Poker Odds Calculator
using visual content

by

Rohit Aila

A Project Report Submitted
in
Partial Fulfillment of the
Requirements for the Degree of
Master of Science
in
Computer Science

Supervised by

Dr. Reynold Bailey

Department of Computer Science

B. Thomas Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, New York

December 2016
I dedicate this project to my supervisor Dr. Reynold Bailey, Graduate Advisor
Dr. Hans-Peter Bischof, the entire CS department, my friends, and family.
Acknowledgments

Thank you to Dr. Reynold Bailey, because of whom, this project exists today. Thank you to Dr. Hans-Peter Bischof, because of whom, I learned to program. Thank you to Prof. Srinivas Sridharan, because of whom, I have developed interest in the field of Computer Vision. Thank you to my friends and family, because of whom, I work harder every day. Thank you to the RIT CS Dept, because of which, I have my Masters degree. Thank you to my Mother and Father, because of whom, I exist.
Abstract

Poker Odds Calculator
using visual content

Rohit Aila

Supervising Professor: Dr. Reynold Bailey

It is a known fact that the number of moves in most of the board/card games is limited. Using this information, artificial intelligence is often used to simulate the probable moves of an opponent and optimal moves for the player to win. However, most of these programs are designed from the information of current game state and are often used in implementing digital games. In that sense, poker is a lot different from other games because the game state is always hidden in a poker game. Hence it is not easy to simulate the opponent’s moves in a poker game. The goal of this project is to predict the probability of winning of a poker hand by using digital camera or images captured through mobile phone. There are a lot of variants in a poker game. Texas Hold’em has been chosen as the variant that will be used in this project because of its popularity. Image processing techniques such as thresholding, edge detection and template matching are applied on the input image in order to detect the suit and rank of playing cards. The last step is to make use of poker statistics to calculate the odds of the ongoing poker game.
## Contents

Dedication .................................................................................. ii

Acknowledgments ...................................................................... iii

Abstract ..................................................................................... iv

1 Introduction ............................................................................. 1
   1.1 Problem Statement .......................................................... 1
   1.2 Motivation ........................................................................ 1
   1.3 Project Summary ............................................................. 2
   1.4 Agenda ............................................................................. 2

2 Background ............................................................................. 3
   2.1 Image Processing ............................................................. 3
   2.2 Preview of Poker ............................................................. 4
      2.2.1 Playing Texas Hold’em Poker .................................. 4
   2.3 Card Detection ................................................................. 5
   2.4 Probability Calculation .................................................... 7

3 Implementation ........................................................................ 8
   3.1 Platform, Tools, and Technology ..................................... 8
      3.1.1 Hardware Requirements and Specification ............ 8
      3.1.2 Software Tools ....................................................... 8
   3.2 System Architecture ....................................................... 9
   3.3 Creating Dictionary ........................................................ 10
   3.4 Thresholding ................................................................. 11
   3.5 Opening ................................................................. 12
   3.6 Edge Detection .............................................................. 12
   3.7 Hough Transform ......................................................... 13
   3.8 Card identification ......................................................... 15
3.8.1 Unsuccessful Attempts ........................................ 15
3.8.2 Implemented Technique ................................. 17
3.9 Probability Calculation ................................. 18

4 Results .............................................................. 20
  4.1 Card Detection ........................................ 20
  4.2 Probability Prediction ................................. 20
  4.3 GUI ....................................................... 21

5 Conclusions ..................................................... 22
  5.1 Current Status ........................................ 22
  5.2 Lessons Learned ........................................ 22

6 Future Work ................................................... 23

Bibliography ..................................................... 24
List of Tables

2.1 Table to compare evaluators of rank function . . . . . . . . . . . . . . . . 7
List of Figures

2.1 Poker Hand Rankings .................................................. 6

3.1 Card Detection Process ................................................. 9
3.2 Card Recognition Process ............................................. 9
3.3 Template of number 8 ................................................... 10
3.4 Template of suit symbol heart ......................................... 10
3.5 Binary format of test image after Otsu’s thresholding ............. 11
3.6 Result of image opening ................................................. 12
3.7 Result of Sobel edge detection ......................................... 13
3.8 Lines on playing cards using Hough Transform .................... 14
3.9 Corner points of cards .................................................. 14
3.10 SIFT matching percentage of card with clubs and diamonds .... 15
3.11 SURF results for suit detection ....................................... 16
3.12 Geometric features for suit detection ............................... 16
3.13 OCR Trainer labeling and result for card 10D ..................... 17
3.14 Rank and Suit Detection using Template Matching ............... 18
3.15 Probability Calculation for a particular set of cards ............. 19

