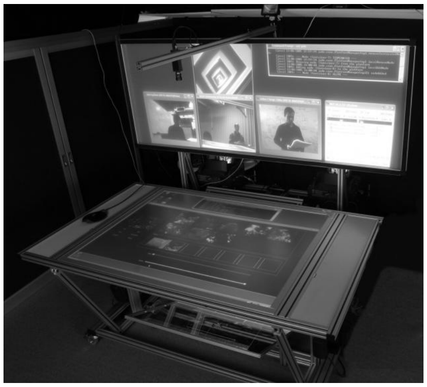


Figure 2.5: Remote multiuser IMT

and catch in Roomware [68] to get distant virtual objects close in the workspace for direct interaction systems.

- Rotation and Translation: Rotation techniques must require little effort, because rotation is a very useful and therefore frequently used procedure. To preserve the role of the non verbal communication of rotation, the rest of the users have to be able to recognize when an user is rotating an object. The problem is how to specify the rotation and translation of objects in a multitouch interaction based IMT [69, 70]:
  - Explicit Rotation: A virtual control on the visual representation of the virtual object permits the specification of the rotation angle, e.g. a text box or scroll-bar.
  - Independent Rotation and Translation: Part of the visual representation of the object is used only for rotation and another part only for translation.
  - Automatic Orientation: Objects are automatically oriented following some kind of criteria, like rotating towards the user or the closest edge on the tabletop.
  - Integral Rotation and Translation: Both can be performed in a single movement and with only one touch. The object is touched and dragged, following several natural conventions.
  - Rotation and Translation with Two Points: The first surface contact point serves as rotation axis point and the second one permits

relative rotation using the relative displacement around the first one. Increasing or decreasing the distance between points is also used to scale the object size.

- Orientation's Role [71]:
  - The user rotates an object towards himself to make it's manipulation and comprehension easier, or rotates it to any other orientation to gain a different perspective of it.
  - Coordination: Users rotate personal objects towards themselves. On the other side, group objects are oriented in another way, therefore establishing an ownership classification of the objects and a segmentation of the space into personal and group areas.
  - Communication: Orienting and object towards himself, the user indicates to the rest of the users that he is working on it. Rotating an object towards another user or user group may mean the purpose of communicating something about that object to the user or user group the object is pointing at.
- Storage: As on conventional tabletops, users tend to create object piles, sort them and move them as an entity. Solutions such as storage bins [65] have been tested.
- Object Control Management: How objects' access rights are handled in the workspace?. In [47] four basic access control protocols are identified. Access is classified as private, public and group based.
  - Release: If an user has the access rights to an object selected, then any other user trying to access it must wait until he releases the object.
  - Relocate: IMT workspace is visually segmented into personal and group areas, and the accessibility level of the objects can tied to their spatial localization, providing an intuitive protocol [47].
  - Reorient: object orientation may also be an indicator of personal or group ownership [71]: if an object is oriented towards the center of the IMT its access is public and if it is oriented to the tabletop external perimeter its access is private.

  - Resize: When an object's size is less than a threshold value, then it becomes private, otherwise it is public.

- Text Writing: Text input is a problem in IMT, which usually lack of conventional keyboards. In [42] this issue is broadly covered.

  - External methods: physical keyboards and voice recognition. Keyboards require additional space on the tabletop and limit the number of maximum simultaneous writing users on the IMT. Voice based writing imposes personal microphones to avoid noise and cocktail party problems.
  - IMT Integrated methods: such as virtual keyboards, hand writing, alphabets of tactile gestures, etc which are represented on the IMT surface and the user interacts there. Some of the issues raised by these methods are:
    - Spatial distribution of characters and symbols on the virtual keyboard.
    - Performance, measured in terms of efficiency and the ease of learning. Efficiency and ease of learning are usually opposite, efficient techniques usually require more learning time.
    - Environmental factors. Each technique has specific space requirements on the IMT, or out of it, which must be considered. Rotation, translation and simultaneous multiple interaction requirements of IMT must also be met. And in the same sense, direct interaction on the IMT, which is one of the grounds of natural interaction.

2.5.2 Tangible User Interfaces (TUI)

The Tangible User Interfaces (TUI) [72, 40] were first introduced by Ishii and Ullmer in the CHI conference in 1997 [73]. They defined them as graphical interfaces which augmented the real physical world by adding digital information to everyday use objects and physical environments. The TUI input events, thus, consists of the manipulation of daily objects with the hands and in the same way as they are used in their usual contexts. The computer system processes the input events, providing the corresponding answer. It has been argued [74] that TUI provide better global view of state of the system than GUI, discovering relations among objects and inside the application in use. Moreover, TUI promote concept communication among users, because they are easily illustrated by means of the physical interaction objects. The design of TUI revolves around the *affordances* of the objects, i.e., the external appearance and physical properties of an object which give an idea about its function and usage. Recent works [57], focus on the haptic aspects, system ubiquity and direct manipulation using physical objects.

The main properties and design requirements of a TUI are:

- Association between the tangible representation and the underlying digital information. One of the difficulties of TUI design is how to associate physical objects and their manipulation method with the digital computation and its response in coherent and clear manner.

- Perceptive association between tangible and dynamic intangible representations. TUI are based on a balance between tangible and intangible representations. The intangible aspect of the representation, usually graphics and sound, provide with great amount of dynamic information created by the underlying computation.

The main benefit of TUI over traditional GUI, is there is an immediate haptic passive response while manipulating a physical object. Without the need to wait for a digital synthetic haptic response, users can accomplish their actions. TUI are trivially persistent, because physical objects are too. Tangible objects also have a physical state, with their physical associated to the digital state they represent.

2.6 Multitouch Tabletops

In this kind of IMT the only interaction method available is multitouch on the tabletop surface, requiring the use of multitouch surfaces, which are able to recognize but many points of contact simultaneously in different positions. One of the reasons of the growth of the researches involving multitouch IMT is the recent development of the Frustrated Total Internal Reflection (FTIR) which is described below and further commented at appendix A. Using this technique, cheap, large sized and easy to build multitouch surfaces can be built.

2.6.1 Multitouch techniques

For an incomplete chronology of multitouch techniques, readers are referred to [75]. Multitouch techniques are very heterogeneous, like Computer Vision and camera based, electric or magnetic field detection based and acoustic disruption. Further details will be given at the appendix A.

1. Computer Vision: This kind of techniques make use of a camera to capture the scene and Computer Vision algorithms to process the images, with the disadvantage of the computing cost inherent to Computer Vision applications. The two main approaches are:

   (a) Frustrated Total Internal Reflection (FTIR): This device is composed by a transparent or at least translucent surface, an infrared camera under it, looking at the surface, and a set of infrared light sources, usually infrared Light Emitting Diodes (LED), placed at the edges of the surface pointing to the middle of it. The emitted light is reflected inside the surface and does not leave it, unless an object touches the surface. In that case the light is reflected across the opposite surface side of the touching point, towards the infrared camera. In this camera, the touching points will appear as bright blobs against a dark background.

   (b) Diffused Illumination (DI): This technique requires a transparent or translucent surface, an infrared camera under the surface pointing at it and a set of infrared constant light source, usually infrared LEDs on the multitouch surface. When there is not any object on the surface, the infrared camera captures a completely bright image, but when an object is on it, or very close to the surface, a shadow appears in that position. In this way multiple touches can be recognized.

2. Capacitive: This techniques are based on the use of low intensity electric fields. There are two type of techniques, called *the human shunt* and *the human transmitter* [76]. The most representative multitouch surface using these techniques is called DiamondTouch [10] and it is based on *the human transmitter* principle. It was developed by Mitsubishi Electric Research Laboratories (MERL). This multitouch surface has been used by several IMT [66, 49, 26, 41, 47, 58, 77]. The most important feature of this surface is that it can identify the user

that touched the surface. Its main disadvantage is that users must wear or touch a special device to create the electric field, reducing the naturalness of interaction.

(a) The human shunt: A potential is created between an oscillator electrode and a virtual ground electrode. Then the intensity between the electrodes, i.e. the electric field, is measured. If the electrodes' size is tiny compared to the distance between them, then the electrodes can be modeled as punctual charges, producing dipole fields. The intensity of the dipole field inversely varies with the distance. When an object with a size much bigger than the size of the dipole approaches to the dipole field, the received intensity reduces, since that intensity is redirected to ground. As the object gets closer, the dipole field intensity reduces. Using this procedure multiple touches can be recognized simultaneously, but also objects in the proximity of the surface and the approximate distance between the object and the surface.

(b) The human transmitter: Low frequency energy is capacitively coupled through a person's body, making him an electric field emitter. Several receptors on the tabletop can perceive the intensity amount received. The greater this intensity the closer the receptor and the emitter are.

2.6.2 Interaction Details

The multitouch IMT opens new potential interactions. For instance, ordinary click and drag operations of a digital mouse can be emulated, extended to several mouse pointers which can coexist in the workspace in the same time. Furthermore, simultaneously touching several different parts of an object can be combined to ease the specification of some actions like rotation, scale and movement of objects. To show some of the potential, a classification of the recognizable touch gestures by a completely multitouch surface is detailed next. We differentiate between pose and gesture interactions. Pose is static in the sense that the action area is restricted and very small while gesture is dynamic because it includes some kind of motion of the hand or the object. In multitouch surfaces multiple poses and gestures of each type in the classification can be present simultaneously and in multiple instances.

1. Monotouch
   (a) Pose.
      i. Ordinary press. The ordinary click action in a mouse.
   (b) Gesture
      i. Ordinary press with a motion. Analogous to the click and drag action of a mouse, but drawing a distinguishable shape with the drag trajectory.
      ii. Contiguous press sequence. Clicking in several near points in a specific sequence.
2. Multitouch
   (a) Pose
      i. Multiple press without specific shape, but with a minimum size, and in a specific position. Pressing of a minimum of adjoining points simultaneously. Only the position is considered.
      ii. Multiple press with a specific shape. Pressing of a minimum of adjoining points simultaneously showing a distinguishable shape. The physical appearance of the pressed surface is used.
   (b) Gesture
      i. Multiple press with a specific motion. The shape of the trajectory is part of the gesture.
      ii. Multiple press with shape and motion. The hand maintains a specific shape during a specific trajectory.
      iii. Multiple press with changing shape and trajectory. The hand follows a trajectory starting with the hand or object showing an initial pose or shape which changes during the trajectory following a specific pattern.

In [78] orientation of the touched surface is also considered as relevant information for multitouch interaction. But that information can not usually be obtained directly from the multitouch device. It is usually an estimation based on the main axises of the shape, which do not usually represent the user desired orientation, and therefore, it is not a reliable information source.

2.6.3 Disadvantages

Multitouch interaction requires touching the tabletop surface, so it produces wearing away and deterioration of the surface. Moreover, dirt accumulates, which leads to rejection by people due to hygienic reasons. Due to social and cultural issues, contempt for touching a surface which has been previously touched by many other people, or even touching a surface while another person is also touching it close by. Finally, being a direct interaction method, reaching distant objects can become a problem as the size of tabletop surface grows. Almost all systems do not recognize the user.

2.7 IMT for Casual Users

This kind of IMT are focused on occasional interaction by any kind of user, in environments of great flow of people. Therefore, naturalness and simplicity of use are the most important design features, as discussed in section 2.1. These requirements include the use of interaction methods and languages which are easy to learn or they do not even require any kind of learning and are also difficult to forget. In [79] several IMT from nine exhibitions are reviewed.

2.7.1 Perception of Naturalness is Not Universal

The most important factor when designing the interaction method for these IMT kind is naturalness, but this feature rarely receives the attention deserved in the literature. Common sense gives people clues about how to interact with an object. This in fact means that several factors, namely the object external appearance, the object usage context, previous user interaction experiences and sociocultural aspects, make people feel an interaction method with a specific object more natural than others.

- The external appearance of an object determines the interaction methods that may seem more natural to a user. The object's size, shape, weight and texture among other characteristics conditions our interaction with it, making some interaction methods seem more convenient, natural or intuitive than others. For example, it is not natural to grab a knife from its blade.

- The context in which an object is used also plays a relevant role. Depending on the environment where an IMT is placed, and the main purpose of that location, specific features and functionalities will be attributed to the IMT. For example if people find a table surrounded by chairs, sitting on them to interact with the table seems natural, or if an IMT is placed in a workshop, it seems natural that users will be able to interact with the IMT using some of their physical tools, like a caliper.

- Sociocultural aspects also affect to the perception of naturalness. On one hand, in some cultures, like Eastern or even North European, body language is very subtle, therefore hand gesture interaction can produce a certain degree of contempt, and what it more, gesture intensity will be low. On the other hand, in cultures like Mediterranean or in many South American countries where there is more tradition to expose sensations and emotions, hand gesture interaction is accepted as natural, since body language is more normalized. A society's technology exposure degree is an important factor for naturalness perception, like it is stated at [62], which also refers to the effects of culture in interaction naturalness perception.

The Ecological Approach is a psychology approach based on the work by Gibson [80], which defends that psychology should be the study of the interaction among people and with their environment. Later on, there was an adaptation of these ideas to the Human-Computer Interaction (HCI) context, carried out by Gaver [81], Kirsh [82], Norman [83], Rasmussen and Rouse [84], Vicente [85] and Woods [86]. According to this adaptation of the theories, interaction naturalness depends on the object use context, and the interaction is determined by the *Ecological Constraints* and the *Affordances*. The *Ecological Constraints* are real world structures which guide people actions. And the *Affordances* are the objects attributes which permit people to know how to use them. Some people for example are reluctant to wear, grab or touch objects that have been previously used by many other people. And intrusive devices can also produce rejection, like fingerprint or eye recognition devices.

2.7.2 Hand Gestures versus Multitouch

Hand gestures and multitouch interfaces are the most natural interaction methods, and the preferred ones for casual users. Therefore, we feel that it is necessary to make a comparison between them, showing the advantages and drawbacks of each method. Let us consider them from the point of view of direct interaction as reference. To decide which of the two interaction design approaches is more direct, we have to consider the combination of the IMT system input-output devices.

Multitouch is conceived for a two-dimensional interaction. Therefore it will be more intuitive when the GUI is represented on a flat surface and shows two-dimensional virtual objects. Under these circumstances the interaction can be direct, since it can be performed on the same dimension where the interface and the virtual objects lie.

If the GUI is not represented on a plane, but in a three-dimensional volume, i.e. a hologram, then hand gesture based interaction seems to be the most natural interaction method of both. Hand gestures are naturally performed on a three-dimensional space, and if they can be used to the manipulation of the GUI three dimensional volume itself, then direct interaction is achieved. Unfortunately, current three-dimensional projection devices impose several restrictions, like enclosing the volume, therefore direct manipulation is not usually available.

When the GUI is shown on a flat surface, but the graphical representation simulates a three-dimensional environment, then both multitouch and hand gestures seem natural. Multitouch interaction on the plane, and hand gestures in the proximity of the surface, otherwise gestures are no longer direct and accurate, neither natural.

Because they are contactless, hand gestures do not produce any physical wear on the IMT, and neither any hygiene nor tidiness issue would be risen. On the other hand, the multitouch interface provides a haptic response which reinforces interaction naturalness. Actions which require high accuracy are more difficult to perform using hand gestures than multitouch, because the user must keep his hands on the air. Moreover, as the user raises his hand over the tabletop surface, the perspective effect makes hand gesture interaction difficult.

The combination of both interaction methods creates a multimodal interface, letting the user choose the method that better suits to his preferences and each task independently. But, if different actions can produce the same effect on the IMT, ambiguity appears. Then the user could have difficulties in distinguishing which of his actions produced a particular effect on the IMT. Therefore, if different actions can produce the same effect, then each of these actions must be clearly linked to a different interaction method.

2.7.3 Interaction Languages for Casual Users

The interaction protocol followed with the IMT can be specified and treated as a language, we can define its lexicon, grammar and semantics. The most important aspects to have into account when defining an interaction language for an IMT focused on casual users those related to the ease with which the user can start interacting efficiently with the system: short learning time and persistence in the mind of the user (once learned, never forgotten). This is achieved using an intuitive and simple language which follows a simple logic without ambiguities.

Natural interaction languages are those with natural lexicon, grammar and semantics in the sense that they are immediately inspired in culturally accepted facts. These interaction languages include the voice communication using natural language. Of course, natural interaction languages are the most adequate for casual users. They can extrapolate their common life knowledge into the interaction with the system and the learning process can have the form of a game very easily. The combination with physical actuators can contribute with the haptic feeling to the naturalness of the interaction, but unfortunately the IMT GUI can not modify the actuators, except from very specific cases like [38].

Besides naturalness, simplicity is another design goal. Simplicity avoids overloading interaction elements (lexicon, grammar) with meanings, and thus avoids ambiguities. Users can accept and learn faster simple and clean interaction languages.

2.7.4 Guiding the User on the Interaction Possibilities

In the case of the casual users, the interaction capabilities perceived by the user must be complete, in the sense of being able to perform a successful interaction, from the first contact with the IMT. The main goal is to guide the user to interact with the IMT immediately, and give him afterwards clues to discover all the remaining interaction possibilities gradually so his interaction experience is as full as possible.

One strategy to perform such a user guidance, consists on the use of a short visual and optional tutorial, in the form of a game with increasing difficulty, showing part of the interaction possibilities available, and helping the user to develop his mental model of the IMT. Once the tutorial is finished, the user would be able to infer the rest of the possibilities. This is a kind of disguised training process, which should not be necessary for very natural interaction languages, but which in fact can be very effective in realistic environments.

Other strategy consists on letting the user interact freely with the IMT. Meanwhile, the user behavior is studied to offer suitable suggestions about how to fulfill the task the user is trying to accomplish, according to the IMT inference system. This user behavior analysis and modeling adds complexity to the underlying IMT computational intelligence system which performs pattern recognition, planning, and other intelligent tasks. Moreover, this kind of systems can be annoying to the user, because it may provide unrequired suggestions, be silent when needed, it can be intrusive in the normal work of the user.

The use of information posters around the IMT can provide a first context about the interaction possibilities available. Although this kind of help is useful to provide a first impression about the IMT, it is not enough, unless extensive information is provided. This approach reduces the naturalness of the interaction, and it is limited to very basic information. However, careful design of the information posters may be very attractive to the users, which is something valuable in the case of the casual users.

2.7.5 Interaction Naturalness Measures for Casual User IMT

It is necessary to have naturalness measures to evaluate in advanced the potential success possibilities of an IMT for casual users in terms of user acceptance. With that in mind, two measures are proposed.

- Time Period Required to Master the Interaction: It is defined as the time period required to achieve an efficient interaction, which means to acquire the ability to perform a task with the minimum resources possible. Furthermore, it is assumed that the user has achieved the ability to perform every required task, using the interaction methods available. The shorter this time period is, the more natural the interaction method is.

- IMT Occupancy Rate: It is a measure of the acceptation or rejection of an IMT by users. It is defined as the time period when the IMT has been in use divided by the time period when people has been in the proximity of the IMT. This measure ranges from zero to one and the bigger this value is, the bigger the acceptance of this IMT is.

2.8 Design proposal

This PhD Thesis work has been done part time at the University and the company Innovae Vision. The context of the PhD is the development of a working prototype IMT for casual user applications for entertainment and didactic applications (edutainment). Envisaged applications are teaching stands at museums, and a platform for advertisement of new products and services at companies.

The basic chosen design includes hand gesture based interaction combined with multitouch like in [87]. This combination allows to perform complex but less accurate actions using hand gestures, while using multitouch for more conventional and accurate interaction. The interaction language lexicon would also permit interaction items composed of both gestures and multitouch. The next sections comment some technical aspects of the design.

2.8.1 Hardware

The first prototype will be composed of a table shaped furniture, enclosing a computer and a flat surface LCD screen. A multitouch surface will be attached to the LCD screen, including a protective layer. And a camera over the table shaped furniture captures the scene, i.e., the volume over the tabletop surface.

- Furniture: Two versions of the prototype IMT are planned to be developed. The first one for indoor environments, lighter and less expensive and a second one, rugged and with a top mounted protective ceiling for outdoor environments. This second version will be completely metallic, with an antireflective coating on the tabletop surface (to avoid sunlight reflections on the surface), and sealed to be rain, water and dirt proof.

- LCD screen: It has been selected due to its higher resolution, improved durability and performance under bad lighting conditions. A commercial 46" LCD screen with built in multitouch capabilities has been chosen.

- Video Camera: Initially an ordinary color camera has been chosen. Due to the limitations of nowadays off the shelf computers, the height of the camera relative to the tabletop surface, and the size of the tabletop surface and ordinary hands, relative to it, capture resolution has been limited to 320x240. The response rate has been initially limited to 24 frames per second, hand scale will not be considered for hand gesture recognition.

- Computer System. Two options are being considered: (a) a combination of the latest multiprocessors of both Intel (Core I7) and AMD (Phenom), and a high end graphic card, and (b) use of the video console PlayStation 3 as a computer because of its high performance vectorial multiprocessor called CELL. There is Linux distribution for this video console called Yellow Dog Linux, optimized for this specific processor.

- Communication: Following the idea of natural communication, wireless interaction between the IMT and external devices is a basic requirement. Consequently WI-FI and Bluetooth interfaces will be available in the computer system. Wired network connections are also required because introduce less delay and permit higher transmission rates. Therefore a Gigabit Ethernet interface will also be included.

2.8.2 Software

Software development includes three kinds of contexts: input recognition, input/output management and natural input/output oriented applications.

Input recognition

Since our multitouch surface solution is commercial, the vendor will provide with all the necessary tools and functions, through the device SDK. However hand gesture recognition is specific of our system, and also the combination between hand gestures and multitouch. The research and development phase of the hand gesture language is the main contribution of this thesis work. Briefly, our hand gesture recognition consists of the following steps:

- Background substraction. Taking advantage of the built in polarizing filter included in LCD screens, we add an additional polarizing filter to the camera, rotated 90º to the filter in the LCD screen. This makes the images shown by the LCD screen black to the video camera. Then it is enough to perform a simple thresholding to obtain a binary image with the silhouettes of the objects on the tabletop.

- Connected set labeling so that each connected shape is extracted.

- Skeleton computation for each valid connected set. We use a combination of Voronoi skeleton and an efficient pruning stage to obtain a simple, connected and stable skeleton. This procedure is presented in Chapter 4.

- Shape recognition based on the computed skeleton and the fusion of the multitouch information.

- Gesture recognition as a temporal sequence of salient shapes for a connected set, also including the path followed by the centroid of the connected set. Initially a finite state machine will be defined, but further improvements will include the use of Hidden Markov Models. The recognition of gestures performed as a combination of both hands' actions will also be explored.

Input/output management

Multitouch and hand gesture based interaction allow several simultaneous users, and each simultaneous user with many simultaneous interaction actions. Consequently, changes must be make in the management of user interaction from the core of the Operating System. In this sense users must be tracked in time, including position, path, pose and gesture, and this requires an efficient message passing and management procedure. Several generic interaction words must be defined for the O.S. graphical interface, and providing with suitable SDK/API for application development using this new interaction paradigm. There are already several initiatives to achieve this goals in the multitouch area, like those described in Appendix 2.6, but they should be extended to include other natural interaction methods. Linux is already modifying its kernel to adapt to the multitouch multiuser environment and Microsoft will also include multitouch support in Windows 7.

Natural input/output oriented applications

Since this is a new development platform, with a new interaction paradigm, applications must be developed which take advantage of all the benefits of this new interaction metaphor.

Two kind of software is intended to be developed, and it differs in the target user type:

1. Casual users. Simple and easy to learn interaction and applications, intended to attract users and focused in leisure time and publicity.

2. Professional users. Focused in efficiency and team work under complex environments like design and visualization. This kind of applications are intended to be developed latter due to their higher complexity.

2.9 Conclusions

Interactive Multimedia Tabletop are a new tool for collective work environments and also for information retrieval in public places. This two environments have very different requirements. While in work environments productivity and efficiency are the most important characteristics, in public places the most important is that interaction must be easy and enjoyable, and an special effort must be put in making the IMT appealing and avoiding any kind of rejection. Therefore a first distinction has been done between IMT focused on regular and casual users.

IMT make use of natural interaction methods, which usually involves the use of hand gestures, physical object manipulation and natural language, i.e. speech. Most of this kind of interaction methods have been reviewed in the context of IMT with an special attention to multitouch interaction, and its combinations with other interaction paradigms like hand gestures or voice.

Aesthetics both in the GUI/TUI workspace and the hardware's external appearance receive a special attention, and ubiquity issues are also present. And multimedia data is the main kind of data manipulated on them. Everything is devoted to natural and easy interaction and to break the barrier between users and computers.

The new multiple simultaneous user interaction models are a challenge to the design of virtual object interaction and collaboration. There is still a lot of work to do in user group management and technical aspects both in hardware and software, but this is a research area which has received much interest.

As a contribution to this research area, the PhD candidate of this document proposes the combination of hand gesture and multitouch recognition on an IMT, which is the context for the rest of the work. The main idea is to use conventional hand gestures for complex but low precision actions over the tabletop and use multitouch gestures for accurate actions on the tabletop. Gestures involving the combination of both interaction methods will also be explored. Future improvements would include speech recognition, but not physical actuators, unless technology limitations which do not easily let the IMT move the actuators yet are overcomed.

In the PhD candidate's opinion IMT will soon become common tools in the areas of education, publicity, science and design, and a milestone in the integration of computers into the society.

Chapter 3

Skeletons in 2D

In this chapter we will give a review on the definition of Skeletons and their computation as found in the literature. We will be dealing with 2D skeletons for real time shape recognition, therefore the emphasis will be on efficiency and stability. Efficiency is related to the computational complexity, that impedes the real time realization. Stability is related to the continuity of the transformation of images into skeletons: it is desired that shapes that look alike have similar skeletons. In other words, small changes in the shape boundary must not induce great changes in the skeleton.

In section 3.1 we give some introductory definitions. In section 3.2 we present the basic algorithms for skeleton computation. In section 3.3 we introduce the two basic skeleton regularization approaches. In section 3.4 we present the basic shape recognition algorithms using skeletal representations. And we give some conclusions in section 3.6.

3.1 Introduction

There are many ways to represent objects in 2D and 3D:

- Feature-based methods, which represent each view as a collection of line segments, curves, corners, regions, etc. The success of such methods depends largely on the extent to which the features are present and can be reliably extracted. The most important methods in this category are further classified into:

    - Landmark representations: An object is represented by an ordered set of geometrically salient locations on the object, which are suitable for automatic recognition.

  - Boundary representations (b-reps): An object is represented by features extracted from its boundary. This can be done by means of decomposition into basis functions like spline fits or other orthogonal basis functions.
  - Region based representation: An object is described by a set of its basic constituent parts and the geometrical and hierarchical relationship among them. Shape skeletons fall into this category.

- Appearance-based methods, which treat the raw image as a single feature in a high-dimensional space. Feature extraction is then formulated as an algebraic transformation between euclidean spaces. Successful approaches in this category include Principal Component Analysis (PCA) and Independent Component Analysis (ICA).

In this PhD Thesis we will be concerned about skeleton representation of shapes. The skeleton of a shape is a thinned version of the original shape, a curve whose points are equidistant to opposite sides of the original shape boundary curve. In figure 3.1 we present the skeleton of the binary image of a hand gesture, one of the images we will be dealing in the application of our works. Skeletons are composed of branches, connected by different joint points. They represent the essential structure of objects and how components are connected to form a whole. Therefore, they are proposed for the recognition, classification and retrieval of shapes. Reconstruction of the original shape is also possible when the minimum distance to the boundary curve is available for each skeleton point. According to Blum [88], skeletons are shape descriptors specially suitable for the description of biological or amorphous shapes present in nature for which other description schemes based in ordinary geometry are inadequate. *Skeletonization* is the procedure of obtaining the skeleton of a shape, which can be defined in a bi-dimensional or a three-dimensional space [4, 89, 90]. This PhD is devoted to the skeletonization of binary shapes in the bi-dimensional space.

There are many different skeleton definitions in the literature. Most of them are equivalent if we work in a continuous image domain, but usually are not equivalent in the discrete image domain case.

The Medial Axis Transform (MAT) is a concept closely related to the skeleton. In the literature, the distinction between the concepts of binary

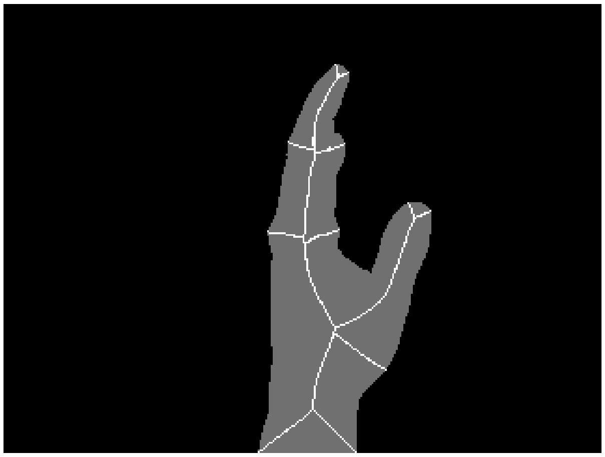


Figure 3.1: **Example skeleton of a hand shape**, computed using the combination of Voronoi skeletonization and pruning presented in this work in section 4. Black color corresponds to the background, gray color corresponds to the original shape and the white color corresponds to the shape skeleton.

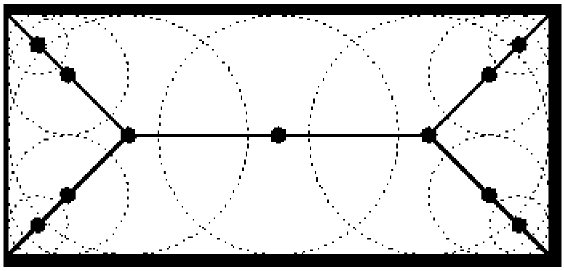


Figure 3.2: **The skeleton of a rectangle**. Dotted line circles correspond to maximal disks and black dots are their centers corresponding to skeletal points.

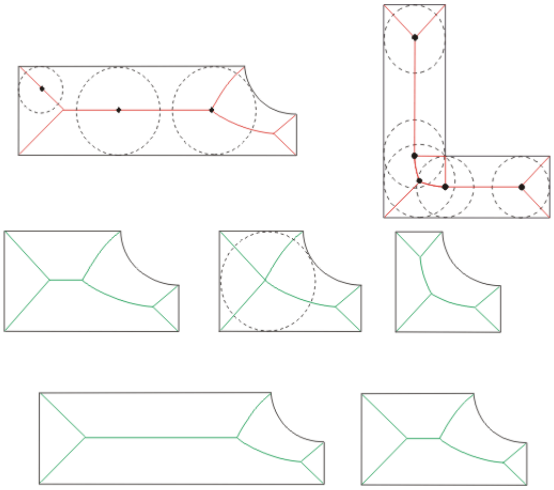


Figure 3.3: **Several skeleton examples**. Dotted line circles correspond to the maximal disks, and the black dots to their corresponding skeleton points.

skeleton and Medial Axis is somewhat blurred, sometimes they are considered equivalent [91, 92, 93], while others [94, 95, 96, 97] consider them similar, but not equal. The Medial Axis can be considered as a particular case of skeleton. The main difference is that the skeleton is given as a binary image distinguishing background and skeleton points, while the Medial Axis is given as a real valued image where each skeleton point has as its value in the MAT image that of the minimum distance to a shape boundary point (i.e. Distance Transformation function value, described in section B.1 from appendix B), and the pixels not belonging to the skeleton take a zero value. Some of the mathematical properties of the Medial Axis Transform are discussed in [98]. Despite all this precisions, we will use the terms Medial Axis and Skeleton as interchangeable terms in many places of this report.

The earliest related work, in 1967 Blum [99] defined the MAT in an intuitive manner. He was the first to describe the MAT of a 2D shape by analogy with a fire front which starts at the boundary of the shape and propagates isotropically towards the interior. The Medial Axis is defined by the locations at which the fire fronts collide. Starting from this intuitive description, several definitions have appeared in the literature which are denoted as the *grassfire* or *eikonal flow* definitions. Some of these definitions are the following ones:

Definition 3.1.1. Blum Skeleton: From the definition of Symmetry set, the

Medial Axis is defined [88][100] as the as the closures of the centers of all the circles which are bi-tangent to the shape boundary curve.

.

Definition 3.1.2. Jain skeleton: In [101] the skeleton definition is based on the centers of maximal disks. A disk B is said to be the maximal disk included in a set A if $B \subseteq A$, and if exists another disk D so that $B \subseteq D$, then $D \nsubseteq A$. The skeleton of a shape A is defined as the set of centers of all the maximal disks in A. Figures 3.2 and 3.3 show both maximal disks and their corresponding centers, as well as the ideal skeleton curve obtained.

.

Definition 3.1.3. Woods skeleton: In [102] a definition based in the centers of bi-tangent circles is proposed. The skeleton of a shape A is defined as the centers of the disks which are bi-tangent to the boundary of A. Therefore, the skeleton points are equidistant to the shape boundary curves.

The following is a definition that will be useful in some of the algorithms that will be described in the next chapters.

Definition 3.1.4. Generative Points: The points where the maximal circles/disks corresponding to skeleton points touch the boundary are called *generative points* of that skeleton point.

.

Definition 3.1.5. Straight Skeleton [103, 104, 105]: The Straight Skeleton consists of straight line segments only. It also has a smaller combinatorial complexity (n—2 internal nodes, with n the number of polygon vertices) than the medial axis ($n + r$—2 nodes, with r the number of reflex vertices). To construct the straight skeleton, we let wavefront edges move parallel to the polygon sides. In contrast to the medial axis, edges incident to a reflex vertex will grow in length. The front remains a polygon, whose vertices during the process trace out the skeleton (see Figure 3.5).

.

Definition 3.1.6. Maxwell set: In [106]the *Maxwell Set* is defined as the set of locations internal to the object with more than one corresponding closest boundary point in the sense of Euclidean distance. Each point in this set is augmented with its distance to the boundary.

The Distance Transformation (DT) function returns the minimum distance to a boundary point for each inner shape point (see section B.1 from appendix B for further details). It can be verified that the set of local maxima in the DT function are skeleton points [94].

Another medial structure is the *Shock Graph*, [107], which is obtained by viewing the Medial Axis as the locus of singularities (shocks) generated during the fire front propagation from the shape boundary. This dynamic view of the Medial Axis associates a flow direction and instantaneous speed to each shock point,[108]. The flow is defined according to the variation of the radius of the maximal disk associated with each Medial Axis point, going in the direction of their increase.Shock points may be classified according to the number of contact points and to the flow direction, as described in [109]: source and sink points determine the nodes of the graph while the links connect source points to sink ones and define the arcs of the graph. In addition, attributes are associated to the shock graph to store both the intrinsic geometry of the portion of shape corresponding to a link and the radius and the flow direction of each node. Analogously to the MAT, the shock graph structure and the corresponding point classification have been extended to 3D shapes [108]. Also, in this case the shock graph structure is not a planar graph. The medial axis and the shock graph differ more in the interpretation of the structure entities than in the geometric abstraction they provide. For example, the shock graph and the MAT of a curved shape can have the same arcs and nodes, but the shock graph also associates the growing direction of the radius of the bi-tangent spheres to each arc, see Figure 3.4. In general, we may consider that the shock graph is a partition of the medial axis. Section B.5, from appendix B, further describes Shock Graphs.

A distinction can be made between *geometric skeletons*, which include all the definitions of skeleton previously presented and the *skeletons derived from topological structures*, which are higher abstractions of the skeletal descriptor, initially without geometrical information, but which can be enriched with it. The Medial Axis transform, as well as the Shock graphs, are the most important examples of geometric skeletons. Although this PhD is only related to geometrical skeletons, for the sake of completeness we mention Reeb graphs [110, 111], which are the most important example of skeletons derived from topological structures. For a review about Reeb graphs, the reader is referred to [112, 113], and an example is shown in fig. 3.6.

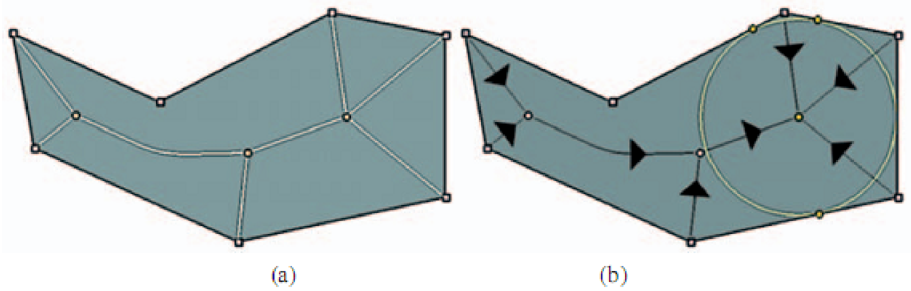


Fig. 2. The medial axis (a) and the shock graph (b) of two simple curves.

Figure 3.4: **Difference between Medial Axis and Shock Graph**. Picture taken from [1]

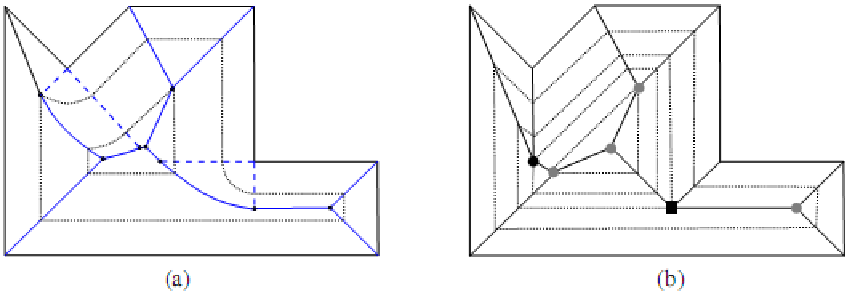


Fig. 12. Medial Axis (a) vs. Straight Skeleton (b). In (b) the black disk marks a reflex edge annihilation, while gray disks mark convex edge annihilations. An edge-edge collision generates the arc between the black box (vertex-edge collision) and a gray disk (convex edge annihilation)

Figure 3.5: **Medial Axis** on the left **and Straight skeleton** on the right. Picture taken from [2]

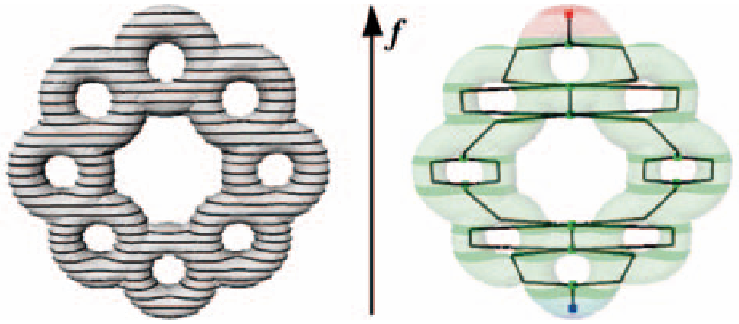


Figure 3.6: **Reeb Graph**. An example of surface, on the left, and its Reeb graph representation with respect to the height function, on the right.

Definitions based on disks, circles and spheres are very sensitive to noise in the shape contour, contrary to the fact that the skeleton that people would intuitively draw is very stable under deformation and noise in the shape boundary. That is, we want that the skeletonization algorithms produce similar skeleton topologies for similar shapes and that the skeleton varies smoothly with modifications on the shape. We call this property stability of the skeletonization algorithm. Usually, the skeletonization procedure is computationally expensive, with exceptions like [4, 114]. Consequently, there has been a special interest in innovative techniques to compute the skeleton, with the purpose of both reducing its computational cost and improve its stability. The goal is no longer to compute the exact Medial Axis of an specific shape, but to be able to compute an acceptable approximation.

Supporting facts of the use of skeletons as shape descriptors are the research works proving that shape skeletons are a relevant component of the human visual models. For example there is literature that support the relation between the part decomposition, i.e. branches, derived by the skeleton and the cognitive process of the brain [115]. And other works show that the shape parts defined by the skeleton which arise from locations of extrema of negative curvature on the boundary are often associated with the visual decomposition of objects [116, 117, 118, 119]. For further information the reader is referred to [120].

