MASTER'S DEGREE
MULTIMEDIA – SPECIALIZATION IN DIGITAL GAMES AND INTERACTIVE TECHNOLOGIES

Tangible tokens for multi-touch interfaces based on extended triangular patterns

André Colares Pinto Ramos

M
2022

FACULDADES PARTICIPANTES:

FACULDADE DE ENGENHARIA
FACULDADE DE BELAS ARTES
FACULDADE DE CIÊNCIAS
FACULDADE DE ECONOMIA
FACULDADE DE LETRAS

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Tangible tokens for multi-touch interfaces based on extended triangular patterns

André Colares Pinto Ramos

WORKING VERSION

Master in Multimedia

Supervisor: Rui Rodrigues

Second Supervisor: Daniel Mendes and Eduardo Magalhães

August 4, 2022

Abstract

In this dissertation it will be explored the world of tangibles, more specifically, their interaction with multi-touch screens, how they work, the different techniques that have been implemented over the years, and the interaction between user and machine that the technology of pattern recognition allows.

In order to explore this technology and its applications, it was created an algorithm capable of recognizing specific triangular patterns as well as expansions of these same patterns, by adding interior touch points to the already existing patterns. Through this technique, it is possible to use and reuse triangles with unique characteristics, maintaining the interactions they normally allow, such as rotation.Therefore, this method extends the usage of touch points on any tangible surface, allowing the user to interact with a custom interface without breaking the patterns that are associated with each object.

The algorithm developed in this thesis explores the spatial area of the screen, identifying the position of the touch points and the angles generated from the patterns inserted in the tokens.

Firstly, the pattern is created in the token, which is then associated with a specific isosceles triangle - represented by its differentiating angle - that the algorithm recognizes and associates with an object, thus deeming it a unique representation of a certain item. The usage of different triangles allows for a recognition rate that is more consistent and less vulnerable to human touches that create unrecognizable patterns. Furthermore, to execute this work it was required the creation of unique tokens, which were made with several materials, such as aluminium, copper, and conductive rubber, to simulate the touch points required for the technology to work. Afterwards, the tokens were tested with different screens, in order to test the algorithm on different touch detection technologies.

Multi-touch screens are screens capable of receiving multiple touch inputs, such as the ones in our smartphones or tablets; these inputs are points of contact with the surface of the screen. Tangibles are physical objects that can be sensed and recognized by touch.

This method explores the capabilities of some of the tangible multi-touch screens available in the market, testing their performance with the proposed technique and how reliable and robust they are when presented with the obstacles and problems that individuals face during everyday usage and that were addressed throughout the development of this work.

Finally, the algorithm must be implemented in a web-based environment, since it can be adapted to create websites with token recognition technology embedded, allowing this application to be used in smartphones, tablets, and other multi-touch displays, thus expanding its usage besides the more traditional tabletop displays, and increase accessibility.

Keywords— tangibles, multi-touch, triangular patterns, object recognition, screens, isosceles triangle

Resumo

Esta dissertação explora o mundo das interfaces tangíveis, destacando as suas interações com os ecrãs multi-toque, o funcionamento dos mesmos, as técnicas implementadas até agora e a interação entre usuário e máquina que a tecnologia de reconhecimento de padrão permite.

De forma a explorar esta tecnologia e as suas aplicações, criou-se um algoritmo capaz de reconhecer padrões triângulares específicos, assim como expansões dos mesmos, através da adição de toques ao padrão preexistente.

Através desta tecnologia, é possível utilizar e reutilizar os diferentes triângulos, mantendo a interação que estes normalmente permitem, como, por exemplo, a rotação. Consequentemente, este método permite a extensão do uso de toques de qualquer superfície tangível, possibilitando que o utilizador interaja com uma interface personalizada sem comprometer os padrões já associados a cada objeto.

O algoritmo criado neste projeto explora o espaço da área do ecrã, identificando a posição dos toques e os ângulos gerados pelos padrões inseridos nos tokens. De forma a que fosse possível representar vários objetos, este método utiliza triângulos isósceles. Primeiramente, o padrão é criado no token, sendo depois associado a um triângulo específico - que, sendo isósceles, é identificado pelo seu ângulo diferenciador - que o algoritmo reconhece e associa a determinado objeto, fazendo assim com que cada triângulo isósceles diferente seja a representação única de um item.

O uso dos triângulos diferenciados permite que a taxa de reconhecimento seja mais consistente e menos vulnerável aos toques humanos, que criam padrões irreconhecíveis pelo algoritmo. Adicionalmente, para executar este projeto, foi necessário criar tokens onde se criam os padrões, feitos com vários materiais, como, por exemplo, alumínio, cobre e borracha condutora para simular os toques necessários para que a tecnologia funcione. Seguidamente, já com os padrões implementados, os tokens foram testados em vários ecrãs, de forma a perceber o funcionamento do algoritmo em diferentes métodos de reconhecimento.

Este método explora as capacidades de vários ecrãs multi-toque disponíveis no mercado, testando a interação do equipamento com este tecnologia e quão fiáveis são quando enfrentam obstáculos presentes ao longo do desenvolvimento.

Sendo o algoritmo criado num ambiente "web-friendly", pode ser adaptado para a criação de websites com tecnologia de reconhecimento de padrão integrada, permitindo a aplicação da mesma em smartphones, tablets e outras interfaces multi-toque, expandindo assim a sua aplicabilidade.

Palavras Chave— tangíveis, multi-toque, padrões triangulares, reconhecimento de objectos, ecrãs, triângulo isósceles

Acknowledgements

First of all, I want to thank Professor Rui Rodrigues and Professor Daniel Mendes, for the guidance, tools and knowledge that they provided me with, during this long dissertation period. Moreover, I would also like to thank Professor Eduardo Magalhães for the opportunity presented, as well as the 3Decide team, especially Carlos Rebelo, the company's CEO, and Carlos Bispo, the CEO's right hand, for all the feedback and advices they provided with throughout this project. Secondly, I thank the colleagues who have accompanied me in my academic journey. Namely to the people that I met during my bachelor's degree in Universidade da Beira Interior and later in Faculdade de Engenharia da Universidade do Porto during my master. Thirdly, I would like to thank my family, especially my parents for always giving me what I needed to become a complete member of society, as well as for the love, the affection, and for teaching me that giving up is not an option. Last but not least, I would like to thank my girlfriend Jessica for being always there for me, through the lows and highs, and, most of all, for making me want to be the better version of myself everyday. Throughout my path, there have been other people that contributed to my motivation to keep going and always giving my best in any circumstance.

"Success is not final, failure is not fatal:
It is the courage to continue that counts."

Winston Churchill

Contents

List of Figures

Abbreviations

TUI	Tangible User Interfaces
TUIC	Tangible User Interface Capacitive
TUIO	Tangible User Interface Optical
MIDI	Musical Instrument Digital Interface
PTPG	Poisson Triangular Pattern Generator
PERCs	Persistently Trackable Tangibles on Capacitive Multi-touch Displays
ATM	Automated Teller Machine
2D	Two Dimensional
API	Application Programming Interface

Chapter 1

Introduction

Tangible multi-touch surfaces have found a place in modern-day applications, enabling the user to interact with digital content by manipulating a physical object, defined as a token throughout this thesis [1]. The exploration of the multi-touch screen and the capacity of touch point recognition is used in several scenarios, from design [2], gaming [3], [4] education [5], [6] and other interactive experiences, serving as a tool that enhances the interaction process for the user.

Tangible User Interfaces (TUIs) systems rely on sensing tokens with certain characteristics on the top of the surface display [7] [8] [9], which are designed with a unique pattern, image or magnet, depending on the approach taken. Thus, the desired object can be easily recognized after the algorithm identifies the pattern on the token, which is associated with an accurate position, direction, and behavior of the token. The recognition must focus on the unique characteristics of the pattern and therefore minimize the possibility of failure in its identification.

Technology is in constant improvement, therefore it seems to be an untapped pool of potential. As such, in this area of pattern recognition technologies, there are important aspects to explore, revolving around the touch point capacity detection, which depends on the current screens, and how these interact with this technology. Considering there are several possibilities for pattern recognition technology, developing this area will broaden the space for creativity and accessibility, as well as allow the creation of new concepts for gadgets and a wider variety of applications.

1.1 Context

Nowadays, interactive displays are integrated with the normal functioning of people's lives. Whether it is through the smaller screens of smartphones or bigger ones in tablets, or even when individuals go to the hospital for a doctor's appointment and must check in at the entrance. As such, it is important to keep exploring this technology so improvements can be made and so that this technology can be more commonly used in gadgets such as smartphones or tablets.

For the past two decades, pattern recognition technology, applied to multi-touch screens, has been suffering constant changes and improvements, which impacts directly the interaction between individuals and technology, since those features are used for the most simple things such as gesture recognition, zoom-in or out in our phones or even in design software that enhance the experience of drawing or painting in tablets, as will be explored in the literature review (chapter 2).

Particularly for this dissertation, the proposal was made by the company 3Decide alongside FEUP, aiming to create an algorithm capable of recognizing and extending common patterns already in use. In order to develop this technique, the algorithm must recognize the touch points in the patterns applied in the tokens as the ones needed to identify the object, as well as discard the user's touch points that are not programmed in the algorithm, always to improve or, at least, maintain the current recognition rates.

Moreover, the algorithm should also be web-friendly, meaning, it should be easily adapted to include in a web-based language, so it can facilitate and improve the creation of websites with this technology embedded. By doing this, it would expand the usage of this technology beyond the traditional applications, such as the Kiosk tabletops, promoting the development of new gadgets and equipment useful for several areas and different types of users.

In the past years, there have been many improvements around the multi-touch screens, since it is a constantly used technology, such as, for example, the birth of the recent InGlass technology applied on big display screens.

Regarding tokens - used as the vehicles for the pattern required for the algorithm - there are two main types, depending if they require external help to function. Active tokens require external help, such as batteries or human contact, whereas passive tokens function on their own. When it comes to the design of tokens, it depends on the identification technique used of the main three (magnetic, optical, or capacitive).

Although pattern recognition has a lot of potential, it also has a few limitations due to the current state of the technology used in multi-touch screens, such as example, the screen touch point limitation and the occlusion of touch points when Infrared sensors are used. These can make the recognition of a pattern more challenging and limited.

Since tokens are necessary for pattern recognition to work, this technique is typically used in big-screen displays. The most common use is to transmit visual experiences and seamless interactions between user and machine. To increase its applications, some of the aforementioned limitations must be tackled.

In order to achieve an easier and more attractive use of this technology, it is necessary to improve its versatility and accessibility, through an increase in the number of possible interactions. Afterwards, it will be possible to expand the content creation and increase the economies of scale so that this technology becomes cheaper and thus more available for common users.

1.2 Research Questions

In order to establish a path for the dissertation, it is important to formulate the research questions. Thus, the following work will be guided according to what it intends to answer or contribute to the field, and the questions are also based on the research made during the literature review (chapter 2). As such, the research questions are the following:

- **H**ow can the triangular patterns be extended to multi-point triangle-based patterns?
- **C**an it be applied to browser-based applications, using the already implemented multi-touch APIs?
- **I**s it possible to create a consistent, light, and reliable algorithm that can recognize the common triangular patterns as well as the extended multi-point triangle-based patterns?

1.3 Objectives

The main goal of this thesis is to tackle some of the problems around the multi-touch screen limitation through the extension of triangular pattern recognition, by increasing the number of unique patterns recognizable, thus extending the traditional triangle patterns with the implemented algorithm. Simultaneously, its efficiency and the recognition rate must be maintained, to increase the number of recognition patterns in a reliable way and expand the creative usage of these methods. As such, it is expected the following in this work:

- **C**reate an algorithm capable of recognizing passive tokens with unique patterns and multiple touch points.
- **E**xpand the triangle patterns with additional touch points.
- **I**ncrease or maintain the efficiency of this algorithm, using spatial-based methods of recognition.
- **M**ake it capable of differentiating human touch from token touch.
- **D**esign a unique algorithm that can be applied to low maintenance hardware and maintains a high performance.
- **E**xplore the usage of passive tokens on multi-touch displays.

1.4 Document Structure

This document will be divided into six main chapters: Introduction, Multi-Touch Surfaces, and Tangibles Literature Review, Proposed Algorithm and System Requirements, Algorithm Development and Module Implementations, Tests and Results, Conclusions and Future Work.

In the Multi-Touch Surfaces and Tangibles Literature Review (chapter 2), we will discuss previous work related to pattern recognition, as well as the goal of this thesis. Moreover, it will be explained the first steps to implement this technology, showing projects that use different techniques and some of the approaches and improvements made in the limitations before mentioned.

In the Proposed Algorithm and System Requirements (chapter 3), the attempt at solving the problem stated previously will be explained in further detail. In this chapter, the solution will start to be explained, alongside the goals this dissertation intends to achieve, the techniques to do so, the methodology behind the testing, the metrics to evaluate the solution, and, finally, the path to follow until the final goal is reached.

In the Algorithm Development and Module Implementations (chapter 4), there will be a detailed explanation of the several development phases in this work. It will state the hardware description, how it interferes and interacts with the program, how the software works and the functionalities required from it, the logical process for its construction, how it will interact with the graphic display, and, finally, the tests made alongside the final results.

In the Tests and Results (chapter 5), it is quickly summed up the whole process of developing this dissertation, and the results of the work.

As for the Conclusions and Future Work (chapter 6), it is suggested some areas that can receive improvements in the short term, and other ones that can help to decrease the limitations of this technology.

Chapter 2

Multi Touch Surfaces and Tangibles Literature Review

Introduction

In order to adequately contextualize this work, firstly, it is necessary to introduce the first steps that have been taken with this technology that has allowed the exploration of tangible surfaces, and have a brief notion of the types of multi-touch screens that have been used, for a better understanding of the detection methods. Afterwards, it will be shown some of the first applications and innovations using the multi-touch screen, since these are the first steps in applications using tangibles. Finally, it will be explored the most recent signs of progress made based on the studies already developed on solutions in the field of tangibles and multi-touch surfaces. Moreover, it will be shown other works that provide solutions for some limitations of this technology.

2.1 Multi-touch surfaces

In 2006, Jefferson Y. Han introduces a multi-touch screen that later would generate the smart screen as we know it (Figure 2.1). This screen was capable of receiving input from several touch-points, allowing the user to interact with it in new ways. With multiple touch features now available, there are more things yet to explore in pattern recognition technology.

In 2007, Steve Jobs presented the first iPhone, a revolutionary gadget, that combined three things: a music player, the iPod, a revolutionary mobile phone, and, thirdly, a breakthrough internet communicator. It was the first step toward one of the greatest things that would emerge in the future years. Although there were already the so-called ”smartphones” in the industry, they were not that smart, since they were extremely hard to use and did not allow the execution of all the desired functions in an easy way. However, with the further development of a larger screen, that occupies the majority of the phone’s front surface and is capable of multi-touch actions, the interface, usability, and interaction with the gadget became game-changing as we can see in our society in the present days.

With the growth of the industry, a large number of technological advancements targeted this technology of multi-touch [10, 11, 12, 13], promoting the appearance of a large number of new studies focused on tangibles [14, 8].

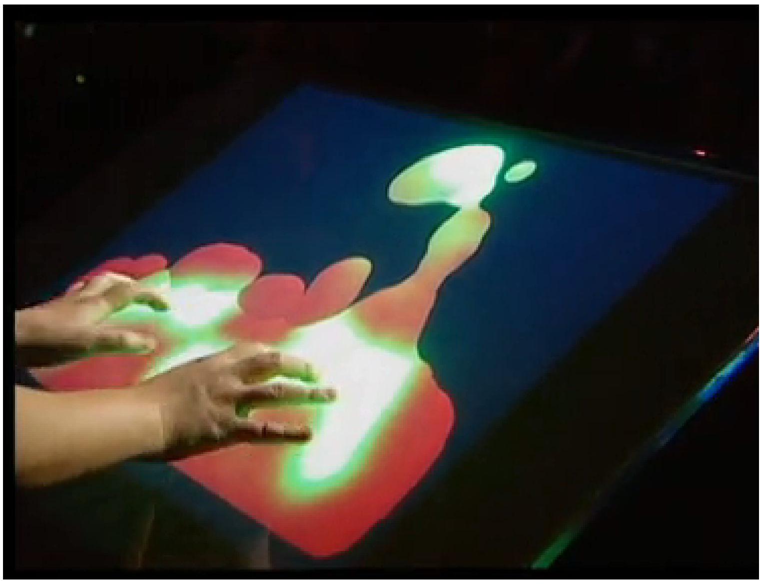

Figure 2.1: Jeff Hans showcase of the multi-touch screen

2.2 Types of multi-touch screens

It is important to understand the functionalities and possibilities of tangibles on multi-touch devices. However, it is also important to know how the different types of screens used in these studies work, in order to better understand the way the systems surface disturbances detection occurs.

2.2.1 Capacitive touch screens

Capacitive touch screens are the most common type of screens, and are present on most smartphones, laptops and tablets. These screens are made of multiple layers of glass and plastic, coated with a conductor material, such as indium tin oxide or copper. These components are capable of conducting electrical signals from a source (for example, people's fingers or batteries) to itself. Therefore they require less pressure to operate than resistive touch screens. When these two make contact with each other, the electrical circuit is completed, thus, the contact point is the exact position where the input is received; once this event is registered, the screen receptors signal it to the operating system, prompting a response from the device. Capacitive touch screens are quite sensible, therefore they are very good to interpret interactions; moreover, they display great imagery and offer more flexibility in functionality and if the screen is cracked, it is still capable of proper functioning.

2.2.2 Resistive touch screens

Resistive touch screens, as the name indicates, use resistance to detect the touches. They have two layers, one made of glass or hard plastic, and another blanketed by a resistive metallic layer that conducts electrical charge. These layers are separated by a small space, an air gap. Once a touch is executed on the screen surface, the two layers are pressed against each other, allowing precise identification of the position of the touch point. These screens are normally used in ATM machines, checkout stands, and post of sales terminals. They are more durable and affordable than capacitive screens, but they cannot register multiple touch points simultaneously.

2.2.3 InGlass touch screens

InGlass technology is one of the latest touch screen technology. These screens use optical emitters and detectors that inject specific light wavelengths, which create a light grid inside the core glass of the screen. With this, any tiny disturbance in the grid created by an object placed on the surface is immediately detected on the receivers. The in-built algorithm determines all touch and pen coordinates in real-time, with almost no delay, creating a seamless experience between users and machine, just like Infrared technology. The system to detect touch points of both InGlass and Infrared work in a very similar way, but the InGlass is not affected by electrical interference, thus making the experience more intuitive.

2.2.4 Infrared touch screens

Infrared technology is considered an entry-level solution in touch screen technology. These screens use an embedded overlay of infrared beams and sensors around the screen frame. When an object interacts, such as a finger or a stylus, the beams are intersected and the device registers a touch point in the intersection. In this work, in which it is necessary for multi-touch features to detect patterns, this technology is more susceptible to touch point occlusion. Consequently, this might be an unreliable technology to use if the patterns have several touch points and the screen has a small area.

2.3 Tangibles

Multi-touch technologies are not a recent discovery. In the late 1960s, touch screens were used in Great Britain to control air traffic. However, the initial programs could only sense one touch at a time. Although the general public only became aware of devices that sensed multiple simultaneous touch locations with the launch of the Apple iPhone in 2007, this capability had already been demonstrated and developed in January 1984 [15].

Even before the multi-touch screen announcement of Jefferson Y. Han, and the introduction of the iPhone by Steve Jobs, the ARTag was published in 2005 [16], which is a marker system that uses a unique algorithm system that recognizes patterns precisely. To recognize such patterns, tokens for the ARTag system use black and white images, quite similar to the QR Code technology, allowing for efficient object recognition (Figure 2.2). Since the tokens have patterns containing a unique ID number encoded with robust digital techniques, ARTag has very low and numerically quantifiable error rates and can encode up to 2002 different unique IDs with no need to store patterns.