4.1 MATLAB GUI showing results of card recognition and probability prediction 21
Chapter 1

Introduction

1.1 Problem Statement

Poker odds calculator can be used either to cheat in a poker game or calculate odds in a live poker game. The main challenge of this problem is to detect the suit and rank of playing cards based on a stream of images which will be captured by using a web-cam hidden in the player’s shirt button or on tie. The suit and rank of the playing cards can be detected by using image processing algorithms such as SIFT, OCR or Template matching. After detecting the rank and suit of playing cards, Poker statistics can be applied to compute the winning odds of a particular player.

1.2 Motivation

In major poker tournaments such as World Series of Poker, the winning odds of players is shown with the information about the cards of each player. But when the games go live, there is nothing as such shown in the games. Hence, there is a need for a tool which gives a winning probability of a player in live poker games. By using the information about rank and suit of playing cards and a little knowledge of poker statistics, such tool can be built which can be applicable to show winning probabilities of a player. These probabilities can be also used to cheat in a poker game if a small camera is placed on the player’s shirt button or tie stealthily.
1.3 Project Summary

For this project, an interactive GUI was developed using Matlab App designer which can be used to capture input images, show the detected cards and to show the winning probability of a player based on detected cards. The images are captured using a webcam present in the RIT graphics lab. A green cloth is placed on the table to capture images to give a visual similarity to a poker table while capturing images. The rank and suit of playing cards have been detected by using image processing techniques such as thresholding, edge detection, morphological processing and template matching. In the 15 images that were used for testing, the accuracy for identification of suit and rank of playing cards has been identified to be a perfect 100%. Other than this there were some test images which failed due to poor lighting conditions and reflections in the images.

1.4 Agenda

In the section 2, this paper will talk about the related work done in card detection as well as probability calculation techniques employed by other researchers. Section 3 will give a detail explanation of the individual steps followed in implementation of the project. Section 4 and 5 will be used to summarize the project with graphical representation of results and conclusion. Future work for this project has been presented in section 6 which will be followed by Bibliography and Appendix.
Chapter 2

Background

2.1 Image Processing

Image processing is a method by which mathematical operations are performed on an image to get an enhanced image or extract useful information from the image. Most image processing techniques involve conversion of image to a two-dimensional or a three-dimensional signal and apply signal processing techniques to the signal. Image processing has seen an increasing demand in diverse applications such as multimedia computing, biomedical imaging, secured image data communication, biometrics, remote sensing, text recognition, pattern recognition, content-based image retrieval etc.

Image processing has mainly been invented to serve the below purposes:

i) **Visualization:** To observe the objects that are not visible.

ii) **Image Sharpening and Restoration:** To enhance and create a better image.

iii) **Image Retrieval:** To find the image of interest.

iv) **Pattern Measurement:** To measure various objects present in an image.

v) **Image recognition:** To distinguish between different objects in an image.

There are two types of image processing techniques i.e. Analog and Digital Image Processing. Analog Image Processing techniques can be used for hard copies such as printouts and photographs. Digital Image Processing can be applicable in the manipulation of digital images using computers.

In this project, Digital Image Processing techniques have been employed to extract information from the images that have been captured using a digital camera.
2.2 Preview of Poker

A verbatim copy of the rules for Texas Hold’em from [2] is included here for convenience: “There are many variants of a poker game. Some of the most popular ones are Texas Hold’em, Razz, Omaha, Seven-card Stud, Five-card Draw etc. To simplify implementation and probability calculation, Texas Hold’em is chosen as the variant that will be used for this project.

