Blackjack Assistant

Robust Card Rank Detection Using Scene-Informed Template Matching

Arbi Tamrazian
Department of Electrical Engineering
Stanford University
Stanford, United States
atamrazian@gmail.com

Kevin Phuong
Department of Electrical Engineering
Stanford University
Stanford, United States
kphuong@stanford.edu

***Abstract*—This paper describes and demonstrates an algorithm for automated recognition and organization of the card ranks of player hands in a blackjack game, as arranged in a typical casino layout. For each player hand in the scene, the detected card values are compared to a "basic strategy" table to determine the optimal game decision. Using an Android application implementation, the original scene image is overlaid with the decisions in order to provide immediate help to new players. The algorithm uses the Hough Transform on a morphological edge-detected image mask to extract, rotate, and provide scaling information of the card ranks in the image. Template matching with our particular deck's rank library is then used to determine rank values. Our algorithm's pipeline aims to provide robustness for scenes featuring various rotations and occlusions of cards, and reduce the complexity and computational cost of template matching in these situations.**

***Index Terms*—blackjack, basic strategy, playing card detection, morphological image processing, edge detection, hough transform, template matching**

I. Introduction

Blackjack, also known as Twenty-one, is a popular casino card game played around the world. The premise is simple: the dealer deals two face-up cards to each player at the table and two cards for himself, only one of which is face-up. Based on this card "shown" by the dealer, players must decide whether they wish to "hit" or "stand" on their hands. "Hit" means receive an additional card; "stand" means receive no more cards. The goal is to reach a total hand value which is greater than the dealer's without exceeding 21, which results in an automatic loss. After the players' turns, the dealer reveals his face-down card and must hit until the sum of his cards is 17 or higher, after which he must stop. If the dealer's total is greater than or equal to a player's, the player loses; if the dealer's total is less than the player's or exceeds 21, the player wins. (Other common decisions include "double", in which players can double their bets to receive a single extra card, and "split", in which players whose two cards form a pair can double their bets and turn each into a separate hand, receiving two more cards to complete each hand.)

While the house always maintains a slightly higher probability of winning than players, it is possible to maximize your chances as a player by employing what is known as "basic strategy". Basic strategy is a table of the optimal decisions to make given what the dealer shows and the player's hand. More advanced players have basic strategy committed to memory or play it by rote.

In this paper, we describe and demonstrate an algorithm, implemented as an Android smart-phone application, which captures an image of a blackjack table card scene and returns to the user an overlay of the optimal decisions for each player hand. The typical casino scene as observed by a player is an overhead view of the table with the dealer cards at the top of the scene and the player hands in a line or arc at the bottom of the scene; additionally, the card ranks on the top left corner of the card typically have prioritized visibility. As such, we constrain our algorithm to accept input images laid out in the same manner. Our algorithm accurately organizes and recognizes the card ranks using template matching. It is furthermore robust against some of the complexities which may be present in a scene, such as various rotations and amounts of overlap of cards as they are dealt to the players.

II. Prior and Related Work

Recognition of playing cards has previously been attempted in different ways, but generally on more simplistic scenes. Hollinger and Ward [3] extracted entire cards from a scene by thresholding, segmenting, and region-labeling an image. Then, central moments and angles were created for each region in order to determine the corners of a bounding box which encompassed each card. The cards could then be rotated back into vertical alignment. To recognize the ranks of cards, the algorithm discriminated between face cards (Jack, Queen, King) from number cards (two, three, etc.) by determining the level of non-background pixels in a thresholded image of a card (which result from texture details present in face cards). Face cards were detected using trained neural networks, and number cards were detected by counting the number of regions in the image (five hearts are present in a five-of-hearts). This algorithm may work well for single cards, or games in which cards are separated spatially (e.g. Texas hold'em poker), but a typical blackjack table will contain partially-occluded player cards. Hollinger and Ward's algorithm will not work in such cases, as face cards may not have enough of a surface for neural network matching and number cards may have one or more of the suit images obscured, preventing any counting scheme.