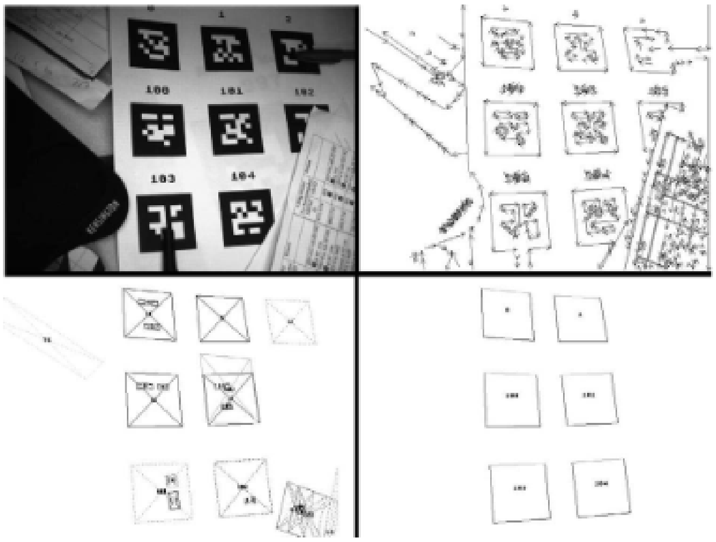

Figure 2.2: ARTag Pattern Recognition

Additionally, in 2005, several authors [17] developed a study that developed a framework to extend tangible interfaces for the field of education - the so-called "Digital Montessori-inspired Manipulatives" - which a specific focus on abstract problem domains. In this work, it was created two sets of manipulatives, physical objects specifically designed to promote learning. The first none is FiMs (Forebel-inspired Manipulatives), which are building toys that allow children to design real-world things, objects, and structures, for example, a castle made of wood blocks. Secondly, there are MiMs (Montessori-inspired Manipulatives), which are computational enhanced building blocks as well, but instead are focused and designed to approach abstract concepts, for example, arranging Cuisenaire rods in different configurations that make numerical proportions more noticeable. Whilst FiMs encourage design behaviors, MiMs strive on making abstract concepts more salient. These Tangibles interfaces looked to bring advantages in learning topics. These brought some advantages such as:

- **Sensory engagement:** the natural way children learn, engaging multiple senses (touch, vision, audition) in a constructive process.
- **Accessibility:** dramatically improves accessibility to younger children and people with learning disabilities.
- **Group learning:** provides multi-hand interfaces and does not give control to a single individual, therefore it facilitates natural group interaction and discussion.

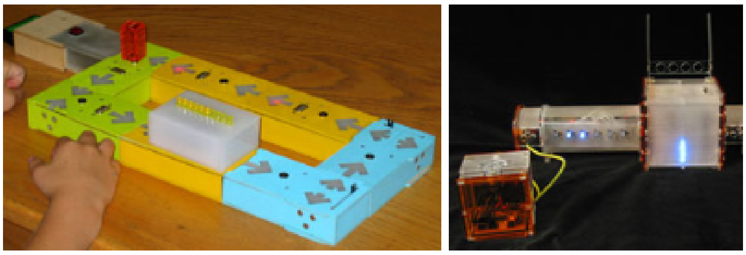

Figure 2.3: FiMs and MiMs

Since then, there has been an increasing trend in the world of tangibles interfaces and more studies were conducted. In 2007, the affirmation of physical interfaces in the learning space had positive results, being necessary that the interaction between the display and the physical object be as seamless as possible. More importantly, these tangibles could be used for a greater amount of applications.

In 2011, several authors [18] wrote a paper regarding the use of tangible surfaces in the Opera-2000 Electronic arenas for performance, culture, arts, and entertainment. This paper studies the interaction between actors on stage through a tangible surface; it links the interactions between the performers to a computer through tangibles in the multi-media opera Amazonas. Essentially, the actors of the opera were given a tangible surface to use in the show, interacting with it, by using obvious and perceptible gestures to an audience. Some of these gestures were: push button, place object, move a finger, moving object, and object placing. Therefore the tangible surface provided the following interaction functionalities:

- **T**rack the position of fingers.
- **T**rack the position and orientation of objects.
- **C**lassify objects according to visual labels (fiducial markers).
- **T**rack the position of acoustic events.
- **C**lassify the acoustic events according to the type of sound; e.g. knocking, brushing, drumming.

The introduction of a tangible surface and the interaction with it by the actors have provided a new way of interaction on stage and the possibility to present information.

Figure 2.4: Amazonas performance with an interactive surface

By this point, there were still technological limitations with multi-touch displays. However, throughout the years there has been a big effort to solve some of these questions, through design changes [2] and new algorithm implementations on token methods [19]. Tangibles have been exploring new ways for a viable, simple and seamless implementation on the multi-touch screen world [20, 21, 22].

To tackle the technical limitation of the multi-touch screen capacity, several studies on the algorithm and the physical body of the token were made. Tokens can be recognized by using different techniques (optical, magnetic, or capacitive), and each one has a unique method. Some of them can be more expensive but more accurate, such as the capacitive technique with an active token [8], whilst others - in the optical technique - use image recognition, which is not suitable for all the used devices [23]. Finally, there is also the magnetic technique, that uses magnetic patterns, which are accurate but require some sort of shielding from the magnets [9]. These are great tools to use, but they are not very accessible for the majority of the producers. Even though they have a great outcome of accuracy and performance, there are cheaper methods that can compete with the technical aspect of these, by using pattern recognition. A simple token can contain a unique pattern, normally composed of three to four touch contact points, to better suit the most common multi-touch screen, which has a restriction of ten to twenty simultaneous touches.

2.4 Identification techniques

Over the years there has been a reasonable amount of diverse techniques used to achieve a number of the features that this technology is capable of. Firstly, it is important to tell apart the various methods used for the token type, which may be active or passive. Passive tokens only use the base's visual/material properties for the recognition and identification, using an image [23], a pattern [24] or even magnetic frequencies [25], with no other supporting technology. Active tokens have the assistance of an additional component, for instance, a small field sensor as in PERCS [19] (Figure

2.5), or a smartwatch circuit board as in Actibles [26] which support the recognition and usage of the token. Besides the classification of the token, there are diverse technologies used for the identification component, which are explained in the following subtopic.

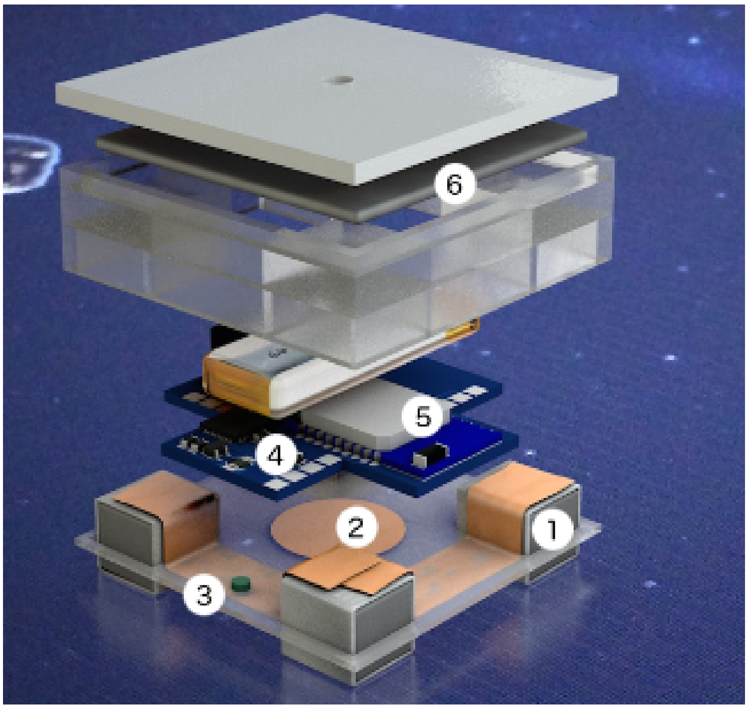

Figure 2.5: Active PERCs Token

The six main components of a PERC tangible are: (1) marker pattern, (2) field sensor, (3) light sensor, (4) micro controller, (5) Bluetooth element, and (6) lead plate.

2.4.1 Optical

In 2007, at the earlier stages of the use of tangibles on multi-touch surfaces, optical techniques were used to recognize the tokens and their patterns, achieving a very precise recognition rate and a large number of recognizable patterns.

In that year, Martin Kaltenbrunner and Ross Bencina developed a study in which they created the ReacTIVision [23], which is a system that uses a camera that is capable of receiving input from the surface of the table.

The tokens used for optical images have fiducial markers, which are objects with a specific image - placed in front of a camera linked to the optical system - that works as a unique pattern and can represent a certain object. Similar to a bar-code reader, the system interprets the black and white image placed in the token, accurately detecting the estimated position and orientation, since it can be scanned in a 3D environment. This feature also allows the fiducial marker to be placed on top of the token, for example, and still be detected. This method has the capability of identifying hundreds of different image patterns.

ReacTIVision is an open-source cross-platform computer-vision framework mainly created to construct table-based tangible user interfaces. This system uses computer vision techniques to detect the token, recognizing its pattern or shape (Figure 2.6). Initially, optical systems primarily used high-contrast pictures composed of black-and-white rectangles to be more easily identified by image recognition techniques [27].

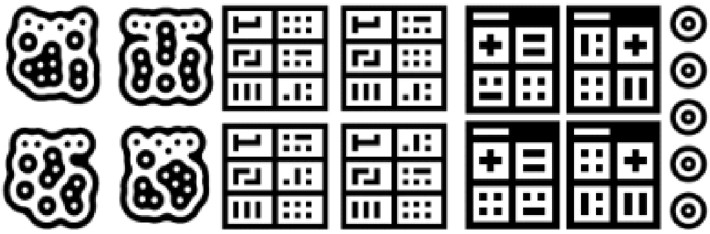

Figure 2.6: Optical Pattern

Through the ReacTIVision system, when a certain object is placed on the table, the table reacts to it, displaying the content it was previously programmed with. However, in this system, the algorithm was written so that the contents it returns after identifying the pattern in the tokens are audios or certain sounds. Once that pattern is properly identified, the programmed instrument would start playing. Moreover, the sound of that instrument could change if the token was moved in certain ways, causing the sound to vary in multiple aspects, such as the velocity of a key, pitch change, frequency, and octaves, amongst others.

This method uses a Tangible User Interface Optical (TUIO) framework, which was designed to satisfy the needs of tabletop tangible users. TUIO was developed as a framework focused on musical interaction, therefore, in order to achieve maximum compatibility with existing musical applications, TUIO allows to send MIDI control messages that can be individually configured. By applying TUIO, it offers a general and versatile communication interface between tangible tabletop controller interfaces and underlying application layers.

Optical patterns require additional cameras to capture images, making optical systems generally unsuitable for mobile or other smaller gadgets devices (Figure 2.7).

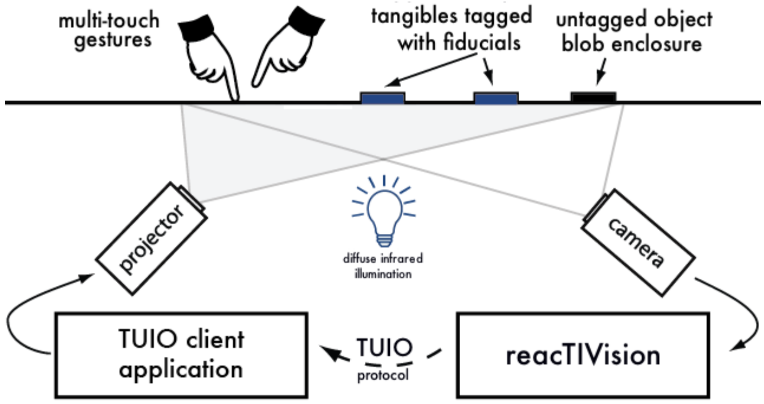

Figure 2.7: ReacTIVision system representation

In 2018, several authors [28] developed the TUIO 2.0 protocol, in which they built an abstraction framework for tangible interactive surfaces. Firstly, this article explores the strengths and shortcomings of the TUIO 1.1 protocol, leading to the construction of an extended abstraction model for tangible interactive surfaces and the specification of the second-generation TUIO 2.0 protocol. In this paper, it is explained that TUIO 1.1, the most used at the time, was a protocol used in optical recognition, as in [23], that fell short in the description of a generalized tangible interaction surface environment. One of the updates it needed, according to these authors of 2018's paper, is the addition of a descriptor of the object geometry, since the Object and Blob profiles remained one of the major limitations for any further protocol extensions. Thus, the TUIO 2.0 is based on an abstraction model designed to describe interactions on, and around, any interactive surface. Moreover, TUIO 1.1 was also missing detailed timing information for gesture analysis, thus justifying, even more, the constitution of an extended model; as such, TUIO 2.0 is meant to approach these limitations, improving the protocol. The version of TUIO 1.1 defines three basic interface components as the main elements of tangible interactive surfaces: Tokens (as objects), Pointers (as cursors), and Bounds (as blobs). On TUIO 2.0, although those three are still core components of the extended model (with a bigger set of attributes), the most recent protocol encapsulates the component's state and attributes, which allowed the creation of a needed semantic description of tangible interface components regardless of the hardware capabilities, allowing the creation of applications that are platform independent. Moreover, to those three core components, it was added an intangible component, a Symbol, and a set of Geometries describe the physical appearance of the object placed on the surface. Furthermore, the new protocol also allowed the encapsulation of an additional component - Controls - as well as the logical and physical description component Associations.

With this updated version of the protocol, it was shown that a big part of tangible platforms and applications can be represented by the components and attributes the TUIO 2.0 allow and, consequently, be encoded within a TUIO 2.0 protocol syntax. As such, it promotes the application of versatile tangible application scenarios.

Thus, the extended abstraction model represents simplified semantics to describe the physical environment abstractly. The model is "hardware independent" and "application invariant", as it gives a normalized representation of the physical interface components. There is an underlying application layer that can build upon the given component abstraction, defining only locally its application model and allowing the installation of the applications on several hardware platforms.

Figure 2.8: TUIO 2.0

2.4.2 Magnetic

In 2013, magnetic tokens were used in a study by a group of authors [25], in which they present GaussBits, a system of passive magnetic tangible designs that allows 3D tangible interactions in the near-surface space of portable displays. When applying a thin magnetic sensor grid attached to the back of the display, the 3D position and partial 3D orientation of the GaussBits can be obtained by the proposed bipolar magnetic field tracking technique. Since the user's hand does not interfere with the magnetic field, users can freely manipulate the GaussBits with their hands or use other non-ferrous tools without causing interferences. The GaussBits had two distinguished token designs: Tiltable Gauss (Figure 2.9), and Rollable Gauss (Figure 2.10). Both types had a slight difference in use and performance; the Tiltable can suffer alterations in the way that it is placed on the surface, and still recognize the object, whilst the Rollable token, as the name suggests, solves the rotation issue of the magnetic token. Last but not least, the GaussBits allow token stacking, which means that, by putting several tokens together, a new frequency is generated, being able to represent a new object.

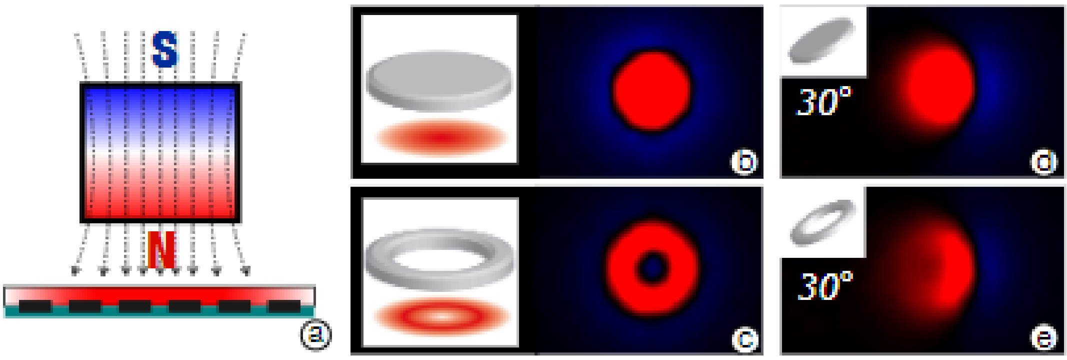

Figure 2.9: Tiltable GaussBits

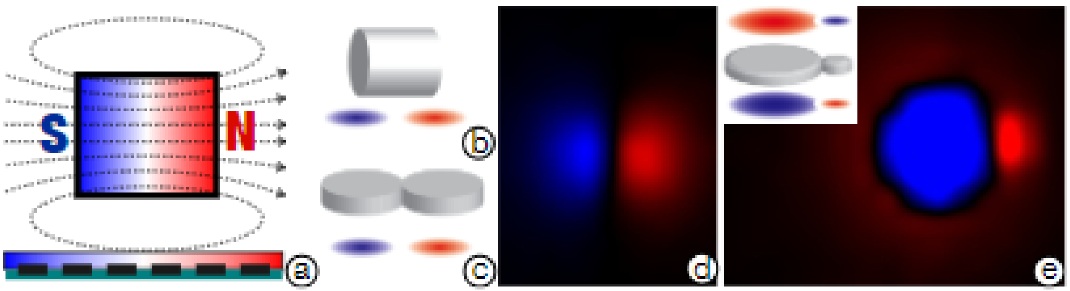

Figure 2.10: Rollable GaussBits

Even with the promising results, GaussBits had a problem with the use of multiple tokens simultaneously. Whether they were stacked or side by side, that proximity could cause interference between them, which can be an advantage when the goal is to get a new frequency for a different object, but a problem when it is supposed to simply represent the object each token is assigned to. Thus, to solve this, the work was later revised.

Hence, the authors developed GaussStones [9], which is a system similar to GaussBits, but with a few improvements, to tackle the frequency issue mentioned above. The authors added a protective layer to the tokens in order to prevent magnetic disturbances between them (Figure 2.11). Consequently, after the implementation of this layer, the magnetic technique became more reliable than its ancestor. Thus, GaussBits and GaussStones represented an improvement in the field of magnetic tokens.

Overall, the design of tangible tokens was always a focus in most of the studies done, since it is believed to be the main factor in the problems that are currently present in this technology.

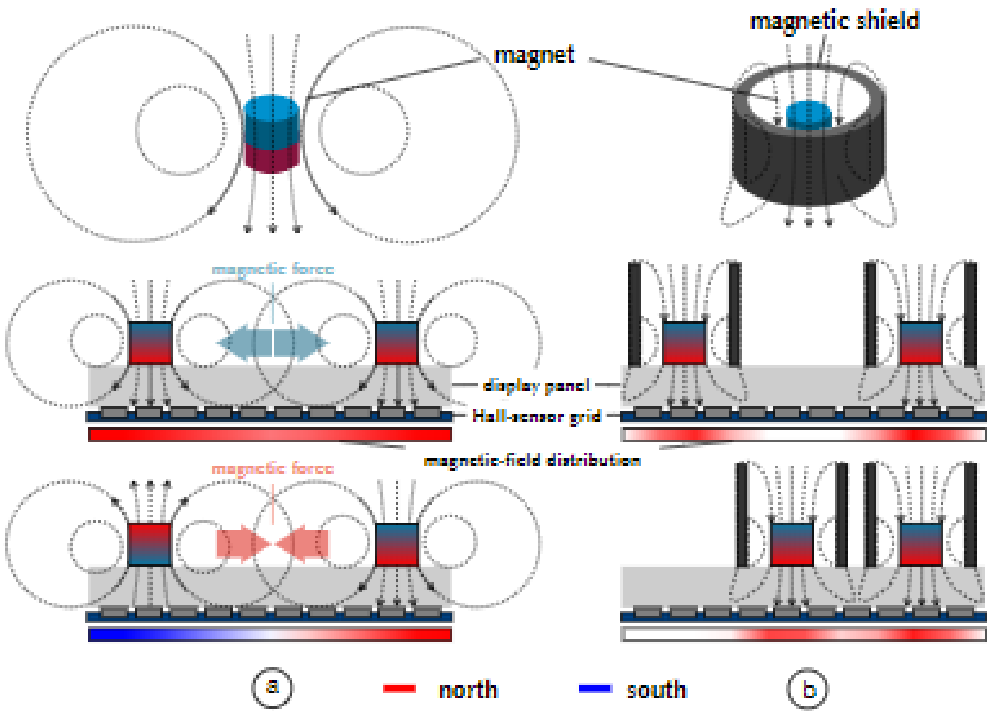

Figure 2.11: GaussStones

In 2016, several authors [29] developed GaussMarbles, a system of spherical magnetic tangibles that exploited the interactions that a spherical object allows with portable physical constraints.

