Chinese Chess State Recognition

Yangyang Yu
Department of Electrical Engineering
Stanford University
Email: yyu10@stanford.edu

***Abstract*—In this paper, we present an algorithm that can correctly recognize the state of a Chinese chess game by processing a photo of the chessboard. Some major steps of the algorithm include chessboard rectification using Hough transformation and homographic transformation, chess piece detection using circular Hough transformation and chess piece recognition using SIFT (Scale-Invariant Feature Transform) descriptors [1], RANSAC (RANdom SAmple Consensus) matching [2] and color classification. The method is robust against inclination, rotation, scaling and non-uniformed lighting within certain ranges.**

I. Introduction

Chinese chess is a very popular strategic board game in many Asian countries. It is played between two players, each in control of one side, black or red. Similar to international chess, the goal of the game is to weaken the other side by eliminating their chess pieces. And eventually win the game by eliminating the general of the other side, which is equivalent to the King in international chess.

On each side, there are 16 chess pieces in 7 different types. So in total, there are 32 chess pieces in 14 different types. The different types of pieces are identified by the Chinese character on the face of the chess piece and its color.

The goal of the project is to develop an algorithm that can automatically recognize the state of the chess game from a photo. An example input is given in Fig. 1. With the essential information extracted from the images, we will be able to save and share the record of a game very efficiently for either educational or recreational purpose. Such algorithm can also be a major component of a human-robot interface for chess-playing robots.

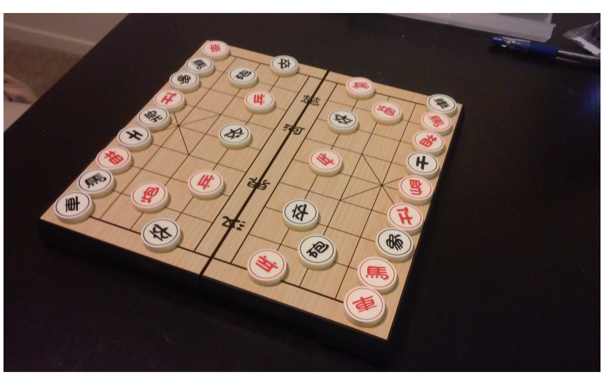


Fig. 1. An example input

II. Related Work

Many researches have been conducted in the field of Chinese-chess recognition. Among these researches, the majority focused on recognition of the individual chess pieces without locating the pieces on the chessboard, with the exception of the following papers.

In [3], the chess locations are calculated from the relative location of the camera and the chessboard, since the camera is attached to the robot arm that can be precisely controlled; In [4], a cross shaped structural element and a circle shaped structural element are used to detect empty intersections and chess pieces. However, this method requires rigid viewing angle of the chessboard; In [5], the authors briefly mentioned that the coordinates are calculated from the four corners of the chessboard without presenting the detailed algorithm.

On the topic of piece recognition, researches proposed various rotation invariant feature extraction methods, since the circular-shaped chess pieces can be placed in any rotated angle on the chess board.

Chen et al. extracted features by equally dividing the chess piece into 360 circular sectors and computing the mean distances from the contours of the character to the center of the piece in each sector [6]. They defined the sector with the max distance to be the first sector to achieve rotation invariance. Hu et al. achieved rotation invariance by transferring the signals into frequency domain and extract only the amplitudes [5]. Du and Huang used the number of crosses with rings of various radius as features [7]. Wang et al. used Radial harmonic Fourier moments as features, which are reported to be invariant to translation, rotation, scaling and intensity [8]. Yang and Wang reported a 3D vector feature extraction method based on the far pixel, near pixel and the angle between the two pixels [9]. Wang and Chen [3], and Wu and Tao [10] chose not to recognize the character on the face of the chess piece. Instead, they recognized and recorded the movements of the pieces and derive the piece information based on the initial game state. This method greatly reduced the image processing requirements but has its limitation of not being able to start recording during a gameplay.

