Determining the state of texas hold ’em in almost to real time

David Molin†, Tomas Forsyth Rosin ††

Abstract—This paper will describe a method which finds and determines suit and rank of playing cards in an image. The image processing pipeline developed in this paper is developed in such a way that it should be possible to make it run on an mobile device in close to real time. The resulting pipeline managed to find and correctly classify images with an accuracy of 66%.

I. INTRODUCTION

The problem of automatically detecting suit and rank of playing cards based on a stream of images could potentially be used for several commercial or non-commercial applications. One such application could be to automatically determine the cards on the table for broadcasting live poker tournaments. Generally when attempting to match objects with a known template in images it is possible to use keypoint detectors such as SIFT [1] or ORB [2] and match these keypoints and do a geometric consistency check through the use of RANSAC. However, due to the fact that the keypoints on poker cards would correspond to the corners of the suit symbols, the ratio test described in [1] would reject most of the matches as there are multiple of each suit symbol on each card.

II. PRIOR WORK

Some prior work in this field has been done. For example, [3] used optical character recognition in order to determine the suit and rank of playing cards. In 2007 [7] used template matching of the symbols in the corners of the cards to determine the rank of playing cards. In the field of image processing there has been several papers published which describes methods which are possible to use for card recognition. In 1995 Kiryati et. al. proposed a method to compute an approximate hough transform by only consider a subset of the points in the image[9]. In 2000 Matas et. al. made improvements to the probabilistic hough transform[8]. Keypoint and feature detection such as SIFT [1] and [2] has also been developed the last decade and these methods can be used to find instances of objects in an image. In 1979, Nobuyuki Otsu developed a method to find a threshold which could separate bright portions of an image from dark portions[5].

III. METHOD

The process used in this paper for extracting the suit and rank of all cards in an image can be divided into two parts. The first part is extracting position of the corners of all cards and then from this position extract an image of the card oriented in an upright position with a known size. The other part of the algorithm is to find the suit and rank from an image of a card placed in an upright position.

A. Extracting the card corners

First the assumption that the cards are considerably brighter than the background is made. This is usually true since poker cards are pieces of white paper and the table which poker is played on is usually covered with a green tablecloth. This motivates why it should be possible to extract the poker cards from the background by using Otsu’s method.

Once Otsu’s method has extracted a mask of the cards it is possible to extract the contours of the cards. In the field of image processing there has been several papers published which describes methods which are possible to use for card recognition. In 1995 Kiryati et. al. proposed a method to compute an approximate hough transform by only consider a subset of the points in the image[9]. In 2000 Matas et. al. made improvements to the probabilistic hough transform[8]. Keypoint and feature detection such as SIFT [1] and [2] has also been developed the last decade and these methods can be used to find instances of objects in an image. In 1979, Nobuyuki Otsu developed a method to find a threshold which could separate bright portions of an image from dark portions[5].
CARD RECOGNITION

lines. These points will usually correspond to the corners of the cards, some false positives might result from this but these can then be rejected at a later stage in the pipeline.

Once these corners are found it is trivial to find and apply an projective transformation which maps the cards to a known shape.

B. Finding rank and suit of a card

1) Determining suit: The color of the card can be extracted by comparing the red channel with some other color channel. For red cards such as diamonds and hearts, the red channel will assume large values at all points on the card, this will not however be the case for black cards such as spades or clubs. This fact can then be used to determine if the suit of a card is red or black.

Since all cards have the suit symbol in the upper left corner it is possible to extract the suit by doing template matching in this region.

2) determining if a card is a face or nonface card: Furthermore it is possible to notice that the appearance of the face cards and nonface cards are quite different, the face cards consist of a large illustration suggesting that keypoint matching such as ORB or SIFT could be used to match these cards by comparing the extracted keypoints to the query card. This suggests that it should be a good idea to compute the rank for face cards and nonface cards in different ways.

In order to determine whether a card is a face card or a nonface card the fact that face cards consist of one large connected blob, while nonface cards consist of several small blobs is used. Blobs are found by using Otsu’s method. Then the size of the largest blob in the image is found. If this size is larger than some predetermined threshold the image is classified as a face card.

3) classifying face cards: When doing keypoint matching for the face cards, some issues might arise for the ratio test in \( \text{ Figure 1 } \) since the bottom half of a face cards is a mirrored copy of the top half and therefore each feature would have a corresponding feature from the mirrored part of the card. This should however not be too much of an issue even if ignored and can be solved by instead of using the third closest feature for the ratio test instead of the second closest feature.

4) classifying nonface cards: For nonface cards keypoints matching would be unsuitable due to the fact that keypoints would mainly be found for the suit symbols and these occur several times on each card and therefore the mathes would not be good. Instead it is possible to count the number of blobs on each nonface card. Each blob will correspond to one suit symbol. In order to suppress noise, the blobs would have to be larger than a given threshold, otherwise the small symbols in the corner of the cards as well as small blobs created by noise would contribute to the rank of the card. The blobs are found by applying Otsu’s method on the grayscale versions of the cards.