This work explores the design aspects of the token, creating a spherical object containing cylindrical magnets. A ball is a generic tangible that can be used in many different ways, in this case, it was applied to a game, similar to a pinball machine, where the play area is constrained with physical boundaries that aim to enrich the experience of the game by increasing kinaesthetic awareness and providing haptic feedback. The ball movement is tracked by external cameras, positioned above the display that can receive a precise input of the physical movement of the ball. Unlike the previous works ([25], [9]), which were developed for in-hand direct manipulation and ID interaction under strict constraints of movements, GaussMarbles exploits the possible uses of spheres for hand-off manipulation (free movement of the token, for example, rolling) with a set of constraints. Throughout the development, there were guidelines to design the magnetic sphere, so that the developer can easily replicate the results using different-sized screens with sensor grids. This works provides a new perspective for TUI researchers, as they can apply new TUI designs to make virtual ball games more physical.

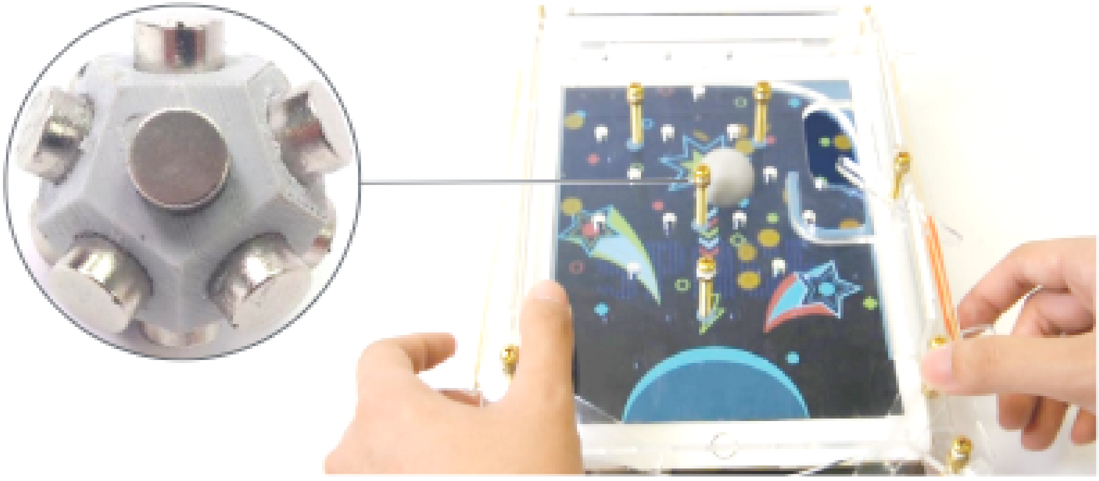

Figure 2.12: GaussMarbles

2.4.3 Capacitive

Within the capacitive technique, there are several studies important to mention due to its innovative or different styles.

In 2011, some authors wrote a study in which they developed the Tangible User Interface Capacitive (TUIC) [30]. TUIC is a system that enables tangible interactions on capacitive multi-touch devices, such as mobiles, tablets, and display tabletops.

It does not require hardware modifications, since TUIC, like other capacitive methods, simulates finger touches, using passive material and active modulation circuits embedded inside the tokens (Figure 2.13).

TUIC uses three approaches for tracking and objects identification: spatial, frequency, and hybrid. The spatial approach, TUIC-2D, uses touchpoints based on distance and spatial methods. In the frequency, TUIC-f, it uses a touch point magnet, such as the magnetic identification method; and, in the hybrid, TUIC hybrid, there is a junction of the spatial and frequency methods.

TUIC-2D tag design is based on the spatial domain. It uses a layout similar to the QR Code. The TUIC-2D tag contains three positioning points which have to be a pre-defined distance at a 90-degree angle so that user touches can be easily distinguished from a token pattern.

TUIC-f tag design is based on a frequency approach. It encodes data in the time domain by simulating finger touches at the same location at various frequencies. The control of these frequencies can dictate if the touch point is active (on) or inactive (off).

The TUIC-Hybrid approach combines the spatial and frequency tags. The spatial touch points provide the object position, and orientation, leaving the token ID to the frequency touch point. TUIC-Hybrid brings tighter the advantages of both methods and enables precise tracking of the movement and rotation.

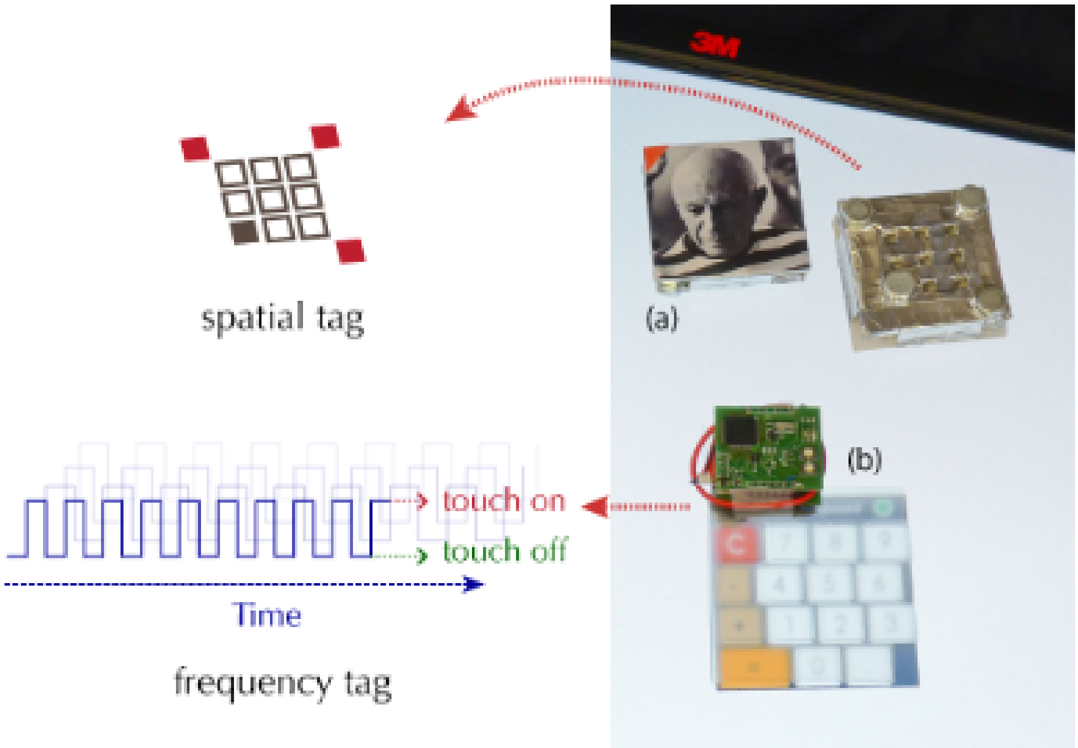

Figure 2.13: TUIC

In 2013, Rachel Blagojevic and Beryl Plimmer developed a study in which they present CapTUI [21]. CapTUI is a system in which the output is a drawing of a geometrical shape. Once again, the tokens have a pattern on a basis that the algorithm identifies and associates with a certain geometrical shape, drawing it as a result. One of the most interesting aspects of this study is the materials used as the tokens and to create the patterns on them; several materials were used, such as aluminum foil, copper, and conductive foam, amongst others, as it is possible to see in Figure 2.14.

CapTUI is a combination of the recognition algorithm and a drawing application, in which there is the recognition of the pattern inserted in the token, which results in an output that is a drawing of a geometrical shape. Moreover, the user can manipulate the token, through slides or rotation, which will affect the output and alter the draws.

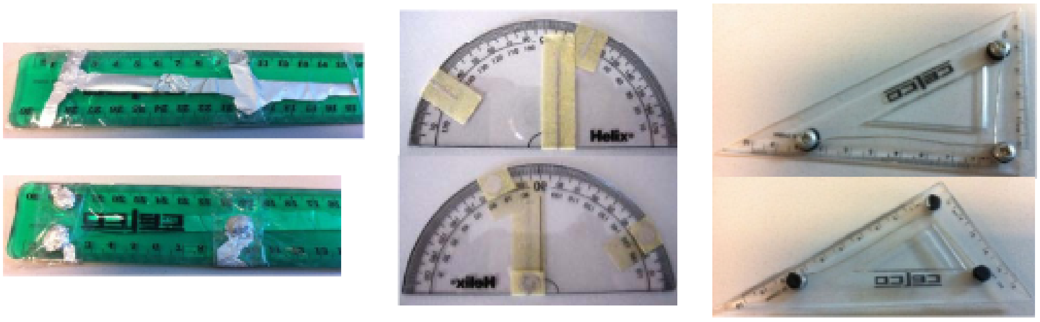

Figure 2.14: CapTUI tokens

In 2016, a few authors developed the TouchTokens [24], which is a system that allows to easily build interfaces that combine tangible and gestural input, by using passive tokens and a multi-touch surface. Whilst in the previous tokens, there typically is a pattern designed on the basis of that token - which would then be identified by the algorithm - with TouchTokens, the tokens have no pattern already inserted. The goal of this study is for the user to be the one that creates the patterns on the tokens through the way he holds them. These tokens have different shapes so that they can constrain the user's grasp and oblige him to hold the token in a different way - thus creating a different pattern. With this technique, the user can perform any kind of gesture based on distinct touch patterns, thereby increasing the richness of input for tactile surfaces. As it is possible to observe in Figure 2.9, the relative spatial configuration of fingers on the surface, theoretically makes it possible to design algorithms that can recognize the resulting touch patterns (Figure 2.15).

The recognition algorithm used in TouchTokens had a 95% accuracy recognition rate. The pattern resulting from the user's grasp on the token has 3 touch points, allowing the construction of triangular patterns. Gestures are used to interact with the token, and multi-touch screens allow other diverse interactions between user, token, and machine.

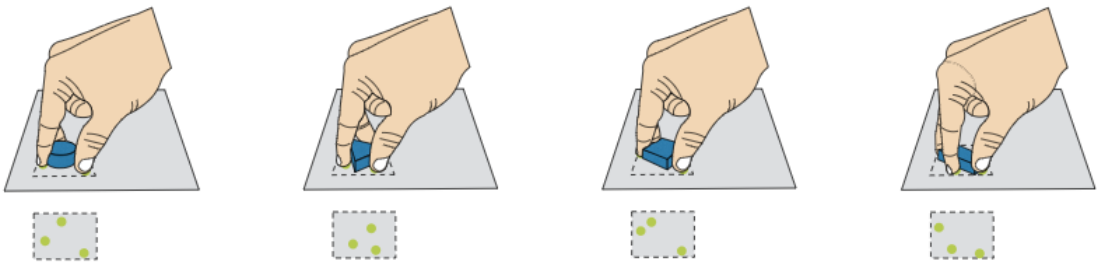

Figure 2.15: TouchTokens

In 2020, several authors developed the Poisson Triangular Pattern Generator (PTPG) [31]. PTPG is an approach that, instead of re-configuring the token's design, focuses on an efficient algorithm and a random pattern generator, which generates multiple patterns that the algorithm is capable of identifying. Some random-generated patterns worked more efficiently than others, which were still capable of achieving remarkable results, extending the number of patterns with a 100% efficiency recognition rate, from 17 to 261.

When an object with a triangular pattern is placed on a tangible surface, the TUI system senses the markers of the object and provides marker positions to the recognition algorithm. The way that PTPG recognizes their patterns, is through sorting the vertices and then calculating the lengths of the triangle sides. Once the sides are calculated, they have obtained the vector f as follows: f=(v1-v2,v2-v3,v3-v1) For a known set of pattern templates, they've used an equation to calculate the distance between f and each template, and the ID of the template that has the smallest distance is obtained as the recognition result.

This algorithm design is capable of expressing identification, location, and direction, and it only requires a small number of touch points. Moreover, it can be applied to several sensing technologies and active and passive tokens. This algorithm is easy to implement, robust and fast, thus, making it one of the most reliable methods in the current technology in this field.

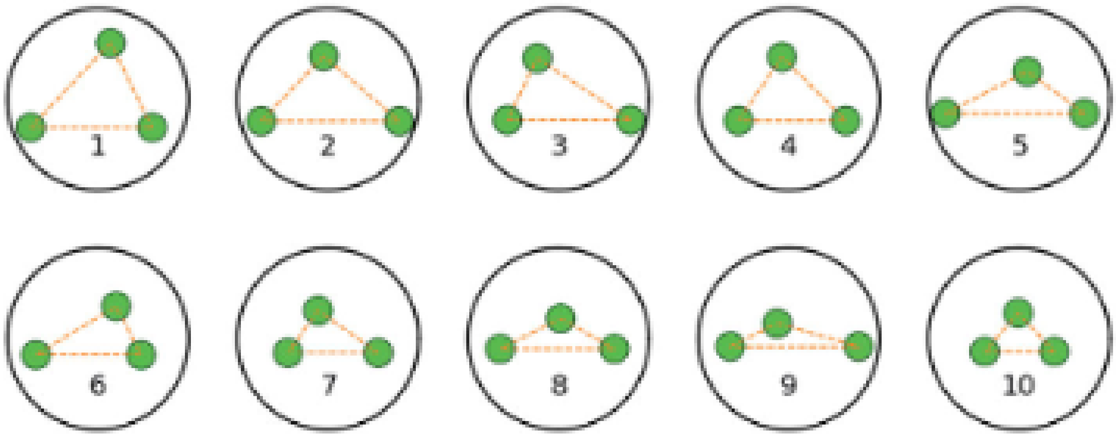

Figure 2.16: PTPG generated patterns

2.5 Discussion

To tackle the technical limitation of the multi-touch screen capacity, several studies focus on the algorithm, the physical body of the token, and diverse techniques of pattern recognition. Tokens can be recognized through different techniques, each one with a unique method. Some of them can be more expensive but more accurate, as previously discussed. These are great tools to use, but they are not very accessible for the majority, since some use more resources, like optical works [23], and can require displays capable of in-grid detection [29]. Thus, even though they have a great outcome in accuracy and performance, there are cheaper methods that can compete with the technical aspect of these [24], by using passive tokens with pattern recognition.

A simple token can contain a unique pattern, normally composed of three to four touch contact points, to better suit the multi-touch screen touch limits, that typically support around ten to twenty simultaneous touches.

Moreover, throughout the analyzed studies, it is possible to see that different shapes of tokens were used, in order to improve the flow of the token, as well as to make the interaction between user and machine as seamless as possible. Additionally, the materials used to build the token also differed in some methods, in pursuit of higher accuracy of the detection rate.

In the following table (Figure 5.2), it was gathered some of the common properties as data, such as the recognition rate, which is a percentage of the success rate in detecting the tokens and identifying the unique object. There is also the number of patterns, which account for the possible different patterns the system can detect, as well as the number of markers, which represent the maximum number of tags, touchpoints, images, etc., used in the token's construction. Moreover, the token size regards the size of the constructed tokens and the token type determines the identification technique (optical, magnetic, capacitive) and classifies if the token is passive or active.

Work	Recognition Rate	Number of Patterns	Number of Markers	Token Size(cm)	Token Type
TUIC-2D	-	511	12	5	capacitive, passive
TouchTokens	>95%	6	3	3 - 5	capacitive, passive
PERCs	100%	unlimited	3	4	capacitive, active
GaussBits	-	128	4	3.8	magnetic, passive
GaussStones	-	6 - 2025	2 - 4	1.5	magnetic, passive
ARTag	95%	2002	dozens	>2	optical, passive
reacTIVision	95%	90 - 300	dozens	2 - 6	optical, passive
PTPG	96.5%	42 - 443	3	4 - 8	capacitive, passive
PTPG (Optimized)	100%	17 - 261	3	4 - 8	capacitive, passive

Table 2.1: Results and description of characteristics and components of previous works

However, multi-touch screens constrain the use of pattern recognition technology due to its limitation in the number of simultaneous touches. In order to alleviate that restriction, there needs to be a solution that can allow a bigger usage of unique patterns with multiple touch points, maintaining a high recognition rate, whilst keeping the interface accessible and easy to use, in order to increase the usage of it.

Finally, it is possible to know that there are several approaches to this problem. The choice for this project depends on the goal it intends to reach; thus, two main factors are cost and accessibility, since the solution should be cheap and accessible. With these criteria, it is possible to exclude some of the techniques previously mentioned, since they can be hard to use, expensive, and inaccessible to the customer and producer.

To conclude, the path followed in this dissertation will be similar to previous works with capacitive methods that use passive tokens, since these showed to be more accessible, and display potential for improvement, both in the recognition rate and the maximization of the usage of the number of markers that the display allows.

Chapter 3

Proposed Algorithm and System Requirements

In this chapter, it will be explored possible paths to follow, based on the needs required to achieve the established goals for this work. To answer some of the research questions, it is required that an algorithm capable of detecting triangular patterns is made, it has to be efficient, reliable, and allow its application to different types of hardware. Moreover, the algorithm needs to increase the detected triangular patterns, by extending them to a multi-point triangle-based pattern, through the use of the maximum hardware touch point capacity. Finally, as an extra, the algorithm has to be capable of running in browser-based applications, since it will open up the accessibility, and application potential. Strongly based on the previous research (chapter 2), it is possible to determine the several solutions that our research can have.

3.1 Extending Triangular Patterns

To solve the challenge that is on the basis of this dissertation, previous studies have tried to generate a unique pattern with a low number of touch points. In addition, some used different tags for identification purposes and to accurately determine the position of the object [30]. Moreover, as another example of studies that aimed to create a unique pattern with fewer touch points, there was also the random triangular pattern generator [31].

Firstly, to detect the triangular patterns it will be used the capacitive methods [24], [32], which are reliable and accessible with great results as shown in Table 5.2.

Then, to further explore these methods, one way to create a higher number of patterns is by using the full capacity of the touch point capability of the interactive surface. By exploring the screen's touch point capacity, it is possible to use the remaining available touches (since the triangular pattern only requires three of these touch points) and use them to augment the current triangular pattern. Recent surfaces with improved technology are capable of supporting more simultaneous touch points, having, consequently, a wider range of unique patterns, based on touch capacity.

However, it is necessary that the location of these points are part of the pattern so they do not get confused with user touches since it is desirable that the user can use a finger touch to interact with any surface instead of exclusively using tokens to interact.

Furthermore, to extend the triangular patterns, there will be additional touch points in the pattern. Although the geometrical shape that determines a pattern will always be a triangle, these triangles will have additional touch points inside their shape. As such the program can detect the triangle as intended and as it would a normal triangular pattern, and then determine if that triangle has any existing active touch points inside it, thus making the unique triangle extendable by the number of touch points capable of fitting in the token and/or pattern, based on the surface touch point capacity.

