Faculdade de Engenharia da Universidade do Porto

FEUP

PokerVision – Perception Layer for a Human-Robot Poker Table

Paulo Sérgio Ribeiro Martins

OFFICIAL VERSION

Thesis performed in the context of the
Master in Electrical and Computer Engineering
Major - Automation

Supervisor: Prof. Dr. Luís Paulo Reis

February 2011

"Know when to hold 'em, and know when to fold 'em"

Kenny Rodgers – *The Gambler*

© Paulo Sérgio Ribeiro Martins, 2011

Resumo

Nos últimos anos, o poker tem-se tornado cada vez mais popular, especificamente o poker online, sendo o segmento com o maior crescimento na indústria do jogo online ao longo dos últimos anos.

O fascínio pelo poker estende-se também à comunidade académica, a qual tem vindo a investigar várias vertentes associadas a este jogo. Um dos ramos das investigações desenvolvidas, lida com a criação de jogadores artificiais capazes de se defrontar, e mesmo superar, jogadores profissionais de poker em salas de jogo online. Apesar dos agentes que têm vindo a ser desenvolvidos serem cada vez mais competitivos, ainda não são capazes de competir, fisicamente, num ambiente de jogo real, com outros jogadores humanos. Como tal, o próximo passo consiste em combinar a inteligência artificial de um jogador de poker vrtual com o hardware adequado, de maneira a ser-lhe possível jogar em ambientes reais, i.e. mesas de poker tradicionais.

Este documento apresenta uma abordagem à identificação de cartas de jogo e contagem de fichas num jogo de poker, recorrendo a um dispositivo de captura de vídeo de baixo custo e algoritmos de visão computacional. A maioria dos trabalhos desenvolvidos sobre identificação de cartas de jogo, baseiam-se num posicionamento óptimo da câmara e em ambientes controlados. No entanto, pretende-se com a abordagem apresentada garantir o funcionamento do sistema em ambiente real, em condições reais, juntamente com todas as restrições que daí advêm.

O método de reconhecimento das cartas de jogo baseia-se na contagem de naipes presentes nas cartas, combinada com a técnica de "template matching". Relativamente à contagem das fichas, a metodologia adoptada baseia-se na segmentação de cores combinada com a Transformada de Hough aplicada a círculos.

Os resultados alcançados no reconhecimento de cartas variam entre 95% e 99%, dependendo principalmente da colocação da webcam em relação às cartas. Quanto ao reconhecimento de fichas de jogo, quando isoladas, a contagem é correcta entre 56% e 100% das vezes, dependendo da cor das fichas consideradas.

Abstract

In recent years, poker has become a mass phenomenon, and its popularity continues to grow, more specifically online poker, which has been the fastest growing segment of the online gambling industry over the last years. Its boom has been phenomenal, and all trends point out to further growth.

The poker hype is not limited to the general population but also to the academic society which has been investing in research aimed to this subject. One branch of this research deals with creating an artificial player able to beat professional poker players in online environments. Although the designed agents are becoming better every time, they are still not able to compete, physically, in a real environment, with human players. Therefore, the next step consists in combining the artificial poker player intelligence with the proper hardware which enables it to play in a real environment, i.e. traditional poker tables.

This document presents an approach to the identification of playing cards and counting of chips in a poker game, using an entry-level webcam and computer vision methodologies. Most of previous works on playing cards' identification rely on optimal camera position and controlled environment. Instead, the presented approach is intended to suit a real and uncontrolled environment along with all its constraints.

The recognition of playing cards lies on suit counting combined with the template matching technique. As for the counting of chips, the adopted methodology relies on colour segmentation combined with the Hough Circles Transform.

The overall results achieved for the playing cards recognition are between 95% and 99%, which depend mainly on the position of the webcam around the table. As for the counting of the chips, when completely visible, it is correct between 56% and 100% of the times, depending on the colour of the chips considered.

Acknowledgements

I would like to thank to my colleague and great friend of mine, André Vidal, for all the feedback and suggestions which allowed me to improve the present project.

I thank to another great friend of mine, Luís Teixeira, for his time and energy spent designing the graphical interface which, unfortunately, was not integrated in the system due to the lack of time.

I thank my supervisor Prof. Dr. Luís Paulo Reis for envisioning the concept of this thesis, which I take a great interest in.

Finally, to my family, who have always been present to show their support and love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis focuses on the development of a perception module for a theoretical humanoid poker player, capable of playing in a real poker game environment. A complete perception layer is able to recognise a standard deck of playing cards, with both ranks and suits uniquely recognized, count casino chips of different colours and follow the poker game play. The latter is based on determining the number of players in a table and the position of the *Dealer*, and by recognising the actions of the players during the betting round. The goal is to couple this perception layer with the decision layer developed in a previous thesis [1]. It is aimed to be coupled with a robotic system, in order to remove the need for the human player to manipulate the cards and/or casino chips.

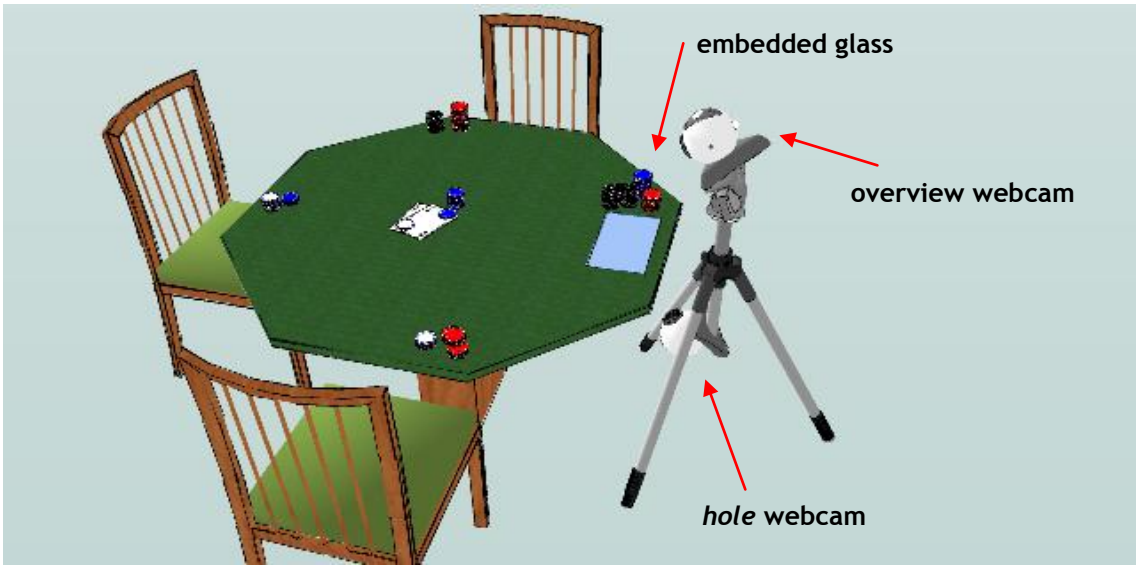


Fig. 1.1 - Illustration of the concept behind PokerVision. The computer which is connected to both webcams and runs the algorithm is not shown

In Fig. 1.1 it is depicted a poker table with four seats, where three of them are for human players and the fourth is for the PokerVision perception module. The PokerVision is physically

conceptualized as a tripod which holds two webcams, where the top one captures the overview of the table, while the bottom one captures the pocket cards which are supposed to be facing down in the glass embedded in the poker player. Both webcams are then connected to a computer which runs the perception algorithm presented in this thesis.

1.1 Motivation

In the past years the online poker has grown surprisingly fast, specifically the online poker which gathered millions of players worldwide [2]. It is said that the “poker boom” is justified, not only but also, by the improvements to the coverage of televised poker tournaments, namely the *hole cam*, which reveals the *hole cards* of each player, and the *lipstick cam*, which is built into the table in front of a player's seat aiming to show the drama around each player [3].

Nowadays there are numerous online poker rooms on the internet and the number keeps growing. Online poker offers the comfort of playing where you want and when you want. There is no need to go to the casino specifically to play it. Furthermore, taking advantage of the fact that online nobody knows each other and what people are doing, numerous poker bots have been developed with the goal of playing winning poker with almost no action from the user. Few were created just for fun, while the majority is intended to be a “money making” machine. Along with this, online communities grew as well to discuss the artificial intelligence embedded in these bots, allowing almost everyone to develop his own bot. *PokerAI* is one of the biggest communities in this context [4].

The downside of online poker and its bots is that happens online. Nothing beats the live feeling with the players in front of each other during the poker game. Online poker is a great way of training, but it is no substitute for live poker, especially at the championship level.

This point of view motivates merging the online playing bots and the live poker, resulting in a “poker humanoid” which embodies intelligence, perception and action, making it able to sit at a real poker table and play with real opponents.

1.2 Objectives

In this thesis, the perception of a Poker game in real and uncontrolled environments is analyzed. Although there has been some research regarding playing cards recognition, this thesis focuses on the development of the perception layer for a poker player agent, reliable on real environments while using low cost equipment.

As referred before, the main goal of this thesis consists in developing a perception layer liable of completely identifying all the actions during the poker game play, such as identifying the playing cards, counting chips and following the complete game play. To achieve such goal, the objectives were defined as follows:

- Develop algorithms to identify the playing card’s Rank and Suit
- Design an algorithm capable of counting the chips present on the table
- Adapt the previous algorithms to a real poker table and environment
- Enable the perception layer to detect players in the table, as well as the *Dealer* position

- Adapt the previous improvement in order to follow the game play and make it completely standalone after the game start
- Test the robustness of the perception layer with the adopted equipment for this work
- Implement a communication protocol, aiming to pass the whole information percept on to the poker agent.

1.3 Thesis Structure

This thesis is divided into eight chapters.

The first chapter (this chapter) presents the thesis theme, as well as the context in which it arises and the generic goals to be achieved.

The second chapter illustrates the Poker game itself and a brief introduction to the game rules and how the game play is processed. It is somewhat important to be aware of the game play in order to understand some decisions made along the implementation.

The third chapter presents the state of the art concerning the recognition of playing cards and chips, not only based on vision recognition, but also by means of other technologies such as RFID or Bar Code Scanning.

In the fourth chapter the complete setup is introduced, both hardware and software, used along the work. The modifications performed to the equipment are detailed as well and the system's requirements and design is reviewed.

The fifth chapter details the algorithms implemented, consisting of the card finding and extraction, card's rank recognition and card's suit recognition, as well as the extraction of chips followed by its counting.

The sixth chapter details the features implemented in order to adapt the system to a real environment and follow the game play. It is determined the number of players and respective positions, the position of the *Dealer button* and the *PokerVision*'s turn to play.

Chapter seven shows the robustness of the chosen algorithm and presents the experimental results. The card recognition and the chip counting are analyzed under various conditions.

Final conclusions and an outlook of possible enhancements are given in chapter eight.

Chapter 2

The Poker Game

Poker is a card game in which players bet that their *hand* is stronger than the hands of their opponents. The round ends with the showdown and, therefore, the best *hand* wins *the pot*, i.e. all the bets made; or all but one player have given up betting and dropped out of play and, therefore, the last person to raise wins the pot without a showdown.

There are many poker variants. The most popular nowadays, and relevant in the context of this work, is Texas Hold'em.

2.1 Texas Hold'em Rules and Game Play

Poker Texas Hold'em is a *community card* game, which means some cards are played face-up in the table and can be used by all the players. A total of five *community cards* are dealt, while each player is given two private cards, called *hole cards*. The *hole cards* are combined with the *community cards* in order to make the best possible five-card hand.

Before the round starts, one player is assigned to be the *dealer* and his seat is marked with a *dealer button* (Fig. 2.1 – player "F"). This position is rotated clockwise at the end of every round.

The two players on the left of the dealer are forced to bet a predetermined amount of money into *the pot (*Fig. 2.1 – players "A" and "B" respectively). The first player on the left of the dealer bets the *small blind*, which is half the minimum bet, while the second player bets the *big blind*, which corresponds to the minimum bet for this table. This procedure ensures that there is action in every round.

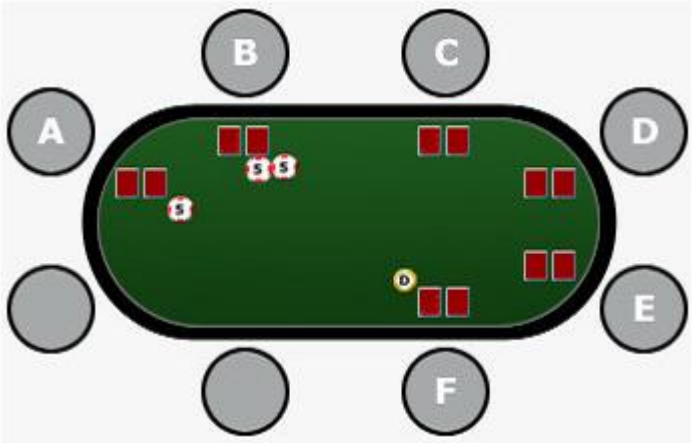


Fig. 2.1 - "F" is the *dealer* and with the *dealer button* assigned; "A" is the *small blind* and "B" is the *big blind*; the seats between "F" and "A" are empty;

Players in a poker game act in turn, in clockwise rotation, during the so called betting round. When it is the turn of a player to act, the first verbal declaration or action he takes binds him to his choice of action. This rule prevents a player from changing his action after seeing how other players react to his initial action. If someone has already made a bet in the present round, then the player can:

Call – Match/equal the other player's bet.

Raise – To increase another player's bet.

Fold – Forfeit cards, thus giving up on the pot.

If no one has betted so far in this round, then the player's options are as follows:

Check – Passing on making an action.

Raise – Put money in *the pot*, forcing the opponents to *Call* the bet if they want to continue in the present round.

Fold – Forfeit cards, thus giving up on the pot. Note: although *folding* is possible if nobody has made a bet yet, it is not an intelligent move since the player can continue in the round "for free", i.e. without having to bet anything.

The players continue to act, in each round, until everyone matches the amount raised by a player. The player who raises may only act again in the same round if another player re-raises him.

The first betting round is called Pre-Flop. The players are given 2 cards faced down which remain private for each player's opponents. These are the *hole cards* (also called *pocket cards*). When the betting round starts, the first player to act is the one on the left of the big blind (Fig. 2.1 – player C). The players have to decide whether to *Call* the big blind, *Raise*, or simply *Fold* his cards. The only player who is able to *Check* during the *Pre-Flop* phase, in case nobody else has raised, is the *big blind* player. When the initial betting round is over, three *community cards* are placed in the center of table. These 3 cards are called *the flop*.

After *the flop* is dealt a new betting round starts. From now on, the first player to bet is the one on the left of the *Dealer* position. The betting round is processed the same way, concerning the players acting in their respective turn.

A fourth card is dealt after the end of the second betting round. This card is refereed as *the turn* and it is followed by a third betting round. At the end, the last card, called *the river*, is placed at the center of the table, and proceeded by the final betting round. When all betting rounds are over and all the five *community cards* are at the table (Fig. 2.2) the *showdown* starts. Each player shows their two *hole cards* and the player who has the best five-card combination wins the hand and, of course, *the pot*.

It is also possible for a player to win before the *showdown* in any part of the game, i.e. pre-flop, flop, turn or river. This happens when all the players fold except for one.

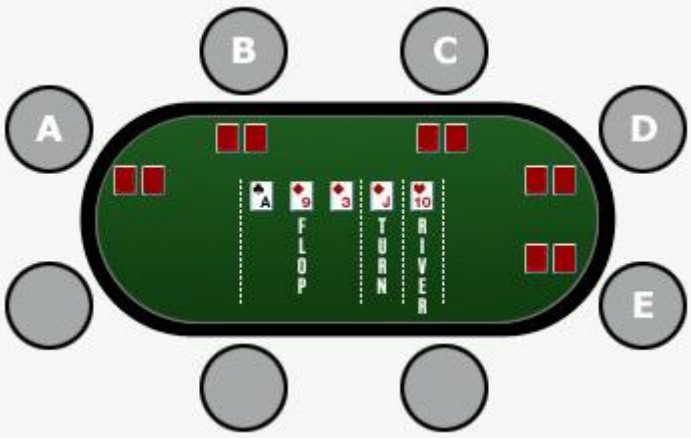


Fig. 2.2 – After the five *community cards* have been dealt

Example Betting Round 1

There are six players at the table (Fig. 2.1):

Player "F" - *Dealer*

Player "A" - *Small blind* (10 cents)

Player "B" - *Big blind* (20 cents)

Start of betting round – *pre-flop*

Player "C" - *Calls* the big blind (20 cents)

Player "D" – *Folds*

Player "E" - *Folds*

Player "F" (*Dealer)* - *Calls* the *big blind* (20 cents)

Player "A" (*small blind)* - *Calls* the *big blind* (since this player already has 10 cents bet, he only has to add another 10 cents, for a total of 20 cents)

Player "B" (*big blind)* - *Checks* (since this player already has the bet matched, he does not need to add more money to *call*)

End of betting round - it is time to deal *the flop*

Note: When Player "A" calls the *big blind,* all players will have the same amount of money in front of them, however Player "B" (the *big blind*) has not had a chance to act, so the betting round is not over. Once Player "B" checks, both conditions are met and the betting round is over.

Example of Betting Round 2 – Re-Raise example

There are six players at the table (Fig. 2.1):

Player “F” - *Dealer*

Player “A” - *Small blind* (10 cents)

Player “B” - *Big blind* (20 cents)

Start of betting round – *pre-flop*

Player “C” - *Calls* the *big blind* (20 cents)

Player “D” - *Raises* (40 cents)

Player “E” – Folds

Player “F” (*Dealer*) – Folds

Player “A” (*small blind*) – Folds

Player “B” (*big blind*) – Re-Raises (this player already has 20 cents in as the big blind. Therefore, he completes the bet of 40 cents, from Player “D”, and adds one additional bet for a total of 80 cents)

Player “C” - Folds (his previous call of 20 cents is now in the pot)

Player “D” - Calls (matches the bet of Player “B”, by adding 40 cents to his previous bet, for a total of 80 cents)

End of betting round – it is time to deal *the flop*

Example of Betting Round 3 – Every player *folds*

There are six players at the table (Fig. 2.1):

Player “F” - *Dealer*

Player “A” - *Small blind* (10 cents)

Player “B” - *Big blind* (20 cents)

Start of betting round – *pre-flop*

Player “C” - *Folds*

Player “D” - *Folds*

Player “E” – Folds

Player “F” (*Dealer*) – Folds

Player “A” (*small blind*) – Folds

Player “B” (*big blind*) – This player wins, since all the players have fold

End of the game

2.2 Hand Ranking

In this section the hand ranking is presents briefly, although it is not crucial for the design of the perception layer.

A poker hand is a set of five cards that identifies the strength of a player in a game of poker. The following card hands and descriptions represent the poker hand ranking, in descending order of strength.

Royal Flush: It is the best possible hand in standard five-card poker. Ace, King, Queen, Jack and 10, all of the same suit.

Fig. 2.3 - Royal Flush example (Hearts)

Straight Flush: Any five-card sequence in the same suit.

Fig. 2.4 - Straight Flush example (Hearts)

Four of a kind: Four cards of the same rank.

Fig. 2.5 - Four of a kind example (Ace)

Full House: Three of a kind combined with a pair. It is ranked by the trips (three of a kind).

Fig. 2.6 - Full House example (Ace-King)

Flush: Five cards of the same suit, but not in sequence. It is ranked by the top card.

Fig. 2.7 - Flush example (Hearts)

Straight: Five cards in sequence, but not of the same suit. The Ace can be either the lowest or the highest card. The Straight is ranked by the top card.

Fig. 2.8 - Straight example. In this case the Ace plays the lowest card

Three of a kind: Three cards of the same rank.

Fig. 2.9 – Three of a kind example (Ten)

Two pair: Two separate pairs and one kicker of a different rank. The kicker is used to decide upon a tie of the same two pairs.

Fig. 2.10 – Two pair example (Ace - Five as the kicker)

One pair: Two cards of the same value. Three kickers.

Fig. 2.11 – One pair example (Ace)

High card: Any hand that does not qualify as one of the hands above. The strength of this hand is ranked by the top card, then the second card and so on.

Fig. 2.12 – High card example

2.3 Positioning at the table

The position of a player at the table is always relative to the *Dealer*. That is one of the reasons the *Dealer* position rotates around the table in poker, so that every player has equal time in every position. Being in the *Dealer* position is the most desirable position in poker, since after the *flop*, the player in the dealer position will always be the last to act in each round of betting. In practice, that player will have the information concerning the bets of the other players, advantaging from that when making a decision.

The remaining positions around the table can be categorized as well, relative to the position of the *Dealer*. The first three positions to the left of the dealer are said to be in "early position" (Fig. 2.13 – red circles). These are considered the worst positions in poker, since these players have to act first without knowing how any of the other players are going to play their hands. The two positions before the dealer plus the dealer position itself are said to be in "late position" (Fig. 2.13 – green circles). Being in "late position" allows the

players to have more information about how the hand is going, since the majority of the table as acted already. The remaining positions, between the “early position” and the “late position”, are said to be in the “middle position” (Fig. 2.13 – yellow circles). These positions are neither too good nor too bad, since a player in this position can see the cards of some players, but there is still the same number of players to follow him [5].

During the game play the “Dealer” position is represented, typically, by a white plastic disc with the word "Dealer", which is called “Dealer Button” (Fig. 2.13 a) and Fig. 2.13 b) - marked with a green circle). Although not officially used, sometimes can be seen as well the “Small Blind Button” and the “Big Blind Button” (Fig. 2.13 b) – the purple and yellow discs respectively, marked with a red circle).

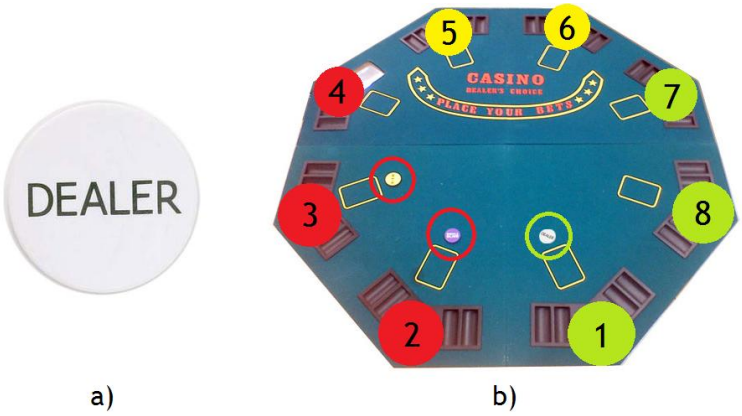


Fig. 2.13 – a) “Dealer Button”; b) Example of the positions relative to the “Dealer” position, which has the number 1 assigned. The red circles represent the “early positions”, yellow circles the “middle positions” and green circles the “late positions”.

2.4 Conclusion

This chapter introduces briefly how the game play of Poker *Texas Hold’em* is processed. However, it should be enough to fully understand the design of the algorithm. For further information please refer to [6, 7, 8].

Chapter 3

State of the Art

There has been relatively little work on playing cards' or chips' recognition based on vision. Instead, the leading technology in this type of recognition and adopted by the gaming industry, e.g. casinos, relies on RFID.

This chapter starts by introducing the broadcasted televised poker tournaments since it represents the major application of video capturing to official poker games. Moreover, non vision based systems for cards and chips recognition will be introduced. Finally some relevant works concerning cards and chips recognition based on vision systems are described as well.

3.1 Televised Poker

For many years, the televised poker coverage was not as robust as it is today. In the early years, viewers at home could not see which cards the players had or follow their progress visually through statistics. Instead, only the overview of the table was covered and the *community cards* displayed (Fig. 3.1 a)). No emotion was transmitted to the viewer and the feeling of watching a poker television show was similar to being in a casino watching the players gamble. At some point the *hole cam* [9] was introduced and, as the name suggests, it was designed to capture the *hole cards* of a player in order to display them to the audience. The introduction of the *hole cam* in televised poker tournaments improved completely the feeling of watching tv poker shows, gathering audience worldwide. Nowadays, the *hole cam* is considered one of the components responsible for the enormity of the poker boom in the last decade [10].

When adapting the *hole cam* to a poker table, its placement can be under a glass panel embedded in the table and players are instructed to place their cards over the glass (Fig. 3.1 b)); while in other cases the rail of the table is elevated and a small camera is placed within the rail and players are instructed to examine their cards in front of the camera (Fig. 3.1 c)). Moreover, the use of a *hole* cam, provided the ability to display interactive statistics and odds of each player, giving a "live sports feel" to the audience (Fig. 3.1 d)).

It is worth mentioning that, although televised poker is intended to provide the audience with a "live feel", actually the majority of the broadcasted poker tournaments are not live. Even the live tournaments have a small delay before their broadcast to the television [11]. This measure is intended to avoid any kind of cheating during the game play. Therefore an

automatic recognition of playing cards would be interesting but not critical, reasoning why the majority of the televised poker does not use automated card recognition systems. Instead, the cards are recognized manually, as well as the display of the players' statistics [11].

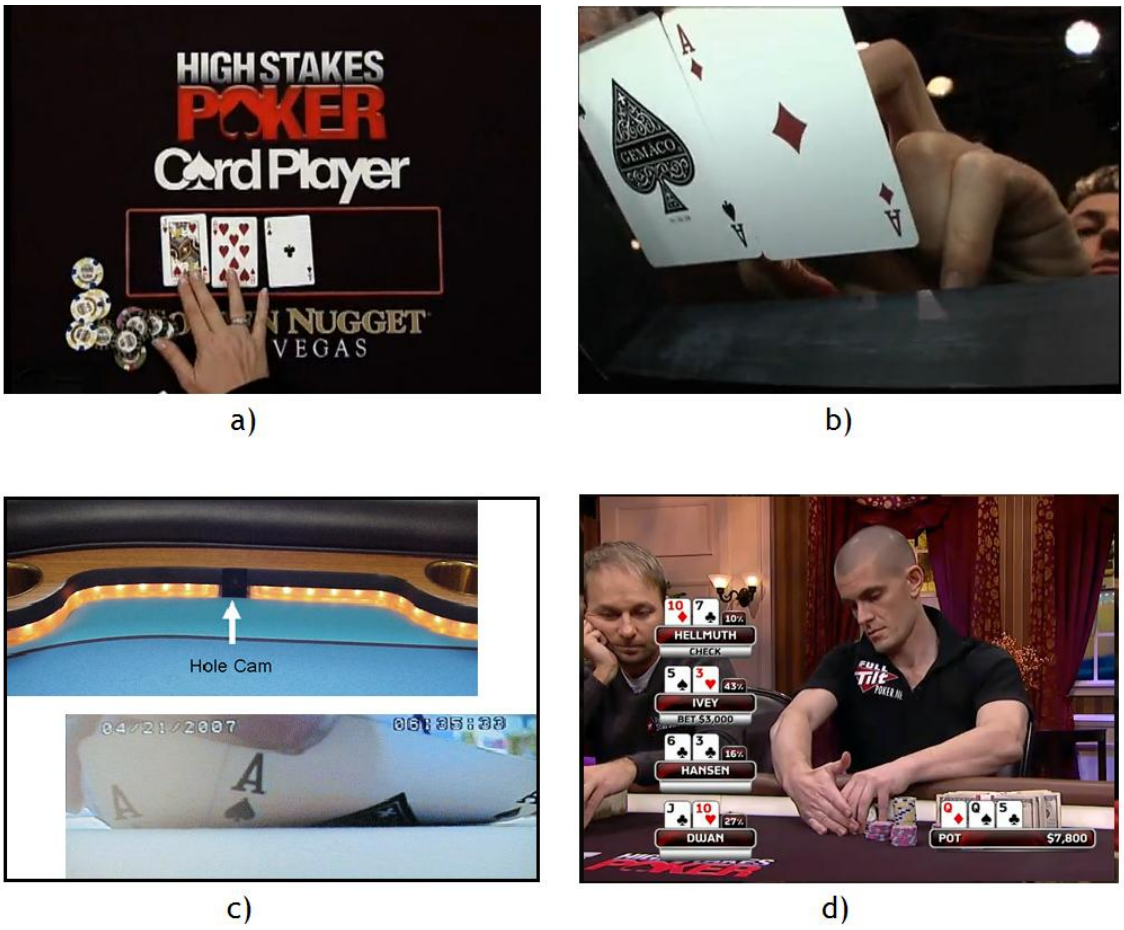


Fig. 3.1 – Televised poker. a) top view of the zone of the table where the community cards are dealt; b) *pocket cam* positioned under a glass panel embedded on the table; c) *pocket cam* positioned in the rail of the poker table and correspondent view of the cards; d) information concerning the hole cards of each player and displayed to the audience

3.2 Automated Card Recognition Systems

In the context of this work this section is split in two parts, the non vision based systems and the vision based systems. The former describes relevant projects and, mainly, commercial products. The latter describes the advances so far in the vision processing field when applied to recognition of playing cards and chips.