References [4] and [5] describe different algorithms utilizing template matching methods of rank recognition. Both use image thresholding followed by edge detection to find contours of the image. A rectangular region-of-interest is then formed around the card, either by fitting rectangles into contours [4], or by detecting corner vertices which form a rectangle [5]. In either case, the corner vertices are used for determining a matrix transformation which rotates the cards back into vertical alignment. In addition to an outer contour for each card's edge, an inner contour is also found which encircles card ranks, suits, and other images present on the card in [4]. To recognize ranks, smaller and smaller rectangles are captured until only a single contour region is confined. Template matching is then done by resizing the image of this rank region to the size of stored templates. These two algorithms also require cards to be nonoverlapping in order for all four rectangle vertices to be accurately detected prior to rotation and template matching.

In order for our algorithm to work on a typical casino blackjack scene, which includes overlap in addition to rotation, we therefore explore recognizing cards without using rectangular regions derived from central moments or contours. In the following sections, we describe a method which instead uses the Hough Transform (following some preprocessing) in order to extract pertinent details of the scene that can be used for rotation and scaling information regardless of overlap. This then greatly simplifies the work required in template matching.

III. Description of the Algorithm

A. Preprocessing and Morphological Edge Detection

Some preprocessing on a card scene image is required before clear card edges can be found for the Hough Transformation. We first convert the scene image from RGB to grayscale and then binarize the image using Otsu's method to determine a threshold. In our experiments, binarization of a grayscale image has been more than adequate for segmenting cards from typical backgrounds. The white regions of a card are generally bright enough to be above the threshold for card surface segmentation. (In a standardized casino scene with a specific color for the blackjack table and the clutter of colored chips, grayscale conversion and binarization can be replaced with HSV conversion to extract white surfaces out of green, red, etc., surfaces and clutter.) Since suit images or textures from the face cards may appear as background within the segmented cards, we perform hole-filling to remove the background pixels from locations within the foreground regions. This leaves us with a mask of a card scene, where the card regions are entirely in the foreground, as shown in Fig. 1b.

Using this card mask, we can obtain an edge mask using morphological image processing in lieu of other edge detection algorithms, which are more computationally intensive and unnecessary for the low level of complexity present in the card mask. We first perform binary image dilation on the card mask with a structuring element in order to enlarge the mask by a small amount. A disk with a two pixel radius worked well as a structuring element. We then subtract the original card mask from the larger dilated card mask in order to obtain a two pixel outline of the edges of card hands in the image (Fig. 1c). A smaller structuring element for dilation results in a one pixel edge, which was often insufficient for forming complete Hough lines for angled cards. A larger structuring element would result in a thicker edge, which multiple Hough lines could fit into, resulting in several redundant lines for each edge.

As a final step in preprocessing, we separate the preprocessing stage images into dealer and player hands based on vertical positioning (given that the typical scene from a player perspective features the dealer at the top). This enables us manipulate the later stages of our pipeline separately and organize the collected information. To obtain the number of player hands present, we perform region labeling and region counting on the player section of the card mask of Fig. 1b. We anticipate that in some cases a player may not have overlapping cards. We therefore first dilate the player binary card mask with a small disk to combine card regions which still correspond to the same hand. The number of player hands will be important when we later cluster card ranks together, since for any given turn a player may have two to maybe five or six cards.

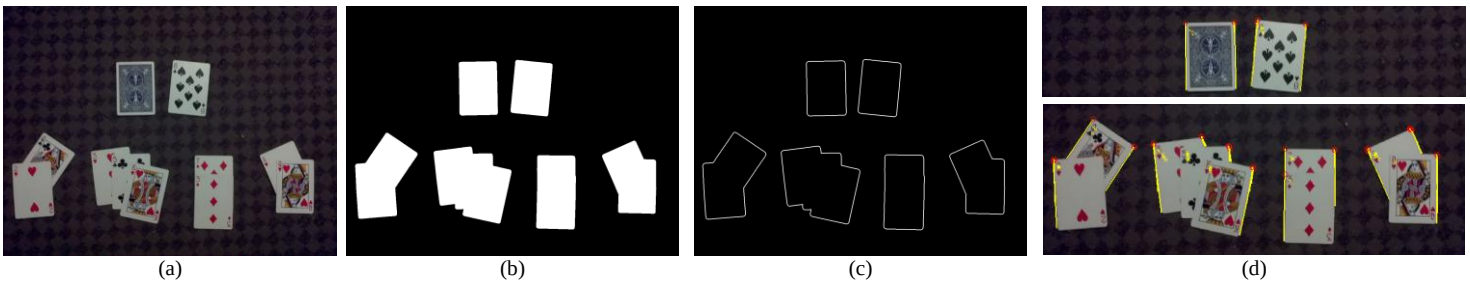
B. Hough Transformation

The Hough transform uses an alternative parameterization of edge pixel locations in an image. Pixel locations in x and y are instead mapped to