Essentially, this dissertation intends to explore the number of touch points for the construction of unique patterns and extend these by using the same token. By increasing the amount of touch points on the token's base, it is intended to expand and extend the traditional triangular patterns. This will increase the number of unique patterns resulting from one triangular shape, due to the possibility of adding different interior points to that triangle.

3.2 Approach

There are several possible techniques and materials to use, thus, different approaches could be taken. The proposed solution relates to extending the traditional triangular patterns by adding interior points to them, which means, there will be more touch points in these patterns and that three touch points are the minimum required to construct a pattern (Figure 3.1).

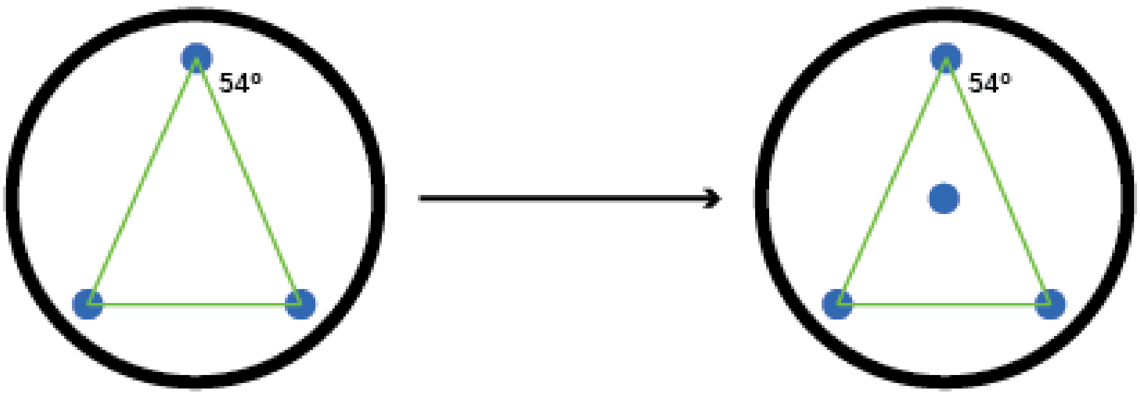

Figure 3.1: Representation of proposed pattern extension method

As expected, this goal brings some challenges, such as distinguishing the pattern touch points and the user touch points. Moreover, the triangle construction might suffer interference with the added touch points, since there is a chance for the algorithm to detect an undesired pattern, with the incorrect triangle. Additionally, the more touch points it is used in the pattern, the higher the chances of existing touch point occlusion; thus, an additional limitation is that the pattern touch points must have a minimum distance between themselves (Figure 3.2).

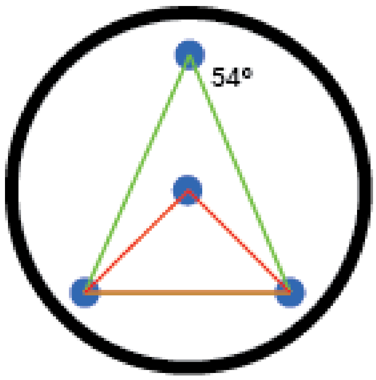

Figure 3.2: Representation of detected triangles

To reach a consistent result and have a strong recognition algorithm the method uses isosceles triangular patterns, since due to its particular characteristics - having two equal sides and, consequently, one unique angle - these triangles can be represented by its differentiating angle, making it a unique representation of a certain item and easier to be identified by the algorithm. Moreover, the use of isosceles triangular patterns also prevents the algorithm from considering a possible unwanted pattern the user could create by placing three finger touches on the surface. As such, this allows to eliminate the recognition of any triangle, and thus have a more accurate detection of the correct pattern.

Additionally, the touch point occlusion can be prevented by spacing the touches in the pattern; since this requires a minimum distance between them, this might mean that the token size has to be big enough to fit several touchpoints, preventing the usage of small tokens, so that there is enough space for our touch points to be spaced and correctly identified (Figure 3.3).

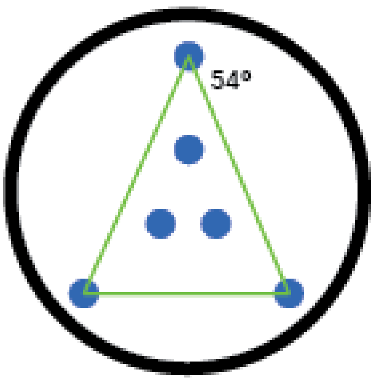

Figure 3.3: Representation of extended triangular pattern

However, by adding interior points to that isosceles triangle, it increases the chances of the algorithm detecting another smaller isosceles triangle amongst the interior points - which is not what is intended. To prevent this from happening, the algorithm was written in a way that, if it detects another isosceles triangle on the pattern on the token's base, it must consider as the desired pattern the isosceles triangle with the bigger area (Figure 3.4).

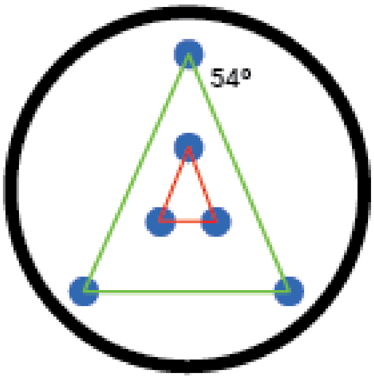

Figure 3.4: Representation of undesired patterns

To conclude, as previously mentioned, to the interior of these isosceles triangles, it will be added other touch points, which will preserve the triangle's unique characteristics and allow the creation of a different unique pattern and, consequently, increasing the possibilities of interactions. Thus, each different isosceles triangle can be expanded to several unique patterns, which also allows better use of the touch point capacity of the hardware. Finally, with the approach established, it is possible to start developing the potential architecture of the application and the algorithm.

3.3 Proposed Technique

To fulfill the purpose of this dissertation, which is to extend the traditional triangle patterns through the use of touchpoints, a new pattern recognition algorithm will be developed.

Afterwards, the algorithm will be submitted to testing in order to guide improvements and adapt whatever is necessary to reach better results. The final algorithm must have a good performance, and good recognition rate and enable the increased usage of the touch point capacity.

Firstly, it will be defined the method and identification technique that better fits the chosen parameters for the algorithm, which are: low-cost, efficient, accurate, accessible, and capable of recognizing the token and user touch points. Additionally, in order to be accessible, it should be capable of running the most current multi-touch devices that tend to use capacitive screens, such as smartphones and tablets.

However, in the case of bigger display surfaces, the type of screen is typically different and uses methods such as Infrared or InGlass; nonetheless, if the program runs on a capacitive screen, it will run on other ones as well. It may present a slight difference in performance, but, as long as it does not require Optical or Magnetic Identification methods, the program and algorithm will work.

Moreover, to further increase accessibility, besides functioning in the common multi-touch devices, it must be capable of running on slower machines (with a small computational capacity). Thus, it will be used as a capacitive passive token, since they require fewer resources to function; also, it might be the cheapest combination of multi-touch surface and token. In addition, the usage of the capacitive method allows the unique rotation characteristic of the token, which adds an extra interaction possibility.

Although an active token works better, from previous works [19], it was possible to conclude that it is a very expensive method and not easily accessible, therefore it does not meet a few requirements for the algorithm desired.

After establishing the basic needs and requirements of the algorithm, it is possible to advance to an early stage of testing, which might lead to obstacles and require further adaptation of the algorithm.

Regarding the graphic interface, it used a simple and minimalist style so that users are compelled to use tokens. Since the objective is to increase or give the user a new fulfilling experience, the interactions must use the token as the "remote" to all of it. Additionally, rotation of the token can be used to select different submenus, and the drag can be used as an interactive element between the user, machine, and content display. Also, token comparison and interaction are possible as long as there are enough available touch points on the screen. Essentially, the creative possibilities are endless alongside the new interactions that this technology allows.

3.4 Software and Hardware requirements

To implement the chosen technique, certain tools are required to build the previously presented components. Once again, these tools can vary, since there are multiple possibilities that can achieve the wanted results. Therefore, to justify the usage of the chosen tools, it is necessary to establish the software and hardware requirements and their limitations.

- **Hardware** — Regarding the hardware, it must have a multi-touch screen, which can vary in size and can use different technologies for surface touch recognition, as explored in (chapter 2). Interactive displays, such as tablets and smartphones, are regularly used in our society, therefore fulfilling this multi-touch screen requirement, should be easy. Moreover, regarding the hardware capacity of the algorithm, at least, it must be in the range of common devices (such as smartphones) with lower performance systems.

- **Software** — The program has the purpose of detecting, from an active set of random touch points, made by the user or a token, and recognizing triangular patterns, as well as being capable of identifying extended patterns of these detected triangles. To fulfill the desired requirements for the proper functionalities, the program needs to be capable of receiving and interpreting touch-point-based input, which can be found in most browsers, and used for proper functionality on mobile devices, or any other multi-touch devices.

- **Development Language** — JavaScript is an accessible, friendly language that is web-based. This language can be used for several purposes, such as website design, interactions, and backend development. It is a very versatile language, and it is possible to use it to fulfill the majority of the requirements of this work. There is a significant amount of documentation on JavaScript language, and constant development of unique libraries and frameworks, each one focused on specific functionality, for example, animation, graphic design, and website structures.

- **Ease-of-use** — To provide an enjoyable experience for the user, the software and hardware must be in sync. For an ease-of-use, the interaction will be dictated by the token, through simple sets of moves, drag, and rotation. Thus, the token needs to fit comfortably in the user's grasp, have a distributed weight, not be too heavy or too light, and finally, it needs to move smoothly. The interactions should be simple and explicit; complex actions with the token might cause confusion on how to use the program.

With the requirements identified, it is easier to know what are the possibilities and limitations, which will make it easier to plan the implementation of the algorithm and deal with future obstacles. Additionally, it already guides the development of both the front-end and back-end.

3.5 Summary

To resume, it will be developed an application that recognizes unique patterns in triangular shapes with added touch points, distinguishing each active touch point correctly. To achieve this, it will be explored an algorithm based on spatial, distance, and area features. This algorithm will need to recognize unique triangular shapes with strict construction rules so that the recognition is efficient and does not get confused with similar geometrical shapes with three touch points. Thus, it is necessary that the algorithm identifies the active touch points, and sorts and distinguishes them to try to detect a triangle shape according to our specific instructions of it being isosceles. Once the triangle is constructed, the algorithm has to analyze the remaining active touch points and comprehend which ones are inside of our triangle pattern and which ones are out. In other words, the algorithm will identify the user touches, the triangle vertices, and the extended touch points, which are the ones inside the triangle. Last but not least, the algorithm will create a matching system that will compare the current active pattern and touchpoints, with our pre-configured system of triangles. Moreover, by determining the axis of the triangle, it will be possible to use the token's rotation functionality and have an accurate identification of its direction. Finally, these functionalities of the algorithm will be evaluated through testing on the main forms of interactive screens, smartphones, tablets, and tabletops, which will allow the evaluation of the efficiency of the algorithm (its recognition rate), and determine which patterns perform best and why.

Chapter 4

Algorithm Development and Module Implementations

This chapter will describe the implementations made for the program to achieve the desired functionalities, some obstacles faced during the development, and how everything was combined. To create an efficient program capable of fulfilling the required tasks, firstly it is necessary to deconstruct the complexity of the actions required, and how this can be affected by the current technology. Moreover, this chapter will be described the creation of the token prototype, according to the specified requirements, and how they work. Additionally, it will be explored the implementation of the algorithm: what functions is it able to do and what are the software requirements, alongside a brief description of how it was built and further justification of the choices made.

4.1 Hardware

To create a capable program, firstly, it is necessary to understand the hardware to which it will be applied. In previous chapters, several approaches to object and pattern recognition techniques were discussed, in which the hardware components differed from one to another and the software reflected a slightly different approach. For this project, the approach must match the main requirements: low hardware capabilities, and an accessible product that does not need help from external hardware components.

Regarding the touchscreen, each one uses different technology to receive the input presented to them; thus, each one has different strengths and weaknesses that make it a better choice for certain applications.

The first steps of the program development were tested on a capacitive screen of a smartphone, thus helping in the algorithm design and debugging. In a later stage, when the algorithm was ready to be executed, it was tested on an InGlass tabletop surface. With this surface, it was possible to retrieve further data for the final tests, which are presented in (chapter 5).

4.1.1 Interactive Displays and Multi-touch

Once the different types of hardware are explored (chapter 2), it is possible to choose the type of screen that better fits the goals of this project, so that the software created can function well on that screen and, hopefully, work in other types of screen as well.

Afterwards, it is time to start programming the software with the method (amongst magnetic, capacitive, or optical) that is most appropriate for us. Thus, the focus is to create functional software for capacitive and InGlass screens, since they use similar ways to calculate touch points' position.

As mentioned before, the hardware dictates how the software works, as different technologies interact with the program in different ways. Firstly, multi-touch is a requisite for the program, otherwise, it would not work the way it is intended to, since the pattern recognition would not work. Thus, since the goal is a multi-point pattern and is required a screen, an essential device is a multi-touch screen. As for the displays, capacitive screens allow multi-touch functionality and can identify every touch accurately; moreover, these are the current, most used screens in consumer products.

Regarding the InGlass technology, it is relatively recent and can also accurately detect any obstruction on the display. The main difference between capacitive and InGlass is the way they detect touches as previously explained. Additionally, the InGlass grid can cause touch point occlusion if these are located in a very small space but also works better when we want to exclude the human contact factor for the proper work of the tokens and program.

As for the algorithm, it was designed to work on all surfaces that use these technologies, since it bases itself on spatial operations.

4.2 Token Prototype

The token construction is as important as any other element of this program since the way it is built will also influence the final results.

For a starter token, bottle caps were covered in aluminium, a conductor material that will allow this token to work on any capacitive screen. These first tokens were small (5 cm in diameter). Although the size restricts the usage of touch points, it showed to be a very adequate size for mobile, since it was capable of receiving the intended touch points and fitted the mobile screen.

With the earlier prototype of the token made, it was possible to ensure some better decision-making in the construction of a bigger, more suitable, tabletop token. A block of wood (9 cm in diameter) was used to make the main structure frame, which was covered in aluminium foil and added aluminium touch points on its base to create the pattern. The size of the touch points has a big weight in the recognition accuracy; they must be as small as possible so that the input is clear and precise. They must also be placed accurately in their base, since the patterns were already written in the algorithm and the slightest displacement on the token's base could alter the unique pattern the algorithm was programmed to detect, thus not recognizing it or recognizing the wrong triangle by mistaking it with a different token, and thus a unique object.

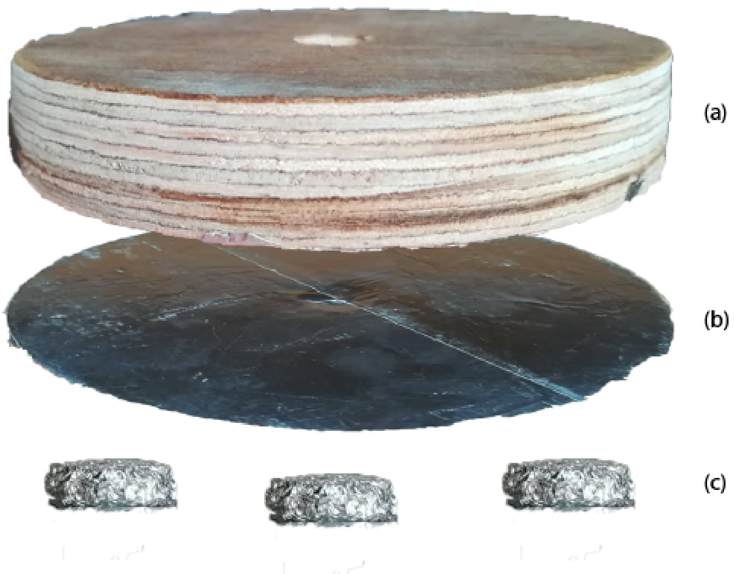

Figure 4.1: Token Build

(a) wood base, (b) aluminium foil, (c) aluminium touch points.

Figure 4.2: Assembled Token (small dimensions)

Figure 4.3: Assembled Token (big dimensions)

4.3 Software

For the implementation of our program, as previously discussed, the software must interpret the physical touch points the hardware used receives. Without the multi-touch function or touch-input data, it is not possible to build the algorithm the way it is intended since the premise of this work is, firstly, to detect patterns on touch point data received, to further extend these same patterns.

Thus, the program needs to recognize patterns defined by touchpoints, which can be placed anywhere on the screen; therefore it requires the algorithm to assimilate the information given and, through strict rules, identify all possible triangular patterns. The rules are based on and worked around the triangle characteristics, more specifically, isosceles triangles, which will make it easier for the algorithm to distinguish pattern touch points from the user touch points (even if the user places three touch points close to each other).

In the next step, the algorithm is required to extend the identified triangular patterns through the addition of the remaining interior touch points. In this part, it is explained how this extension is achieved, with the help of rules regarding the area.

Finally, when the entire pattern recognition is done and there is a positive match between the detected pattern and the programmed patterns, it is possible to determine the orientation of the token through their axis. This feature will allow the rotation interaction of the system, thus using that information in an interactive way or application.

To sum up, the program needs to recognize patterns defined by touch points, sort them, identify them, distinguish the pattern touches from user touches, match the found pattern with the patterns written in the system and then return the associated output or object.

4.3.1 Program Implementation and Modules

4.3.1.1 Touch Points and 2D Vectors

Given a set of active coordinates, first of all, it is necessary to process the input data, which are the touch points, made by the user or the tokens. Every active touch point will be stored in an array where the touch point will be converted to an X/Y coordinate, to simplify working with touch points.

Given the Cartesian coordinates (X/Y) of our points, these will tell the software where the touch point was executed, with the precise location on the screen, according to its specific position. Thus, the touch points are the trigger to begin executing the program (Figure 4.4).

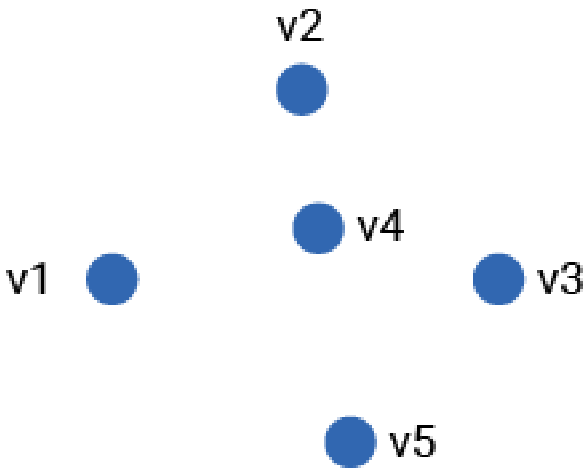

Figure 4.4: Set of Active Touch Points

Then, to this active set of data, it is applied the first constriction for the pattern recognition, which is based on the distances between the points. Since our triangular patterns regard only isosceles triangles, two sides of the triangles have to be equal in distance.

Moreover, when writing the program, minimum and maximum distances were established according to the dimension of the token, so that the algorithm does not consider possible sets of three-finger touches from the users. Then, considering these pre-programmed distances, the length between active points will be analyzed, to identify the combinations that constitute an isosceles triangle. In other words, the v1 will be connected to every other active point (v2, v3, v4, v5), and the algorithm will only consider the lengths that fit the minimum and maximum established distances (Figure 4.5).

Afterwards, from the sets of points that do form a side with a distance within the limits, the algorithm will check if there exists another side, constituted by at least one different point from the previous one, that has the same distance to form the isosceles triangle. Thus, these sides with equal lengths will be considered as a possible triangle and stored in another array.

As such, we get closer to determining if possible triangles are present in our active set of touch points.