3.2.1 Non Vision based systems

So far the solutions for automated card recognition not based on vision systems, namely RFID and Bar Code, are proven to be the most reliable. Therefore, such solutions can be found in commercialized products which aim to automatically identify the cards and chips on a poker game. These techniques can be seen as a valid improvement to implement in televised poker.

RFID (Radio Frequency Identification)

RFID is a technology which uses communication via radio waves to exchange data between a reader and an electronic tag attached to an object, for the purpose of identification and tracking [12].

In 2004, Lewis et al. [11] patented a recognition system based on RFID adapted to a poker table, aiming to improve the recognition of cards during the broadcast of poker events to television, thus replacing the *pocket cam*.

In 2008, Andrew Milner, developed a poker table capable of automatically recognize cards by using RFID technology [13]. Milner tagged the playing cards with 52 RFID 'smart labels'. Moreover the RFID antennas are installed on the baseboard of the poker table. This project was covered by the online media and had positive feedback all over the internet.

Tangam Gaming [14] produced an automated card recognition system named MindPlay21. This system relied on a range of specialized hardware which included 14 cameras, invisible ink, and RFID tags. Cameras were used to scan the cards as they were dealt, as each card had been marked with a unique barcode painted in special ink [15]. The cost of $20,000 per table, the unreliable components and the slow speed of operation led the company to discontinue MindPlay21 in 2005.

Poker Tronic GmbH [16] and Flashlight Design Electronic System GmbH [17], both commercialize poker tables capable of identifying playing cards, where the latter's table is able to identify chips as well. Both rely on RFID technology.

a) b)

Fig. 3.2 – a) PokerTronic's table, where it is possible to see the RFID readers in the first half of the table; b) Flashlight Design Electronic System's poker table, the RFID readers can be spotted by the dark grey squares;

The RFID implementation in this context allows automatic recognition of the cards and/or chips without a camera. It is not necessary to lay the cards side by side on a glass plate, or to hold all the cards into a rail camera. Therefore, the RFID brings advantages to the televised poker tournaments concerning the identification and following the cards. However, a system of this kind requires major modifications to the poker table in order to install the *TAG readers*. The cards and chips need to be adapted with RFID *tags* as well.

Barcode

In 2010, Press Association Sport introduced an automated card recognition system named "Perlego" [18], which is based in infrared barcode and camera technology. In this system an invisible but reflective barcode is placed in the back of each card, which enables the system to read the face-down cards the moment they are dealt, even before they have been picked up.

This system has been used already, mainly for trial tests, in broadcasted poker tournaments [19]. If it proves to be reliable and accepted by the audience, it can even substitute the hole cameras introduced years ago when it comes to card recognition.

3.2.2 Vision Based Systems – Relevant Works

Zaworka [20] in 2001 tracked a Blackjack game by detecting cards and players' chip stacks as they are bet, in real time. Overall accuracy was 97.5% for detecting playing cards and chip stacks, even with occlusion. However, the system only detected the presence, not the values, of cards and chip stacks. Moreover, the system did not rely solely on vision since it used an electronic chip tray.

Zheng [21] in 2007 proposed a scale and rotation invariant recognition system, based on template matching. Despite the recognition rate was 99.8% over a range of rotations, the presented technique did not handle face cards well. Moreover, chips identification was not covered and a final usable system was not designed.

Hollinger and Ward [22] built a card recognition system using computer vision for card game. The recognition used relies on counting the suits, in case of the value cards, and on a neural network to differentiate between the king, queen and jack. Concerning the results, the authors claim that *"Overall, this project was a great success (...)"*, however the results of their card recognition system are not stated.

Zutis and Hoey [23] implemented a vision system aiming to detect card counters in the game of Blackjack. A combination of contour analysis, template matching and the SIFT algorithm was used to detect and recognise cards. Stereo imaging was also implemented, in order to calculate the height of chip stacks on the table, allowing the system to determine the amount of each bet. The card recognition accuracy claimed is over 99%. For the system to work properly, the camera has to be placed on top of the table and perpendicular to it.

Chen [24] used a four-orientation connectivity run-length value (FOCRLV) method to calculate connectivity features. Additionally, a weighted compacted energy band feature using "discrete wavelet transform" and "discrete cosine transforms" is used for improving poker image recognition.

In 2007, Sony Computer Entertainment launched The Eye of Judgement [25], which is a turn-based card battle video game. It fuses special playing cards with videogame technology. The game play relies on placing the cards on a board, which are then captured and recognized by the system setup. Each card features a special pattern easily recognizable by the camera. However, in middle of 2010 Sony Computer Entertainment shutdown the online services for this game [26], due to the not so good acceptance among the gamers.

3.3 Conclusion

In this chapter were described the techniques currently used as well as relevant works, aiming to recognize playing cards and casino chips.

In the context of the present work, it is possible to conclude that, although the RFID has been in the market for more years and proven to be more reliable than vision based systems, it is not a viable solution for this work. Not only it differs from the initial concept, but also reasoned by the impossibility of adapting it to any poker table. A vision based system at the same time makes the whole implementation on the table much more transparent from the point of view of the human opponents and capable of “sit and go” in almost any ordinary poker table.

Concerning the vision based systems, it is worth mentioning that all the relevant works on that field were all designed and implemented considering that the camera is positioned perpendicularly to the table and, therefore, to the cards. Moreover, none of the described works implement the perspective invariant feature, which is approached in this thesis.

Chapter 4

System Analysis

In this chapter, the description of the hardware and software may help to better understand some of the decisions made, as well as to be familiarized with the complete setup. Here one can find the details on the equipment found in an ordinary poker game and used in this work as well, consisting of the deck of playing cards, the casino tokens and the poker table. Still within the hardware, the capturing devices are introduced. The modifications performed to the hardware, namely the poker table, are described as well.

Moreover, the system's requirements are reviewed, as well as the design of the perception layer.

4.1 Hardware

Playing Cards

There are different kinds of playing cards, however the most used and standardized in poker, and used along this work, is the Anglo-American kind (Fig. 4.1). These are constituted by fifty two different playing cards, which include thirteen ranks of each of the four suits, diamonds, spades, hearts and clubs. Each suit includes an ace, depicting a single symbol of its suit; a king, queen, and jack, each depicted with a symbol of their suit; and ranks two through ten, with each card depicting that number of symbols of its suit. The common size for this type of cards, poker version, is 63 x 88 (mm).

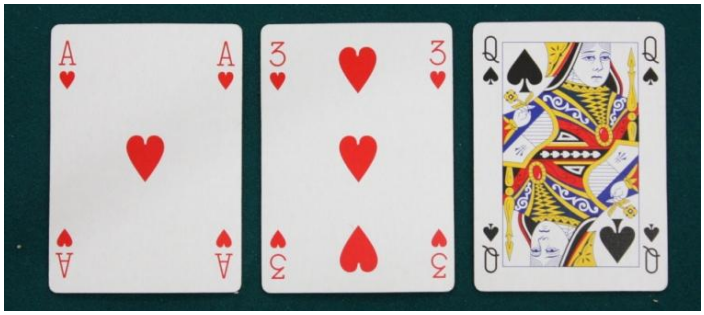


Fig. 4.1 –Cards from the main deck used in this work

Casino Chips

Casino chips, also known as casino tokens, are small discs used as the currency in casinos, as well as in casual and tournament games.

Chips are employed in casinos for several reasons. They are more convenient to use than currency, and also make theft and counterfeiting more difficult. Because of the uniform size, regularity and colouring patterns of stacks of chips, they are easier to count in stacks compared to paper currency when used on a table. Furthermore, it is observed that players gamble more when using chips than when using cash. Nowadays, the chips are considered to be part of the casino environment, therefore replacing them would be unpopular [27].

The amateur use of chips, i.e. in casual tournaments, is reasoned as well for being more convenient than using money. Moreover, it gives to the amateur player a similar feeling as playing in a casino, making the game play more enjoyable.

In this work, the chips used are made of plastic and are proper for casual games. A total of five different types of chips are used, differing in colour from each other and where each colour is associated with a monetary value (Fig. 4.2). Although in amateur games the value assigned to each colour is arbitrary, often the assignment is ordered as shown in Fig. 4.2.

Fig. 4.2 – Chips used along the work. These are ordered by value, where the white chip has the lowest value and the black chip represents the highest value

Poker Table

A poker table, as the name suggests, is a table specifically designed for playing poker. It can be either an actual table or a fold-out tabletop surface. The ones used in casinos are usually oval-shaped, with the players sitting around a curve of the table with a dealer facing them in an indented area of the table (Fig. 4.3 a)). In amateur poker, the table can be either an actual table or a fold-out tabletop surface. Amateur poker tables are often round or octagonal and use a rotating dealer position, i.e. there is no specific place for the dealer since there is not a dedicated *Dealer* with the task of dealing the cards, has it happens in professional poker.

The table used along this work is a poker tabletop surface, thus amateur type, octagonal shaped and foldable, Fig. 4.3 b). It measures 122 x 122 (cm) and features 8 seats for players. In each seat there are pre-defined spots, square shaped, which mark the place where the player should put his *hole cards*.

Fig. 4.3 – a) example of a professional poker table, similar to the ones used in casinos; b) amateur poker table – fold-out tabletop surface used to implement the work

Video Capturing Device

With the purpose of capturing the complete game play, a total of two we*bcams* were used. The first one is intended to get an overview of the poker table where the *community* part of the game is happening, while the second one acts as a *pocket cam*, i.e. a camera that displays the player's face-down cards.

The *pocket cam* chosen for this work features a native resolution of 640x480 pixels, which is the resolution used as well. The *webcam* which captures the overview of the table features a resolution up to 1280x720 pixels, but for computational cost's reasons only the resolutions of 640x480 and 800x600 are used and tested.

It is worth mentioning that when changing the resolution of the camera, while maintaining the same point of view, the region captured is different. More specifically, when setting the resolution of the capturing to 640x480, the region captured will be a crop of the image when it is set to 800x600. Moreover, the same region captured at 640x480 in one situation, occupies about 690x525 when the resolution is set to 800x600. Please refer to Fig. 4.4 for a better understanding. This fact is related with the combination of the pixel size, resolution and optical format of the webcam's sensor. For further explanation please refer to [28]. This situation should be taken into account during the analysis, when comparing the tests' results of different resolutions.

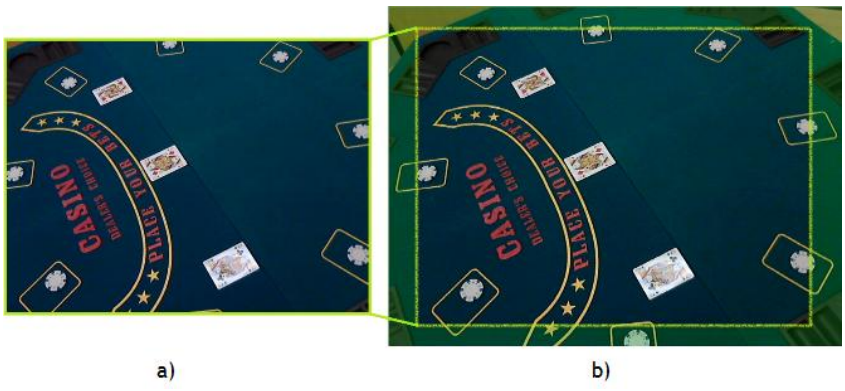


Fig. 4.4 - Region captured by the *webcam* in a fixed position, but set with different resolutions, where a) is set at 640x480 and b) is set at 800x600.

4.2 Software

The scope of this thesis limits itself to the study of the perception layer for the poker game play. Although the decision layer is briefly introduced, as follows, it is part of another work [1], therefore, this thesis involvement in such context was always kept to a minimum.

Poker Agent

A Poker Agent is a piece of software that contains a Poker AI that is used to play Poker autonomously without human interference. In the context of this thesis, the poker agent is part of the decision layer and is provided with the game play information determined in the perception layer. Analogously speaking, the poker agent can be thought as being the brain. The Poker Agent adopted in this work was developed in a previous work [1].

Poker Simulation Platform

A poker simulation platform is capable of simulating poker games automatically, where the players can be either human or automated poker agents. A platform of this kind was adopted in [1] to create the environment needed for the poker agent to play.

It is worth mentioning that it would be of interest to test the work developed in real gaming conditions with human opponents, aiming to obtain the most trustworthy results as possible within this work's concept. However, the current development of the whole concept has not reached yet the physical autonomy, making it impossible to accomplish some tasks such as placing chips on the table. Therefore the simulation platform overcomes the physical issues, when it comes to test the implemented work.

Further information regarding the simulation platform can be found in [1].

4.3 Modifications

Although all the equipment used is off-the-shelf, aiming to create the most realistic environment as possible, some adaptations were needed to adapt the concept of this work to real life and its conditions. These modifications aim to overcome the motionless property of this system, enabling it to capture the complete game play from its stationary position. The modifications performed tried to be as smooth as possible in order to maintain the essence of the game from the opponent human players' point of view.

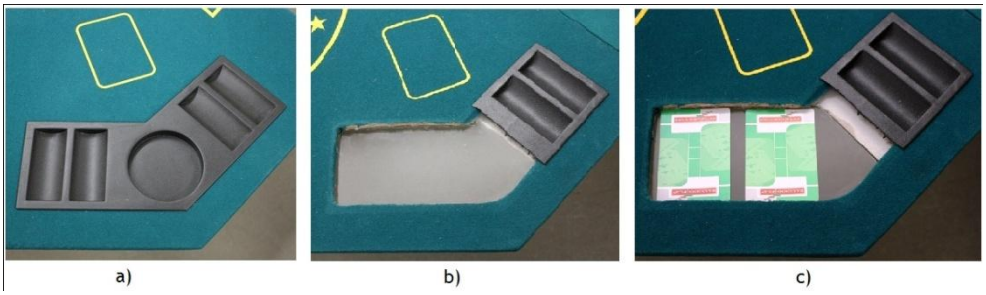
Poker Table

During a poker game, the players' cards, *hole cards*, are facing down when lying on the table. Whenever a player wants to look at the cards to see which hand he has, he grabs them from the table and looks at them. However, in this thesis, implementing something of that kind is out of context since the main idea relies on a motionless system. On the other hand, it is not interesting to have the cards facing up, since it would reveal the hand to the opponents, taking the essence of the game. In order to overcome this situation and provide the required conditions for the *pocket cam* to operate successfully, the poker table was slightly modified. This modification was inspired in the tables used in televised tournaments, in which the camera is placed under an embedded glass panel in the table, Fig. 4.5.

a) b)

Fig. 4.5 – a) example of a poker table adapted for televised tournaments; b) *pocket cam* capturing the player's cards

The modification performed to the poker table relied on swapping the cup and chips holder by an acrylic panel, Fig. 4.6.



a) b) c)

Fig. 4.6 – Modification done to the poker table. a) before - with the chips and cup holder; b) after - chips and cup holder partially removed and acrylic panel adapted; c) after – with the cards facing down in the acrylic panel

Tripod

A tripod was adopted with the purpose of holding both *webcams,* Fig. 4.7 a). It simplifies the task of setting up the whole system, more specifically the *webcams'* position, every time it is needed to do so.

One of the *webcams* was mounted on the top of the tripod, Fig. 4.7 b). When setting up the height of this camera, must be taken into account that it must be high enough to capture the overview of the poker table; more specifically, the goal is to capture neither more nor less than all the pre-defined marks for the players' hole cards depicted on the table.

The second camera, the *pocket camera*, is intended to capture the *hole cards;* hence it was mounted on the lower part of the tripod, more specifically in one of its legs, Fig. 4.7c).

a) b) c)

Fig. 4.7 - a) used tripod with the *webcams* mounted – red circles; b) *webcam* installed on the top of the tripod – captures the overview of the poker table; c) *webcam* installed on the lower part of the tripod – used as the *pocket cam*

a) b)

Fig. 4.8 – Examples of captured images from both cameras used; a) top mounted *webcam;* b) lower mounted *webcam*

4.4 Functional Requirements Specification

The functional requirements of this work are based on the concept presented in Chapter 1 and in what is expected from the system.

From the implementation point of view, it helps to keep focus on the important parts of the system when designing and implementing it, by prioritizing some functionalities over others.

Table 4.1 - Functional Requirements

Nr.	*Requirement*	*Priority*
1	Card rank identification from video stream	High
2	Card suit identification from video stream	High
3	Counting of different coloured chips present on the table	High
4	Counting of players in the game and respective positions	Medium
5	Determine the position of the *Dealer*	Medium
6	Identification of the players' actions during the betting round by using Speech Recognition	Medium
7	Stand alone system by following the complete game play	Medium

4.5 System's Design

The complete system is designed as follows:

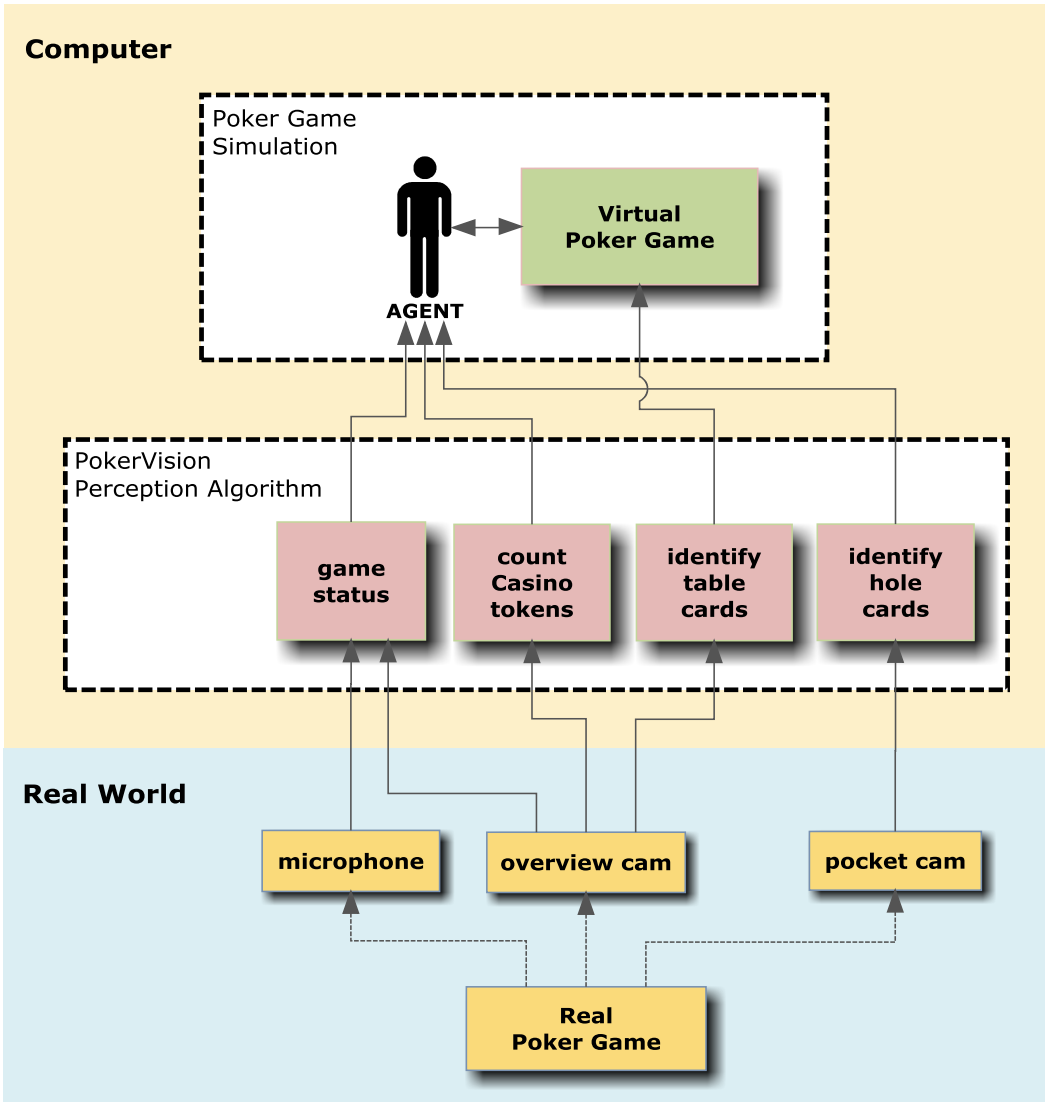


Fig. 4.9 - System Design

The design of the whole system can be split in two parts, the hardware part and the software part.

The first consists of every element physically existent, which is represented in Fig. 4.9 by the “Real World” sub-division. This part includes both the hardware inherent to an ordinary poker game such as the poker table, the playing cards and the casino tokens, as well as the hardware used which makes the perception possible, such as the microphone, the overview *webcam* and the *pocket webcam*.

The software part is present on the second sub-division of the system’s design – “Computer”. It represents both the software that was already implemented, i.e. the poker game simulator, as well as the algorithm implemented in this work, i.e. the “Perception Algorithm”. This “Perception Algorithm” is sub-divided into four blocks, which constitute the main parts of the implemented algorithm and, therefore, can be seen as corresponding to the functional requirements described in 4.4 Functional Requirements Specification.

It is worth recalling that the “Poker Game Simulation” depicted in Fig. 4.9 is not covered in the context of this thesis.

4.6 Conclusion

This chapter was intended to describe the setup used, so the approach taken in each step of the implementation is better understandable. Although the implemented algorithm tries to be as robust as possible, the setup detailed was always the reference during the design of the algorithm.

Chapter 5

Recognition of Playing Cards and Chips

This chapter presents the approach to identify the rank and suit of each card, as well as the approach to determine the total amount of chips on the table.

Before having an overview of the approach, the reader should be aware that the algorithm about to be presented was always focused on the setup described in Chapter 4. This means that one can only expect the most trustworthy output from the system when the setup is the same or, at most, with some minor differences. There is no "one solution fits all", but the approach shown here is the skeleton of the solution and can be easily adapted to meet any conditions imposed.

Concerning the system's adaptability, in order to ease any future modification or adaptation, the algorithm is designed as a set of blocks. Each block, containing a well defined task, can be edited, improved or replaced. Later, this modular design will also allow analysing the experimental results clearly and coming to a conclusion about the reliability of each block.

5.1 Introduction to Low Level Image Processing

In this section, common techniques of low level image processing are described, such as smoothing, image binarization and linear stretching. These techniques are frequently used along the work supporting this thesis.

The need for low level image processing arises when there is a too large set of data and there is the need to simplify the analysis by reducing the amount of data to be processed, while at the same time preserving useful structural information [29]. This can be done either by removing part of the data available or by emphasizing the interesting regions.

In the context of this work, although there are no critical time constraints to image processing and feature detection, all the work is directed to be as much as region-of-interest oriented. This will both reduce computational cost and at the same time prevent misidentifications, where applicable.

5.1.1 Smoothing

Smoothing is a simple and frequently used image processing operation and it is usually done to reduce noise or camera artifacts, i.e. distortion. Diverse kinds of smoothing exist for specific applications, however the Gaussian filtering was the most used along the developed work [30].

Mathematically speaking, the Gaussian filtering is done by convolving each point in the input array, i.e. input image, with a Gaussian kernel, example in Fig. 5.1 b). The kernel used is a discrete approximation of the Gaussian distribution, example in Fig. 5.1 a), which puts more weight towards the central pixel. Because of this, a Gaussian smoothing provides gentler smoothing and preserves edges better than similarly sized filters. The Gaussian filter also acts as a low-pass, meaning it removes high spatial frequency components from an image [29]. In practice this will remove noise while preventing the edge detection from finding false edges.

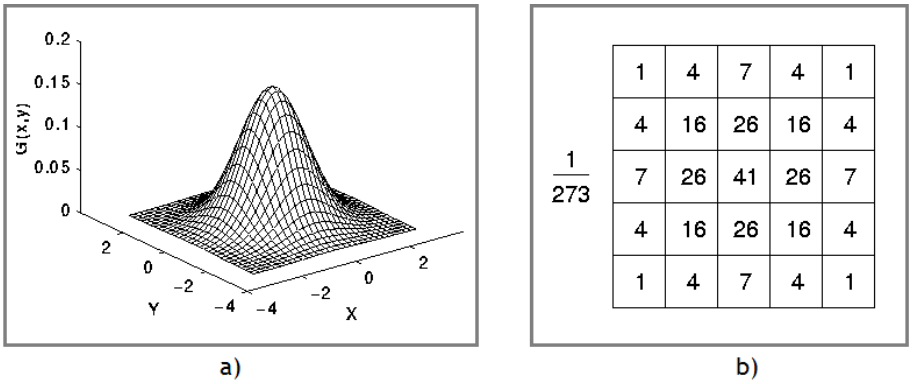


Fig. 5.1 a) example of 2-D Gaussian distribution with mean (0,0) and σ=1.0; b) discrete approximation to the Gaussian distribution in a)

5.1.2 Image Binarization

In many vision applications it is useful to separate out the regions of the image corresponding to objects, from the regions of the image that correspond to background. This is called image binarization and it is used to remove parts of the image that fall within a specified intensity range, by converting an image of up to 256 gray levels to a black and white image, i.e. either a level of 0 or 255 [29].

The simplest way to use image binarization is called Thresholding, which consists in choosing a threshold value and classify all pixels with values above this threshold as white - foreground, and all other pixels as black - background. However, in many cases finding one threshold value compatible to the entire image is impossible, mostly when there is a strong illumination or reflectance gradient, where it is needed to threshold relative to the general intensity gradient. In the example of Fig. 5.2 the image in a) was binarized with the threshold level set to 80 and 120, resulting in b) and c) respectively. Clearly, simple thresholding does not provide good results in this situation, due to the severe illumination gradient across the image [29].