These are the major advantages of using skeletal representations of objects, according to [121]:

1. Since it is an interior representation of the shape, it is subject to both

geometric and mechanical operations applicable on the object's interior, such as bending, widening, elongation, and warping.

2. It provides rich geometric information, giving simultaneously positional, orientational, and metric (size) description in any locality of the interior and near exterior of an object.

3. It provides a basis for description at multiple spatial scales and thus provides efficiency of computation and efficiency in the number of population samples needed to estimate object geometry probabilities.

4. It allows one to distinguish object deformations into along-object deviations, namely elongations and bendings, and across-object deviations, namely bulgings and attachment of protrusions or indentations.

5. Its branches at the larger scales divide objects in a way that makes automatic object recognition effective [122].

6. It provides descriptions of objects and their geometric transformations that are intuitive to non mathematical users.

7. It generates object-relative coordinate systems for object interiors and their near neighborhoods that provide useful correspondences across instances of an object.

8. It provides a means for describing the locational, orientational, and size relations between one object and a neighboring region of another object within a complex of objects.

The main disadvantages of using skeletons for image representation are:

- Computational complexity of the skeletonization algorithms,

- Lack of robustness against boundary noise,that is the lack of stability of the algorithms,

- Ambiguity, when the skeleton is represented as a binary image different shapes can produce the same skeleton (see fig. 3.7).

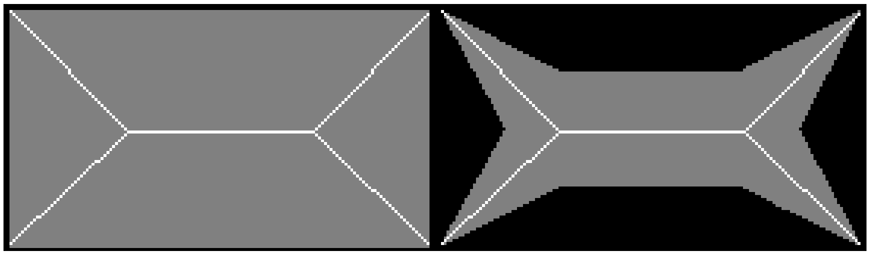


Figure 3.7: **Skeleton ambiguity**. Two different shapes producing the same skeleton. Background is shown in black, the shape in gray and the skeleton in white.

3.1.1 Digital discrete space

Translating the skeletonization ideas from the continuous space into a digital discrete space produces several issues which must be considered. A different connectivity type for the object (foreground) and for its complement (background) must be used. If not, a topological paradox may appear: a closed curve/surface may not divide the background into disjoint parts, or the background may be divided into disjoint parts by an open curve/surface.

In the discrete space it is impossible to give a precise solution to apparently easy tasks, such as identifying the middle point in a segment. If the segment consists of an even number of elements, any of two elements, or both, will be considered as the middle point if we want a discrete solution, i.e. a solution contained in the image pixel space. This fact produces problems in obtaining the skeleton of a rectangle with an even side length. The skeleton will be two pixel width, if a distance transform based approach is considered, or will disappear if the bi-tangent definition is considered, since there is not any bi-tangent circle. Consequently, the exact medial axis cannot always be computed in a discrete space.

Since in practice the image domain is discrete, several definitions are provided, in order to establish the framework for further descriptions.

First of all, lets consider a digital image I, whose pixel sites $x \in \mathbb{Z}^2$. The original images, from which skeletons are computed, are binary, i.e., $I(x) \in \{0, 1\}$, where 0 valued pixels correspond to the background and 1 valued pixels correspond to the foreground.

Definition 3.1.7. *Adjacency* relations. Two pixel sites $p = (x_p, y_p)$ and $q = (x_q, y_q)$ of I , such that $(|x_p - x_q| \leq 1) \wedge (|y_p - y_q| \leq 1)$, are said to be 4-

adjacent if $|x_p - x_q| + |y_p - y_q| = 1$, and 8-adjacent if $0 < |x_p - x_q| + |y_p - y_q| \leq 2$. This adjacency relations produce 4-connected and 8-connected discrete spaces. We denote the adjacency relation as $|p, q|_A$.

.

Definition 3.1.8. A connected set $P = \{p_0, ...p_N\}$ is a set so that $\forall p_i, p_j \in P$ we have that $\exists Q \subset P$ where $Q = \{q_0, ..., q_M\}$ so that $q_0 = p_i \wedge q_M = p_j$ and $\forall 0 \leq k < M$ we have$|q_k, q_{k+1}|_A$, according to one of the adjacency relations previously defined.

.

Definition 3.1.9. A *shape* S_o is a binary image: we can define it as the subset of the image domain with value $I(x) = 1$. The complement of S_o, relative to the image domain, denoted $\overline{S_o}$, is called the *background.* To avoid segmentation paradoxes S_o is 8-connected and $\overline{S_o}$ is 4-connected. Sometimes a shape is a single connected object, which can have holes in it.

.

Definition 3.1.10. The *digital distance* between two pixel sites x and y in the $\mathbb{Z}^2$ image domain is the length of the shortest path connecting x to y, where the path consists of steps between close neighbors. The distance thus depends on the chosen neighborhood relation and the definition of the path length.

3.2 Skeleton Computation

The different definitions of the image object's skeleton, and the diverse points of view on its computation have derived into diverse *skeletonization* procedures. These techniques include procedures for the prepocessing of the original boundary in order improve the robustness and the efficiency of the skeletonization; and skeleton regularization procedures applied on the computed skeleton, aiming to reduce its complexity, removing spurious branches in order improve the robustness and stability of the final result.

The desirable properties of a skeletonization procedure include:

- Completeness of the skeleton. There are not regions not represented by the skeleton. In other words, each point in the object's boundary has a corresponding point in the skeleton, in a many-to-one correspondence.

- Connectivity of the skeletal points. The skeleton of a connected component is a connected component.
- Thinness: ideally one pixel thick.
- Simplicity. The skeleton must not be cluttered with too much detail that will hamper the subsequent processes. The algorithms must conform to an underlying minimum information principle.
- The resulting skeleton must be centered inside the shape.
- Each skeleton branch represents a significant part of the shape.
- Robustness to rigid body transformations of the object (i.e. scaling, rotation and translation).
- Stability: Smooth variation of the skeleton under minor changes on the shape contour. In other words, similar skeletons should represent similar objects, and different objects should have different skeletons.

Reversibility, i.e. the possibility to reconstruct the original shape from the skeleton, is also a desirable property of the resulting skeleton. However an exact reconstruction is sometimes not possible from simple, regularized, one pixel thick skeleton.

3.2.1 *Marching Front* Skeleton (MFS)

Marching Front Skeleton (MFS) algorithms *[123]* simulate the *grassfire transform* [124], of Blum's first Medial Axis definition. The object boundary is propagated iteratively inwards. The points where two or more fronts collide constitute the object's skeleton. There are two kind of techniques following this approach, those based on morphological thinning and those based on curve evolution.

3.2.1.1 MFS based on morphological thinning

The main idea of this technique is to peel off the object boundary curve iteratively, removing them like the layers of an onion (see fig. 3.8). Morphological operators are used for its computation [95, 125] [126, 127] . It is a simple technique which produces connected and complete skeletons, forces

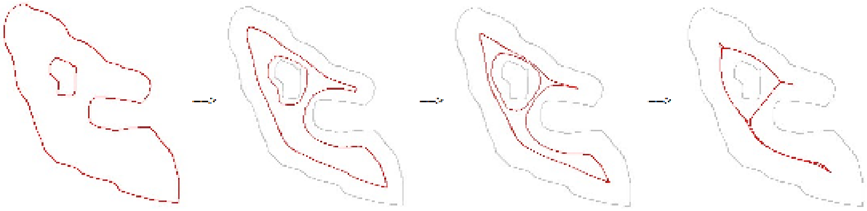


Figure 3.8: **Thinning**. Thinning iteration example. Red lines correspond to the pealed shape at each step.

the skeleton to be placed in the middle of the object, preserves the shape and topology of the original object, and in general produces one pixel wide skeletons [128], although some approaches ensure one pixel width skeletons [129]. One of the main difficulties of this kind of technique is the definition of the detection and stopping criteria to avoid either overbranching or branch shortening. On the other hand, and due to the characteristics of the morphological operators, it is not robust under Euclidean transformations, needs not to be homotopic and can produce branches wider than one pixel. The sequential homotopic thinning does not have the previously mentioned disadvantages but its computation complexity is one order higher than that of the technique described in [95]. In general, they are computationally expensive algorithms, Therefore, efficient parallel implementations [130, 131, 132, 133] have been proposed.

3.2.1.2 MFS based on *curve evolution*

This technique models the evolving curve of the grassfire transform from the Hamilton-Jacobi equations using a partial differential equation, the Eikonal equation, defined by:

$$\|\nabla u\,(x)\| = F\,(x)\,,\ x \in \Omega \tag{3.1}$$

subject to $u\mid_{\partial\Omega}= 0$, where Ω is an open set in $\mathbb{R}^2$ with well-behaved boundary, $F\,(x)$ is a positive valued function ,∇ denotes the gradient and $\|\cdot\|$ is the Euclidean norm. Here, the right-hand side $F\,(x)$ is the input data. Physically, the solution $u\,(x)$ is the shortest time needed to travel from the boundary $\partial\Omega$ to x inside Ω, with $F\,(x)$ being the time cost (not speed) at x.

In practice this approach raises nontrivial issues of flow numerical discretization, and the treatment of the evolution singularities detected . The first proposed method using this technique [134] represented the object boundary as an active contour, which is partitioned at locations of positive curvature maxima corresponding to the medial axis end points. The various segments of the active contour then propagate inwards driven by a potential function modeled by the negative gradient of the Euclidean distance function the boundary. The active contours slow down in the vicinity of the medial axis, where the magnitude of the numerical gradient is small. Other techniques also use splines for the shape boundary description [135, 136].

In general, this approach is robust under Euclidean transformations, but requires a functional description of the shape boundary. Moreover, it requires to fit a curve to the original contour, increasing the complexity and boundary noise sensitivity. The quality of the skeleton is proportional to the quality of the boundary curve description functions [123, 137]. Algorithms based on this approach, can be efficient, with a computational cost of $O\left(n \log n\right)$ on the number of shape pixels n.

3.2.2 Distance Transform Function (DT) Skeletons

The Distance Transformation function (DT) of a binary digital image assigns to each pixel the minimum distance to a shape boundary curve. Figure 3.9 presents a pictorial representation of the DT computed on the binary image of a hand. The initial algorithms for the computation of this transformation had a high computational cost, because the need to compute the Euclidean distance between all shape points to the boundary. Several alternative metrics have been proposed to alleviate this computational burden, like Manhattan, Chessboard, Chamfer,... [138, 139, 140], which are approximations of the Euclidean distance. Unfortunately, this metrics are orientation dependent, therefore this approximations are less robust than the Euclidean. ,There are exact methods to compute the Euclidean Distance Transform function with complexity $O\left(n\right)$ where n is the number of pixels in the image shape region [141, 142]. Additional information about the Distance Transform Function is presented in section B.1 of the appendix B.

The main idea to obtain the shape skeleton from the DT consists of finding the ridges in the DT function, i.e. the location of the local maxima. This kind of skeletonization procedures can be very efficient like the one proposed in [4] that has $O(n)$ time complexity. In general, the DT based skeletons permits the reconstruction of the original shape. However, two pixel wide branches can appear, and the resulting skeletons are not guaranteed to be

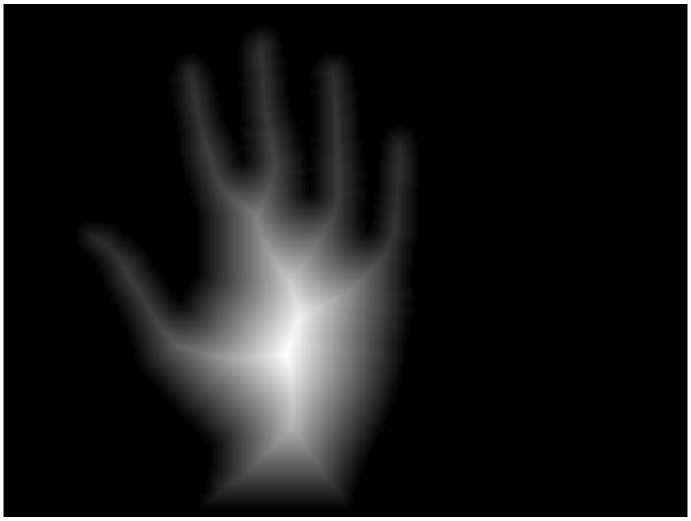


Figure 3.9: Visual representation of the **Distance Transform** of an open hand shape. The gray level indicates the DT value. Brighter corresponding to higher DT value. A contrast enhancement has been performed on this image.

connected, which exceptions [143].

There are examples of hybrid approaches where the grassfire transform is simulated on the Distance Transform function [134], using an active contour which shrinks to obtain the skeleton using the Distance Transform function as energy function. This approach guarantees connectivity, but it is difficult to obtain a one-pixel wide skeleton.

3.2.3 Voronoi Skeleton

The Voronoi Tessellation is a partition of the space into convex regions on the basis of a set of points, called Voronoi Sites, so that every point in a Voronoi Region (called Voronoi polygon) around a Voronoi site is closer to that Voronoi site that any other Voronoi site. The Voronoi Edges are the boundaries between Voronoi Regions. The Voronoi Tessellation if further described in section B.3 of appendix B.

Let us consider a shape $F \subset \mathbb{Z}^2$, and $C \subset \mathbb{Z}^2$ the set of points in the boundary curves of the shape, ($C \subset F$), after computing the Voronoi Tessellation of image domain assuming the set $V_{sites} \subset C$ as the Voronoi sites, the part of the Voronoi Edges lying inside the shape F becomes a better approximation of the Medial Axis of F, as the number of points in V_{sites} increases. When the number of Voronoi sites equals C, the Voronoi skeleton becomes the Medial Axis [144][145] [146]. Formally, Voronoi Skeleton is defined as follows:

$$S_v = \bigcup_{i,j \in V_{sites}} \left[s_{ij} \cap F \right], \tag{3.2}$$

where F is the set of pixel sites (points in a continuous space formulation) belonging to the shape and s_{ij} is a Voronoi Edge between Voronoi regions $V(i)$ and $V(j)$ defined by Voronoi sites v_i and v_j, respectively. Because Voronoi edges are connected and one pixel thick, the skeleton computed using this procedure is always connected and one pixel thick. Unfortunately, since any new boundary pixel generates a new Voronoi polygon, which in turn generates new Voronoi edge, this skeletonization procedure is specially sensitive to noise in the boundary. The Voronoi Tessellation can be computed in time $O(n \log n)$ [147], but an optimal lower bound $O(n)$ has been proved theoretically feasible [148], with n being the number of Voronoi sites $v_i \in V_{sites}$. For convex polygons the computational cost has been proved to be $O(n)$ [149].

The duality relationship between the Voronoi Tessellation and the Delaunay Triangulation [150] (see section B.3), has been exploited to propose a pruning procedure for skeletons which can be computed in $O(n)$. Each Delaunay edge is crossed by a Voronoi edge, the efficient storage of this information [151] leads to a pruning procedure based in the partition of the Delaunay edges using the concepts of *crust* and *anti-crust*. If a Delaunay edge defines a circle that is empty of Voronoi sites, then the Voronoi edge will cross the Delaunay edge, passing entirely through that circle, and the Delaunay edge will be a portion of the crust according to the criterion in [152]. As can be appreciated in figure 3.10, the set of Delaunay edges belonging to the crust form the polygonal representation of the shape boundary whose vertices are the points selected by its subsampling for the Voronoi Tessellation computation. The rest of the Delaunay edges constitute the anti-crust. The Voronoi edges crossing the Delaunay edges in the anti-crust form the skeleton of the shape, divided into the external (out of the shape) and internal (inside the shape) skeleton, by the Voronoi edges crossing the crust. Only the internal skeleton is of interest for shape representation.

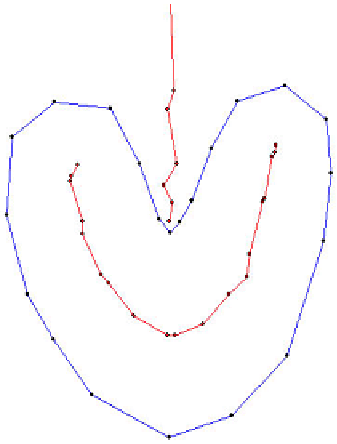


Figure 3.10: **Crust and anticrust.** Crust in blue, composing a polygonal approximation of the shape. It divides the Voronoi skeleton into exo and endo skeletons in red. Image extracted from [3].

For a basic but complete Voronoi skeleton computation and pruning algorithm example the reader is referred to the research work in [153].

3.2.3.1 Shape Boundary Curve Sampling for Efficient Voronoi Skeleton Computation

Since the computational cost of the Voronoi Tessellation depends on the number of boundary points considered, it is natural to use a subsampling of the object boundary as the Voronoi sites [154]. This subsampling, which can be performed following many criteria and density, conditions the degree of approximation to the true skeleton. In fact, only some subsampling procedures lead to a skeleton which includes all the relevant geometry of a shape. In this section several subsampling methods are described as well as the conditions under which produce skeletons representing the most important parts of the original shape. Some theoretical foundations have been extracted from [155].

Definition 3.2.1. An *uniform sampling* D of the shape boundary curve C is defined as a sampling so that the distance from any point on the curve C to the nearest sample in D is smaller than a constant δ.

$$\forall c_i \in C \exists d_j \in D \,|\, \|c_i - d_j\| \leq \delta \tag{3.3}$$

Uniform sampling produces skeletons with the same level of detail in the whole skeleton, since it has more density (i.e., pixels per area unit) in the shape parts with more curvature, which also require more Voronoi sites [156, 157, 158]. These results are in good correspondence with perceptual evidence resulting from psychophysical investigations [159].

r-Regular Shapes

This kind of shapes were defined in [145, 160]. These works are concerned about fitting a curve or surface to a scattered point set, under the assumption that it follows a r-regular shape, using the Delaunay triangulation and the Voronoi diagrams. Some of their results are relevant to determine the appropriate subsampling of the shape boundary curves for Voronoi skeleton computation.

Definition 3.2.2. r-regular shape[161, 160]: Let B_0 be a circle of radius one, and rB_0 denote a circle of radius r. A shape F is said to be r-regular if it is morphologically invariant to opening and closing with a disk of radius r, with $r > 0$.

$$F = (F \ominus rB_0) \oplus rB_0 = (F \oplus rB_0) \ominus rB_0 \tag{3.4}$$

r-regular shapes have several interesting properties:

1. Each boundary curve point of a r-regular shape has a tangent and a radius of curvature greater than or equal to r.

2. The boundary of a r-regular shape divides any circle with center on the boundary and radius $2r$ into two connected components.

3. From the previous property, it follows that if the shape boundary is uniformly sampled with density $\delta < 2r$, the polygon induced by the sampling retains all the topological properties of original shape boundary curves.

4. Any circle passing through three distinct boundary points has radius greater than r.

According to [160] if we uniformly sample a r-regular shape with a sampling density so that the distance between neighbor samples $\|x_i, x_{i+1}\| < 2r$, then the Voronoi edges inside the shape are topologically equivalent to its skeleton. Another conclusion of that work is that, given a shape boundary sampling, if the Delaunay edges dual to the Voronoi edges crossing the shape boundary curve, form a topologically correct approximation of the shape boundary curves, then the Voronoi skeleton is also topologically correct respect to the medial axis of the shape.

λ-Medial Axis and Uniform Sampling

Definition 3.2.3. λ-*Medial Axis* [162]: the set of Medial Axis points m for which $\beta(m) \geq \lambda$, where $\beta(m)$ is the radius of the minimum circle which includes all the generative points of m. For Medial Axis points with two generative points c_i and c_j, it is half of the distance between them $\beta(m) = \frac{1}{2}\left\|c_i - c_j\right\|$. For Medial Axis points with more generative points this value will be half of the largest distance between two generative points. The λ-Medial Axis varies with the scale of the shape.

The theorem 3.2.5 proved in the work by Chazal and Lieutier[162] relates the approximation, in the sense of the Hausdorff distance, between curves to the approximation of their corresponding Medial Axis. This theorem states that if the Hausdorff distance $d_H(C, D)$ between a boundary curve C and some other set D is at most a constant $\delta = O(\lambda^4)$, then the Hausdorff distance between the λ-medial axis of C and the λ-medial axis of D is $O\left(\sqrt{\delta}\right)$. Let us recall the definition of the Hausdorff distance:

Definition 3.2.4. Hausdorff distance: It is a measure of the distance between two sets A, M. If we define the minimum distance between a point x to the set A as $d_h(x, A) = \min_{y \in A}\|x - y\|$, then the Hausdorff distance between sets A and B, $d(A, B)$ is defined as,

$$d_H(A, M) = \max\left[\max_{x \in A} d_h(x, M), \max_{y \in M} d_h(y, A)\right] \tag{3.5}$$

Theorem 3.2.5. *[162]. The set M consisting of the closure of the points v of the Voronoi diagram (including points in edges and 2D faces), for which the distance between two of the nearest samples (the generative points) to v is at least δ, is an approximation of the λ-medial axis of C, with $\lambda = O\left(\delta^{\frac{1}{4}}\right)$, and where the Hausdorff distance between M and the λ-medial axis is $O\left(\delta^{\frac{1}{8}}\right)$.*

Therefore, the λ-medial axis is the stable part of the skeleton. This result also confirms the convergence of the Voronoi skeleton towards the exact medial axis as $\delta \to 0$. The specific value of δ where convergence is achieved is addressed in [163].

Definition 3.2.6. λ-*offset surfaces of* C: They are the λ iso-surfaces of the distance function:

$$\{x \| d_h(x, C) = \lambda\} \tag{3.6}$$

Where C is the shape boundary curve. When C is smooth and λ is small, there are two connected components of the λ-offset surface, one inside the shape and the other outside the shape. At larger values of λ one or more components overlap.

The values at which the topological changes of the offset surfaces occur are the *singular values* of this distance function. Zero is a singular value. In [163] they define the *weak feature* size as the next smallest singular value. In [163] it is proved that the λ-Medial Axis is homotopy equivalent to the medial axis when λ is less than the weak feature size.

γ-Medial Axis and Non-Uniform (Scale-Invariant) Sampling

Definition 3.2.7. γ-*Medial Axis*: The set of medial points m such that $\gamma(m) \geq \gamma$ for some constant γ.

$$\gamma(m) = \max_{p_1, p_2 \in B_{gen}(m)} \angle(p_1, m, p_2) \tag{3.7}$$

with $B_{gen}(m)$ being the set of generative points of m, and $\angle(p_1, m, p_2)$ is the angle defined by points p_1, p_2 and m. The γ-Medial Axis is scale invariant, in contrast to the λ-Medial Axis.

Definition 3.2.8. ε-sample [164]: It is a subsampling D from a shape boundary curve C so that for every point $c_i \in C$ the minimum distance from c_i to any point in D is at most $\varepsilon f(c_i)$, where $f(c_i) = d_h(c_i, MAT(C))$, and $MAT(C)$ is the medial axis transform of the shape enclosed by the curve C.

There are several algorithms that guarantee an approximation quality of the γ-Medial Axis for $\varepsilon \leq \varepsilon_0$, where ε_0 is a constant independent of any particular curve C. The main problem of this sampling method is that it is only useful for smooth surfaces, because in a sharp corner we have that $f(c_i) = 0$, which implies an infinitely dense sampling.

3.3 Skeleton regularization

Skeleton regularization is a any kind of procedure aiming to produce more stable and compact skeletons. These procedures start from the skeleton computed at a certain point, and either try to remove spurious branches of the skeleton (pruning) or to detect unstable inner branch configurations due to shape concavities.

3.3.1 Skeleton Pruning

Most of the skeletonization procedures are highly sensitive to noise in the boundary, which is usually originated by the segmentation procedures and the discrete image domain. This noise produces many spurious branches in the skeleton corresponding to non-significant parts of the boundary. An example of this overbranching is shown in fig. 3.11. The purpose of the pruning stage, as a post-processing or embedded in the skeleton computation procedure, is to remove these branches, in order to preserve only the stable parts of the skeleton. A preprocessing of the shape boundary curves to denoise it can also be applied, like boundary smoothing [165, 166], but they may remove both noise and salient boundary features. It gives less control on the final skeleton than pruning .

The general process of the pruning algorithms is as follows: extreme points on the skeleton are successively removed until a stopping condition is satisfied. This condition may be (1) a threshold on the difference between the initial shape and the shape reconstructed from the simplified medial axis [145, 167, 168, 169], or (2) it may be based on an estimate of the stability of portions of the medial axis [170, 171, 172, 173, 174, 175].

In the first case this measure can be taken in two different ways, by means of computing the area or the elongation of the portion of the skeleton lost in the reconstruction from the pruned skeleton.

In the second case a local salience measure is computed for each skeletal point. Two common measures are:

- The minimum distance to the boundary or the radius ρ of the maximal disk corresponding to the skeletal point v_{sk},

$$\rho\left(v_{sk}\right) = \min_{p_b \in C}\left(\left\|v_{sk} - p_b\right\|\right) = DT\left(v_{sk}\right) \tag{3.8}$$

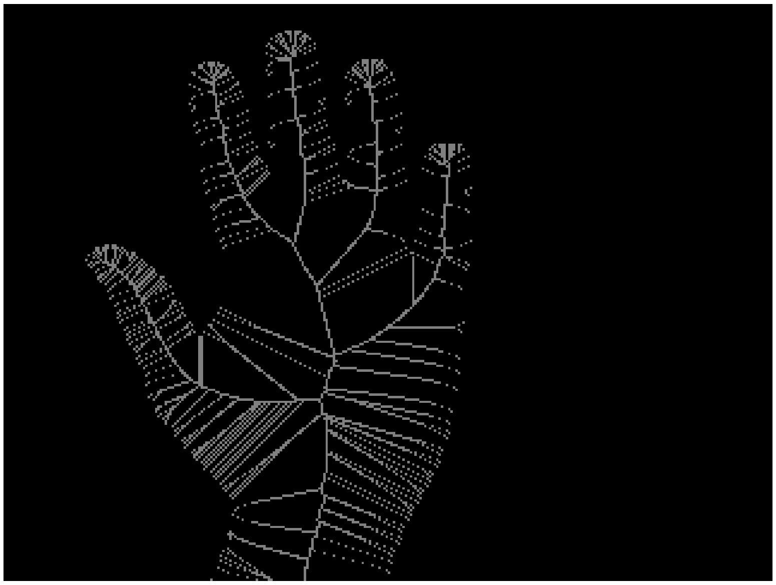


Figure 3.11: **Skeleton based in DT**. Raw overbranched skeleton obtained using an algorithm based on the Distance Transform approach [4].

- The maximum angle formed by the skeletal point v_{sk}, and two of its generative points,

$$\theta\left(v_{sk}\right) = \max_{a,b \in B_{gen}\left(v_{sk}\right)} \angle\left(a, v_{sk}, b\right) \tag{3.9}$$

where C corresponds to the shape boundary curves and $B_{gen}\left(v_{sk}\right)$ corresponds to the generative point set of the skeletal point v_{sk} . It is generally accepted that pruning methods should have the following properties:

1. It should preserve topology (homotopy type).
2. It should be continuous, i.e., small differences in the significance measure should result in small changes to the computed skeleton.
3. The significance measure should be local on the Medial Axis locus.

Some pruning approaches

Shaked and Bruckstein [165] present a systematization of pruning in the continuous space, including acceptable pruning methods in terms of preservation of topology, being well-conditioned and locality of significance. According to their findings pruning can only be done from the skeleton end points inwards, and the decision of whether a pixel must be removed or not has to be performed by local operations. Significance measures of the skeleton points are related to the area or elongation of the portion of the original shape lost in the reconstruction of the skeleton when a skeleton point is removed, and the boundary length corresponding to a skeleton branch.

In [175] a saliency measure is proposed for the for Voronoi edges belonging to the Voronoi Skeleton. They use the difference between the distance along the shape boundary curve and the Euclidean distance between the two generative points of the Voronoi edges.

A recent pruning approach was presented by Bai et al. in [6]. It makes use of the Discrete Curve Evolution (DCE) procedure (explained in section B.2 of appendix B), which approximates the shape boundary curve by a simple polygon which represents the most relevant geometry of the shape, whose vertices constitute a subsampling of the shape boundary curve. They compute the DCE of the shape boundary curves until certain, application specific, ending criteria is met, obtaining the polygon P_{DCE}. Then, the convex point set of the DCE boundary downsampling is used to define a convex polygon P^C_{DCE}. The set of generative points of a skeleton point p_i are the shape boundary points at minimum distance from p_i, denote it $Q = \{q_o, ..., q_n\}$. Considering the shape boundary curve points as a point sequence C, indexed by an arbitrary origin and direction, and P^C_{DCE} a partition of it, which can assign the closest edge of the polygon P^C_{DCE} to each point in $p \in C$, using a function $Edge(P^C_{DCE}, p)$, the idea consists of removing any skeleton point whose generative points do not belonging to, at least, two different edges of the P^C_{DCE} polygon,

$$\exists q_i, q_j \in Q \left| q_i \neq q_j \wedge Edge\left(P^C_{DCE}, q_i\right) \neq Edge\left(P^C_{DCE}, q_j\right) \right. \Rightarrow p_i \in S_{DCE}$$

where S_{DCE} is the pruned skeleton.

This pruning method is able to remove complete spurious branches without shortening any important branch. It is also valid for skeletons computed using any of the techniques described in this section while two criteria are met: (1) every skeleton point must be the center of a maximal disk and (2) the generative points in the shape boundary curves must be available. We will use this pruning method as the benchmarking for our own proposal.

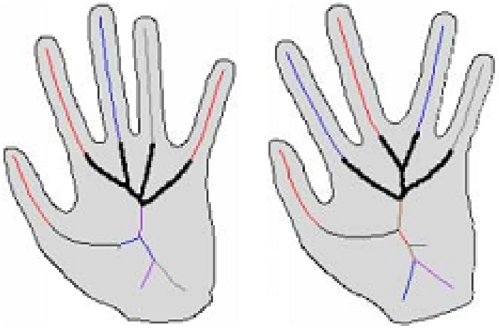


Figure 3.12: **Skeleton ligature example**. The different separation of the two middle fingers in both images induces a strong topological change in the related skeleton graph.

3.3.2 Ligature Analysis

Another source of skeletal instability , is that shape boundary concavities produce so-called ligature branch segments whose points are related only to the concave boundary points[176]. When a ligature segment spans the entire branch, it is called a ligature branch [88, 176]. Small positional changes of such concavities can cause significant ligature branch structural changes of the skeleton (see fig. 3.12), which ultimately give rise to significant differences in their corresponding graphs. August et al. [177] show that these internal skeleton instabilities cannot be removed by boundary smoothing alone.

Removing this branches improves the stability of the skeleton, at the cost of reducing the accuracy of the shape representation. This problem has been dealt in [178, 179], focusing in shock graphs. In both cases the procedure first detects skeletal segments with low contribution to the shape, then removes those branches and finally reconnects the remaining branches among them.

3.4 Skeletons in Shape Recognition

The skeleton is a pixel set, which is not always guaranteed to be a single connected component, except in specific cases. Most of shape classification techniques based in skeletal representations (e.g. Medial Axis, Shocks, Symmetry Sets...) make use of a higher abstraction layer than that of the set of pixels from the skeleton, which includes enough information to reconstruct the original skeleton, and even to make a correspondence between skeleton branches and shape parts. Moreover, this abstraction provides robustness to translation, rotation and scaling, and shape part deformations like articulations, elongations or widening.

Most of the shape recognition approaches use some kind of structural characterization of the skeleton. The structural representation captures the internal relations between skeleton parts identified by union and ending points. Both local and global information is stored.

The most usual structural representation is a graph where links represent the skeleton branches and nodes represent both the end points and joint points in the skeleton. In the thesis work [123], several different techniques for shape recognition using skeletons and graph representations are described.

3.4.1 Skeletal Graphs for Shape Recognition

When the skeleton is represented as a graph, the shape recognition is done through a graph matching algorithm. A review of graph matching is given in Appendix B, section B.4. The skeleton graph representation arises naturally from a classification of the skeleton pixels into *joint points*, *branch points* and *end points*.

- A joint point is the intersection point of three or more branches of the skeleton.
- A branch point is a point belonging to a branch.
- An end point is the final point of a branch which is not connected with any other branch.

To formalize these definitions mathematically, assuming a 8-neighborhood connectivity, we have that $N(p_{x,y})$ is the sum of the neighboring pixels of $p_{x,y}$, where pixels belonging to the shape have a one value and pixels not belonging to the shape have a zero value. Considering a one pixel thick skeleton, then, a branch point is defined as,

$$Branch := \{q \in S \mid N\left(q\right) = 2\} \tag{3.10}$$

where q is a image pixel and S is the skeleton of the shape. A joint point is defined as,

$$Joint := \{q \in S \mid N\left(q\right) \geq 3\} \tag{3.11}$$

and an end point is defined as,

$$End := \{q \in S \mid N(q) = 1\} \tag{3.12}$$

In a skeleton's graph (V, E), the joint points and end points are the vertices of the graph V, and the connected set of branch points between them form the edges of the graph E. These graphs are undirected. When the shape has no holes in it, the skeleton has no cycles, and the corresponding graph is acyclic. Each hole in the shape produces a cycle in the graph. Both, the vertices and the edges in the graph usually have attributes. Some authors call this representation *attributed skeletal graph* (ASG) [180]. This is the usual graph representation method, but there are exceptions like in [123], where graph nodes correspond to branches in the skeleton and graph links represent the adjacency between those branches. Another alternative graph representation are Shock Graph, described in the appendix B, in section B.5.

After the transformation of the skeleton into a graph, the shape recognition problem is turned into a graph matching problem, where the classification of a given sample graph G_s is formulated as the search of the minimum distance between G_s and one from a set of shape template graphs $G_T = \left\{G_{T_0}, ..., G_{T_p}\right\}$ corresponding to the shape classes.

Many different attributes and matching algorithms have been proposed for the ASG. In [180] graph nodes are assigned the DT value of their skeletal point normalized to the maximum value, and graph links, composed by the skeletal point set corresponding to that skeletal branch, are assigned a set of features: variation of the curvature along the branch; orientation of the branch normalized to the orientation of the whole skeleton, size of the branch normalized to the size of the whole skeleton; strength of the branch measured as the ratio of the branch length to the Euclidean distance between the branch endpoints; distance function variation along the branch; size of the shape part corresponding to the branch compared to the object size. Then, the ASG are matched using a *graduated assignment* algorithm.

Luo and Hancock [181] proposed a purely structural approach for inexact graph matching between two point sets, which is formulated as a maximum-likelihood estimation. It can only handle small rotation and non-rigid shape deformations. Zhu and Yuille [182] constructed object skeleton model in terms of the principal deformation modes for object recognition. The algorithm is sensitive to noise on primitive segmentation and computationally demanding with multiple parameters to be tuned.

The recent work in [183] presents a complex set of attributes for both the ASG nodes and branches, combined to obtain local and global features, and also a shape matching technique using this specific ASG. This work uses the Multiresolution Gradient Vector Field (MGVF) skeleton framework for the computation of the skeleton, and the special properties of the skeleton obtained with this framework.

An alternative use of the skeleton graph for shape matching is presented in [184]. There, the shortest path between end nodes is computed, and that spatial information including the DT value is used for matching, by means of an algorithm called Optimal Subsequence Bijection (OSB), which performs the elastic matching of two sequences of different lengths m and n. More specifically, for two finite sequences of end nodes of skeletons $a = (a_1, ..., a_m)$ and $b = (b_1, ..., b_n)$ the goal is to find sub-sequences $a\prime$ of a and $b\prime$ of b such that $a\prime$ best matches $b\prime$. Skipping (not matching) some elements of a and b is necessary, because both sequences may contain some outlier elements. Another example of different use of the skeleton graph is presented in [185]. There the skeletal graph representations is further simplified into a string representation, considering the end and joint nodes adjacent to each joint node. Then a string matching algorithm is used, which reduces the complexity of the matching stage. Unfortunately they do not represent any local information about skeleton branches, apart from the connectivity between them, therefore losing shape descriptive capabilities.

When the shapes to be recognized do not have holes, i.e. there are not cycles in the graph definition, and a node ordering is defined the graph representation in the previous section can be turned into a tree representation [185]. Tree matching complexity is lower than the graph matching complexity [186]. Tree matching procedures, being an special case of graph matching can also be performed as an isomorphism or homomorphism.

3.4.2 Shock Graphs for Shape Recognition

The basic description of this skeleton representation, which is an special case of graph representation is covered in section B.5 of Appendix B. In this section several work extending the initial definition are described.

The matching algorithms of the shock graph representations are formulated as assignment algorithms [187], finding subgraph isomorphism [188, 189] or edit-distance algorithms [190]. The shock segmentation and matching algorithms are computationally expensive and sensitive to noise. A further simplification of the Shock Graph into a rooted tree structure, called Shock Tree, is usually employed to reduce the matching procedure complexity. Unfortunately, this representation is not valid for shapes with holes in them. An alternative shock graph representation is introduced in [191], where shock graph nodes nodes correspond to the shock sequence forming a skeleton branch, and links represent the adjacency between branches.

3.4.3 Skeleton Abstraction

A set of skeleton metrics describing the shape can also be computed in order to perform a loose shape classification. This kind of measures do not permit accurate shape recognition, but they are usually easier and less complex to compute.

Skeleton topology can be described [128] using:

- The pixel or point number (i.e., end points, branch points and joint points).
- The distance between end and joint points and the center of mass in the shape.
- The angle between the center of mass, the farthest node and the actual node.
- The number of neighbor nodes and the length of their branches.

This information or a subset of it can then be used to define the signature of the shape for shape matching. The comparison of this signature or part of it information can be used as a first shape matching step, in order to guide the graph and tree matching algorithms performed in a second step, and also to distinguish between generic shape classes in a database, before a more exhaustive class specific matching.

3.5 General framework

In this section we give a general framework of a shape recognition process based on Skeletons and their corresponding graph representation. Furthermore, we will be advancing some of our choices and proposals described in the next chapter. From the previous review in this chapter, the best general choice, suitable for connected shapes which can have holes on them, robust under deformations on the shape boundary curves and computationally efficient would be composed of the next processing pipeline:

1. Initially the shape boundary curves would be subsampled by some kind of procedure. We will discuss the Discrete Curve Evolution (DCE) technique as our own choice for this step. As stated in [168, 6, 192] the boundary points with highest curvature can be used to obtain the most salient branches of the skeleton. The purpose of this processing stage is to limit the effect of noise in the boundary curves while keeping the most portion of the shape characteristics.

2. Compute the shape skeleton using a DT based approach like [4]. Or take advantage of the shape boundary curve subsampling, which reduces the computational cost of skeletonization algorithms based in the Voronoi Tessellation and its dual Delaunay Triangulation [114] (Our own choice).

3. Skeleton regularization. During this research work, a variation of the skeleton pruning method proposed in [6] has been developed. It offers reasonably good results with a low computational cost, but many other pruning algorithms are available. Ligature branch removal has been initially discarded due to its computational cost.

4. Graph representation of the skeleton. Build an Attributed Skeletal Graph (ASG). The presence of holes does not usually permit tree representations, due to the presence of cycles in the graph. Some research should be devoted to the representation of graph cycles in order to permit more simple representations which also require more simple matching procedures. Alternative representations would include rooted trees and strings.

5. Class template indexing. In order to efficiently classify shapes into classes, as their number increases, indexing algorithms should be employed to avoid the computation of the similarity measure between the sample skeleton and all the class templates in the database.

As final comment, the use of skeletons for shape recognition would only be recommended when shapes are amorphous, like biological shapes, deformations, articulations and occlusions are present, and high accuracy is required.