III. Algorithm

In this paper, we propose a chessboard and chess piece recognition algorithm that is robust against various scaling factors, viewing angles, image brightness, and chess piece orientations. The algorithm details are presented in the following sections.

A. Chessboard Segmentation

To separate the chessboard from the back ground, we first binarize the image using Otsus Method. Here, we assume the background is darker than the chessboard and a global threshold is used. Then we perform a series of morphological operations and small area removal to remove the extra background, reflections, details on the chess board and obtain a single connected component that represent the board.

We also experimented with another approach of using the lines on the chessboard for identification and segmentation. However, the result is not ideal. The facts that the fine lines on the chessboard can be easily affected by lighting and reflection; that chess pieces placed on the intersection usually blocked a major portion of the lines, and that the characters on the chess pieces usually create a lot of noise edges all contributed to the non-ideal result.

Fig. 2 shows the intermediate and final results of chessboard segmentation.

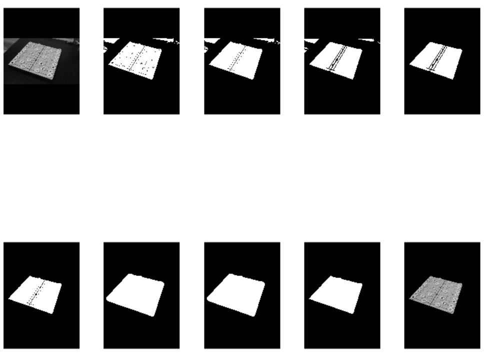


Fig. 2. The intermediate and final results of chessboard segmentation

B. Chessboard Rectification

After we separate the chessboard, we would like to rectify the board into a top-down view that match a known golden reference. The golden reference is a perfect top-down view with corner and intersection coordinates pre-calculated, as shown in Fig. 3.

To perform the match, we need to first find the four corners of our input board as key points for computing the appropriate homographic transformation. We abstract the four edges using Hough transformation and calculate the coordinates of the vertices. [11] Since the corner coordinates of the golden reference are known to us, we can now calculate the transformation matrix and transform our input image to match the golden reference.

Fig. 4 shows the result of a successful transformation. When we overlay the transformed image on top of the reference, the lines on the board are very closely matched. As a result, we can now utilize our knowledge of the intersection locations on the reference image to locate the intersections on the input image. Additionally, since the original image is transformed to a known scale and viewing angle, this step helps the algorithm to be more robust against scaling and viewing angle variance.

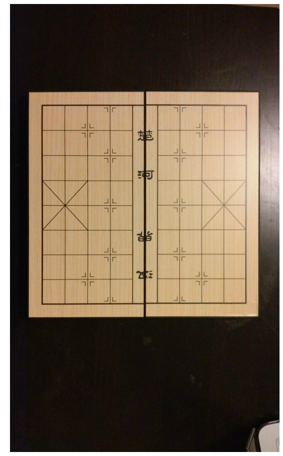


Fig. 3. The golden reference

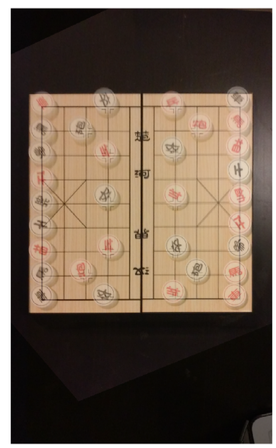


Fig. 4. The intermediate and final results of chessboard segmentation

C. Chess Piece Detection

Since the intersection locations are known after the chessboard is transformed to match the golden reference, we can focus on each intersection and decide if a piece is placed there.

To achieve that, we sample a small window of image around each intersection, as shown in Fig. 5. And perform circular Hough transformation to detect circles, which would most likely to be our chess pieces.

