|  |
| --- |
| RIT, Deptartment of Computer Engineering |
| Pocket Billiards Trainer |
| Senior Projects I High-Risk Investigation: Image Processing Component |
| Beau Sattora  Ryan Gault  Collin Reeser  Greg Dias |
| **Beau Sattora** |
| **11/4/2014** |

|  |
| --- |
| The image processing component of this project shall identify the play area of the table, the pockets and rails of the table, and the location of all balls on the table and their colors. |

Contents

[1. Overview 2](#_Toc402843332)

[2. Risk Specification 2](#_Toc402843333)

[I. Risk Overview 2](#_Toc402843334)

[II. Associated Customer Needs/Marketing Requirements 2](#_Toc402843335)

[III. Associated Engineering Requirements 3](#_Toc402843336)

[IV. Justification 3](#_Toc402843337)

[3. Risk Investigation 3](#_Toc402843338)

[I. Table Feature Detection 3](#_Toc402843339)

[II. Ball Detection 4](#_Toc402843340)

[III. Ball Color Identification 4](#_Toc402843341)

[4. Risk Mitigation Design 5](#_Toc402843342)

[I. Overview 5](#_Toc402843343)

[II. Process Flow 5](#_Toc402843344)

[III. Edge Detection 6](#_Toc402843345)

[IV. Hough Transform 6](#_Toc402843346)

[V. Histogram Comparison 8](#_Toc402843347)

[5. Parts List 8](#_Toc402843348)

[6. Testing Strategy 8](#_Toc402843349)

[I. Table Feature Detection 8](#_Toc402843350)

[II. Ball Detection 9](#_Toc402843351)

[III. Ball Color Identification 9](#_Toc402843352)

[7. Uncertainties 9](#_Toc402843353)

[I. Table Feature Detection 9](#_Toc402843354)

[II. Ball Detection 9](#_Toc402843355)

[III. Ball Color Identification 9](#_Toc402843356)

[A. Appendices 10](#_Toc402843357)

[I. References 10](#_Toc402843358)

[II. Diagrams 10](#_Toc402843359)

# **Overview**

The pocket billiards trainer project requires an image processing component capable of analyzing the play area of the pocket billiards table. This component shall identify the play area of the pocket billiards table, which includes the pockets and rails of the table. This component shall also identify the location of all balls on the table and the color of each ball. The game information shall be obtained by an overhead camera and analyzed on an embedded Linux computer using computer vision and image processing techniques.

# **Risk Specification**

## **Risk Overview**

The image processing component of this project has many risks associated with it. The functionality of this component holds significant importance in that it provides vital information to many of the other components of the system. In particular, the game calculation and shot prediction components rely heavily on the information obtained from the image processing component. These risks are outlined below with their associated engineering and marketing requirements.

|  |  |  |
| --- | --- | --- |
| **Risk** | **Marketing Requirements** | **Engineering Requirements** |
| The Image Processing Component shall detect the table’s pockets and rails. | 1, 2, 5, 7, 8, 13 | A, G |
| The Image Processing Component shall detect each ball on the table. | 1, 2, 6, 7, 13 | A, B, G |
| The Image Processing Component shall identify each ball’s color. | 1, 2, 6, 7, 13 | A, D, G |

## **Associated Customer Needs/Marketing Requirements**

* + 1. The system shall not physically interfere with gameplay.
    2. The system shall not be interrupted by normal gameplay.

1. The system shall not physically modify or alter the table.
2. The system shall not permanently modify game tools (balls, rack, cue, etc.).
3. The system shall work with standard pocket billiards equipment set (balls, rack, cue, etc.).
4. The system shall work on a standard size billiards table.
5. The system shall feel responsive in casual gameplay.

## **Associated Engineering Requirements**

* + - * 1. The system shall have a minimum overhead display distance from the table.
        2. The system shall be able to locate all balls on the playing surface.