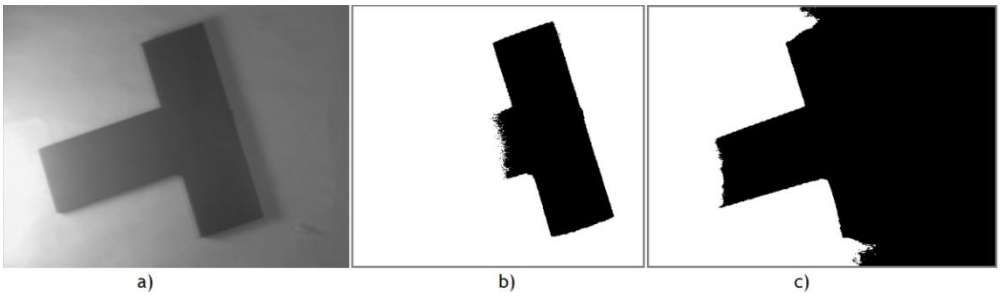


Fig. 5.2 – Examples of poor image binarization using one level Threshold [29]

Therefore, Adaptive Threshold is adopted so the threshold level is itself variable for different regions in the image. This technique is set on a pixel-by-pixel basis by computing a weighted average of a region around each pixel location minus a constant. The region to be computed is defined by a kernel with a pre-defined size, which can follow a mean distribution or a Gaussian distribution. If the kernel follows a mean distribution, then all the pixels in the area are weighted equally. If the kernel follows a Gaussian distribution, then the pixels in the region around the pixel to be computed are weighted according to a Gaussian function of their distance from that centre point (Fig. 5.1).

5.1.3 Contrast Stretching

This technique, also called normalization, can be applied to images where there is a substantial lack of contrast. This method attempts to improve the contrast by "stretching" the range of intensity values it contains to span a desired range of values, *e.g.* the full range of pixel values from 0 to 255 in the 8 bit images [29].

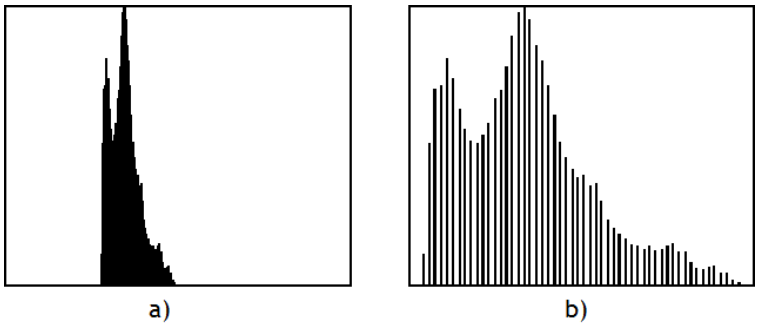


Fig. 5.3 – Example of a histogram before a) and after b) the linear stretch

In the context of this work, the normalization is used to stretch the histogram of the gray level images, in order to use the complete range of brightness from 0 to 255, Fig. 5.3. When applied, the function scans the image to find the lowest and highest pixel values present in the image [30]. Thereafter, the brightness of each pixel is scaled using the following function:

$$dst(i,j) = \frac{(src(i,j) - \min(src)) \times (b' - a')}{\max(src) - \min(src)} + a' \quad (5.1)$$

Where:
b'=upper limit of the contrast – 8bit image: 255
a'=lower limit of the contrast – 8bit image: 0

min(src) and max(src) are the global minimum and maximum, respectively, of the input array, computed over the whole array or the specified subset of it.

5.1.4 Mathematical Morphology

The aim of the morphological operations is to transform the images into simpler ones either by removing irrelevant information, such as noise or non interesting objects, or by joining disparate elements in an image. This operation is often done on binary images that result from thresholding [29].

The basic morphological transformations are called dilation and erosion, which are both commonly used along the work.

Binary Dilation

The basic effect of the dilation operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels. Thus areas of foreground pixels grow in size while holes within those regions become smaller.

Mathematically, dilation is a convolution between two pieces of data inputs. The first is the image which is to be dilated, while the second is a set of coordinate points called structuring element, also known as kernel. This kernel determines the precise effect of the dilation, which is that of a local maximum operator. As the kernel is scanned over the image, it is computed the maximal pixel value overlapped by the kernel. This maximal value computed replaces the image pixel value overlapped by the centre cell of the kernel [30]. In practice, this causes bright regions within an image to grow. The dilation of A by B is denoted by A $\oplus$ B and is defined by:

$$A \oplus B = \cup_{b \in B} (A)_b \quad (5.2)$$

Binary Erosion

Erosion is the inverse of dilation, therefore the basic effect of this operator on a binary image is to erode away the boundaries of regions of foreground pixels. Thus, areas of foreground pixels shrink in size and holes within those areas become larger.

The action of the erosion operator is equivalent to computing a local minimum over the area of the kernel. As the kernel is scanned over the image, it is computed the minimal pixel value overlapped by the kernel. This minimal value computed replaces the image pixel value overlapped by the centre cell of the kernel [30]. In practice, this causes bright regions within an image to shrink. The erosion of A by B is denoted by A $\ominus$ B and is defined by:

$$A \ominus B = \cap_{b \in B} (A)_{-b} \quad (5.3)$$

5.2 Playing Cards' Recognition

The playing cards' recognition algorithm consists of three main stages, Fig. 5.4. The first stage finds and extracts/isolates the cards present on the captured image of the table overview. The proceeding blocks, "identify rank" and "identify suit", process each card individually and output the identified rank and suit respectively.

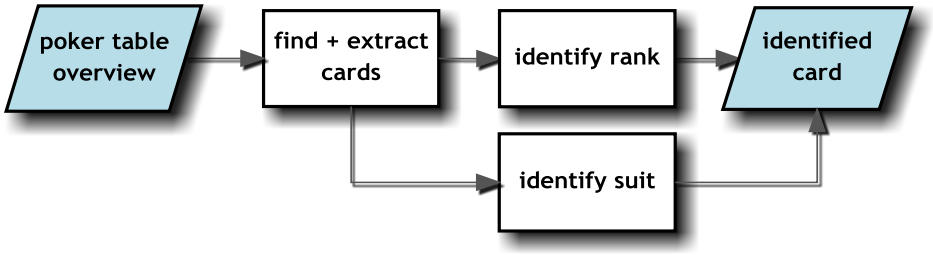


Fig. 5.4 – Card Recognition algorithm overview

5.2.1 Card Finding and Extraction

The diagram in Fig. 5.5 represents the high level overview of the algorithm responsible for finding and extracting the cards that are present on the captured image.

This algorithm relies on the great contrast between the poker table and the cards lying on it, where the former features dark colours, e.g. dark green, and the latter has always a white colour. The greater the contrast between the cards and the poker table, the stronger the edges between both will be, i.e. the border between the card and the table.

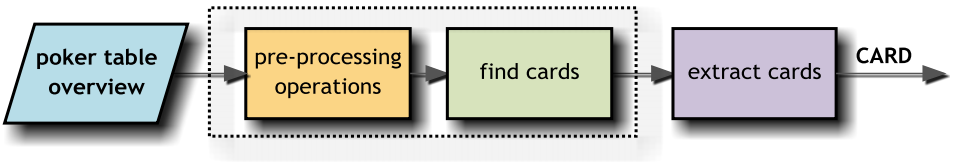


Fig. 5.5 – Card Finding and Extraction overview

Pre-Processing Operations

The first part of the algorithm, the image pre-processing operations (Fig. 5.6), is where the captured image is submitted to low level image processing. After converting the image to gray scale, the "Gaussian Smoothing" helps removing unwanted noise naturally present on the input image. In order to increase the contrast between both elements, the overall contrast of the image is enhanced by a linear stretch to the histogram. In this specific case, the contrast enhancement has the purpose of providing the edge detector block (Fig. 5.7 – first block) with strong edges, i.e. boundaries with a high gradient.

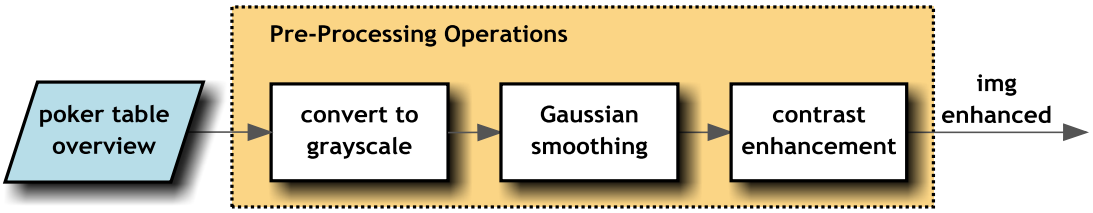


Fig. 5.6 - Pre-Processing Operations block in detail

Card Finding

The "Card Finding" block aims to scan the captured image of the table overview for cards. In case these are found, the block outputs their corners' coordinates. Determining the presence of cards relies on edge detection. Edges in images are areas with strong intensity contrasts, i.e. abrupt change in intensity from one pixel to the next. Such contrast is expected considering the boundary between the poker table cloth and the card.

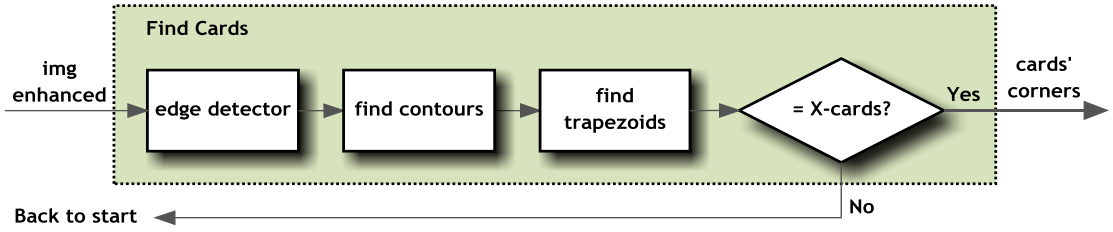


Fig. 5.7 – Find Cards block in detail

The edge detector used is the Canny edge detector [31]. The Canny algorithm computes the first derivatives in x and y coordinates and then combines them into four directional derivatives. The points where these directional derivatives are local maxims are then candidates for assembling into edges. Thereafter it tries to assemble the individual edge candidate pixels into contours, by applying a hysteresis threshold to the pixels. This hysteresis helps to ensure that noisy edges are not broken up into multiple edge fragments. The edge detector produces as output an image showing the positions of tracked intensity discontinuities [30].

From the resulting edges, the "find contours" block scans for all the external contours present in the image. All the closed shapes, e.g. squares, are selected and the rest, e.g. lines, are discarded. In Fig. 5.8, it is shown an example of how the block works; on the left image, a), the red square and the red circle meet the requirements since both are external contours; marked in green are the inner contours and non contours to be discarded. The image on the right, b), shows the selected contours by the algorithm. The output from this block is a sequence containing all the sequences of points of all the external contours.

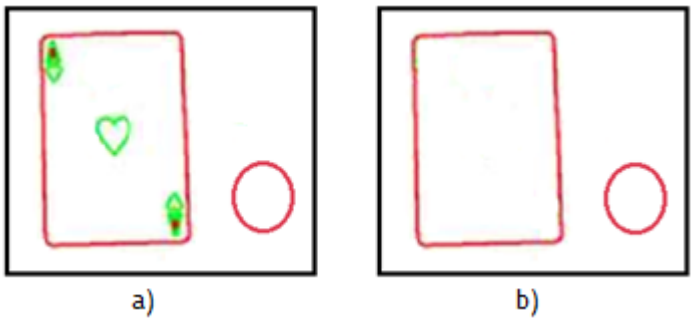


Fig. 5.8 – Representation of an image and its contours - a), and the same image with only the external contours represented - b)

Finally, the "find trapezoids" approximates all the contours by polygons. Those which have four vertices, corresponding to the four vertices of a card, are kept; the remaining polygons are discarded. At this point only trapezoids remain.

Since cards have the shape of a rectangle, thus four right angles, the algorithm filters the trapezoids by their inner angles. All the trapezoids which inner angles are close to 90º are considered to be cards, while the remaining trapezoids are discarded. Accepting inner angles within a range around 90º makes the system robust enough to be in any angle relatively to the cards. If all the inner angles had to be exactly 90º, it would be required to place the camera perpendicular to the cards on the table.

During a poker game round, there are three stages in which it is needed to identify the *community cards* - the *flop,* the *turn* and the *river;* where in each stage, the number of cards present in the table is respectively three, four and five. In practice, when scanning the input image for cards, it is expected to find the number of cards to be three, four or five. Therefore, the cards found with the previously detailed algorithm are only considered valid if its number is concordant with the number of cards expected. Otherwise, the cards' coordinates are not considered and the rest of the steps concerning card recognition are skipped.

In practice, this implementation is useful when a playing card is inadvertently occluded, thus not found. It also prevents situations where one of the cards is not detected due to noise present in the captured image. When one of these two situations occurs, the algorithm continues searching for the correct number of cards to be found.

Card Extraction

The "extract rank" block (Fig. 5.9) aims to isolate each one of the cards found with the purpose of simplifying the proceeding operations concerning the card identification. This algorithm consists of two parts, where the first one identifies the position of the card regarding the camera, while the second one reverses the perspective view of the card.

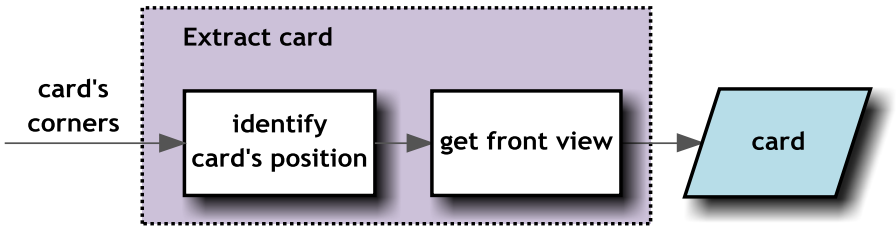


Fig. 5.9 – Card Extraction block diagram

When the tripod is setup in one of the seats around the table, the position relative to the cards is not calibrated. Therefore the PokerVision does not know in advance in which angle it is relative to the *community* cards (Fig. 5.10).

Fig. 5.10 – Examples of perspective views of cards

In order to make this system robust enough to be positioned in any place around the poker table, thus capable of extracting the cards whatever their position, the block "identify card's position" was implemented. The position of each card is computed using the coordinates' information of its corners and their position relative to each other and to the global coordinate system (Fig. 5.11). The first step to identify the card's position relies on numbering the card's corners. The algorithm starts by scanning the image for the corner which features the lowest y-coordinate (Fig. 5.11 – red circle) corner number one. In case two of the corners have the same y-coordinate, the algorithm chooses one of them to be the corner number one. Thereafter, it is determined the closest corner to the corner assigned before - corner number two. The third and fourth corners are determined as well, relative to the distance to the first corner (Fig. 5.11 – yellow and blue circles).

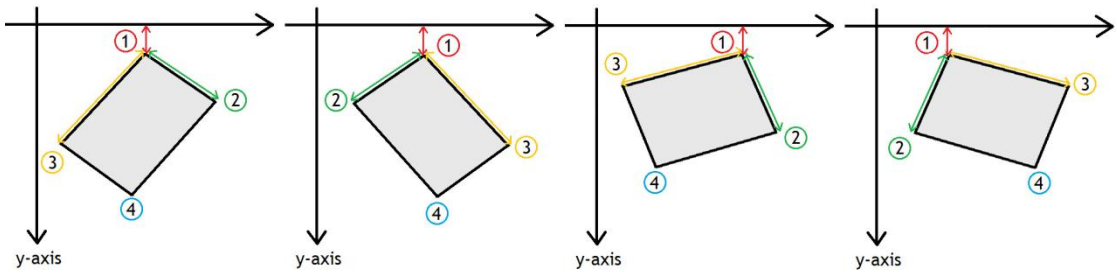


Fig. 5.11 – Possible positions of the cards

With the previous procedure – numbering of corners - one of the four situations depicted in Fig. 5.11 can occur. It is of interest to normalize the numbering so this information is

correctly used by the next block. The normalization procedure assures that the corner number one is always to the left of number two and corner number three is always to the left of number four. If not, these corners are swapped respectively (Fig. 5.12).

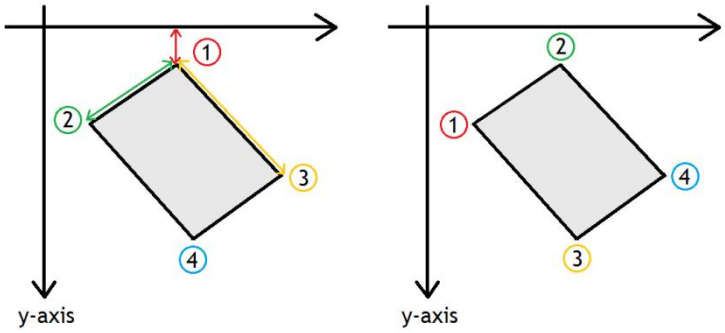


Fig. 5.12 - Example of corner numbering normalization, corner one swaps position with corner two and corner three swaps position with corner four

The numbered and ordered card's corners are used in the next block, "get front view". This block is intended to reverse the perspective view of the card, outputting the image of the card with the correct size proportions, as a regular playing card, and without the perspective view effect (Fig. 5.13).

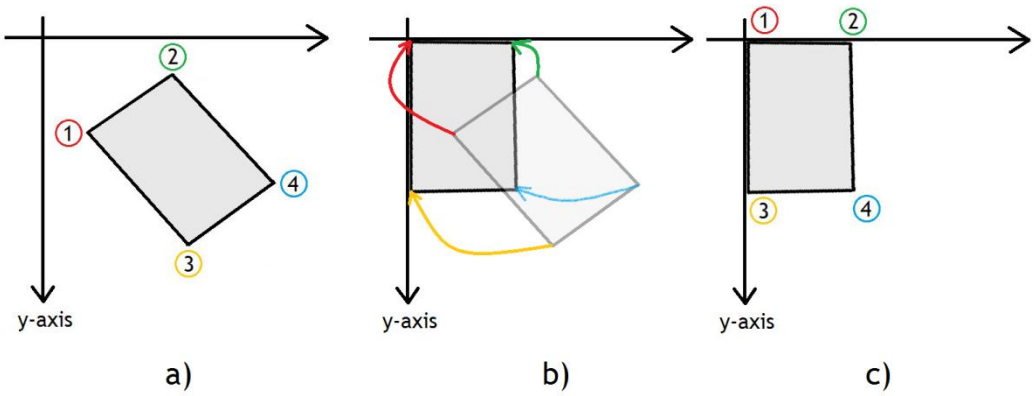


Fig. 5.13 - Inverse perspective process

The algorithm calculates the 3x3 matrix of perspective transform based on the starting points and on the destination points. The former correspond to the card's corners on the original image (Fig. 5.13 a)) while the latter correspond to the corners of the new rectangle shaped (Fig. 5.13 c)). It is now perceptible the importance of the normalization of the corners' numbering performed in the previous block. The calculated matrix contains the constants which will rotate, stretch and translate the original image.

$$(t_i \bullet \mathrm{x}_\mathrm{i}' , t_i \bullet \mathrm{y}_\mathrm{i}', t_i)^T = map\_matrix \bullet (x_i, y_i, 1)^T \tag{5.4}$$

where: $dst(i) = (x_i', y_i')$, $src(i) = (x_i, y_i)$, $i = 0..3$

With the matrix generated, it is now possible to reverse the perspective.

$$dst(x', y') \leftarrow src(x, y)$$

$$(t \bullet \mathrm{x}', \mathrm{t} \bullet \mathrm{y}', t)^T = map\_matrix \bullet (\mathrm{x}, \mathrm{y}, 1)^{\mathrm{T}} + \mathrm{b} \tag{5.5}$$

The result is a rectangle shaped card with the standardized proportions between width and height. The common size for poker cards, the ones used in this work, is 63 x 88 (mm) which represents a ratio of 0.72. This ratio was used to fix both width and height of the output image in 285 x 398 (pixels) respectively.

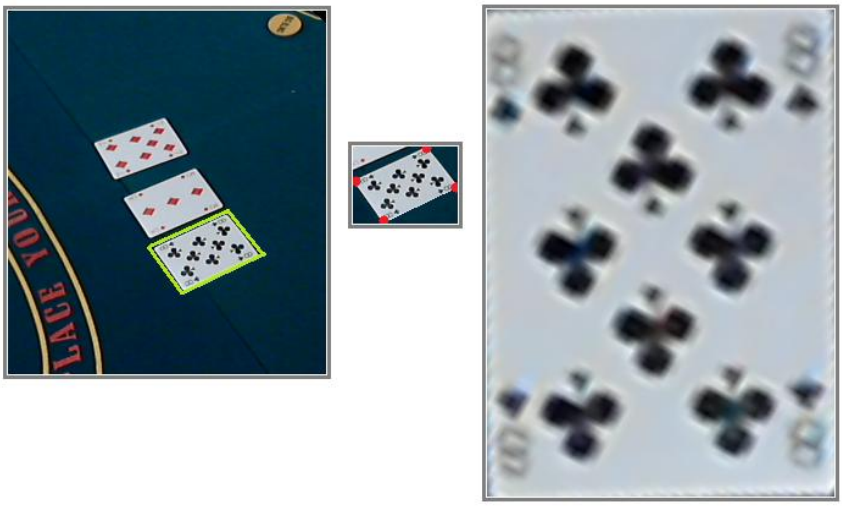


Fig. 5.14 – Starting image and resulting image after the "get front view" block. Although the image was reduced, the ratio between before and after is at scale.

In Fig. 5.14 it is noticeable the loss of quality or, at least, the low quality of the resulting card image. This result is due to the enlargement inherent in the transformation. The new area to be occupied is reconstructed from the original one using bilinear interpolation in which the loss of quality is inevitable.

5.2.2 Rank Recognition

In this section the reader can find two algorithm contributions for recognition of card ranks. The first approach, known as Template Matching, is based on the comparison between the original card rank and the pre-defined templates of each rank. The second approach is based on the counting of suits depicted on each card.

5.2.2.1 Template Matching approach

Template Matching is a pattern recognition technique that allows detecting the presence of a specific object in an image. It can produce reliable results but needs to know exactly what to look for since it is based on the comparison of the image against a pre-defined template. In the context of rank recognition, since the shape and size of the rank of the

cards is almost invariant, even considering different standardized card decks, this approach fits the problem and happens to be a promising choice theoretically speaking.

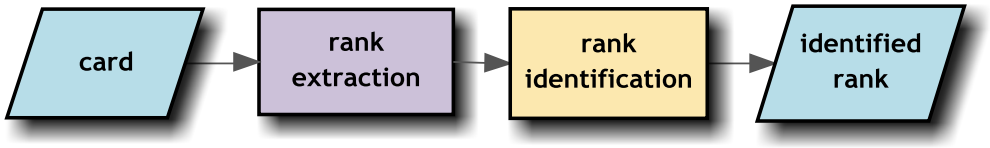


Fig. 5.15 – Template matching High level overview – block diagram

Card Rank Extraction

This block aims to extract two of the four ranks present on the card and prepare them for the template matching. By defining the region of interest, it enables the algorithm to restrict the region to be subject of template matching, benefiting the rank identification reliability. Not only it prevents possible misidentifications with something else drew on the card than the rank itself, but also enables the identification to compute much faster since the image area to be analyzed is reduced significantly. On the other hand, extracting two ranks instead of one makes the algorithm more reliable to illumination reflectance and makes the recognition more reliable since it is performed to two samples instead of one.

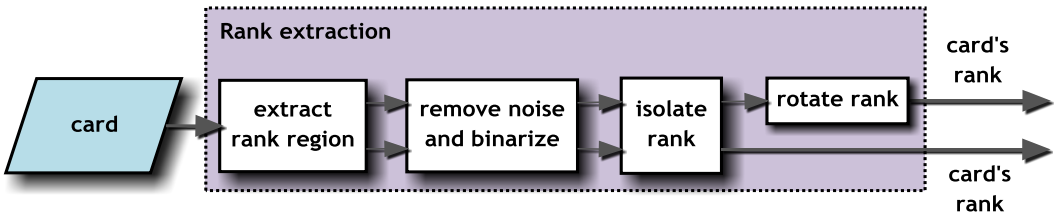


Fig. 5.16 – Card Rank Extraction - block diagram

By extracting two ranks from the same card and submitting both to recognition, it is expected the Template Matching to be more reliable since more samples per card are computed. Moreover, since the extraction occurs in opposite corners of the card, Fig. 5.17, it is possible to overcome situations where illumination reflectance affects one of the corners, blinding the camera in that region, but not the other.

Fig. 5.17 – Position of the ranks extracted from each card

The position and size of the rank is the same across the cards of the same deck. However, every time a new frame is captured that position is itself variable. This is due to the processing the card undergoes when it is extracted, *5.2.1 Card Finding and Extraction*. To overcome this issue, the cropped region features a bigger size than the rank's size, in order to prevent the rank from being sliced during this operation (Fig. 5.18 a) b)).

After the region of interest is extracted, the algorithm removes unwanted noise by smoothing it with a Gaussian filter. Thereafter binarizes the resultant image with an Adaptive Threshold. The resulting image after binarization is eroded in order to remove some remaining thin noise.

In order to normalize the position of the rank, along with removing the remaining noise, the image is submitted to the "isolate rank" block. This block aims to tightly isolate the rank of the card from the rest of the region previously extracted (Fig. 5.18 c)). This block searches for all the elements, i.e. contours, present on the binarized image and encloses them in a box. Afterwards, the algorithm considers the sizes of the enclosed contours and their centroids, discarding all the contours which feature a non reasonable size. From the remaining, it selects the one which centroid is the closest to the centre of the image. The resulting image features the rank tightly isolated (Fig. 5.18 d)).

Fig. 5.18 – Part of the rank extraction process, a) low part of the card, b) region of interest cropped in gray, c) binarized rank region still with unwanted edges, d) rank isolated from noise

In fact, the algorithm as described up until now has a major flaw, which relies on the card ranked ten. The rank "10" is constituted by two characters instead of one, which means the algorithm when iterating through the conditions referred above, discards one of the characters (Fig. 5.19 a-1) and a-2)). The work around for this issue consists in drawing one horizontal line 1pixel thick, wide enough to superimpose the characters 1 and 0 (Fig. 5.19 b-1) – red line). This misleads the algorithm which searches for contours, to interpret both characters as the same one since both are connected through the line.

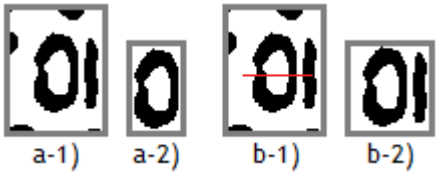


Fig. 5.19 – "10" upside down - a-1) and a-2) issue when isolating the rank, b-1) and b-2) solution found to keep both characters together

This step is performed in every card since it is not known previously if the rank to be isolated is the “10” or not. However, this does not affect the other ranks in any way. It is worth mentioning that the line drawn will not be present on the output image (Fig. 5.19 b-2)) in order to be as neutral as possible regarding the shape of the rank.

Finally, one of the extracted ranks is rotated in order to match the orientation of the templates used in the Template Matching.