Sometimes, false positives are given when pieces placed on the nearby intersections are partially shown in the window. Two examples are shown in Fig. 6. As we can see, even though the two intersections in the left column are empty, the partially sampled pieces placed on the intersections in the right column result in three false positives.

These false positive are eliminated by thresholding the distance between the center of the detected circle and the intersection. Since each intersection is supposed to be covered by the piece placed on top, ideally, the distance between the center of the detected chess piece and the intersection cannot be larger than the radius of the piece.

In this way, false positives are eliminated as shown in Fig. 7.

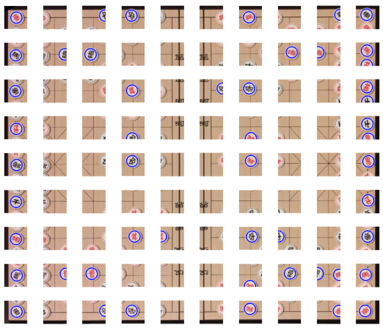


Fig. 5. Samples around intersections

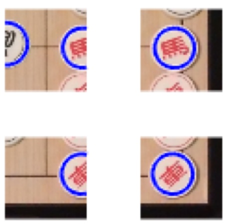


Fig. 6. Piece detection false positives

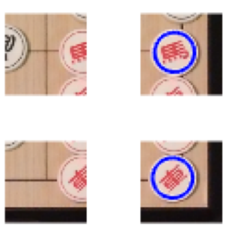


Fig. 7. Piece detection false positives eliminated

D. Chess Piece Recognition

To eliminate any noise in the background, all the pixels outside of the detected circle are masked out. Then we perform the SIFT descriptor [1] and RANSAC matching [2] algorithm on each of the chess pieces, to compare the piece with our reference database that contains the reference images of all 14 types of pieces. The queried image is classified as the same type as the reference image that results in the most matches.

Fig. 8 shows a typical result when two pieces are classified as the same. Fig. 9 shows a typical result when two pieces are classified as different. Because SIFT descriptor and RANSAC matching algorithm is very robust against rotation and lighting variance, our algorithm is able to recognize the casually placed chess pieces in various orientation.

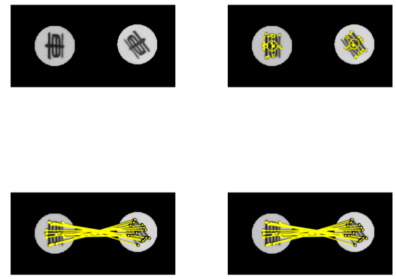


Fig. 8. Typical matched result

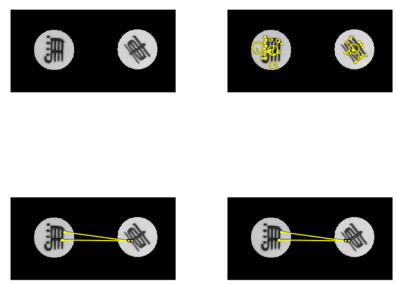


Fig. 9. Typical not-matched result

E. Color Classification

SIFT descriptor and RANSAC matching can correctly classify most of the pieces in this set. However, there are two pairs of piece types that share the same characters on the face. They are differentiated only by their different colors. These pieces are shown in the Fig. 10. On the left are a black chariot and a red chariot. On the right are a black horse and a red horse.

To differentiate these pieces, we need to perform color classification. To eliminate the effect of non-white lighting, instead of using a pre-calculated color classification standard, the classification standard is computed based on the pieces that are already uniquely identified in the last step. More specifically, for each identified piece, the normalized RGB sum is calculated. For each queried piece, the same normalized RGB sum is also calculated. Then we find its nearest identified neighbor in the RGB space and classify the queried piece as the same color as its nearest neighbor. In the rare cases

Fig. 10. Pieces share the same characters

when not enough identified pieces are available, which is not possible in normal game play, a default pre-calculated standard is used.