3.6 Conclusions

This chapter has presented skeletons of objects in binary bidimensional images. The skeleton is a thinned version of an object, which lies in the middle of it. It is suitable for shape representation and recognition, and when the Distance Transform value is available for every skeletal point it also permits the reconstruction of the shape. It is a part based shape representation technique which also represents the relationship among parts. Therefore it is robust under shape articulations and occlusions. We have described the main Skeleton computation approaches, with an emphasis in Voronoi approaches which will be used in the core contribution of the thesis. All the methods lack robustness against noise on the boundary in the discrete space and have high computational complexity. We have recalled some regularization methods aiming to speed up the algorithms, reduce sensitivity to noise and provide more stability. Finally, shape recognition based on the skeleton has been introduced as an attributed graph matching problem, with variations like Shock Graphs/Trees, and some exceptions like the definitions of shape signatures based on the skeleton.

This PhD Thesis work has been carried out having in mind hand gesture recognition for a hand-based interaction with Multimedia Interactive tabletops, described in chapter 2.7, as the target application. In that context the system response must be obtained in real-time. The main goal is to recognize the silhouette of a hand, segmented without the presence of noise. The hand is an object which can have articulations, deformations, and has a limited detail level. The most adequate technique combination for the stable recognition of hand shapes, considering its specific characteristics.

Chapter 4

Efficient and Stable Voronoi Skeleton

In this chapter we present the main contribution of this PhD work, the definition of an efficient skeletonization algorithm, which performs in real time and is more stable than previous algorithms found in the literature. This chapter will be concentrated in the formal definition of the algorithm, and the proof of several key results that back its performance. We prove that only the extreme points of Voronoi segments needed to be tested in order to discard the entire segment. This result justifies a simplification that improves the efficiency of the algorithm to the point of allowing real time processing. The next chapter will be more concerned with the empirical validation of the algorithm improvement over other algorithms in terms of stability and patter recognition performance.

Section 4.1 gives some introductory remarks and defines some notation. Section 4.2 describes the algorithm. Section 4.3 contains the proofs of the main theoretical results that support the algorithm. Section 4.4 gives some conclusions and discussion.

4.1 Introduction and notation

A skeleton computation and pruning procedure is introduced in this chapter which permits obtaining simple and stable skeletons efficiently in bi-dimensional binary images. The skeleton computation is performed using the Voronoi Tessellation and the pruning procedure is divided into two stages, the first one removing the Voronoi edges not contained inside the shape, and the second one is an enhanced version of the pruning procedure presented in [6], taking advantage of certain characteristics of Voronoi Skeletons.

An image is a function $I : D \rightarrow \mathbb{R}$, and a shape is a connected set of pixel sites $F \subseteq D$ identified through some segmentation procedure, which can include holes in it. This segmentation procedure usually adds some kind of noise to the shape boundary. In the continuous case $D \subset \mathbb{R}^2$ a shape is a closed region of the image domain. In the discrete case, the image plane is tessellated into square pixels $D \subset \mathbb{Z}^2$.

Definition 4.1.1. In this discrete space the shape F is defined as a connected component:

$$F = \forall p, q \in F\ \exists G = \{g_1, ..., g_n\} \subset F\ s.t.\ (g_1 = p) \wedge (g_n = q) \wedge (\forall g_i \in G\, ||g_i, g_{i+1}|_A)\,, \tag{4.1}$$

where $|a, b|_A$represents the adjacency relationship between pixels a and b (see definition 3.1.7).

.

Definition 4.1.2. We consider the shape boundary curves denoted C, which can be constituted by one or several connected components of one-pixel width curves $C = \{C_1, ..., C_n\}$. It is formally defined in the continuous case as:

$$C = \left\{p \in \mathbb{R}^2\, |\forall \xi > 0, \exists a \in F, \exists b \notin F\, s.t.\, \|p - a\| < \xi \wedge \|p - b\| < \xi\right\}\,, \tag{4.2}$$

where p, a and b are point in the image domain, and F is the region of the image domain corresponding to the shape.

.

Definition 4.1.3. In the discrete image domain case $C \subset \mathbb{Z}^2$ each of the boundary curves is represented as a sequence of connected (8-connected in this case, see definitions 3.1.7 and 3.1.8) pixels $C_i = \{p_1, ..., p_n\}$, so that p_{j+1} is the next neighbor pixel of p_j $\left(\forall 1 \leq j < n\ \ |p_j, p_{j+1}|_A\right)$. This sequence corresponds to visiting the pixels in the boundary curve in clockwise or anticlockwise direction, starting from an arbitrary position. Each of the boundary curves is closed so that $|p_n, p_1|_A$. The boundary condition is mathematically stated as follows:

$$\forall p_k \in C\left(\exists q \in F \wedge \exists r \in \overline{F}\ s.t.\ |p_k, q|_A \wedge |p_k, r|_A \wedge (p_k \in F)\right) \tag{4.3}$$

4.2 Skeleton Computation Algorithm

Our algorithm follows the next steps:

1. *Shape boundary subsampling:* The shape boundary curves are subsampled along their arclengths to obtain the set of Voronoi sites $V_{sites} \subset C$ which will be used to compute our Voronoi Tessellation. In our implementation and experiments an uniform subsampling is performed keeping one pixel out of four from the initially 8-connected shape boundary curves. Some restrictions of the set of pixels V_{sites} to guarantee a correct shape skeleton have been presented in section 3.2.3.1.

2. *Voronoi Tessellation computation:* The Voronoi edges $\{s_{ij} \mid i, j \in V_{sites}\}$ of the Voronoi Tessellation induced by V_{sites} are computed.

3. *Discrete Curve Evolution computation (DCE):* A Discrete Curve Evolution procedure (see section B.2 in Appendix B) is performed on the original shape boundary curves C, with a termination criterion of a minimum number of vertices for each boundary curve C_i. The external boundary curve minimum number of vertices is θ_E, while the boundary curves of the shape holes have always a θ_I number of vertices. In general $\theta_E \gg \theta_I$, because the external boundary curve is considered more representative for the shape description than the boundary curves of the shape holes and also because the length of the external boundary curve is usually bigger. A constant threshold can reduce the complexity of further steps for shape recognition, but also limits the maximal detail resolution of a shape. In our tests we used $\theta_E = 15$ and $\theta_I = 6$, for shapes in 320x240 resolution images.

4. *Pruning:* The spurious Voronoi edges of the external skeleton branches are removed in a two stage pruning procedure. The set of Voronoi edges is sequentially processed checking the next two conditions in order, preserving only the segments fulfilling both of them. Consequently, the final pruned skeleton is defined as:

$$S_{Vpruned} = S_{VA} - S_{VDCE} \tag{4.4}$$

   (a) *First Pruning Stage:* We only keep the set of Voronoi segments entirely contained inside the original shape F as the initial Voronoi Skeleton. This is an initial pruning procedure which was defined and described by [161]. Figure 4.2 shows that this initial pruning procedure removes most of the spurious branches in the skeleton, corresponding to the red branches in the figure. This pruning procedure is formally described as:

$$S_{VA} = \bigcup_{i,j \in V_{sites}} (s_{ij} \subset F) \tag{4.5}$$

(b) *Second Pruning Stage:* As a second pruning step, we apply a DCE based pruning procedure similar to [6] (described in section 3.3.1). First, the convex hull of the DCE polygon (i.e. only the sequence of convex points in the DCE subsampling $C_{DCE} = \{c_1^*, ..., c_{n'}^*\} \subset C$ are used to compose a polygon) is obtained $H_{DCE} = (E_{DCE}, C_{DCE})$, so that each edge $e_i \in E_{DCE}$ is associated to a subsequence of the original shape boundary curve $e_i = \{c_p, ..., c_{p+q}\} \subset C$, such that $c_p = c_i^*$ and $c_{p+q} = c_{i+1}^*$. For $e_{n'}$ the end points are c_1^* and $c_{n'}^*$. Therefore, there will be some j such that $v_i \in e_j$. The function $Edge\,(polygon, point)$ returns the identity of the H_{DCE} edge for a given Voronoi site. Then, the pruning criterion consists of removing the Voronoi segments s_{ij}, whose generative points (i.e., closest Voronoi sites) v_i and v_j correspond to the same edge in H_{DCE}. Mathematically, our pruning procedure can be defined as,

$$S_{VDCE} = \{s_{i,j}\,|Edge\,(H_{DCE}, v_i) = Edge\,(H_{DCE}, v_j)\}$$

where S_{VDCE} is the Voronoi DCE pruned skeleton.

Both requirements of the DCE pruning procedure listed in section 3.3.1 are met by the Voronoi skeletonization procedure, which are: (1) every skeleton point must be the center of a maximal disk and (2) the generative points in the shape boundary curves must be available.

A visual representation of the pruning stages is shown in figure 4.2. This figure corresponds to the Voronoi Tessellation on the figure 4.1. The red segments correspond to Voronoi segments removed by the first pruning stage. The yellow Voronoi segments correspond to Voronoi segments removed by the DCE pruning stage. The segments in green correspond to the final pruned skeleton. The resulting skeleton can be seen in figure 4.1.

The first pruning phase, described in the step (4.a) of the algorithm, which involves checking if a Voronoi edge is completely contained inside the shape (which can contain holes) can be performed efficiently because testing if both endpoints, i.e., Voronoi vertices, of a Voronoi segment are inside the shape is enough to guarantee that the whole segment is contained inside the shape. We prove formally this statement and its restrictions in section 4.3.

The second pruning phase can also be performed efficiently thanks to the definition of Voronoi segment. According to it, every point in a Voronoi segment shares the same pair of Voronoi sites, say v_i and v_j which are also their generative points, used by the DCE pruning procedure, and therefore it is enough to check the DCE condition once for the whole Voronoi segment.

The algorithmic pseudo code specification of the procedure is shown in 4.1, where $S_{Vpruned}$ is the shape's Voronoi skeleton obtained by our procedure and composed of Voronoi segments, A_I and $A_F$the end points or Voronoi vertices of a Voronoi segment s_{ij}. And the functions are defined as:

- $Subsample\left(C\right)$: Obtains a subsampling of the points in the original shape boundary curves C.
- $VoronoiTesselation\left(V_{sites}\right)$: Computes the Voronoi Tessellation on the V_{sites}point set, which includes obtaining the Voronoi segment set.
- $DCE(C)$: Computes the Discrete Curve Evolution procedure for each of the boundary curves in C.
- $P\left(DCE_{convex}\right)$: Is the polygon formed by the convex vertices of the polygon obtained by the $DCE(C)$ function.
- $EndPoints\left(s_{ij}\right)$: Obtains the end points, i.e. Voronoi vertices, of the Voronoi segment s_{ij}.
- $Shape\left(p\right)$: Returns true if the point p is contained in the shape, and false otherwise.
- $DCECrit\left(s_{ij}\right)$: Performs the adapted version of the DCE pruning procedure. It checks if the generative points v_i and $v_j$of the Voronoi segment s_{ij} belong to different edges of $P\left(DCE_{convex}\right)$. If so, the function returns true, and false otherwise.
- $ConvexHull(poly)$: Computes the convex hull of a polygon, returning another polygon.

Algorithm 4.1 Skeletonization procedure

```
$V_{sites} := Subsample\,(C)\,;$
$V := VoronoiTesselation\,(V_{sites})\,;$
$P\,(DCE_{convex}) := ConvexHull(DCE(C));$
foreach $s_{ij}$ in $V$ do
{
    $(\alpha_{ij}, \omega_{ij}) := EndPoints\,(s_{ij})\,;$
    if$((Shape\,(\alpha_{ij}))\ AND\ (Shape\,(\omega_{ij})))$
    {
        if$(DCECrit\,(s_{ij}))$
        {
            $S_{Vpruned} := S_{Vpruned} \cup s_{ij};$
        }
    }
}
```

The final Voronoi skeleton obtained by means of this algorithm $S_{Vpruned}$ is a connected set, simple and robust under noise. Being the Voronoi skeleton specially noisy with many spurious branches (also called hairs) and also highly affected by noise in the shape boundary curves, this result is remarkable. The external branches of the skeleton do not connect with the boundary curves, because the first pruning stage shortens the skeleton ending branches. But the shortening degree is low and limited as it can be appreciated in figure 4.3.

4.3 Theoretical support

One of the main speedups of the algorithm described above is the fact that we can consider Voronoi edges as a whole when testing their pruning conditions. For the first pruning stage, the Voronoi edge preservation can be decided looking only a its extreme points. In this section we elaborate the proof that testing if both endpoints or Voronoi vertices of a Voronoi segment are inside the shape is enough to say that the whole segment is contained in the shape. As this proof is not straightforward in the general discrete image domain

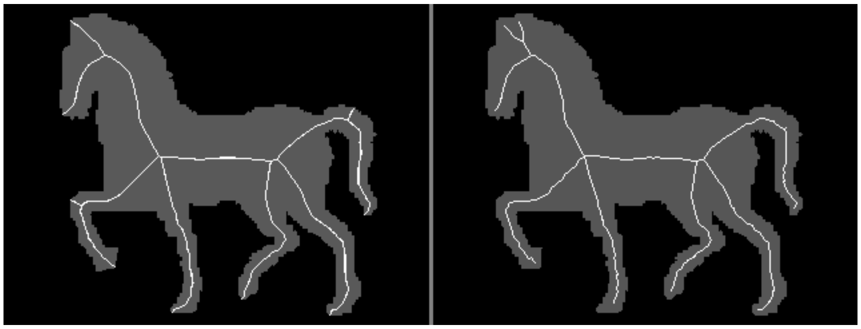


Figure 4.1: Left: skeleton computed using the Matlab implementation in [5]. Right: Voronoi based approach presented in this paper. The binary image is taken from [6].

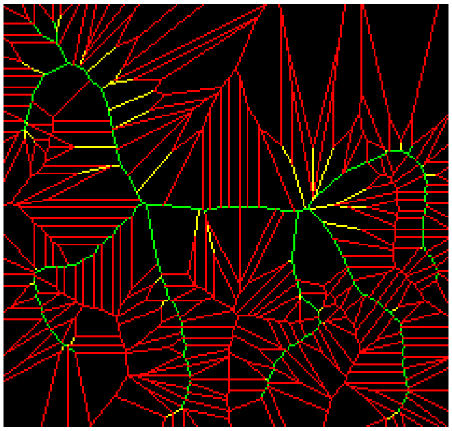


Figure 4.2: **Pruning procedure** stages of our algorithm of the image in Figure 4.1. Red: Voronoi segments removed in first pruning stage. Yellow: Voronoi segments removed after the second pruning stage, DCE pruning. Green: final skeleton.

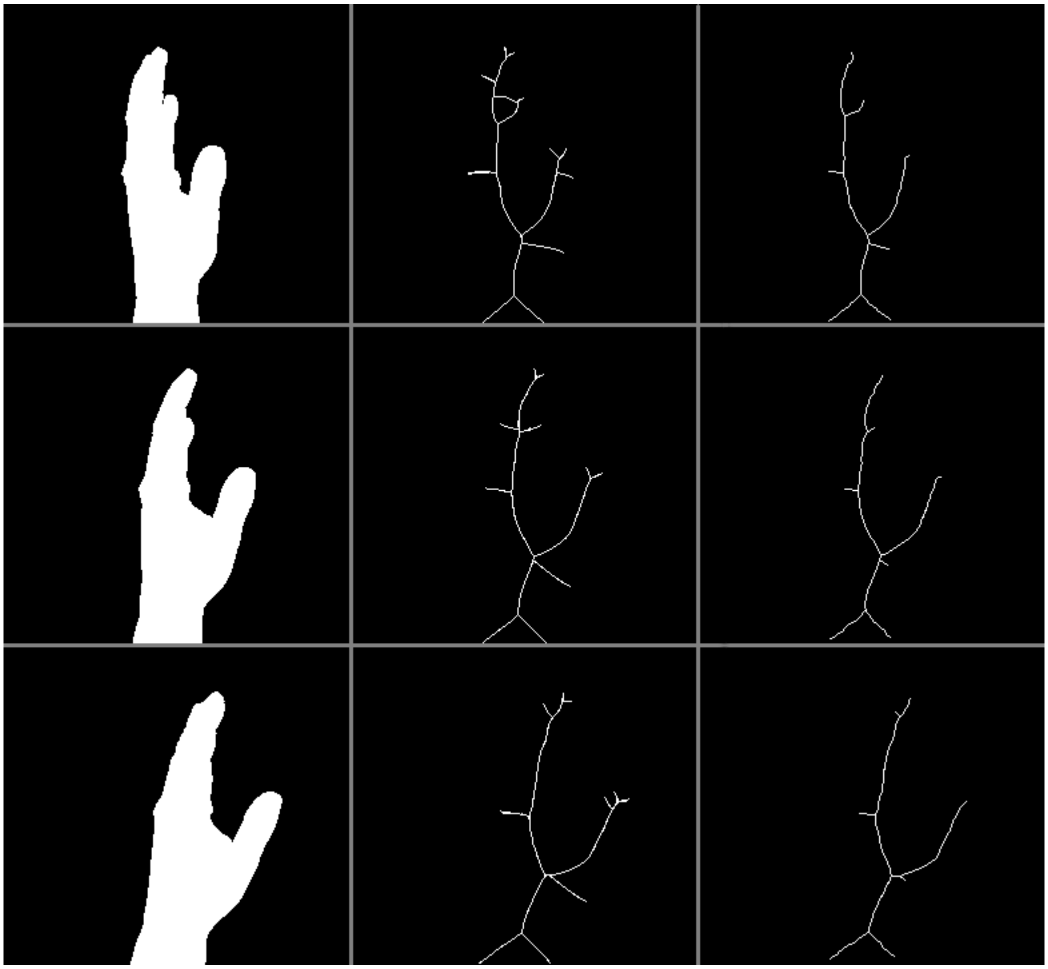


Figure 4.3: Left column: Hand gesture binary image sequence from [7]. Middle Column: Skeletons obtained using the implementation of algorithm [6] provided in [5]. Right column: Skeletons obtained using the approach in this paper.

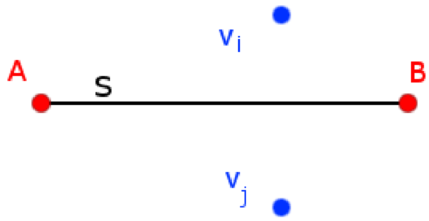


Figure 4.4: Voronoi segment representation.A corresponds to α_{ij}, B to ω_{ij} and S to s_{ij}.

case, we propose and prove several Theorems. The first theorem proving this statement when $V_{sites} = C$, in the continuous image domain case, and the second theorem generalizes this result when $V_{sites} \subset C$. First we state a useful lemma for the remaining proofs.

Lemma 4.3.1. *A Voronoi site does not belong to any Voronoi segment, in other words, the distance from a Voronoi site to (the points of) a Voronoi segment is never zero.*

$$\forall p \in s_{ij} \wedge V_{sites} = \{v_1, ..., v_n\} \wedge \forall v_k \in V_{sites} \,|\, \|p - v_k\| > 0 \tag{4.6}$$

Proof. By definition the distance between two different points is greater than zero. Also by definition, the Voronoi segment s_{ij} corresponds to a set of points which are equidistant to two different Voronoi sites v_i and v_j, and there is not any Voronoi site closer. Consequently the minimum distance between v_i and (any point in) s_{ij} is half of the distance between v_i and v_j, which is always greater than zero. □

Theorem 4.3.2. *Let $\alpha_{ij}, \omega_{ij} \in (T \cap s_{ij})$ be the end points of a Voronoi segment s_{ij} (i.e. its Voronoi vertices), determined by Voronoi sites v_i and v_j. If both end points α_{ij}, ω_{ij} belong to the shape F, then the whole segment s_{ij} is contained in F, when the Voronoi site set is equal to the whole shape boundary curve $V_{sites} = C$, and the image domain is continuous. Formally,*

$$\alpha_{ij} \in F \wedge \omega_{ij} \in F \Rightarrow s_{ij} \subset F \tag{4.7}$$

Proof. Let us start assuming that there is a point p in a Voronoi segment s_{ij} whose end points fall inside the shape F, but p does not belong to the shape,

$$\alpha_{ij} \in F \wedge \omega_{ij} \in F \tag{4.8}$$

$$\exists p \in s_{ij} \wedge p \notin F \tag{4.9}$$

The proof is by contradiction, showing that s_{ij} does not conform with the definition of Voronoi segment under this condition. By eq. 4.8 and 4.9we have that,

$$\begin{aligned} &\alpha_{ij} \in F \wedge \omega_{ij} \in F \wedge \exists p \in s_{ij} \wedge p \notin F \Rightarrow \\ &\qquad \exists q \in \overline{\alpha_{ij} p} \wedge \exists m \in \overline{\alpha_{ij} q} \wedge \exists n \in \overline{qp} \\ &\qquad |m \in F \wedge n \notin F \wedge \|q - m\| < \epsilon \wedge \|q - n\| < \epsilon \wedge (\epsilon \to 0) \end{aligned} \tag{4.10}$$

where $\overline{ab}$ denotes a segment between a and b. Therefore q belongs to C by definition 4.2. □

Finally, since the whole shape boundary curves are used as Voronoi sites $V_{sites} = C$ and q belongs to the shape boundary curves $q \in C$, then it is also a Voronoi site $q \in V_{sites}$. Moreover, because q belongs to the Voronoi segment , i.e. $q \in s_{ij}$, then q is a Voronoi site which is at zero distance from the Voronoi segment. contradicting lemma 4.3.1. Therefore, we can conclude that s_{ij} is not a Voronoi segment, contradicting the theorem statement.

$$q \in s_{ij} \wedge q \in V_{sites} \Rightarrow \exists v_k = q \wedge \exists x = q \,|\|x - v_k\| = 0 \tag{4.11}$$

Theorem 4.3.3. *Let $\alpha_{ij}, \omega_{ij} \in (T \cap s_{ij})$ be the end points of a Voronoi segment s_{ij} (i.e. its Voronoi vertices). Let the Voronoi sites be a uniform sampling of the contour $V_{sites} \subset C$ according to eq. 3.3, with δ the sampling density value. If the Voronoi segment s_{ij}, which is part of the Voronoi skeleton, is at distance greater than δ from any $v_i \in V_{site}$, then we can guarantee that if both α_{ij}, ω_{ij} belong to the shape F, then the whole s_{ij} is contained in F.*

Proof. The proof of this theorem is by contradiction, following a parallel reasoning as that in Theorem 4.3.2. We start assuming that there is a background point $p \notin F$ belonging to a Voronoi segment s_{ij} whose endpoints belong to the shape $(\alpha_{ij}, \omega_{ij} \in F)$. The conclusion of eq. 4.10 that q is a boundary point still holds, but now the boundary curve point q does not necessarily belong to V_{sites}. However, by the definition of the uniformly downsampled boundary curve in eq. 4.10, and by the continuity of the boundary curves, there must be at least a $v_k \in V_{sites}$ at distance equal or less than δ from q, therefore,

$$q \in C \Rightarrow \exists v_k \in V_{sites} \; s.t. \; \|q - v_k\| \leq \delta \tag{4.12}$$

By the Voronoi segment definition B.6 and the previous equation 4.12 we have three cases regarding the distance of this Voronoi segment point q to its generating pair of Voronoi sites, let us consider one of them, v_i, either $\delta < \|q - v_i\|$, or $\delta > \|q - v_i\|$, or $\delta = \|q - v_i\|$.

Case 1. $\delta < \|q - v_i\|$: In this case by equation 4.12 there is a Voronoi site v_k which is closer than any of the segment generating Voronoi sites v_i, v_j to a point q belonging to the Voronoi segment s_{ij}.

$$\exists v_k \in V_{sites} \wedge \exists x = q \; s.t. \; \|x - v_k\| < \|x - v_i\| \tag{4.13}$$

Consequently s_{ij} is not a Voronoi segment, contradicting the theorem statement. Thus the theorem is proved.

Case 2. $\delta > \|q - v_i\|$: In this case the fact that point q is a boundary point does not contradict the fact that s_{ij} is a Voronoi segment. That is, we can not assert that $\exists v_k \in V_{sites}$ such that $\|q - v_k\| < \|q - v_i\|$. Therefore, in shape folded regions, where Voronoi edges are closer to the boundary curves than the sampling density, we may have Voronoi edges which cross the boundary and are not detected and pruned.

Case 3. $\delta = \|q - v_i\|$: Let us also consider that $\|q - v_k\| = \delta$. In this case if v_k is one of the Voronoi sites of this Voronoi segment $(v_k = v_i) \vee (v_k = v_j)$, the fact that point q is a boundary point does not contradict the fact that s_{ij} is a Voronoi segment. If v_k is not one of the generating site, then there are three equidistant Voronoi sites to the point q in the Voronoi segment s_{ij}, and q is not an endpoint of the segment, contradicting that s_{ij} is a Voronoi segment and the theorem holds. If $\|q - v_k\| < \delta$ it is plain that the theorem holds.

□

We can state the theorem in terms of the distance transform, so that for those segments that fulfill the condition

$$\min_{x \in s_{ij}} DT(x) > \delta \tag{4.14}$$

we have that if both segment endpoints are inside the shape, then we can assert that the whole segment is inside the shape. If we perform a shape normalization through opening ensuring that the shape is δ-regular, then we will have that theorem 4.3.3 holds everywhere in the shape, when we subsample it uniformly with density δ.

This result is in good correspondence with the shape boundary curve sampling methods for efficient Voronoi Skeleton computation described in section 3.2.3.1.

4.4 Conclusions

The skeleton computation method proposed here obtains a stable, simple and robust skeleton with a reduced computational complexity, therefore it is suitable for realtime scenarios, like hand gesture recognition in video images.

First of all, the shape boundary curves are subsampled in order to reduce the complexity of the Voronoi Tessellation computation, removing part of the noise in the shape boundary. The subsampling methods used guarantee that the Voronoi Tessellation correctly approximates the skeleton of the shape.

Then the skeleton is pruned in two stages. The first stage involves removing Voronoi edges not completely included in the original shape. This pruning method shortens some skeleton branches, but the shortening is limited, and can be performed efficiently thanks to theorem 4.3.3 proved in section 4.3. If both end points of a Voronoi segment are included in the shape, then the whole segment is included inside the shape, under certain sampling conditions.

The second pruning stage is related to the Discrete Curve Evolution procedure and the pruning procedure presented in [6]. Our method outperforms this pruning procedure, because the computations required are performed at Voronoi segment level, instead of pixel level, and because they are only computed on Voronoi segments fulfilling the first pruning stage.

Chapter 5

Results

In the previous chapter we gave the formal definition of the algorithm, and proved some central results for its realtime performance. In this chapter we will focus on the empirical evaluation of the algorithm attending to skeleton stability and pattern recognition results. We emphasize the application of the algorithm for tabletop gesture recognition, which was our original goal, performing an exhaustive computational experimentation. In the first set of tests the realtime performance of the algorithm has been checked. Next, our skeletonization approach, *Beris* henceforth, has been tested against the algorithm in [6], called *Bai* henceforth, by two means: first comparing the stability of the skeletons (branching variation between similar shapes), and then checking their suitability for shape recognition using a quick greedy matching approach (introduced in section5.3).

In section 5.1 we give some words about the implementation and its real time efficiency. In section 5.2 we present the two benchmarking databases employed. In section 5.3 we present the graph matching algorithm used in our pattern recognition experiments. Section 5.4 describes the real time performance obtained. Section 5.5 gives the skeleton stability results over the two databases and section 5.6 gives the pattern recognition results. Finally, section 5.7 gives some summary conclusions of the chapter.

5.1 Some Words about Implementation

Our basic skeletonization and pruning procedure, *Beris*, is encapsulated into a $C++$ class ($B\_ G\_Skeletonization$). Some assumptions about the maximum number of shape boundary points in a shape boundary have been made in order to improve the efficiency, considering images of a 320x240 resolution, besides that, the code is not optimized at all. This class makes use of the Open Source Computer Vision (OpenCV) library. We have used the Voronoi Tessellation function implementation provided in this library, as well as some other functions for shape boundary point sequence extraction, image visualization and low level image processing. Our implementation requires a shape to be connected, but it can have holes in it. If a hole's area is less than a threshold, it is discarded for the skeleton computation. The output of the implemented procedure is the shape skeleton drawn in an image, like the ones in the right column of figure 4.3. This prototype implementation has been used to compute the skeletons in figures 4.3 and 4.1.

We have also included an implementation of the graph computation procedure described in section 5.3. Theoretically, obtaining the skeleton graph from the skeleton is a trivial task. But in practice, the discrete space introduces several problems: distinguishing between branch and joint points is not trivial and neither it is to follow a skeleton branch from end to end, due to 8-connectivity limitations.

For the tests involving the algorithm in [6], *Bai*, we have used the Matlab code implementation of their algorithm, publicly available at [5]. A problem of this implementation is that it cannot deal with holes in the shapes, ignoring them for the skeleton computation.

5.2 Data Sets

Two image databases have been used in the empirical evaluation. The first one corresponds to a hand gesture image database designed for visual hand gesture interaction on tabletops developed by us during this research work. The second one is a well known database used for shape matching algorithms benchmarking.

5.2.1 Hand Gesture Set for Tabletop Interaction

We have made this image database available at [7]. It contains a set of binary images corresponding to the even frames of the video recordings of three different dynamic hand gestures. These gestures are dynamic in the sense that include motion and pose changes over time, as opposed to a static pose,

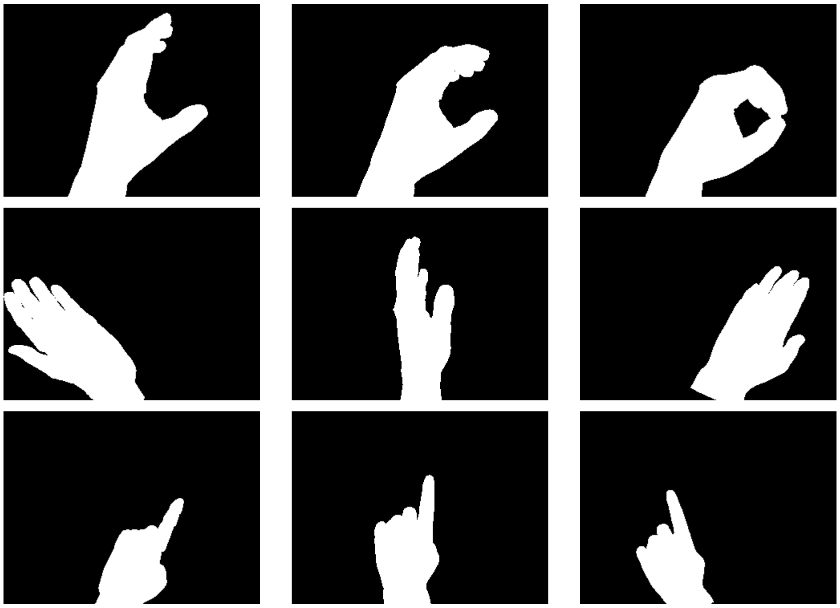


Figure 5.1: **IMT database examples**. Image examples in database [7]. Hand gestures for natural IMT interaction.

which can be represented by only one image. The hand gesture database has been built during this PhD work in order to be used as a benchmark for shape skeletonization and recognition in the context of natural hand gesture interaction in IMT (described in chapter 2). The three gestures in the database correspond to the basic hand gestures defined for the IMT design proposal in section 2.8 of chapter 2. Those gestures are: grabbing an object, turning a page and pointing to some object. Examples of the initial, intermediate and final frames of each gesture are shown in fig. 5.1. This database includes 200 repetitions of the three basic gestures, each repetition represented by a sequence of 15 images, up to a total of 200x3x15=9000 images. All of them are binary images, stored in Windows bitmap format (.bmp) with a resolution of 320x240 pixels. We will call this database *Tabletop database* henceforth. All the images are upright oriented, although there are orientation variation in the three dimensions.

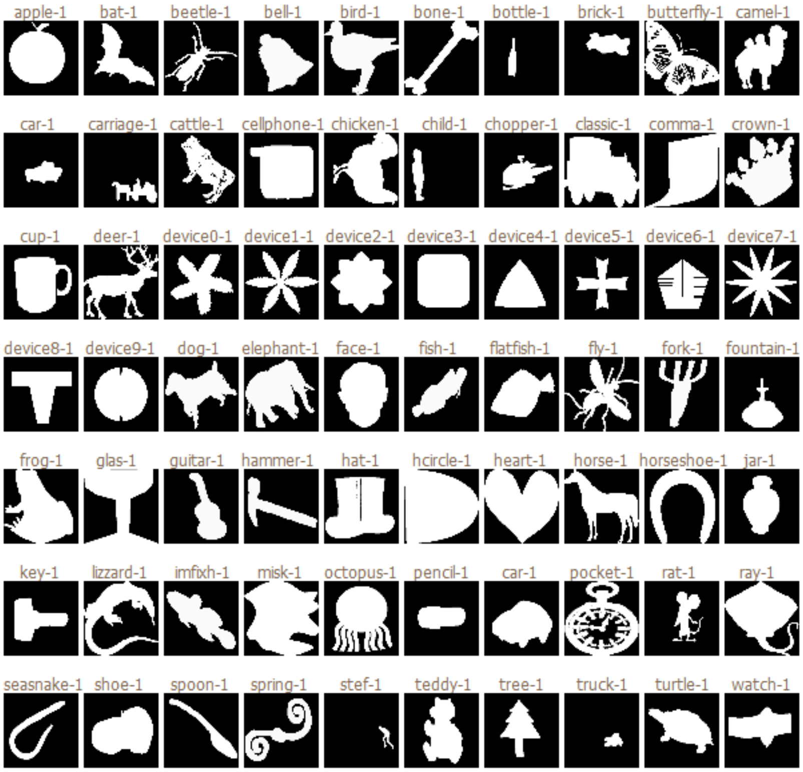


Figure 5.2: First image of each class in the MPEG-7 Core Experiment CE-Shape-1 Test Set

5.2.2 MPEG-7 Core Experiment CE-Shape-1 Test Set

This image database is commonly used for benchmarking shape matching algorithms and includes binary images grouped into categories by their content, not their appearance or shape. Because of this fuzzy categorization, there are categories, like car or bird, which include images corresponding to the same concept, but showing noticeable different shapes. Each category includes 20 samples. Shape variations inside a category may include rotation, size and position, and even image resolution variations. Some images have holes in it, while others do not. Figure 5.2 shows the first sample of each category in the database. We will denote this database as *MPEG-7 database* henceforth.

5.3 Skeletal Graph Matching using a Greedy Algorithm

This section presents a greedy based attributed graph matching algorithm used to compare the shape matching performance of both skeleton computation and pruning approaches *Beris* versus *Bai*. The only reasons to choose an algorithm of this characteristics is its simplicity to be implemented, and its computational efficiency, not its recognition performance.

We consider a conventional mapping of the skeleton into an undirected graph (see Appendix B), which can be cyclic or acyclic, depending on the shape having holes or not. Skeleton points are categorized into three categories: *end points* of branches, points with only one skeleton point in their 8-neighborhood, *branch points*, points in a branch with two skeleton points in their 8-neighborhood, and *joint points*, connecting three or more branches, i.e. with more than two points in their 8-neighborhood. A graph G_{skel}will be constructed as follows: the skeleton end points and joint points are the graph nodes, and the skeleton branches correspond to the links of the graph, labeled with the length of the branch.

5.3.1 Definitions

This subsection introduces the notation and definitions for the specification of the algorithm, including the attributes stored by the skeleton graph, as well as some data structures for a simpler management of skeleton graphs.

5.3.1.1 Skeleton Point Information (SPI)

It is composed of the skeleton point position and DT value.

- Position in polar coordinates:
  - r: radius, normalized to the maximum distance to the skeleton centroid.
  - θ: angle normalized between 0 and 1.
- Distance Transform function value:
  - dt: normalized to the maximum value in the skeleton.

5.3.1.2 Node

Corresponds to the end points of each of the skeleton branches. They include all the information of a SPI and also the list of all their adjacent nodes.

5.3.1.3 Link

Corresponds to a skeleton branch. It includes the sequence of skeleton points in the branch and their distance transform function value (nodes), and its two end points (node with adjacency).

- SPI sequence (ordered from one end point to the other by neighborhood relationship between consecutive points).
- Two nodes.

5.3.2 Some Variable Value Normalizations

First we normalize the DT value. The maximum value of the DT function in the skeleton is computed and each value is normalized to this maximum value:

$$\forall s \in Skel\ dt'(s) = dt(s) \,/ \max_{k \in Skel}(dt(k))$$

so that $dt'(s) \in [0,1]$.

Second, the position, which is initially stored in Cartesian coordinates, is normalized in two stages. First, translation invariance is achieved using the skeleton centroid c_{skel}, which is the average position of all the skeleton points.

$$\forall s \in Skel\ s' = s - c_{skel}$$

Second, the skeleton point coordinates are transformed into the polar coordinate system $p = (r, \theta)$.

$$r = \sqrt{x^2 + y^2} \text{ and } \theta = \begin{cases} \arctan\left(\frac{y}{x}\right) & x > 0 \wedge y \geq 0 \\ \arctan\left(\frac{y}{x}\right) + 2\pi & x > 0 \wedge y < 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & x < 0 \\ \frac{\pi}{2} & x = 0 \wedge y > 0 \\ \frac{3\pi}{2} & x = 0 \wedge y < 0 \\ 0 & x = 0 \wedge y = 0 \end{cases}$$

Third, the scale invariance is achieved normalizing the distance between the skeleton centroid c_{skel} and each point to the maximum distance value.

$$\forall p \in Skel\ r' = r / \max_{k \in Skel} (\|k - c_{skel}\|)$$

Fourth, orientation is then normalized to range $[0, 1]$

$$\forall p' \in Skel\ \wedge \theta' = \frac{\theta}{2\pi}$$

5.3.3 Node Distance Measure

It is a weighted distance between position difference and distance transform value difference. If we have two nodes $A = (A_r, A_\theta, A_{dt})$ and $B = (B_r, B_\theta, B_{dt})$, the distance between them is measured as,

$$|A, B|_{node} = \alpha \, |A_{dt} - B_{dt}| + (1 - \alpha) \sqrt{A_r^2 + B_r^2 - 2 A_r B_r \cos (A_\theta - B_\theta)}$$

so that $|A, B|_{node} \in [0, 1]$.

5.3.4 Graph Matching Algorithm

Consider that we have a set of template graphs $T = \{T_1, ..., T_n\}$, and a sample graph S. The graph matching problem consists of finding the most similar graph to S in T. For this purpose, we apply an time efficient greedy approach. Subgraph isomorphism and homomorphism search to solve the graph matching problem, are well known NP-hard problems, which are not feasible for realtime recognition systems. The pseudo code formulation of our greedy algorithm to match a template graph T_i with a sample graph S is shown in figure 5.1:

As it can be appreciated in the piece of code of figure 5.1, nodes in both graphs are sorted by their normalized geometrical position, starting from the nodes closest to the image domain boundaries, visiting the graph nodes towards the center of the image domain. The `Match()` structure stores the set of node pairs matched between the sample graph and the template graph. This sorting of the nodes optimizes the graph visiting process for the hand gesture recognition on IMT (see appendix 2). In this context the user arm enters the image from its domain boundaries towards inside, and the arm's entrance point in the image is used as the origin for the node sorting.

Algorithm 5.1 Greedy matching algorithm pseudo code

```
int numMatches=0;
//sort nodes in Nodes(Ti) and Nodes(S)
//so that (n(i).y>n(i+1))OR
//((n(i).y=(n(i+1).y)&&(n(i+1).x<=n(i).x))
foreach n in Nodes(Ti)
{
        foreach m in Nodes(S)
        {
                if(notAlreadySelected(m)
                        AND
                |Neighbors(n)|<=|Neighbors(m)|)
                {
                        if(d(n,m)<d(n,Min))
                                Min:=m;
                }
        }
        Match[numMatches]:=(n,Min);
        numMatches:=numMatches+1;
}
```

5.3.5 Graph Matching Result Evaluation

The main idea is to count the number of nodes paired from the total number of nodes of both graphs. Since the node distance ranges $[0, 1]$ with 0 corresponding to the same node, its opposite value (i.e. 1-distance or similarity), the similarity, is similar to a matching probability between nodes. Therefore, instead of only counting the number of matched nodes, the sum of this similarity is computed.