Card Rank Identification

Template matching is a technique in digital image processing for finding small parts of an image which match a template image and consists in “sliding” the template over the input image (Fig. 5.20). The convolution output is lowest (better results) at places where the image structure matches the mask structure. This technique can be easily performed on grey images or edge images, or it can also be applied to full colour images with the trade-off of a heavy computational cost. One of the causes for the high computational cost is the necessity of using multiple templates to cover the appearance’s variability of the objects, in the case of rank identification it is necessary one template for each rank.

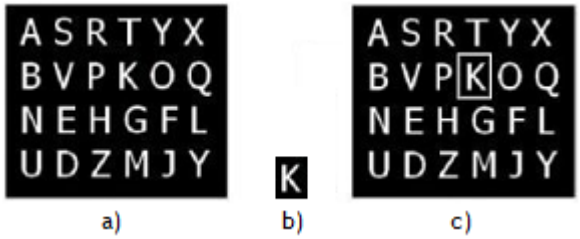


Fig. 5.20 - Template matching example [32]: Applying a matching in the search image (a) with the template (b) gives a maximum response in the center pixel of the surrounding rectangle in (c)

The Template Matching method is not scale and rotation invariant, which forces the implementation of additional measures intended to dispose the extracted rank with the same orientation as the templates available (Fig. 5.21). The implementation which overcomes the rotation and the perspective view of the card is described in 5.2.1. To overcome the “scale invariant” characteristic of this approach, the template match must be preceded by the adjustment of scale between the extracted rank and the template to be used. This task is performed by the “resize block”, which resizes the extracted rank to match the size of the templates used (Fig. 5.22).

Fig. 5.21 – The thirteen templates used for the template matching

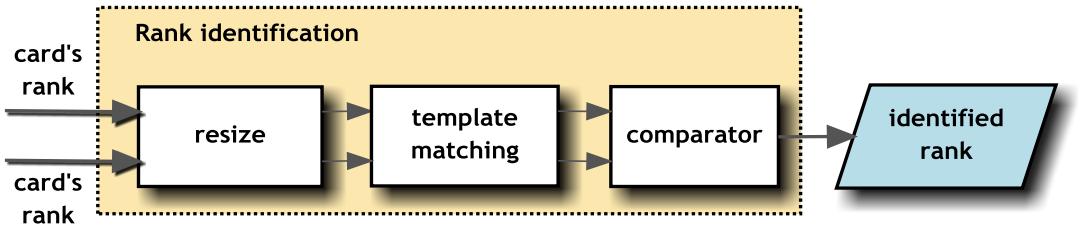


Fig. 5.22 – Card Rank Identification - block diagram

The resizing reduces as well the computational cost of the template matching procedure, since every time the template "slides" one pixel over the image, the match is computed. Therefore, if both the template and the image have the same size, then only one computation is performed. The number of iterations can be calculated as (W-w+1)x(H-h+1), where W and H are the width and height of the input image and w and h are the width and height of the template [30].

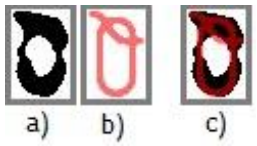


Fig. 5.23 – a) Rank extracted – real sample, b) template, c) both superimposed. Both represent the "Queen" rank.

Within the software used for the image processing, OpenCV, the computation of the match can be one of three types, Square difference matching, Correlation matching or Correlation coefficient matching. In the purpose of this work the chosen method was the Square difference matching, where a perfect match scores 0, while bad matches score larger values than 0 and up to 1.

$$R\_sqdiff\ (x,y) = \sum_{x',y'} [T(x',y') - I(x+x',y+y')]^2 \tag{5.6}$$

For each of the three methods just described, there are also normalized versions [33]. The normalized methods are useful on reducing the effects of lighting differences between the template and the image. In the context of this work, the normalized square difference matching was used:

$$R(x,y) = \frac{\sum_{x',y'}[T(x',y') - I(x+x',y+y')]^2}{\sqrt{\sum_{x',y'} T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}} \tag{5.7}$$

In order to prevent misidentifications, it was defined that every match which scores above 0.45 is not considered. If for any reason, after computing the match against all the

thirteen templates, the best score is above 0.45, it is considered that no match was found. This threshold was empirically defined.

Both the “resize” and the “template matching” are performed for both card ranks extracted previously, resulting in two identifications and respective Template Matching scores. The “comparator” has the task of selecting the identification which scores the best match and output it.

Improvement to the algorithm

It is known that captured images using an entry-level webcam are not noise free. Considering two consecutive captured images, it is most likely that both are different at the pixel scale, even considering there were not any changes concerning the position of the camera and/or scenario. This can greatly affect the reliability of the identification since a minor difference on the capturing can become a major difference after all the pre-processing the image is submitted to. Therefore, making the result of the Template Matching non consistent with the remaining iterations.

In order to assure that the identification is consistent along several captured frames, it was implemented a loop of successive capturing and extraction succeeded by matching along N-frames, during which the best matches found are stored. This loop is proceeded by a decision function which uses the stored matches to compute the weighted mean; it considers the identified ranks and the number of times each one occurred. The card rank which match is the closest to the weighted mean result, is the best match found for the correspondent card. This improves the algorithm’s reliability since the identification is not dependent on the identification of only one frame, but a set of them, more precisely fifteen which was empirically defined. Also, with the logged data the decision function implemented makes the best decision possible by considering the history of the matches occurred. The revised overview of the algorithm is shown in Fig. 5.24.

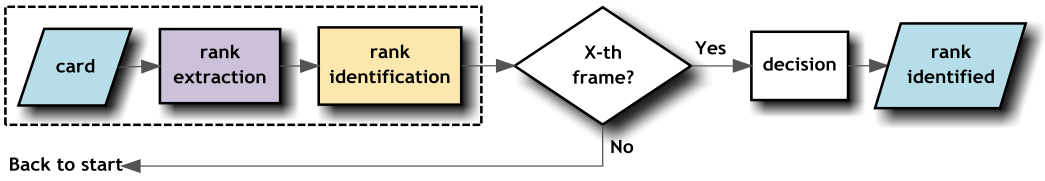


Fig. 5.24 – Revised algorithm block diagram after the improvements

5.2.2.2 Suit Counting Approach

The second approach to identify playing cards is based on counting the suits drawn on the card subject to identification. It is worth mentioning that this method is valid only for the ranks two through ten plus the Ace, since only these cards have the same number of suits depicted as its rank, e.g. the card which rank is 6 has six suits depicted.

Although the face cards are not covered by this method, these benefit from it indirectly. With this approach the templates correspondent to the ranks two through ten plus the ace

can be discarded, which prevents misidentifications when the template matching is performed for the face cards.

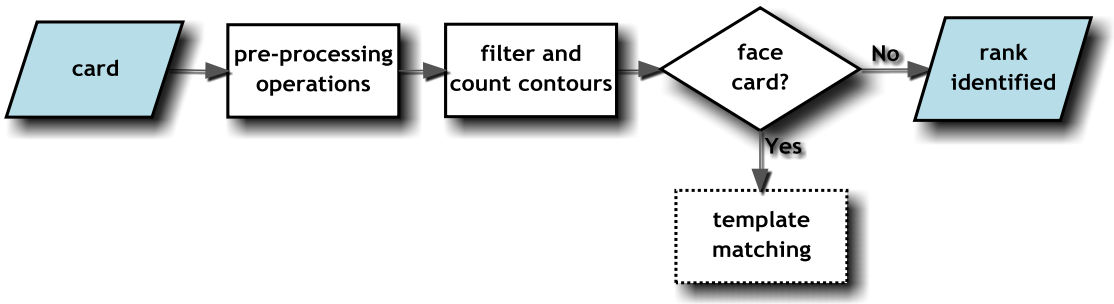


Fig. 5.25 – Suit counting - algorithm overview

The image undergoes low level image pre-processing in the pre-operations block, which is the same block detailed in *5.2.1 Card Finding and Extraction*.

Follows the "filter and count contours" block, where the algorithm finds the contours present on the binarized image. If the binarized image of the card submitted to this block is not a face card, each contour found represents a drawn suit present on the card, Fig. 5.26 a); in case it is a face card, contours are found as well but in this case corresponding to the holes and small elements of the picture drawn (Fig. 5.26 b)).

It is possible to distinguish both types of cards by analyzing the size of each element found on the card, since all the suits drawn on the cards have the same size. Therefore, each found contour is enclosed in a box and the algorithm checks if one of the enclosing boxes features either a width of height higher than half of the image's width or height, example Fig. 5.26 b-2) – biggest rectangle. If such happens, the card is classified as a face card and immediately redirected to the template matching approach. Nevertheless, because of light reflectance, it can happen that the figure drawn on the card is not correctly binarized, thus misleading the algorithm to accept it as a value card.

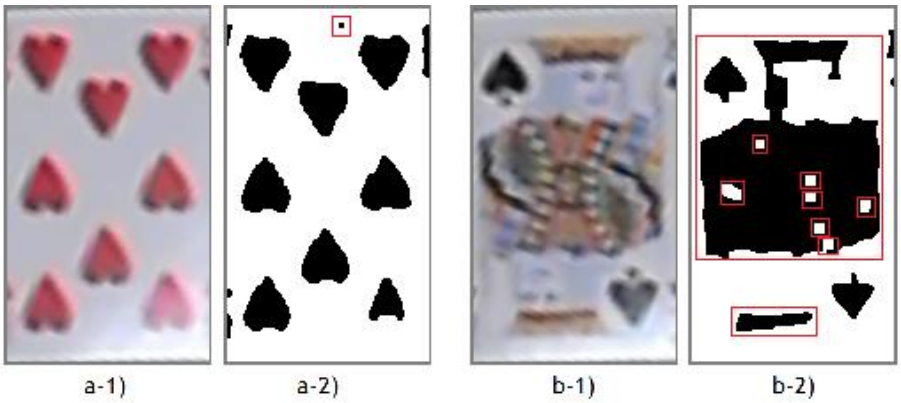


Fig. 5.26 – Two examples of the card categorization algorithm, where in a) only one contour was rejected, thus considered as a non face card; b) nine contours were rejected, thus considered a face card.

To double check the nature of the card, a second procedure was implemented. This procedure organizes the enclosing boxes in one of two categories, the rejected or the approved. This categorization is based on the dimensions of the enclosing box, which is compared with the acceptable size of a suit, empirically defined. If the size is within the acceptable dimensions it is considered a suit, thus categorized as approved; otherwise, it is not considered a suit, thus counting as rejected. If more than two rejected boxes are counted, the card is identified as a face card, redirecting the identification to the Template Matching method (Fig. 5.26 b-2)). Otherwise, it is finally considered a card from two through ten or the Ace and in this case the algorithm outputs the counting of the approved boxes as the card's rank (Fig. 5.26 b-1)).

Improvements to the algorithm

With the suit counting method an additional problem arises. In the case of illumination reflecting on the card and affecting the captured image, it can happen that some suits are not visible, thus not being counted. In order to overcome these situations, it was implemented an algorithm which advantages from the vertical symmetry of the suit's disposition in a card. This algorithm aims to partially reconstruct a card in case of light reflectance.

It is possible to categorize the cards by the number of columns (Fig. 5.27). First category consists of the cards which have only one column of suits in the middle of the card, more specifically the cards ranked two and three (Fig. 5.27 a)); second, the card ranked four, which has two (outer) columns of suits (Fig. 5.27 b)), being the only one in this category; finally, the ones which have three columns, one in the middle and two outer columns, which occurs in the remaining cards (Fig. 5.27 c)).

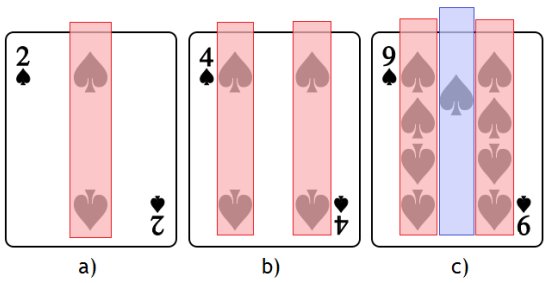


Fig. 5.27 – Representation of the imaginary columns of suits. a) represents the card 2 and 3, which have only one column in the middle; b) represents the card 4 itself; c) represents the cards 5 through 10

The first case, constituted by the card two and three, the symmetry of the suits is only horizontal and therefore these cards are not covered by this improvement. The latter two cases are vertically symmetric concerning the outer columns, since each card features the same number of suits in the outer columns – red coloured columns. Therefore, the algorithm is able to "reconstruct" these cards in case one of the outer columns is unreadable, while the other is fully readable. The algorithm counts the number of suits by column and makes the comparison of the counting of the outer columns, correcting it if needed. In case one of the

outer columns has fewer suits compared to the other (Fig. 5.28 b-2)) its value is discarded and assigned the counting of the other. The identified rank is the sum of the three columns.

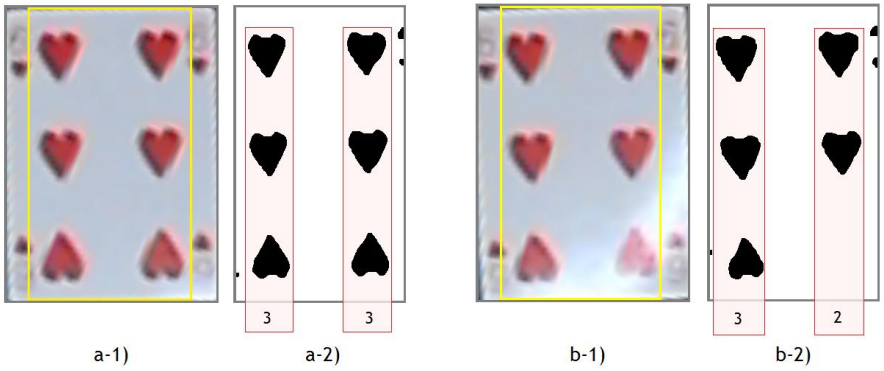


Fig. 5.28 – Examples on the suit counting algorithm, where in a) there is no light reflection and in b) there is light reflection

Having a look at the example in Fig. 5.28 a-1) and a-2) there are no reflection disturbances thus the counting does not have to be corrected; in b-1) and b-2) can be noticed a slight disturbance affecting the detection of the suit on the bottom right, however, with this improvement implemented the maximum value of counted suits prevails over the other, resulting in a correct identification as aforesaid.

This method of counting does not cover the situation where, at least, two suits of different columns are unreadable or if the inner column is completely or partially unreadable (Fig. 5.29).

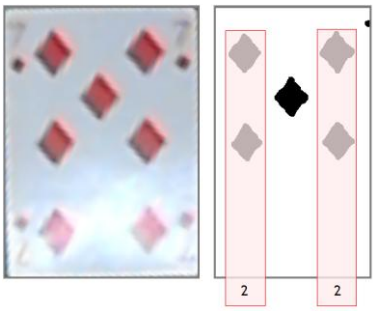


Fig. 5.29 – Example of the suit counting method failing

5.2.3 Suit Recognition

After a correct identification of the card's rank, comes the recognition of the suit. The approach used relies on Template Matching as well, for the same reasons stated on *5.2.2 Rank Recognition*. For this reason, the steps of this algorithm follow a similar path as the rank recognition using Template Matching detailed on *5.2.2.1 Template Matching approach*.

It is worth mentioning that part of the success of this algorithm depends on a successful rank identification.

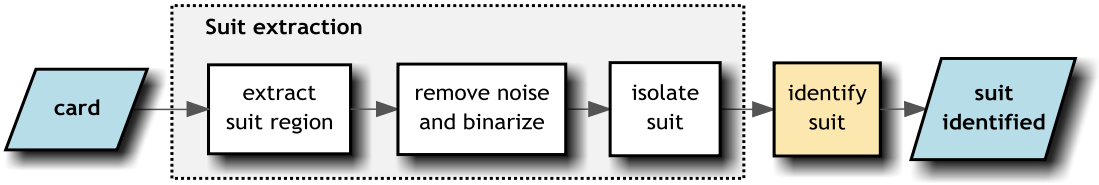


Fig. 5.30 – Suit Recognition overview - block diagram

Suit Extraction

Taking a look at a complete deck of playing cards, one can conclude that absolutely all of them have one small sized suit under its rank and, at least, one bigger suit elsewhere in the card (Fig. 5.31).

Fig. 5.31 – Example of the small suit and big suit on the Queen card

Since the position of the latter varies from card to card, extracting and analyzing the suit from the first situation would be the best solution since it is irrespective to the card on analysis. However, this option had to be discarded since the quality of the extracted region, which included the suit, did not produce satisfactory results.

Proceeding to the second option, it was clear that the results would be more satisfactory since the extracted suit is big enough to be identified clearly. As aforesaid, extracting the bigger drawn suit has some conditions which depend on the card's rank. Analyzing the cards, it was found three different locations for the suit, on different cards, which needed to be covered (Fig. 5.32).

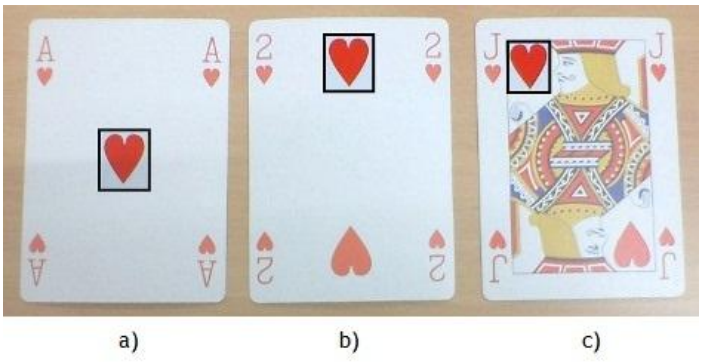


Fig. 5.32 – The 3 different positions of the suit

The first location of the suit is in the middle of the card, which is suitable only for the Ace (Fig. 5.32 a)); the second location covers the cards ranked two and three, which have a common place for the suit at its top centre (Fig. 5.32 b)); for all the remaining cards the common place for the suit is situated right next to its rank (Fig. 5.32 c)).

After extracting the suit the algorithm first converts to gray scale and then removes unwanted noise with a Gaussian smoothing filter. The contrast is enhanced by linearizing the histogram. Since there is unwanted noise remaining, the suit is isolated using the same algorithm presented in *5.2.2.1 Template Matching approach – Rank Extraction*.

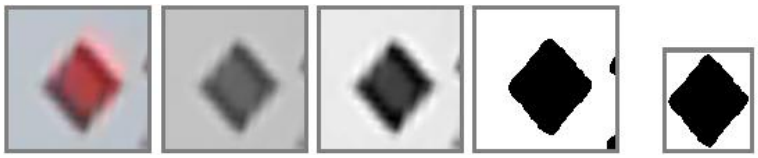


Fig. 5.33 – Suit extraction process; the image is first converted to gray, then the contrast is enhanced by linear stretching and followed by a binarization, finally the suit is isolated

Suit Identification

The procedures the suit undergoes are similar to the ones the rank undergoes (Fig. 5.34).

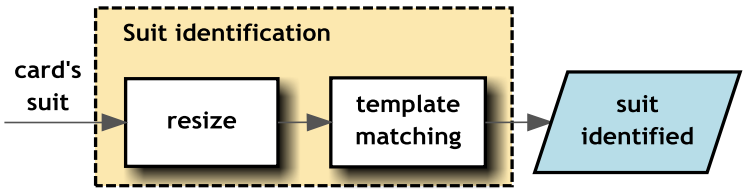


Fig. 5.34 – Suit Identification block diagram

The accuracy of Template Matching applied in this situation is expected to be higher than when used to identify the rank of the cards. The reason for this is that there are only 4 different suits which differ greatly in shape from each other, furthermore the size of the suits extracted have a bigger size than the ranks extracted, thus a better definition.

The template matching in this case follows exactly the same procedure as described in *5.2.2.1 Template Matching approach – Card Rank Identification*.

Fig. 5.35 – Templates used for the suit identification, *Diamonds, Hearts, Clubs and Spades* respectively

5.3 Chips Identification

In order to build a complete humanoid poker player, the information about the chips on the table is essential since it has a role on the decision during game play.

When a human player counts the value of the chips on the table, the counting is ordered by colour, i.e. what he is actually doing is isolating the chips by colour and performing the counting. In a similar fashion to the way humans make the counting of chips, the approach presented here relies on colour segmentation [34], to perceive the chips in the environment, followed by counting of circles, i.e. chips, using the Hough Circles Transform. In Fig. 5.36 can be seen the overview of the algorithm, which is represented by three main blocks.

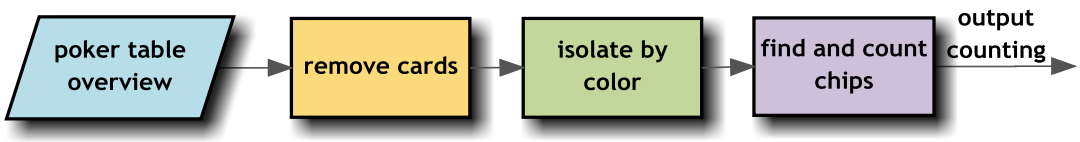


Fig. 5.36- Chips Identification block diagram

The isolation by colour is intended to separate chips of different values on the game, which are only distinguishable by the colour. Processing the counting before isolating the chips would output the total number of chips on the table, which is useless since only the total value on the table is interesting and not the total counting of chips.

The block “find chips” performs the counting of the chips by advantaging from the rounded shape characteristic of the chips. This method allows as well the counting of chips even if partially superimposed.

5.3.1 Removing Playing Cards from the Captured Image

During a poker game, the chips and the playing cards share the *community area* of the table, as there are no pre-defined spots for placing both. The vision system has to be capable of handling the random placement of both and prevent misidentifications. In this specific case, the algorithm segments the captured images by the colour of the chips (Fig. 5.36 – second block) which helps to focus the finding of chips in the interesting regions. However, parts of the playing cards remain after segmentation, in case the algorithm is performing the colour segmentation of the red or the black colour. This is caused by the black and/or red elements present on each playing card. To prevent misidentifications of the elements of the cards with chips, the first block was implemented aiming the removal of the *community* cards. The removal relies on a mask containing the cards’ positions, which filters the image containing the overview of the table (Fig. 5.37). The information with the position of the cards is fed by the algorithm which finds the cards on the table, detailed *5.2.1 Card Finding and Extraction*.

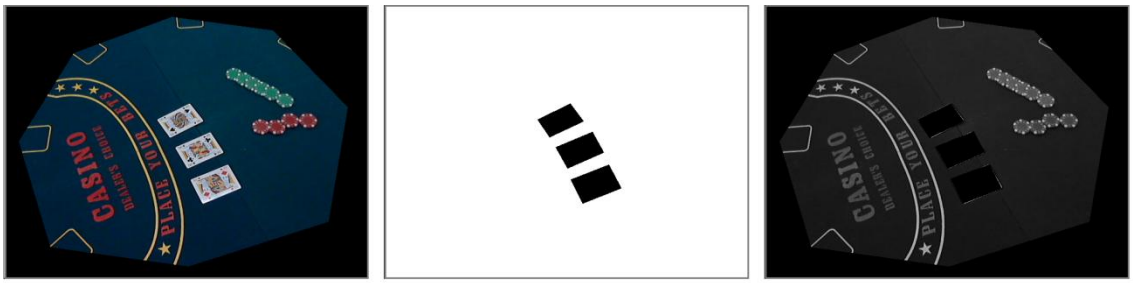


Fig. 5.37 – a) overview of the table with cards and chips in the *community* region; b) mask created based on a); c) resulting image after applying the mask – playing cards removed

5.3.2 Isolate Chips by Colour – Colour Segmentation

Image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics. In this case, it was adopted the colour segmentation, since the chips differ in colour from each other.

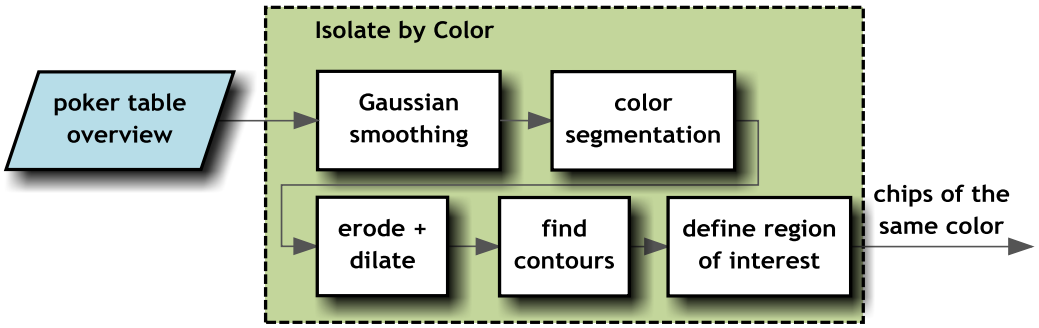


Fig. 5.38 – Isolation of chips block diagram

To work with images and colour representations, a colour model is always used. A colour model describes colours as tuples of numbers, typically as three or four values or colour components, e.g. RGB (Red Green Blue), HSV/HSL (Hue Saturation Value/Lightness), CMYK (Cyan Magenta Yellow Key), YUV (Y - luma component (the brightness), U and V - chrominance (colour) components). To achieve the goal of colour segmentation is preferable to use the HSV or HSL colour model over alternative models such as RGB or CMYK. This model is the best colour space concerning the emulation of the way humans perceive colour. Thinking of how someone describes a colour, normally the words used are the name of the colour itself and an adjective such as “light” or “dark”, e.g. light blue or dark red. The HSV colour space is modelled this way, where the “Hue” represents the actual colour, e.g. blue, yellow, purple, etc, and the “Value” or “Lightness” represents how light or dark the colour. In terms of a spectral definition of colour, value describes the overall intensity or strength of the light. The last dimension of colour that describes our response to colour is “Saturation”, which refers to the dominance of “Hue” in the colour [35].

The RGB colour space, initials for Red, Green and Blue, is an additive colour model, which models the way primary colour lights or pigments combine to form new colours when mixed. It is worth mentioning that the raw image format supplied by the camera is RGB itself. This is an advantage of working with the RGB colour space since the computational cost of converting from the raw format colour space, RGB, to another colour space is rather high. The RGB colour space can be represented as a cube (Fig. 5.39).