Fig. 11 shows the distribution of the identified pieces and the queried pieces in the RGB space. Red dots represent the identified red pieces. Blue dots represent the identified black pieces. Green dots represent the queried pieces. As we can see, the different colored pieces are well separated. Thus we can be quite confident on our classification results.

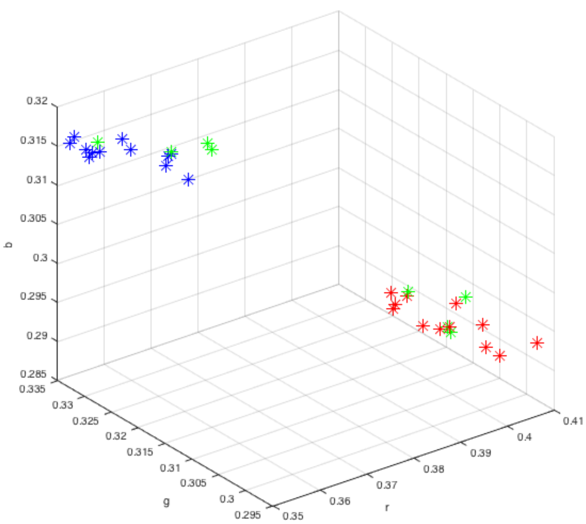


Fig. 11. Distribution of the identified pieces and the queried pieces in the RGB space

IV. Experimental Results

To analyze the algorithms performance under various conditions, the following test images are used. (Fig. 12, Fig. 13, Fig. 14, Fig. 15, Fig. 16)

A. View 1

This test image shows the initial game play set up in a top-down view. It tests the algorithms most basic functionality. It is also worth noticing that even though the viewing angle and chess piece placement are very basic, the lighting of the image is not uniformed. A noticeable bright reflection section can be seen towards the right bottom corner.

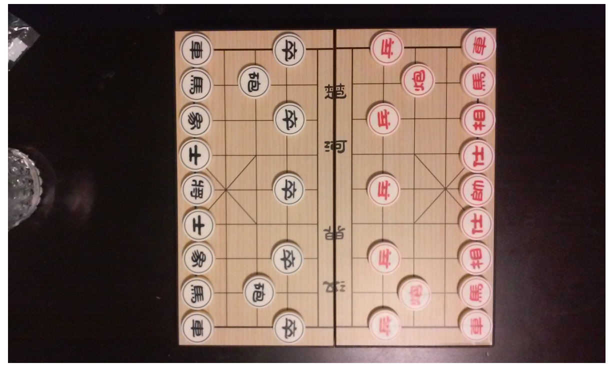


Fig. 12. Test image: View 1

B. View 2

This test image also shows the initial game play set up in a top-down view. However, this view poses an additional challenge in comparison to View 1. The bright reflection section is very close to the right edge. Since the chessboard segmentation and rectification algorithm is based on the edges of the chessboard, such reflection creates a challenge to our algorithm.

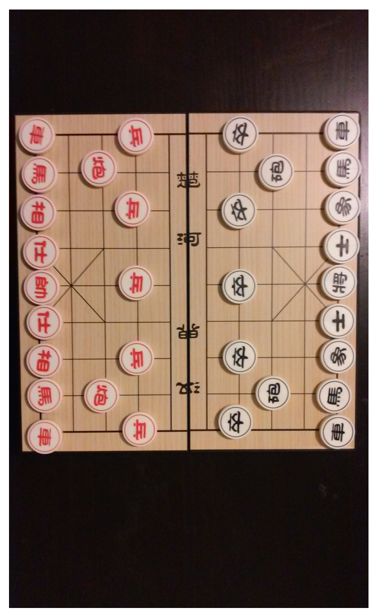


Fig. 13. Test image: View 2

C. View 3

This test image shows the initial game play set up at a viewing angle of about 50 degrees. This test image is used to test the algorithms robustness against viewing angle variance. The non-uniformed lighting condition also helps us verify the algorithms robustness against brightness variance.