The matching result is evaluated taking into account the number of nodes in the template graph matched $|Match\,(T_i, S)|$ and the distance between the matched nodes. Mathematically, the similarity value $similarity(T_i, S)$, obtained from the matching $Match\,(T_i, S)$ is defined as,

$$similarity\,(T_i, S) = \frac{2\sum\left(1 - \left|Match\,(i)_{T_i}, Match\,(i)_S\right|_{node}\right)_{1\leq i\leq|Match|}}{|Nodes\,(T_i)| + |Nodes\,(S)|}$$

where the distance between nodes is normalized to $[0, 1]$, and in order to simplify notation, $Match(i)$ corresponds to the i-th match between the sample graph S and the template graph T_i, and $Match\,(i)_{T_i}$ corresponds to the id of the node in T_i in that matching, and $Match\,(i)_S$ corresponds to the id of the node in S in that matching.

5.4 Algorithm Computational Performance Tests

Two software prototypes have been built to test the robustness and realtime performance of our procedure. They have been developed in Windows XP. Both use the same code implementation of our skeletonization and pruning algorithm, Beris. The difference is that the first one is an offline version of the algorithm that works on preprocessed static binary images, and the second one is a realtime, which is able to capture images from a digital video camera, perform a chroma key based background substraction, an image thresholding, a morphological operation for noise removal in the shape boundary curves, and then apply our algorithm.

For the online implementation, a green chroma key was used in our testing indoor environment with no special lighting and a Sony EVI D-70 camera. The video resolution used was 320x240 and the prototype followed this algorithm for each image:

1. Substract the objects on the chroma key using the HSV color space.
2. Threshold the image, obtaining a binary image with the objects on it.
3. Find all the connected components,i.e., shapes, in the image.
4. For each shape in the image:
    (a) If its area is bigger than a threshold value:
        i. Obtain the shape boundary point sequence. It is a list of sequences, including the external boundary and the internal boundaries (i.e. holes).
        ii. Check the size of the holes in the shape to include them or not in the final skeletonization.
        iii. Compute the skeletonization procedure described in chapter 4 using the methods in *B_ G_ Skeletonization.*
        iv. Show the resulting skeleton on a window.

The realtime prototype is able to compute the previous algorithm at 60 frames per second on a Intel Pentium IV processor (4 GHz.) with 1 GB. memory and Windows XP SP2. Therefore it proves that it is possible to make a realtime implementation of our algorithm. Example images of the prototype is use are shown in figure 5.3. Our research group web page also hosts a video example showing the realtime performance of our prototype available at [193].

5.5 Algorithm Stability Results

We consider a conventional skeleton mapping approach into a undirected graph, which can be cyclic or acyclic, if the shape has holes on it or not. Skeleton points are categorized into three categories: *end points* of branches, points with only one skeleton point in their 8-neighborhood, *branch points*, points in a branch with two skeleton points in their 8-neighborhood, and *joint points*, connecting three or more branches, i.e. with more than two points in their vicinity. A graph G_{skel}can be constructed where the end points and joint points are the nodes and the branches joining them form the edges of the graph, labeled with the length of the branch.

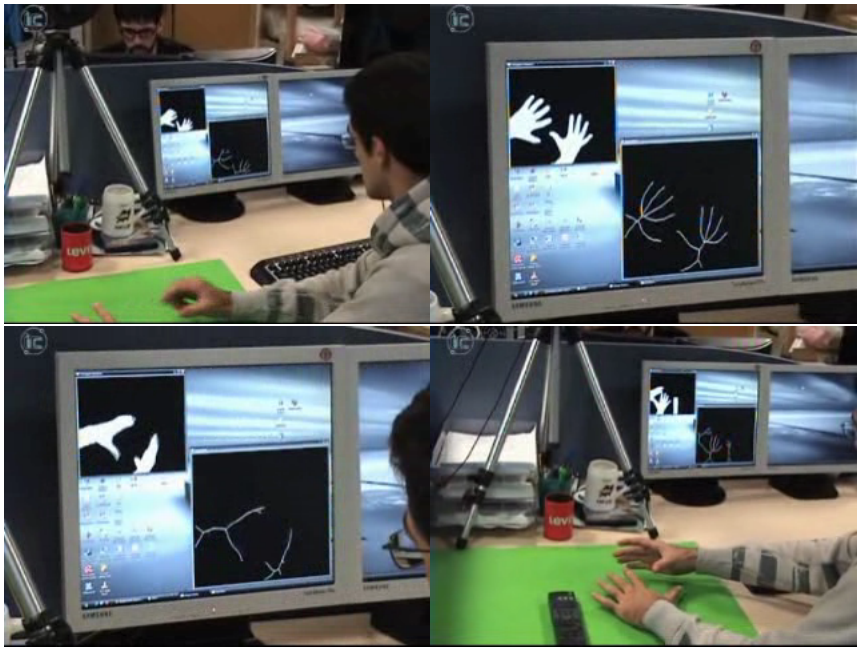


Figure 5.3: **Realtime prototype**. Pictures of the realtime skeletonization and pruning software prototype.

	Bai		Beris	
	Avg.	Std. Deviation	Avg.	Std. Deviation
Grab	15.94	2.40	12.84	2.08
Point	13.88	2.47	10.87	1.54
Turn page	16.28	3.40	11.93	2.80
Global	**15.37**	**2.76**	**11.88**	**2.14**

Table 5.1: **Branch number 15 DCE pruning**. Skeleton branch number difference between this approach and the approach by BAI in [6], denoted as BAI. The average and the standard deviation are shown by gestures and globally.

	Bai		Beris	
	Avg.	Std. Deviation	Avg.	Std. Deviation
Grab	20.50	2.73	15.56	2.75
Point	18.89	2.49	12.04	2.07
Turn page	20.65	3.92	13.63	3.15
Global	**20.02**	**3.05**	**13.74**	**2.66**

Table 5.2: **Branch number 20 DCE pruning**. Skeleton branch number difference between this approach and the approach by BAI in [6], denoted as BAI. The average and the standard deviation are shown by gestures and globally.

Using the graph G_{skel} an intuitive measure of the skeleton complexity is the number of graph links, corresponding to the number of branches in the skeleton.

5.5.1 Tabletop Database

First we computed the average branch number and the standard deviation for every image in the database partitioned by gestures. Tables 5.1, 5.2 and 5.3 show the results for DCE pruning values of 15, 20 and 25, respectively. According to the these tables, *Beris* systematically produces simpler skeletons for the same parameters (i.e., DCE pruning level) than *Bai* [6], and with less variability.

We have a dynamic gesture database, so we can compute another measure

	Bai		Beris	
	Avg.	Std. Deviation	Avg.	Std. Deviation
Grab	24.55	3.04	17.67	3.14
Point	22.73	3.03	12.61	2.52
Turn page	24.33	4.55	14.92	3.45
Global	**23.87**	**3.54**	**15.07**	**3.03**

Table 5.3: **Branch number 25 DCE pruning**. Skeleton branch number difference between this approach and the approach by BAI in [6], denoted as BAI. The average and the standard deviation are shown by gestures and globally.

	Bai		Beris	
	Avg.	Std. Deviation	Avg.	Std. Deviation
Grab	2.17	1.80	1.62	1.46
Point	2.67	1.98	1.40	1.17
Turn page	3.05	2.42	2.11	1.96
Global	**2.63**	**2.06**	**1.75**	**1.53**

Table 5.4: **Branch number difference 15 DCE pruning.** Skeleton branch number variation between consecutive images, comparing this approach and the approach by BAI in [6], denoted as BAI. The average and the standard deviation of the difference are shown by gestures and globally.

of skeleton stability. That is, we computed the branch number variation between consecutive images corresponding the the same gesture sample in the database. Because the shape variation between frames will be small, we expect that the corresponding variation in the skeletons will also be small. Therefore, the difference in the number of skeleton branches between frames is a good measure of the skeleton stability. Results are shown in tables 5.4, 5.5 and 5.6 for DCE pruning values of 15, 20 and 25 grouped by gestures and then a global measure is included. According to them our algorithm is also more stable, because for images corresponding to consecutive frames, which are also similar between them, the difference is lower, and also is the variation of this difference. Consequently, we conclude that our algorithm produces a simpler and also more stable skeleton than that produced using the algorithm in [6].

	Bai		Beris	
	Avg.	Std. Deviation	Avg.	Std. Deviation
Grab	2.52	1.99	1.90	1.63
Point	2.61	2.06	1.60	1.29
Turn page	3.39	2.72	2.38	2.06
Global	**2.84**	**2.26**	**1.96**	**1.66**

Table 5.5: **Branch number difference 20 DCE pruning.** Skeleton branch number variation between consecutive images, comparing this approach and the approach by BAI in [6], denoted as BAI. The average and the standard deviation of the difference are shown by gestures and globally.

	Bai		Beris	
	Avg.	Std. Deviation	Avg.	Std. Deviation
Grab	2.79	2.16	2.10	1.72
Point	3.24	2.50	1.75	1.47
Turn page	3.89	3.06	2.58	2.20
Global	**3.30**	**2.57**	**2.15**	**1.80**

Table 5.6: **Branch number difference 25 DCE pruning.** Skeleton branch number variation between consecutive images, comparing this approach and the approach by BAI in [6], denoted as BAI. The average and the standard deviation of the difference are shown by gestures and globally.

5.5.2 MPEG-7 Database

The images in this database contain samples with holes. The implementation of the *Bai* algorithm available from the authors does not take into account the existence of holes in the shapes, whereas *Beris* does. That imposes some combinatorics in the experimental design. We have considered the following cases:

- Case A corresponds to the whole dataset,
- Case B corresponds to the whole dataset when *Beris* ignores the holes in the shapes,
- Case C corresponds to the image set resulting after removing the shapes with holes.

We computed the average branch number and the standard deviation for every image in the database grouped by categories for the three cases described above. Tables 5.7, 5.8 and 5.9 show the total average and standard deviation for DCE pruning values 15, 20 and 25, respectively. These tables show that the highest difference between Beris and Bai occurs for a pruning value of 25, and Case B (see last row in table 5.9), which corresponds to the case when the whole database is used, but the holes in the shapes are not considered by neither of the algorithms. Next, graph plots in figures 5.4 and 5.5 show the average branch number and standard deviation of the shape skeletons grouped by classes.

As the tables show, our method produces simpler skeletons for the same pruning value. The higher standard deviation values of our algorithm in Case A correspond to the fact that our algorithm is taking into account the branches produced by holes on the shape, while the implementation of the Bai approach does not. In the rest of cases the Beris approach has lower standard deviation values, so it is more stable.

5.6 Shape Recognition Tests

These tests compares the shape recognition performance using a simple but efficient skeleton graph matching algorithm, between skeleton graphs obtained using the approach by *Bai* [6] and our approach, *Beris.* The graph matching algorithm used in this tests has been presented in section 5.3. Three

	Average		Std. deviation	
	Bai	Beris	Bai	Beris
Case A	17.23	16.48	4.25	5.06
Case B	17.23	12.61	4.25	2.46
Case C	15.95	13.43	3.51	3.33

Table 5.7: **Branch number average and standard deviation for the MPEG database for DCE pruning value 15**. Case A corresponds to the whole dataset, Case B corresponds to the whole dataset without considering the holes in the shapes, and Case C corresponds to the set resulting after removing the shapes with holes.

	Average		Std. deviation	
	Bai	Beris	Bai	Beris
Case A	21.95	19.60	4.97	5.74
Case B	21.95	15.80	4.97	3.36
Case C	20.56	16.60	4.12	4.14

Table 5.8: **Branch number average and standard deviation for the MPEG database for DCE pruning value 20**. Case A corresponds to the whole dataset, Case B corresponds to the whole dataset without considering the holes in the shapes, and Case C corresponds to the set resulting after removing the shapes with holes.

	Average		Std. deviation	
	Bai	Beris	Bai	Beris
Case A	26.44	22.77	5.71	6.29
Case B	26.44	19.03	5.71	3.99
Case C	25.20	19.84	4.81	4.74

Table 5.9: **Branch number average and standard deviation for the MPEG database for DCE pruning value 25**. Case A corresponds to the whole dataset, Case B corresponds to the whole dataset without considering the holes in the shapes, and Case C corresponds to the set resulting after removing the shapes with holes.

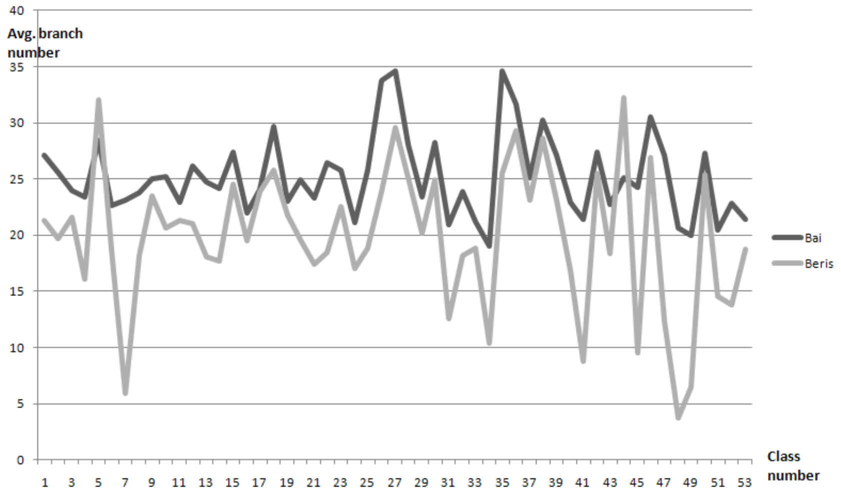


Figure 5.4: Branch number average grouped by class for Case B and DCE pruning value 25.

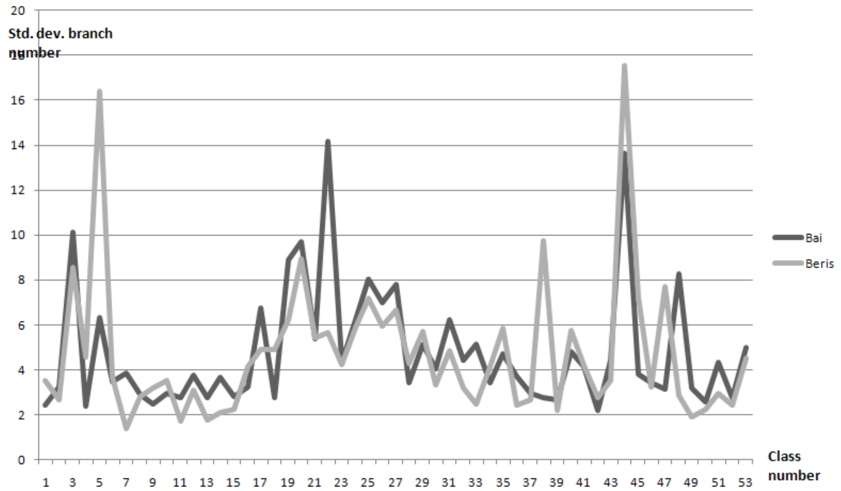


Figure 5.5: Branch number standard deviation grouped by class for Case B and DCE pruning value 25.

tests have been carried out modifying the node similarity measure parameter α, for each of the three pruning values (i.e., 15, 20 and 25).

5.6.1 Tabletop Database

In this database, each of the three gestures is composed of 15 frames, therefore having 45 initial classes. Since each gesture image set corresponds to a dynamic hand gesture over the time, consecutive images are very similar. Therefore, a class grouping has been performed, so that consecutive frames in each gesture are grouped into 5 class groups. For example, frames 1 to 5 of the grab gesture are grouped into the class group 1, frames 6 to 10 are grouped into the class group 2 and so on for the rest of the frames and gestures.

5.6.1.1 Validation based on a k-NN classifier

The classifiers used are 1-NN, 3-NN and 5-NN, selecting randomly one, three and five gesture samples for each gesture as the class representatives, and classifying the remaining images of the database into the three classes described previously. We performed a hundred repetitions of the experiment to estimate the generalization of the classifiers shown in the tables. Notice that these experimental designs do not correspond to a n-fold crossvalidation, because there is no systematic partition of the database into n subsets. Total recognition (averaging the recognition success over all classes) results are shown in tables 5.10, 5.11 and 5.12, for 45, 9 and 3 classes (one for each gesture). We first tried to classify all the 15 frames of each gesture as separate classes, assuming that they may be considered as identifiable objects. Given the results of table 5.10 we considered the need to aggregate the classes showing greater confusion and being part of the same gesture. Then we obtained 9 classes, whose classification results, following the above 1-NN, 3-NN and 5-NN procedures, are shown in table 5.11. Some of the classes showed a low recognition rate, and we decide to further aggregate classes, reaching the minimum number of classes: the number gestures we want to recognize.

The confusion matrices for the best results of each table (highlighted in bold characters in the tables) are shown in figures 5.6 and 5.7 for 45 and 9 classes as grayscale images for better reading and table 5.13 for 3 classes. These images correspond to a visual representation of the normalized values of the cells in the confusion matrices, so that brighter colors correspond to

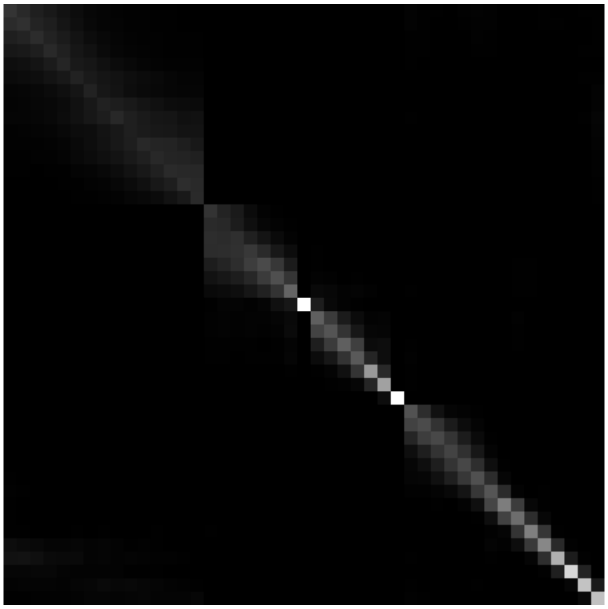


Figure 5.6: **Total recognition Normalized Confusion Matrix Image for the best case of our algorithm (Beris) with 45 classes: pruning 15, alpha 0.75 and 5NN.** Brighter colors represent higher values.

higher values. Separate recognition results for each of the three gestures, for 45, 9 and 3 classes are show in tables 5.14, 5.15, 5.16 for the "grab" gesture; 5.17, 5.18 and 5.22 for the "point" gesture; and 5.20, 5.21 and 5.22 for the "turn page" gesture.

Several conclusions can be extracted from this tables:

- Our algorithm (*Beris*) outperforms the Bai algorithm recognition in most of cases.
- Higher values of the α parameter produce better results. This results mean that the geometrical position of nodes is more important for matching than the DT value. Notice that, in the particular case of this database, all the objects in the database are upright oriented. Therefore there are little rotation like noise effects in the images,
- As the number of training examples grow, the results improve, but the computational cost is also higher.

Pruning	Alpha	Classifier	Bai	Beris
15	0.25	1NN	15.59	15.93
15	0.25	3NN	21.74	21.88
15	0.25	5NN	24.81	25.25
15	0.50	1NN	17.85	20.24
15	0.50	3NN	24.83	27.42
15	0.50	5NN	28.00	31.12
15	0.75	1NN	19.80	23.50
15	0.75	3NN	26.95	31.55
15	**0.75**	**5NN**	**30.17**	**35.42**
20	0.25	1NN	16.26	16.10
20	0.25	3NN	22.14	21.46
20	0.25	5NN	25.18	24.50
20	0.50	1NN	19.02	20.15
20	0.50	3NN	25.63	26.62
20	0.50	5NN	28.88	30.03
20	0.75	1NN	20.94	23.02
20	0.75	3NN	27.63	30.44
20	0.75	5NN	30.75	33.96
25	0.25	1NN	16.52	16.06
25	0.25	3NN	22.40	21.13
25	0.25	5NN	25.25	24.21
25	0.50	1NN	19.30	19.94
25	0.50	3NN	25.76	26.30
25	0.50	5NN	28.88	29.68
25	0.75	1NN	21.22	22.83
25	0.75	3NN	27.72	30.01
25	0.75	5NN	30.82	33.58

Table 5.10: Total recognition results for 45 classes.

Pruning	Alpha	Classifier	Bai	Beris
15	0.25	1NN	41.51	42.06
15	0.25	3NN	50.94	51.68
15	0.25	5NN	54.90	56.13
15	0.50	1NN	48.01	53.00
15	0.50	3NN	58.61	63.38
15	0.50	5NN	62.90	67.64
15	0.75	1NN	52.93	59.26
15	0.75	3NN	63.77	69.45
15	**0.75**	**5NN**	**67.85**	**73.38**
20	0.25	1NN	42.01	42.92
20	0.25	3NN	51.71	50.91
20	0.25	5NN	55.79	54.64
20	0.50	1NN	49.78	53.02
20	0.50	3NN	60.35	61.86
20	0.50	5NN	64.53	65.69
20	0.75	1NN	54.89	58.83
20	0.75	3NN	65.36	67.83
20	0.75	5NN	69.13	71.40
25	0.25	1NN	42.78	43.35
25	0.25	3NN	52.54	50.67
25	0.25	5NN	56.05	54.64
25	0.50	1NN	50.48	52.93
25	0.50	3NN	61.03	61.41
25	0.50	5NN	64.65	65.28
25	0.75	1NN	55.46	58.60
25	0.75	3NN	65.72	67.18
25	0.75	5NN	69.07	70.87

Table 5.11: Total recognition results for 9 classes.

Pruning	Alpha	Classifier	Bai	Beris
15	0.25	1NN	72.06	70.74
15	0.25	3NN	80.53	80.21
15	0.25	5NN	83.67	83.81
15	0.50	1NN	78.30	80.73
15	0.50	3NN	87.01	89.17
15	0.50	5NN	89.84	91.96
15	0.75	1NN	82.30	84.78
15	0.75	3NN	90.71	92.27
15	**0.75**	**5NN**	**93.25**	**94.57**
20	0.25	1NN	72.04	70.58
20	0.25	3NN	80.93	79.61
20	0.25	5NN	83.82	82.71
20	0.50	1NN	79.77	80.85
20	0.50	3NN	88.23	88.60
20	0.50	5NN	90.79	91.08
20	0.75	1NN	83.34	84.97
20	0.75	3NN	91.29	92.03
20	0.75	5NN	93.60	94.01
25	0.25	1NN	72.81	71.40
25	0.25	3NN	81.73	79.69
25	0.25	5NN	84.14	82.83
25	0.50	1NN	81.11	81.18
25	0.50	3NN	89.41	88.79
25	0.50	5NN	91.41	91.19
25	0.75	1NN	84.95	85.30
25	0.75	3NN	92.44	92.17
25	0.75	5NN	94.39	94.12

Table 5.12: Total recognition results for 3 classes.

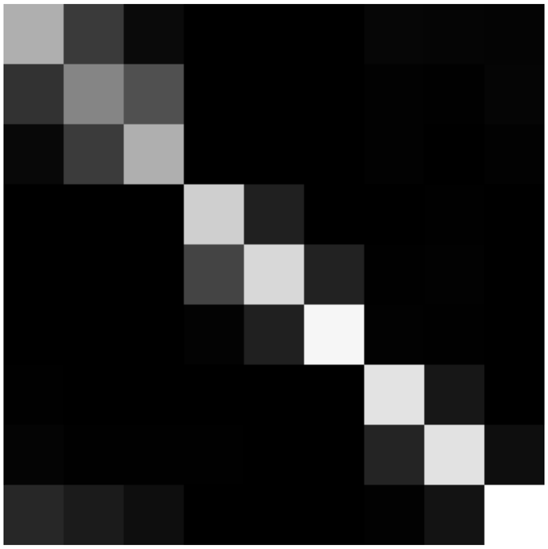


Figure 5.7: **Total recognition Normalized Confusion Matrix Image for the best case of our algorithm (Beris) with 9 classes: pruning 15, alpha 0.75 and 5NN.** Brighter colors represent higher values.

	Grab	Point	Turn Page
Grab	**266725**	603	32672
Point	82	**298783**	1135
Turn Page	10607	3731	**285662**

Table 5.13: **Total recognition Confusion Matrix for the best case of our algorithm (Beris) with 3 classes: pruning 15, alpha 0.75 and 5NN.**

Pruning	Alpha	Classifier	Bai	Beris
15	0.25	1NN	6.52	6.13
15	0.25	3NN	9.20	9.41
15	0.25	5NN	11.03	11.10
15	0.50	1NN	7.61	7.73
15	0.50	3NN	11.07	11.91
15	0.50	5NN	13.10	14.07
15	0.75	1NN	8.77	8.99
15	0.75	3NN	12.83	13.51
15	**0.75**	**5NN**	**15.11**	**15.97**
20	0.25	1NN	6.49	5.78
20	0.25	3NN	9.39	8.63
20	0.25	5NN	11.05	10.39
20	0.50	1NN	7.59	7.33
20	0.50	3NN	11.15	10.95
20	0.50	5NN	13.21	13.07
20	0.75	1NN	8.43	8.42
20	0.75	3NN	12.47	12.49
20	0.75	5NN	14.78	14.85
25	0.25	1NN	6.61	5.89
25	0.25	3NN	9.59	8.54
25	0.25	5NN	11.14	10.17
25	0.50	1NN	8.15	7.29
25	0.50	3NN	11.50	10.66
25	0.50	5NN	13.27	12.69
25	0.75	1NN	9.11	8.26
25	0.75	3NN	12.82	12.08
25	0.75	5NN	14.85	14.49

Table 5.14: Grab gesture recognition results for 45 classes.

Pruning	Alpha	Classifier	Bai	Beris
15	0.25	1NN	27.84	26.61
15	0.25	3NN	36.09	36.90
15	0.25	5NN	39.56	40.81
15	0.50	1NN	32.53	33.42
15	0.50	3NN	42.47	46.46
15	0.50	5NN	47.14	51.48
15	0.75	1NN	37.07	37.48
15	0.75	3NN	48.90	51.78
15	**0.75**	**5NN**	**54.00**	**57.22**
20	0.25	1NN	28.22	24.74
20	0.25	3NN	36.45	33.54
20	0.25	5NN	39.95	36.91
20	0.50	1NN	32.74	30.86
20	0.50	3NN	43.10	42.13
20	0.50	5NN	47.37	46.49
20	0.75	1NN	36.09	35.12
20	0.75	3NN	47.98	47.62
20	0.75	5NN	52.74	52.25
25	0.25	1NN	28.60	25.60
25	0.25	3NN	37.45	33.47
25	0.25	5NN	40.04	36.57
25	0.50	1NN	34.51	31.24
25	0.50	3NN	44.59	41.68
25	0.50	5NN	47.79	45.80
25	0.75	1NN	38.26	35.25
25	0.75	3NN	49.55	46.81
25	0.75	5NN	53.09	51.62

Table 5.15: Grab gesture recognition results for 9 classes.

Pruning	Alpha	Classifier	Bai	Beris
15	0.25	1NN	61.66	61.75
15	0.25	3NN	72.71	75.56
15	0.25	5NN	77.03	78.98
15	0.50	1NN	65.79	68.50
15	0.50	3NN	77.46	82.63
15	0.50	5NN	81.89	86.52
15	0.75	1NN	69.88	70.14
15	0.75	3NN	82.74	84.76
15	**0.75**	**5NN**	**87.33**	**88.90**
20	0.25	1NN	60.21	59.56
20	0.25	3NN	70.87	79.61
20	0.25	5NN	74.63	76.74
20	0.50	1NN	65.32	66.64
20	0.50	3NN	77.37	80.13
20	0.50	5NN	81.51	83.60
20	0.75	1NN	68.08	69.14
20	0.75	3NN	81.79	83.22
20	0.75	5NN	86.33	86.72
25	0.25	1NN	59.97	61.89
25	0.25	3NN	72.25	74.38
25	0.25	5NN	74.19	77.09
25	0.50	1NN	67.16	68.06
25	0.50	3NN	79.38	80.33
25	0.50	5NN	81.73	83.67
25	0.75	1NN	71.48	70.61
25	0.75	3NN	84.18	83.46
25	0.75	5NN	87.35	87.01

Table 5.16: Grab gesture recognition results for 3 classes.

Pruning	Alpha	Classifier	Bai	Beris
15	0.25	1NN	22.10	23.97
15	0.25	3NN	28.43	29.98
15	0.25	5NN	31.96	33.33
15	0.50	1NN	23.82	27.36
15	0.50	3NN	30.42	34.35
15	0.50	5NN	33.96	37.94
15	0.75	1NN	25.54	30.63
15	0.75	3NN	31.85	39.04
15	**0.75**	**5NN**	**35.18**	**42.87**
20	0.25	1NN	21.89	24.94
20	0.25	3NN	27.54	30.69
20	0.25	5NN	30.91	33.96
20	0.50	1NN	24.41	27.94
20	0.50	3NN	30.47	34.42
20	0.50	5NN	33.92	37.67
20	0.75	1NN	26.79	30.24
20	0.75	3NN	32.73	38.22
20	0.75	5NN	35.63	41.82
25	0.25	1NN	22.03	25.64
25	0.25	3NN	27.99	31.15
25	0.25	5NN	30.88	34.46
25	0.50	1NN	24.21	28.46
25	0.50	3NN	30.54	34.78
25	0.50	5NN	33.78	38.04
25	0.75	1NN	26.59	30.68
25	0.75	3NN	32.63	38.30
25	0.75	5NN	35.71	41.93

Table 5.17: Point gesture recognition results for 45 classes.

Pruning	Alpha	Classifier	Bai	Beris
15	0.25	1NN	55.55	57.66
15	0.25	3NN	62.04	63.79
15	0.25	5NN	65.34	67.13
15	0.50	1NN	60.09	66.97
15	0.50	3NN	66.67	72.26
15	0.50	5NN	69.70	74.76
15	0.75	1NN	63.57	71.01
15	0.75	3NN	69.42	76.60
15	**0.75**	**5NN**	**72.10**	**79.22**
20	0.25	1NN	53.19	60.57
20	0.25	3NN	61.39	65.40
20	0.25	5NN	65.26	68.69
20	0.50	1NN	60.76	68.38
20	0.50	3NN	68.46	72.28
20	0.50	5NN	72.05	74.90
20	0.75	1NN	67.37	71.15
20	0.75	3NN	73.87	75.73
20	0.75	5NN	76.19	78.48
25	0.25	1NN	53.05	61.97
25	0.25	3NN	61.35	66.30
25	0.25	5NN	64.55	69.66
25	0.50	1NN	59.10	69.15
25	0.50	3NN	67.50	72.78
25	0.50	5NN	70.55	75.16
25	0.75	1NN	65.33	71.55
25	0.75	3NN	72.08	75.48
25	0.75	5NN	74.43	78.13

Table 5.18: Point gesture recognition results for 9 classes.

Pruning	Alpha	Classifier	Bai	Beris
15	0.25	1NN	90.31	89.71
15	0.25	3NN	92.78	94.25
15	0.25	5NN	94.13	96.16
15	0.50	1NN	94.47	97.49
15	0.50	3NN	96.56	98.79
15	0.50	5NN	97.44	99.16
15	0.75	1NN	97.34	99.07
15	0.75	3NN	98.45	99.50
15	0.75	5NN	98.88	99.59
20	0.25	1NN	87.12	88.76
20	0.25	3NN	92.06	94.03
20	0.25	5NN	93.50	96.09
20	0.50	1NN	92.98	97.03
20	0.50	3NN	96.36	98.80
20	0.50	5NN	97.28	99.31
20	0.75	1NN	96.80	99.01
20	0.75	3NN	98.56	99.65
20	0.75	5NN	98.93	99.74
25	0.25	1NN	88.09	89.25
25	0.25	3NN	92.49	93.43
25	0.25	5NN	93.84	95.65
25	0.50	1NN	94.13	97.11
25	0.50	3NN	97.13	98.78
25	0.50	5NN	98.80	99.26
25	0.75	1NN	97.66	99.10
25	0.75	3NN	98.98	99.67
25	**0.75**	**5NN**	**99.29**	**99.76**

Table 5.19: Point gesture recognition results for 3 classes.

Pruning	Alpha	Classifier	Bai	Beris
15	0.25	1NN	18.16	17.71
15	0.25	3NN	27.58	26.23
15	0.25	5NN	31.45	31.34
15	0.50	1NN	22.10	25.62
15	0.50	3NN	32.99	36.00
15	0.50	5NN	36.95	41.36
15	0.75	1NN	25.09	30.96
15	0.75	3NN	36.18	42.10
15	**0.75**	**5NN**	**40.21**	**47.41**
20	0.25	1NN	20.40	17.58
20	0.25	3NN	29.48	25.07
20	0.25	5NN	33.59	29.17
20	0.50	1NN	25.05	25.17
20	0.50	3NN	35.26	34.49
20	0.50	5NN	39.51	39.36
20	0.75	1NN	27.61	30.99
20	0.75	3NN	37.69	40.62
20	0.75	5NN	41.84	45.21
25	0.25	1NN	20.94	16.64
25	0.25	3NN	29.64	23.70
25	0.25	5NN	33.74	27.99
25	0.50	1NN	25.55	24.08
25	0.50	3NN	35.25	33.47
25	0.50	5NN	39.59	38.32
25	0.75	1NN	27.97	29.54
25	0.75	3NN	37.72	39.65
25	0.75	5NN	41.90	44.31

Table 5.20: Turn page gesture recognition results for 45 classes.

Pruning	Alpha	Classifier	Bai	Beris
15	0.25	1NN	41.14	41.90
15	0.25	3NN	54.70	54.34
15	0.25	5NN	59.80	60.44
15	0.50	1NN	51.42	58.60
15	0.50	3NN	66.70	71.43
15	0.50	5NN	71.86	76.68
15	0.75	1NN	58.15	69.28
15	0.75	3NN	72.99	79.98
15	**0.75**	**5NN**	**77.46**	**83.71**
20	0.25	1NN	44.62	43.45
20	0.25	3NN	57.28	53.78
20	0.25	5NN	62.14	58.33
20	0.50	1NN	55.84	59.82
20	0.50	3NN	69.49	71.18
20	0.50	5NN	74.18	75.66
20	0.75	1NN	61.22	70.23
20	0.75	3NN	74.24	80.15
20	0.75	5NN	78.47	83.47
25	0.25	1NN	46.70	42.49
25	0.25	3NN	58.83	52.24
25	0.25	5NN	63.57	57.58
25	0.50	1NN	57.83	58.39
25	0.50	3NN	70.99	69.78
25	0.50	5NN	75.62	74.88
25	0.75	1NN	62.80	68.99
25	0.75	3NN	75.53	79.25
25	0.75	5NN	79.71	82.85

Table 5.21: Turn page gesture recognition results for 9 classes.

Pruning	Alpha	Classifier	Bai	Beris
15	0.25	1NN	64.20	60.47
15	0.25	3NN	76.10	70.84
15	0.25	5NN	79.84	76.30
15	0.50	1NN	74.63	76.20
15	0.50	3NN	87.02	86.09
15	0.50	5NN	90.20	90.22
15	0.75	1NN	79.70	85.14
15	0.75	3NN	90.93	92.54
15	0.75	5NN	93.56	95.22
20	0.25	1NN	68.81	63.41
20	0.25	3NN	79.87	71.33
20	0.25	5NN	83.34	75.29
20	0.50	1NN	81.02	79.89
20	0.50	3NN	90.97	86.88
20	0.50	5NN	93.57	90.32
20	0.75	1NN	85.13	86.77
20	0.75	3NN	93.52	93.21
20	0.75	5NN	95.54	95.56
25	0.25	1NN	70.37	63.05
25	0.25	3NN	80.45	71.27
25	0.25	5NN	84.38	75.77
25	0.50	1NN	82.04	78.37
25	0.50	3NN	91.71	87.25
25	0.50	5NN	94.72	90.65
25	0.75	1NN	85.71	86.21
25	0.75	3NN	94.15	93.39
25	**0.75**	**5NN**	**96.54**	**95.58**

Table 5.22: Turn page gesture recognition results for 3 classes.

- As opposed to what intuition suggests, less DCE pruning does not produce better recognition results. This result is related to the maximum complexity of the shapes produced by hands, which is limited. So it seems that higher DCE pruning parameter values add clutter in the representation, instead of discriminating information.
- There is an inherent difficulty to distinguish between some gesture classes (e.g., grab and page turn), so the collection of gesture definitions should be reconsidered.

5.6.1.2 PNN and 2-Fold Crossvalidation

The next test set involved a classification using Probabilistic Neural Networks (PNN) [194]. Like in the previous subsection, recognition has been performed using 45, 9 and 3 classes. We separate their results because there was a change in the experimental design: 20 experiments, using a 2-Fold Crossvalidation on the 40 percent of the database were performed. Two values of the PNN's variance parameter were tested: 0.1 and 2.0. Total (all classes) average recognition results for 45, 9 and 3 classes are shown in tables 5.23, 5.24 and 5.25. Separate recognition results for the "grab" gesture are presented in tables 5.26, 5.27 and 5.28. Separate recognition results for the "point" gesture are presented in tables 5.29, 5.30 and 5.31. Separate recognition results for the "turn page" gesture are shown in tables 5.32, 5.33 and 5.34.

Since our graph distance measure takes values in the range $[0, 1]$, it is natural that results with a variance parameter value 2.0 are worse than with a value 0.1 . Overall, these tests agree with the conclusions of the k-NN tests.

Pruning	α	σ	Bai	Beris
15	0.25	0.1	17.78	21.33
15	0.25	2.0	15.22	16.83
15	0.50	0.1	22.39	27.50
15	0.50	2.0	18.94	23.44
15	0.75	0.1	27.44	30.72
15	0.75	2.0	23.11	26.11
20	0.25	0.1	22.39	23.22
20	0.25	2.0	20.50	19.28
20	0.50	0.1	27.06	28.89
20	0.50	2.0	24.61	25.17
20	0.75	0.1	30.50	30.61
20	0.75	2.0	27.94	27.00
25	0.25	0.1	21.78	23.50
25	0.25	2.0	19.44	20.06
25	0.50	0.1	27.83	27.50
25	0.50	2.0	25.44	25.39
25	0.75	0.1	32.28	30.89
25	0.75	2.0	29.33	28.56

Table 5.23: PNN global recognition results for 45 classes.

Pruning	α	σ	Bai	Beris
15	0.25	0.1	37.83	51.17
15	0.25	2.0	32.89	46.33
15	0.50	0.1	48.56	63.78
15	0.50	2.0	42.33	59.00
15	0.75	0.1	58.06	72.00
15	0.75	2.0	50.94	66.94
20	0.25	0.1	44.44	54.33
20	0.25	2.0	40.50	50.17
20	0.50	0.1	56.50	64.33
20	0.50	2.0	50.72	61.00
20	0.75	0.1	64.78	69.00
20	0.75	2.0	59.56	65.00
25	0.25	0.1	44.61	54.17
25	0.25	2.0	40.67	49.89
25	0.50	0.1	57.33	62.56
25	0.50	2.0	53.50	60.11
25	0.75	0.1	65.22	67.39
25	0.75	2.0	61.39	64.28

Table 5.24: PNN global recognition results for 9 classes.

Pruning	α	σ	Bai	Beris
15	0.25	0.1	64.67	75.11
15	0.25	2.0	60.50	71.11
15	0.50	0.1	73.72	82.28
15	0.50	2.0	69.17	78.61
15	0.75	0.1	80.11	87.94
15	0.75	2.0	75.17	83.11
20	0.25	0.1	70.56	79.11
20	0.25	2.0	67.50	76.94
20	0.50	0.1	78.06	85.44
20	0.50	2.0	74.00	83.28
20	0.75	0.1	84.89	86.28
20	0.75	2.0	80.78	83.29
25	0.25	0.1	70.78	81.00
25	0.25	2.0	67.94	78.33
25	0.50	0.1	81.28	84.89
25	0.50	2.0	78.22	83.22
25	0.75	0.1	87.61	85.67
25	0.75	2.0	84.67	83.61

Table 5.25: PNN global recognition results for 3 classes

Pruning	α	σ	Bai	Beris
15	0.25	0.1	4.17	6.00
15	0.25	2.0	3.33	4.67
15	0.50	0.1	7.83	9.50
15	0.50	2.0	6.00	6.83
15	0.75	0.1	11.83	12.17
15	0.75	2.0	7.83	8.67
20	0.25	0.1	4.00	8.17
20	0.25	2.0	3.17	6.67
20	0.50	0.1	7.50	11.67
20	0.50	2.0	5.83	8.83
20	0.75	0.1	10.33	10.67
20	0.75	2.0	9.00	8.17
25	0.25	0.1	5.67	8.67
25	0.25	2.0	4.00	7.33
25	0.50	0.1	10.83	8.33
25	0.50	2.0	10.17	7.83
25	0.75	0.1	14.17	8.67
25	0.75	2.0	13.50	8.50

Table 5.26: PNN grab gesture recognition results for 45 classes.