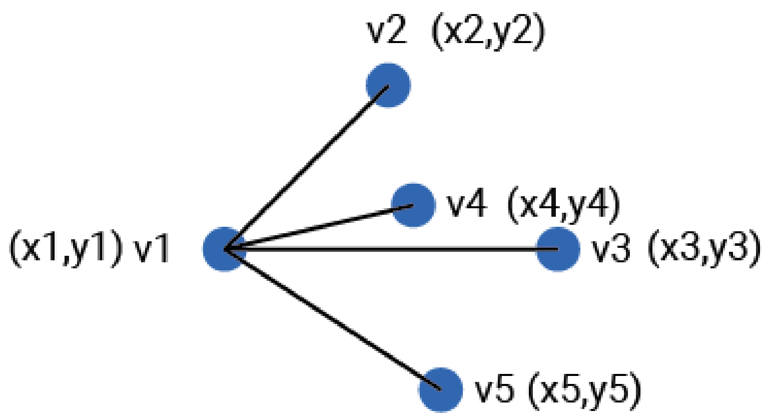

Figure 4.5: Setting Lines Between Touch Points According to Distances

4.3.1.2 Triangle

Once the program has the active touch points that might constitute one of our unique triangle patterns, it is possible to create a triangle through a constructor, which creates an object on the program. Then, it is also necessary to have three points; for this, the program fetches the already paired ones and builds all the possible outcomes of triangles with the missing point. To build the triangle, there are some conditions written in the algorithm that must be validated, as well as the established rules of the isosceles triangle.

Regarding which conditions to write in the code, it was necessary to have some to increase the precision and avoid constructing and detecting non-desirable shape objects, such as three-point shapes made by the user while interacting with our system. Thus, a specific set of isosceles triangles will be used. Isosceles triangles have one unique angle - defined as the apex angle. In the following steps, this is the angle that will determine if the constructed triangle matches one of those implemented in the system, allowing each unique angled triangle to be associated with the desired object.

Isosceles triangles also have two equal sides, length-wise, and, consequentially, the opposing angles are also equal. These characteristics ensure that it is possible to implement enough restrictions to not construct non-desired triangles.

However, the matter regarding having two sides with equal length was already tackled in the previous subsection, so, by this point, the points that constitute the sides of the triangle are already selected.

If our sides have common points we can assure that there is an isosceles triangle present, on the set of active points. To better explain this, if the side point (a) to point (b) has the same length as point (c) to point (b), we have a triangular 2D geometrical shape constituted by three touch points, with two equal sides and one unique length side (Figure 4.6).

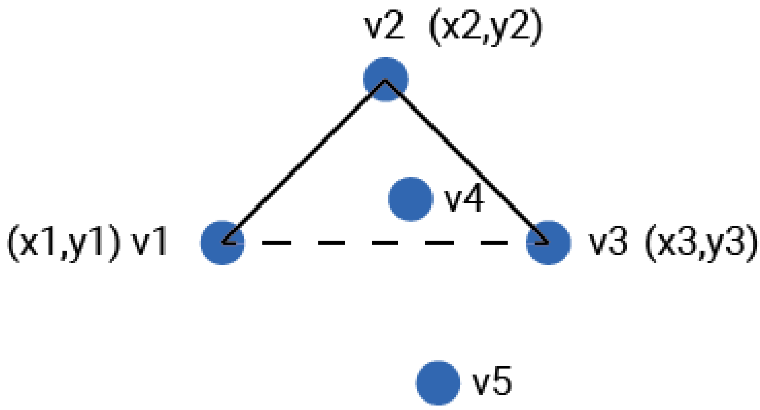

Figure 4.6: Triangle Construction

4.3.1.3 Recognizer

The main goal of this dissertation is to recognize different patterns according to the touch points, thus enabling the possible extension of triangular patterns. Therefore, it is necessary to apply several conditions so that the program can distinguish what points are part of the extension of the triangular patterns and the ones that do not belong to the extension. Moreover, in order for the program to be able to recognize the patterns, it is first necessary that the pre-determined triangles that will be used to constitute the patterns are implemented in the algorithm. In this case, the following eight triangles with a certain unique angle were implemented: 18º, 36º, 54º, 72º, 90º, 108º, 126º, and 144º (Figure 4.7).

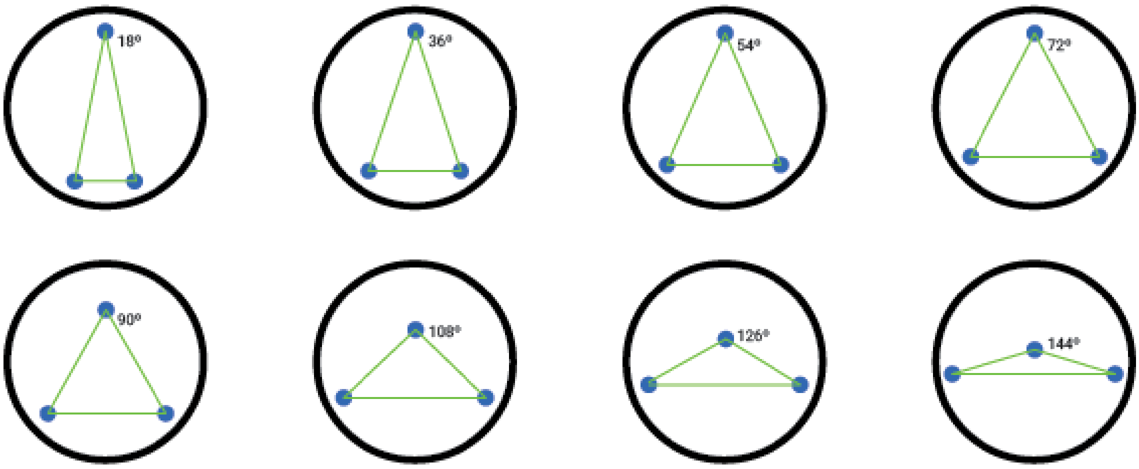

Figure 4.7: Unique Triangle Patterns

Then, the recognition mechanism needs to check some of the predetermined matches, which include the previously discussed conditions of the triangle, such as the apex angle, which will then be used to identify which triangle it corresponds to (each pre-determined triangle has an identification number, ranging from 1 to 8), as well as the inside points.

The inside points are recognized through the area; firstly, the program will identify in the pattern one of the eight pre-determined triangles through the apex angle. Then, a candidate point will be added to that triangle, leaving the pattern with four touch points instead of three. Consequently, it is now possible to form three new triangles with the different combinations of those four points. The next step, to verify if that candidate point belongs to the initial isosceles triangle pattern, is to calculate the individual areas of those new three triangles. After having the three areas, those values are summed in order to compare them to the area of the initial isosceles triangle. If the three summed areas are higher than the area of the initial isosceles triangle, then the candidate point does not belong to that triangle. Otherwise, if the sum of the area of the three new triangles is lower than the area of the main isosceles triangle, it is possible to conclude that that candidate point does belong to that triangle, making it an interior point (Figure 4.8). Since the token itself is a physical object, no user touches can occur in the area that it occupies on the screen, thus making this method a reliable way to achieve the goals of this thesis.

Figure 4.8: Touch Point Detection for Pattern Extension

In the final stage of our recognition, the program requires a match from the detected triangle, detected angle, and detected inside points. This match will make it easy for the detected pattern to be associated with one of our already programmed unique patterns, according to the end results. Once the recognition is complete and there is a successful output, it is only necessary to associate the desired display of information/content to the right pattern/object (Figure 4.9).

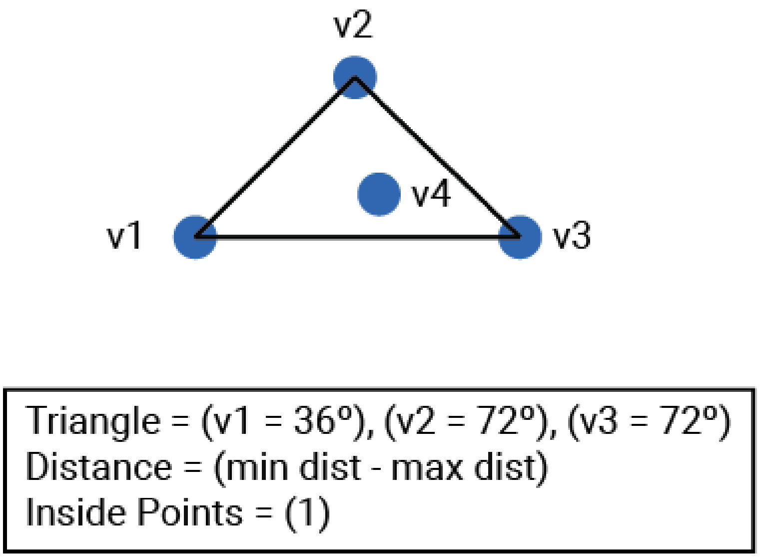

Figure 4.9: Matching of Detected Triangle Pattern

The recognition rate will determine the success of the program; moreover, it is also important that the program runs smoothly so it can be used in several types of hardware.

4.3.2 Development Environment

Throughout the initial phases of the development, all testing was made on a capacitive screen of a smartphone, for a couple of reasons. Firstly, the small screen provides a more restricted area to work with since the algorithm functions, not exclusively, through the area, and distance; therefore, if the algorithm functions on a small screen, it will most likely work on larger display screens, as long the screen resolution is similar, and, if not, the token size and respective touch points need to be calibrated to fit the new display surface.

Secondly, the capacitive screen, currently, is the most used technology on multi-touch surfaces, and, as such, our algorithm is capable of working on a large market of the consumer/producer.

Then, when the algorithm was operational and working properly on the capacitive screen, the testing phase was initiated on a large display screen with InGlass technology. As expected, the algorithm worked after a few tweaks on the distance thresholds and the token construction, touch point-wise.

Some important notes are that the screens work in different ways, even though the algorithm worked on both of them. With the InGlass screen, there were more difficulties in recognizing the object associated with the smaller-area triangles with a larger number of touch points.

4.4 Summary

In the initial phase, the main focus was to determine the path to take and how to approach the several functionalities that the program needed to be based on hardware and software requirements.

To sum up the core of this work, the following images guide the process of the algorithm creation, with the required steps and functionalities. The current program (Figure 4.10) has several functionalities:

- **I**t is capable of receiving a set of active points and analyzing them.
- **G**iven a set of active points, it is capable of detecting unique isosceles triangular patterns.
- **I**dentifies the unique isosceles triangle patterns through its differentiating angle.
- **R**ecognizes extensions of the same triangular patterns.
- **F**or the extension of the patterns, it recognizes candidate points, and identifies them as an inside point of the triangle or not.
- **C**an be applied to a touch screen of any hardware device with multi-touch technology.
- **C**an be used in several areas for different purposes: learning, playing, etc.
- **E**nhances the user experience with the token interaction.
- **A**llows rotation, movement, and drag interactions.

During the development, there were several challenges that appeared during the development, mainly, the extension of the triangular patterns. On the other hand, the construction of the pattern was relatively straightforward, connecting the dots when the conditions were met. For this part, adding the rule of the triangles having to be isosceles was crucial, otherwise, every kind of triangle presented would trigger the recognition pattern and the process would be much slower since very few would actually be the desired triangles.

Finally, once the whole recognition process is done, the algorithm checks for matches; if all of the parameters of the matching are checked, it recognizes the object.

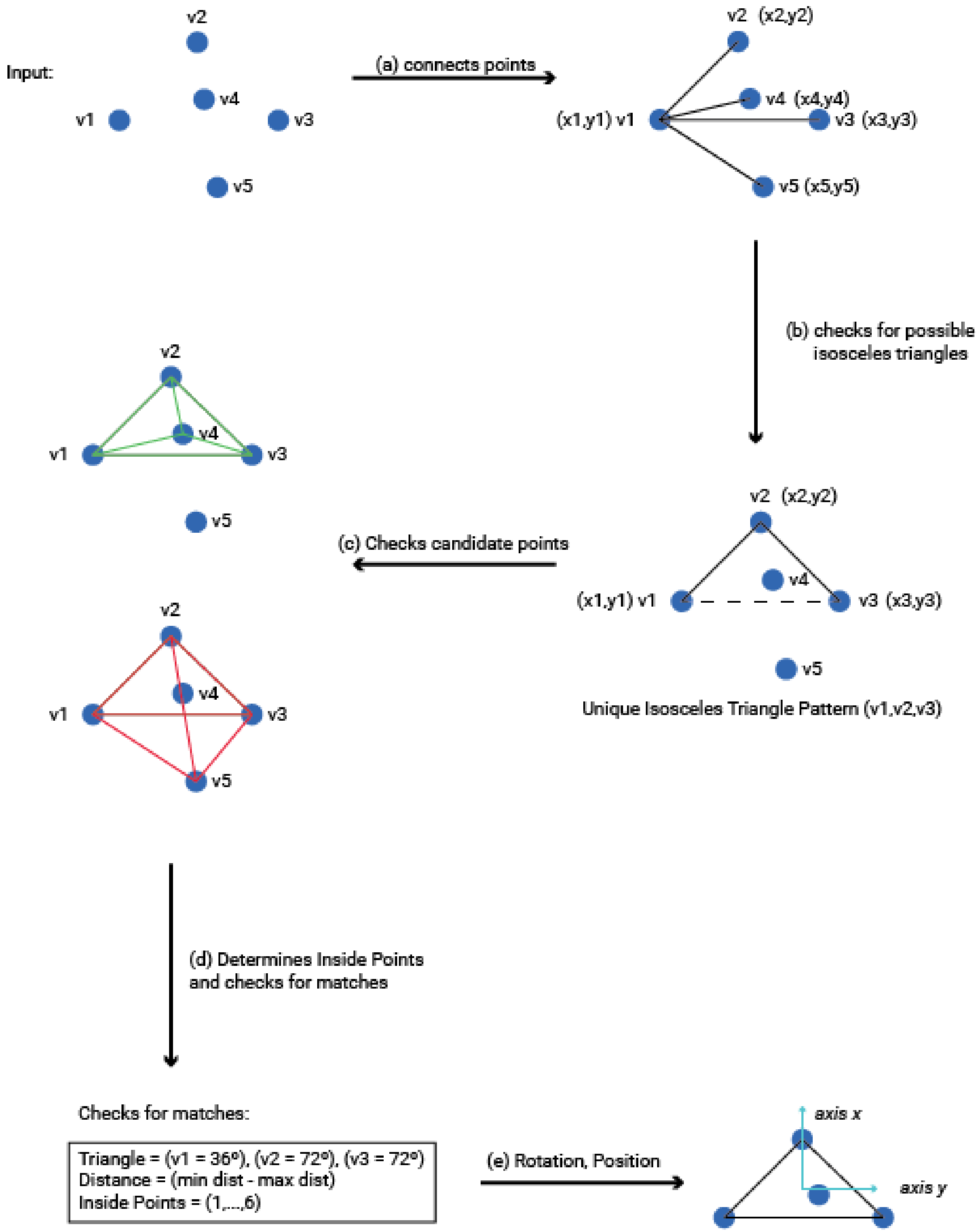

Figure 4.10: Algorithm Explanation

(a) Creates possible lines, (b) Calculates distance between points according to our distance threshold, (c) Checks for candidate points that can be part of the pattern, (d) Determines Inside Points and checks for Matches, (e) Calculates orientation and position of the matched triangle

To conclude, it was created a functional algorithm capable of extending the original triangular patterns with three points to multi-point triangular patterns (see Figure 4.11). The algorithm will be tested in the following Chapter (chapter 5) to retrieve additional performance data and study where it can be improved and what could be made different to reflect better results if necessary.

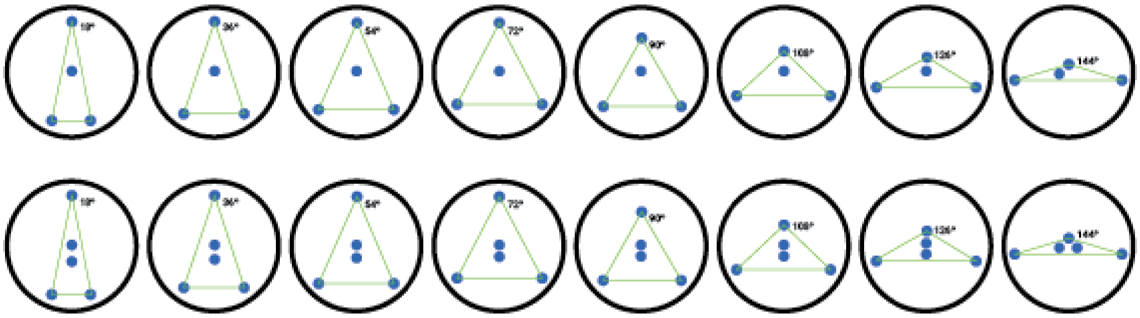

Figure 4.11: Extended Triangle Patterns

Chapter 5

Tests and Results

To test the algorithm, the program was subjected to stress tests, which allow the computation of the speed of the algorithm (which represents the time it takes for the program to run and identify the object through pattern recognition) and the recognition rate, which quantifies the precision with which the algorithm can recognize the pattern and properly identify the object. Moreover, the basic interactions were also tested, such as token rotation and drag.

Regarding the type of display surface used for these tests, it was a Samsung Flip2 55", which has a touch limitation of five simultaneous touch points and uses InGlass technology for recognition of surface disturbances. More details can be found in the following image, with more specifics of this gadget.

Panel

Diagonal Size	Type	Resolution	Brightness(Typ.)
55"	60Hz New Edge	3840*2160 (Landscape)	350 (without glass), 220 (with glass) nit
Contrast Ratio	**Viewing Angle(H/V)**	**Response Time(G-to-G)**	**Operation Hour**
4000:1 (Typ.)(without glass)	178/178	8ms(Typ.)	16/7

Figure 5.1: Samsung FLIP2 55” specs

To complete these tests, besides the display surface, it was also necessary to have at least eight tokens, to represent the patterns for the eight main isosceles triangles created.

Initially, each available triangle pattern was tested with the patterns of only three touch points, in which the tokens were put on the surface (20 times per each of the eight triangles) and the number of times that the algorithm was able to identify the correct triangle was registered. Then, the same procedure was made but, this time, another touch point was added in the middle of the triangle pattern on the token’s base, so that it was possible to test the recognition precision with four touch points. Finally, since the limit of simultaneous touches on that display was five, another set of tests was made, but with five touch points, in which the initial three-point pattern was extended with two interior touch points.

Essentially, it tested the precision with which the algorithm could properly identify the eight isosceles triangles, using a pattern with three, four, and five touch points. As previously mentioned, to have a significative sample of attempts, per each triangle, the following happened: the token with a three touch point pattern was put on the table twenty times, the token with a four touch point pattern was put on the table twenty times and the same for the token with five touch points; thus, per triangle, the token was put on the table sixty times to test the recognition.

As the following subsections will show, interesting results were achieved.

5.1 Visualization of the experimental tests

Before there is a more in-depth overview of the results, it is important to explain what is shown on the tests, so that it is easier to understand the observations. In the picture below (Figures 5.2, 5.3, 5.4), the active touch points are represented by the light grey circles, next to a drawing of the active represented triangle pattern. Then, the details of the triangle recognized in the pattern are shown in the box below, which, on the top, has the number of the triangle (ranging from 1 - 8) and its unique angle; then, in the middle of the box, there is the orientation of the triangle, its area, center, and common angles.

To have a better understanding of how the detection was performing, was added the information regarding how many active points the screen was detecting and, from these active points, how many made were a part of the pattern.

Moreover, the box is also painted with a specific color, depending on how many active touch points the pattern has (for a pattern with three touch points, it is yellow; with four touch points, it is red and with five touch points it is green).

Once again, it is important to mention that these active touches are only the ones on the base of the token, the user touches are not recognized by the program.