1. The system shall be able to know each ball’s identity.
2. The system shall be powerful enough to perform all calculations in less than 10 seconds per turn.

## **Justification**

1. *The Image Processing Component shall detect the table’s pockets and rails*.

This risk specification justifies Engineering requirements A and G.

* Requirement A: A minimum distance from the table is required for the camera to acquire a reasonable image which contains the entire play area.
* Requirement G: Depending on the image processing techniques utilized, the detection of the table’s features needs to fit within the defined time constraint.

1. *The Image Processing Component shall detect each ball on the table*.

This risk specification justifies Engineering requirements A, B and G.

* Requirement A: A minimum distance from the table is required for the camera to acquire a reasonable image which contains the entire play area.
* Requirement B: The image processing component is the primary component for detection of the balls on the playing surface.
* Requirement G: Depending on the image processing techniques utilized, the detection of all balls on the table needs to fit within the defined time constraint.

1. *The Image Processing Component shall identify each ball’s color.*

This risk specification justifies Engineering requirements A, D and G.

* Requirement A: A minimum distance from the table is required for the camera to acquire a reasonable image which contains the entire play area.
* Requirement D: The image processing component is the primary component for determining each of the ball’s identities.
* Requirement G: Depending on the image processing techniques utilized, the detection of all balls on the table needs to fit within the defined time constraint.

# **Risk Investigation**

## Table Feature Detection

Detection of the table’s features (rails, pockets) is necessary in order to define the boundaries of the play area for accurate game calculations. This information is used by the game calculation component to locate goal regions (pockets) and barriers (rails) for calculation of ball physics on the play surface.

1. Concepts Considered/Chosen
2. Hough Transform
3. Rationale

The Hough transform was chosen for identifying the table’s features. This method was chosen because the Hough transform is able to locate all straight lines and circles in an image. This process may need to be separated into two separate transforms, one for straight lines and another for circles. This will allow for simple detection of the table’s borders and pockets. No other techniques were considered because the Hough transform is extremely robust, easy to compute, and has an implementation in OpenCV.

Examples of the Hough transform being used to detect lines and circles in images can be seen in Figures A1 [2] and A2 [3] of [Appendix A.II](#_Diagrams), respectively.

## Ball Detection

Detection of the size and location of all balls on the table is necessary in order for the game calculation component to calculate meaningful shot predictions for the player.

1. Concepts Considered/Chosen
2. Hough Transform
3. Rationale

The Hough transform was chosen for locating the balls on the table. This method was chosen because the Hough transform is able to locate all circular objects in an image. The Hough transform can also be limited to search a specified parameter space. This could be useful in minimizing computation time by limiting the parameter space of the Hough transform to circles that only fit the radius of pool balls from the calibration phase. No other techniques were considered because the Hough transform is extremely robust, easy to compute, and has an implementation in OpenCV.

An example of the Hough transform being used to detect circles in an image can be seen in Figure A2 [3] of [Appendix A.II](#_Diagrams).

## Ball Color Identification

Detection of each ball’s color is necessary in order for the game calculation component to track different modes of play and to calculate shots for both striped and solid balls.

1. Concepts Considered/Chosen
2. Histogram Comparison
3. Color Thresholding
4. Rationale

Comparison of histograms was chosen because a histogram of the ball from the calibration phase can be used as a unique feature to describe each ball. This makes determining the ball’s identity a much simpler task by comparing the histogram of the current ball in question to a lookup table of expected histograms, where the minimum absolute error between histograms indicates the closest match. There are many histogram comparison techniques which can be performed and histogram comparison has multiple implementations in OpenCV.

The concept of color thresholding the image to find each ball would be a tedious approach and could require a large amount of processing time with no guarantee of finding the intended ball.

An example of a sample histogram from two pool balls can be seen in Figure A3 [1] of [Appendix A.II](#_Diagrams).

# **Risk Mitigation Design**

## Overview

The image processing component will attempt to mitigate risk by following a simple process flow to solve each of the addressed problems. In the case that there is a failure, this should make it simple to determine where the component is failing.