IV. Results

A. Test set

The method for determining the suit and rank of the cards described in the previous section, is applied to a test set of 20 images, each containing 3 cards taken in an indoor environment with conditions similar to the ones for a real pokertable. In other words, clutter will be present and the images will not be close ups of poker cards taken straight from above. Some photographs will include motion blur as this will generally be present when a mobile device is used for determining cards in real time.

<table>
<thead>
<tr>
<th>Real class</th>
<th>Card</th>
<th>Non card</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detected class</td>
<td>Card</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>NonCard</td>
<td>6</td>
</tr>
</tbody>
</table>

TABLE I

The confusion matrix shows that the results for detecting the cards is good. Using the method described earlier on the test set the accuracy for suit detection is 83% for all of the detected cards. The main problem in suit detection seems to be that clubs are detected as spaces as this happened in 7 out of 9 cases where the suit was incorrectly detected. The correct rank is detected for 92.5% of the detected cards. The detected rank was usually off by one to the correct rank of the card. The accuracy of the whole pipeline is 66%.

B. Undetected cards

In the test set, 10% of the cards were not detected at all. One example of that is shown in figure 5 where the 4 of Clubs is not detected. If the card is not outlined properly, the window used for counting the number of blobs in the center of the card might be too small to detect large enough blobs.
On the 4 of Clubs, all blobs are spread out, which might lead to no blobs counted at all and results in a non-card output.

Another example is shown in figure 6. To avoid false positives from for example the backside of a card, there is a threshold on the minimum number of matching keypoints required to be detected as a face card at all. In this case, the Queen of Diamonds reached too few matching keypoints, and is instead recognized as a non-card.

<table>
<thead>
<tr>
<th>7 of Spades</th>
<th>9 of Spades</th>
<th>3 of Diamonds</th>
</tr>
</thead>
</table>

Fig. 7. Blurry Clubs card, detected as a Spade.

<table>
<thead>
<tr>
<th>7 of Diamonds</th>
<th>J of Spades</th>
<th>2 of Spades</th>
</tr>
</thead>
</table>

Fig. 8. Sharp Clubs card, detected as a Spade.

Another incorrect suit classification is shown in figure 9. The 5 of Hearts, is detected as the 5 of Diamonds. In this case, the blur has made the corner suit symbols warped, which leads to the error. However, the two other cards are even more blurred, but are still correctly classified.

<table>
<thead>
<tr>
<th>6 of Spades</th>
<th>Q of Hearts</th>
</tr>
</thead>
</table>

Fig. 6. An example of an undetected card due to the narrow search window.

Another incorrect suit classification is shown in figure 9. The 5 of Hearts, is detected as the 5 of Diamonds. In this case, the blur has made the corner suit symbols warped, which leads to the error. However, the two other cards are even more blurred, but are still correctly classified.

C. Wrong suit

The similarity between the shape of Clubs and Spades are a great cause of incorrect suit classifications. In figure 7 the 7 of Clubs is detected as the 7 of Spade. It might seem that this is because of the blur, making the classification harder, but this is not always the case. In figure 8 the card is captured sharply, and still it is detected incorrectly.

D. Wrong rank

Cases of incorrectly classified ranks of cards are significantly more rare than in the case of suits. The most common case is that too much of the surroundings of the card is included in the detected area, which leads to either a corner pushed into the search window, or a part of the background detected as a blob. An example of this is shown in figure 10 where the 3 of Hearts is detected as a 4.

There was one incorrect classification of a face card in the test image set. Figure 11 shows a King detected as a 10. This is because the size of the largest blob in the binarized version
of the card was too small to be a face card. If the number of blobs exceeds 10, the card is classified as a 10. Another card in the image is tilted 90°, but still correctly classified.

Fig. 11. A face card, classified as a non-face card.

A. Future work

Future work could include to find a way to make this method work with occluded cards and for cases where the cards are placed on top of each other. It should also be possible to improve the results by making a better corner detector as the corners are not perfectly detected by the current pipeline which means that the recognition step gets a slightly warped card as an input. One of the major reasons for why the suit detection was bad was that clubs were often identified as spades and if a better method for discriminating inbetween spades and clubs the results would improve significantly. Since the cards used in the test set were reflective, some cards in the test set had specular highlights, this problem is similar to occlusion since it was not possible to determine the actual colors of the cards at these points.

VI. CONCLUSION

The method described in this paper managed to classify playing cards, while still being resistant to clutter as well as perspective transformations. The accuracy of the method is currently quite low, but by improving some minor steps in the algorithm, it might be possible to get significantly better results.

REFERENCES