$$x\cos\theta + y\sin\theta = \rho, \tag{1}$$

where ρ is the length of the line from the origin to lines that can be formed from a pixel, and θ is the angle of the line formed. When the pixels of an image form a line, a strong peak is observed in the Hough transformation for the length ρ corresponding θ of the line which forms a 90 degree angle with the image line. We apply the Hough transformation to our dealer and player edge masks and detect peaks of the transformation, which correspond to the lines in the card edges. We first apply the transformation to the dealer edges. This gives us the lines corresponding to the horizontal and vertical edges of dealer cards. We take the maximum lengths of lines between -85° and 85° and between -5° and 5° relative to vertical axis. These give us the vertical and horizontal lengths of a "reference" card in the image scene, which we use for scaling during template matching. For a given deck, we know the dimensions of the cards, as well as the size and location of the rank.

We then limit the range of valid Hough lines to those between -50° and 50°, which typically correspond to vertical edge lines, even when cards are rotated. For each vertical line present in the image, we determine whether the top endpoint is a corner of the card by checking whether a location above the card in the binary card mask returns true. We also check if it is a left or right corner by checking against the binary mask for a card in both directions. Top corners are relevant because they provide the most accurate information about rank location; due to overlaps, vertical edges in the image may not always lead to useful information about the card. These checks are performed after a rotation corresponding to the angle of the Hough line,



(a) (b) (c) (d)

Figure 1. The major steps that lead to rank location markers in the algorithm. (a) The input image. (b) The binarized and hole-filled card mask. (c) The results of morphological edge detection. A small structuring element (disk of two pixel radius) was used to dilate the card mask of (b). The original mask (b) was then subtracted from the dilated mask to create (c). (d) The results of rank location marking. Vertical card edges whose endpoints correspond to top corners of cards are shown in yellow. The endpoints are labeled with red circles. The rank locations determined using the endpoints and the Hough line angles for rotation are shown as yellow crosses. The image shows how multiple yellow crosses may be localized around each rank due to redundancies that must be removed.

which can be done using a standard counterclockwise rotation matrix with the endpoint as a vertex. Based on whether a line is determined to be the left or right edge, it is known from the reference card dimensions (and deck card information) where a rank location can be found relative to the top endpoint of the line. We also rotate this location based on the Hough line angle to compensate for the card's orientation. This process occurs for each Hough line so that markers are placed at each rank location for both dealer and player hands.

Once we have the rank location markers, we extract a small region around them to obtain rank images. These images are rotated using bicubic interpolation, again using the angles of the Hough lines from which the rank markers were determined, to return card rank images to vertical orientations for template matching. Card edges may contain multiple Hough lines, and a card whose left and right edges are both visible would have a very similar rank location marker contribution from each, so in general the rank location markers may be redundant. To save on computation time, these are removed prior to rank image extraction and rotation.

C. K-means Hand Clustering

The raw rank location markers need to be clustered into corresponding hands. We remove redundant marker locations by calculating the Euclidian distance matrix and eliminating points that do not satisfy an experimentally determined distance threshold. We then cluster these unique points into hands using k-means as our clustering method. We note that other methods of clustering (e.g. hierarchical clustering) can also achieve accurate results. The k-parameter is given by the number of player hands determined using region counting.

D. Template Matching

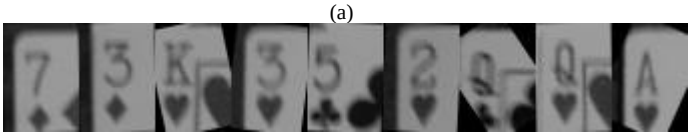
Values were assigned to the extracted rank images using template matching. Templates were obtained by photographing thirteen cards (one for each rank) with the Android smart-phone used in algorithm implementation. We recorded the heights of each template card to allow for proper scaling of the templates with respect to the "reference" card height in a query image. (Extraction of the query image height was described in section *C*.) Scaling of templates is critical in that it allows the algorithm to be scale invariant. We crop out the ranks from the thirteen images to form a 50x30 template using photo-editing software. These images are then stored in the server.