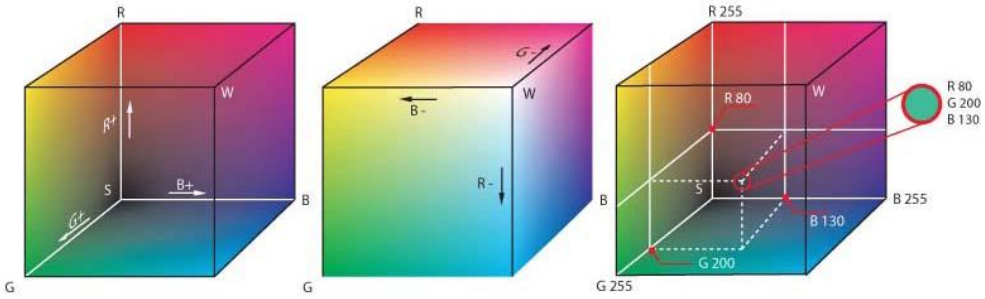
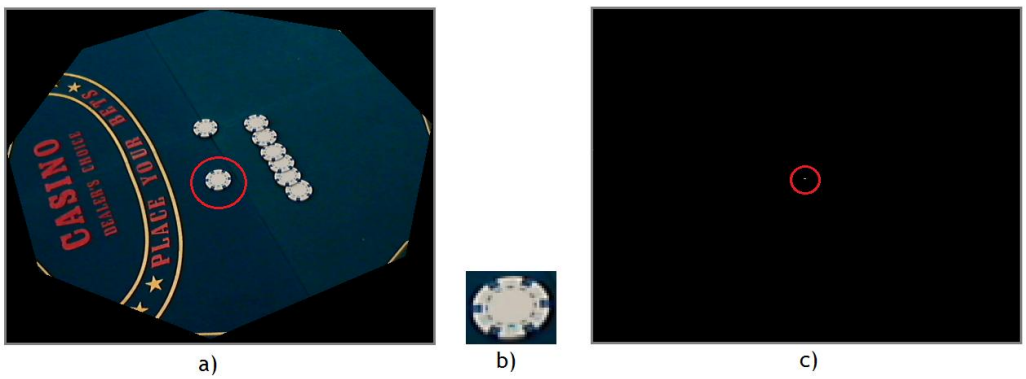


Fig. 5.39 – The RGB colour space representation [36]

Although the HSV colour space would be the obvious choice concerning the easiness on the manipulation of colours, the choice fell in the RGB colour space. This is reasoned by two things; firstly it is worth mentioning that the purpose of the segmentation in this case is merely to isolate the chips and not the counting itself. Secondly, if using the RGB colour space there is no need for converting between colour spaces, saving computational time.

The colour segmentation process relies on making use of the desired object physical properties such as light absorption, reflection, or emission spectra to identify the object on an image [35]. However, this type of segmentation is not as straight forward as picking one vector of RGB components for each colour, since it would not produce interesting results. As an example, in Fig. 5.40 the image in a) was input for colour segmentation. The RGB values chosen were R=196, G=200, B=184, which correspond to the colour components of the centre of the chip in b). The resultant image after colour segmentation using the specified RGB components is Fig. 5.40 c), meaning that only one pixel was found, in the entire image, with that combination of RGB components. This pixel is the one from which the RGB components were extracted.



a) b) c)

Fig. 5.40 – Example of colour segmentation using only one vector of RGB values

In this example it is observed that although the chip is, apparently, uniform with respect to the colour, minor differences in the RGB components will prevent the segmentation from returning interesting results. These minor differences on the colour of the object are related on both the physics of the object in its environment, intensity and angles of illumination, and the characteristics of the capturing device which adds some colour interference [35].

In order to overcome this situation, first a smoothing filter is applied with the purpose of neutralizing high variations on the pixel values. However, this procedure is not enough since the RGB components will still vary even if the pixels correspond to the same chip. In case the pixels are from chips of the same colour which are in two opposite regions of the table, this variation will be even higher. So, it is needed to create a window of possible values centred on each RGB component, which will be used for the segmentation process. In Fig. 5.41 can be seen a similar example to Fig. 5.40, with the difference that now the "window" of values was used for the segmentation. The RGB vector selected as a reference corresponds to a random pixel from the middle of a white chip (Fig. 5.41 a) – red circle). The results are noticeably better, since all the white chips were found and displayed in Fig. 5.41 b). Should be noticed that in the same image, marked in blue, can be seen a playing card that was partially visible on the original captured image. This playing card was filtered since it has white spots on it, therefore confused with the white coloured chips.

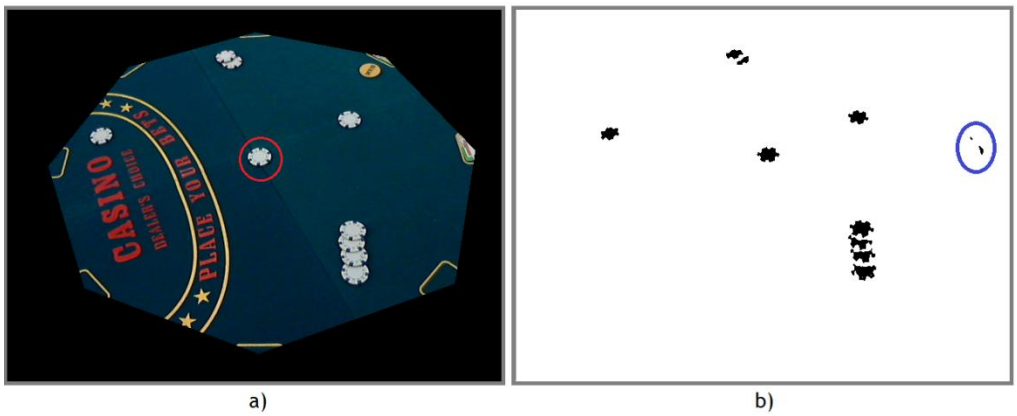


Fig. 5.41 – Colour segmentation after defining a range of RGB values

In case the capturing device is exchanged or the devices' configuration is modified, the same problem arises since the acquired RGB values will be different. To overcome this situation a calibration block was created and should be executed whenever there are great changes in the illumination, different chips, camera settings and/or camera. The calibration consists in placing the chips of each colour on the poker table and, using the interface, the user should click on each chip and define how much it will be worth during the game play. The information about the RGB components of the clicked pixels, one for each colour, is stored in an external file along with the respective value. Whenever the colour segmentation is executed this information is used as reference for the "window" of RGB values.

The resultant binarized image, example in Fig. 5.41 b), undergoes morphological operations, i.e. series of eroding and dilating operations, in order to remove noise and strengthen the contours of the chips found (Fig. 5.42 a)).

In order to define the region of interest as tight as possible, aiming to isolate only the interesting chips, the found contours are enclosed in rectangles (Fig. 5.42 b)). Afterwards, a "mask" is created with the boundaries of the rectangles, which is then used to filter the interesting regions in the original gray scale image. Resulting in an image where each not null region contains the chips of the isolated colour to be counted (Fig. 5.42 b)).

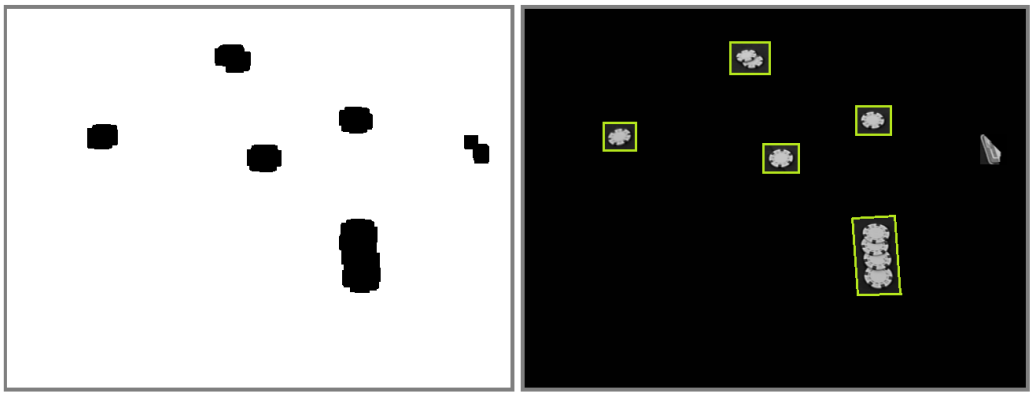


Fig. 5.42 – Result of the colour segmentation a) and resulting areas of interest b). In b) the boundaries used to delimit the chips are depicted in green

5.3.3 Count Chips – The Hough Circle Transform

The previous block spots the chips as well as other kinds of objects of the same colour. In case the latter are present on the captured image, these must be discarded. Moreover, it is intended to design the algorithm as robust as possible by making the detection of partially superimposed chips, Fig. 5.44, since it is common to happen during the game play. All these requirements are achieved using the Hough Circles Transform.

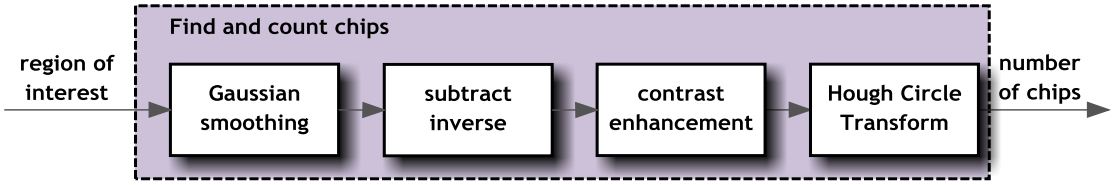


Fig. 5.43 – Counting of chips block diagram

Fig. 5.44 – Example of superimposed chips

Before finding the chips with the Hough Circle Transform, the algorithm performs some operations in order to emphasize the edges of the chips. The first of these operations is Gaussian smoothing, detailed in *5.1.1 Smoothing*. The next step consists of computing the absolute difference between the gray image of the chips with itself. Only difference is that the first has the brightness inverted, while the second does not (Fig. 5.45 a) and b)). This procedure results in a great contrast between the chips and the table (Fig. 5.45 c)).

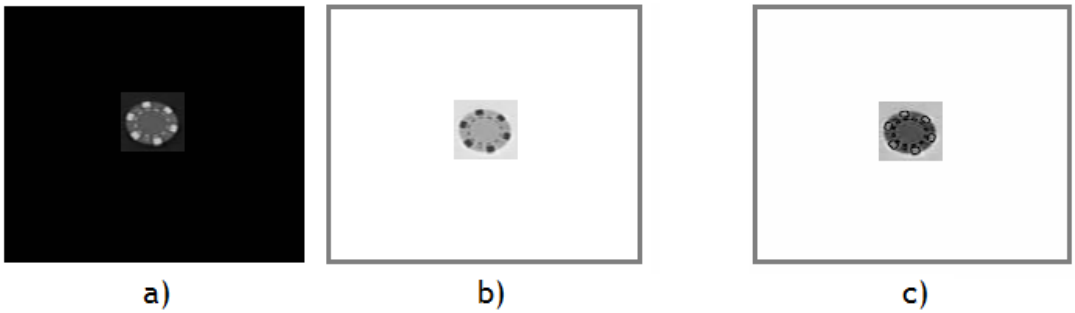


Fig. 5.45 – a) image with the brightness inverted – negative image; b) positive gray image; c) result after computing the absolute difference between a) and b)

In order to enhance the contrast even more, the resultant image undergoes a linear stretch of the histogram. With the contrast between the chips and the poker table enhanced, it is now possible to count the chips accurately.

Since the chips have a rounded shape, the Hough Circle Transform is used to detect them accurately. The detection is also successful even if the chips are partially occluded, but with their boundaries visible enough to be detected.

The Hough Circle transform relies on determining the parameters of a circle when a number of points that fall on the perimeter are known. A circle with radius R and center (a,b) can be described with the parametric equations:

$$x = a + Rcos(\theta) \tag{5.8}$$

$$y = b + Rsin(\theta) \tag{5.9}$$

When the angle θ sweeps through the full 360 degree range, the points (x, y) trace the perimeter of a circle. If an image contains many points, some of which fall on perimeters of circles, then the task of the search algorithm is to find the triplets (a, b, R) which describe each circle. However, if the radius R of the circles are known, then the search is reduced to find the (a, b) coordinates of the centres of each circle [37].

The position of (a, b) points in the parameter space is reduced to a circle of radius R centred at (x, y). The true centre point will be common to all parameter circles, and can be found with a Hough accumulation array [37].

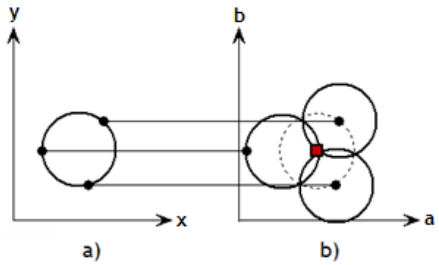


Fig. 5.46 – Each point in geometric space a) generates a circle in parameter space b). The circles in parameter space intersect at the (a, b) that is the centre in geometric space.

Multiple circles with the same radius can be found with the same technique. The centre points are represented as red cells in the parameter space drawing. Overlap of circles can cause false centres to be found as well, such as at the blue cell. However, false circles can be removed by matching to circles in the original image [37].

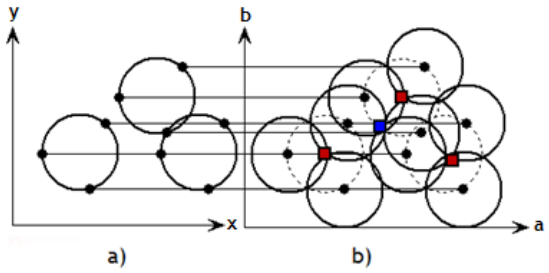


Fig. 5.47 – Each point in geometric space a) generates a circle in parameter space b). The circles in parameter space intersect at the (a, b) that is the centre in geometric space.

As demonstrated, the Hough circles transform itself was conceived to detect circles. Therefore, in order to produce interesting results, the image submitted must be an edge image. However, the edge detection function, in this case Canny edge detector, is embedded in the Hough Circles function, so it is not needed to be implemented in the present algorithm. An example of how the Hough circles transform works can be seen in Fig. 5.48.

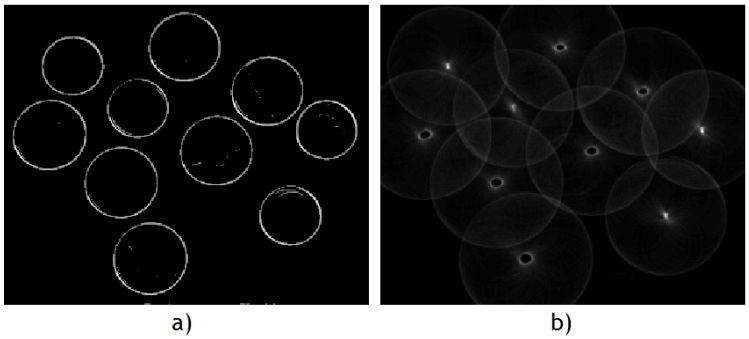


Fig. 5.48 – a) Random image with circles submitted to the Hough Circles Transform – after the canny edge detector. b) Resultant image where the centre of each circle is marked.

In Fig. 5.48 b) the brightness of each centre of circle represents the accumulator. The brighter it is the more votes it received during the computation of the Transform and more confidence there is about being a circle.

The number of circles, i.e. chips, found of each colour is stored along N frames. In the end the average of the counted chips is calculated and output. This method of count averaging was implemented since the Hough Circles Transform is slightly inconsistent along consecutive frames, due to the parameters of the function specified. These parameters were chosen in order to cover several setups of the *webcam* position and positions of the chips, therefore the trade-off is a slight inaccuracy in some situations.

5.4 Conclusions

In this chapter was presented the major part of the work developed, corresponding to the algorithms designed with the goal of identifying playing cards and chips.

During the implementation, low level image processing methods were used in order to pre-process the captures images before any identification is performed. Within the method used, one can find smoothing, image binarization, contrast stretching and mathematical morphology (including dilation and erosion).

After the pre-processing, the captured image is submitted to an algorithm which finds cards present on the table and isolates them. Thereafter, each card is categorized either as a *face card* (i.e. King, Queen or Jack), or as a *value card*. In case of a *face card*, its rank is extracted and *template matching* is performed; in case of a *value card,* the suits depicted in the card are counted. In order to identify the suits, the methodology used was *template matching*.

As for the hole cards, the procedure is not mentioned specifically since it does not differ from the identification of the community cards in any aspect, only the capturing device used is different.

Concerning the chips identification, the captured images are first segmented by the colour of the chips to be used during the game play, one colour at a time. This procedure aims to isolate the zone of the image which comprises the chips of a determined colour. Thereafter, the algorithm performs the Hough Circles Transform in order to detect and count the circles present in the table.

Chapter 6

Additional features

During the game play of a poker game, it is possible to define two kinds of considerations when a player has to make a move. The first kind, and the most obvious, consists in the cards dealt, both to table and to the player, along with the total pot, i.e. amount in chips on the table. The second kind, not so obvious yet important, consists in such things as the position of the player relative to the dealer and the number of players still in the game.

The first kind of considerations is already covered by the algorithm detailed so far, which provides to the higher level, i.e. the brain which makes the decisions on the poker moves, the major information about the game. However, in order to successfully implement this system in a real environment, the implementation has to focus not only on the card and chips recognition, but also on the perception during the whole poker game play. It is of interest to have the system following the game play during the betting round, i.e. know whose turn is it to play. Otherwise the system would be completely dependent of external action by humans.

The system's self-reliance is based on determining the number of players in the game and respective positions, *dealer* position and *Speech Recognition* as well. These features enable the perception layer to follow the game in real time, in a similar fashion to how humans do.

6.1 Configuration of the Table

6.1.1 Calibration of the table

It would be nearly impossible to make this whole system table independent if no calibration was performed. The calibration implemented relies on determining the seats' coordinates available on the table. These coordinates are valuable information for the features implemented which percept a great part of the game play during the betting round.

The calibration implemented is not fully automatic, instead it is required the user to place one white coloured chip in each spot reserved for the cards – which will be called the seat of the player henceforward - Fig. 6.1 a); no other white objects should be placed on top of the table when performing it, in order to avoid an incorrect calibration. The algorithm binarizes the image by colour segmentation, resulting in a binary image with only the spots of the white chips remaining (Fig. 6.1 b)). The chips used feature a white colour, which should

correspond to the RGB vector (255,255,255), however it is considered as white all the values above (160,160,160), in order to overcome different illumination gradients present in different ambient conditions.

Fig. 6.1 – a) overview of the table with the white chips placed on the pre-defined squares for the cards; b) and the resulting image after colour segmentation and eroding+dilation

From the resulting binarized image, the algorithm searches for contours (Fig. 6.1 b) – white elements) where each one represents the seat of each player. These spots are now ordered according to the y-coordinate, starting on the PokerVision's seat, (Fig. 6.1 b) - in red) which has always the highest y-coordinate, i.e. the bottom of the image. Then, the remaining positions are ordered clockwise, resulting in the list of the 8 seats ordered along with its coordinates.

The calibration should be run every time the system is setup in order to overcome any changes in the position since the last calibration made.

6.1.2 Game Area filter

When the *webcam* is capturing the overview of the table, including the seats of the players, part of the capture is of no interest. This part consists, roughly, of the image's corners, while the centre region of the table is where the action is, i.e. community region where the cards are dealt and the chips are thrown (Fig. 6.2 a)).

During the recognition of the *community cards* or the counting of chips there is no need to process the corners of the captured image, therefore these parts are filtered using a *mask*. This process is similar to the one used when removing the cards during the counting of chips, *5.3.1 Removing Playing Cards from the Captured Image*. In this case the *mask* is parameterized with the players' seats coordinates, determined during the calibration of the table, corresponding to its corners. Therefore, instead of a circle the resulting mask will be octagonal due to the number of seats being eight (Fig. 6.2 b)).

This procedure was implemented mainly to improve the reliability of the counting of chips which was described in *5.3 Chips Identification*. The players' chips are normally placed around the community area (Fig. 6.2 a) – zone in red) therefore its capture is inevitable. It is interesting to discard the part where these chips are placed since it prevents the players' stacks to be misidentified as bets.

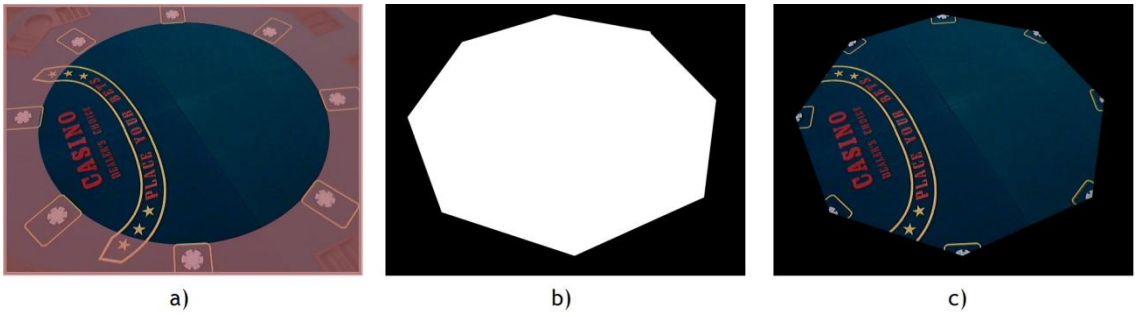


Fig. 6.2 – a) community area of the game play; b) *mask* created based on the coordinates of the chips/seats in a); c) resulting image after applying the *mask* in b) to the original image in a)

6.2 Status of the Players

This feature determines the number of players in the game and their respective positions around the table. As referred in *4.1 Hardware*, the poker table used has pre-defined spots for the cards. Moreover, during game play, when a player "folds" he must give the hand cards back to the dealer. Advantaging from both characteristics, the algorithm relies the determination of the number of player on detecting if their hand cards are on the pre-defined spot on the table.

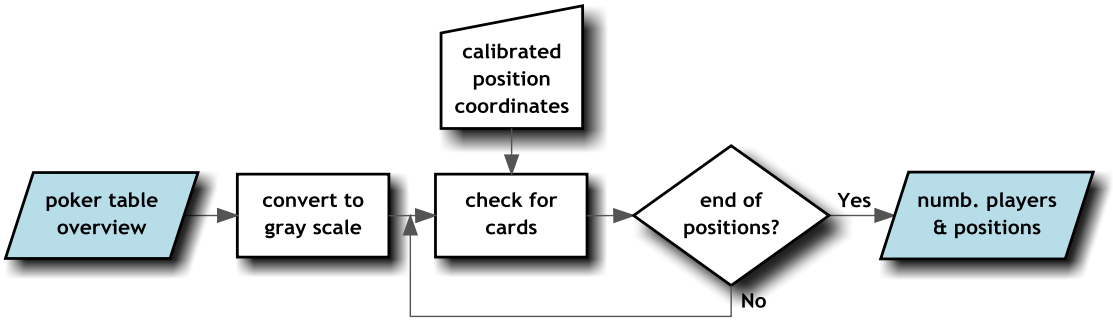


Fig. 6.3 - Fluxogram representing the algorithm which checks for players in a table

Checking the status of a player can rely on the colour of the pixel defined in the calibration as its position. In the specific case of this work, the colour of the poker table is significantly different from the colour of the card, where the first one features a dark green colour and the latter a light green colour (Fig. 6.4). When converted to gray scale, the first features a low brightness level, while the latter features a high brightness level. Therefore, since analyzing gray scale images is computationally easier and, in this case, effective as well, the algorithm relies on the brightness level of each pixel.

Fig. 6.4 – Picture of a card laying on the poker table, from the same deck used in this work. Although not perceptible, dark green is the actual colour of the poker table

The first step of the algorithm is to convert the image to gray scale. Afterwards, and along with the information previously calibrated, the algorithm checks each position existent, which in the case of the octagonal table corresponds to a total of eight positions. Each position has its own correspondent pixel, which is checked for its value of brightness.

When the algorithm finds the brightness level of a certain pixel to be low, means there are no cards in that spot and the player is out of the game. If the value of brightness happens to be high, then there is a card laying on the pre-defined spot, meaning the player is still on the game.

The algorithm presents some drawbacks which were known right from the beginning of the implementation, however these are not easily solvable since some of them rely on how the free will of the players during the game, e.g. the way they sit, the place where they put their hands, the way they place their cards.

The first two robustness problems rely on the different types of poker tables existent. Although this algorithm is table independent, it happens that some tables do not have a pre-defined spot for the cards (Fig. 6.5) becoming unpredictable the exact place where the player will put the cards. Moreover, as aforesaid, the algorithm will take advantage from the difference brightness levels between the back of the card and the poker table. In case the back of the card and the table have similar brightness values (Fig. 6.5) its reliability is not assured.

Fig. 6.5 – Example of situation with two issues. First there are no predefined spots for the cards. Second, the dark blue colour of the poker table along with cards featuring dark blue stripes can affect the reliability

Another weakness of this algorithm is its dependency on the free will of the players. In other words, while a player is in the game he can check the cards dealt to him whenever he intends to, which can interfere with the algorithm checking the players status. If, for

example, one player is looking at his cards, having the cards in his hands, the algorithm determines that he is not in the game anymore since it does not find the cards where they are supposed to be, i.e. the pre-defined square for the cards.

6.3 Dealer Position

The *Dealer* position is determined by finding the *Dealer button,* also known as *Button,* on the table and comparing its coordinates with the positions calibrated for the cards of the players. Moreover, the *Button* only changes position every time a new round starts, which is when the cards are collected to be shuffled and the chips are collected as well by the winning player, remaining only the b*utton* on the table. Since the poker table features a dark colour and the *Button* is white coloured, the identification of its coordinates could rely only on segmentation by white colour. However, during the game play one issue would arise when determining the total amount of chips in the table. Since there are white coloured chips to be identified, the *Button* would be identified as one white chip, which is not interesting.

One way to overcome this issue could rely on spotting the *Button* and create a mask based on its position and shape. Afterwards this mask would be used to discard the *button* every time the counting of white chips was executed. This would be similar to the process of removing the cards when the counting of chips is performed, *5.3.1 Removing Playing Cards from the Captured Image*. Unfortunately, the mask had to be generated in the beginning of a round and would be applied until the end of that round. If, for any reason, the *Button* was displaced, the mask would be useless and the problem would persist.

In order to overcome this issue the "Big Blind button" was adopted instead, which is similar to the *Dealer button* except for the colour, which is yellow. The yellow colour is suitable for this implementation since there are no yellow coloured chips, thus preventing it to be identified as a chip.

Fig. 6.6 – The "Big Blind button"

Finding the "Big Blind button" is approached in a similar fashion as how the calibration is done, i.e. colour segmentation.

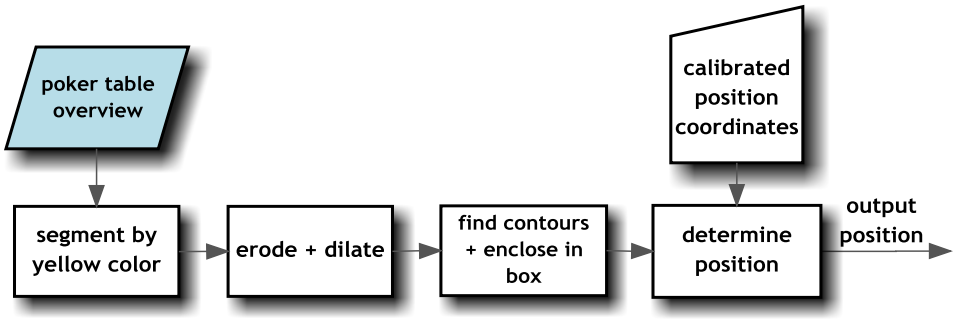


Fig. 6.7 – "Big Blind" position finding algorithm

Therefore the captured image, which contains the overview of the table, is first segmented by the yellow colour. The colour space used was, again, RGB, and the yellow colour was defined as being in a range of RGB values centred on the vector correspondent to pixels of the "Big Blind button", percept by the camera and found after some tests.

The poker table used along this work has a drawing stamped on it, which included yellow parts. These elements persisted after applying colour segmentation of the yellow colour, therefore it is interesting to remove these parts of the drawing. It was possible to do it by applying the morphological operation of eroding. After this operation, dilating was also applied in order to make the "Big Blind button" region stronger.