## Process Flow

Read Image

Canny Edge Detection

Hough Transform (Circles)

Hough Transform (Lines)

Locate Ball Circles

Locate Pocket Circles

Locate Rectangles

Identify Rails

Identify Pockets

Identify Balls (Histogram)

Send Object and Coordinate Data

* The image is read from the camera for processing.
* Canny edge detection is performed to obtain an edge image.
* Hough transforms are applied to the edge image to obtain a Hough line image and a Hough circle image.
* Rectangles are located in the Hough line image.
* Pocket circles are located in the Hough circle image.
* Ball circles are located in the Hough circle image.
* Rails are identified in the Hough line image.
* Pockets are identified in the Hough circle image.
* Balls are identified using comparison of histograms.
* Location data for all objects is sent for game calculations.

## Edge Detection

Canny edge detection is used in obtaining an edge image from the input image from the camera. This algorithm operates on the following four-step process [4]:

1. Blur the image with Gaussian smoothing to remove noise from the image.
2. Apply a gradient operator to obtain the gradients’ intensity and direction.
3. Determine if a pixel is a candidate for an edge by use of non-maximum suppression.
4. Find the beginning and end of edges using hysteresis thresholding.

An example Canny edge image of a pool table is shown below in Figure 1.

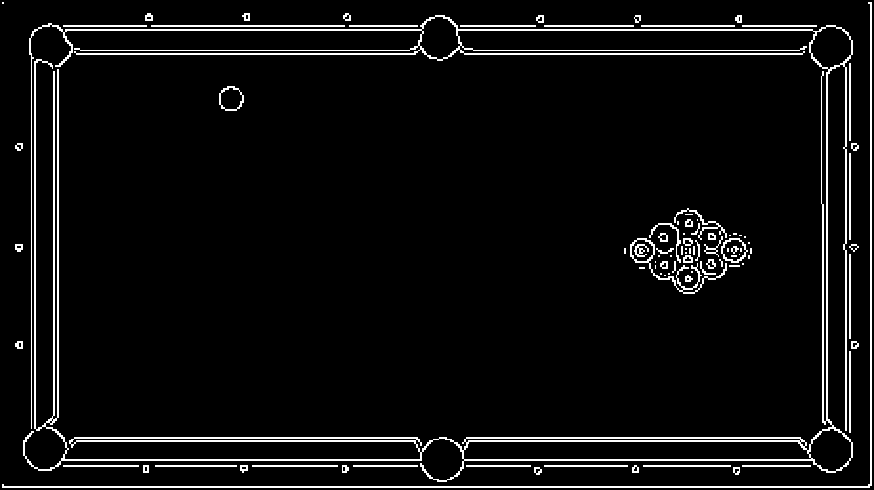


Figure 1: Canny Edge Image

## Hough Transform

The Hough transform is used in locating straight lines and circles in the edge image.

1. Hough Lines

Locating lines in the image is performed by using the Hough transform to solve for all lines in a polar coordinate system where . Candidate lines are voted for by assessing the number matching lines through a set of points.

An example image of the Hough transform used to detect lines in a pool table image is shown below in Figure 2.



Figure 2: Hough Line Image

1. Hough Circles

Locating circles in the image is performed by using the Hough transform to solve for all circles such that and, solving for *a*, *b* and *R*. Similar to the line detection, candidate circles of a specified radius are voted for by assessing the number of matching circle edge pixels a distance *R* away from the center pixel.

An example image of the Hough transform used to detect pool balls in an image is shown below in Figure 3 [1].

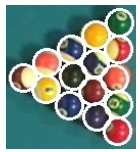


Figure 3: Hough Circle Image [1]

## Histogram Comparison

A comparison of histograms is used to identify the color of each ball in the image. This method will use a table of reference histograms to define the ideal distribution for each ball based on the calibration phase. Once a ball has been located from the Hough transform, the color data for that ball’s region will be analyzed and compared to the histogram of each ball in the table. The comparison which yields the highest correlation will be the candidate for that ball’s identity. An example of the histogram data from balls in Hue Saturation Brightness (HSB) space is shown in Figure 4 [1] below.