Pruning	α	σ	Bai	Beris
15	0.25	0.1	17.17	27.00
15	0.25	2.0	12.17	24.50
15	0.50	0.1	28.83	38.67
15	0.50	2.0	23.17	32.33
15	0.75	0.1	39.50	52.17
15	0.75	2.0	32.17	43.33
20	0.25	0.1	17.33	30.83
20	0.25	2.0	13.83	26.67
20	0.50	0.1	28.50	38.00
20	0.50	2.0	23.33	32.67
20	0.75	0.1	40.83	41.67
20	0.75	2.0	33.67	36.67
25	0.25	0.1	20.83	32.83
25	0.25	2.0	17.33	29.33
25	0.50	0.1	37.67	39.17
25	0.50	2.0	33.67	35.50
25	0.75	0.1	48.17	43.50
25	0.75	2.0	44.17	38.50

Table 5.27: PNN grab gesture recognition results for 9 classes.

Pruning	α	σ	Bai	Beris
15	0.25	0.1	27.67	45.00
15	0.25	2.0	21.33	39.83
15	0.50	0.1	45.83	53.33
15	0.50	2.0	38.33	46.50
15	0.75	0.1	59.17	66.83
15	0.75	2.0	50.33	55.67
20	0.25	0.1	26.50	59.00
20	0.25	2.0	20.50	58.17
20	0.50	0.1	40.67	63.00
20	0.50	2.0	32.00	60.50
20	0.75	0.1	59.17	63.00
20	0.75	2.0	49.67	58.50
25	0.25	0.1	30.00	60.67
25	0.25	2.0	23.83	60.17
25	0.50	0.1	81.28	62.83
25	0.50	2.0	46.17	60.50
25	0.75	0.1	68.33	63.33
25	0.75	2.0	62.17	59.83

Table 5.28: PNN grab gesture recognition results for 3 classes.

Pruning	α	σ	Bai	Beris
15	0.25	0.1	20.50	24.17
15	0.25	2.0	17.50	20.17
15	0.50	0.1	24.17	28.17
15	0.50	2.0	20.33	24.50
15	0.75	0.1	28.33	29.67
15	0.75	2.0	23.17	25.17
20	0.25	0.1	28.67	25.67
20	0.25	2.0	26.33	22.67
20	0.50	0.1	32.17	29.67
20	0.50	2.0	29.67	25.50
20	0.75	0.1	33.17	31.33
20	0.75	2.0	30.83	27.17
25	0.25	0.1	26.33	27.50
25	0.25	2.0	23.67	23.67
25	0.50	0.1	30.50	30.33
25	0.50	2.0	28.00	26.50
25	0.75	0.1	34.67	32.00
25	0.75	2.0	30.50	29.67

Table 5.29: PNN point gesture recognition results for 45 classes.

Pruning	α	σ	Bai	Beris
15	0.25	0.1	52.67	65.33
15	0.25	2.0	49.17	63.00
15	0.50	0.1	59.83	71.83
15	0.50	2.0	54.50	69.67
15	0.75	0.1	67.67	76.17
15	0.75	2.0	61.50	74.00
20	0.25	0.1	63.17	71.00
20	0.25	2.0	57.00	68.83
20	0.50	0.1	73.50	74.00
20	0.50	2.0	67.67	72.17
20	0.75	0.1	77.83	76.67
20	0.75	2.0	76.33	73.00
25	0.25	0.1	61.33	69.67
25	0.25	2.0	57.17	66.00
25	0.50	0.1	67.33	72.83
25	0.50	2.0	65.00	72.67
25	0.75	0.1	71.83	74.17
25	0.75	2.0	69.17	72.00

Table 5.30: PNN point gesture recognition results for 9 classes.

Pruning	α	σ	Bai	Beris
15	0.25	0.1	96.33	98.67
15	0.25	2.0	96.17	98.50
15	0.50	0.1	98.83	99.83
15	0.50	2.0	97.83	99.83
15	0.75	0.1	100.00	99.83
15	0.75	2.0	99.83	99.83
20	0.25	0.1	93.67	98.17
20	0.25	2.0	91.67	98.50
20	0.50	0.1	98.33	99.83
20	0.50	2.0	96.67	100.00
20	0.75	0.1	99.67	100.00
20	0.75	2.0	99.67	100.00
25	0.25	0.1	95.33	98.50
25	0.25	2.0	94.33	96.83
25	0.50	0.1	99.17	100.00
25	0.50	2.0	98.67	100.00
25	0.75	0.1	99.83	100.00
25	0.75	2.0	99.83	100.00

Table 5.31: PNN point gesture recognition results for 3 classes.

Pruning	α	σ	Bai	Beris
15	0.25	0.1	28.67	33.83
15	0.25	2.0	24.83	25.67
15	0.50	0.1	35.17	44.83
15	0.50	2.0	30.50	39.00
15	0.75	0.1	42.17	50.33
15	0.75	2.0	38.33	44.50
20	0.25	0.1	34.50	35.83
20	0.25	2.0	32.00	28.50
20	0.50	0.1	41.50	45.33
20	0.50	2.0	38.33	41.17
20	0.75	0.1	48.00	49.83
20	0.75	2.0	44.00	45.67
25	0.25	0.1	33.33	34.33
25	0.25	2.0	30.67	29.17
25	0.50	0.1	42.17	43.83
25	0.50	2.0	38.17	41.83
25	0.75	0.1	48.00	52.00
25	0.75	2.0	44.00	47.50

Table 5.32: PNN turn page gesture recognition results for 45 classes.

Pruning	α	σ	Bai	Beris
15	0.25	0.1	43.67	61.17
15	0.25	2.0	37.33	51.50
15	0.50	0.1	57.00	80.83
15	0.50	2.0	49.33	75.00
15	0.75	0.1	67.00	87.67
15	0.75	2.0	59.17	83.50
20	0.25	0.1	52.83	61.17
20	0.25	2.0	50.67	55.00
20	0.50	0.1	67.50	81.00
20	0.50	2.0	61.17	78.17
20	0.75	0.1	75.67	88.67
20	0.75	2.0	68.67	85.33
25	0.25	0.1	51.67	60.00
25	0.25	2.0	47.50	54.33
25	0.50	0.1	67.00	75.67
25	0.50	2.0	61.83	72.17
25	0.75	0.1	75.67	84.50
25	0.75	2.0	70.83	82.33

Table 5.33: PNN turn page gesture recognition results for 9 classes.

Pruning	α	σ	Bai	Beris
15	0.25	0.1	70.00	81.67
15	0.25	2.0	64.00	75.00
15	0.50	0.1	76.50	93.67
15	0.50	2.0	71.33	89.50
15	0.75	0.1	81.17	97.17
15	0.75	2.0	75.33	93.83
20	0.25	0.1	91.50	80.17
20	0.25	2.0	90.33	74.17
20	0.50	0.1	95.17	93.50
20	0.50	2.0	93.33	89.33
20	0.75	0.1	95.83	95.83
20	0.75	2.0	93.00	93.17
25	0.25	0.1	87.00	83.83
25	0.25	2.0	85.67	78.00
25	0.50	0.1	91.67	91.83
25	0.50	2.0	89.83	89.17
25	0.75	0.1	94.67	93.67
25	0.75	2.0	92.00	91.00

Table 5.34: PNN turn page gesture recognition results for 3 classes.

5.6.2 MPEG-7 Database

In this case 100 experiments using 1-NN, 3-NN and 5-NN classifiers have been executed to obtain the average performance results shown in the tables. Since the Matlab implementation of the *Bai* algorithm made available by the author does not consider holes in the shapes but this database contains images with holes in them, two experimental design scenarios have been proposed. In the first one the whole database has been used, but the *Beris* algorithm ignores the holes as the *Bai* algorithm. We will call this approach *Complete*. The results for the Complete case are shown in table 5.35. In the second case, all of the classes containing image samples with holes in them have been discarded. We will denote this approach as *Partial*. The results for the Partial approach are shown in table 5.36.

Reviewing this results, we come to several conclusions:

- In this particular case the best results are obtained for the higher DCE pruning values (lower DCE pruning, less information is discarded).

- The recognition rate increases with the number of sample images, but increasing the computational cost.

- *Beris* outperforms *Bai* for lower values of Alpha. The explanation of this behavior may lye in the fact that there are a lot of rotational variations in the image samples, and the normalization procedure of our graph computation procedure does not include orientation normalization. Therefore, rotated shapes are treated as different shapes by our recognition algorithm. Consequently, in this case the DT value is more discriminant because it is more robust against rotations in the shape. That is, in this database shapes are better identified by their distribution of DT values than by the position of the skeleton singular points (endpoints and joint points).

5.7 Conclusions

The tests in this chapter focused in three different aspects of : realtime skeleton computation, and skeleton stability and shape recognition performance comparison. We compare our skeleton computation and pruning algorithm *Beris* against a well known recent approach, *Bai* .

Test results confirm that *Beris* is a computationally efficient procedure, stable under noise in the shape boundary, which means that similar shapes produce similar skeletons, and also suitable for shape recognition.

Pruning	Alpha	Classifier	Bai	Beris
15	0.25	1NN	23.31	22.59
15	0.25	3NN	38.02	36.92
15	0.25	5NN	71.98	74.60
15	0.50	1NN	28.79	28.38
15	0.50	3NN	44.93	45.26
15	0.50	5NN	70.52	70.66
15	0.75	1NN	30.38	30.85
15	0.75	3NN	46.32	48.56
15	0.75	5NN	61.57	59.86
20	0.25	1NN	25.02	23.35
20	0.25	3NN	40.28	37.62
20	0.25	5NN	72.04	69.19
20	0.50	1NN	30.33	29.14
20	0.50	3NN	46.93	46.08
20	0.50	5NN	73.80	73.36
20	0.75	1NN	32.02	32.24
20	0.75	3NN	48.42	48.18
20	0.75	5NN	62.98	59.22
25	0.25	1NN	26.06	24.00
25	0.25	3NN	37.54	34.17
25	0.25	5NN	49.50	56.06
25	0.50	1NN	31.51	29.45
25	0.50	3NN	43.23	40.88
25	0.50	5NN	55.53	53.61
25	0.75	1NN	32.65	31.49
25	0.75	3NN	44.41	43.22
25	0.75	5NN	56.32	45.87

Table 5.35: Total recognition results for the whole database not considering the holes in the shapes.

Pruning	Alpha	Classifier	Bai	Beris
15	0.25	1NN	24.95	26.08
15	0.25	3NN	46.07	46.61
15	0.25	5NN	64.21	65.01
15	0.50	1NN	31.29	32.91
15	0.50	3NN	55.35	56.85
15	0.50	5NN	74.88	74.44
15	0.75	1NN	33.78	36.10
15	0.75	3NN	57.90	60.89
15	0.75	5NN	77.50	81.56
20	0.25	1NN	26.61	26.36
20	0.25	3NN	48.03	46.26
20	0.25	5NN	66.07	64.66
20	0.50	1NN	32.99	33.03
20	0.50	3NN	57.29	55.85
20	0.50	5NN	76.70	75.71
20	0.75	1NN	35.05	36.41
20	0.75	3NN	59.60	60.00
20	0.75	5NN	79.15	80.34
25	0.25	1NN	29.51	27.38
25	0.25	3NN	50.84	47.45
25	0.25	5NN	69.14	66.20
25	0.50	1NN	35.63	33.95
25	0.50	3NN	58.99	57.43
25	0.50	5NN	78.16	77.52
25	0.75	1NN	36.91	36.70
25	0.75	3NN	60.45	61.50
25	0.75	5NN	79.79	81.57

Table 5.36: Total recognition results for part of the database removing image classes containing holes on any of their samples.

Chapter 6

Conclusions

The PhD research work presented in this report has been motivated by the immersion of the PhD candidate in the industry. This special environment has positive and negative aspects. On the positive side, the problems addressed have an actual application to a real product. On the negative side, conditioning the research work to the company strategic policies causes shifts of focus and some disorientation. From a methodological point of view, the company and the academy are almost orthogonal. Where the emphasis for the company would be the realization of a working product without further assessment of the virtues and limitations of the approach, the academy emphasis is on formal description, proof and systematic validation. The PhD candidate has been fortunate to have a taste of both worlds, which is reflected on this thesis report.

From the industry point of view, this PhD work has produced a working real-time prototype of hand gesture recognition that could be embedded in a working tabletop. Further work in this line of interest would be the actual physical realization of the tabletop, which implies assembling the different elements (cameras, multitouch surface, projectors, computers, in a working whole that shows some of the desired properties of the IMT. This realization would also provide an experimental workbench to gather measures of the performance of the proposed recognition system on a realistic environment with real users doing real interactions.

From the academy point of view, we have achieved several of the proposed objectives:

- We have set the stage for the research on tabletops, doing a deep and broad review of IMT fundamental ideas and realizations.

- We have performed an exhaustive review of the literature of skeleton computation and regularization, including shape matching.
- We have proposed an innovative efficient and stable skeletonization algorithm, based on Voronoi skeletonization and Discrete Curve Evolution. The algorithm speedup is due to two facts
  - To decide that the entire Voronoi edge is interior to the shape we need only to know if its extreme points are interior to the shape, which we have formally proved in chapter 4.
  - The entire Voronoi segment can be pruned with only one test, we do not need to process all the points separately.
- We have quantitatively assessed the stability improvement of our algorithm over a state of the art algorithm.
- We have proposed a greedy algorithm to compute a distance between graphs, which can be tuned to give more importance to the matching error over the regularization.
- We have performed systematic classification experiments to asses the improvement of our algorithm over a state of the art algorithm, We have followed a systematic methodology:
  - We have collected a experimental database of hand gesture images,
  - We have computed the skeletons and the graph representations based on competing skeletonization algorithms. That is, we have successfully implemented our algorithm and the competing one.
  - We have performed two fold crossvalidation experiments to asses the expected generalization of the classification algorithms. To avoid biases due to the classifier system and its learning algorithm, we have used the simplest non parametric ones: 1-NN, 3-NN, 5-NN, and the Probabilistic Neural Networks (PNN).
- From the point of view of the proposed application, IMT visual interaction, we have obtained good results on the recognition of a hand gesture language composed of three dynamic gestures.

- The algorithm sensitivity to a key parameter, the DCE stopping condition has been explored. It has been shown that skeleton stability and pattern recognition improves decreasing it. The balance between the pruning and the pattern recognition performance has not exhaustively explored (i.e. testing very low values) but it is already surprising that the detail implied by higher values of this parameter do not seem to provide additional discriminative power.

- We performed additional tests on a well known freely available image database which has been used for the definition of video standards. For this database the improvements in pattern recognition were not conclusive. We think that the labeling noise of the database has had some effect in the algorithms performance.

- From the point of view of comparison with other algorithms, our approach has improved the competing algorithm almost in all cases.

- It is one of the few real time skeletonization algorithms proposed, and the only one we know whose code has been published.

Lack of time, has prevented us to explore more sophisticated distances among graphs and matching procedures, as well as skeleton mapping into graphs. Most of them are not feasible for real time process at the current computing resources, but some of them could be in a near future.

Another issue for further research would be the application of the approach to other gesture languages and environments, such as body poses and body language for affective computing.

We find that the idea of clustering shapes based on their skeleton features may be an exciting and fruitful avenue of research when enhanced skeletonization algorithms have been proposed. This idea has been proposed and discarded all along the years, mainly because the poor stability of the skeletonization algorithms did not ensure the preservation of the perceptual similarity between shapes in the skeleton representation. Therefore, clustering skeletons might lead to identifying quite different (perceptually) shapes. Stable skeletonization algorithms provide this perceptual continuity which may allow the realization of meaningful clustering algorithms. One potential application would be the on-line unsupervised or semisupervised learning of gestures in IMT interaction, both for training, and for application tailoring or user modeling.

Finally, we have found in the literature that much of the work done in the past decade on skeletonization was in the area of computer graphics and surface modeling in 3D. The application of our algorithm in such fields may deserve further efforts and may lead to interesting results.

Everything should be made as simple as possible, but not simpler.

Albert Einstein

Appendix A

Multitouch displays

This appendix is structured as follows: section A.1 presents this kind of displays, and the rest of sections present different multitouch display technologies, optical in section A.2, capacitive in section A.3, digital resistive in section A.4 and acoustic in section A.5.

A.1 Introduction

> *Multi-touch (or multitouch) denotes a set of interaction techniques which allow computer users to control graphical applications with several fingers.*
>
> *Multi-touch consists of a touch screen (screen, table, wall, etc.) or touchpad, as well as software that recognizes multiple simultaneous touch points, as opposed to the standard touchscreen (e.g. computer touchpad, ATM), which recognizes only one touch point. This effect is achieved through a variety of means, including but not limited to: heat, finger pressure, high capture rate cameras, infrared light, optic capture, tuned electromagnetic induction, ultrasonic receivers, transducer microphones, laser rangefinders, and shadow capture[195].*
>
> *Many applications for multi-touch interfaces exist and are being proposed. Multi-touch is often associated with Apple Inc's iPhone and iPod Touch but is also used in many other products such as Apple's MacBook and MacBook Pro notebook line. Other products with multi-touch technology include Microsoft Surface,*

> *Asus EEE PC, Meizu M8 and the upcoming Zune HD.*
>
> *Modern multi-touch controllers support Single-Touch and Multi-Touch All-Point touchscreen applications which allow functions such as playing video games on a mobile handset, using GPS to key in multiple locations, etc.*[1][2]

After this introductory definition of multitouch, extracted from the Wikipedia, it must be remarked that this appendix is focused only on multitouch screens or devices which can transform an ordinary screen into a multitouch device. Any kind of multitouch surface which does not permit direct interaction on a GUI shown on or just below it is not covered since this research work is focused on natural interaction, and direct interaction plays an important role to achieve it. Another comment on this definition is the fact that not only fingers can be used for interaction in many multitouch surfaces, but also other kind of objects.

To fit multitouch into a historical context, the first multitouch attempts date back to 1982, when the University of Toronto developed the first finger pressure multitouch display[196]. A year later, Bell Labs at Murray Hill published what is believed to be the first paper discussing touch-screen based interfaces [197]. The same year Krueger presented his Video Place and Video Desk [198], which represented a big step forward for multitouch, whose later works gave even more momentum to the research on multitouch devices[29, 199]. After those pioneering, the interest of about multitouch interfaces of the research community slowly vanished. Later, in 2005, Han introduced FTIR multitouch technology and several valuable applications [200], bringing the interest back to the research community. And in 2007 Apple introduced multitouch screens into mobile phones with his iPhone, and Microsoft presented his multitouch tabletop, called Surface, and coined the term Surface Computing, giving a huge impulse to the interest on multitouch interaction. At this moment, several research groups are working using FTIR multitouch technology, and are focused in creating natural interaction paradigms, exploiting all the possibilities of multitouch interaction. Graphical User Interfaces are also a hot research topics for interactive multitouch, in spite of being a well established technology. For a deeper revision of multitouch history, the reader is suggested to review Bill Buxton's work, available

[1] Extracted from the Wikipedia (http://en.wikipedia.org/wiki/Multi-touch)

[2] Notice that the term tactile display is used to refer to displays offering a haptic feedback in the form of some tactile effect simulation.

at his personal web page [201].

On the other hand multitouch can be seen as a logic evolution of touch displays. Taking this point of view as basis, touch interaction can be categorized into three levels, mono-touch, multi-point and multi-touch, and each of this levels includes all the interaction possibilities of the previous.

Monotouch interaction is analog to ordinary electronic mouse pointer interaction. There is only one pointer and right click is not naturally available. Usually, only touch coordinates are considered, but in some cases drag paths are also recognized. Multi-point is a generalization to many points of monotouch interaction. The position of n touches is recognized and the paths followed by drag operations of each point are also recognized. Finally, real multi-touch interaction should also include other information sources, and combinations with all of the rest information available. In this way and according to[78] multi-touch gestures can be categorized based in their input properties

- Position property: Fingers' contact position (x, y) is used, for example, to select an object.
- Motion property: Both velocity and acceleration can be used, for example to use physical elements in the GUI like forces and inertia.
- Physical property: This includes the size of the contact area, the shape, orientation and pressure degree. Orientation has never been used, mainly because it is not available for most of the devices. The use of the contact area shape is also rare in the literature. And the use of the rest of the properties in this category is not frequent. Despite this lack of interest, we think that new interactions could be based in this kind of information. For example touching with the whole hand, palm down, and the fingers spread could be used to show a contextual menu.
- Event property: This category includes tap and flick. Usually to select or activate objects.

Gestures combining different properties have not been explored yet.

There are several multitouch gestures that have become *de facto* standards to specify how to resize, rotate and move objects on the GUI. This gestures are available in the Microsoft Surface and the Apple iPhone, and in many other actual systems. Touching an object with two or more fingers and

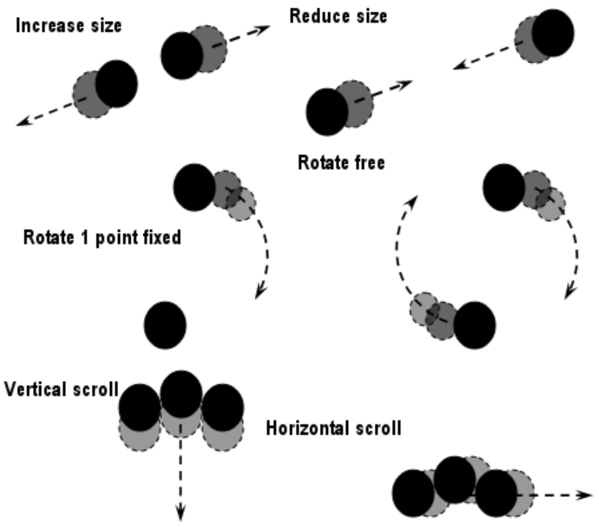


Figure A.1: **Common multitouch gestures**. Ellipses represent pressure positions. Black ellipses represent initial pressure positions, and brighter ellipsis represent intermediate and final pressure positions.

approaching them reduces the size of an object. Putting the fingers apart increases the object's size. Touching an object with several fingers and rotating them, or using one as pivot and rotating the other one, rotates the object. Touching an object with several fingers and dragging is used for scrolling (see fig. A.1 for some graphical example drafts).

Some considerations about multitouch interaction:

- It is a direct interaction method, therefore it is natural, and has some inherent limitations like reaching distant objects.
- It can produce rejection in people concerned about hygiene, because there are many people touching the same surface.
- Many simultaneous users must be well managed by the system.
- Graphical User Interfaces must be adapted to support multiple simultaneous users, and to permit comfortable finger interaction, since fingers are bigger and less accurate than the mouse pointer.

As it is documented in [195, 9], there are several technologies that can be used to build multitouch surfaces. These technologies can be categorized as either optical or Computer Vision based, resistance based, capacitive based and acoustic based. Next, most multitouch technologies available nowadays are described, and their strengths and weaknesses are pointed out.

A.2 Optical Multitouch Surfaces

A set of techniques are covered under this name, which consist of an optical sensor, usually a camera, a light source, usually infrared, and a visual feedback usually in the form of a video projection but also in the form of an LCD screen. Currently most of the available multitouch devices make use of this technology because of its scalability, low cost, and easy setup [9]. The Natural User Interface Group (NUI), has recently proposed a document, which is still under revision stage, describing most of this techniques. Part of this section is extracted from that book, which is freely accessible via web at the official web page of the group [202].

There are many kind of optical multitouch techniques which are described next, but first some of the characteristics of the elements used by these techniques must be introduced. The main reason for this introduction is that the performance of the multitouch surface greatly depends on the proper selection of the components that make it up.

Infrared Light Sources

Infrared (IR henceforth) is a portion of the light spectrum that lies just beyond what can be seen by the human eye, therefore it is invisible for the human eye. It is a range of wavelengths longer than visible light, but shorter than microwaves. *Near Infrared* (NIR) is the lower end of the infrared light spectrum and typically considered wavelengths between 700nm and 1000nm (nanometers).

Most digital camera sensors are also sensitive to at least NIR and are often fitted with a filter to remove that part of the spectrum so they only capture the visible light spectrum. By removing the infrared filter and replacing it with one that removes the visible light instead, a camera that only sees infrared light can be created. In this way, ordinary cameras can be used instead of specific infrared cameras, which are usually more expensive.

Most of the techniques explained later could also be used with light sources belonging to any other part of the light spectrum, including the human visible spectrum portion. The reason for this choice is twofold, first because the visual feedback is emitted (projector or LCD) in the same physical surface where the light source for multitouch recognition is shown. The optical sensor pointing at that surface needs a way to distinguish between the visual feedback and the multitouch light source. A way to avoid interferences between the light coming from that visual feedback and the one coming from the multitouch system is to use infrared light for the multitouch system and a camera which can only "see" that light spectrum portion. Consequently, the camera will only see the light used by the multitouch technique. The second reason is because IR is invisible to human eyes, so it does not interfere with users' view of the visual feedback shown in the surface.

Usually IR LEDs are used, because they are durable, have low heat emission and produce constant light. Several single IR LEDs can be wired, but IR LED ribbons are also available, which are easier to setup. And for certain techniques, like Diffused Illumination, even LED emitters can also be used.

Recommended characteristics for IR LEDs for multitouch screens:

1. Wavelength: 780-940nm. LEDs in this range are easily seen by most cameras and visible light filters can be easily found for these wavelengths. The lower the wavelength, the higher sensitivity which equates to a higher ability to determine the pressure.

2. Radiant intensity: Minimum of 80mw. The ideal is the highest radiant intensity you can find, which can be much higher than 80mw.

3. Angle for FTIR: Anything less than +/- 48 will not take full advantage of total internal refection, and anything above +/- 48 degrees will escape the acrylic. In order to ensure there is coverage, going beyond +/- 48 degrees is fine, but anything above +/- 60 is really just a waste as (60 - 48 = +/- 12 degrees) will escape the acrylic.

4. Angle for diffused illumination: Wider angle LEDs are generally better. The wider the LED angle, the easier it is to achieve even illumination.

Infrared Cameras

Most of web-cams and regular cameras can sense IR light but need to be modified first. These cameras have an IR light blocking filter which must be removed. Even with this filter on them, part of the near infrared light can be sensed by the cameras, but their sensitivity to IR light will improve removing the filter. That IR blocking filter needs to be replaced by a band pass filter which only lets IR light pass. This procedure is usually simple for cheap cameras, opening the camera pops out the filter, but more expensive cameras can have the filter applied directly to the lens. In this case, the lens needs to be replaced.

There are also IR light sensitive cameras, which are best suited for this task. The main reason to use regular cameras is that they are usually cheaper than specific IR cameras. On the other hand, IR cameras offer much better sensitivity to the IR light spectrum portion. Another characteristic of specific IR cameras is that they also capture “human visible” light spectrum portion, but they are more sensitive to light in the wavelengths of IR light. Therefore, this cameras also need a band pass filter.

The most important characteristics to select a camera for an optical multitouch screen are:

- Resolution: Measured in pixels, defines the number of columns and rows of colored pixels in which the camera subsamples the captured scene. The precision of the touch device depends directly on this parameter. For small multi-touch surfaces a low resolution web-cam (320 x 240 pixels) can be sufficient. Larger surfaces require cameras with a resolution of 640x480 or higher in order to maintain the accuracy.

- Frame rate: Measured in frames per second (FPS) indicates the frequency in which a camera can capture images of the scene. In order to cope with fast movements and responsiveness of the system a camera with at least a frame rate of 30 FPS is recommended. Higher frame rates provide a smoother and more responsive experience, but require higher computer overhead, since more images must be processed in one second.

- Interface: Basically USB and IEEE 1394 (FireWire). Currently FireWire can be found in more expensive cameras, but provides lower latency and higher data transmission rates than USB, therefore higher resolutions and frame rates can be achieved. And data compression can be avoided during transmission. Both USB and FireWire are in a continuous improvement, defining new standards, therefore current specifications of each technology should be reviewed before choosing one or other method. Other important interfaces include Gigabit Ethernet (GigaE), or even wireless (WI-FI).

- Lens type: Cheap cameras include a lens and an IR blocking filter behind it, which can be removed. More expensive cameras can have that filter applied to the lens directly and professional cameras do not usually include a lens. In these two cases a new lens is required. Fortunately most of the cameras use the same kind of lens mount, namely M12, C or CS mount. Then the suitable focal Length of the lens must be calculated based on the size and distance between the camera and the surface to capture.

- Camera sensor & IR bandpass filter: Before even using the camera a bandpass filter is required to block any other light which does not belong to the IR light spectrum portion (around 880 nm.). This occurs even with specific IR cameras which are specially sensitive to the IR light but can also sense other parts of the light spectrum.

Liquid Crystal Display (LCD) Technology

Although the use of video projectors is more extended for optical multitouch devices, LCD screens are also being used. Compared to video projectors, LCD panels offer better durability, higher video resolution and are less affected by environmental lighting.

In order to understand how some of the the techniques described next can be combined with the use of LCD screens it is essential to explain how this devices work.

LCD screens use electronically controlled Red-Green-Blue (RGB) filters to set the color of each pixel. LCD panels are essentially transparent when no current is running through the screen. In the front and back of the panel there are two polarizing filters criss-crossing, i.e. with a rotation of 90º, which give the black color to the panel. But polarizing filters do not polarize light in the IR spectrum. So, although it is opaque to human's eye, IR light passes through the panel. A back-light is necessary in order to illuminate the LCD pixels. For panels of less than 23" a long thin fluorescent light bulb which lines the length of the monitor is used. Attached to the bulb there is an acrylic sheet (called the light guide) which has a honey-comb pattern of white dots. Based on the principle of total-internal reflection, the light from the fluorescent tube travels inside the acrylic sheet until it reflects off the white dot. For monitors of 27" or larger, a rail of thin fluorescent lights bulbs is used.

To improve lighting conditions a layering of several different filters which modulate and affect the back light in various ways are included. Most common filters include:

- Diffuser, to disperse the light in every direction.
- Fresnel lens, based on the Fresnel principle, can magnify light with a shorter focal length in different directions. This filter is used to disperse light in 180 degrees.
- White Reflector, an opaque white filter which reflects any light that may have escaped the filters.

The only filter which impedes IR light, and it is concern when developing optical multi-touch surfaces is the last white opaque filter. All the rest can and should remain to keep optimal viewing performance.

A.2.1 Frustrated Total Internal Reflection (FTIR)

This technique was introduced by Han in 2005 [200]. It makes use of the Total Internal Reflection principle, introduced in [203], and described next.

[3] *Total internal reflection is an optical phenomenon that occurs when a ray of light strikes a medium boundary at an angle larger than the critical angle with respect to the normal to the surface. If the refractive index is lower on the other side of the boundary, no light can pass through and all of the light is reflected. The critical angle is the angle of incidence above which the total internal reflection occurs.*

When light crosses a boundary between materials with different refractive indexes, the light beam will be partially refracted at the boundary surface, and partially reflected. However, if the angle of incidence is greater (i.e. the ray is closer to being parallel to the boundary) than the critical angle — the angle of incidence at which light is refracted such that it travels along the boundary — then the light will stop crossing the boundary altogether and instead be totally reflected back internally. This can only occur where light travels from a medium with a higher refractive index to one with a lower

[3] *Extracted from the Wikipedia (http://en.wikipedia.org/wiki/Total_internal_reflection#Frustrated_total_interne*

refractive index. For example, it will occur when passing from glass to air, but not when passing from air to glass.

The specific angle at which this occurs depends on the refractive indexes of both materials, and is known as the critical angle, which can be calculated mathematically using Snell's law [204].

$$\frac{\sin\theta_1}{\sin\theta_2} = \frac{v_1}{v_2} = \frac{n_2}{n_1} \tag{A.1}$$

where $\theta_1$is the incidence angle, θ_2 is the refraction angle, $v_1$and$v_2$are the velocities in both media and n_2 and n_1 are the indexes of refraction. Frustrated Total Internal Reflection, FTIR henceforth, describes the canceling of the Total Internal Reflection of the light, when the material which has lower refraction index is replaced by another with higher refraction index than both of them. In this case, the light contained now in the material with lower refraction index reflects on the material with highest refraction index in the direction of the normal of the plane separating both materials.

For multitouch surfaces, IR light is used, invisible to the human eye which is Totally Internally Reflected inside an acrylic surface (multitouch surface), which is also called Plexiglas, surrounded by air, which has a lower refractive index. The acrylic surface is IR light flooded using several IR LEDs around the surface emitting towards inside the surface. When an object with higher refraction index than the acrylic surface, like a finger, touches the surface, the light reflects on the finger and a camera beneath the surface recognizes the touching point as a bright spot in the IR light spectrum portion. Using Computer Vision techniques like image substraction, smoothing, morphological operators and thresholding a binary image can be obtained where every touched point is highlighted. Using this technique, recognition resolution and frequency, and maximum simultaneously recognized touch amount are only limited by the intrinsic characteristics of the camera and the computational cost of the computer vision processes involved.

The basic hardware setup for a FTIR multitouch surface is shown in fig. A.2. Usually, a diffuser layer is added in order to use a projector to show the visual output on the multitouch surface, and to limit the effect of environmental IR light. And also a "compliant layer" on top of the acrylic surface to improve the coupling between objects and the surface. The compliant surface is an overlay placed above the acrylic waveguide in a FTIR based multi-touch system. In this configuration the compliant surface overlay needs to be made of a material with a similar refractive index than that of the acrylic waveguide, because Total Internal Reflection will extend to its volume, and one that will "couple" with the finger or object under pressure and produce the FTIR effect, and then "uncouple" once the pressure is released. The scheme is shown at figure A.3. Han points out that the selection of the right acrylic, diffuser and compliant layer is very important and it is still a work in progress. Anyway, he gives some advices about brands an procedures to improve the final results. In [205] some different "compliant layers" are discussed. Another configuration includes a projection surface on top, then a compliant layer under it and finally the acrylic surface. In this case the compliant layer refraction index needs to be higher than that of the acrylic surface, so when pressure is done on the projection surface the acrylic and the compliant layer "couple" and the FTIR effect is set. But also has to "uncouple" when the pressure is released on the projection surface. This scheme is shown at figure A.4.

The benefits of using a compliant surface in any of the two configurations include:

- Protects the expensive acrylic from scratches.
- Blocks most of light pollution.
- Provides consistent results (the effectiveness of the bare acrylic touch seems to be down to how sweaty/greasy the hands are).
- Zero visual disparity between the touch surface and the projection surface.
- Pressure sensitive.
- Seems to react better for dragging movements.
- Brighter blobs to track, as there is no longer a diffuser between the IR blob light and the camera.

Advantages

- Blobs have strong contrast.
- Allows for varying blob pressure.
- With a compliant surface, it can be used with something as small as a pen tip

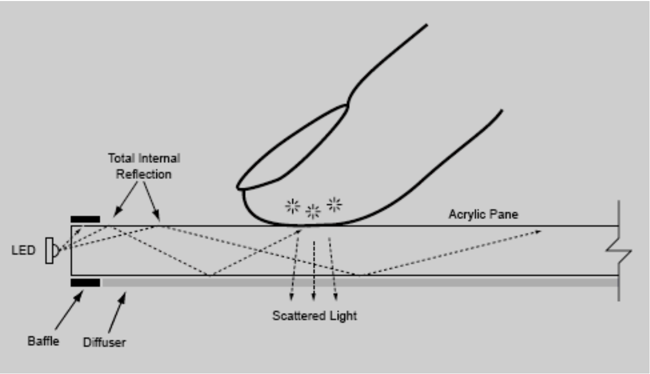


Figure A.2: **Basic FTIR description scheme**. Copyright Jeff Han.

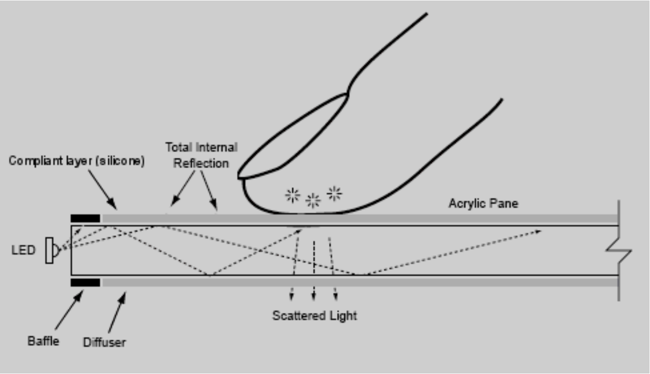


Figure A.3: **FTIR with compliant surface on top**.

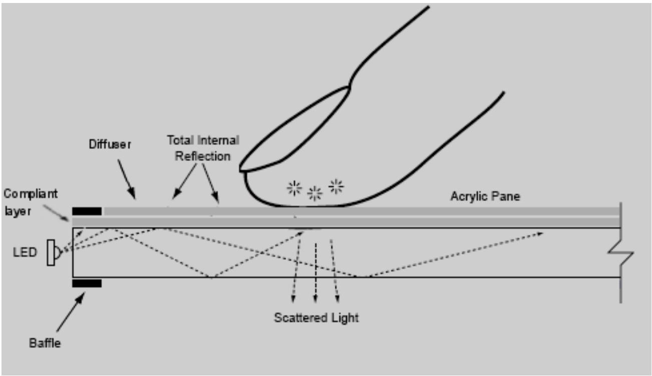


Figure A.4: **FTIR with diffuser on top and compliant layer in the middle**.

Disadvantages

- Setup calls for some type of LED frame.
- Requires a compliant surface (silicone rubber) for proper use.
- Cannot recognize objects or fiducial markers, only for objects which couple with the surface.
- Performance depends on environmental lighting.

A.2.1.1 Testing experiences

During the present research work a FTIR multitouch surface was built, in order to test the performance of this technique. The prototype used two acrylic surfaces mounted one on top of the other, one transparent and 10 mm. thick to test the FTIR effect and beneath it another 4 mm thick translucent panel playing the role of a diffuser to project the visual output. The prototype has 70" and a 4:3 ratio, and two IR LED (880 nm.) rows have been mounted in the longest sides of the surface rectangle. LEDs were distributed evenly with a separation of 20 mm. between them. An infrared camera Jai CV-M50-IR was used, which captures most of the IR spectrum portion, with a resolution of 640x480 and a capture rate of 25 FPS. A band pass filter to limit capture to IR spectrum portion has also been added.

With this setup, results were not as impressive as those presented at Han's paper. To achieve the FTIR effect hard pressure was required on the surface. And also the dragging operation was difficult to perform for the same reason. Many research teams working on FTIR multitouch agree that the only problem with this technique is that the right material for the surface has not been found yet, but they add that the use of a compliant layer, like silicone rubber, on top of the surface improves the recognition performance [205].

Another problem was environmental IR light pollution. Even at an indoor environment, sunlight coming from windows and fluorescent lighting added a lot of infrared light to the camera, producing noisy images. Therefore, FTIR should only use in IR free environments, like dark rooms. Finally, and under suitable lighting conditions, the Computer Vision algorithms required low computational resources.

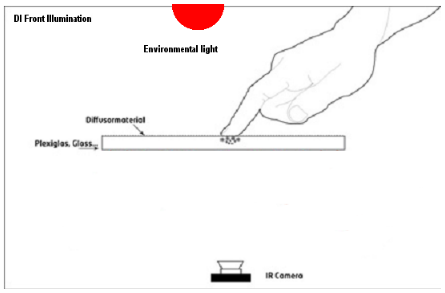


Figure A.5: **Front Diffused Illumination**

A.2.2 Diffused Illumination (DI)

Diffused Illumination comes in two flavors, Front Diffused Illumination and Rear Diffused Illumination. In both cases the contrast between the silent image and the finger that touches the surface is detected, which also lets recognizing not only touching objects, but also objects in the proximity of the surface.