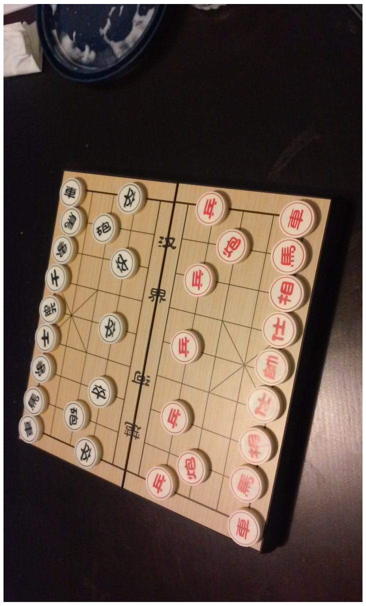


Fig. 14. Test image: View 3

D. View 4

This test image shows a set up with the two-colored chess pieces placed at various locations on the chessboard at a top-down view. This test image is used to verify the color classification algorithm.

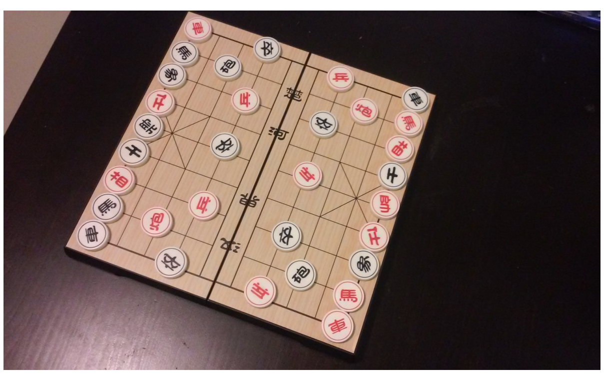


Fig. 15. Test image: View 4

E. View 5

This test image shows a set up with the two-colored chess pieces placed at various locations on the chessboard at a viewing angle of about 50 degrees. This test image is used to verify the combination of the entire workflow.

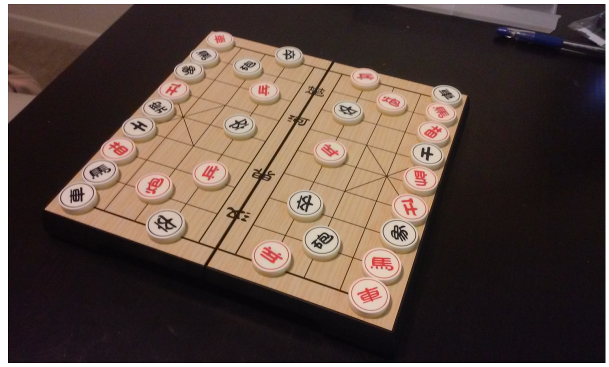


Fig. 16. Test image: View 5

The results are shown in Fig. 17 and Fig. 18.

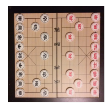
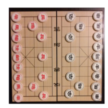
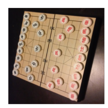
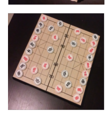
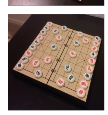
	Scaling Factor	1			0.5			0.3		
		Correct #	Total #	Rate (%)	Correct #	Total #	Rate (%)	Correct #	Total #	Rate (%)
	View 1	32	32	100	31	32	96.8	30	32	93.8
	View 2	31	32	96.8	32	32	100	31	32	96.8
	View 3	32	32	100	32	32	100	31	32	96.8
	View 4	32	32	100	32	32	100	31	32	96.8
	View 5	32	32	100	32	32	100	31	32	96.8