After these operations only a strong region will remain, which corresponds to the "Big Blind Button", along with some noise. In fact, because of the words "BIG BLIND" written in black colour, occupying great part of the "Big Blind button", the region in the binarized image of the "Big Blind button" will be split in two (Fig. 6.8 c) – in red). Therefore it is needed to find the "Big Blind button" by associating both parts of it, while at the same time discarding any remaining noise. In order to achieve that, the algorithm makes a search for the contours present on the binarized image and encloses them in a box. Depending on the dimensions of the box, it is discarded or not. It is worth mentioning that these dimensions are concordant with the expected size of each one of the parts which constitute the "Big Blind button", determined empirically.

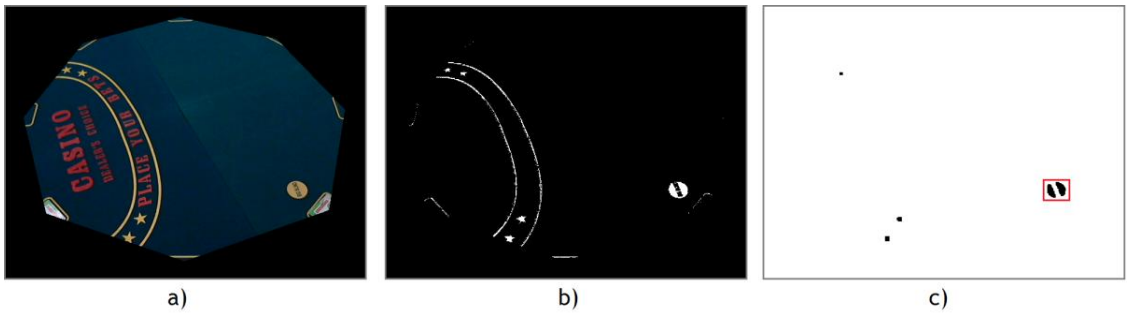


Fig. 6.8 - Low level operations to find the "Big Blind button". a) captured overview of the table; b) segmentation by colour – yellow colour; c) binarized image after eroding, dilating and colour inversion

The remaining contours are then compared with each other concerning the coordinates, in order to find the two parts of the “Big Blind button”. The closest pair, i.e. the two contours with the shortest distance between them, is considered the “Big Blind button”.

Finally, with the coordinates of the “Big Blind button” determined, it is now possible to determine to which player it has been assigned. The previously calibrated positions of the players are used to calculate the distance between each one and the position of the “Big Blind button”. The position which has the shortest distance to the “Big Blind button” is considered to be in the “Big blind” position.

Although the “Big Blind” player was determined instead of the “Dealer”, it is possible to obtain the latter indirectly. The rules of poker state that the two players to the left of the dealer are called the small blind and the big blind, respectively. Therefore, determining the “Dealer” relies on finding the player which is two positions to the right of the “Big Blind”. This can be done by combining the information about the “Big Blind” position and the information concerning the players in the game, in the starting of each the round.

6.4 Perception of Players taking Action - Speech Recognition

The game play will continue very restrictive and unattractive for human players, while the system is not able to percept when a player acts in the betting round. Therefore, it was decided to implement a module capable of understanding the players’ actions, but at the same time transparent for the players.

It is worth mentioning that the module to be implemented is of interest to provide the other players the most transparent and natural, i.e. non artificial, experience as possible. It is also of interest to keep the game play in its original form, avoiding the imposition of restrictions to it, as well as maintaining the original form of the game’s equipment, i.e. poker table, chips, cards.

In order to implement a perception system of this kind, a speech recognition module was implemented. This solution upgrades the perception module while at the same time preserves the game play, i.e. overcomes any physical implementation of non standard elements, such as a set of buttons to each player, in the poker table or game area. The only restriction imposed to the players is to declare the action by words, i.e. avoid declaring the action by gestures.

In this context the Microsoft SAPI arises as a valid option, since it allows the use of speech recognition within Windows applications. SAPI can be viewed as an interface which sits between the application and the speech engine.

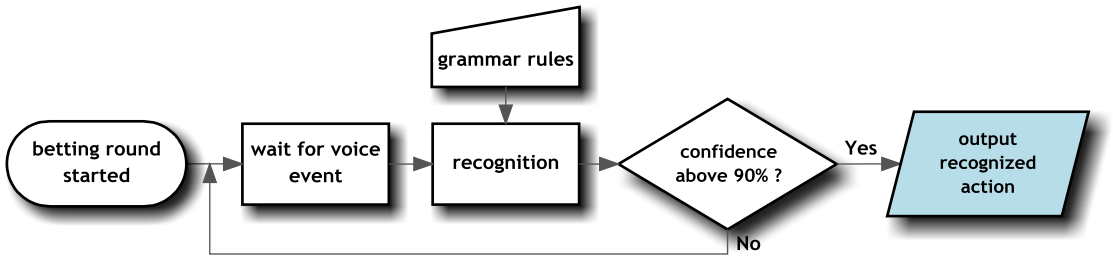


Fig. 6.9 – Flow chart of the Speech Recognition module

During the dealing of the cards there is no need for speech recognition since the players are not supposed to act/talk. During the betting round they are supposed to act or “talk”, therefore the speech recognition is activated by that time, waiting for a recognition event. Every time a spoken sentence is captured an event is raised, informing the recognition block to process the captured sound, i.e. sentence. The recognition block splits the phrase in single elements and tries to match the elements with predefined grammar rules.

Grammar rules help the SAPI engine recognize what is being said, since it provides the engine with what to expect. It limits the set of recognition hypothesis to a custom set of voice commands, substantially increasing the recognition match accuracy. A specific grammar was created with the purpose of identifying the actions during the betting round, which covers the actions a player can take in his turn during the betting round. Therefore, there would be a total of 4 commands to be implemented consisting of “call”, “check”, “fold” and “raise”, however since these commands are composed each one by only one word, the result during an ordinary poker game can be unreliable. The unreliability would consist of misidentifications where a random word could be interpreted by the recognition engine as one of the words in the grammar, in case both were phonetically similar. This could happen, for example, during an ordinary conversation between players during the betting round. In this same situation could happen, as well, in the middle of the conversation a player pronouncing one of the words present in the grammar, i.e. a valid poker action, but with no intention of taking that action, e.g. “If I was you, I would **call** this”. In order to overcome both situations, each of the grammar rules, i.e. actions, is composed by a sentence and not only a single word (Fig. 6.10):

Call: “I call now”

Raise: “I raise now”

Check: “I check now”

Fold: “I fold now”

This makes the speech recognition more reliable since a sentence has a more specific phonetic form when compared to an isolated word. Moreover, the content of the sentences selected act as a confirmation of the player, since they affirm specifically which action the player is taking.

```
<GRAMMAR LANGID="409">
  <RULE ID="1" TOPLEVEL="ACTIVE">
    <P>I Fold npw</P>
  </RULE>
  <RULE ID="2" TOPLEVEL="ACTIVE">
    <P>I Call now</P>
  </RULE>
  <RULE ID="3" TOPLEVEL="ACTIVE">
    <P>I Check now</P>
  </RULE>
  <RULE ID="4" TOPLEVEL="ACTIVE">
    <P>I raise now</P>
  </RULE>

</GRAMMAR>
```

Fig. 6.10 – Grammar rules used for the Speech Recognition used in this work

In case one of the actions, present on the grammar, has been identified, the recognition engine returns the identified word along with the confidence rating calculated. This confidence is used by the algorithm implemented to discard the results which score less than 90% of confidence, while the remaining are accepted as the recognized action. This percentage was found empirically and aims discarding results which present a low confidence rating.

It is worth mentioning that the implementation of the sentences aforesaid, will add another restriction to the declaration of the actions of the players, making the game somehow more artificial from the point of view of the human players.

6.5 Game Status

This section focuses on merging the features described so far, in order to provide the ability of following in real time the game play.

In the beginning of each round and after the hole cards are dealt, the algorithm determines how many players are in the table and in which positions, as well as the position of the big blind.

Crossing the information relative to the players in the game, with the information concerning the position of the big blind, it is possible to determine who the first player is to act in each betting round. It is worth reminding that the first player to act in the betting round is ruled as referred in the game play rules.

With the speech recognition module active, as soon as a player declares his action, the system detects it. Depending on the action of the player, it can be determined if he continues on the game, in case he “raises” “calls” or “checks”, or if he gives up of this round by “folding”. Therefore, it is possible to update the number of players in the table in real time and accurately, by making use of a finite state machine.

The state machine implemented makes use of all the features implemented along the work, resulting in a system almost independent from user interaction and capable of following the complete game play. The flow chart along with some pseudo-code of the state machine can be seen in Fig. 6.11.

It is worth mentioning that the flow chart in Fig. 6.11 has two blocks which have not been referred so far, “Upload information” and “high level decision”. Both were implemented in the flow chart in order to make the complete state machine logical.

The first block has the task of sending the complete set of information concerning the game to the upper level. The upper level is represented by the “high level decision” which has the task of making the decision on which action is the best.

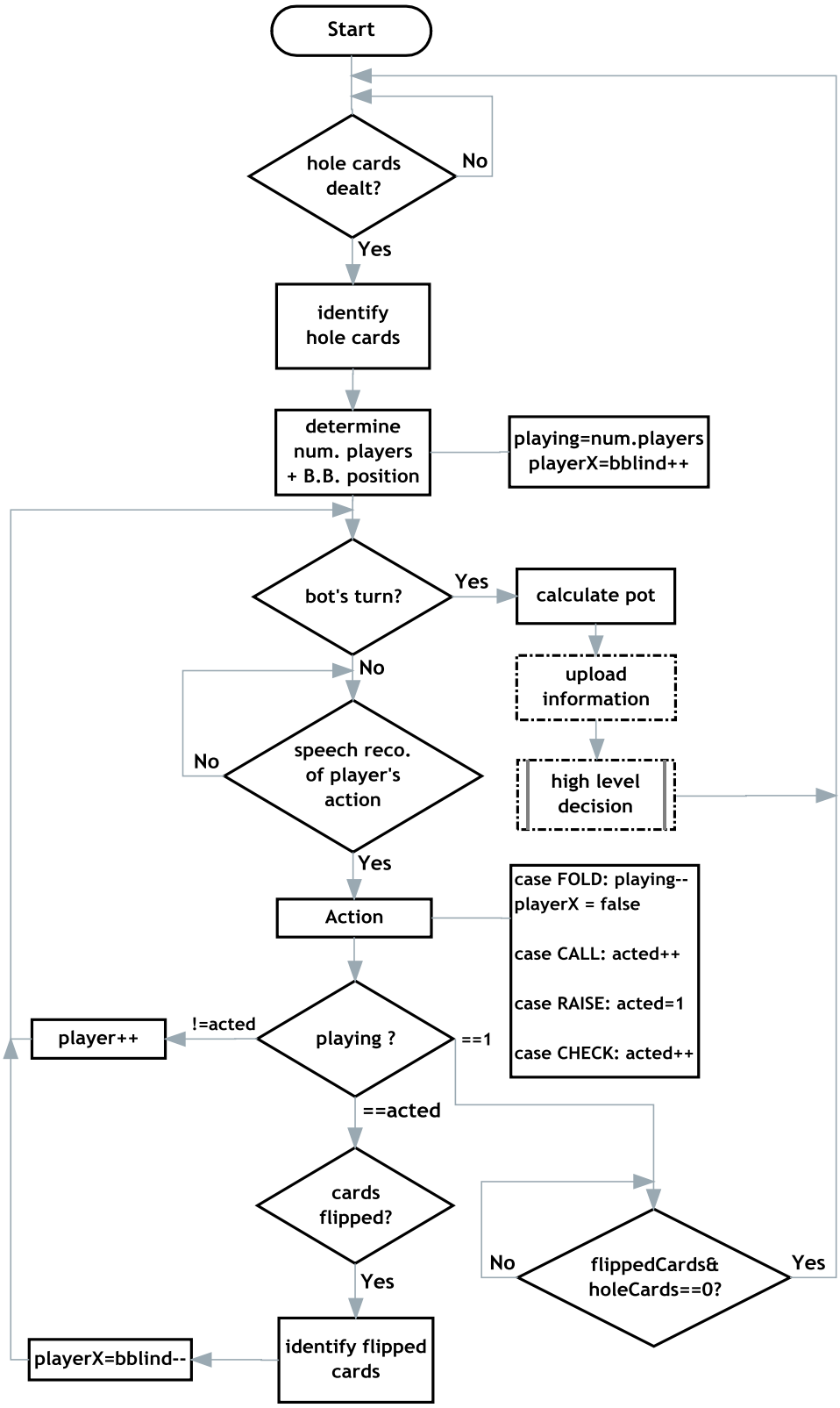


Fig. 6.11 – Finite state machine which follows the game play

6.6 Conlusion

In this chapter were presented the algorithms which make possible the following of the complete game play. The features implemented consist of the counting of players in a round of the game, determining the position of the dealer/big blind and following the player's actions by speech recognition. Additionally, it was implemented a game area filter which restricts the analysis of the playing cards and chips to the *community* region of the table, since the rest of the table is of no interest. All these features implemented are combined in a state machine which represents a simple, but efficient, method of keeping track of the game.

Chapter 7

Experimental Results and Discussion

This chapter presents the experiments performed using the algorithm developed and the results achieved, as well as their discussion. The tests performed and respective conclusions cover the recognition's accuracy of the playing cards under different conditions, such as the placement of the webcam in different positions around the table, different resolutions and different cards. As for the chips, the tests relied on checking the accuracy of the counting relative to the distance from the webcam, both when placed separated from each other and when partially occluded.

7.1 Card Recognition

In order to determine the card's recognition accuracy, the algorithm was tested with the camera in three different positions around the table (Fig. 7.1). The main goal was to test how the algorithm performed relative to the distance, angle and perspective from the webcam to the cards being identified.

Procedure during the tests made:

- Concerning the disposition of the cards during the tests, it was delimited a region in the table within which the *community cards* were disposed. This region was defined empirically (Fig. 7.1 – green square). In professional poker the cards are dealt the same way, i.e. always disposed in the same region of the table, Fig. 7.2;
- The recognition was performed three cards at a time, which were placed along the *community cards'* region (Fig. 7.1 – green square)
- The recognition of a suit was only considered, and its result registered, after a correct identification of the card's rank, since the suit's extraction is dependent on the success of the card's recognition. If, for example, a playing card ranked "four" is submitted to recognition, but the algorithm identifies it as a "two", then the algorithm will define the region of interest erroneously, thus it will not even extract the rank successfuly.
- All the tests were performed under artificial lightning in order to simulate a poker environment.

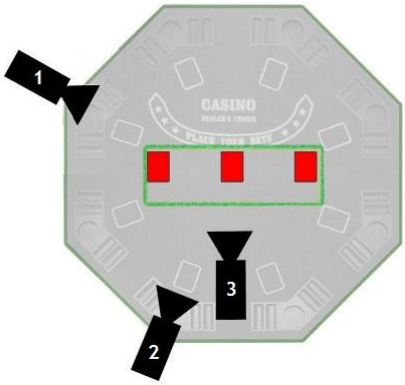


Fig. 7.1 – Overview of the table; the camera icons and respective numbers represent the tested positions/scenario of the overview webcam; the red rectangles represent the disposition of the cards

Fig. 7.2 – Pre-defined region for disposing the *community cards* in a professional poker game

For a better evaluation of the system's performance, and recalling the different methods used when identifying the cards, i.e. template matching and suit counting, the tests were split in two categories:

One of the tests covers the *face cards,* where each *face* was submitted for identification. This kind of testing is interesting since the method used for identification of this type of cards is based on templates, which will help to conclude on how the shape of the characters "J", "Q" and "K" influences the identification.

The results obtained were categorized as:

- **Correct**
- **Incorrect**
- **No match found** - if the template matching score was too high, i.e. bad match;
- **Identified as a *value card*** - in case the card was misidentified as a value card, thus submitted erroneously to the counting of suits;

The other test covers the identification of *value cards*, which is based on counting the suits within the card. Therefore, since this method is not dependent on special features of shapes, i.e. the suits are detected and counted or not, then these cards can be evaluated altogether. The results were categorized as:

- **Correct**
- **Incorrect**
- **Identified as a *face card*** - in case the card was misidentified as a *face card*

7.1.1 Scenario One:

The scenario and testing conditions were as follows:

First Test:

- The *overview webcam* was placed as close as possible to the table, with the tripod supported on the floor, i.e. positioned in a similar way to how human players sit at the poker table;
- The *webcam* was set to capture tightly the overview of the poker table, i.e. both the *community region* and the seats of each player;
- The capturing resolution was set to 640x480;

Second Test:

- The overview *webcam* was kept in the same place as in the first test;
- The capturing resolution was set to 800x600. Therefore the captured region at this resolution is broader than in the first test. This test is referred in the results as "800x600";

Third Test:

- The capturing resolution was set to 800x600;
- The *webcam* was set to capture tightly the overview of the poker table, i.e. both the *community region* and the seats of each player, as in the first test. This test is referred in the results as "800x600-2";

In Table 7.1, are listed the distances from the overview webcam to each card (Fig. 7.3 a) - a1/b1/c1) as well as the height of the camera relative to the poker table.

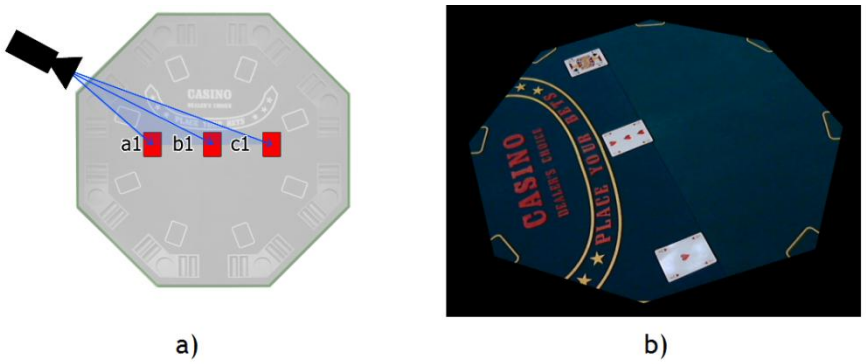


Fig. 7.3 - position of the overview *webcam a)* and its points of view b) @640x480 (800x400 is similar but the view is closer to the table). It is possible as well to see in both pictures the placement of the cards

Table 7.1 - Different setups for the tripod and for the different capturing resolutions. All the measurements are in centimeters and depicted in Fig. 7.3 a)

	640x480	800x600	800x600-2
a1	88	88	79
b1	103	103	93
c1	119	119	110
height	81	81	71

Results Obtained

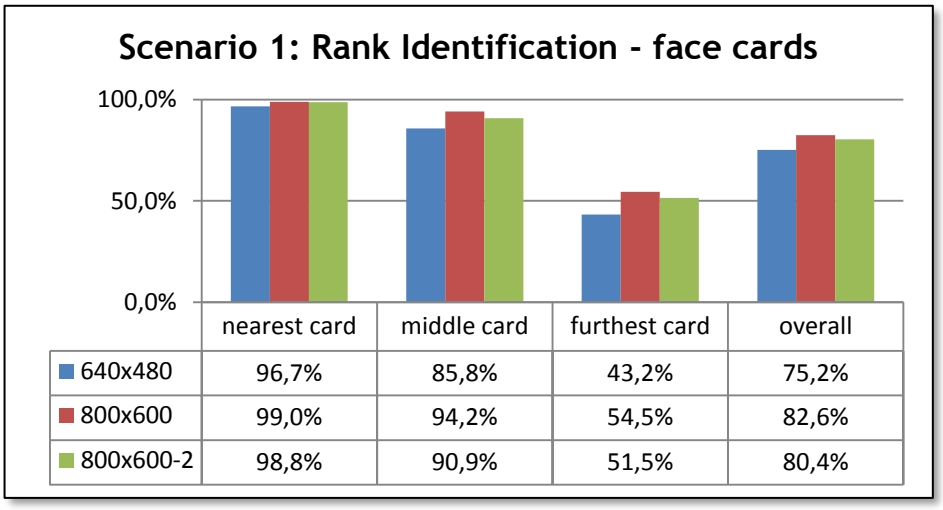


Fig. 7.4 - Results for the rank identification of the *face cards* - Test 1

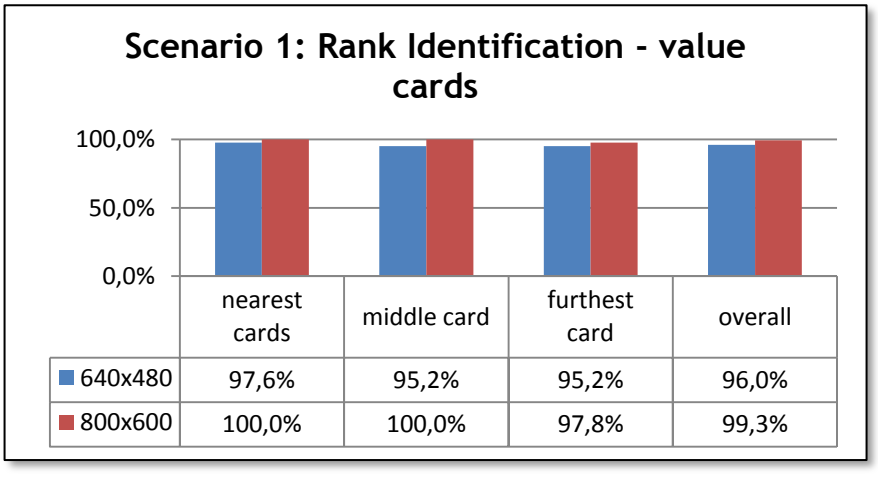
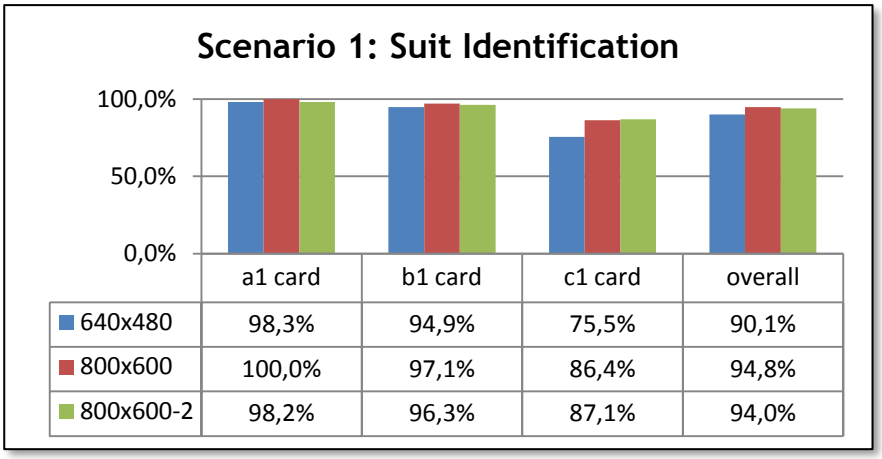


Fig. 7.5 - Results for the rank identification of the *value cards* plus the *Ace* - Test 1



	a1 card	b1 card	c1 card	overall
640x480	98,3%	94,9%	75,5%	90,1%
800x600	100,0%	97,1%	86,4%	94,8%
800x600-2	98,2%	96,3%	87,1%	94,0%

Fig. 7.6 - Results for the suit identification of the *face cards* - Test 1

Discussion

In these tests, the *webcam* was placed in the seat which offers the worst capturing conditions and angle of view to the *community cards*. More specifically, for the first and second tests, the furthest card is placed up to 119cm far from the *webcam*, making the recognition of the card through the captured image very difficult, even for a human being (Fig. 7.7). These tests were intended to conclude on how robust the perception system is, relative to the distance and to the rotation.

Fig. 7.7 - Example of the furthest card extracted from a captured image at 640x480. The image is at full scale

It is possible to conclude that better results were achieved in both tests with a resolution set to 800x600, in comparison with the results when the resolution was set to 640x480. This was already expected, since a higher resolution results in a more detailed image, thus a better recognition rate.

As expected, it is also possible to conclude that, the nearer the cards are, the better the recognition rates. The drop on the recognition rate of the *face cards* from the nearest to the furthest card, is situated between forty and fifty percent on all resolutions, which is a significant difference. On the other hand, the *value cards* do not suffer such a significant drop in the recognition rate along the three distances tested, since the method used for its recognition is not based on template matching but, instead, counting of suits.

Furthermore, it is interesting to notice that, although the 800x600 setup is positioned further from the cards, it features slightly better recognition rates than the 800x600-2. This is reasoned by the fact that the former is positioned higher, which results in a lower perspective distortion effect.

Within the *face cards*, the “King” features the best recognition rates on all resolutions, while the “Jack” appears to be the card with the worst recognition rate. This is reasoned by

the shape of the "Jack" as being thin, which can easily lose definition when it is extracted from the captured image or when it undergoes morphological operations.

Concerning the identification of suits, the furthest card presents again a lower accuracy when compared to the remaining. The resolution seems to affect the accuracy in a similar way as aforementioned, i.e. when the resolution is set to 800x600 better results are achieved than if it was set to 640x480. Furthermore, it is possible to spot that this low accuracy is mostly originated by the "Hearts" and "Spades" suits. In the case of "Hearts" it is recurrent being unable to find a match; in the case of "Spades" it is common to be misidentified as "Diamonds". The latter happens since due to the extraction process and morphological operations, the "Spades" suit sometimes has its shape eroded more than it would be desired, resulting in a different shape (Fig. 7.8)

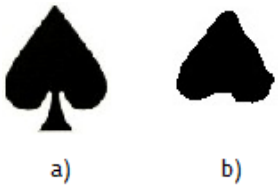


Fig. 7.8 - a) original shape of the "Spades" suit; b) situation in which the suit loses its characteristic shape;

7.1.2 Scenario Two:

Scenario Two represents the positioning of the webcam as in Fig. 7.1 – camera 2. In this test it is possible to capture a more frontal view of the cards than in the previous scenario. The first row of tests with this scenario covers different resolutions and tripod setups; in the second row is tried a deck of playing cards similar to the default one. The latter was intended to compare results and conclude on the robustness of the designed algorithm when different decks are used.