A.2.2.1 Front Diffused Illumination

Visible light (often from the ambient surroundings) is shined at the screen from above the touch surface. A diffuser is placed on top or on bottom of the touch surface. When an object touches the surface, a shadow is created in the position of the object. The camera, located under the surface, senses this shadow. The diffuser limits the light amount which traverses the surface and also helps to distribute it more evenly. The scheme of this configuration is shown in A.5.

Advantages:

- No need for a compliant surface, just an diffuser/projection surface on top/bottom.
- Can use any transparent material like glass (not just acrylic).
- No LED frame required.
- Can track fingers and hovering.
- Simplest setup.

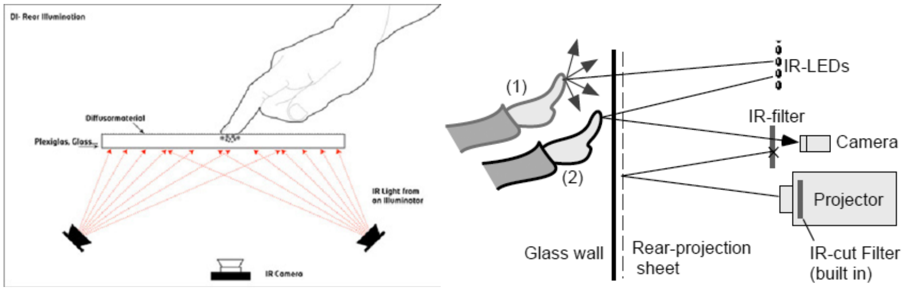


Figura A.6: Left: Simple Rear Diffused Illumination scheme. Right: Real Design[8].

Disadvantages:

- Cannot track objects and fiducials.
- Difficult to get even illumination.
- Great chance of 'false blobs'.
- Not reliable, because it relies heavily on ambient lighting environment.

A.2.2.2 Rear Diffused Illumination

In this case an IR light emitter is located under the touch surface, to illuminate the touch surface. When there is nothing on the touch surface, the IR camera next to the IR emitter captures a bright image. But when an object approaches to the surface, a shadow is recognized. Using a diffuser like in the Front Diffused Illumination technique, the light amount coming from above the surface is limited and also the infrared light emitted is more evenly distributed. In figure A.6a typical configuration is presented.

Anyway, the main problems of Front and Rear diffused Illumination are to achieve an uniform light distribution on the surface, so the recognition performance does not depend on the section of the touch surface where the object is located, and reducing the effect of external light pollution. This can be accomplished carefully selecting the diffuser and controlling the lighting conditions, and also by software, using advanced Computer Vision techniques, which usually are more computer demanding and difficult to implement.

Advantages:

- No need for a compliant surface, just an diffuser/projection surface on top/bottom.
- Can use any transparent material like glass (not just acrylic)
- No LED frame required.
- Building is simple.
- Can track objects, fingers, fiducials and hovering (no touch required).

Disadvantages:

- Difficult to get even illumination.
- Blobs have lower contrast (harder to pick up by software).
- Greater chance of 'false blobs'
- Performance depends on environmental lighting.

A.2.3 Diffused Surface Illumination (DSI)

The hardware configuration is essentially the same as FTIR. A translucent surface, several IR LEDs around it and a camera below. There is no need to use any compliant surface. The only difference is that a special acrylic is used. This acrylic uses small particles that are inside the material, acting like thousands of small mirrors. When IR light is shined into the edges of this material, the light gets redirected and spread to the surface of the acrylic. The effect is similar to DI, but with even illumination, no hot-spots, and same setup process as FTIR A.7. Manufacturers offer this kind of Endlighten panels ranging from 6 mm. to 10 mm. thick.

Advantages:

- Even distribution of light, and easy to set-up.
- An FTIR set-up can be converted to DSI easily.
- No need to build a special case for the setup.

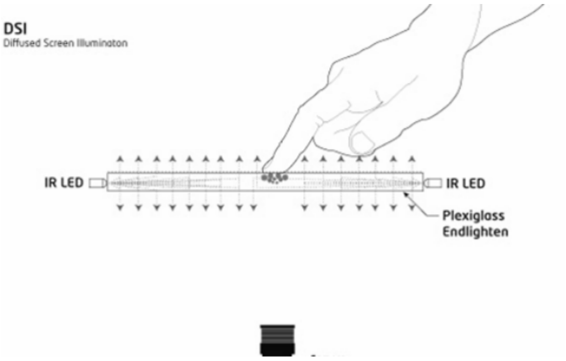


Figure A.7: **Diffused Surface Illumination scheme**.

- Fiducial, object and hovering tracking is possible.
- It is pressure sensitive.

Disadvantages:

- Less contrast compared to normal DI set-ups as the surface material also redirects the IR towards the camera.
- Potentially more problems with ambient IR because of less contrast.
- Size restrictions because of the softness of the surface material.
- Endlighten Acrylic costs more than regular acrylic (but the some of the cost can be made up since no IR illuminators are needed).
- Performance depends on environmental lighting.

A.2.4 Laser Light planE (LLP)

Infrared light from a laser or set of lasers is shined just above the surface. The laser plane of light is about 1mm thick and is positioned right above the surface, when the finger just touches it, the device will register it as a IR blob A.8.

Infrared lasers are an easy and usually inexpensive way to create a multitouch setup using the LLP method. Most setups go with two up to four

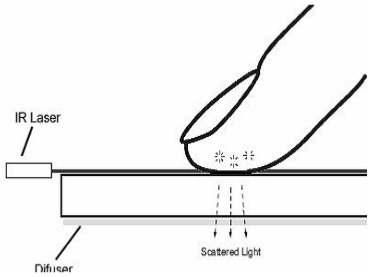


Figure A.8: **Laser light planE (LLP) scheme**.

lasers, positioned on the corners of the touch surface. The laser wattage power rating (mW,W) is related to the brightness of the laser, so the more power the brighter the IR plane will be. Laser modules need to have line lenses on them to create a light plane. The 120 degree line lens is most commonly used, so as to reduce the number of lasers necessary to cover the entire touch surface. Using lasers require some safety considerations to avoid possible damages.

Advantages:

- No compliant surface (silicone)
- Can use any transparent material like glass (not just acrylic)
- No LED frame required
- An enclosed box is not required
- One of the simplest setup.

Disadvantages:

- Cannot track traditional objects and fiducials.
- Not truly pressure sensitive (since light intensity does not change with pressure).
- Can cause occlusion if only using 1 or 2 lasers where light hitting one finger blocks another finger from receiving light. Anyway the number of simultaneous touches is limited, and also several touch combinations are not recognizable.

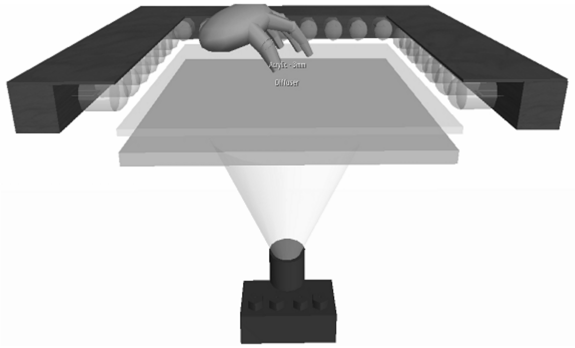


Figure A.9: **LED-LP 3D Schematic created in SecondLife**.

- Performance depends on environmental lighting.

A.2.5 LED Light Plane (LED-LP)

LED-LP is setup the same way as an FTIR setup except that the thick acrylic that the infrared light travels through is removed and the light travels over the touch surface. This picture A.9 shows the layers that are common in an LED-LP setup. Similar to LLP, this technique creates a IR plane all over the surface. The difference is that LED produce a conical light beam, therefore not only touching objects, but also objects in the proximity can be detected. This effect can be restricted using software filters, but also using a bezel which limits the light emission aperture size.

LED-LP is usually only recommended when working with an LCD screen as there are better methods such as Rear DI when using a projector that usually don't work with an LCD screen. Like Rear DI and LLP the touch surface need not be thick like in FTIR, but only as strong as it needs to support the forces from working on the touch surface.

Advantages:

- No compliant surface (silicone).
- Can use any transparent material like glass (not just acrylic).
- No LED frame required.
- An enclosed box is not required.

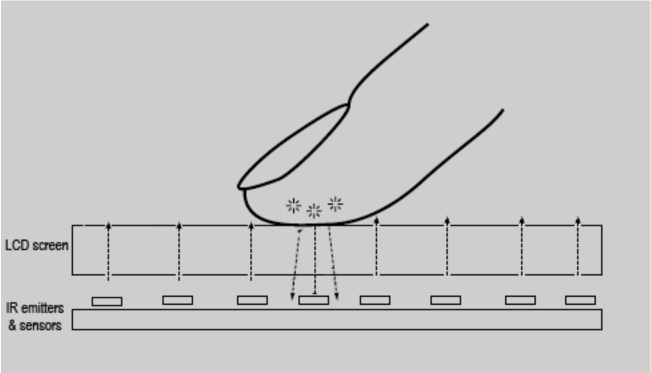


Figure A.10: **Matrix of IR Transceivers scheme**

- Could be slightly cheaper than other techniques.

Disadvantages:

- Hovering might not be detected.
- Does not track objects or fiducials.
- LED frame required.
- Only narrow-beam LEDs can be used, no ribbons.
- Performance depends on environmental lighting.

A.2.6 Matrix of IR Transceivers

This technique is conceived for LCD screens. There is a grid of IR transceivers behind the LCD panel. Each IR transceiver consists of an IR emitter and an IR detector. The emitter pulses IR light at a certain frequency which the sensor can detect. When any object touches the surface, reflects the light, like highlighted in the figure A.10.

Advantages:

- Thin form factors
- Permits tangible tracking because the fiducial pattern can be distinguished.

Disadvantages:

- Is it not easy to self-construct.
- It is not a scalable solution (e.g. a 32" screen would require thousands of sensors and a fast-processor to read through all the sensors).
- Performance depends on environmental lighting.

A.3 Capacitive Multitouch Surfaces

The main component used in this kind of devices is the capacitor. A simple parallel plate capacitor consists of two metal plates of about the same size, facing each other without touching. One of the plates is usually connected to ground while the other is attached to a voltage source. When applying a voltage to those plates, the resulting current charges the capacitor. The amount of charge it can hold depends on the voltage applied and on the capacitance of the capacitor:

$$Q = CU \tag{A.2}$$

where Q is the charge in coulomb, C is the capacitance in farad and U is the voltage in volt. The capacitance in turn can be calculated using equation.

$$C = \varepsilon_r * \varepsilon_0 * A/d \tag{A.3}$$

where d is the distance between the capacitor plates in cm, A is the area of each plate in cm^2 , ε_0 is the dielectric constant of vacuum, and ε_r is the dielectric value of the medium between the plates. This equation is only valid for capacitors where the value of A is much larger than d. By measuring the capacitance of a capacitor, one can detect:

- A change in plate size or overlapping area (A). This can be caused e.g. by sliding one of the plates.
- Change in distance between the plates (d). This can be caused by movement of one of the plates.
- A change in the dielectric value of the medium (εr). This can be caused by a object with a different dielectric constant being placed between the plates.

This means that capacitance changes can have entirely different causes. On one hand this results in ambiguous data. On the other hand it makes capacitive sensing a very versatile technique. By controlling two of the variables the third one can be measured. Common sensor designs utilize one or more of these three effects.

Taking advantage of the capacitor's characteristics mentioned above, two kind of multitouch techniques have been invented. These techniques are based on the use of low intensity electric fields, and are described in [76], and they are called *the human shunt* and *the human transmitter*. Later, a technique called *Projected Capacitive* was also discovered.

The most representative multitouch surface using these techniques is called Diamond Touch [10] and it is based on *the human transmitter* principle. It was developed by Mitsubishi Electric Research Laboratories (MERL). This multitouch surface has been used by several IMT [66, 49, 26, 41, 47, 58, 77].The most important feature of this surface is that it can identify each touch with the user which did it. On the contrary, users must wear or touch a device to create the electric field, reducing the naturalness of interaction. On the rest of the practical examples of these techniques there are still plenty of restrictions in recognition resolution, frequency and robustness.

A.3.0.1 The human shunt

A potential is created between an oscillator electrode and a virtual ground electrode. Then the intensity between the electrodes, i.e. the electric field, is measured. If the electrodes' size is tiny compared to the distance between them, then the electrodes can be modeled as punctual charges, producing dipole fields. The intensity of the dipole field inversely varies with the distance. When an object with a size much bigger than the size of the dipole approaches to the dipole field, the received intensity reduces, since that intensity is redirected to ground. As the object gets closer, the dipole field intensity reduces. Using this procedure multiple touches can be recognized simultaneously, but also objects in the proximity of the surface and the approximate distance between the object and the surface. This technique is broadly known as *Surface Capacitance*, and in figure A.11 the scheme of a multitouch surface capacitive panel is shown.

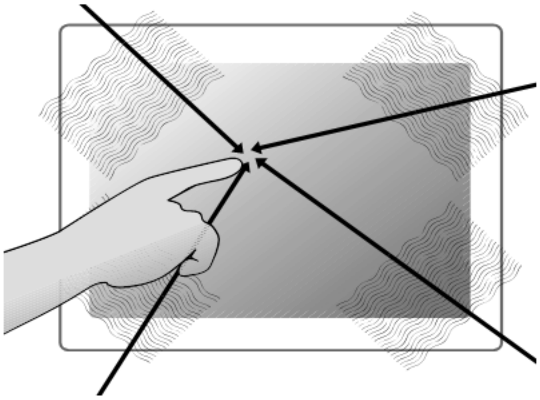


Figure A.11: **Surface Capacitance scheme** [9].

Advantages:

- High durability (a uniform conductive coating on a glass layer.).
- High clarity can be achieved by using indium tin oxide as the conductive material (it is transparent as well as colorless when used in very thin layers).
- Can sense objects in the proximity of the surface and the estimated distance to the surface.
- Not affected by environmental lighting.

Disadvantages:

- Only for conductive objects.
- Can be affected by electromagnetic interferences in the environment.
- Multitouch interpretation is not easy, and it is usually limited due to occlusions.

A.3.0.2 The human transmitter

Low frequency energy is capacitively coupled through a person's body, becoming him an electric field emitter. Several receptors on the tabletop can

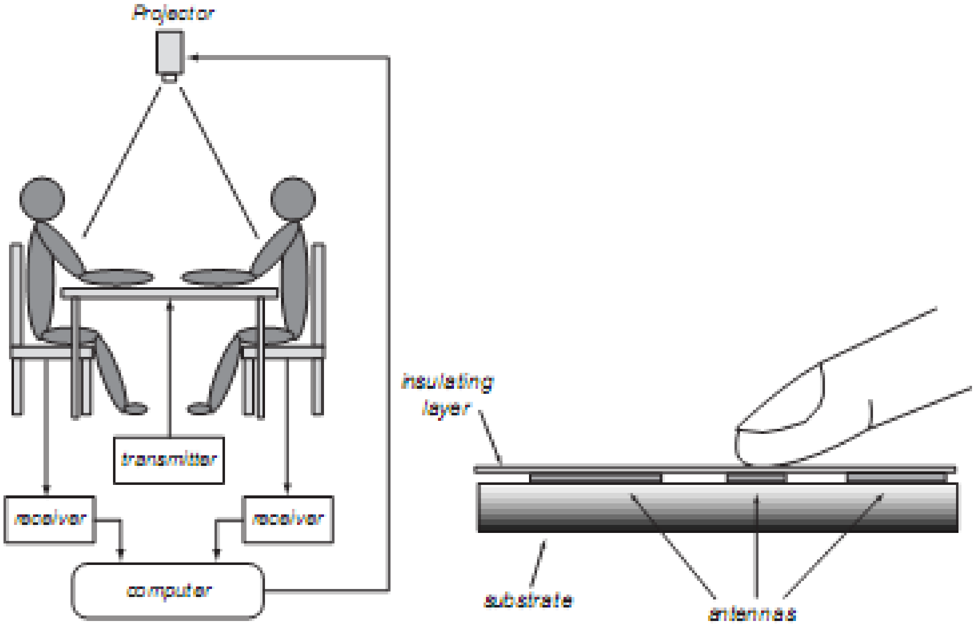


Figure A.12: **Diamond Touch scheme** [10]

perceive the intensity amount received. The greater this intensity the closer the receptor and the emitter are.

[9]Diamond Touch works by transmitting signals through antennas in the table; these signals are used to identify the parts of the table each user is touching. This information can then be used to calculate the finger's position. Usually a user touches several antennas at once. For this reason the signals have to be separable (in technical terms orthogonal). This can be achieved by frequency-division multiplexing, time-division multiplexing or code-division multiplexing. The antenna pattern consists of two layers similar in design, but with one rotated by ninety degrees. The rows/columns (antennas) of each layer are composed of diamond shapes connected in one direction and isolated in the other. In this way, the covered surface is maximized and the shielding effect minimized. Usually there is an antenna every five millimeters (which is in result the maximum pointing accuracy), illustrated in fig. A.12. Due to image projection from above the only obstructions are shadows cast on the table by objects (i.e. hands or arms) inserted into the projector's light beam.

Advantages:

- Can sense objects in the proximity of the surface and the approximated distance to the surface.
- Can identify users (antenna associated to each user).

Disadvantages:

- Only for conductive objects connected to the surface.
- Can be affected by electromagnetic interferences in the environment.
- Users must wear an additional device or at least touch it to close the circuit.

A.3.0.3 Projected Capacitive

This kind of devices are the most expensive capacitive surfaces to produce. Their performance is rather worse than many of the other approaches described, however they afford superb mechanical resilience. Projected capacitive surfaces can also be covered by a non-conductive material (with a maximum thickness of around 20mm) without negatively impacting on their functionality. When used for multitouch surfaces, a very thin grid of microphone wires is installed between two protective glass layers (see figure 6). When touched, capacitance forms between the finger and the sensor grid and the touch location can be computed based on the measured electrical characteristics of the grid layer. The accuracy of projected capacitive technology is similar to surface capacitive technology although light transmission is superior because the wire grid can be constructed such that it is nearly transparent. The technology is also highly suitable for rugged environments such as public installations, as a protective layer (such as thick glass) may be added with out drastically decreasing the sensitivity. Finally, multiple simultaneous touches can be more easily interpreted com pared to surface capacitive based technology.

Advantages:

- High clarity (very thin wires).

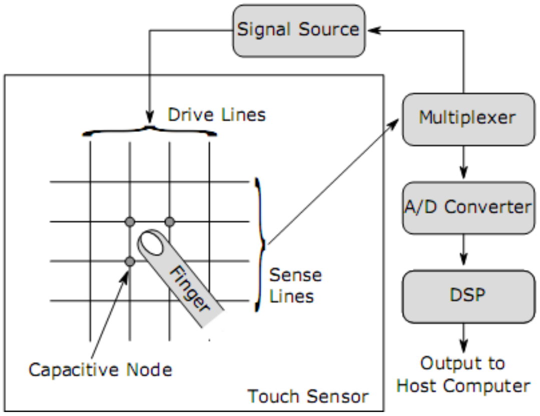


Figure A.13: **Projected Capacitive scheme** [9].

- Can sense objects in the proximity of the surface and the approximated distance to the surface.
- Can identify users (antenna associated to each user).
- High durability. Can be covered by a non-conductive material (with a maximum thickness of around 20mm) without negatively impacting on their functionality.
- Multitouch can easily be interpreted.

Disadvantages:

- Only for conductive objects connected to the surface.example
- Can be affected by electromagnetic interferences in the environment.
- Expensive.

A.4 Digital resistive

Generally two conductive layers are coated with substances such as indium tin oxide. These layers are separated by an insulating layer, usually made of tiny silicon dots. The touchable panel is typically made of a flexible hard coated outer membrane while the back panel is often a glass substrate, as illustrated

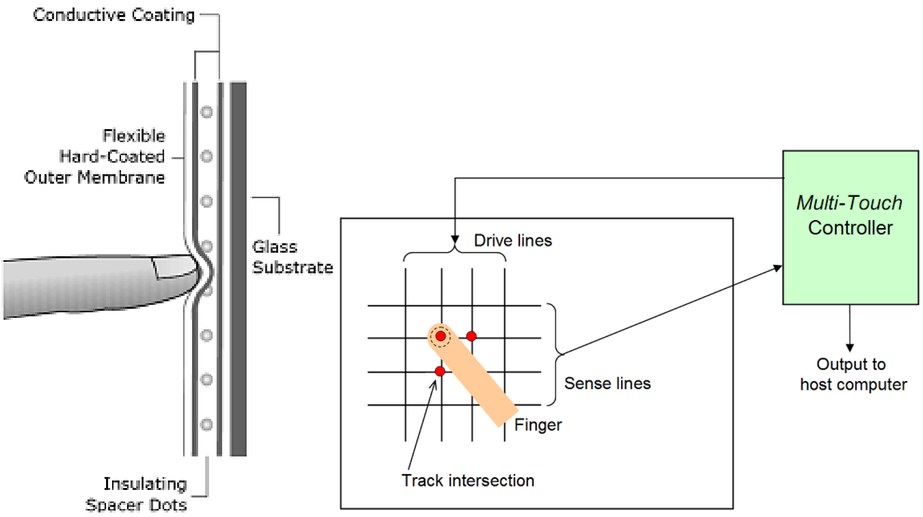


Figure A.14: Left: Resistive layering scheme [9]. Right: **Multitouch Resistive** surface scheme.

in fig. A.14. A controller swaps between giving electric current to one of the conductive layers and measuring it on the other. When users touch the display, the conductive layers are connected, establishing an electric current measured horizontally and vertically thanks to the controller swapping.

Advantages:

- Low power consumption.
- Can be used with fingers, stylus and any other object.

Disadvantages:

- Low transparency surfaces.
- Low durability of the flexible outer membrane.
- Usually low resolution.

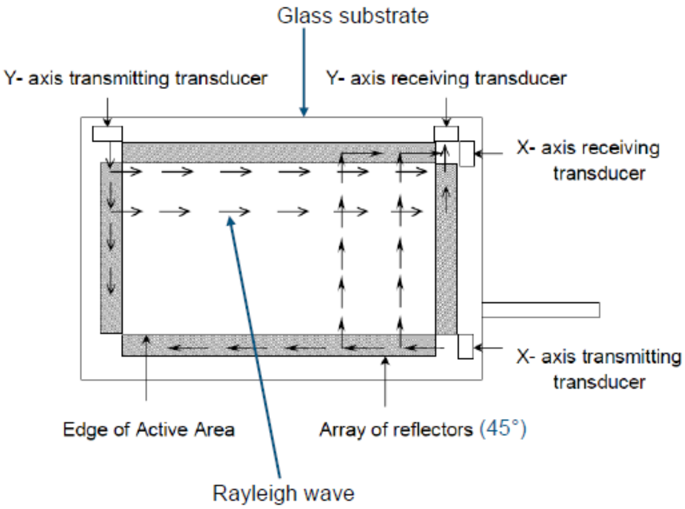

Glass substrate
Y- axis transmitting transducer
Y- axis receiving transducer
X- axis receiving transducer
X- axis transmitting transducer
Edge of Active Area
Array of reflectors (45°)
Rayleigh wave

Figure A.15: **Surface Acoustic Wave scheme**.

A.5 Surface Acoustic Wave (SAW)

Systems that use surface wave technology are similar to those that use infrared grid technology. Transmitting and receiving piezoelectric transducers, for both the X and Y axes, are mounted on a faceplate and ultra-sonic waves on a glass surface are created and directed by reflectors, represented in fig. A.15. By processing these to electronic signals and observing the changes when the faceplate is touched, it is possible to calculate the position of that interaction. Most SAW systems can support at most dual touch, therefore these kind of displays are not really multitouch, but are explained for completion.

Advantages:

- Very durable (can be ruggedized).
- High clarity surface.
- Fingers, stylus and objects can be used.

Disadvantages:

- Very sensitive to any surface contamination, including water.

- Requires sound absorbing object. Not any object.
- Projects slightly above the surface (~1-2 mm.), therefore touch events are recognized before even touching the surface.
- Size restrictions (less than 50").
- Limited number of simultaneous touches.

Appendix B

Support Knowledge

This appendix only serves as an introductory chapter to present several concepts which appear through the PhD Thesis Report. The distance Transform Function is presented in section B.1. The Discrete Curve Evolution shape boundary curve sampling procedure is presented in section B.2. The Voronoi Tessellation procedure is introduced in section B.3. Graph matching is introduced in section B.4. And finally Shock Graphs are presented in section B.5. For a deeper talk on each of the sections the reader is referred to the technical report of our research group web page hosted at [?].

B.1 Distance Transform function (DT)

A distance transform, also known as distance transformation, distance map or distance field, is a representation of a digital image. The choice of the term depends on the point of view on the object in question: whether the initial image is transformed into another representation, or it is simply endowed with an additional map or field. The map supplies each pixel of the image with the distance to the nearest obstacle pixel. A most common type obstacle pixel is a boundary pixel in a binary image. Usually the transform/map is qualified with the chosen metric. For example, one may speak of Manhattan distance transform, if the underlying metric is Manhattan distance. Common metrics are:

- Manhattan distance, City block distance or Taxicab geometry (d_1).
- Chessboard distance, Chebyshev distance or Tchebychev distance (d_∞).

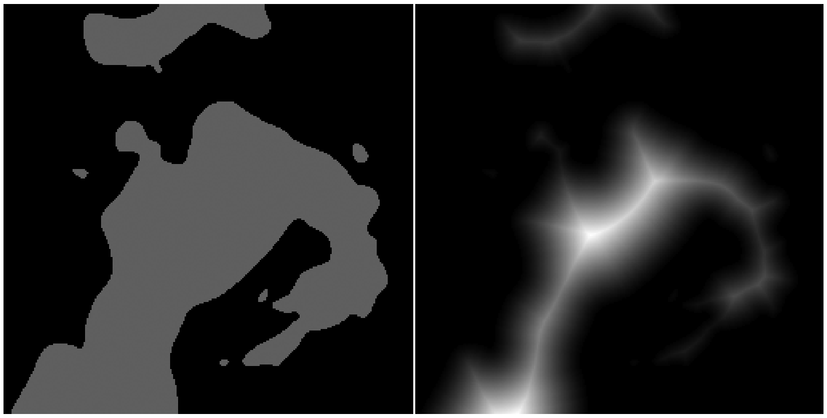


Figure B.1: **Graphical representation of the Distance Transform function**. The left image represents several connected components in gray, and the background in black. And the image on the right shows the distance transform value for each pixel in the left image. Brighter pixel color represents higher Distance Transform value.

- Chamfer distance, or weighted distance, with weights $(a, b)(d_{a,b})$.
- Euclidean distance.

Given a binary image, the distance transform function value of an image point $x \in \mathbb{R}^2$, denoted as $DT\left(x\right)$is the minimum distance of x to any image background point. Formally, let a binary image $I : \mathbb{R}^2 \rightarrow \{0, 1\}$where 0 corresponds to the background B and 1 to the foreground F (i.e., the shape), and $I = B \cup F$, and a distance measure $\left|.\right|_d$. Then the distance transform function is a function $DT\left(x\right)$ so that,

$$DT\left(x\right) = \min\left(\left|x - y\right|_d\right) \ s.t. \ y \in B \tag{B.1}$$

Initial Distance Transform computation algorithms made use of distance metrics alternative to the Euclidean in order to improve the efficiency, at the cost of less accuracy (according to [134] there is distance measurement error between 40 and 5 percentage with the Euclidean distance) and also orientation dependency. The most rotational invariant metric is the Euclidean. Fortunately, actual algorithms like like [141, 206, 142] can compute the Euclidean Distance Transform in $O\left(n\right)$. A graphical representation of this transform can be seen in fig. B.1.

B.2 Discrete Curve Evolution (DCE)

Discrete Curve Evolution [207], DCE henceforth, is a shape boundary curve downsampling method that keeps the most meaningful geometrical information for shape recognition [208, 209]. Boundary curves of digital binary images usually include some degree of noise due to the digitization and segmentation errors. And the set of boundary points usually includes redundant points, or at least many points which provide very low information for the shape description, i.e. they have low importance or salience. The main idea of the DCE procedure is to downsample iteratively the shape boundary curves' point set, removing the point with less contribution to the global shape at each step of the iteration, therefore reducing the boundary curve complexity while keeping the most important information of the shape.

Any digital curve can be described as a polygon, with the enough number of vertices, and the main idea of the DCE procedure consists of removing one edge of that polygon at each step of the procedure by replacing the two consecutive segments connected to the point with the global minimum relevance value v_{min} by one connecting the two points left after the removal of that v_{min}.

The salience of each vertex is given by the relevance value $K(S_i, S_{i+1})$, which computes the salience of the polygon vertex incident to the consecutive polygon segments S_i and S_{i+1}. It is defined as:

$$K(S_i, S_{i+1}) = \frac{\beta(S_i, S_{i+1}) l(S_i) l(S_{i+1})}{l(S_i) + l(S_{i+1})}, \tag{B.2}$$

where $\beta(S_i, S_{i+1})$ is the turn angle at the common vertex of segments S_i and S_{i+1} and l is the length function normalized with respect to the total length of a polygonal curve C. The value $K(S_i, S_{i+1})$ is directly proportional to the contribution of the arc $S_i \cup S_{i+1}$ to the shape. The cost function $K(S_i, S_{i+1})$ is monotonically increasing with respect to the relative lengths and the total curvature of segments S_i and S_{i+1}

[207]Let $D_m = s_0, ...s_{m-1}$be a decomposition of a digital curve C into consecutive digital line segments. The algorithm that computes the decompositions D_k for each stage of the discrete curve evolution $k > 3$ until $D_{k-1}$is convex is shown in figureB.1:

This algorithm is guaranteed to terminate because the number of initial segments is finite and one segment is removed at each step of the algorithm. It is also proved in [207] that the DCE procedure converges into a convex

Algorithm B.1 Discrete Curve Evolution Algorithm (D_m)

```
$k = m$;
**Do**
**Find** in ($D_k$) a pair $s_i$ , $s_{(i+1)mod(k)}$ such that $K(S_i, S_{i+1})$ is minimal
**{**
    $D_{k+1} = D_k$ with segments $s_i$,$s_{i+1}$ replaced by $s'$ joining the endpoints
        of $s_i U s_{i+1}$;
    $k = k - 1$;
**}until** $D_{k-1}$ is convex;
```

polygon. Although the original algorithm ends when the convexity of the DCE polygon is achieved, other ending criteria is possible, e.g. defining a constant number of iterations or final segment number. In [209] a more sophisticated criteria is employed based on a threshold value for the difference between the original shape and the DCE approximation.

Properties of the Discrete Curve Evolution procedure:

- It leads to the simplification of shape complexity, in analogy to evolutions guided by diffusion equations.

- There are no blurring (i.e., shape rounding) effects and no dislocation of relevant features, because vertices do not change their position.

- The relevance value $K(S_i, S_{i+1})$ is stable under deformations on the shape boundary curves, because noise elimination takes place in the early stages of the evolution.

- It allows to find line segments in geometrical objects even under noisy images, due to the relevance order of the repeated processes of digital linearization.

B.3 Voronoi Tessellation in $\mathbb{R}^2$

A Voronoi Tessellation, also called Voronoi diagram, Dirichlet diagrams or Thiessen polygons, is a partition of the space into convex regions called Voronoi polygons, each around a generator or Voronoi site, (belonging to a finite set of points in the space with at least two points), so that the pixels in the Voronoi polygon o an specific Voronoi site are closer to it than to any other Voronoi site. This partition is illustrated in figure B.2.

The Voronoi Tessellation can be computed in $O(n \log n)$[147], but an optimal lower bound in $O(n)$ is proved to be theoretically feasible in [148], with n being the number of $v_i \in V_{sites}$. For convex polygons the computational cost is $O(n)$, as proved by Aggarwal in [149]. Although this tessellation can be generalized to n-dimensional spaces, this section is focused in the bi-dimensional space. Next, the Voronoi Tessellation is mathematically defined, as well as its dual, the Delaunay triangulation.

B.3.1 Planar Ordinary Voronoi Tessellation/Diagram

Definition B.3.1. *Planar ordinary Voronoi diagram.* Given a set of two or more but a finite number of distinct points in the Euclidean plane, we associate all locations in that space with the closest member(s) of the point set with respect to the Euclidean distance. The result is a tessellation of the plane into a set of the regions associated with members of the point set. We call this tessellation the planar ordinary Voronoi diagram generated by the point set, and the regions constituting the Voronoi diagram ordinary Voronoi polygons. Mathematically,

Let the Voronoi Tessellation V consists of a decomposition of the image plane domain D into convex regions around a set of points, the Voronoi sites $V_{sites} = \{v_1, ..., v_n\}$, and called *Voronoi polygon* $V_{poly}(i)$. A Voronoi polygon $V_{poly}(i)$ around a Voronoi site v_i is defined as:

$$V_{poly}(i) = \left\{x, v_i, v_j \in \mathbb{R}^2 \mid \|x - v_i\| \leq \|x - v_j\| \, \forall v_j \neq v_i\right\} \tag{B.3}$$

Then, the Voronoi Tessellation is mathematically defined as,

$$V = \bigcup_{i=1}^{n} V_{poly}(i) = \{V_{poly}(1), ..., V_{poly}(n)\} \tag{B.4}$$

Each Voronoi polygon is bounded by several Voronoi segments and vertices.

Definition B.3.2. A *Voronoi segment* is the locus of the intersection of exactly two and no more different Voronoi polygons, and it is therefore determined only by two Voronoi sites. Consequently the Voronoi segment shared by the Voronoi sites v_i and v_j, denoted as s_{ij} is defined by the next equation:

$$s_{ij} = V_{poly}\left(i\right) \cap V_{poly}\left(j\right) \tag{B.5}$$

or alternatively,

$$s_{ij} = \left\{x \in \mathbb{R}^2 \mid \|x - v_i\| = \|x - v_j\| \wedge \forall k \neq i \neq j, \|x - v_k\| > \|x - v_i\|\right\} \tag{B.6}$$

Definition B.3.3. A *Voronoi vertex* is the intersection of three or more Voronoi polygons. The set of Voronoi vertices is defined as:

$$T = \left\{x \in \mathbb{R}^2 \middle| W \subset V_{sites} \wedge ||W| \geq 3| \; s.t. \; x \in \bigcap_{ij, \in W} s_{ij}\right\} \tag{B.7}$$

In fig. B.2 a graphical representation of a simple planar Voronoi Tessellation is shown. These definitions show that there must be at least two points in order to compute the Voronoi Tessellation.

B.3.2 Delaunay Triangulation

A Voronoi diagram has its ‘dual tessellation’, called a Delaunay triangulation, which is represented in fig. B.3. We consider a Voronoi diagram in the Euclidean plane, and assume that generator points of the Voronoi diagram are not on the same line, and also that there are at least three Voronoi sites, but they are finite $3 \leq |V_{sites}| < \infty$, otherwise the Delaunay triangulation is not possible.

Most of the Delaunay triangulation algorithms can perform in $O\left(n \cdot \log\left(n\right)\right)$ time, but in [210] a method is proposed to compute it in expected $O\left(n\right)$ time. And the conversion between the Voronoi Tessellation and the Delaunay triangulation can be performed in lineal time ($O\left(n\right)$).

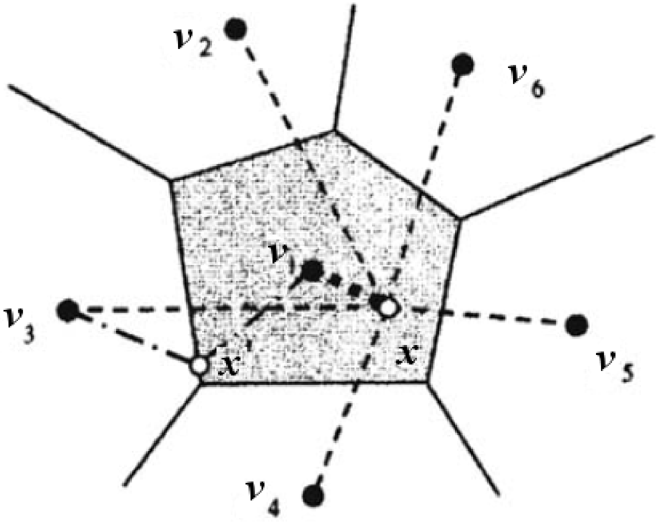


Figure B.2: **Graphical representation of a simple planar Voronoi Tessellation**. v correspond to Voronoi sites, while x correspond the ordinary points. $x\prime$ belongs to s_{13}.

B.4 Graph Matching

Most of the information contained in this section has been summarized from chapters 2 and 6 of the thesis work by Bengoetxea [211]. For more detailed explanations the user is referred to that work.

B.4.1 Introduction

Let $G\,(V, E)$ be a *graph* where V is the set of vertices or nodes and $E \rightarrow V \times V$ (also defined as $E[V]^2$ in the literature) is the set of edges (also known as arcs, links or lines). The order (or size) of a graph G is defined as the number of vertices of G and it is represented as $|V|$ and the number of edges as$|E|$, or also $|G|$ and $\|G\|$. If two vertices in G, say$u, v \in V$, are connected by an edge $e \in E$, this is denoted by $e = (u, v)$ and the two vertices are said to be *adjacent* or neighbors. Edges are said to be *undirected* when they have no direction, and a graph G containing only such types of graphs is called undirected. When all edges have directions and therefore (u, v) and (v, u) can be distinguished, the graph is said to be *directed.* Usually, the term *arc* is used when the graph is directed, and the term edge is used when it is

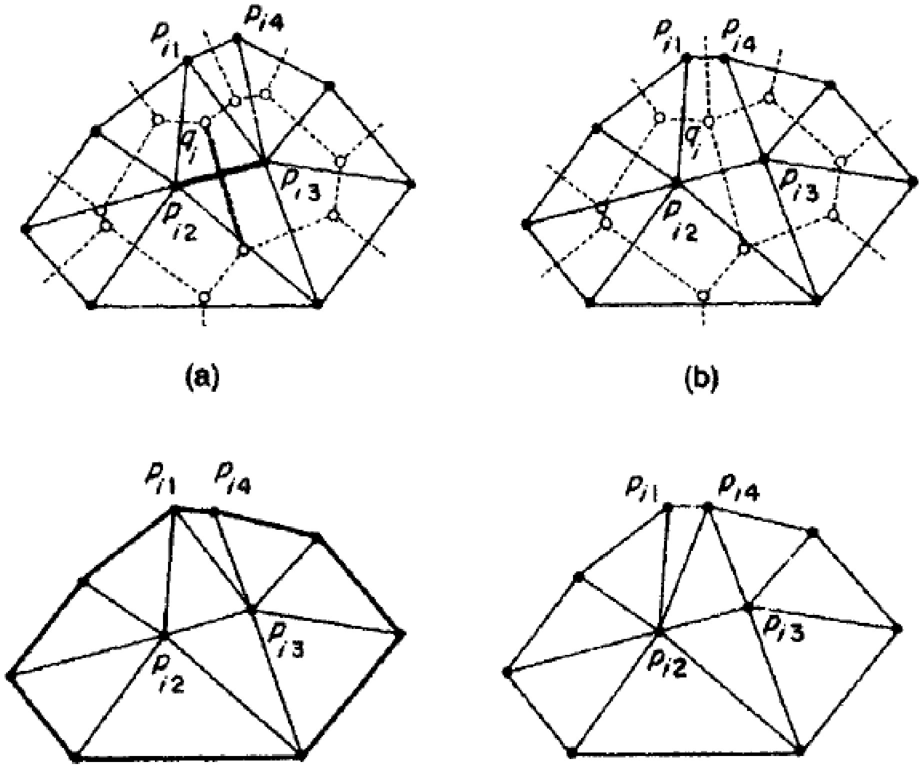


Figure B.3: **Delaunay triangulation example**. Filled circles correspond to Voronoi sites, empty circles correspond to Voronoi vertices, dotted lines correspond to the Voronoi polygons and filled lines correspond to the delaunay triangulation. Figure a shows a complete triangulation, while b is a pretriangulation and the figures below are two possible final Delaunay triangulations computed from b.

