Determining the state of thexas hold 'em in almost to real time

David Molin[†] , Tomas Rosin Forsyth [††]

***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 TODO%**

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 commerical or non-commerical 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 doing 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 [4] 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 whihc describes methods which are possible to use for card recognition. In 95 Kiryati et. al. proposed a method to compute an approximate hough transform by only consider a subset of the points in the image[7]. 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 beeend developed the last decade and these methods can be used to find instances of objects in an image. In 79 Nobuyuki Otsu developed a method to find a threshhold 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 a image of the card oriented in a upright position with a known size. The other prat of the algorithm is to find the suit and rank from a image of a card placed in a upright position.

Fig. 1. Outline of the process used for detecting cards in this paper

A. Extracting the card corners

Fig. 2. Pipeline for identifying the cards and finding their 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 a green tablecloth, or in some cases on a wooden table. This motivates why it should be possible to extract the poker cards from the background by using Otsu's method[5].

Once Otsu's method has extracted a mask of the cards it is possible to extract the contours of the card by applying the method described in [6]. A card will have a contour consisting of 4 lines. Since playing cards are small and the distance to the playing cards are much larger than the size of the card there will be pairs of almost paralell lines for the contours of the cards. This stucture can be used by finding the paralell lines by using a Hough transform. In practice the hough transform is too slow for almost real time applications, therefore the progressive probabilistic hough transform described in [8], This will give results similar to the one in the original Hough transform, but using less computation. From the hough transform it is possible to find the endpoints of long paralell 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 whch maps the cards to a default size.

[†] dmolin@stanford.edu

[††] tomfo@stanford.edu

Fig. 3. Pipeline for determining the rank and suit of a card

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 appearence of the face cards and nonface cards are quite different, the face cards consist of a large image 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 wheather 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 threshhold 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 [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 closes feature.

4) classifying nonface cards: For nonface cards keypoints mathcing 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 possbile to count the number of blobs on each nonface card. Each blob will correspond to one suit symbol. In order to supress noise the blobs would have to be larger than a given threshhold, 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 blue channel of the cards.

IV. Results

A. *Test set*

IMAGES OF THE TEST SET TODO

The method for determining the suit and rank of the cards described in method is in this section 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 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.

		Real class	
		Card	Non card
Detected class	Card	a	b
	NonCard	c	-

TABLE I
CONFUSION MATRIX OF THE CARDS DETECTED BY THE ALGORITHM

The confusion matrix shows that the results for detecting the cards is good.

When runnning the entire pieline the accuracy is $TODO\%$
CARDS & IMAGES INCorectly CLASSIFIED IMAGES
COMMENTS ON the incorrectly classified images
CARDS & IMAGES Corectly CLASSIFIED IMAGES
COMMENTS ON the correctly classified images
TODO

V. Discussion

As can be seen in the results section this approach gives pretty good results. Although they are slightly worse than the results in [3] the training set for that paper was taken is a more controlled environment. TODO

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 eachother. 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.

VI. Conclusion

TODO

References

[1] Lowe, David G. ”Distinctive image features from scale-invariant keypoints.” International journal of computer vision 60.2 (2004): 91-110.

[2] Rublee, Ethan, et al. ”ORB: an efficient alternative to SIFT or SURF.” Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011.

[3] Martins, Paulo, Lus Paulo Reis, and Lus Tefilo. ”Poker vision: playing cards and chips identification based on image processing.” Pattern Recognition and Image Analysis. Springer Berlin Heidelberg, 2011. 436-443.

[4] Zheng, Chunhui, and Richard Green. ”Playing card recognition using rotational invariant template matching.” Proc. of Image and Vision Computing New Zealand. 2007.

[5] Otsu, Nobuyuki. ”A threshold selection method from gray-level histograms.” Automatica 11.285-296 (1975): 23-27.

[6] Suzuki, Satoshi. ”Topological structural analysis of digitized binary images by border following.” Computer Vision, Graphics, and Image Processing 30.1 (1985): 32-46.

[7] Kiryati, Nahum, Yuval Eldar, and Alfred M. Bruckstein. ”A probabilistic Hough transform.” Pattern recognition 24.4 (1991): 303-316.

[8] Matas, Jiri, Charles Galambos, and Josef Kittler. ”Robust detection of lines using the progressive probabilistic hough transform.” Computer Vision and Image Understanding 78.1 (2000): 119-137.