Fig. 17. Recognition rates on each view with different scaling factors

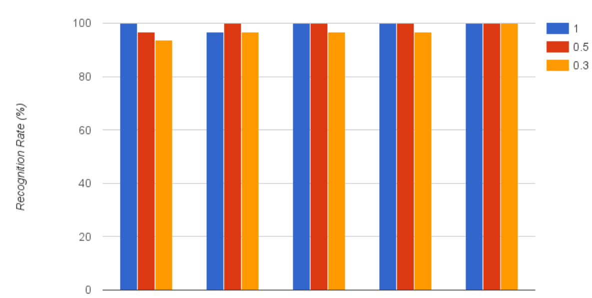


Fig. 18. Chart of recognition rates

V. Result Analysis and Comparison to Alternative Approaches

As shown in the result section, the proposed algorithm is able to achieve very high recognition rate under various scaling factors, viewing angles and lighting conditions. More test cases might be necessary to achieve a more reliable analysis of the proposed algorithm, but the current experimental results demonstrated better recognition rate and higher robustness against variance in comparison to the previous researches mentioned in the related work section. However, the proposed algorithm does require more intensive computation. The current MATLAB implementation takes about 135 seconds on a MacBook Air (Processor: 1.4 GHz Intel Core i5, Memory: 4 GB 1600 MHz DDR3).

VI. Future Work

One major area for future improvement is the chessboard segmentation and corner localization algorithm, As the early steps of the algorithm, all the following recognition steps are largely dependent on its correctness to provide a proper homographic transformation. Most of the errors in the results happen when segmentation result is not ideal due to reflection on the edge of the chessboard. Consequently, the homographic transformation is not able to match the input image with the golden reference image well. The final results thus fail to identify one of the chess pieces. One way to achieve very accurate chessboard segmentation results is to consider taking user input as segmentation assistant, since the algorithm is designed for mobile application.

Another implementation improvement would be porting the entire algorithm onto an Android device. So that the application is no longer dependent on a laptop. The setup will be largely simplified.

Acknowledgment

The author would like to thank Professor Gordon Wetzstein and Kushagr Gupta for their guidance and instruction throughout the course.

References

[1] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ser. ICCV '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 1150–. [Online]. Available: http://dl.acm.org/citation.cfm?id=850924.851523

[2] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[3] X. Wang and Q. Chen, *Vision-Based Entity Chinese Chess Playing Robot Design and Realization*. Springer International Publishing, 2015, vol. 9246, pp. 341–351.

[4] D. Jun-li, Z. Jing-fei, and H. Xin-han, "Chess-board recognition based on vision," *Computer Engineering and Applications*, vol. 43, no. 34, pp. 220–222, 2007.

[5] P. Hu, Y. Luo, and C. Li, "Chinese chess recognition based on projection histogram of polar coordinates image and fft," in *Pattern Recognition, 2009. CCPR 2009. Chinese Conference on*, Nov 2009, pp. 1–5.

[6] W.-Y. Chen, S.-Y. Heish, C.-Y. Yen, and D.-Y. Kuo, "The chinese-chess image identification techniques on spatial domain," in *Intelligent Control and Automation (WCICA), 2011 9th World Congress on*, June 2011, pp. 970–974.

[7] D. J. Li and H. X. Han, "Design of chinese chess robot vision system," *APPLICATION OF ELECTRONIC TECHNIQUE*, vol. 9, pp. 133–136, 2007. [Online]. Available: http://58.68.130.89:8080/JournalSearch/17522340

[8] W. Kejia, Z. Honggang, P. Ziliang, and H. Ying, "Chinese chess character recognition with radial harmonic fourier moments," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, Sept 2011, pp. 1369–1373.

[9] T.-N. Yang and S.-D. Wang, "A rotation invariant printed chinese character recognition system," *Pattern Recognition Letters*, vol. 22, no. 2, pp. 85 – 95, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865500000891

[10] W. Gui and T. Jun, "Chinese chess recognition algorithm based on computer vision," in *Control and Decision Conference (2014 CCDC), The 26th Chinese*, May 2014, pp. 3375–3379.

[11] S. Weatherford, "Pool cue guide," 2013, unpublished.