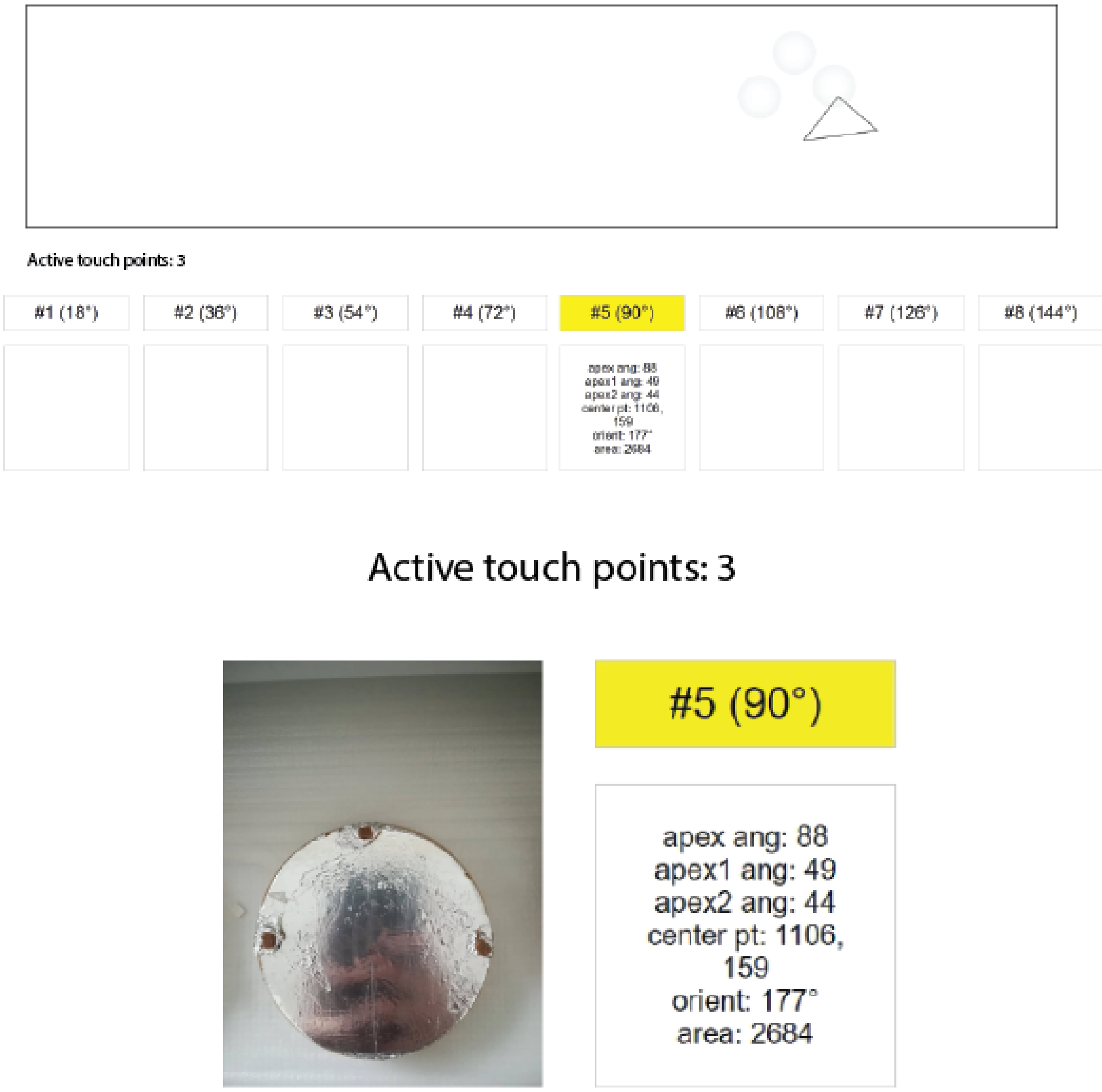

Figure 5.2: Test Results - 3 touch points pattern

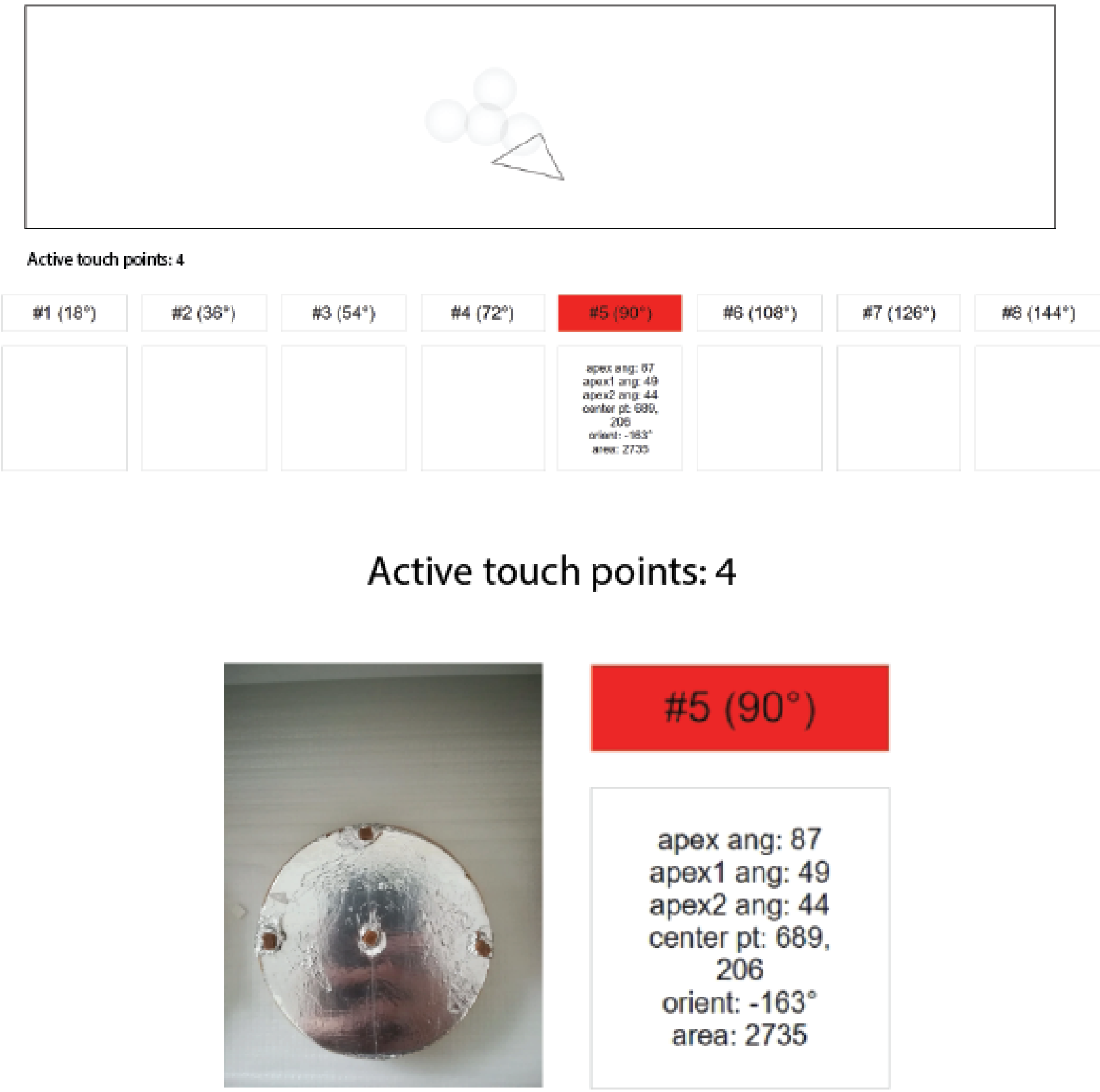

Figure 5.3: Test Results - 4 touch points pattern

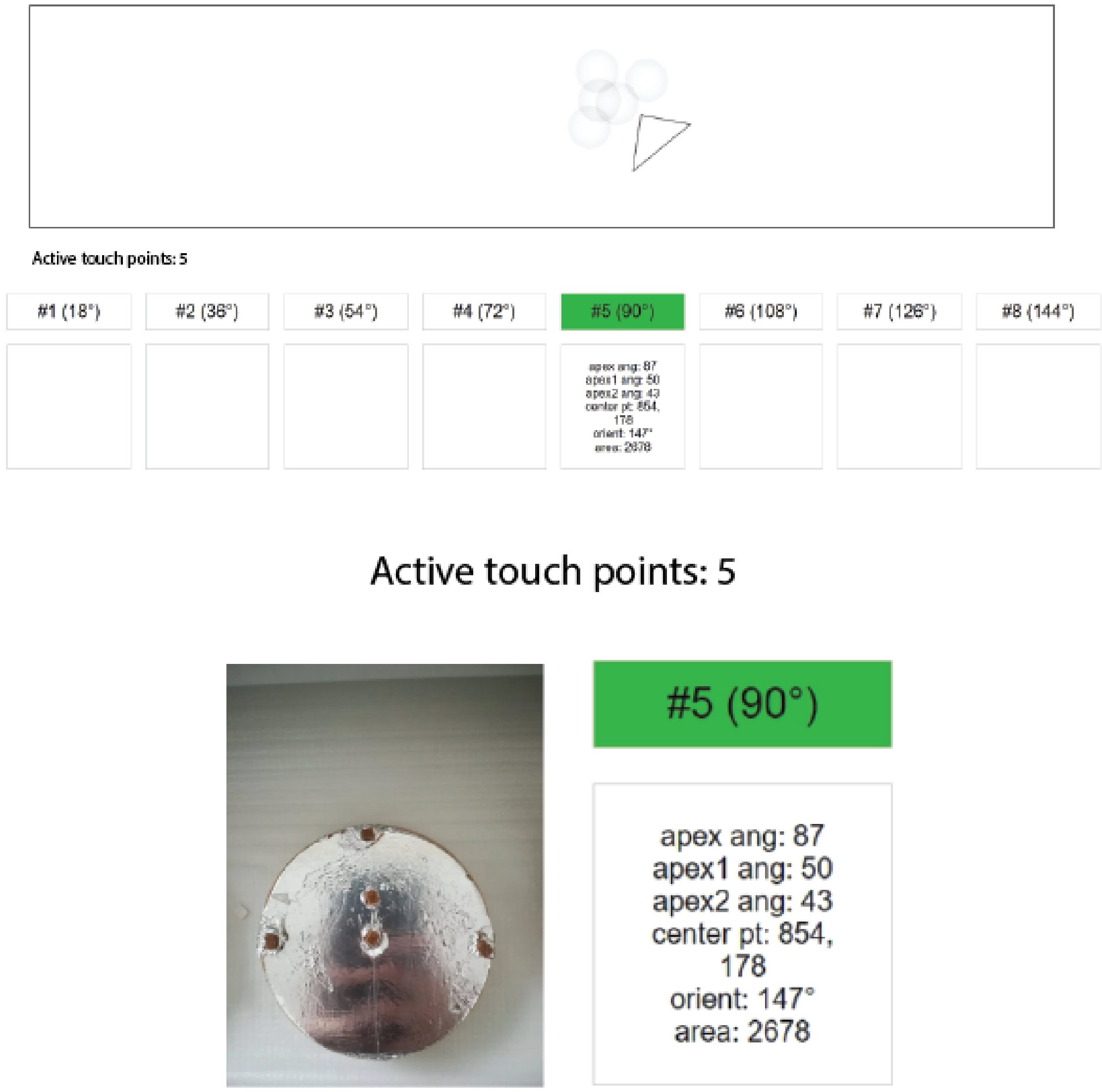

Figure 5.4: Test Results - 5 touch points pattern

In Appendix A it is presented the other patterns of the tokens test results.

5.2 Observations

5.2.1 Recognition Rate

To provide an overview of what was explained in the previous points regarding the recognition rate, a table was constructed in which it is possible to see how many times did the algorithm recognize the pattern on the base of the token, per triangle, and per number of touch points.

As previously explained, each pattern in the tokens was tested twenty times, so that there could be some quantification regarding the recognition rate of the algorithm from the moment the token is put on the display surface. If the pattern was correctly identified as soon as it was set on the interactive table, it would be considered a successful attempt.

There are two main factors that seem to affect the precision of the algorithm: the area of the triangle and the number of touch points. Both in triangles with smaller areas or in patterns with a higher number of touch points, the algorithm could stumble a little, since the distance between the touch points in both cases can be smaller and the algorithm may not be able to recognize all the touch points on the base of the token. Thus, it made sense to analyze the overall recognition rate for both factors: per triangle and per number of touch points. In order to compute the recognition rate, it was divided the number of successful attempts (in which the algorithm recognized the pattern correctly) by the number of total attempts (20 per triangle and number of touch points).

5.2.1.1 3 touch point unique patterns

After putting each of the eight tokens with a three touch points pattern on the display surface twenty times, it was possible to conclude that the algorithm's accuracy for this number of touch points is very good, presenting a 100% recognition rate for all the eight tested tokens with a pattern of three touch points. However, some tokens were somewhat unbalanced due to the positioning of the tags (aluminium touch points) on the token's base. Since their placement had to simulate the respective isosceles triangle and they were at the token's base, with a few triangles, the base did not have well-distributed support and could lean to the side where it did not have the support of a tag, which would cause some confusion on the program since that part of the base would touch the display surface and cause a disruption, it would appear to be a touch point. Nonetheless, this is a problem that can happen with any number of touch points.

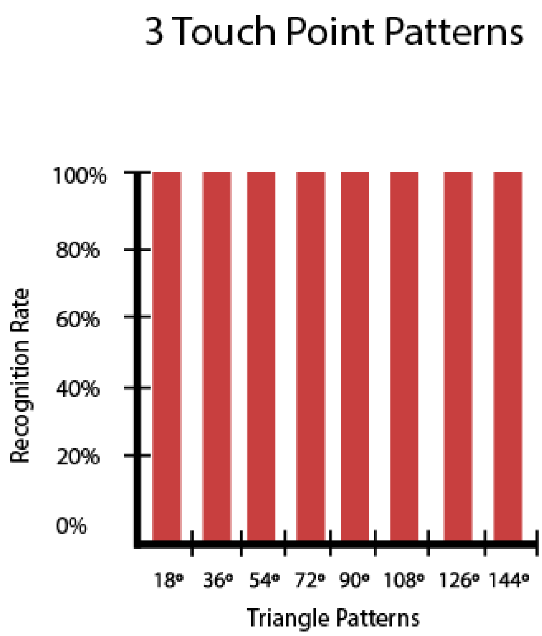

Figure 5.5: 3 Touch Point Pattern Recognition Rate

5.2.1.2 4 touch point unique patterns

Regarding the patterns with four touch points, the results were quite similar to the ones with three. On average, the recognition rate with four touch points was around 94%, which means that all the touch points were recognized at almost every try in all of the triangles. Once again, the basic interactions, such as drag and rotation had no problems.

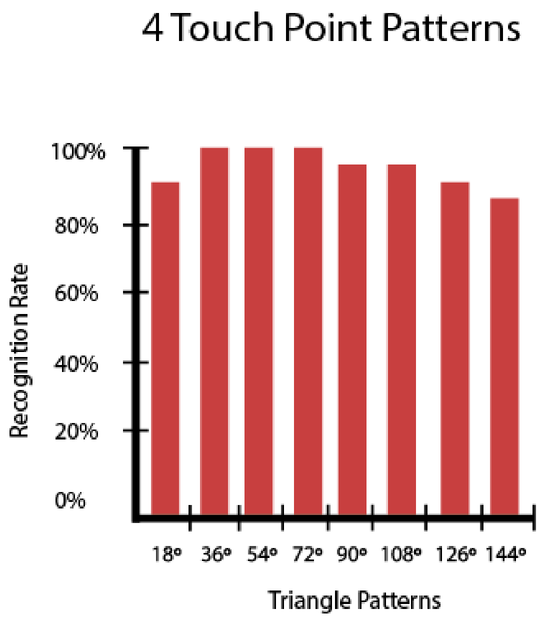

Figure 5.6: 4 Touch Point Pattern Recognition Rate

5.2.1.3 5 touch point unique patterns

In regards to the tests with patterns of five touch points, the scenario changed and the algorithm started to show a few precision problems. Although it can recognize the pattern in the first instance when the token is put on the display surface, with certain movement functionalities - such as drag or rotation - it would confuse the touch points and sometimes even identify two triangles instead of one. Additionally, in triangles with a smaller area (such as the ones with the differentiating angles of 18º and 144º), since some of the touch points on the base of the token are quite close, there could be occlusion of touch points and the algorithm would detect a pattern with four or three touch points instead of five, thus recognizing a different triangular pattern than the one it was supposed to.

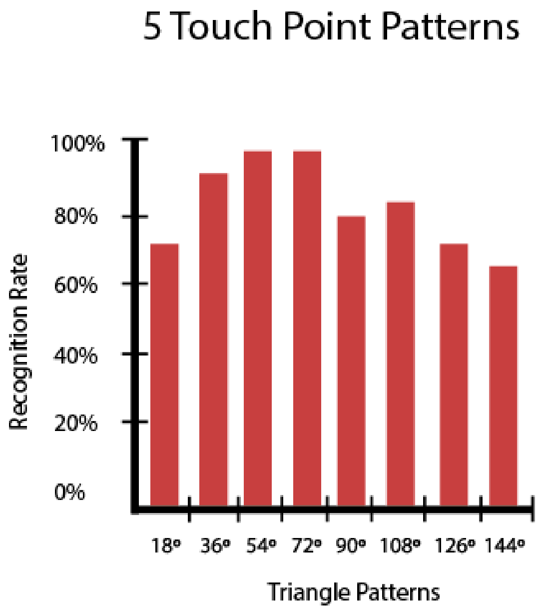

Figure 5.7: 5 Touch Point Pattern Recognition Rate

5.2.1.4 Overview of the Recognition Rate

As it is possible to observe in the table and previously explained, the patterns with three touch points presented a 100% recognition rate, showing no trouble in properly identifying the correct pattern. Regarding the patterns with four or five touch points, the recognition rate drops a bit as expected, due to the problem already explained of the proximity of the touch points - since triangular patterns with smaller areas have also a smaller spatial area in which it is possible to add interior touch points, it can make them too close and cause occlusion of some.

Moreover, as expected, the fewer tags (aluminium touch points) the token has, the more accurate is the algorithm at recognizing the pattern. Thus, if the goal is to extend the number of touch points in triangular patterns, it is recommended that the triangles used are the ones that provide a bigger area of construction (such as the triangles with the differentiating angles of 36º, 54º, and 72º).

	3 Touch Point	4 Touch Point	5 Touch Point	Recognition Rate (Triangle)
18° Triangle	20	18	14	87%
36° Triangle	20	20	18	97%
54° Triangle	20	20	19	98%
72° Triangle	20	20	19	98%
90° Triangle	20	19	16	92%
108° Triangle	20	19	17	93%
126° Triangle	20	18	14	87%
144° Triangle	20	17	13	83%
Recognition Rate (Touch Points)	100%	94%	81%	

Table 5.1: Recognition Rate

5.2.2 Drag and Rotation

It was observed how the drag and rotation functionalities behaved, to check if these motions would break the successful pattern recognition. In the majority of the cases where the patterns showed a high recognition rate, the pattern recognition did not break or miss-matched the recognized triangle, and the drag and rotation motions functioned well, providing a smooth interaction.

On the other hand, with the patterns that had a lower recognition rate, namely the extended triangular patterns 18° and 144°, there were difficulties in maintaining the detected intended pattern, since, in some instances, it broke during a rotation motion.

Regarding the drag motion, it did not show this problem, and the tracking of the object was accurate and precise. It only had a small delay, which is somewhat normal, since this delay represents the time that the operations and tracking took of the active touch points.

5.2.3 Comparison

With the obtained results it is possible to compare our work with previous studies, that use different methods. Also a clearer comparison allows to take conclusions more easily.