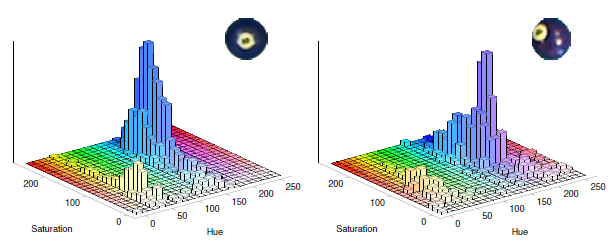


Figure 4: HSB Histogram Data for #2 and #4 Balls [1]

# **Parts List**

There are no parts associated with this component. The image processing component is implemented in software and requires information from the physical components of the system.

# **Testing Strategy**

The overall testing strategy for the image processing component is similar for each individual risk. The goal is to develop a base algorithm using known computer vision and image processing techniques, apply the algorithm to a large set of images for analysis under various conditions, then integrate the working code with the remaining components for use during live gameplay.

## Table Feature Detection

* Develop feature detection algorithm(s) in Matlab.
* Test algorithm(s) on a large set of test images.
* Port working Matlab code to C/C++ using OpenCV.
* Test C/C++ port on test images and compare results.
* Integrate with related components.
* Test live on a table during gameplay.

## Ball Detection

* Develop ball detection algorithm(s) in Matlab.
* Test algorithm(s) on a large set of test images.
* Port working Matlab code to C/C++ using OpenCV.
* Test C/C++ port on test images and compare results.
* Integrate with related components.
* Test live on a table during gameplay.

## Ball Color Identification

* Develop color identification algorithm(s) in Matlab.
* Test algorithm(s) on a large set of test images.
* Port working Matlab code to C/C++ using OpenCV.
* Test C/C++ port on test images and compare results.
* Integrate with related components.
* Test live on a table during gameplay.

# **Uncertainties**

## Table Feature Detection

There is a minimal amount of uncertainty in this area, however it is uncertain how the system will handle interference from outside objects entering the frame such as a human or pool cue.

## Ball Detection

The detection of balls on the table has some uncertainty when balls are clustered together or when lighting situations cause the ball to lose edge information and blend in with the background.

## Ball Color Identification

Identification of each ball based on its color holds the highest amount of uncertainty in this component due to the variation of ball orientation, the color variation due to lighting, and the color variation due to striped/solid balls.

# **Appendices**

## References

[1] Baekdahl, J. & Have, S. (2011). Detection and Identification of Pool Balls using Computer Vision. Aalborg University, Denmark.

[2] <http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html>

[3] <http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/hough_circle/hough_circle.html>

[4] <http://en.wikipedia.org/wiki/Canny_edge_detector>

## Diagrams

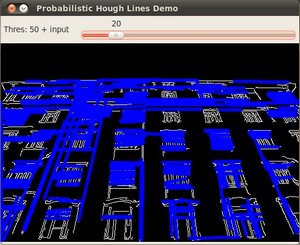


Figure A1: Hough Lines Example Image [2]



Figure A2: Hough Circles Example Image [3]

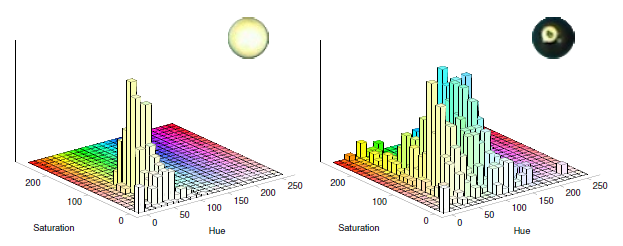


Figure A3: Sample Histogram of White & Black Balls in HSB Color Space [1]