After the templates are created we begin the template matching procedure by preprocessing the query images. After rank extraction from the overall image we were left with a blurry, low contrast image. This is due to the limitations in resolution/focusing ability of the android device. To combat this we first perform localized histogram equalization on each extracted rank image to improve contrast. Localized histogram equalization is preferred to global histogram equalization since it allows for non-uniform lighting conditions. After histogram equalization we binarize both the query and template image using a threshold determined by Otsu's method for each.

After the query images have been preprocessed, we being our template matching procedure by first minimizing the following mean-squared error,

$$E[p,q] = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} (s[x,y] - t[x-p, y-q])^2 \qquad (2)$$

(a)



(b)

(c)

Figure 2. Extracted and aligned rank images. (a) The dealer rank location images. (b) The player rank location images. (c) An example of a rank template for a particular deck. After redundant rank location markers are removed, a small region around each marker (size determined by "reference" card dimensions) is extracted. An image rotation with bicubic interpolation is performed based on the Hough line angle associated with the rank marker to return the rank image to its proper alignment.

where s[x,y] is the image matrix and t[x,y] is the template image. Effectively we wish to maximize the correlation between s[x,y] and t[x,y]. This operation is equivalent to maximizing the convolution between s[x,y] and t[-x,-y]. We remove the mean from all templates and query images to avoid bias toward bright image areas. We loop through each template image and convolve the flipped template with the query image. The value corresponding to the maximum of all thirteen correlation results is recorded as a possible candidate. A value will be assigned to a query image if an experimentally determined SNR threshold is met. An SNR constraint is needed only for the player hands since a Hough line misdetection may occur, causing a faulty image extraction from a location without a rank. We improve robustness of dealer card detection by assigning the value corresponding to the maximum correlation results of all query images. The dealer will always have query images that fall into two categories: face down rank extraction/ faulty rank extraction, which will have a low correlation result; one face up rank extraction which will have a high correlation result. We take the high correlation result as the value for the dealer hand.

E. Basic Strategy Look-up

Once each hand (including the dealer) has been paired with its card value(s) as determined by template matching, we compare them to our coded version of the basic strategy table [2]. This returns an optimal decision for each hand.

F. Practical Implementation

We implemented the algorithm in MATLAB (Mathworks, R2011b) on a Xeon dual-core server. We sent query images to the server for processing using a 5MP images from an Android device (Motorola ATRIX 4G, 2011). The processed image was returned to the Android device for viewing. The processed image includes the original card scene with text indicating the numerical value of each rank in a hand and its optimal decision displayed over the respective hand, as shown in Fig. 3.

IV. RESULTS

We used our algorithm on several sample images to test rotational invariance, and scale invariance. Given the constraints in the scenes meant to be encountered by our algorithm, we similarly constrained each experiment in the following way: 1) One top corner of a card must be exposed; 2) the top left rank must not be occluded; 3) the image must be taken directly above the cards; 4) individual cards may not be rotated more than 50 degrees relative to the vertical axis. Each card is distributed with an orientation independent of the other cards and each rank therefore corresponds to a single sample point of rotation in our experiments. The experiments presented below will contain four hands (including dealer) with three cards each. This will correspond to a sample size of thirteen (N = 13) for each experiment.

A. Rotational Invariance

For each experiment we distribute four hands (three cards in each hand) and a dealer card at random relative (to each other) rotations. We capture an image for processing using the Android device and record the number of correct/ incorrect template matching values as well as the number of Hough line misdetections. A Hough line misdetection is classified as a fault in determining the rank location marker (ie. a "rank" detected in the middle of a card. This will result in a faulty rank extraction causing a bad/null result to be reported as the value. We conduct ten experiments giving us a large number of trials (N = 130) to test rotational invariance. The results are summarized below (Table 1). These data show that the algorithm is extremely robust to changes in relative rotations (92.3% total correct detection).

B. Scale Invariance

An image is captured at various distances above the distributed cards using a single card distribution (four hands with three cards per hand). We record the number incorrect/ correct template matching values as well as Hough line misdetections. We conduct four experiments (N = 52) at each distance above the distributed cards and report the results in Fig. 4. For distance between 2.5 and 3.5 the mean percent correct detection was 98.1%. Hence, we find that the algorithm is robust against scaling up to distances of 3.5 feet. We see a dramatic decrease of reliability after 3.5 with a minimum value of 0.77% at 5 feet above the card values. We believe that the decrease in repeatability was mainly due to the poor resolution/lens quality found on the android device.