The scenario and testing conditions were as follows:

- The *overview webcam* was placed with the tripod supported partially on the poker table, without disturbing the gaming area (Fig. 7.9 a)). This positioning makes the webcam to be closer to the cards, than in "scenario one", while at the same time reducing perspective view effect;
- The *webcam* was set in order to capture the complete overview of the poker table, i.e. both the *community region* as well as the seats of each player (Fig. 7.9 b)). Due to being closer to the cards, the webcam will be positioned higher than in "scenario one";

Tests with the default deck:

- Two resolutions were used, 640x480 and 800x600. The *webcam* was set to capture tightly the overview of the poker table, i.e. both the *community region* and the seats of each player, nothing more;

Tests with a different deck:

- Only the resolution of 800x600 was used. Again, the *webcam* was set to capture tightly the overview of the poker table, i.e. both the *community region* and the seats of each player, nothing more;

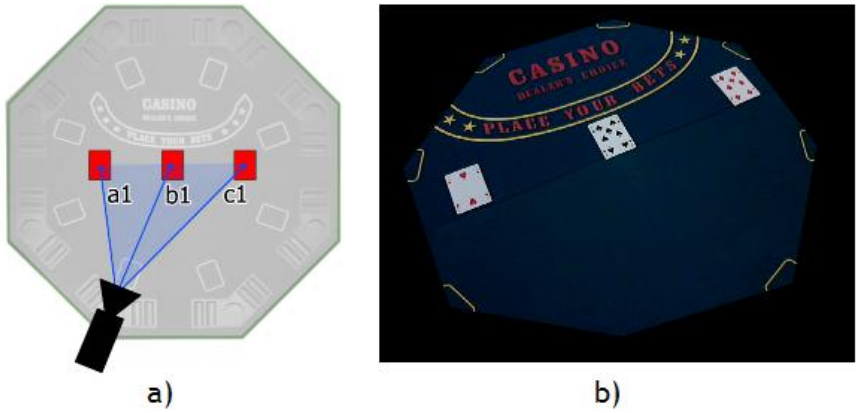


Fig. 7.9 - position of the overview *webcam a)* and its point of view b) @800x600. It is possible to see, in both pictures, the placement of the cards

Table 7.2 - Different setups for the tripod and for the different capturing resolutions. All the measurements are in centimeters and depicted in Fig. 7.9 a)

	640x480	800x600
a1	102	95
b1	104	98
c1	113	107
height	89	79

Default deck of playing cards: Results and Discussion

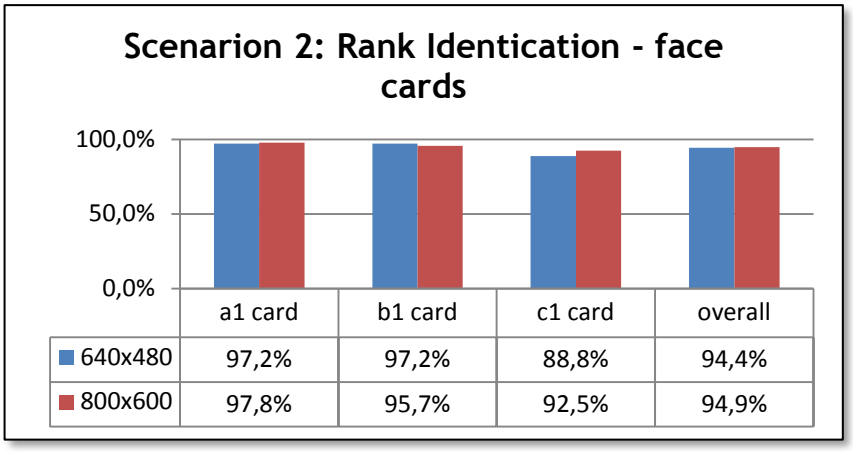


Fig. 7.10 - Accuracy results achieved for the rank identification (face cards) for both resolutions and relative to the distance to each card

The difference in accuracy between the resolutions of 800x600 and 640x480 obtained is not as remarkable as in Test 1. In the worst scenario, i.e. when comparing the rates for the furthest card, it does not exceed four percent, with the higher resolution prevailing.

The accuracy results achieved with the setup of this test, considering the furthest card from the camera, are remarkably better than the results obtained in Test 1. In "Test 1: 800x600-2" the furthest card is placed 110cm far from the camera and nevertheless presents worse results when compared to the furthest card in "Test 2: 640x480" which is placed 113cm away from the camera (and with less resolution). The gap in the accuracy between both is between thirty and forty percent. This gap can be as large as around fifty percent, comparing the results for the furthest card from "Test 2: 800x600" with "Test 1: 640x480".

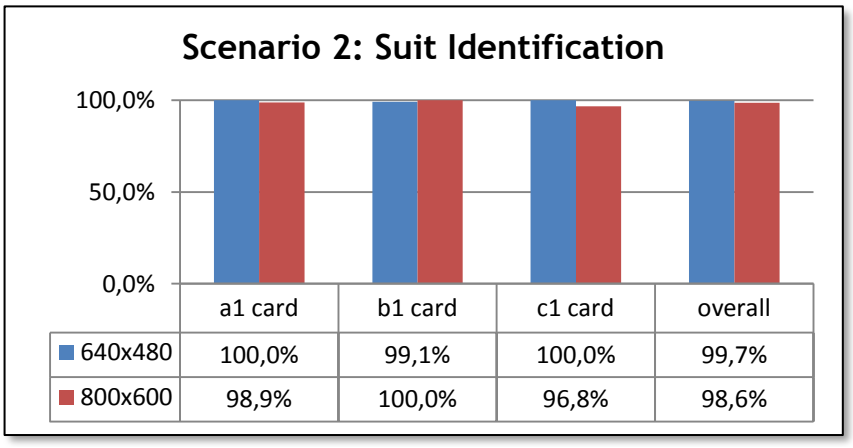


Fig. 7.11 - Accuracy results achieved for the suit identification for both resolutions and relative to the distance to each card

In this test the accuracy of the suit identification between both resolutions is about the same. However, comparing the results against the achieved in Test 1, it is noticeable that the accuracy achieved with this setup is better for the "b1 card" and remarkably better for the

"c1 card", where in the latter's the difference in performance is between ten and twenty percent.

Different Deck of Playing Cards: Results and Discussion

In this test was used a deck of cards slightly different from the original deck used along the work. No modifications were performed to the algorithm in order to recognize this new deck. The test with this new deck, in these conditions, is intended to test the robustness of the algorithm when slightly different cards are used.

Table 7.3 – Comparison table between the deck used along the work and the new deck.

Rank	The new cards feature a stronger type of letter.
Rank position	The new cards have only two suits depicted instead of four; however the algorithm is designed to crop exactly those two suits. The positioning of the rank within the card is the same for both.
Suit	The difference between the new suits and the templates used is small and practically unnoticeable
Suit position	The new deck's suits are positioned slightly towards the middle than the deck used.
other	The new deck has not a reflective surface, thus avoids light reflectance and camera blinding

The results achieved were compared against the results from previous test @800x600 and are shown in Fig. 7.12 and Fig. 7.13.

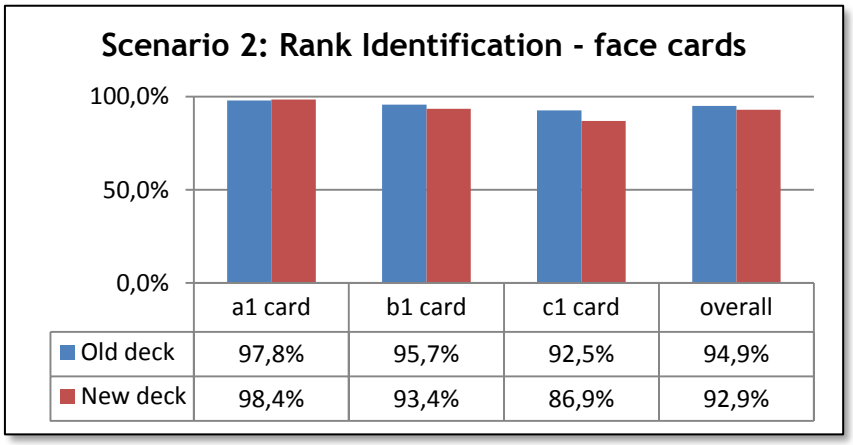


Fig. 7.12 - Accuracy results achieved for the rank identification (face cards) for both decks, relative to the distance to each card @800x600

The results achieved for the nearest two cards show that the recognition performance of the new deck is about the same as for the old deck. Moreover, the furthest card's

performance is about five percent better for the old deck; nevertheless such difference does not allow coming to a consistent conclusion about which deck performs best.

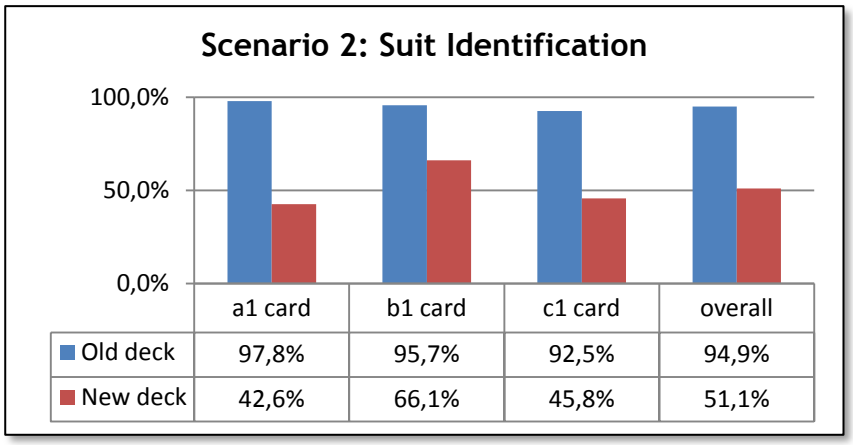


Fig. 7.13 - Accuracy results achieved for the suit identification for both decks, relative to the distance to each card @800x600

The performance of the suit recognition for the new deck is notoriously weak, comparatively to the old deck. This is mainly due to how close the suit is to the picture in the card (Fig. 7.14 a) b)). After converting to gray, binarizing and performing the morphological operations, the suit ends connected to the rest of the figure (Fig. 7.14 d)). Therefore, when the isolation of the suit is performed, it results in a null image not feasible of being identified.

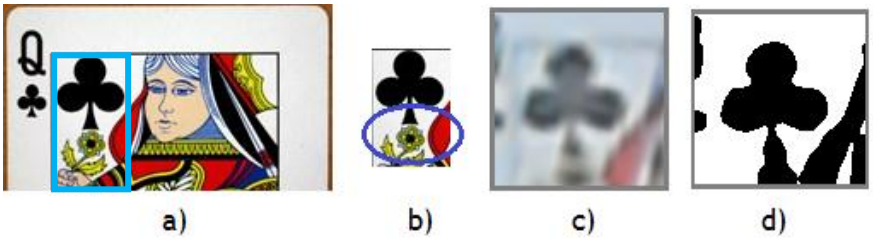


Fig. 7.14 – Unsuccessful suit extraction

7.1.3 Scenario Three: Optimal Position

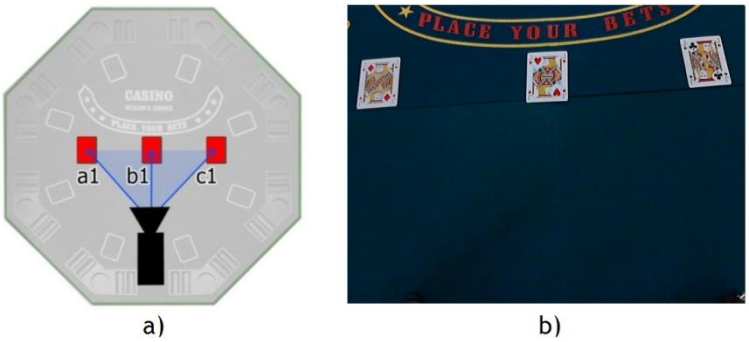


Fig. 7.15 – a) disposition of the *webcam* and its point of view b)

In this test the tripod was fully supported on the poker table, invading slightly the *community area*. Moreover, the placement of the tripod was not according with the placement of the seats, instead it was completely aligned with the disposition of the *community cards*, Fig. 7.15 a) and b).

Table 7.4 - distances from the camera to the cards and height of the camera

	640x480
a1	51
b1	47
c1	51
height	67

Although the setup of the tripod in this test is not in conformity with the concept described in Chapter 1, the aim of this test was to determine if the algorithm presents some kind of limitation. Considering how it is positioned and the distance of the webcam to the cards, it is expected the algorithm to be one hundred percent accurate, if not then the algorithm is limited somehow.

The results of this test are not stated in the form of a table **since it achieved one hundred percent of accuracy in the recognition of the *face cards*, *value cards* and suits**. This result covers all the rotations possible, except when the card is horizontally transversal to the *webcam*. In this situation the algorithm does not handle the identification of the card's position correctly and produces unexpected results. Therefore, the block which identifies the position of the card should be revised.

7.1.4 Scenario Four: KEM Playing Cards

The KEM playing cards (Fig. 7.16) are the most used in casino poker rooms in the USA and are as well the official cards of the World Series of Poker [38], which is one of the two biggest poker tournaments in the world. The KEM cards feature bigger sized ranks

comparatively to the cards used along the work, and also have a big sized suit under the rank. Moreover, unlike the playing cards used along the work, the KEM cards are not light reflective, thus not blinding the camera (Fig. 7.17).

In a later stage of this work a deck of this kind of cards was available, therefore the algorithm was slightly modified to enable the recognition of the KEM cards.

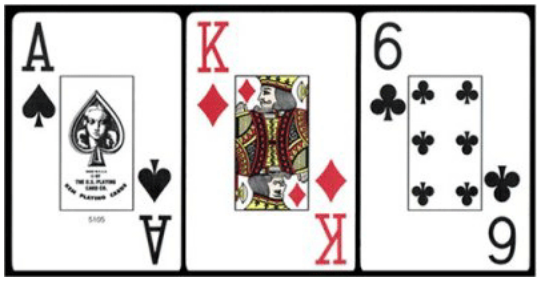


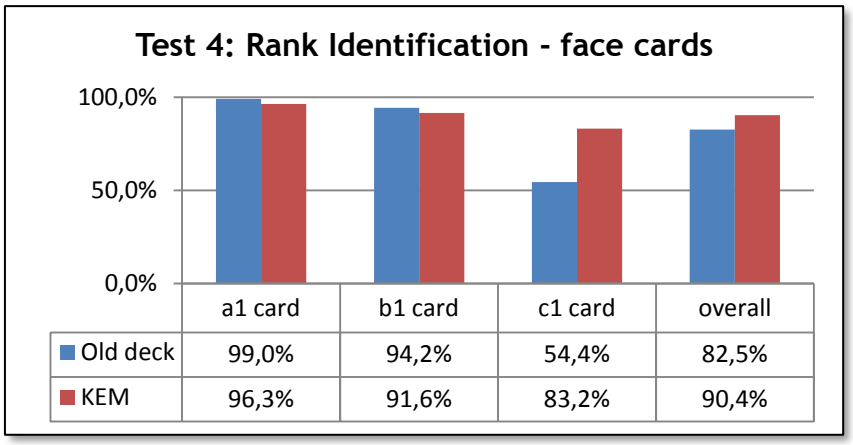
Fig. 7.16 – Example of the KEM cards

Fig. 7.17 – a) one random card from the deck used along the work, reflecting the light and slightly blinding the camera; b) a random KEM card, under the same illumination as a) and in exactly the same place, but not reflecting the light.

Testing conditions:

- The position on the poker table used was the same as in Scenario One;
- The resolution was set to 800x600 and the tripod was set accordingly;
- For the recognition of the cards, **the count of suits technique was not used** due to the smaller size of the suits depicted. Instead, the template matching method was used for the complete range of cards. Therefore each rank extracted is compared against thirteen templates each time the algorithm is run. The templates used were the same used so far, i.e. no templates were created with the purpose of specifically testing these cards;

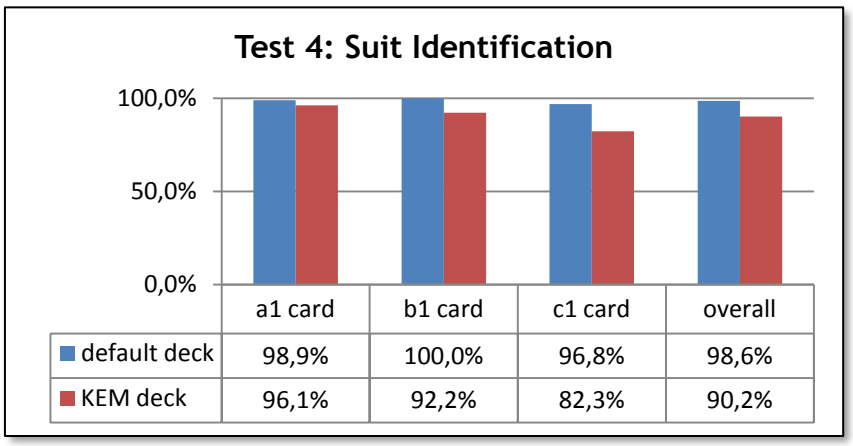
Results and Discussion



	a1 card	b1 card	c1 card	overall
Old deck	99,0%	94,2%	54,4%	82,5%
KEM	96,3%	91,6%	83,2%	90,4%

Fig. 7.18 - Accuracy results achieved for the rank identification (face cards) for both decks, relative to the distance to each card @800x600

The KEM playing cards achieve a more homogeneous accuracy along the three distances tested, comparatively to the deck used along the work. For the first two cards the accuracy achieved is about the same for both kinds of cards. The biggest difference in the accuracy concerns the furthest card, in which the KEM cards achieve an accuracy about 30% higher than the old deck. This is reasoned by the bigger sized rank, comparatively to the former deck. Moreover, the non-reflective surface of these cards benefit the identification as well.



	a1 card	b1 card	c1 card	overall
default deck	98,9%	100,0%	96,8%	98,6%
KEM deck	96,1%	92,2%	82,3%	90,2%

Fig. 7.19 - Accuracy results achieved for the suit identification for both decks, relative to the distance to each card @800x600

In this test the KEM cards do not reveal to be as good as the default deck, when it comes to the suit identification. The biggest difference relies, again, in the furthest card, with a gap between both of about 15%. It is worth reminding that the templates used for the suit identification are specific for the default deck. Therefore, it is expected that the accuracy achieved would improve if proper templates were prepared for the KEM cards.

7.2 Chips Identification

Two tests were performed to the chips' counting algorithm. The first was intended to determine the accuracy of the counting when the chips are disposed along the table and fully visible. The second test was intended to determine the accuracy of the counting when the chips are partially occluded.

In both tests, ten chips of each colour were used and submitted to counting. The number of chips returned by the algorithm was recorded in order to analyze how the colour and its contrast affect the counting and how accurate is the counting of each colour in each situation. This was performed thirty times for each placement and colour.

The setup of the tripod and resolution was the same as used in Test 1: 640x480. No variations were performed to this setup along the tests, since the perspective view of chips does not influence the recognition as it happens in the case of the cards. Within the captured area of the poker table two regions were defined by splitting the captured image overview of the poker table concerning the distance to the *webcam,*

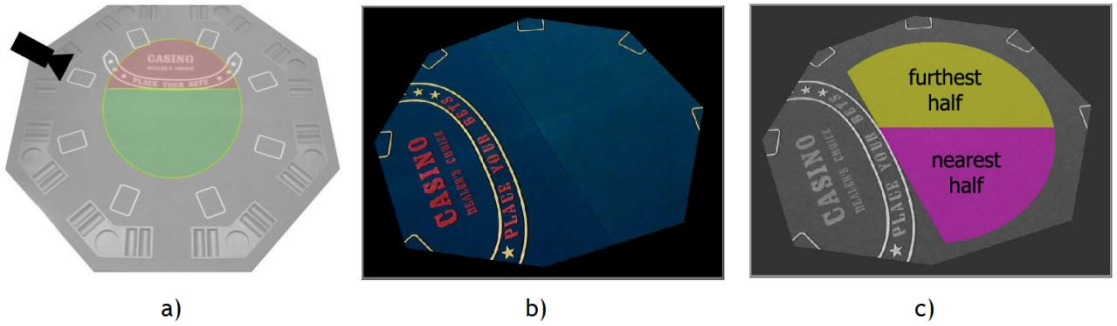


Fig. 7.20 – a) position of the camera during the tests. The green region represents the region used to place the chips; the red region was not used to place the chips due to the letters drawn, which would conflict with the counting; b) point of view of the *webcam* during the tests; c) regions defined and used during the tests to analyze the results relative to the distance of the *webcam*

7.2.1 Test 1: Chips Fully Visible

For this test, ten chips of each colour were placed randomly in the table (one colour at a time). The counting was then performed and the resultant counting result was registered. This was performed twenty five times for each colour.

The segmentation of the black chips conflicts with the colour of the poker table, therefore making the analysis impossible, Fig. 7.21.

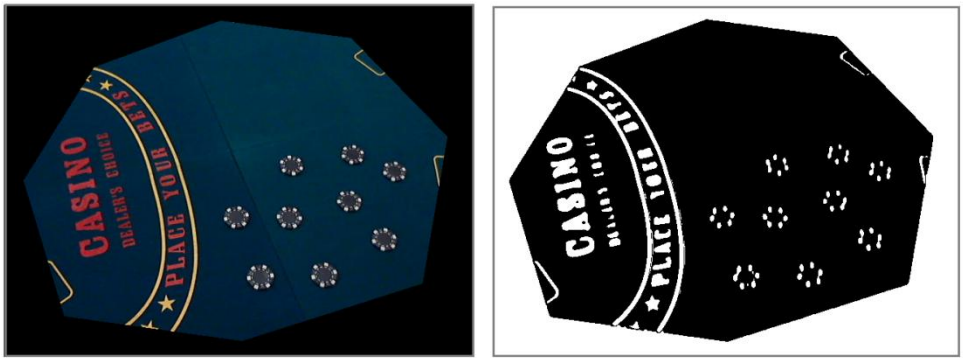


Fig. 7.21 – Overview of the poker table with the black chips disposed in a) and resultant binarized image in b). The chips in a) are placed in the nearest half of the table

The segmentation of the red colour conflicts with the red letters painted on the table (Fig. 7.22). However the common professional poker tables do not feature such characteristic, therefore during the tests performed the region of the table where the letters are was covered.

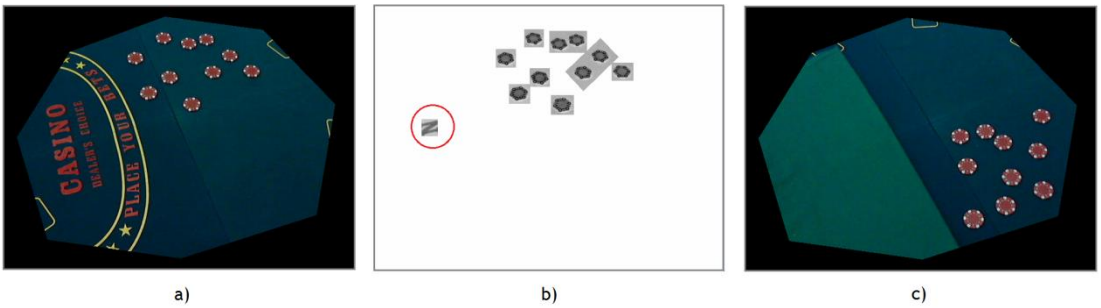


Fig. 7.22 - Overview of the table and the red chips a) and respective isolated red objects in b), including one of the letters of the poker table marked in a red circle; c) region with the letters covered. Moreover, the chips in a) are placed in the furthest half of the table, while in c) are placed in the nearest half

Results

Table 7.5 - White Chips: resultant counting relative to the *webcam's* distance; 10 chips is the correct number of chips placed in the table;

	10
Furthest half	**100,0%**
Nearest half	**100,0%**

Table 7.6 – Red Chips: resultant counting relative to the *webcam's* distance; 10 chips is the correct number of chips placed in the table;

	11	***10***	*9*	*8*	*7*
Furthest half	0,0%	**12,0%**	28,0%	40,0%	20,0%
Nearest half	32,0%	**68,0%**	0	0	0

Table 7.7 – Blue Chips: resultant counting relative to the *webcam's* distance; 10 chips is the correct number of chips placed in the table;

	10	*9*	*8*	*7*	*6*
Furthest half	**4,0%**	8,0%	52,0%	24,0%	12,0%
Nearest half	**96,0%**	4,0%	0	0	0

Table 7.8 – Green Chips: resultant counting relative to the *webcam's* distance; 10 chips is the correct number of chips placed in the table;

	10	*9*
Furthest half	**68,0%**	32,0%
Nearest half	**100,0%**	0,0%

Discussion

The white chips revealed 100% accuracy when placed anywhere in the table. This is due to the high contrast between the chips and the poker table. Undoubtedly, the white chips revealed the best accuracy among the different colours of chips.

As for the remaining chips, the accuracy on counting precisely 10 chips was better when these were placed in the nearest half of the table. Namely the red and blue chips, which feature an increase in the number of times the algorithm counted precisely 10 chips of about 55% and 80% respectively, when moving the chips from the furthest to the nearest half.

It is curious to observe that, when counting the red chips, 11 chips were counted 32% of the times when placed in the nearest half of the table. This is explained by the small dents featured in the design of the chip (Fig. 7.23), which are misidentified as circles in some situations.

Fig. 7.23 – Small elements liable of being detected as circles

7.2.2 Test 2: Chips Partially Occluded

For this test, ten chips of one colour were placed in the table forming a row and superimposing each other. The tests and registered results covered an occlusion of each chip around fifty percent, as well as an occlusion of thirty percent, Fig. 7.24. This procedure was performed twenty five times to each one of the five different colours of chips. It was registered how many chips were counted by the algorithm each time it was run.

Fig. 7.24 – Ten chips, where each one is about fifty percent occluded

Results

Table 7.9 – Results for partially occluded white chips

White Chips					
	50% occluded				
	10	9	8	7	6
	0,0%	5,0%	57,5%	27,5%	10,0%
	30% occluded				
	12	11	10	9	8
	5,0%	2,5%	**85,0%**	5,0%	2,5%

Table 7.10 - Results for partially occluded red chips

Red Chips				
	50% occluded			
	5	4	3	2
	12,5%	25,0%	40,0%	22,5%
	30% occluded			
	11	10	9	8
	10,0%	**62,5%**	27,5%	0,0%

Table 7.11 - Results for partially occluded blue chips

Blue Chips				
	50% occluded			
	8	7	6	5
	8,0%	76,0%	16,0%	0,0%
	30% occluded			
	10	9	8	7
	4,0%	68,0%	28,0%	0,0%

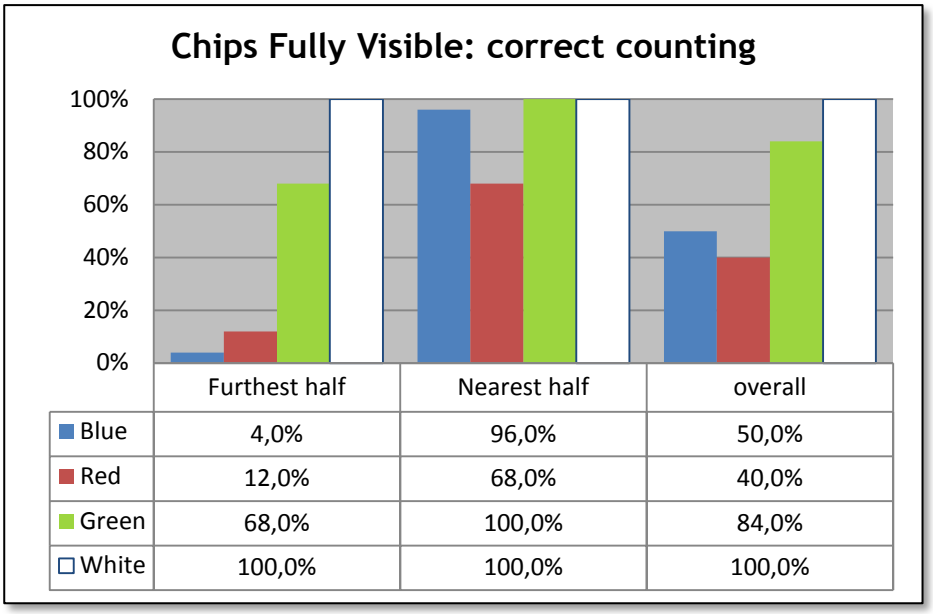
Table 7.12 - Results for partially occluded green chips

Green Chips				
	50% occluded			
	8	7	6	5
	12,0%	36,0%	48,0%	4,0%
	30% occluded			
	10	9	8	7
	48,0%	12,0%	4,0%	0,0%

Discussion

These tests were performed only on the nearest half of the poker table (Fig. 7.20 c)) since the algorithm revealed to be completely unreliable when the chips were placed on the furthest half of the table. The majority of results in this situation, i.e. furthest half of the table, output a counting between none and one chip found, whatever the occlusion percentage was. Therefore, it can be concluded that the algorithm is not able to count partially occluded chips, if these are placed more than about 100 cm from the *webcam*.