2.2.1 Playing Texas Hold’em Poker

In a Texas Hold’em Poker game, each player is handed two cards which are also known as hole cards. The visibility of these cards is limited to the player. Then the dealer spreads five cards—three at once, then another and then another depending on the round which helps the players to make their best possible five-card hand. In each round the players check, call, fold or raise based on their playing strategies. The below instructions will help one familiarize themselves with the rules of Texas Hold’em poker.

i) Blinds: In Hold’em, a marker called the button or the dealer button indicates which player is the nominal dealer for the current game. Before the game begins, the player immediately clockwise from the button posts the “small blind”, the first forced bet. The player immediately clockwise from the small blind posts the “big blind”, which is typical twice the size of the small blind, but the blinds can vary depending on the stakes and betting structure being played.

ii) Player Betting Options: In Hold’em, as with other forms of poker, the available actions are fold, check, bet, call or raise. Exactly which options are available depends on the action taken by the previous players. If nobody has yet made a bet, then a player may either check (decline to bet, but keep their cards) or bet. If a player has bet, then subsequent players can fold, call or raise. To call is to match the amount the previous player’s bet. To raise is to not only match the previous bet but to also increase it.

iii) Pre-Flop: After seeing his or her hole cards, each player now has the option to play his or her hand by calling or raising the big blind. The action begins to the left of the big
blind, which is considered a live bet on this round. That player has the option to fold, call or raise.

**iv) Flop:** Now, three cards are dealt face-up on the board. This is known as the flop. In Hold’em, the three cards on the flop are community cards, available to all players still in the hand. Betting on the flop begins with the active player immediately clockwise from the button. The betting options are similar to pre-flop, however, if nobody has previously bet, players may opt to check, passing the action to the next active player clockwise.

**v) Turn:** When the betting action is completed for the flop round, the turn is dealt face-up on the board. The turn is the fourth community card in Hold’em. Another round of betting ensues, beginning with the active player immediately clockwise from the button.

**vi) River:** When betting action is completed for the turn round, the river or Fifth Street is dealt face-up on the board. The river is the fifth and final community card in a Hold’em game. Betting again begins with the active player immediately clockwise from the button, and the same betting rules apply as they do for the flop and turn, as explained above.

**vii) The Showdown:** If there is more than one remaining player when the final betting round is complete, the last person to bet or raise shows their cards, unless there was no bet on the final round in which case the player immediately clockwise from the button shows their cards first. The player with the best five-card poker hand wins the pot. In the event of identical hands, the pot will be equally divided between the players with the best hands. Hold’em rules state that all suits are equal.”

### 2.3 Card Detection

Some prior work has been done in extracting information from playing cards using the images of cards. The paper used optical character recognition in order to detect the suit and rank of the playing cards[4]. This method also makes use of hough transform for extraction of cards. It also implements detection of coins by using hough circles. This method
achieved an overall accuracy of 94% for rank detection but fails to do suit detection since optical character recognition can be only used to detect characters. Stern et al Aizik makes use of Artificial Neural Networks in order to detect the rank and suit of playing cards[1]. This method makes use of hough transform and edge detection techniques to extract a rectangular portion of cards. After identification of card corners, the image was flattened by homography and the cards are cropped from image surroundings. The cards were filtered by using color detection to differentiate between red and black symbols and suits can be differentiated using shape detection techniques. In this approach, ANN with 20 layers was used to detect the suit and rank of playing cards with an accuracy of 81%. Another approach for recognizing playing cards which involve character segmentation, affine transformation, edge detection and template matching has been proven as an effective technique in card recognition[5]. This project adopts most of its card recognition approaches from this paper. Rotation and scaling are also considered when designing an algorithm for card recognition. To solve the rotation problem, the templates rotated by 180 degrees is also taken into consideration along with the original templates[8]. To solve the scaling problem, all the cards are scaled to 300 pixels height and the width of the cards is also adjusted correspondingly. To perform character segmentation either the top left portion of cards or the bottom right corner of the cards can be extracted. Later these extracted rectangular portions of cards can be used for template matching by comparing the templates of dictionary cards with the templates of captured cards. This approach has been able to recognize the ranks of cards with an accuracy of 99.79 percent in low noise cases and an accuracy of
81.06 percent for suit recognition.

### 2.4 Probability Calculation

Many systems have been designed to predict the results of card/board games using artificial intelligence[7]. But it is difficult to build such systems for poker since it has to deal with hidden information, deception and risk management. In order to calculate the winning probability of a player, it is important to compute the value of the rank function. There were many implementations provided that calculates the rank function. The most popular implementations are Poker-Eval, Cactus Kev, Paul Senzee, and TwoPlusTwo[3]. Among these methods, TwoPlusTwo rank function proved to be the fastest method available. All the rank function implementations require heavy computing and a large number of calculations. Hence, most of these evaluators store precomputed values and fetches these values from look-up tables in order to increase the speed of the algorithm.