Work	Recognition Rate	Number of Patterns	Number of Markers	Token Size(cm)	Token Type
TUIC-2D	-	511	12	5	capacitive, passive
TouchTokens	>95%	6	3	3 - 5	capacitive, passive
PTPG	96.5%	42 - 443	3	4 - 8	capacitive, passive
Our Work	92%	8 - 320	3 - 40	5 - 9	capacitive, passive

Table 5.2: Comparison table of similar studies

As showed in the table, it is possible to see that the recognition rate as maintain a similar percentage of success, the unique possible patterns detected drastically increased from some of the traditional pattern recognition methods, and finally the number of markers has gained a wider usage in comparison to other studies. This method has an increasing potential due to the improvements of multi-touch capacity of future screens.

5.2.4 Improvements and technology limitations

Although positive results were obtained, there are still improvements to be made. Firstly, there could be another restriction implemented on the algorithm, so that it cannot recognize multiple triangles in one triangular pattern as it sometimes happened with five touch points.

Moreover, the construction of the token is also another point to focus on, considering both the size of the base of the token and of the pattern, so that there is enough space to add interior points and no occlusion occurs. The bigger the area working at, the more precise will the recognition be. Additionally, it is also important to keep in mind the balance of the token, so that, in some patterns, the token can still be balanced and nothing besides the desired touch points actually interacts with the display surface.

5.3 Summary

In this dissertation, it was possible to reach most of the initial goals set. An algorithm capable of recognizing unique triangular patterns was created, that works on capacitive screens of different sizes, as well as in InGlass screens.

The algorithm is capable of extending the number of touch points in the initial triangular patterns, thus increasing the number of possible patterns and maximizing the usage of touch points.

Although the recognition rate did not increase when compared to the ones analyzed in the previous studies, it maintained its values. The ideal scenario would be to have a 100% recognition rate, which did not happen in all instances of the stress tests for the reasons already explained.

Additionally, the algorithm is capable of differentiating the pattern touch points on the base of the token from the user touch points, which was achieved through a large number of restrictions, necessary for the usage of the program.

Moreover, regarding the flexibility of the algorithm working in both high and low-spec machines, the results were positive since it did not require a high-spec one to work properly.

Finally, by exploring the usage of passive tokens on different displays, it is possible to conclude that some types of screens are better adapted to execute this technology than others. Nevertheless, the algorithm worked on every device tested, with some having bigger limitations than others as in the case of, for example, having several touch points in a small area, which can confuse the algorithm.

Chapter 6

Conclusions and Future Work

In the initial chapter, it was established the presence and usage of tangible interfaces, by analyzing where they were applied over the years, and their technological applications.

Consequently, when investigating the related work, a few questions and problems that were already identified in the technology of multi-touch points in tangible surfaces were explored. Alongside the problems, some of the solutions implemented in conducted studies were also explored.

Afterwards, having researched previous works, it was possible to define the main goals for this dissertation. Then, it was established a possible path and a theoretical solution that could be implemented to achieve those goals, according to our needs and the technological developments already made.

Furthermore, after the theoretical solution was explored, it was implemented, tested, and adapted to face the challenges that appeared during the development of the algorithm.

Finally, once the program was functional and executed properly, the final testing phase was done, in order to be able to register the main observations and take some conclusions from the algorithm developed. The final results were satisfactory, and it was possible to extend the initial unique isosceles triangular patterns of three points to five points in the Samsung Flip 2 55".

6.1 Future Work

There is still potential to improve, mainly the robustness of the recognition, since one of the greatest limitations here, is the limit of simultaneous touch points that an interactive surface can take. Small area tokens, and small area patterns, are also tricky to work with since this method requires a minimum distance between touch points to be effective in the recognition.

Moreover, the token's design and construction itself can be an important area to explore. For example, with the algorithm created in this dissertation to extend common triangular patterns with three touch points to multi-point triangular patterns, it would be interesting to design tokens that allowed the insertion of an interior point by only pressing a button. The base of the token would have the three touch points of the isosceles triangle and, once the button was pressed, it would add a touch point to the interior of that pattern, thus reducing the need for more tokens and allowing the insertion of different patterns in one single token. As such, the token maintains the main triangle pattern but switches it to an extended pattern in a quick and seamless motion, creating a more interactive tool for tangibles.

This proposition of changing the design of the physical token can be a great way of reducing the number of tokens used, and it would be a perfect fit for the usage of the created algorithm.

Overall, this work had satisfactory results and the main goal - extending triangular patterns to multi-point triangular patterns - was achieved. Nonetheless, there are still improvements that could be made to optimize the algorithm and its application, such as the creation of more complex tokens that could reduce the need to have one token per pattern (as mentioned in the previous example of the token with buttons), or even by working more in the algorithm and doing further tests regarding its velocity and the functionality in several different devices.

Although pattern recognition is already quite explored, there is still untapped potential regarding pattern extension and the applicability of this technology on several multi-touch surfaces. The solution developed can be further worked on to create more complex versions that, ultimately, can be applied to as many things as the programmer's creativity allows. As long as there is a multi-touch screen and capable hardware, there is much to explore within this field.

References

[1] Orit Shaer and Eva Hornecker. Tangible user interfaces: Past, present, and future directions. *Foundations and Trends in Human-Computer Interaction*, 3:1–137, 2009.

[2] Hiroshi Ishii, Carlo Ratti, Ben Piper, Yao Wang, Assaf Biderman, and Eran Ben-Joseph. Bringing clay and sand into digital design - continous tangible user interfaces. *BT Technology Journal*, 22:287–299, 2004.

[3] Ulrich Von Zadow, Daniel Bösel, Duc Dung Dam, Anke Lehmann, Patrick Reipschläger, and Raimund Dachselt. Miners: Communication and awareness in collaborative gaming at an interactive display wall. pages 235–240. Association for Computing Machinery, Inc, 11 2016.

[4] Clément Pillias, Raphaël Robert-Bouchard, and Guillaume Levieux. Designing tangible video games: Lessons learned from the sifteo cubes. pages 3163–3166, 2014.

[5] Paul Marshall. Do tangible interfaces enhance learning? pages 163–170, 2007.

[6] Oren Zuckerman, Saeed Arida, and Mitchel Resnick. Extending tangible interfaces for education: Digital montessori-inspired manipulatives, 2005.

[7] Simon Voelker, Kosuke Nakajima, Christian Thoresen, Yuichi Itoh, Kjell Ivar Øvergård, and Jan Borchers. Pucs: Detecting transparent, passive untouched capacitive widgets on unmodified multi-touch displays. pages 101–104. Association for Computing Machinery, 2013.

[8] Nicolas Villar, Daniel Cletheroe, Greg Saul, Christian Holz, Tim Regan, Oscar Salandin, Misha Sra, Hui Shyong Yeo, William Field, and Haiyan Zhang. Project zanzibar: A portable and flexible tangible interaction platform. volume 2018-April. Association for Computing Machinery, 4 2018.

[9] Rong Hao Liang, Han Chih Kuo, Liwei Chan, De Nian Yang, and Bing Yu Chen. Gauss-stones: Shielded magnetic tangibles for multi-token interactions on portable displays. pages 365–372. Association for Computing Machinery, 10 2014.

[10] Evren Bozgeyikli and Lal Lila Bozgeyikli. Evaluating object manipulation interaction techniques in mixed reality: Tangible user interfaces and gesture. pages 778–787. Institute of Electrical and Electronics Engineers Inc., 3 2021.

[11] Jivago Serrado Nunes, Nelson Castro, Sergio Gonçalves, Nélson Pereira, Vitor Correia, and Senentxu Lanceros-Mendez. Marked object recognition multitouch screen printed touchpad for interactive applications. *Sensors (Switzerland)*, 17, 12 2017.

[12] Leonardo Angelini, Denis Lalanne, Elise van den Hoven, Omar Abou Khaled, and Elena Mugellini. Move, hold and touch: A framework for tangible gesture interactive systems. *Machines*, 3:173–207, 9 2015.

[13] Andrea Bottino, Andrea Martina, Francesco Strada, and Amirhosein Toosi. Gaine - a portable framework for the development of edutainment applications based on multitouch and tangible interaction. *Entertainment Computing*, 16:53–65, 7 2016.

[14] Simon Voelker, Kjell Ivar Øvergard, Chat Wacharamanotham, and Jan Borchers. Knobology revisited: A comparison of user performance between tangible and virtual rotary knobs. pages 35–38. Association for Computing Machinery, Inc, 11 2015.

[15] S K Lee, W Bu on, and K C Smith. A multi-touch three dimensional touch-sensitive tablet, 1985.

[16] Mark Fiala. Artag, a fiducial marker system using digital techniques. volume II, pages 590–596. IEEE Computer Society, 2005.

[17] Oren Zuckerman, Saeed Arida, and Mitchel Resnick. Extending tangible interfaces for education: Digital montessori-inspired manipulatives, 2005.

[18] Matthias Wölfel, Bernd Lintermann, and Nikolaus Völzow. Using tangible surfaces in opera-2000 electronic arenas for culture, performance, arts and entertainment view project teilhabe-digital view project using tangible surfaces in opera, 1997.

[19] Simon Voelker, Christian Cherek, Jan Thar, Thorsten Karrer, Christian Thoresen, Kjell Ivar Øvergåard, and Jan Borchers. Percs: Persistently trackable tangibles on capacitivemulti-touch displays. pages 351–356. Association for Computing Machinery, Inc, 11 2015.

[20] Emmanuel Courtoux, Caroline Appert, and Olivier Chapuis. Walltokens: Surface tangibles for vertical displays. 2021.

[21] Rachel Blagojevic and Beryl Plimmer. Captui: Geometric drawing with tangibles on a capacitive multi-touch display. pages 511–528, 2013.

[22] Liwei Chan, Stefanie Müller, Anne Roudaut, and Patrick Baudisch. Capstones and zebrawidgets: Sensing stacks of building blocks, dials and sliders on capacitive touch screens. pages 2189–2192, 2012.

[23] Martin Kaltenbrunner and Ross Bencina. reactivision: A computer-vision framework for table-based tangible interaction.

[24] Rafael Morales Gonzalez, Caroline Appert, Gilles Bailly, and Emmanuel Pietriga. Touchtokens: Guiding touch patterns with passive tokens. pages 4189–4202, 2016.

[25] Rong Hao Liang, Kai Yin Cheng, Liwei Chan, Chuan Xhyuan Peng, Mike Y. Chen, Rung Huei Liang, De Nian Yang, and Bing Yu Chen. Gaussbits: Magnetic tangible bits for portable and occlusion-free near-surface interactions. pages 1391–1400, 2013.

[26] Brien East, Ahmed Arif, Sean DeLong, Ali Mazalek, and Roozbeh Manshaei. Actibles: Open source active tangibles. pages 469–472. Association for Computing Machinery, Inc, 11 2016.

[27] Desney S. Tan, Bo. Begole, Wendy. Kellogg, and SIGCHI (Group : U.S.). *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems : 2011 proceeding, Vancouver, BC, Canada - May 07-12, 2011*. ACM Press, 2011.

[28] Martin Kaltenbrunner and Florian Echtler. The tuio 2.0 protocol: An abstraction framework for tangible interactive surfaces. *Proceedings of the ACM on Human-Computer Interaction*, 2, 6 2018.

[29] Han Chih Kuo, Rong Hao Liang, Long Fei Lin, and Bing Yu Chen. Gaussmarbles: Spherical magnetic tangibles for interacting with portable physical constraints. pages 4228–4232. Association for Computing Machinery, 5 2016.

[30] Neng Hao Yu, Li Wei Chan, Seng Yong Lau, Sung Sheng Tsai, I. Chun Hsiao, Dian Je Tsai, Lung Pan Cheng, Fang I. Hsiao, Mike Y. Chen, Polly Huang, and Yi Ping Hung. Tuic: Enabling tangible interaction on capacitive multi-touch display. pages 2995–3004, 2011.

[31] Zanzhen Huang, Minyong Shi, Guangzheng Fei, and Yaxin Zhu. Ptpg: A poisson triangular pattern generator for tokens on tangible surfaces. *IEEE Access*, 8:76019–76027, 2020.

[32] Caroline Appert, Emmanuel Pietriga, Eléonore Bartenlian, Rafael Morales González, Éléonore Bartenlian, and Rafael Morales. Custom-made tangible interfaces with touchtokens. 2018.

REFERENCES

Appendix A

Appendix

A.1 Token Pattern Detection Results

The following images show the full results of the performed tests and the correspondent token with the detected pattern. As mentioned before, the color scheme of the box dictates the number of inside points detected inside each triangle pattern, yellow translates to zero, red translates to one and green translates to two.