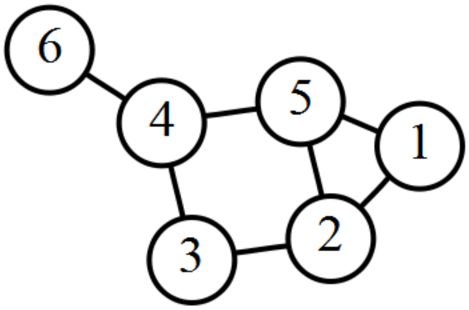


Figure B.4: **Example of non directed cyclic labeled graph**. The circles represent the nodes and the lines represent the links. The numbers in the nodes correspond to their labels.

undirected. In addition, a directed graph $G = (V, E)$ is called *complete* when there is always an edge$(u, u\prime) \in E = V \times V$ between any two *vertices* $u, u\prime$ in the graph. A graph $G\prime\,(V\prime, E\prime)$ is called a *subgraph* of $G\,(V, E)$ if $V\prime \subseteq V$ and $E\prime \subseteq E$.

Graph vertices and edges can also contain information. When this information is a simple label (i.e. a name or number) the graph is called *labeled graph*. Other times, vertices and edges contain some more information. When only nodes include attributes the graph is said to be *vertex-attributed*, or simply an *attributed graph* and when only the edges are attributed it is called *edge-attributed* or *weighted graph*. When both edge and vertex attributes are present, the graph is called an *attributed relational graph*.

A path between any two vertices $u, u\prime \in V$ is a non-empty sequence of k different vertices $< v_0, v_1, ..., v_k >$ where $u = v_0, u\prime = v_k$ and $(v_{i-1}, v_i) \in E, i = 1, 2, ..., k$. Finally, a graph G is said to be *acyclic* when there are no cycles between its edges, independently of whether the graph G is directed or not. An example of a graphical representation of a graph is shown in fig. B.4

B.4.2 Isomorphism and Homomorphism Graph Matching

Given two graphs $G_s = (V_s, E_s)$ and $G_{T_i} = (V_{T_i}, E_{T_i})$, with $|V_s| = |V_{T_i}|$, the graph matching problem is to find a one-to-one mapping $f : V_s \rightarrow V_{T_i}$ such that$(u, v) \in E_s \; iff \; (f\,(u)\,, f\,(v)) \in E_{T_i}$. When such a mapping f exists, this

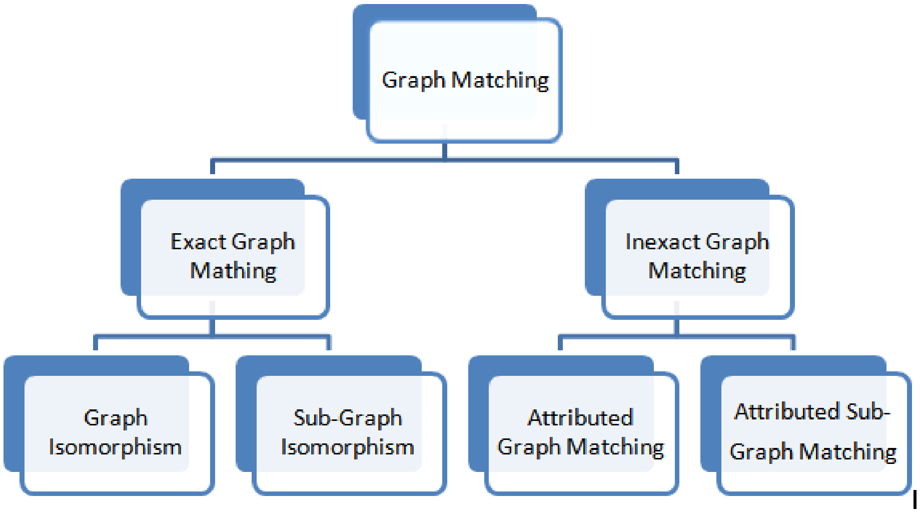


Figure B.5: Graph matching type classification

is called an *isomorphism*, and G_s is said to be isomorphic to G_{T_i}. This type of problems is said to be *exact graph matching*.

The term *inexact* applied to some graph matching problems means that it is not possible to find an isomorphism between the two graphs to be matched. This is the case when the number of vertices is different in both the model and data graphs. This partial matching problem leads to a class of problems known as *inexact graph matching*, or *homomorphism*. In that case, the matching procedure aims at finding a non-bijective correspondence between a data graph and a model graph. In an inexact graph matching problem the goal is to find a mapping $f\prime : V_s \rightarrow V_{T_i}$ such that $(u, v) \in E_s \; iff \; (f(u), f(v)) \in E_{T_i}$, assuming that $|V_{T_i}| < |V_s|$. This corresponds to the search for a small graph within a big one. An important sub-type of these problems are subgraph matching problems, in which we have two graphs $G = (V, E)$ and $G\prime = (V\prime, E\prime)$, where $V\prime \subseteq V$ and $E\prime \subseteq E$, and in this case the aim if to find a mapping $f\prime : V\prime \rightarrow V$ such that $(u, v) \in E\prime$ iff $(f(u), f(v)) \in E$. When such a mapping exists, this is called a *subgraph matching* or *subgraph isomorphism*.

The classification of the different graph matching problems is graphically shown in fig. B.5.

B.4.3 Graph Matching Techniques

Graph matching problems are known to be NP-complete, with an increasing complexity starting by exact isomorphism, following with subgraph isomorphism and finishing with attributed graph and sub-graph homomorphism matching which are the most complex. Consequently the efficiency of graph matching algorithms is one of the most important goals, including many heuristics, and even trying to transform the graph into a rooted tree when possible, because its lower matching complexity.

Graph Matching using dummy nodes is a special case of the exact graph matching, when $|G_1| \neq |G_2|$. In such cases artificial nodes can be added, so that $|G_1| = |G_2|$, and given values such that the similarity between nodes is very low, to penalize their use in the matching process. It is a one-to-one graph vertex matching, although it is also a homomorphic graph matching, just in the limit with the isomorphic graph matching procedure. It is less computationally complex than conventional many-to-many graph node matching procedures, i.e. graph homomorphic matching, but imposes much more restrictions to the graphs being matched.

Some other graph matching problems allow many-to-many matches, that is, given two graphs$G_{T_i} = (V_{T_i}, E_{T_i})$ and $G_s = (V_s, E_s)$, the problem consists on searching for a homomorphism $f : V_s \rightarrow W$ where $W \in P\left(V_{T_i}\right) \setminus \{\emptyset\}$ and $W \subseteq V_{T_i}$. In case of using also dummy vertices, W can take the value $\emptyset$ and therefore $W \in P(V_{T_i})$. This type of graph matching problems are more difficult to solve, as the complexity of the search for the best homomorphism has much more combinations and therefore the search space of the graph matching algorithm is much bigger.

B.4.3.1 Elastic matching

Elastic graph matching is a graph representation and matching technique that takes into account the possible deformation of objects to be recognized. It usually consists of two steps. First of all both graphs are matched with a rigid grid, and then the grid is deformed, permitting certain flexibility. In shape matching, the second step permits certain deformation, rotation and scale between the template and the sample. In a [212] several algorithms of this type are compared in a practical problem.

B.4.3.2 Morphological Graph Matching

Morphological graph matching applies hyperplanes or deformable spline-based models to the skeleton of non-rigid discrete objects. The graph is built from the most salient point set in the skeleton. Partial shape recognition is possible interactively defining shape features using a sub-graph matching algorithm.

B.4.3.3 Graph Edit Distance

The graph edit distance between two graphs is defined as the number of modifications that one has to undertake to arrive from one graph to be the other. The distance between two graphs is defined as the weighted sum of the costs of edit operations (insert, delete, and relabel the vertices and edges) to transform one graph to the other. The fact of applying these concepts and removing vertices or edges in graphs is analyzed in many works, as removal will lead to smaller graphs and therefore the graph matching problem can be reduced in complexity.

B.4.3.4 Error Correction Graph Matching

In error-correcting graph matching (or error-tolerant graph matching as it is also called) one considers a set of graph edit operations, and defines the edit distance of two graphs G_1 and G_2 as the shortest (or least cost) sequence of edit operations that transform G_1 into G_2 . Error-correcting graph matching is a powerful concept that has various applications in pattern recognition and machine vision, and its application is focused on distorted inputs. It constitutes a different approach very similar to other graph matching techniques. In [213] this topic is addressed and a new distance measure on graphs that does not require any particular edit operations is proposed. This measure is based on the maximal common subgraph of two graphs. A general formulation for error-correcting subgraph isomorphism algorithms is presented in [214] in terms of adjacency graphs, and [215] presents a study on the influence of the definition of fitness functions for error correcting graph matching.which reveals guidelines for defining fitness functions for optimization algorithms in error correcting graph matching. In addition, in [216] an algorithm for error-correcting subgraph isomorphism detection from a set of model graphs to an unknown input graph is introduced.

B.4.3.5 Other Approaches

Many other techniques have been applied to the graph matching problem. Evolutionary algorithms, and specifically genetic algorithms [217, 218, 219], and the use of techniques based on probability theory [220, 221, 222], including the application of probabilistic relaxation to graph matching [223] and the Expectation Maximization (EM) algorithm [224]. Decision trees have also been used for graph matching like in [225], Neural Networks [226, 227], clustering techniques [228], etc [229, 230, 231, 232, 233, 216].

B.5 Shock Graph

B.5.1 Introduction

A Shock graph is a shape representation method. It is based in the Medial Axis but includes additional information, producing a graph representation of the shape which keeps global and local information of the shape, but also the relationship between parts.

B.5.2 Shapes and shocks

Shocks [234] are entropy satisfying entities, and the locus of shock positions forms the Blum's Medial Axis. The categorization of shocks according to the local variation of the radius function along the medial axis produces a labeled Medial Axis which provides a much richer shape descriptor than an unlabeled skeleton. To illustrate the labeling, imagine traversing a path along the medial axis. At a 1-shock the radius varies monotonically, as is the case for a protrusion. At a 2-shock the radius function achieves a strict local minimum such that the medial axis is disconnected when the shock is removed, e.g., at a neck. At a 3-shock the radius function is constant along an interval, e.g., for a bend with parallel sides. Finally, at a 4-shock the radius function achieves a strict local maximum, as is the case when the evolving curve annihilates into a single point or a seed. A visual representation of each category can be seen in figure B.6.

Formally, this medial axis labeling based in shock categories is defined as follows. Let X be the open interior of a simple closed curve, and $Me\left(X\right)$its medial axis (the set of points reached simultaneously by two or more fire fronts). Let $B\left(x,\epsilon\right)$ be an open disk of radius ϵ centered at $x \in X$, and

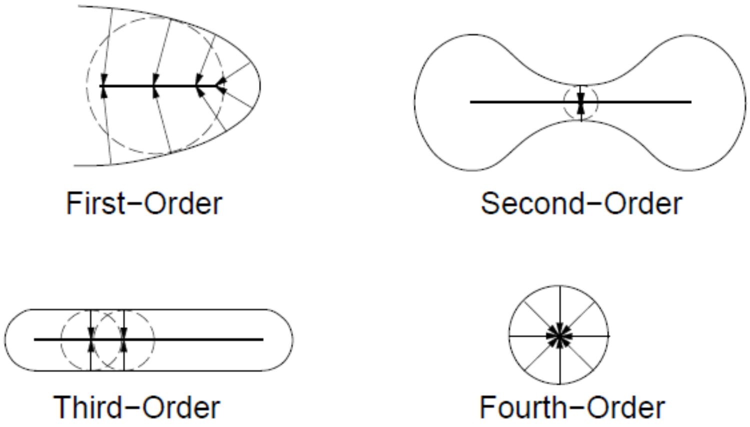


Figure B.6: **Shock categories**. First-order(1-shock): derives from a protrusion, and traces out a curve segment of first-order shocks. Second-order (2-shock): arises at a neck, and is immediately followed by two 1-shocks flowing away from it in opposite directions. Third-order (3-shock): correspond to an annihilation into curve segment due to a bend. Fourth-order (4-shock): an annihilation into a point or a seed. The loci of these shocks gives Blum's Medial Axis.

let $R(x)$denote the radius of the largest such disk contained in X. Let $N(x, \epsilon) = Me(X) \cap B(x, \epsilon) \setminus \{x\}$define a "punctured" ϵ-neighborhood of x, one that does not contain x itself. A medial axis point $x \in Me(X)$ is

1. Type 4 if $\exists \epsilon > 0 \; s.t. \; R(x) > R(y) \;\; \forall y \in N(x, \epsilon)$;
2. Type 3 if $\exists \epsilon > 0 \; s.t. \; R(x) = R(y) \;\; \forall y \in N(x, \epsilon)$ and $N(x, \epsilon) \neq 0$;
3. Type 2 if $\exists \epsilon > 0 \; s.t. \; R(x) < R(y) \;\; \forall y \in N(x, \epsilon)$ and $N(x, \epsilon) \neq 0$ and $N(x, \epsilon)$ is not connected; and
4. Type 1 otherwise.

It should be clear that there is a relationship between the above labeling and the velocity function $\frac{dR}{dx}$ along the Medial Axis [95].

B.5.3 The Shock Graph

It is an abstract representation of the shocks in a Medial Axis. The shock types will label each vertex in the graph and the shock formation times (i.e.

maximal disk radius or Distance Transform function value) will direct edges to provide an ordering for matching, and a basis for subgraph approximation.

By the shock labeling in the previous section it can be seen that 2-shocks and 4-shocks are isolated points, whereas 1-shocks and 3-shocks are neighbored by other shocks of the same type. To build the shock graph shocks of the same type that form a connected component shall be grouped together, denoting the groups with labels $\tilde{1}, 2, \tilde{3}$,4,[1] and breaking apart $\tilde{1}$'s at points with three or more generative points (i.e., the points where its maximal disk touches the shape boundary curve). Let each shock group be indexed by a distinct integer i and let t_i denote its time (or times) of formation (i.e., distance transform function value or minimum distance to a shape boundary curve point), corresponding to the radius function evaluated at the shocks in the group. Hence, t_i will be an interval for a $\tilde{1}$; for 2's, $\tilde{3}$'s (i.e., it is a set but all have the same value) and 4's it will be a single number. Finally, let # denote a start symbol and Φ a terminal symbol. The Shock Graph (SG) is a connected graph, rooted at a vertex labeled #, such that all other (non-terminal) vertices are shock groups, and directed edges to non-terminal vertices indicate the genesis of new shock groups.

Formally a Shock Graph of a 2-D shape, $SG(\mathcal{O})$, is a labeled graph $G = (V, E, \gamma)$, with:

- vertices $V = \{1, ..., n\}$;
- edges $(i, j) \in E \subseteq V \times V$ directed from vertex i to vertex j if and only if $i \neq j$, $t_i \geq t_j$, and $i \cup j$ is connected in the plane;
- labels $\gamma : V \rightarrow l$, with $l \in \left\{\tilde{1}, 2, \tilde{3}, 4, \#, \Phi\right\}$; and
- topology such that, $\forall j \in V$ with $\gamma(j) \neq \#, \exists i \in V$ with $(i, j) \in E$

The SG is built in descendant order of the $t_i$values, the distance transform values, because the shocks with higher values correspond to the most significant (central) features. The graph is rooted in the unique vertex labeled #, having as children the last shock groups formed during the grass-fire analogy, i.e., the shock group with the biggest maximal disk. And vertices with label Φ are leaves of the SG, whose parents are the first shock groups to form. Any 2D shape $\mathcal{O}$ has a unique corresponding shock graph $SG(\mathcal{O})$. This uniqueness is proved in [188].

[1] $\tilde{}$stands for a set of points, and the rest cases represent one single point.

B.5.4 The Shock Graph Grammar

The set of rules presented in the previous section have been grouped according to the semantic processes that they characterize, i.e., the birth, combination and death of shock groups, obtaining a small set of rules shown in figure B.7.

The Shock Graph Grammar, SGG, is a quadruple $G\left(V, \Sigma, R, S\right)$, with

1. $V = \left\{\tilde{1}, 2, \tilde{3}, 4, \#, \Phi\right\}$, the alphabet;
2. $\Sigma = \{\Phi\}$, the set of terminals;
3. $S = \#$, the start symbol; and
4. $R = \{R_1, ..., R_{10}\}$, the set of rules given in figure B.7.

The grammar in figure B.7 operates by beginning at the start symbol # and repeatedly replacing the left-hand side of a rule by the corresponding right-hand side until no further replacements can be made [235]. The rewrite rules of the Shock Graph Grammar are sufficient to derive the shock graph of any 2D shape $\mathcal{O}$.

Several consequences of these definition:

- Since the same shock cannot be born at two distinct times there exists no path from a vertex back to itself. Consequently, the Shock Graph is a directed acyclic graph (DAG). The problem of searching into acyclic graphs is computationally much simpler than in graphs with cycles on it.
- Since there exist rules in the SGG whose left-hand sides do not consist of single non-terminals, the SGG is not context-free.
- The rewrite rules indicate that a 2-shock and a 4-shock can only be added by rules 5 and 1 respectively, and that semantically equivalent rules exist for a $\tilde{3}$(rules 6 and 1). Hence, a 2-shock and a 4-shock are each semantically equivalent to a $\tilde{3}$ in a specific context. Following this observation, only label types $\tilde{1}$and $\tilde{3}$ have been explicitly assigned. A $\tilde{3}$ with a parent $\tilde{1}$ at each end acts as a 2 (a neck), and a $\tilde{3}$ with a # as a parent acts as a 4 (seed).

Later in [11] a modification of this grammar is proposed in order to include joint points (points where three or more skeleton branches intersect), since

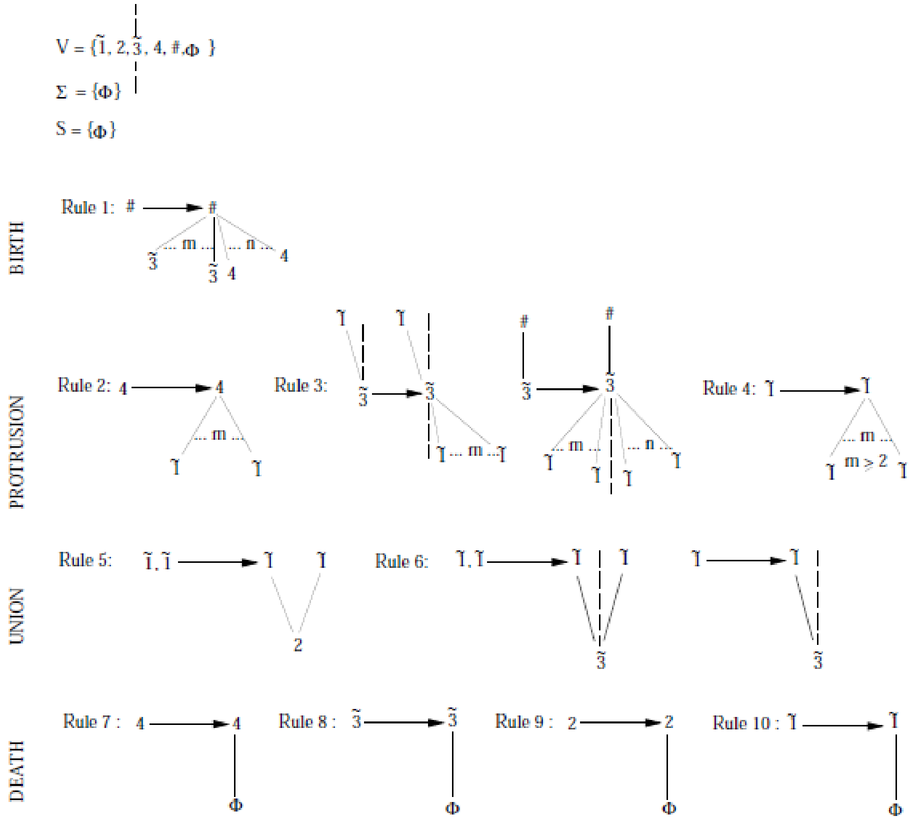


Figure B.7: **The Shock Graph Grammar, SGG**. Dashed lines partition distinct ends of a $\tilde{3}$. The rules are grouped according to the different semantic processes (on the left) that they characterize. Note that the grammar is not context-free, e.g., rule 3 indicates that a $\tilde{1}$ can only be added onto an end of a $\tilde{3}$ that has no parent $\tilde{1}$.

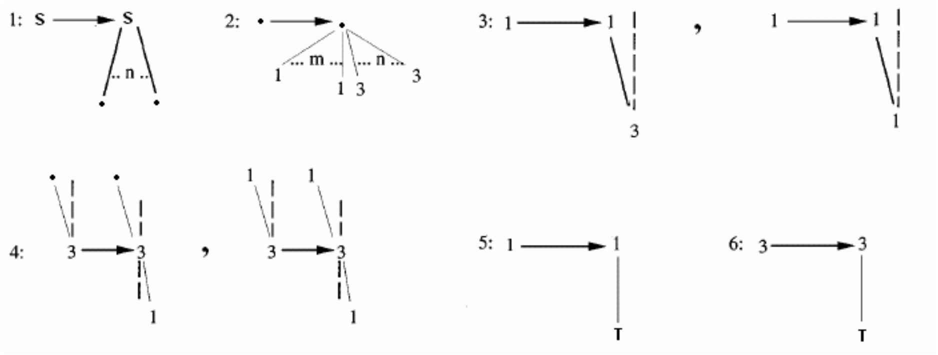


Figure B.8: Shock Grammar modification by [11]

they are usually located in the internal part of the skeleton. Consequently they are usually more stable under noise on the boundary curves, so their representation is proved to be more stable under noise than the original definition. Because the points of type 2 and 4 are always isolated points and not point sets, they ignore them in their grammar. Their grammar is $V(1, 3, \cdot, S, T)$, where "1" and "3" correspond to 1-shock and 3-shock groups, "$\cdot$" refers to a joint point, and symbols "S" and "T" correspond to the root and leave nodes respectively. Set $\sum = \{T\}$ contains terminal of the grammar and set R contains rules of the new grammar given in fig. B.8. Set $S = \{s\}$ contains start symbol of the grammar.

Each joint point is assigned to a joint node so that all of the end nodes are children of the root node, and the root node has only children of type joint node. Then, for each end point "a" on the skeleton trace skeletal curves branching out from it with identical speed and place shock groups as children of the end node "a". Direction of tracing always is from end points to the terminal points or other end points. Tracing is stopped at terminal points or when two tracer agents started from two distinct end points, reach to a common shock group. In the last case, common shock group is placed as the child of the two end nodes twice. An example of this kind of shock graph is shown in fig. B.9.

B.5.5 Shock Graph Matching

Given two shock graphs, one representing an object in the scene (V_2) and one representing class template object (V_1), we seek a method for computing

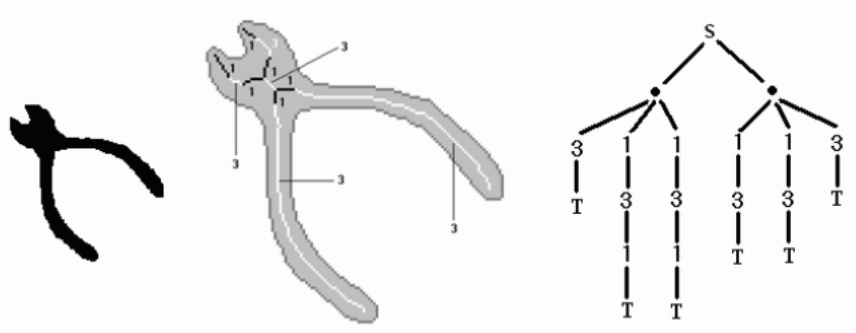


Figure B.9: Shock graph example using the modified Shock Grammar in [11]

their similarity. Unfortunately, due to occlusion and clutter, the shock graph representing the scene object may, in fact, be embedded in a larger shock graph representing the entire scene. Therefore, we have a largest subgraph isomorphism problem, stated as follows: Given two graphs $G = (V_1, E_1)$ and $H = (V_2, E_2)$, find the maximum integer k, such that there exists two subsets of cardinality k, $E_1' \subseteq E_1$ and $E_2' \subseteq E_2$, and the induced subgraphs $G' = (V_1, E_1')$ and $H' = (V_2, E_2')$, are isomorphic. Moreover, since shock graphs are labeled graphs, consistency between node labels must be enforced in the isomorphism. Graph matching techniques are presented in Chapter 3 in the section B.4about shape recognition and graphs obtained from the skeleton.

This graph matching problem is know to be NP-hard for general graphs [236], however, polynomial time algorithms exist for the special case of finite rooted trees [237, 238, 239]. In the next section a method is presented to obtain a unique rooted tree from a shock graph, to improve the efficiency of the matching procedure while keeping consistency. In addition, a depth-first search is performed on the underlying shock trees, to perform a matching beginning with the most meaningful parts of the shape towards its details.

B.5.6 Shock Graphs to Shock Trees

This section presents a method to convert a DAG representing a shock graph into a unique vertex labeled rooted tree whose size is polynomially bounded by the size of the original shock graph.

Let $G = (V, E)$ be a DAG representing a shock graph on n vertices. A loop L is a subgraph of G formed by the intersection of two directed paths.

more formally, L originates at a vertex b, follows two paths P_1 and P_2, and ends at the vertex t. We denote b as the base of L, t as the tip of L, and P_1 and P_2 the wings of L. Due to the shock graph grammar rules the authors of [188] conclude that the tips of all loops are adjacent to nodes having type Φ in G, and each such tip participates in exactly one loop.

The reduction can be obtained therefore as follows. For each tip node t duplicate copies t_1 and t_2 are maintained, and L is redefined to be the union of b and two new disjoint paths $P_1' = P_1 \cup \{t_1\} \cup \{\Phi\}$and $P_2' = P_2 \cup \{t_2\} \cup \{\Phi\}$. This reduction is unique and produces a directed, or equivalently, a rooted tree. This reduction can be computed in linear time, since G has only $O(n)$ tips, and as it is derived from the SGG, consists of checking the in-degree of any $\tilde{3}$'s and 2's, and duplicate them if necessary.

B.5.7 The Distance Between Two Vertices

Since both shock graphs and trees are labeled, part of the matching procedure includes node label matching, which corresponds to the geometrical properties of shape parts. In this particular case nodes are shock sequences, which are compared one to one. Each shock in labeled by its position, its time of formation (i.e., maximal disk radius), and its direction of flow (or orientation in the case of $\tilde{3}$'s) using the shock detection algorithm in [240].

The main idea is to interpolate a low dimensional curve through their respective shock trajectories, and assign a cost $C(u, v)$ to an affine transformation that aligns one interpolated curve with the other.

Assume that S and S' are two (sampled) shock sequences of the form $S = (s_1, ..., s_p)$ and $S' = \left(s_1', ..., s_p'\right)$, where each shock point $s_i$is represented by a 4-tuple (x, y, t, α) , corresponding to its Euclidean coordinates (x, y), formation time t, and direction α. For samples from a $\tilde{1}$, the sequence is ordered by time of formation, while for a $\tilde{3}$ there is a partial order to the samples, but no preferred direction. In the latter case, both directions will have to be tried. In order to find the 4D-simplex corresponding to the basis for the affine transformation (in a 4D space) between the two sets, they choose three equidistant points on the chains formed by partial orders $(s_1 \prec ... \prec s_p)$ and $\left(s_1' \prec ... \prec s_q'\right)$. To preserve the partial order of the points in each sequence, $s_1$should be transferred to s_1', and s_p to s_q'.

Let (A, B) be the transformation pair for this partial order and, without loss of generality, assume that $p \leq q$. They apply the transformation (A, B) to sequence S to form the sequence $\hat{S} = (\hat{s}_1, ..., \hat{s}_p)$. Once the curves

$\Psi\left(\widehat{S}\right)$and $\Psi\left(S'\right)$, which denote the interpolated 4D curves passing through the points of the sets $\hat{S}$ and S', are defined, the Hausdorff distance measure is computed between them,

$$\Delta\left(\Psi\left(\widehat{S}\right),\Psi\left(S'\right)\right)=\sum_{x\in\widehat{S}}\inf_{y\in\Psi(S')}\|x-y\|_2+\sum_{x\in S'}\inf_{y\in\Psi\left(\widehat{S}\right)}\|x-y\|_2 \qquad \text{(B.8)}$$

B.5.8 Algorithms for Shock Tree Matching

In the original paper describing shape recognition based in Shock Graphs and Medial Axis [188] a depth-first algorithm is used to obtain the maximum subgraph correspondence between two Shock Graphs in their tree form. Depth-first is an algorithm for traversing or searching a tree, tree structure, or graph. One starts at the root (selecting some node as the root in the graph case) and explores as far as possible along each branch before backtracking. See fig. B.10 for a graphical representation of the order followed by this procedure. It is therefore an exhaustive exploration paradigm.

Later, Sebastian et al. [190] proposed the use of the edit distance, which is a way to measure the minimum deformations needed to transform each of the shock graphs into the other. This creates a high dimensional search space. They define a shape cell as a collection of shapes which have identical shock graph topology, and a shape deformation bundle is the set of one-parameter families of deformations passing through an identical sequence of shock transitions. Using this two definitions the search space is partitioned and sampled into cells and the transitions between them are limited to the most simple, the one which includes less cells. In order to obtain this sequence, intermediate shock graphs between both graphs are computed.

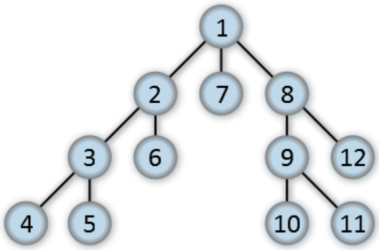


Figure B.10: Example of tree traversing order by depth-first procedure. Image courtesy of the Wikipedia

Bibliography

[1] S. Biasotti, D. Attali, J. Boissonnat, H. Edelsbrunner, E. G., M. Mortara, G. Sanniti di Baja, S. M., M. Tanase, and V. R., *Shape Analysis and Structuring.* Springer, 2008, ch. Skeletal Structures, p. 149. (document), 3.4

[2] ——, *Shape Analysis and Structuring.* Springer, 2008, ch. Skeletal Structures, p. 162. (document), 3.5

[3] C. Gold and J. Snoeyink, "A one-step crust and skeleton extraction algorithm," *Algorithmica*, vol. 30, no. 2, pp. 144–163, 2001. (document), 3.10

[4] W. H. Hesselink and J. B. Roerdink, "Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 12, pp. 2204–2217, 2008. (document), 3.1, 3.1, 3.2.2, 3.11, 2

[5] X. Bai. (2007) Matlab software for skeleton pruning. [Online]. Available: http://www.cis.temple.edu/~latecki/Programs/skeletonPruning07.htm (document), 4.1, 4.3, 5.1

[6] X. Bai, L. J. Latecki, and W. Y. Liu, "Skeleton pruning by contour partitioning with discrete curve evolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 449–462, 2007. [Online]. Available: http://dx.doi.org/http://dx.doi.org/10.1109/TPAMI.2007.59 (document), 3.3.1, 1, 3, 4.1, 4b, 4.1, 4.3, 4.4, 5, 5.1, 5.1, 5.2, 5.5.1, 5.3, 5.4, 5.5.1, 5.5, 5.6, 5.6

[7] A. Beristain and M. Graña. (2008, november) Synthetic hand gesture images for tabletop interaction. [Online].

Available: http://www.ehu.es/ccwintco/index.php/Synthetic_hand_gesture_images_for_tabletop_interaction (document), 4.3, 5.2.1, 5.1

[8] J. Rekimoto, "Perceptual surfaces: Towards a human and object sensitive interactive display," (Sony), 1997. (document), A.6

[9] J. Schöning, P. Brandl, F. Daiber, F. Echtler, O. Hilliges, J. Hook, M. Löchtefeld, N. Motamedi, L. Muller, P. Olivier, T. Roth, and U. von Zadow, "Multi-touch surfaces: A technical guide," Tech. Rep., October 2008. (document), A.1, A.2, A.11, A.3.0.2, A.13, A.14

[10] P. Dietz and D. Leigh, "Diamondtouch: a multi-user touch technology," in *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology.* New York, NY, USA: ACM, 2001, pp. 219–226. (document), 2, A.3, A.12

[11] H. Zaboli and M. Rahmati, "An improved shock graph approach for shape recognition and retrieval," in *AMS '07: Proceedings of the First Asia International Conference on Modelling & Simulation.* Washington, DC, USA: IEEE Computer Society, 2007, pp. 438–443. (document), B.5.4, B.8, B.9

[12] A. Beristain, P. Ayala, M. Pajares, and M. Grana, "Innovae emotional trainer: an interactive system to measure people's acting skills," in *Virtual Environments, Human-Computer Interfaces and Measurement Systems, Proceedings of 2006 IEEE International Conference on*, July 2006, pp. 172–177. 1.1

[13] J. Lopez, I. Cearreta, N. Garay, A. Beristain, and L. de Ipina Z., "Creación de una base de datos emocional bilingüe y multimodal," in *Interacción*, 2006. 1.1

[14] A. Beristain and M. Grana, "Emotion recognition based on the analysis of facial expressions," *New Mathematics and Natural Computation (NMNC)*, vol. 5, no. 2, pp. 513–534, 2009. 1.1

[15] M. Weiser, "The computer for the 21st century," *Scientific American*, 1991. 2.1

[16] ——, "Some computer science issues in ubiquitous computing," *Commun. ACM*, vol. 36, no. 7, pp. 75–84, 1993. 2.1

[17] ——, “The computer for the 21st century,” pp. 933–940, 1995. 2.1

[18] ——, “The computer for the 21st century,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 3, no. 3, pp. 3–11, 1999. 2.1

[19] R. Datta, D. Joshi, J. Li, and J. Z. Wang, “Image retrieval: Ideas, influences, and trends of the new age,” *ACM Comput. Surv.*, vol. 40, no. 2, pp. 1–60, 2008. 2.1

[20] S. McDonald and J. Tait, “Search strategies in content-based image retrieval,” in *SIGIR ’03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval.* New York, NY, USA: ACM, 2003, pp. 80–87. 2.1

[21] R. Zhang and Z. Zhang, “Addressing cbir efficiency, effectiveness, and retrieval subjectivity simultaneously,” in *MIR ’03: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval.* New York, NY, USA: ACM, 2003, pp. 71–78. 2.1

[22] S. Chatty, A. Lemort, and S. Vales, “Multiple input support in a model-based interaction framework,” Oct. 2007, pp. 179–186. 2.1.2

[23] S. Scott, K. D. Grant, and R. L. Mandryk, “System guidelines for co-located, collaborative work on a tabletop display,” in *European Conference Computer-Supported Cooperative Work (ECSCW 2003)*, 2003. [Online]. Available: http://ilpubs.stanford.edu:8090/612/ 2.1.2, 2.3

[24] P. Tuddenham and P. Robinson, “Distributed tabletops: Supporting remote and mixed-presence tabletop collaboration,” Oct. 2007, pp. 19–26. 2.1.2, 2.2.2, 2.3

[25] L. Liu, H. Erdogmus, and F. Maurer, “An environment for collaborative iteration planning,” July 2005, pp. 80–89. 2.1.2

[26] A. Pauchet, F. Coldefy, L. Lefebvre, S. Louis Dit Picard, L. Perron, A. Bouguet, M. Collobert, J. Guerin, and D. Corvaisier, “Tabletops: Worthwhile experiences of collocated and remote collaboration,” Oct. 2007, pp. 27–34. 2.1.2, 3, 2, A.3

[27] G. Morrison, "Interactive wall displays: interaction techniques and commercial applications," in *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*. New York, NY, USA: ACM, 2007, pp. 54–64. 2.1.2

[28] F. Guimbretière, M. Stone, and T. Winograd, "Fluid interaction with high-resolution wall-size displays," in *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*. New York, NY, USA: ACM, 2001, pp. 21–30. 2.1.2

[29] M. W. Krueger, T. Gionfriddo, and K. Hinrichsen, "Videoplace: An artificial reality," *SIGCHI Bull.*, vol. 16, no. 4, pp. 35–40, 1985. 2.2.1, A.1

[30] P. Wellner, "Interacting with paper on the digitaldesk," *Commun. ACM*, vol. 36, no. 7, pp. 87–96, 1993. 2.2.1, 1

[31] J. C. Tang and S. L. Minnenan, "Videodraw: A video interface for collaborative drawing," in *Proc. of CHI-90*, Seattle, WA, 1990, pp. 313–320. 2.2.1

[32] J. C. Tang and S. L. Minneman, "Videodraw: a video interface for collaborative drawing," *ACM Trans. Inf. Syst.*, vol. 9, no. 2, pp. 170–184, 1991. 2.2.1

[33] H. Ishii and N. Miyake, "Toward an open shared workspace: computer and video fusion approach of teamworkstation," *Commun. ACM*, vol. 34, no. 12, pp. 37–50, 1991. 2.2.1

[34] H. Ishii, M. Kobayashi, and J. Grudin, "Integration of interpersonal space and shared workspace: Clearboard design and experiments," *ACM Trans. Inf. Syst.*, vol. 11, no. 4, pp. 349–375, 1993. 2.2.1

[35] N. A. Streitz, J. Geisler, J. M. Haake, and J. Hol, "DOLPHIN: Integrated meeting support across local and remote desktop environments and liveboards," in *Computer Supported Cooperative Work*, 1994, pp. 345–358. [Online]. Available: citeseer.ist.psu.edu/streitz94dolphin.html 2.2.1

[36] S. Elrod, R. Bruce, R. Gold, D. Goldberg, F. Halasz, W. Janssen, D. Lee, K. McCall, E. Pedersen, K. Pier, J. Tang, and B. Welch,

"Liveboard: a large interactive display supporting group meetings, presentations, and remote collaboration," in *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems.* New York, NY, USA: ACM, 1992, pp. 599–607. 2.2.1

[37] G. W. Fitzmaurice, H. Ishii, and W. A. S. Buxton, "Bricks: laying the foundations for graspable user interfaces," in *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems.* New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 442–449. [Online]. Available: citeseer.ist.psu.edu/fitzmaurice95bricks.html 2.2.1

[38] J. Patten and H. Ishii, "Mechanical constraints as computational constraints in tabletop tangible interfaces," in *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems.* New York, NY, USA: ACM, 2007, pp. 809–818. 2.2.2, 2.7.3

[39] G. Pangaro, D. Maynes-aminzade, and H. Ishii, "The actuated workbench: computer-controlled actuation in tabletop tangible interfaces," in *Interfaces, Proceedings of Symposium on User Interface Software and Technology*, 2002, pp. 181–190. 2.2.2, 2

[40] M. Kim and M. Maher, "The impact of tangible user interfaces on spatial cognition during collaborative design," *Design Studies*, vol. 29, no. 3, pp. 222–253, May 2008. [Online]. Available: http://dx.doi.org/10.1016/j.destud.2007.12.006 2.2.2, 2.5.2

[41] D. Wigdor, G. Perm, K. Ryall, A. Esenther, and C. Shen, "Living with a tabletop: Analysis and observations of long term office use of a multi-touch table," Oct. 2007, pp. 60–67. 2.2.2, 1, 2, A.3

[42] U. Hinrichs, M. Hancock, C. Collins, and S. Carpendale, "Examination of text-entry methods for tabletop displays," Oct. 2007, pp. 105–112. 2.2.2, 2.5.1

[43] F. Chen, P. Eades, J. Epps, S. Lichman, B. Close, P. Hutterer, M. Takatsuka, B. Thomas, and M. Wu, "Vicat: Visualisation and interaction on a collaborative access table," *Horizontal Interactive Human-Computer Systems, International Workshop on*, vol. 0, pp. 59–62, 2006. 2.2.2, 3