The accuracy of the chips' counting is not as critical as the identification of the cards, since the pot size is not directly related with winning or losing a hand as the *community cards*/*hole cards* are. The algorithm can be better analysed if it is contextualized with the importance of the correct counting of the chips during the game play. Therefore, the counting of precisely 10 chips was grouped with an error of 10%. More specifically, in the specific case of the tests performed, it is considered correct a result of nine, ten or eleven chips.

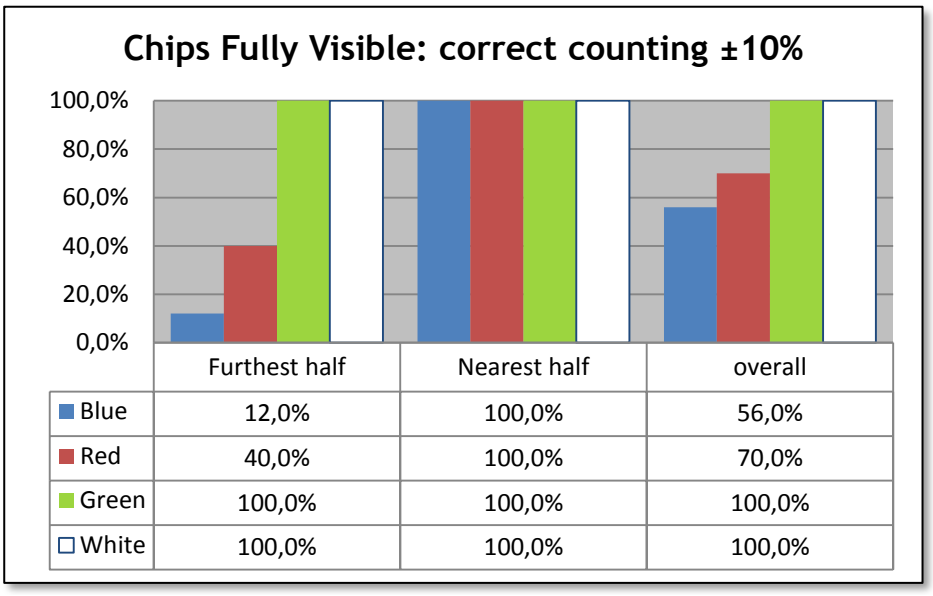


	Furthest half	Nearest half	overall
Blue	4,0%	96,0%	50,0%
Red	12,0%	68,0%	40,0%
Green	68,0%	100,0%	84,0%
White	100,0%	100,0%	100,0%

Fig. 7.25 – Accuracy of the chips' counting when these are fully visible

The performance of the chips' counting for the blue and red chips, presents a high variance considering the placement on the poker table. When positioned in the furthest half, both kind of chips present detection accuracies between five and fifteen percent; on the other hand, when positioned in the nearest half, the accuracy is situated between seventy and ninety percent, with the blue chips having an exceptional increase in the accuracy around ninety percent.

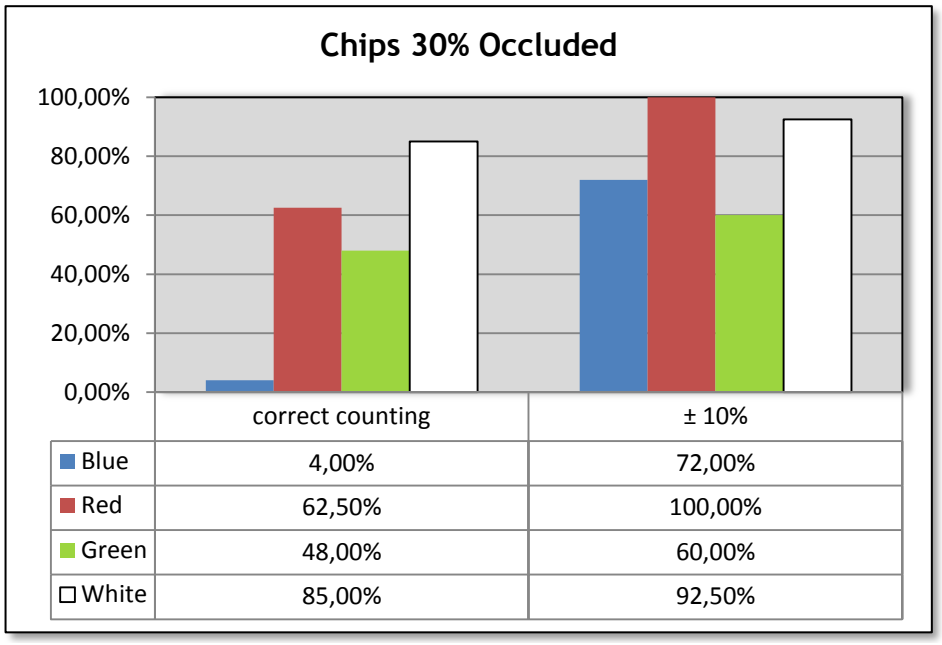
Considering the green and white chips, the accuracy rate is more homogeneous along the range of distances possible. Moreover, the white chips achieve total accuracy in both parts of the poker table.



	Furthest half	Nearest half	overall
Blue	12,0%	100,0%	56,0%
Red	40,0%	100,0%	70,0%
Green	100,0%	100,0%	100,0%
White	100,0%	100,0%	100,0%

Fig. 7.26 – Accuracy of the chips' counting when these are fully visible, with an acceptable mistake up to 10%

Although the counting of the green chips, placed in the furthest half, and the counting of the red and blue chips, both placed in the nearest half, did not achieve 100% counting accuracy in the first test, it is possible to conclude that every time the counting was wrong, the mistake was only by 10%. The same does not apply to the red and blue chips placed on the furthest half, and analyzing the chart it is possible to conclude that more than 60% of the counting outputs had an error higher than 10%.



	correct counting	± 10%
Blue	4,00%	72,00%
Red	62,50%	100,00%
Green	48,00%	60,00%
White	85,00%	92,50%

Fig. 7.27 – Accuracy of the chips' counting when these are partially occluded and positioned in the nearest half of the table; results grouped by mistake acceptance – 0% and 10%

The improvement in the percentage of correct counting is noticeable after considering as correct a range of 10%, namely when counting the blue and the red chips. This proves that although the algorithm fails to precisely count the chips in some occasions, the incorrectness, most of the times, is no more than 10%, which is completely acceptable since the number of chips, i.e. total value in the table, does not play a major role during the decision within the game play.

7.3 Conclusion

This chapter presented the tests performed to the algorithms designed to identify playing cards and chips.

The playing cards identification algorithm revealed its weakness when tested in the conditions of Scenario One, where it achieved its worst result of about 50% with concern to the furthest card being recognized. However, this situation was proven to be overcome if the KEM playing cards are used, which happen to be the most used playing cards in professional poker games. In the remaining situations, the results achieved vary between 95% and 99%.

The recognition of chips was found to be dependent on the distance to the webcam and on the contrast between their colour and the colour of the poker table. If the chips which feature a low contrast with the poker table, such as the red or blue chips, are placed in the furthest half of the table, then the counting cannot be expected to be reliable. Moreover, if there are occluded chips, it cannot be over than 30% and have to be placed in the nearest half of the table for a correct counting.

Chapter 8

Conclusions and Future Work

8.1 Goal Achievement

The purpose of this research work was to create a vision based perception layer for a poker game, reliable in a real and uncontrolled environment. The general requirements of such layer included the recognition of playing cards and chips and, furthermore, the complete following of the game play. All the gathered information is supposed to be sent to the upper layer, i.e. the poker agent – out of this thesis' context.

In this document the game of poker was introduced briefly, together with some research work on automated card recognition systems. This overview enabled to conclude that although some work has been developed around the recognition of cards and almost none on chips, the concept which originated this thesis has not been developed yet to the present date.

The first task, and the most important, to be developed was the recognition of playing cards. The approach developed relies on a combination of template matching with suits' counting. The recognition of the playing cards was accomplished with an overall accuracy for the rank recognition around 95% for the face cards and slightly more than 99% for both the value cards and the suits' recognition. There is a total of fifty two cards, where twelve of them are face cards and the forty remaining are the cards two through ten plus the ace. Therefore, the weighted mean of the rank recognition accuracy for the fifty two cards is about 98%. These accuracy rates correspond to the test which scored the best accuracy.

A full template matching approach was tested with the KEM cards, since these feature ranks big enough to be extracted with no major loss of quality, while at the same time the suits depicted in these cards are too small to perform the suit counting. The tests performed only concerned the face cards, however the results achieved surpassed the results achieved with the default cards.

The counting of chips was also a priority during the implementation, since it provides useful information during the game play when it comes to making a decision. The results achieved show that with the presented method it is possible to expect a reliability between

50% and 100%, depending on the colour, in case the chips are fully visible, even if placed along the whole table; however, in the case the chips are partially occluded, up to 30%, the reliability floats between 60 and 100%, in the case these are placed near the camera.

Although the results achieved for the counting of chips are not as remarkable as the results for the card recognition, both should not be compared against each other due to their different natures. When counting chips, the biggest challenge relies on detecting the occluded ones, specifically when completely occluded. It is worth mentioning that, when it comes to counting chips in an ordinary poker game, human players not only rely on their vision but also on the feel of their hands to sense how much chips they are holding, which in the context of this work is not possible.

Concerning the adaptability of the system to different illumination scenarios, it is far from successful. Actually, the perception layer does not handle light dimming well, requiring the webcam to be re-configured every time there are changes in the scenarios.

The speech recognition was tested in a scenario with one person dictating the sentences present in the grammar rules, along with random phrases and sounds. The speech recognition did not misidentify any random phrase or sound as one of the grammar rules implemented. However, when stating one of the grammar rules, occasionally it was needed to repeat the sentence.

The speech recognition implemented was not subject to a stress test, such as a scenario with multiple players speaking at the same time; therefore it is not possible to conclude about its reliability in a real poker environment.

As for determining the players in the start of a game, as well as the Big Blind position, the implementation was successful. However, the reliability is dependent on the behaviour of the human players, which have to wait for all the *hole cards* to be dealt before checking them, in order for the algorithm to determine at once who is in or out of the game.

8.2 Future Work

Throughout this project, the goal was always the development of a robust playing cards' identification and chips' counting system. After performing the tests, conclusions were made and some questions were answered. However, the work presented in this thesis is far from being completely developed, actually there are features implemented which can be refined, as well as additional features which can be implemented.

As referred previously, the implemented perception layer aimed at the cards described in Chapter 4. In a later stage of this work, the KEM cards were available and tested after little modifications to the implemented algorithm. The accuracy achieved with these cards was better than the best accuracy achieved with the former cards. Since these cards are widely used in official poker tournaments, an improved version of this perception layer should explore the features of these cards in order to achieve the best accuracy rating possible.

One of the most important tasks to develop in the future, concerns the implementation of a block which verifies the output of the cards' recognition and detects possible repeated

identifications of cards. During a poker game it is known that the cards are all from the same deck, thus there are no repeated cards. However, the recognition algorithm was designed to handle a group of cards as single cards instead of a group of cards from the same deck. Therefore, since the recognition algorithm is not completely fail proof yet, it is possible that within a group of different cards subject to recognition, two or more different cards are identified as the same, which is impossible if the cards are from the same deck as they are supposed to be. Therefore, it is missing a block which compares the final recognition results performed to a group of cards, preventing this kind of situation by selecting only one of the cards. This filtering should be based on the recognition score.

It is of interest to implement card differentiation by colour, in order to prevent possible misidentifications between black and red suits. As an example, it was proven to be frequent the “Spades” suit (black) being misidentified as “Diamonds” (red).

The setup used along this work is unreliable when the playing cards reflect the light directly to the camera lens, thus blinding the camera. The blinding effect can be reduced by adopting an infra red light source, pointing directly at the *community* region of the poker table. The infra red light source is the most suitable for such environment since the human eye cannot see light in the infra red spectrum, but cameras can. In case non reflective cards are used, such as the KEM cards, the reflectance is diminished. Nevertheless, should be adopted a capturing device which allows the basic configuration, such as brightness, contrast, white balance, to be modified externally by software. This way it would be possible to automatically adjust the camera’s configuration.

Concerning the counting of chips, an approach based on stereo vision could improve the accuracy. Stereo vision is intended to give the sense of depth of field, which combined with 2d images could improve substantially the detection of overlapping chips and even stacks of chips [39].

References

[1] L. Teófilo, "Building a Poker Playing Agent based on Game Logs using Supervised Learning", MSc Thesis, Faculdade de Engenharia, Universidade do Porto, Porto, 2010.

[2] "GOING ALL IN FOR ONLINE POKER", *Newsweek*, August 2005.

[3] J. Kirk, "The Poker Boom Part 1: Where it all began", *Poker Listings*, February 2008.

[4] *PokerAI*. Available at: www.pokerai.org (last accessed on January 2011)

[5] T. Bochan. (2006). *Understand Position in Poker - Poker Position Explained*. Available at: http://poker.about.com/od/poker101/ss/pokerposition.htm (last accessed on January 2011)

[6] PokerListings.com. (2007). *Texas Hold'em Rules and Game Play*. Available at: www.pokerlistings.com/poker-rules-texas-holdem (last accessed on January 2011)

[7] J. McLeod. (2005). *Rules of Poker*. Available at: www.pagat.com/poker/history.html (last accessed on January 2011)

[8] J. McLeod. (2005). *Poker Betting and Showdown*. Available at: www.pagat.com/poker/history.html (last accessed on January 2011)

[9] H. Orenstein, "Poker tournament", United Sates of America Patent 5,451,054, 1995.

[10] (June 2006, January 2011). History of the Hole Card Camera. Available at: http://www.hollywoodpoker.com/green-room/poker-lifestyle/poker-news/pocket-cam-20060601.html

[11] F. Lewis, K. Lewis, and L. Rogers, "Automated Playing Card Identification System For Casino-Type Card Games", United States of America Patent, 2005.

[12] *What is RFID?* Available at: http://rfid.net/basics/190-what-is-rfid (last accessed on January 2011)

[13] A. Miller. (2008). *The Video Poker Table*. Available at: http://videopokertable.net/rfid/index.html (last accessed on January 2011)

[14] *Tangam Gaming*. Available at: www.tangamgaming.com (last accessed on January 2011)

[15] D. Terdiman, "Who's Holding the Aces Now? ", *New York Post*, 2003.

[16] *Poker Tronic GmbH*. Available at: www.pokertronic.de (last accessed on January 2011)

[17] *Flashlight Design Electronic System GmbH*. Available at: www.value-poker.com (last accessed on January 2011)

[18] *Perlego automated card recognition and broadcasting services*. Available at: www.pressassociation.com/sport/broadcast-services/perlego.html (last accessed on January 2011)

[19] H. Swains, "UKIPT Nottingham: PokerStars to trial new hole card technology", *PokerStars News*, May 2010.

[20] A. Zaworka, "Machine Vision Driven Real-time Blackjack Analysis", MSc Thesis, Institut für Maschinelles Sehen und Darstellen Technische Universität Graz, Graz, August 2001.

[21] C. Zheng and R. Green, "Playing Card Recognition Using Rotational Invariant Template Matching", Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand, December 2007.

[22] G. Hollinger and N. Ward, "Introducing Computers to Blackjack: Implementation of a Card Recognition System using Computer Vision Techniques", Department of Computer Science, Colby College, Waterville, Maine, 2003.

[23] K. Zutis and J. Hoey, "Who's Counting?: Real-Time Blackjack Monitoring for Card Counting Detection", School of Computing, University of Dundee, 2009.

[24] W.-Y. Chen and C.-H. Chung, "Robust poker image recognition scheme in playing card machine using Hotelling transform, DCT and run-length techniques", in *Digital Signal Processing*. vol. 20, 3 ed, May 2010, pp. 769-779.
[25] R. Clements, "Eye of Judgment Review", ed: IGN - UK Edition, October 2007.
[26] G. Mastrapa, "Sony Shuts Down Online Eye of Judgement PS3 Game", *Wired,* July 2010.
[27] *Casino Token*. Available at: http://en.wikipedia.org/wiki/Poker_chip (last accessed on January 2011)
[28] D. Merrill. *Pixel Size*. Available at: www.outbackphoto.com/dp_essentials/dp_essentials_02/essay.html (last accessed on January 2011)
[29] R. Fisher, S. Perkins, A. Walker, and E. Wolfart. (2000). *Image Processing Operator Worksheets*. Available at: http://homepages.inf.ed.ac.uk/rbf/HIPR2/wksheets.htm (last accessed on January 2011)
[30] G. Bradski and A. Kaehler, *Learning OpenCV*. Beijing: O'Reilly Media, 2008.
[31] J. Canny, "A Computational Approach to Edge Detection", *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, pp. 679-714, 1986.
[32] MathWorks. Available at: www.mathworks.com (last accessed on January 2011)
[33] J. Rodgers and A. Nicewander, "Thirteen Ways to Look at the Correlation Coefficient", *The American Statistician,* vol. 42, pp. 59-66, 1988.
[34] M. Bénallal and J. Meunier, "Real-time color segmentation of road signs", Department of Computer Science, University of Montreal, Montreal, May 2003.
[35] G. H. Joblove and D. Greenberg, "Color Spaces for Computer Graphics", in *SIGGRAPH '78 Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, New York, USA, 1978.
[36] *RGB color space*. Available at: http://en.wikipedia.org/wiki/RGB_color_space (last accessed on January 2011)
[37] H. Rhody, "Lecture 10: Hough Circle Transform", Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, October 2005.
[38] *World Series of Poker*. Available at: www.wsop.com (last accessed on January 2011)
[39] S. Barnard and M. Fischler, "Computational Stereo", *ACM Computing Surveys,* vol. 14, pp. 553-572, December 1982.

Appendix A

Card Recognition: Test 1 @640x480

		J	*Q*	*K*	*correct*	*total*	*acc*
88 cm	Correct	59	60	59	177	183	**96,7%**
	Incorrect	1		2			
	No Match	2	1				
103 cm	Correct	49	51	57	157	183	**85,8%**
	Incorrect	12	4	3			
	No Match		5	1			
	Value card		1				
119 cm	Correct	21	21	37	79	183	**43,2%**
	Incorrect	37	18	13			
	No Match	2	14	8			
	Value card	1	8	3			
stats	Tot. correct	128	132	153			
	Total tests	183	183	183			
	Accuracy	**69,9%**	**72,1%**	**83,6%**			**75,2%**

		Hearts	***Diamonds***	***Spades***	***Clubs***	*correct*	*total*	*accuracy*
88 cm	Correct	44	44	44	45	177	180	**98,3%**
	Incorrect			1				
	No Match		1	1				
103 cm	Correct	39	44	40	44	167	176	**94,9%**
	Incorrect			4				
	No Match	5						
119 cm	Correct	26	32	28	27	111	147	**75,5%**
	Incorrect			6				
	No Match	12	6	1	9			
stats		107	120	112	116			
		126	127	125	125			
		84,9%	**94,5%**	**89,6%**	**92,8%**			**90,1%**

Card Recognition: Test 1 @800x600

		J	*Q*	*K*	*correct*	*total*	*accuracy*
88 cm	Correct	33	34	35	102	103	**99,0%**
	Incorrect						
	No Match	1					
103 cm	Correct	30	32	35	97	103	**94,2%**
	Incorrect	4	2				
	No Match						
119 cm	Correct	10	14	32	56	103	**54,4%**
	Incorrect	16	16	1			
	No Match	6	5	2			
	Value card	1					
	Tot. Correct	73	80	102			
	Total tests	101	103	105			
	accuracy	**72,3%**	**77,7%**	**97,1%**			**82,4%**

		Hearts	***Diamonds***	***Spades***	***Clubs***	*correct*	*total*	*accuracy*
88 cm	Correct	26	26	26	24	102	102	**100,0%**
	Incorrect							
	No Match							
103 cm	Correct	22	28	24	26	100	103	**97,1%**
	Incorrect			1				
	No Match	2						
119 cm	Correct	18	23	17	19	77	89	**86,4%**
	Incorrect			4				
	No Match	6	1	1				
	Tot. correct	66	77	67	69			
	Total tests	74	78	73	69			
	accuracy	**89,2%**	**98,7%**	**91,8%**	**100,0%**			**94,8%**

Card Recognition: Test 1 @800x600-2

		J	*Q*	*K*	*correct*	*total*	*accuracy*
79 cm	Correct	53	55	55	163	165	**98,8%**
	Incorrect						
	No Match	2					
93 cm	Correct	48	51	51	150	165	**90,9%**
	Incorrect	7	3	2			
	No Match		1	2			
110 cm	Correct	21	24	40	85	165	**51,5%**
	Incorrect	31	26	13			
	No Match	3	5	2			
	Tot. correct	122	130	146			
	Total tests	165	165	165			
	accuracy	**73,9%**	**78,8%**	**88,5%**			**80,4%**

		Hearts	*Diamonds*	*Spades*	*Clubs*	*correct*	*total*	*accuracy*
79 cm	Correct	37	37	43	43	160	163	**98,2%**
	Incorrect							
	No Match	3						
93 cm	correct	40	41	36	39	156	162	**96,3%**
	Incorrect			1				
	No Match	3			2			
110 cm	Correct	33	39	31	32	135	155	**87,1%**
	Incorrect			10	1			
	No Match	7	1		1			
	Tot. correct	110	117	110	114			
	Total tests	123	118	121	118			
	accuracy	**89,4%**	**99,2%**	**90,9%**	**96,6%**			**94,0%**

Card Recognition: Test 2.1 @640x480

		J	*Q*	*K*	*correct*	*total*	*accuracy*
102 cm	Correct	35	34	35	104	107	**97,2%**
	Incorrect						
	No Match		3				
104 cm	Correct	34	35	35	104	107	**97,2%**
	Incorrect	1	2				
	No Match						
113 cm	Correct	31	30	34	95	107	**88,8%**
	Incorrect	4	7	1			
	No Match						
	Tot. correct	100	99	104			
	Total tests	105	111	105			
	accuracy	**95,2%**	**89,2%**	**99,0%**			**94,4%**

		Hearts	***Diamonds***	***Spades***	***Clubs***	*correct*	*total*	*accuracy*
102 cm	Correct	27	25	26	26	104	104	**100,0%**
	Incorrect							
	No Match							
104 cm	Correct	26	28	27	25	106	107	**99,1%**
	Incorrect							
	No Match	1						
113 cm	Correct	27	26	26	28	107	107	**100,0%**
	Incorrect							
	No Match							
	Tot. correct	80	79	79	79			
	Total tests	81	79	79	79			
	accuracy	**98,8%**	**100,0%**	**100,0%**	**100,0%**			**99,7%**

Card Recognition: Test 2.1/2.2/4 @800x600

		J	*Q*	*K*	*correct*	*total*	*accuracy*
95 cm	Correct	30	30	31	91	93	**97,8%**
	Incorrect	1	1				
	No Match						
98 cm	Correct	31	28	30	89	93	**95,7%**
	Incorrect		3	1			
	No Match						
107 cm	Correct	27	29	30	86	93	**92,5%**
	Incorrect	4	1	1			
	No Match						
	Value Card		1				
	Tot. correct	88	87	91			
	Total tests	93	93	93			
	accuracy	**94,6%**	**93,5%**	**97,8%**			**94,9%**

		Hearts	***Diamonds***	***Spades***	***Clubs***	*correct*	*total*	*accuracy*
95 cm	Correct	21	24	22	25	92	93	**98,9%**
	Incorrect				1			
	No Match							
98 cm	Correct	23	25	25	20	93	93	**100,0%**
	Incorrect							
	No Match							
107 cm	Correct	25	20	25	20	90	93	**96,8%**
	Incorrect			3				
	No Match							
	Tot. correct	69	69	72	65			
	Total tests	69	69	75	66			
	accuracy	**100,0%**	**100,0%**	**96,0%**	**98,5%**			**98,6%**

Card Recognition: Test 2.2 @800x600 – new deck

Table 8.1

		J	*Q*	*K*	*correct*	*total*	*accuracy*
95 cm	Correct	20	20	20	60	61	**98,4%**
	Incorrect						
	No Match	1					
98 cm	Correct	19	18	20	57	61	**93,4%**
	Incorrect	1					
	No Match						
	Value	1	2				
107 cm	Correct	18	18	17	53	61	**86,9%**
	Incorrect	2					
	No Match						
	Value	1	2	3			
	Tot. correct	57	56	57			
	Total tests	63	60	60			
	accuracy	**90,5%**	**93,3%**	**95,0%**			**92,9%**

Table 8.2

		Hearts	*Diamonds*	*Spades*	*Clubs*	*correct*	*total*	*accuracy*
95 cm	Correct		9	11	6	26	61	**42,6%**
	Incorrect			1				
	No Match	15	5	5	9			
98 cm	correct	7	12	11	7	37	56	**66,1%**
	Incorrect			1				
	No Match	8	2	1	7			
107 cm	Correct	3	11	8	5	27	59	**45,8%**
	Incorrect			3				
	No Match	10	4	6	9			
	Tot. correct	10	32	30	18			
	Total tests	43	43	47	43			
	accuracy	**23,3%**	**74,4%**	**63,8%**	**41,9%**			**51,1%**

Card Recognition: Test 4 @800x600 – KEM deck

		J	*Q*	*K*	*correct*	*total*	*accuracy*
88 cm	Correct	35	33	35	103	107	**96,3%**
	Incorrect		2				
	No Match		1	1			
103 cm	Correct	35	30	33	98	107	**91,6%**
	Incorrect		5				
	No Match		1	3			
119 cm	Correct	32	28	29	89	107	**83,2%**
	Incorrect	2	7	4			
	No Match	1	1	3			
	Tot. correct	102	91	97			
	Total tests	105	108	108			
	accuracy	**97,1%**	**84,3%**	**89,8%**			**90,4%**

		Hearts	*Diamonds*	*Spades*	*Clubs*	*correct*	*total*	*accuracy*
88 cm	Correct	25	26	22	26	99	103	**96,1%**
	Incorrect			4				
	No Match							
103 cm	correct	24	30	15	25	94	102	**92,2%**
	Incorrect			8				
	No Match							
119 cm	Correct	25	25	14	15	79	96	**82,3%**
	Incorrect			5	6			
	No Match		1	1	4			
	Tot. correct	74	81	51	66			
	Total tests	74	82	69	76			
	accuracy	**100,0%**	**98,8%**	**73,9%**	**86,8%**			**90,0%**