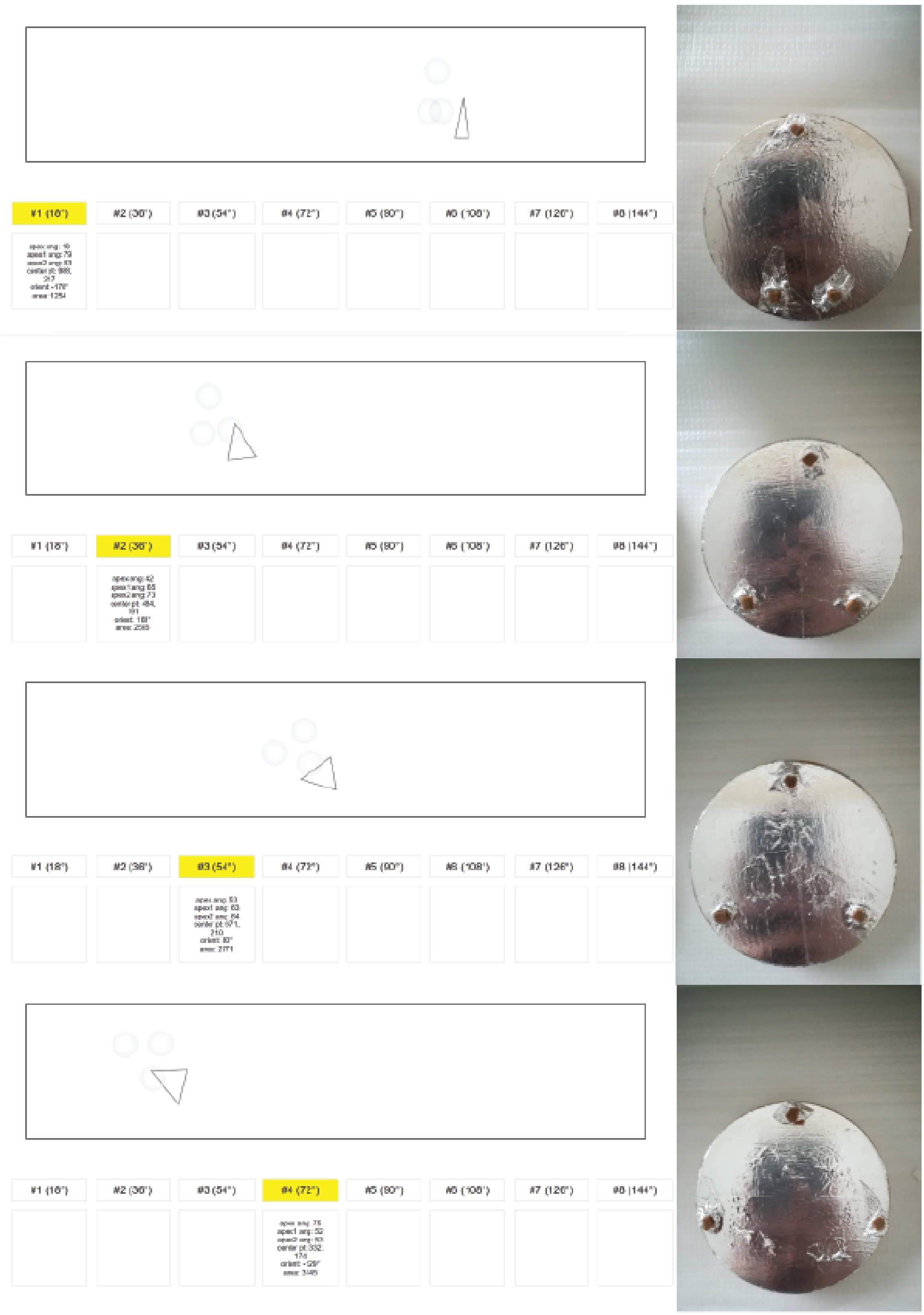

Figure A.1: 3 Touch Point Pattern Recognition

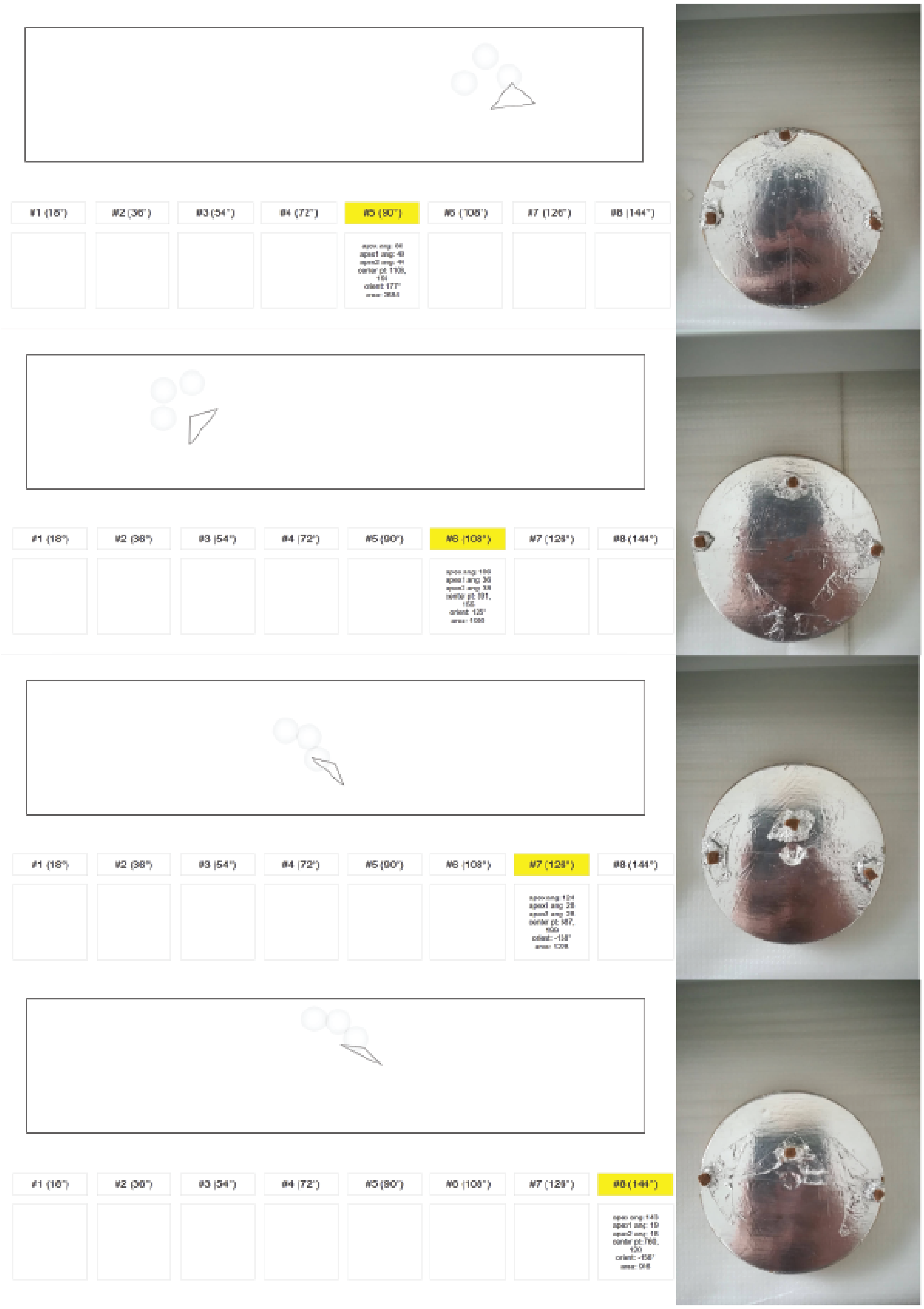

Figure A.2: 3 Touch Point Pattern Recognition

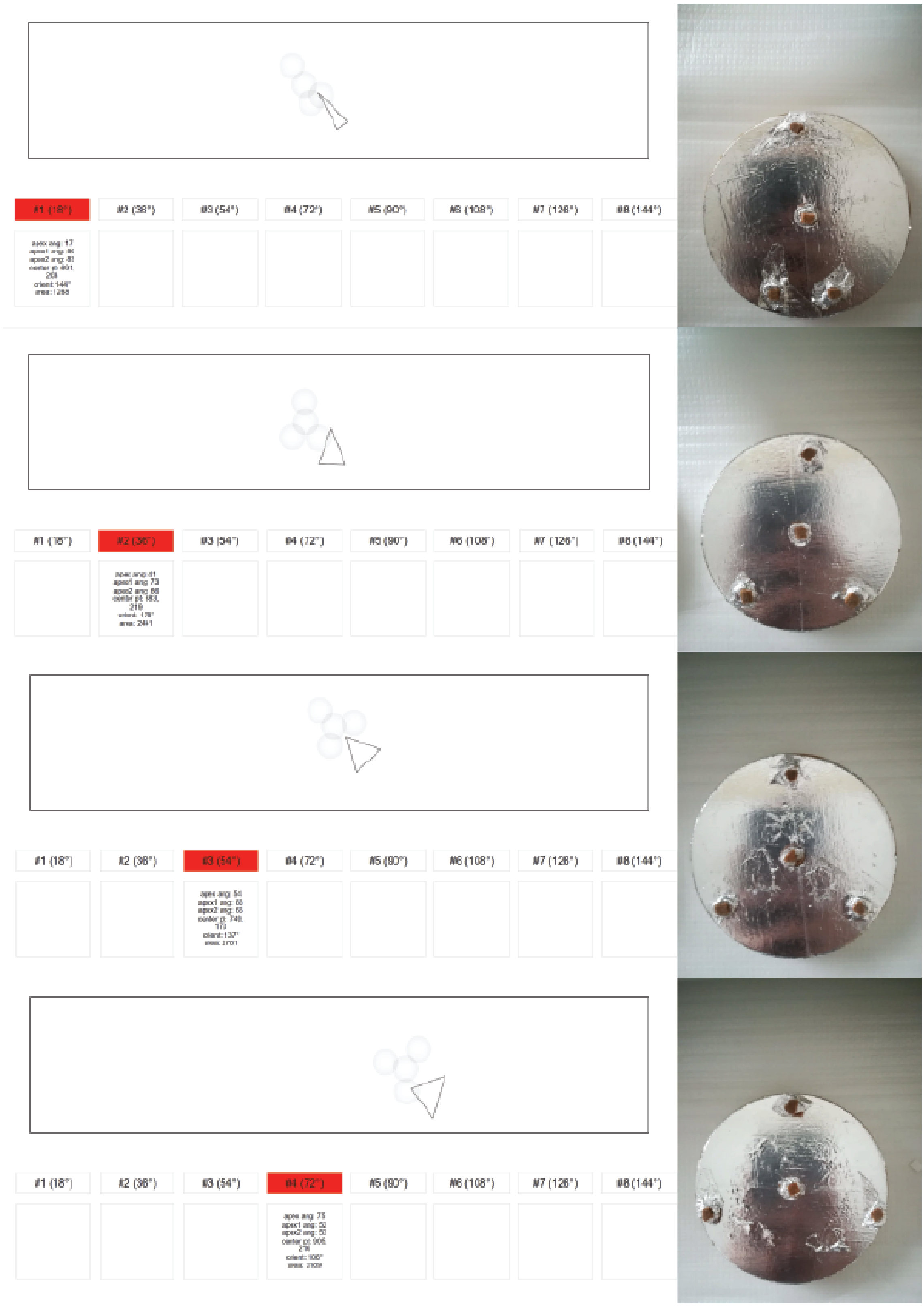

Figure A.3: 4 Touch Point Pattern Recognition

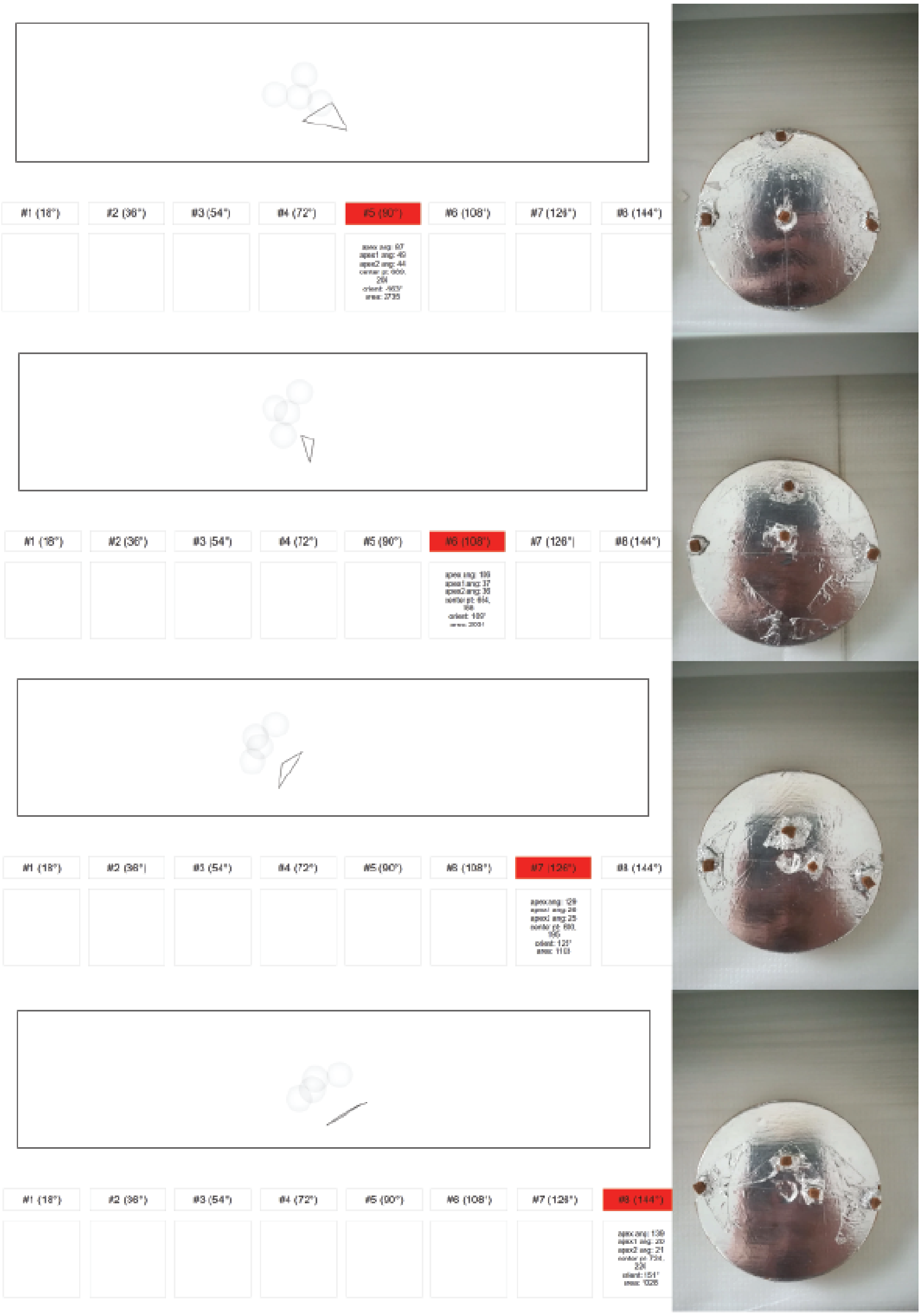

Figure A.4: 4 Touch Point Pattern Recognition

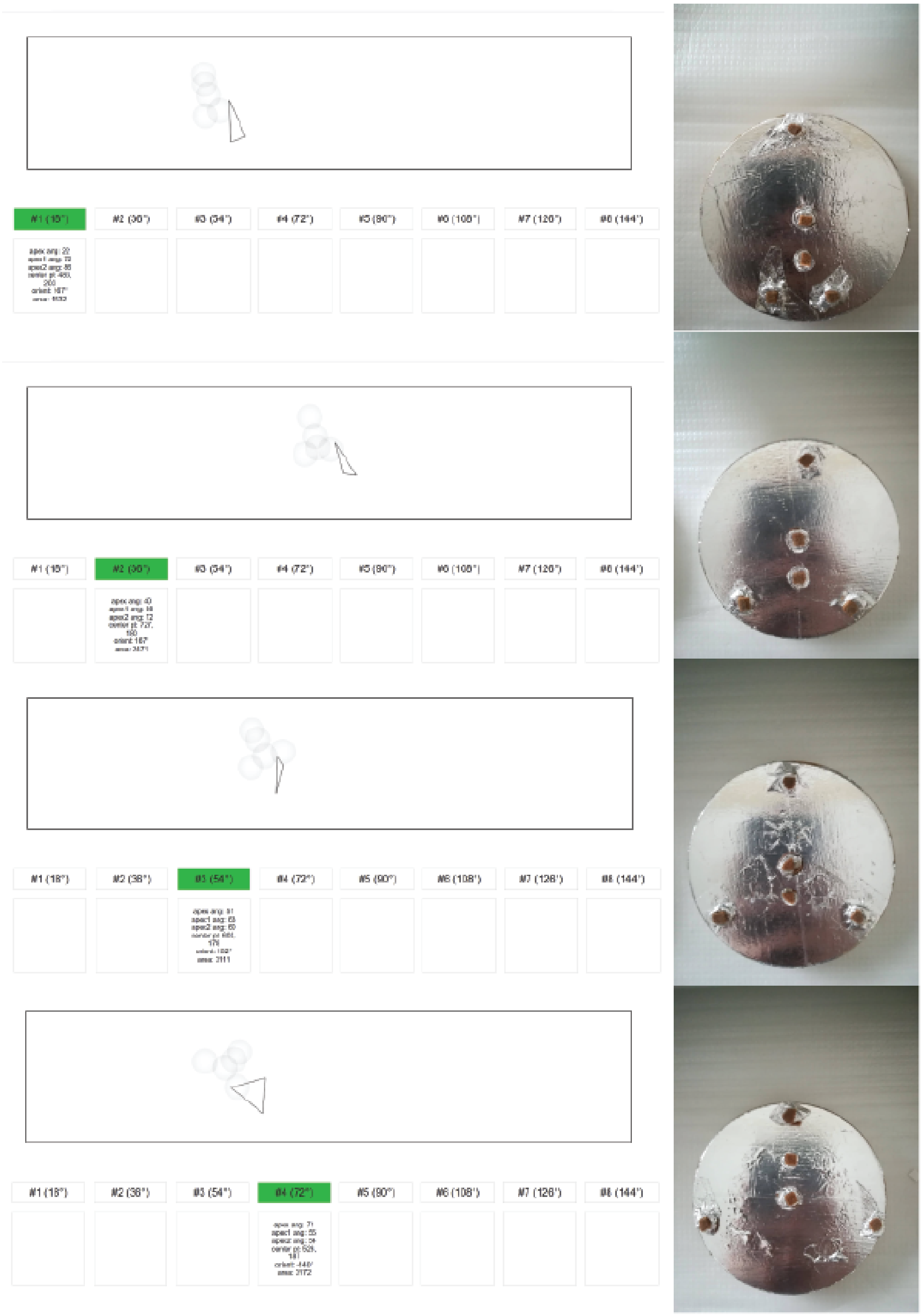

Figure A.5: 5 Touch Point Pattern Recognition

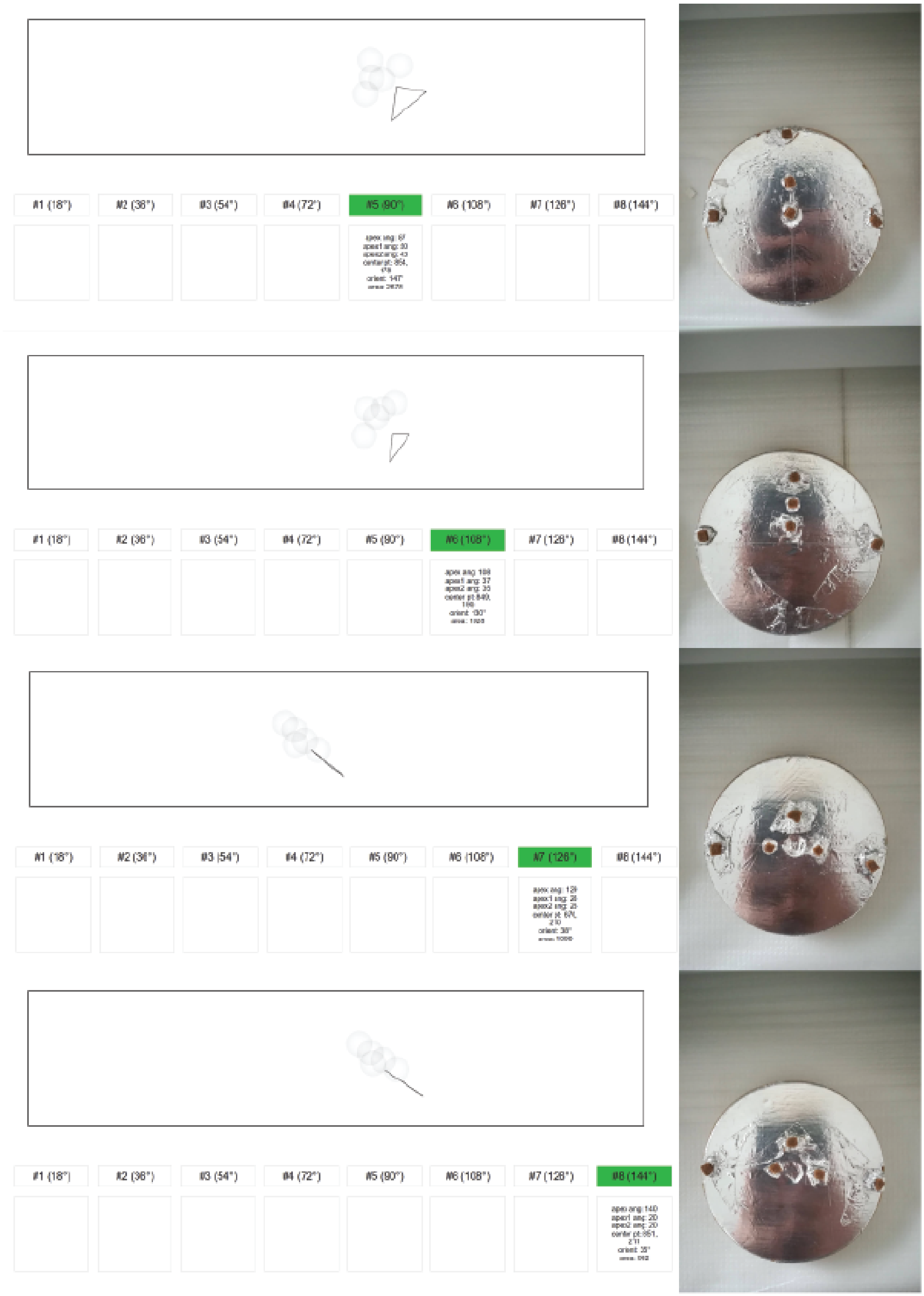

Figure A.6: 5 Touch Point Pattern Recognition

A.2 Flowchart

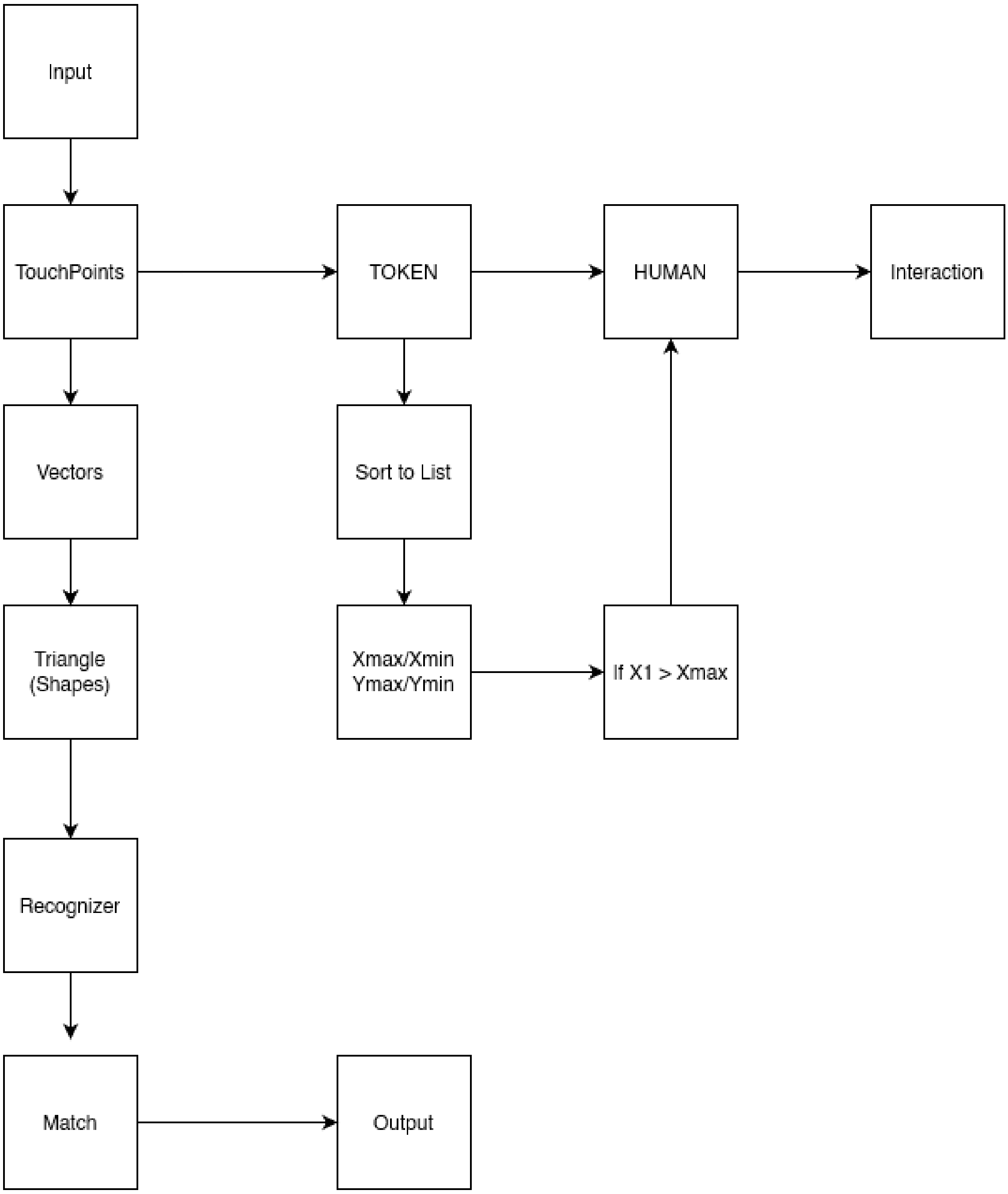

Figure A.7: Algorithm flow