Other than these evaluators there is another algorithm known as Chen method which was developed by a professional poker player William Chen. This algorithm is used to compute the relative value of the hand before unveiling the shared cards. These computed hand rankings and rank function values can be used to predict the winning the probability of a player by iterating the algorithm a large number of times. Teofilo has done a lot of research in calculating odds in a poker game. He tried to compute the odds using Chen’s algorithm and a model that takes hand strength, hand potential with opponent modeling and evaluators to calculate the rank function.

<table>
<thead>
<tr>
<th>Hand Rank Program</th>
<th>Average elapsed time for 1,000 trials in milliseconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cactus Kev</td>
<td>807.13</td>
</tr>
<tr>
<td>Paul Senzee</td>
<td>403.04</td>
</tr>
<tr>
<td>Pokersource</td>
<td>2,520.44</td>
</tr>
<tr>
<td>TwoPlusTwo</td>
<td>91.09</td>
</tr>
</tbody>
</table>

Table 2.1: Table to compare evaluators of rank function
Chapter 3

Implementation

3.1 Platform, Tools, and Technology

3.1.1 Hardware Requirements and Specification

The poker odds calculator application was developed on a 2015 Macbook Pro running OSX El Capitan 10.11.6. The dictionary images and test images of playing cards are captured using Samsung Galaxy S6 Edge camera which is a 16 Megapixel camera with an image resolution of 1440 x 2560 pixels. In order to capture the test images, a webcam that supports USB has been taken from the graphics laboratory at Rochester Institute of Technology has been used. In spite of choosing Macbook for working on this project, the project can be run on Windows devices as well since the format for the Matlab project remains the same.

3.1.2 Software Tools

This project requires MATLAB 2016b which happens to be the latest version of Matlab available, but this is the minimum version required for a user to run this project. That is because this project makes use of some advanced functions which were introduced in MATLAB 2016b. Also, it is required to install Computer Vision toolbox, Image Processing toolbox, Statistics toolbox and a few common toolboxes that are required to work on the image processing applications. In order to work with USB camera to capture the test images, an add-on USB Webcam Support with Matlab has to be installed. A GUI has been built for this application by making use of Matlab’s inbuilt App Designer which enables a user to simply drag and drop GUI items and the code for these items will be automatically
generated for use. Then code can be added into the GUI items to add functionality to them.

3.2 System Architecture

System Architecture in this project involves two parts i.e. Card detection and Card Recognition. The below image shows the process flow for card detection.

![Card Detection Process](image)

**Figure 3.1: Card Detection Process**

The below image shows the algorithm flow followed for playing card recognition. All the individual processes and algorithms used in this project will be explained in the below sections.

![Card Recognition Process](image)

**Figure 3.2: Card Recognition Process**
3.3 Creating Dictionary

One deck of playing cards is required to play a poker game excluding the Joker card. Each deck consists of 52 playing cards. Each of these cards has been captured using Samsung S6 Edge camera and then the rectangular portion of the cards have been extracted using preview application in MacOS. After these cards are saved in a folder, the top left portion of the cards that contains the numbers and suit symbols are saved as dictionary images. The number portion and the suit portion of the cards are extracted and saved as separate images in the dictionary. Each of these images is used in template matching while comparing the test images to classify the card that matches the most with the dictionary cards as that card.

Figure 3.3: Template of number 8

Figure 3.4: Template of suit symbol heart
3.4 Thresholding

Thresholding is the easiest method to perform image segmentation. Image segmentation is the process of partitioning an image into multiple segments and showing them in different colors so that they can be easy to analyze. In this project, thresholding is required to separate the white portion from the darker portion of the image such that the borders of cards can be extracted from its background. Hence, first, the images have to be converted from RGB to grayscale. Then, the gray-scale images have to be segmented into binary images using thresholding. MATLAB provides an inbuilt function to perform thresholding in images. This function makes use of Otsu’s method internally to segment the gray-scale or RGB images into binary format[6]. Otsu’s method is a clustering based thresholding algorithm which reduces the gray-scale images to binary images. Otsu’s method is used to segment images into two clusters i.e. white and black based on a threshold value. The part of the image which has value above the threshold is clustered as black and the part below the threshold is clustered as white.