[44] A. D. Wilson, “Depth-sensing video cameras for 3d tangible tabletop interaction,” in *Horizontal Interactive Human-Computer Systems, 2007. TABLETOP '07. Second Annual IEEE International Workshop on*, 2007, pp. 201–204. [Online]. Available: http://dx.doi.org/10.1109/TABLETOP.2007.13 2.2.3

[45] C. Lee, S. DiVerdi, and T. Höllerer, “An immaterial depth-fused 3d display,” in *VRST '07: Proceedings of the 2007 ACM symposium on Virtual reality software and technology.* New York, NY, USA: ACM, 2007, pp. 191–198. 2.2.3

[46] A. Tang, M. Tory, B. Po, P. Neumann, and S. Carpendale, “Collaborative coupling over tabletop displays,” in *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems.* New York, NY, USA: ACM, 2006, pp. 1181–1190. 2.3

[47] M. Ringel, K. Ryall, C. Shen, C. Forlines, and F. Vernier, “Release, relocate, reorient, resize: fluid techniques for document sharing on multi-user interactive tables,” in *CHI '04: CHI '04 extended abstracts on Human factors in computing systems.* New York, NY, USA: ACM, 2004, pp. 1441–1444. 2.3, 2.5.1, 2, A.3

[48] S. D. Scott, M. Sheelagh, T. Carpendale, and K. M. Inkpen, “Territoriality in collaborative tabletop workspaces,” in *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work.* New York, NY, USA: ACM, 2004, pp. 294–303. 2.3

[49] V. Ha, K. Inkpen, R. Mandryk, and T. Whalen, “Direct intentions: the effects of input devices on collaboration around a tabletop display,” Jan. 2006, pp. 8 pp.–. 2.3, 4, 2, A.3

[50] Y. Kakehi, M. Iida, T. Naemura, Y. Shirai, M. Matsushita, and T. Ohguro, “Lumisight table: an interactive view-dependent tabletop display,” *Computer Graphics and Applications, IEEE*, vol. 25, no. 1, pp. 48–53, Jan.-Feb. 2005. 2.3

[51] J. J. J. LaViola, “A survey of hand posture and gesture recognition techniques and technology,” Brown University, Providence, RI, USA, Tech. Rep. CS-99-11, 1999. [Online]. Available: citeseer.ist.psu.edu/laviola99survey.html 1b

[52] C. von Hardenberg and F. Bérard, "Bare-hand human-computer interaction," in *PUI '01: Proceedings of the 2001 workshop on Perceptive user interfaces.* New York, NY, USA: ACM, 2001, pp. 1–8. 1b

[53] S. Mitra and T. Acharya, "Gesture recognition: A survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 3, pp. 311–324, May 2007. 1b

[54] R. Berry, M. Makino, N. Hikawa, and M. Suzuki, "The augmented composer project: The music table," in *ISMAR '03: Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality.* Washington, DC, USA: IEEE Computer Society, 2003, p. 338. 2

[55] B. Ullmer and H. Ishii, "The metadesk: models and prototypes for tangible user interfaces," in *UIST '97: Proceedings of the 10th annual ACM symposium on User interface software and technology.* New York, NY, USA: ACM, 1997, pp. 223–232. 2, 5c

[56] A. Mazalek, M. Reynolds, and G. Davenport, "Tviews: An extensible architecture for multiuser digital media tables," *Computer Graphics and Applications, IEEE*, vol. 26, no. 5, pp. 47–55, Sept.-Oct. 2006. 2

[57] H. Ishii, "Tangible bits: beyond pixels," in *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction.* New York, NY, USA: ACM, 2008, pp. xv–xxv. 2, 2.5.2

[58] E. Tse, S. Greenberg, C. Shen, J. Barnwell, S. Shipman, and D. Leigh, "Multimodal split view tabletop interaction over existing applications," Oct. 2007, pp. 129–136. 3, 2, A.3

[59] A. Stafford, W. Piekarski, and B. Thomas, "Hog on a wim," March 2008, pp. 289–290. 3

[60] D. Jurafsky and J. H. Martin, *Speech and Language Processing (2nd Edition) (Prentice Hall Series in Artificial Intelligence)*, 2nd ed. Prentice Hall, May 2008. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20\&amp;path=ASIN/0131873210 3

[61] P. Nguyen, "Techware: speech recognition software and resources on the web," *Signal Processing Magazine, IEEE*, vol. 26, no. 3, pp. 102–105, May 2009. 3

[62] M. Ghazali and A. Dix, "Knowledge of today for the design of tomorrow," in *Proceedings of the 2nd International Design and Engagibility Conference (IDEC)*, Edinburgh, September 2005. 4, 2.7.1

[63] K. Ryall, A. Esenther, C. Forlines, C. Shen, S. Shipman, M. Morris, K. Everitt, and F. Vernier, "Identity-differentiating widgets for multiuser interactive surfaces," *Computer Graphics and Applications, IEEE*, vol. 26, no. 5, pp. 56–64, Sept.-Oct. 2006. 5c, 2

[64] K. Oka, Y. Sato, and H. Koike, "Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems," in *FGR '02: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition.* Washington, DC, USA: IEEE Computer Society, 2002, p. 429. 1

[65] S. Scott, M. Carpendale, and S. Habelski, "Storage bins: mobile storage for collaborative tabletop displays," *Computer Graphics and Applications, IEEE*, vol. 25, no. 4, pp. 58–65, July-Aug. 2005. 2, 2.5.1

[66] K. Ryall, C. Forlines, C. Shen, and M. R. Morris, "Exploring the effects of group size and table size on interactions with tabletop shared-display groupware," in *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work.* New York, NY, USA: ACM, 2004, pp. 284–293. 2.5.1, 2, A.3

[67] A. Toney and B. H. Thomas, "Considering reach in tangible and table top design," *Horizontal Interactive Human-Computer Systems, International Workshop on*, vol. 0, pp. 57–58, 2006. 2.5.1

[68] N. Streitz, P. Tandler, C. Müller-Tomfelde, and S. Konomi, *Human-Computer Interaction in the New Millennium.* Addison-Wesley, 2001, ch. Roomware: Towards the Next Generation of Human-Computer Interaction based on an Integrated Design of Real and Virtual Worlds, pp. 553–578. 2.5.1

[69] M. S. Hancock, S. Carpendale, F. D. Vernier, D. Wigdor, and C. Shen, "Rotation and translation mechanisms for tabletop interaction," in

TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems. Washington, DC, USA: IEEE Computer Society, 2006, pp. 79–88. 2.5.1

[70] M. Hancock, S. Carpendale, and A. Cockburn, "Shallow-depth 3d interaction: design and evaluation of one-, two- and three-touch techniques," in *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems.* New York, NY, USA: ACM, 2007, pp. 1147–1156. 2.5.1

[71] R. Kruger, S. Carpendale, S. D. Scott, and S. Greenberg, "How people use orientation on tables: comprehension, coordination and communication," in *GROUP '03: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work.* New York, NY, USA: ACM, 2003, pp. 369–378. 2.5.1

[72] K. Fishkin and P. Kenneth, "A taxonomy for and analysis of tangible interfaces," *Personal Ubiquitous Comput.*, vol. 8, no. 5, pp. 347–358, 2004. 2.5.2

[73] H. Ishii and B. Ullmer, "Tangible bits: towards seamless interfaces between people, bits and atoms," in *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems.* New York, NY, USA: ACM, 1997, pp. 234–241. 2.5.2

[74] M. L. Maher and M. J. Kim, "Studying designers using a tabletop system for 3d design with a focus on the impact on spatial cognition," in *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems.* Washington, DC, USA: IEEE Computer Society, 2006, pp. 105–112. 2.5.2

[75] B. Buxton. (2009, november) Multi-touch systems that i have known and loved. [Online]. Available: http://www.billbuxton.com/multitouchOverview.html 2.6.1

[76] T. G. Zimmerman, J. R. Smith, J. A. Paradiso, D. Allport, and N. Gershenfeld, "Applying electric field sensing to human-computer interfaces," in *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems.* New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 280–287. 2, A.3

[77] K. Everitt, C. S., K. Ryall, and C. Forlines, "Multispace: enabling electronic document micro-mobility in table-centric, multi-device environments," Jan. 2006, pp. 8 pp.–. 2, A.3

[78] F. Wang and X. Ren, "Empirical evaluation for finger input properties in multi-touch interaction," in *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems.* New York, NY, USA: ACM, 2009, pp. 1063–1072. 2, A.1

[79] T. Geller, "Interactive tabletop exhibits in museums and galleries," *Computer Graphics and Applications, IEEE*, vol. 26, no. 5, pp. 6–11, Sept.-Oct. 2006. 2.7

[80] J. Gibson, *The Ecological Approach to Visual Perception.* Hillsdale, NJ: Lawrence Erlbaum Associates, 1986. 2.7.1

[81] W. W. Gaver, "Technology affordances," in *SIGCHI 1991.* ACM Press, 1991, pp. 79–84. 2.7.1

[82] D. Kirsh, "The context of work," *Human-Computer Interaction*, vol. 16, no. 2, pp. 305–322, 2001. 2.7.1

[83] D. A. Norman, *The Psychology of Everyday Things.* Basic Books, April 1988. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20\&amp;path=ASIN/0465067093 2.7.1

[84] J. Rasmussen and W. B. Rouse, *Human detection and diagnosis of system failures*, J. Rasmussen and W. B. Rouse, Eds. Plenum Press, 1981. 2.7.1

[85] K. J. Vicente, *Global Perspectives on the Ecology of Human-Machine Systems.* Hillsdale, New Jersey Hove, UK: L. Erlbaum Asso, 1995, ch. A few implications of an ecological approach to human factors, pp. 54–67. 2.7.1

[86] D. Woods, *Global Perspective on the Ecology of Human-Machine Systems.* L. Erlbaum Associates, 1995, ch. Toward a theoretical base for representation design in the computer medium: ecological perception and aiding cognition, pp. 157–188. 2.7.1

[87] K. C. Dohse, T. Dohse, J. Still, and D. Parkhurst, "Enhancing multi-user interaction with multi-touch tabletop displays using hand tracking," *International Conference on Advances in Computer-Human Interaction*, vol. 0, pp. 297–302, 2008. 2.8

[88] H. Blum, "Biological shape and visual science," *Theoretical Biology*, vol. 38, pp. 205–287, 1973. 3.1, 3.1.1, 3.3.2

[89] M. Couprie, D. Coeurjolly, and R. Zrour, "Discrete bisector function and euclidean skeleton in 2d and 3d," *Image Vision Comput.*, vol. 25, no. 10, pp. 1543–1556, 2007. 3.1

[90] K. Siddiqi, S. Bouix, A. Tannenbaum, and S. W. Zucker, "Hamilton-jacobi skeletons," *Int. J. Comput. Vision*, vol. 48, no. 3, pp. 215–231, 2002. 3.1

[91] R. Jain, R. Kasturi, and B. G. Schunck, *Machine vision.* New York, NY, USA: McGraw-Hill, Inc., 1995, ch. 2.5.10, p. 55. 3.1

[92] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001, ch. 11.1.5, p. 650. 3.1

[93] E. R. Dougherty, *An Introduction to Morphological Image Processing*, S. of Photo-optical Instrumentation Engineers, Ed. SPIE Optical Engineering Press, 1992. 3.1

[94] A. K. Jain, *Fundamentals of digital image processing.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989, ch. 9.9, p. 382. 3.1, 3.1

[95] J. Serra, *Image Analysis and Mathematical Morphology.* Orlando, FL, USA: Academic Press, Inc., 1982. 3.1, 3.2.1.1, B.5.2

[96] J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science (Cambridge ... on Applied and Computational Mathematics).* Cambridge University Press, June 1999, ch. 17.5.2, p. 234. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20\&amp;path=ASIN/0521645573 3.1

[97] J. C. Russ, *The Image Processing Handbook, Fifth Edition (Image Processing Handbook).* Boca Raton, FL, USA: CRC Press, Inc., 2006, ch. Processing Binary Images (skeletonization), pp. 498–499. 3.1

[98] H. Choi, S. Choi, and H. Moon, "Mathematical theory of medial axis transform," *Pacific J. Math*, vol. 181, pp. 57–88, 1997. 3.1

[99] H. Blum, "A transformation for extracting new descriptors of shape," in *Models for the Perception of Speech and Visual Form*, W. W. Dunn, Ed. Cambridge: MIT Press, 1967, pp. 362–380. 3.1

[100] J. Bruce, P. Giblin, and C. Gibson, "Symmetry sets," vol. 101A, 1985, pp. 163–186. 3.1.1

[101] A. K. Jain, *Fundamentals of digital image processing.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989, ch. 9.9, p. 387. 3.1.2

[102] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001, ch. 9.5.7, p. 543. 3.1.3

[103] O. Aichholzer, F. Aurenhammer, D. Alberts, and B. Gärtner, "A novel type of skeleton for polygons," *Journal of Universal Computer Science*, vol. 1, no. 12, pp. 752–761, 1995. 3.1.5

[104] O. Aichholzer and F. Aurenhammer, *Voronoi's Impact on Modern Sciences II.* Proc. Institute of Mathematics of the National Academy of Sciences of Ukraine 21, 1998, ch. Straight skeletons for general polygonal figures in the plane, pp. 7–21. 3.1.5

[105] P. Felkel, "Straight skeleton implementation," in *Proceedings of Spring Conference on Computer Graphics*, 1998, pp. 210–218. 3.1.5

[106] J. Mather, "Distance from a submanifold in euclidean space," in *Symposia in Pure Mathematics*, ser. 2, vol. 40, no. 2, 1983, pp. 199–216. 3.1.6

[107] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, "Shapes, shocks, and deformations i: the components of two-dimensional shape and the reaction-diffusion space," *Int. J. Comput. Vision*, vol. 15, no. 3, pp. 189–224, 1995. 3.1

[108] P. J. Giblin and B. B. Kimia, "On the local form and transitions of symmetry sets, medial axes, and shocks," *Int. J. Comput. Vision*, vol. 54, no. 1-3, pp. 143–156, 2003. 3.1

[109] P. Giblin and B. B. Kimia, "A formal classification of 3d medial axis points and their local geometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 238–251, 2004. 3.1

[110] G. Reeb, "Sur les points singuliers d'une forme de pfaff complèment intégrable ou d'une fonction numérique," in *Comptes Rendus de L'Académie ses Seances*, vol. 222, 1946, pp. 847–849. 3.1

[111] Y. Shinagawa, T. L. Kunii, and Y. L. Kergosien, "Surface coding based on morse theory," *IEEE Computer Graphics and Applications*, vol. 11, no. 5, pp. 66–78, 1991. 3.1

[112] S. Biasotti, D. Attali, J. Boissonnat, H. Edelsbrunner, E. G., M. Mortara, G. Sanniti di Baja, S. M., M. Tanase, and V. R., *Shape Analysis and Structuring.* Springer, 2008, ch. Skeletal Structures, pp. 172–175. 3.1

[113] V. Pascucci, G. Scorzelli, P.-T. Bremer, and A. Mascarenhas, "Robust on-line computation of reeb graphs: simplicity and speed," *ACM Trans. Graph.*, vol. 26, no. 3, p. 58, 2007. 3.1

[114] F. Y. Chin, J. Snoeyink, and C. A. Wang, "Finding the medial axis of a simple polygon in linear time," in *ISAAC '95: Proceedings of the 6th International Symposium on Algorithms and Computation.* London, UK: Springer-Verlag, 1995, pp. 382–391. 3.1, 2

[115] I. Rock and C. M. Linnett, "Is a perceived shape based on its retinal image?" *Perception*, vol. 22, no. 1, pp. 61–76, 1993. 3.1

[116] D. Hoffman and R. W.A., "Parts of recognition," *Cognition*, vol. 18, pp. 65–96, 1984. 3.1

[117] I. Biederman, "Recognition-by-components: a theory of human image understanding," *Psychol Rev*, vol. 94, no. 2, pp. 115–147, Apr 1987. 3.1

[118] M. L. Braunstein, D. D. Hoffman, and A. Saidpour, "Parts of visual objects: an experimental test of the minima rule," *Perception*, vol. 18, pp. 817–826, 1989. 3.1

[119] K. Siddiqi, B. B. Kimia, and K. J. Tresness, "Parts of visual form: psychophysical aspects," *Perception*, vol. 25, no. 4, pp. 399–424, 1996. 3.1

[120] K. Siddiqi and M. Pizer, *Medial Representations: Mathematics, Algorithms and Applications*, ser. Computational Imaging. Springer, 2008, vol. 37, ch. Introduction, pp. 21–25. 3.1

[121] ——, *Medial Representations: Mathematics, Algorithms and Applications*, ser. Computational Imaging. Springer, 2008, vol. 37, ch. Introduction, p. 6. 3.1

[122] A. Lieutier, "Any open bounded subset of rn has the same homotopy type than its medial axis," in *SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications.* New York, NY, USA: ACM, 2003, pp. 65–75. 5

[123] A. Torsello, "Matching hierarchical structures for shape recognition," Ph.D. dissertation, University of York, July 2004. 3.2.1, 3.2.1.2, 3.4, 3.4.1

[124] Y. Xia, "Skeletonization via the realization of the fire front's propagation and extinction in digital binary shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 10, pp. 1076–1086, Oct 1989. 3.2.1

[125] C. Giardina and E. Dougherty, *Morphological methods in image and signal processing.* Prentice Hall, 1988. 3.2.1.1

[126] B. Jang and R. Chin, "Analysis of thinning algorithms using mathematical morphology," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 6, pp. 541–551, 1990. 3.2.1.1

[127] L. Lam, S. Lee, and C. Suen, "Thinning methodologies: A comprehensive survey," *PAMI*, vol. 14, no. 9, pp. 869–885, September 1992. 3.2.1.1

[128] R. Sora, "Object recognition method using a network oriented skeleton computation," vol. 2, May 2006, pp. 381–385. 3.2.1.1, 3.4.3

[129] P. Rockett, “An improved rotation-invariant thinning algorithm,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1671–1674, Oct. 2005. 3.2.1.1

[130] Z. Guo and R. W. Hall, “Parallel thinning with two-subiteration algorithms,” *Commun. ACM*, vol. 32, no. 3, pp. 359–373, 1989. 3.2.1.1

[131] A. Rosenfeld, “A characterization of parallel thinning algorithms,” *InfoControl*, vol. 29, pp. 286–291, November 1975. 3.2.1.1

[132] R. Stefanelli and A. Rosenfeld, “Some parallel thinning algorithms for digital pictures,” *J. ACM*, vol. 18, no. 2, pp. 255–264, 1971. 3.2.1.1

[133] W. Deng, S. S. Iyengar, and N. E. Brener, “A fast parallel thinning algorithm for the binary image skeletonization,” *Int. J. High Perform. Comput. Appl.*, vol. 14, no. 1, pp. 65–81, 2000. 3.2.1.1

[134] F. Leymarie and M. Levine, “Simulating the grassfire transform using an active contour model,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 1, pp. 56–75, 1992. 3.2.1.2, 3.2.2, B.1

[135] H. Tek and B. Kimia, “Symmetry maps of free-form curve segments via wave propagation,” vol. 1, 1999, pp. 362–369. 3.2.1.2

[136] Z. S. G. Tari, J. Shah, and H. Pien, “Extraction of shape skeletons from grayscale images,” *Comput. Vis. Image Underst.*, vol. 66, no. 2, pp. 133–146, 1997. 3.2.1.2

[137] A. Torsello and E. Hancock, “Correcting curvature-density effects in the hamilton jacobi skeleton,” *IEEE Transactions on Image Processing*, vol. 15, pp. 877–891, Apr. 2006. 3.2.1.2

[138] C. Arcelli and G. S. Di Baja, “A one-pass two-operation process to detect the skeletal pixels on the 4-distance transform,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 4, pp. 411–414, 1989. 3.2.2

[139] A. Rosenfeld and J. L. Pfaltz, “Sequential operations in digital picture processing,” *J. ACM*, vol. 13, no. 4, pp. 471–494, 1966. 3.2.2

[140] C. Arcelli and G. S. Di Baja, “A width-independent fast thinning algorithm,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-7, no. 4, pp. 463–474, July 1985. 3.2.2

[141] A. Meijster, J. Roerdink, and W. H. Hesselink, "A general algorithm for computing distance transforms in linear time," in *Mathematical Morphology and its Applications to Image and Signal Processing.* Kluwer, 2000, pp. 331–340. 3.2.2, B.1

[142] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman, "Linear time euclidean distance transform algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 529–533, 1995. 3.2.2, B.1

[143] W. Choi, K. Lam, and W. Siu, "Extraction of the euclidean skeleton based on a connectivity criterion," *Pattern Recognition*, vol. 36, pp. 721–729, 2003. 3.2.2

[144] M. Schmitt, *Geometry and Robotics.* Tolouse, France: Springer, 1989, vol. 391, ch. Some Examples of Algorithms Analysis in Computational Geometry by Means of Mathematical Morphology Techniques, pp. 225–246. 3.2.3

[145] J. W. Brandt and V. R. Algazi, "Continuous skeleton computation by voronoi diagram," *CVGIP: Image Underst.*, vol. 55, no. 3, pp. 329–338, 1992. 3.2.3, 3.2.3.1, 3.3.1

[146] R. Ogniewicz and M. Ilg, "Voronoi skeletons: theory and applications," in *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, Jun 1992, pp. 63–69. 3.2.3

[147] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Comput. Surv.*, vol. 23, no. 3, pp. 345–405, 1991. 3.2.3, B.3

[148] B. Aronov, "A lower bound on voronoi diagram complexity," *Inf. Process. Lett.*, vol. 83, no. 4, pp. 183–185, 2002. 3.2.3, B.3

[149] A. Aggarwal, L. Guibas, J. Saxe, and P. Shor, "A linear time algorithm for computing the voronoi diagram of a convex polygon," in *STOC '87: Proceedings of the nineteenth annual ACM symposium on Theory of computing.* New York, NY, USA: ACM, 1987, pp. 39–45. 3.2.3, B.3

[150] B. Delaunay, "Sur la sphère vide," *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, vol. 7, pp. 793–800, 1934. 3.2.3

[151] L. Guibas and J. Stolfi, "Primitives for the manipulation of general subdivisions and the computation of voronoi," *ACM Trans. Graph.*, vol. 4, no. 2, pp. 74–123, 1985. 3.2.3

[152] J.-D. Boissonnat, "Geometric structures for three-dimensional shape representation," *ACM Trans. Graph.*, vol. 3, no. 4, pp. 266–286, 1984. 3.2.3

[153] N. Mayya and V. T. Rajan, "Voronoi diagrams of polygons: a framework for shape representation," *J. Math. Imaging Vis.*, vol. 6, no. 4, pp. 355–378, 1996. 3.2.3

[154] J. Rocha, "Perceptually stable regions for arbitrary polygons," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 33, no. 1, pp. 165–171, Feb 2003. 3.2.3.1

[155] K. Siddiqi and M. Pizer, *Medial Representations: Mathematics, Algorithms and Applications*, ser. Computational Imaging. Springer, 2008, vol. 37, ch. 7, pp. 227–232. 3.2.3.1

[156] H. Asada and M. Brady, "The curvature primal sketch," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 1, pp. 2–14, 1986. 3.2.3.1

[157] I. Weiss, "Curve fitting withoptimal mesh point placement," Computer Vision Laboratory. University of Maryland, Tech. Rep. CAR-TR-22, 1986. 3.2.3.1

[158] ——, "Shape reconstruction on a varying mesh," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 4, pp. 345–362, 1990. 3.2.3.1

[159] F. Attneave, "Some informational aspects of visual perception," *Psychological Review*, vol. 61, no. 3, pp. 183–193, May 1954. [Online]. Available: http://view.ncbi.nlm.nih.gov/pubmed/13167245 3.2.3.1

[160] D. Attali, "r-regular shape reconstruction from unorganized points," in *SCG '97: Proceedings of the thirteenth annual symposium on Computational geometry.* New York, NY, USA: ACM, 1997, pp. 248–253. 3.2.3.1, 3.2.2, 3.2.3.1

[161] J. W. Brandt, “Convergence and continuity criteria for discrete approximations of the continuous planar skeleton,” *CVGIP: Image Underst.*, vol. 59, no. 1, pp. 116–124, 1994. 3.2.2, 4a

[162] F. Chazal and A. Lieutier, “The "$lambda$-medial axis",” *Graph. Models*, vol. 67, no. 4, pp. 304–331, 2005. 3.2.3, 3.2.3.1, 3.2.5

[163] ——, “Weak feature size and persistent homology: computing homology of solids in rn from noisy data samples,” in *SCG ’05: Proceedings of the twenty-first annual symposium on Computational geometry.* New York, NY, USA: ACM, 2005, pp. 255–262. 3.2.3.1, 3.2.3.1

[164] N. Amenta, M. Bern, and D. Eppstein, “The crust and the beta-skeleton: Combinatorial curve reconstruction,” *Graphical models and image processing: GMIP*, vol. 60, no. 2, pp. 125–135, 1998. [Online]. Available: citeseer.ist.psu.edu/amenta98crust.html 3.2.8

[165] D. Shaked and A. M. Bruckstein, “Pruning medial axes,” *Comput. Vis. Image Underst.*, vol. 69, no. 2, pp. 156–169, 1998. 3.3.1, 3.3.1

[166] F. Mokhtarian and A. K. Mackworth, “A theory of multiscale, curvature-based shape representation for planar curves,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 8, pp. 789–805, 1992. 3.3.1

[167] S. W. Choi and H.-P. Seidel, “Linear onesided stability of mat for weakly injective 3d domain,” in *SMA ’02: Proceedings of the seventh ACM symposium on Solid modeling and applications.* New York, NY, USA: ACM, 2002, pp. 344–355. 3.3.1

[168] A. R. Dill, M. D. Levine, and P. B. Noble, “Multiple resolution skeletons,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 4, pp. 495–504, 1987. 3.3.1, 1

[169] G. Sanniti di Baja and E. Thiel, “A multiresolution shape description algorithm,” in *CAIP ’93: Proceedings of the 5th International Conference on Computer Analysis of Images and Patterns.* London, UK: Springer-Verlag, 1993, pp. 208–215. 3.3.1

[170] D. Attali and J. O. Lachaud, “Delaunay conforming iso-surface, skeleton extraction and noise removal,” *Computational Geometry: Theory and Applications*, vol. 19, pp. 175–189, 2001. 3.3.1

[171] D. Attali and A. Montanvert, “Modeling noise for a better simplification of skeletons,” in *In Proc. Internat. Conf. Image Process*, vol. 3, 1996, pp. 13–16. 3.3.1

[172] T. K. Dey and W. Zhao, “Approximate medial axis as a voronoi subcomplex,” in *SMA ’02: Proceedings of the seventh ACM symposium on Solid modeling and applications.* New York, NY, USA: ACM, 2002, pp. 356–366. 3.3.1

[173] H. Edelsbrunner and E. P. Mücke, “Three-dimensional alpha shapes,” *ACM Trans. Graph.*, vol. 13, no. 1, pp. 43–72, 1994. 3.3.1

[174] M. Foskey, M. C. Lin, and D. Manocha, “Efficient computation of a simplified medial axis,” in *SM ’03: Proceedings of the eighth ACM symposium on Solid modeling and applications.* New York, NY, USA: ACM, 2003, pp. 96–107. 3.3.1

[175] R. Ogniewicz, “Skeleton-space: a multiscale shape description combining region and boundary information,” in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR ’94., 1994 IEEE Computer Society Conference on*, Jun 1994, pp. 746–751. 3.3.1, 3.3.1

[176] J. August, K. Siddiqi, and S. Zucker, “Ligature instabilities in the perceptual organization of shape,” in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2, 1999, pp. –48 Vol. 2. 3.3.2

[177] J. August, A. Tannenbaum, and S. Zucker, “On the evolution of the skeleton,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1, 1999, pp. 315–322 vol.1. 3.3.2

[178] M. v. Eede, D. Macrini, A. Telea, C. Sminchisescu, and S. S. Dickinson, “Canonical skeletons for shape matching,” in *ICPR ’06: Proceedings of the 18th International Conference on Pattern Recognition.* Washington, DC, USA: IEEE Computer Society, 2006, pp. 64–69. 3.3.2

[179] D. Macrini, K. Siddiqi, and S. Dickinson, “From skeletons to bone graphs: Medial abstraction for object recognition,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8. 3.3.2

[180] C. Di Ruberto, "Recognition of shapes by attributed skeletal graphs," *Pattern Recognition*, vol. 37, no. 11, pp. 21–31, 2004. 3.4.1

[181] B. Luo and E. R. Hancock, "Structural graph matching using the em algorithm and singular value decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 10, pp. 1120–1136, 2001. 3.4.1

[182] S. C. Zhu and A. L. Yuille, "Forms: a flexible object recognition and modeling system," *Int. J. Comput. Vision*, vol. 20, no. 3, pp. 187–212, 1996. 3.4.1

[183] W.-B. Goh, "Strategies for shape matching using skeletons," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 326–345, 2008. 3.4.1

[184] X. Bai and L. Latecki, "Path similarity skeleton graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1282–1292, 2008. 3.4.1

[185] L. He, C. Han, and W. Wee, "Object recognition and recovery by skeleton graph matching," in *Multimedia and Expo, 2006 IEEE International Conference on*, July 2006, pp. 993–996. 3.4.1

[186] A. Lozano, R. Y. Pinter, O. Rokhlenko, G. Valiente, and M. Ziv-Ukelson, "Seeded tree alignment," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 5, no. 4, pp. 503–513, 2008. 3.4.1

[187] D. Sharvit, J. Chan, H. Tek, and B. B. Kimia, "Symmetry-based indexing of image databases," in *CBAIVL '98: Proceedings of the IEEE Workshop on Content - Based Access of Image and Video Libraries.* Washington, DC, USA: IEEE Computer Society, 1998, p. 56. 3.4.2

[188] K. Siddiqi, A. Shokoufandeh, S. Dickenson, and S. Zucker, "Shock graphs and shape matching," Jan 1998, pp. 222–229. 3.4.2, B.5.3, B.5.6, B.5.8

[189] M. Pelillo, K. Siddiqi, and S. W. Zucker, "Matching hierarchical structures using association graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 11, pp. 1105–1120, 1999. 3.4.2

[190] T. B. Sebastian, P. N. Klein, and B. B. Kimia, "Recognition of shapes by editing their shock graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 550–571, 2004. 3.4.2, B.5.8

[191] M. Fatih Demirci, A. Shokoufandeh, and S. J. Dickinson, "Skeletal shape abstraction from examples," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 944–952, 2009. 3.4.2

[192] F. Leymarie and M. D. Levine, *Progress in Image Analysis and Processing.* Singapore: World Scientific, 1989, ch. Snakes and skeletons, pp. 186–193. 1

[193] A. Beristain. (2009, March) Skeletonization prototype in use in real time. [Online]. Available: http://www.ehu.es/ccwintco/uploads/8/80/SkeletonPrototypeInUse_divX_Montaje.avi 5.4

[194] D. F. Specht, "Probabilistic neural networks," *Neural Netw.*, vol. 3, no. 1, pp. 109–118, 1990. 5.6.1.2

[195] J. Pennock and M. N. Tabrizi, "A survey of input sensing and processing techniques for multi-touch systems." in *CDES*, H. R. Arabnia, Ed. CSREA Press, 2008, pp. 10–16. A.1, A.1

[196] N. Metha, "A flexible machine interface," Ph.D. dissertation, Department of Electrical Engineering, University of Toronto, 1982. A.1

[197] L. H. Nakatani and J. A. Rohrlich, "Soft machines: A philosophy of user-computer interface design," in *CHI '83: Proceedings of the SIGCHI conference on Human Factors in Computing Systems.* New York, NY, USA: ACM, 1983, pp. 19–23. A.1

[198] M. Krueger, *Artificial Reality.* AddisonWesley, 1983, p. 312 pp. A.1

[199] W. Krueger, Myron, *Artificial Reality II.* Addison-Wesley, 1991. A.1

[200] J. Y. Han, "Low-cost multi-touch sensing through frustrated total internal reflection," in *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology.* New York, NY, USA: ACM, 2005, pp. 115–118. A.1, A.2.1

[201] B. Buxton. (2009, november) Multi-touch systems that i have known and loved. [Online]. Available: http://www.billbuxton.com/multitouchOverview.html A.1

[202] Multi-touch technologies. Natural User Interface Group (NUI). [Online]. Available: http://nuicode.com/projects/wiki-book/files A.2

[203] W. E. Gettys, K. F. K., and M. J. Skove, *Physics: Classical and Modern.* New York: McGraw-Hill, 1989, ch. Total Internal Reflection, p. 779. A.2.1

[204] R. Rashed, "A pioneer in anaclastics: Ibn sahl on burning mirrors and lenses." *ISIS*, vol. 81, pp. 464–491, 1990. A.2.1

[205] J. D. Smith, T. Nicholas Graham, D. Holman, and J. Borchers, "Low-cost malleable surfaces with multi-touch pressure sensitivity," *Horizontal Interactive Human-Computer Systems, International Workshop on*, vol. 0, pp. 205–208, 2007. A.2.1, A.2.1.1

[206] Y. Lucet, "A linear euclidean distance transform algorithm based on the linear-time legendre transform," in *CRV '05: Proceedings of the 2nd Canadian conference on Computer and Robot Vision.* Washington, DC, USA: IEEE Computer Society, 2005, pp. 262–267. B.1

[207] L. J. Latecki and R. Lakämper, "Convexity rule for shape decomposition based on discrete contour evolution," *Comput. Vis. Image Underst.*, vol. 73, no. 3, pp. 441–454, 1999. B.2, B.2, B.2

[208] ——, "Shape similarity measure based on correspondence of visual parts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1185–1190, 2000. B.2

[209] ——, "Application of planar shape comparison to object retrieval in image databases," *Pattern Recognition*, vol. 35, pp. 15–29, 2002. B.2, B.2

[210] K. Buchin, "Delaunay triangulations in linear time? (part i)," *CoRR*, vol. abs/0812.0387, 2008. B.3.2

[211] E. Bengoetxea, "Inexact graph matching using estimation of distribution algorithms," Ph.D. dissertation, Ecole Nationale Supérieure des Télécommunications, Paris, France, 2003. B.4

[212] N. Joshi, G. Sita, A. G. Ramakrishnan, and S. Madhvanath, "Comparison of elastic matching algorithms for online tamil handwritten character recognition," in *IWFHR '04: Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition.* Washington, DC, USA: IEEE Computer Society, 2004, pp. 444–449. B.4.3.1

[213] H. Bunke and K. Shearer, "A graph distance metric based on the maximal common subgraph," *Pattern Recogn. Lett.*, vol. 19, no. 3-4, pp. 255–259, 1998. B.4.3.4

[214] J. Lladoós, E. Martí, and J. J. Villanueva, "Symbol recognition by error-tolerant subgraph matching between region adjacency graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 10, pp. 1137–1143, 2001. B.4.3.4

[215] H. Bunke, "Error correcting graph matching: On the influence of the underlying cost function," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 917–922, 1999. B.4.3.4

[216] B. Messmer and H. Bunke, "A new algorithm for error-tolerant subgraph isomorphism detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, pp. 493–504, 1998. B.4.3.4, B.4.3.5

[217] K. G. Khoo and P. N. Suganthan, "Evaluation of genetic operators and solution representations for shape recognition by genetic algorithms," *Pattern Recogn. Lett.*, vol. 23, no. 13, pp. 1589–1597, 2002. B.4.3.5

[218] S. Auwatanamongkol, "Inexact graph matching using a genetic algorithm for image recognition," *Pattern Recogn. Lett.*, vol. 28, no. 12, pp. 1428–1437, 2007. B.4.3.5

[219] R. Myers and E. R. Hancock, "Genetic algorithms for ambiguous labelling problems," in *EMMCVPR '97: Proceedings of the First International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition.* London, UK: Springer-Verlag, 1997, pp. 345–360. B.4.3.5

[220] E. Hancock and J. Kittler, "Edge-labeling using dictionary-based relaxation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 2, pp. 165–181, 1990. B.4.3.5

[221] J. Kittler, W. Christmas, and M. Petrou, "Probabilistic relaxation for matching problems in computer vision," in *Computer Vision, 1993. Proceedings., Fourth International Conference on*, May 1993, pp. 666–673. B.4.3.5

[222] L. B. Shams, M. J. Brady, and S. Schaal, “Graph matching vs mutual information maximization for object detection,” *Neural Netw.*, vol. 14, no. 3, pp. 345–354, 2001. B.4.3.5

[223] R. Wilson and E. Hancock, “Structural matching by discrete relaxation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 6, pp. 634–648, Jun 1997. B.4.3.5

[224] H.-Y. Kim and J. H. Kim, “Hierarchical random graph representation of handwritten characters and its application to hangul recognition,” *Pattern Recognition*, vol. 34, pp. 187–201, 2001. B.4.3.5

[225] K. Shearer, H. Bunke, and S. Venkatesh, “Video indexing and similarity retrieval by largest common subgraph detection using decision trees,” *Pattern Recognition*, vol. 34, no. 5, pp. 1075 – 1091, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/B6V14-42810JK-C/2/97ba992f6082cb8d8b267ac80fc327ad B.4.3.5

[226] D. Rivière, J.-F. Mangin, D. Papadopoulos-Orfanos, J.-M. Martinez, V. Frouin, and J. Régis, “Automatic recognition of cortical sulci using a congregation of neural networks,” in *MICCAI ’00: Proceedings of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention.* London, UK: Springer-Verlag, 2000, pp. 40–49. B.4.3.5

[227] D. Rivière, J. Mangin, D. Papadopoulos-Orfanos, J. Martinez, V. Frouin, and J. Régis, “Automatic recognition of cortical sulci of the human brain using a congregation of neural networks,” *Elsevier, Medical Image Analysis*, vol. 6, p. 7792, 2002. B.4.3.5

[228] A. Sanfeliu, R. Alquézar, and F. Serratosa, “Clustering of attributed graphs and unsupervised synthesis of function-described graphs,” *Pattern Recognition, International Conference on*, vol. 2, p. 6022, 2000. B.4.3.5

[229] T. Caelli and S. Kosinov, “An eigenspace projection clustering method for inexact graph matching,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 4, pp. 515–519, 2004. B.4.3.5

[230] S. Medasani, R. Krishnapuram, and Y. Choi, “Graph matching by relaxation of fuzzy assignments,” *Fuzzy Systems, IEEE Transactions on*, vol. 9, no. 1, pp. 173–182, Feb 2001. B.4.3.5

[231] Y. Keselman, A. Shokoufandeh, M. Demirci, and S. Dickinson, “Many-to-many graph matching via metric embedding,” in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1, June 2003, pp. I–850–I–857 vol.1. B.4.3.5

[232] A. Hlaoui and S. Wang, “A new algorithm for inexact graph matching,” in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 4, 2002, pp. 180–183 vol.4. B.4.3.5

[233] M. A. Eshera and K. S. Fu, “An image understanding system using attributed symbolic representation and inexact graph-matching,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 5, pp. 604–618, 1986. B.4.3.5

[234] P. Lax, *Contributions to Nonlinear Functional Analysis.* New York: Academic Press, 1971, ch. Shock waves and entropy. B.5.2

[235] H. R. Lewis and C. H.-H. Papadimitriou, *Elements of the Theory of Computation*, 1st ed. Englewood Cliffs, NJ: Prentice-Hall, 1981. B.5.4

[236] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness.* New York, NY, USA: W. H. Freeman & Co., 1990. B.5.5

[237] J. Edmonds and D. W. Matula, “An algorithm for subtree identification,” *SIAM Rev.*, vol. 10, pp. 273–274, 1968, abstract. B.5.5

[238] R. M. Verma and S. W. Reyner, “An analysis of a good algorithm for the subtree problem, correlated,” *SIAM J. Comput.*, vol. 18, no. 5, pp. 906–908, 1989. B.5.5

[239] J. Hopcroft and R. Karp, “An n(5/2) algorithm for maximum matching in bipartite graphs,” *SIAM J. Comput.*, pp. 225–231, 1975. B.5.5

[240] K. Siddiqi and B. Kimia, “A shock grammar for recognition,” in *CVPR 96*, 1996, pp. 507–513. B.5.7