V. CONCLUSION

In this paper we present an effective method to detect, identify and characterize card rank values to advise users on basic black jack strategy. Unlike previous works [4,5,6], our implementation allows for card overlap and implements an efficient algorithm by first extracting out card ranks using Hough transforms. This allows for quick template matching due to the drastic reduction in the size of the image matrix. Furthermore, we have shown that our algorithm is robust against rotations and scaling. We obtain 92.7% total correct detection rate with rotations ranging from -50 to 50 degrees relative to the vertical axis. An average total correct detection rate of 98.1% is obtained from distances of 2 to 3.5 feet above the distributed cards.

Future works would involve exploring adaptations to our algorithm pipeline in order to account for perspective.

TABLE I. ROTATIONAL INVARIENCE

Condition	***Number of ranks***	***% of ranks***
Correct Template Matching Value[a]	120/124	98.7
Hough Line Misdetection	6/130	—
Total Correct Detection[b]	120/130	92.3

a Compares ranks resulting from correct Hough line detection.
b Includes Hough line misdetections.

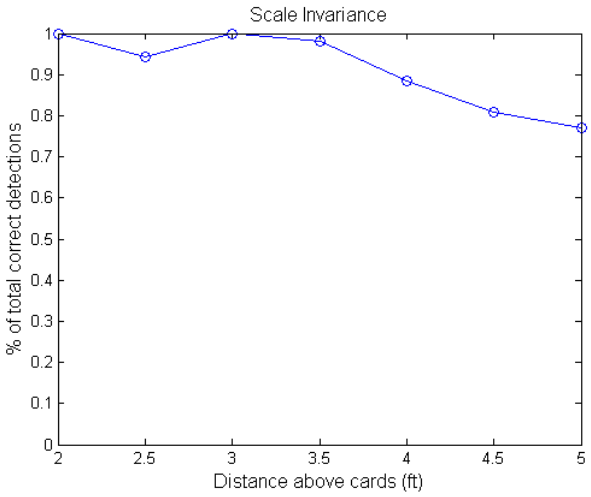


Figure 4. A plot of the percentage of total correct detections of card ranks as a function of distance that the Android smart-phone pictures were taken from the scene. The algorithm demonstrates strong scale invariance until resolution and blurring begin to affect template matching.

Perspective invariance would allow for scenes to be captured from a more natural player perspective, rather than from an overhead view. The Android application can also be used to store the values of card ranks recorded in previous turns. This would enable the algorithm to factor card counting into its decision making, which changes the optimal strategy based on the probability of the dealer being able to have certain cards as its face-down card. With the combination of card counting and basic strategy, a user would stand a much better chance of winning in blackjack.

ACKNOWLEDGMENT

The authors thank Professor Bernd Girod and course teaching assistant Derek Pang for their instruction during EE368. The authors especially thank course teaching assistant David Chen for his helpful ideas for the implementation of our algorithm and technical assistance on the Android application implementation.

APPENDIX

Both authors proposed the project. K. Phuong implemented the Preprocessing and Hough Transform sections of the algorithm. A. Tamrazian implemented the K-means Clustering and Template Matching sections. Both authors implemented the Android application. Both authors collected experimental results. A. Tamrazian prepared experimental data for figures and table. K. Phuong prepared images for figures. Both authors contributed to the written report.

REFERENCES

[1] K. Smith. 2009. *Blackjack info*. [On-line]. Available: www.blackjackinfo.com/blackjack-rules.php. [June 6, 2012].

[2] *Basic Strategy: 4-8 decks, dealer stands soft 17*. [On-line]. Available: http://www.blackjackinstitute.com/store/Basic_Strategy_Chart.php [June 6, 2012].

[3] G. Hollinger and N. Ward, "Introducing computers to blackjack: Implementation of a card recognition system using computer vision", unpublished.

[4] C. Zheng, R. Green, 'Playing card recognition using rotational invariant template matching', Proceedings of Image and Vision Computing New Zealand 2007, pp. 276–281, Hamilton, New Zealand, December 2007.

[5] P. Martins, L. Reis, L. Teófilo: Poker vision: playing cards and chips identification based on image processing. IbPRIA'11 Proceedings of the 5th Iberian conference on Pattern recognition and image analysis, pp. 436–443, Springer-Verlag Berlin, Heidelberg, 2011.