![Figure 3.5: Binary format of test image after Otsu’s thresholding](image)
3.5 Opening

Opening is a morphological process which is the dilation of erosion of an image by the structuring element. Opening is used to remove small objects from the foreground of an image and place them in the background. Opening is generally used to find shapes in an image. In this project, opening is used to extract the rectangular portion of the playing cards in an image. Dilation is one of the basic techniques in morphology. Dilation adds pixels to the boundaries of objects in an image. Erosion removes pixels on object boundaries.

![Figure 3.6: Result of image opening](image)

3.6 Edge Detection

Edge detection is an image processing technique used to find the boundaries of objects within images. The set of points at which the image brightness changes sharply is identified as edges. Common applications for edge detection include image segmentation and data extraction in areas of image processing, computer vision, and machine vision. The most
common edge detection algorithms are Canny, Prewitt, Sobel, Roberts and fuzzy logic. This project makes use of Sobel Edge Detection algorithm in order to extract the edges in the input images. MATLAB provides an inbuilt function to perform edge detection on an image.

![Figure 3.7: Result of Sobel edge detection](image)

### 3.7 Hough Transform

The Hough transform is a feature extraction technique which is commonly used in Image Processing and Computer Vision applications. Originally hough transform was concerned only with the identification of lines in the image, but later it has been extended to identify arbitrary shapes like circles and ellipses. The main advantage of Hough transform techniques is that it has tolerance for gaps in feature descriptions at boundaries and is relatively unaffected by noise in the image. In this project, hough transform was used to detect the lines in the cards which are arranged in the form of parallel lines. Then, the intersection of
horizontal and vertical lines is found to determine the corners of cards. The below images show the lines in the cards and corner points in the cards.

Figure 3.8: Lines on playing cards using Hough Transform

Figure 3.9: Corner points of cards
3.8 Card identification

3.8.1 Unsuccessful Attempts

3.8.1.1 SIFT

Scale Invariant Feature Transform is a computer vision algorithm that is mainly used to detect and describe local features. For any image, it is enough to store the important points in an image which are given by a feature descriptor in order to identify that object in a different image. In order to perform reliable recognition, it is important that these features have to be recognized under changes in image scale, noise and illumination. These points usually lie on the high contrast regions of the images such as edge points of objects. In this project, card detection has been attempted by using SIFT. There were two main drawbacks observed in using SIFT for card detection: 1) It is very slow. 2) It gives a bad accuracy for suit detection since different suit symbols look the same.

![SIFT matching percentage of card with clubs and diamonds](image)

Figure 3.10: SIFT matching percentage of card with clubs and diamonds

3.8.1.2 SURF

Speeded Up Robust Features is also a computer vision algorithm that is known to be local feature detector and descriptor. SURF is partly inspired by SIFT but SURF is several times faster than SIFT and is also more robust against image transformations. SURF is mainly used for applications that require object recognition, image registration, classification or 3D
reconstruction. SURF algorithm computes an integer approximation of the determinant of Hessian blob detector in order to detect the interest points. In this project SURF was used to detect suit symbols of playing cards. In spite of being faster and more robust against image transformations when compared to SIFT, SURF has produced poor results in suit detection and has proven to be slow.

![Figure 3.11: SURF results for suit detection](image)

### 3.8.1.3 Geometric Features

Each shape has its unique geometric aspect that makes it easier to identify a particular shape. A few such geometric factors include perimeter, area, form factor, aspect ratio, squareness, and circularity. The above factors were calculated for each suit symbol and an attempt was made to identify the patterns in these factors for each suit symbol. Although the results were good for diamonds and clubs, it was very difficult to differentiate between hearts and spades using this method.

![Figure 3.12: Geometric features for suit detection](image)
3.8.1.4 OCR Trainer

MATLAB has an inbuilt function to recognize text. This function is called OCR which stands for Optical Character Recognition. The ocr function provided by Matlab is a generalized function and has a limited number of training samples. Therefore, MATLAB provided a new app called OCR Trainer where we can train our images and label each character in the image to a particular character. OCR Trainer was used in this project in an attempt to detect the rank and suit of playing cards. OCR Trainer was not successful in correctly identifying the characters in the playing cards and since the accuracy was poor, this method was not used in the final implementation.

![OCR Trainer labeling and result for card 10D](image)

Figure 3.13: OCR Trainer labeling and result for card 10D

3.8.2 Implemented Technique

3.8.2.1 Template Matching

Template matching is an image processing technique which places the template image on different parts of the image and tries to find a match. Template matching is usually used in manufacturing as a part of quality control, to navigate a robot or to detect edges in an image. In this project, the templates are the dictionary images that have been extracted from the set of playing cards which consists of images of ranks and suit symbols of cards. These symbols will be placed on the cards identified in the test images and will be checked for matches. In order to be robust against rotation of images, the templates are checked
in normal angle as well as the one shifted in 180 degrees. After trying several methods, Template matching proved to be an effective technique in identifying the rank and suit of the playing cards.

![Figure 3.14: Rank and Suit Detection using Template Matching](image)

### 3.9 Probability Calculation

This project has been done for Texas hold’em poker and only flop round has been accounted for simplicity. So the known information in this game is the player’s cards and three cards on the table. So in order to calculate the winning probability of a player, the remaining cards have to be assigned randomly and also the other player’s cards have to be assigned randomly. Then, these cards are sent to a poker evaluator which evaluates the results based on the cards. This process has to be repeated for 30000 iterations in order to get the probability for which a player wins a particular round of poker. In this project, an evaluator developed by Oliver Rice has been used to evaluate the results of a poker game. The output will be a floating point number less than 1 which gives the winning probability of the player. The closer the probability to 1, the more the chances for the player to win the
game. Based on the probability number, a player can choose to check, call, raise or fold in a poker game.

Figure 3.15: Probability Calculation for a particular set of cards
Chapter 4

Results

4.1 Card Detection

Template matching was used for determining the rank and suit of playing cards. Under good lighting conditions and a plain and dark background, the recognition of cards has been successful. The used method has managed to classify the playing cards correctly while providing resistance to visual clutters and perspective transforms. The suit and rank of playing cards has been recognized with a perfect accuracy of 100% for the 15 test images used in the project.

4.2 Probability Prediction

After the ranks and suits of playing cards has been identified by using template matching, this information is fed to the algorithm for calculating the winning probability of a player. So in order to calculate the winning probability of a player, the remaining cards have to be assigned randomly and also the other player’s cards have to be assigned randomly. Then, these cards are sent to a poker evaluator which evaluates the results based on the cards. This process has to be repeated for 30000 iterations in order to get the probability for which a player wins a particular round of poker. These results are compared to those of few android apps that serve the same purpose and the results were observed to be very close.
4.3 GUI

A GUI has been built on MATLAB using MATLAB App Designer where the GUI items can be built easily using drag and drop option. Then, the functionality for the GUI items can be manually provided by the programmer. The GUI accepts either a camera input to capture the input image or an upload button to select the input image from files. After that, on clicking the get results in button, the results for identified rank and suit of playing cards is displayed and also the winning probability of the player is displayed.

Figure 4.1: MATLAB GUI showing results of card recognition and probability prediction
Chapter 5

Conclusions

5.1 Current Status

Of all the methods that have been tried for playing card recognition, template matching has proven to be an effective method for classifying the cards correctly and with a perfect accuracy of 100%. The suit and ranks of playing cards has been recognized perfectly with the interference of clutter and different perspectives. Also, the recognition of cards took only 3s making it applicable in real-time applications.

5.2 Lessons Learned

- SURF and SIFT cannot be used for card recognition because of its speed limitation and also poor accuracy in recognition.

- Using OCR Trainer was a new experience but this method also yielded poor results in card recognition.

- Initially, this project was intended to be built for the android platform. But since few functions did not support the conversion from Matlab to C, this could not be done. However, to build computer vision applications in Android platform it is better to build an application using OpenCV Android or OpenCV Java.
Chapter 6

Future Work

Future work for this project may include creating this project to work for all rounds of poker and after that maybe create an implementation to make it work for different variants of poker or other games. One another improvement that could be made is to make this algorithm to work for all playing cards using one kind of training images. Also, the whole algorithm could be implemented for Android platform so that it can be used for real-time applications.
Bibliography


