A Framework for Recognition and Animation of Chess Moves Printed on a Chess Book

Süleyman Eken, Abdülkadir Karabaş, Hayrunnisa Sarı, and Ahmet Sayar
Department of Computer Engineering, Kocaeli University, Turkey

Abstract: *The work presented in this paper proposes a set of techniques to animate chess moves which are printed on a chess book. Those techniques include (1) extraction of chess moves from an image of printed page, (2) recognition of chess moves from the extracted image, and (3) displaying digitally encoded successive moves as an animation on a chessboard. Since all the moves are temporally related, temporal animations show change of spatial patterns in time. Moreover, it becomes easier to understand how the moves are played out and who leads the game. In this study, we animate chess moves printed in Figurine Algebraic Notation (FAN) notation. The proposed technique also eliminates false recognition by means of controlling possible moves in accordance with the rules of chess semantics.*

Keywords: *Animating chess moves, chess character recognition, chess readings, chess document image analysis.*

Received April 22, 2015; accept August 12, 2015

1. Introduction

Pattern recognition has been studied in many fields including psychology, psychiatry, biometrics, bioinformatics and gene expression analysis, cognitive science, traffic flow, handwriting recognition in criminology and banking, Optical Character Recognition (OCR), computational finance, stock markets and many more. OCR systems are divided into two categories:

1. Task-specific systems.
2. General purpose systems.

In task-specific systems, whole document is considered to be text, and certain portions of a document are digitized. This approach is widely used in bank check readers, account processing systems, and airline ticketing readers. In general purpose OCR systems, the document is separated as text and non-text blocks. Text blocks are split into lines, words, and characters. These systems are known as document image analysis [2, 13, 21].

The proposed OCR system is applied on extraction of chess moves recorded as text in Chess Informant documents. Chess Informant is a publishing company produces books of the same name periodically. Each issue offers several hundred games or fragments of games from master play, mostly annotated by the players themselves. A Chess Informant page not only includes game moves, but also many other texts and images scattered around the page. So, the chess moves need to be separated from the other parts of the page [6]. Within this concept, the recognition process we are dealing with falls in the second general class of OCR system mentioned above. Some of the related works on extraction and recognition of chess characters as follows.

Nabiyev [14] initially represented chess readings turning several notations to common notation called Figurine Algebraic Notation (FAN). After chess readings analysis, the transformations of recognized chess characters have been performed. Chess moves have been animated on chessboard, so mistakes in recognition stage have been eliminated by controlling possible moves in accordance with the rules of chess. For character recognition, Nabiyev utilized the heuristic solution together with the figural information related to the notations. Chen [5] developed a feature comparison method for the Chinese-chess object by calculating the distance between the contour of the character and the center of the chess object. Basic image analysis methods such as the noise filter, object extraction, normalization, Feature Calculation (FC) and Maximum Energy Slop (MES) are used to achieve robust Chinese-chess recognition. Zhu *et al.* [24] presented a Chinese chessman pattern recognition system based on rate-connectivity and concentric circle algorithms. According to their experimental results, concentric circle algorithm is more suitable then rate-connectivity algorithm in terms of complexity and accuracy. Wang *et al.* [22] proposed Radial Harmonic Fourier Moments (RHFMs) based algorithm to recognize Chinese chess characters. They evaluated the efficiency of their method to experiment on both toy images and real chess images with recognition rate of 99.49% and 99.57% respectively. Nai-qiang [15] presented Chinese chessboard recognition method considering colour recognition. Colour recognition provides maintaining and restoring of state matrix that can get the information of chessboard. Hu *et al.* [9] proposed a technique recognizing Chinese chess via computer vision methods. This technique firstly utilize Rober operator to detect the edge information of chess then mathematical morphology and template circle method to detect and segment of the chess. Finally, their technique use projection histogram of polar coordinates image and Fast Fourier Transform (FFT) to extract feature of chess characters. All aforementioned studies focus on recognition of Chinese chess characters. These characters are all the same and represented with a single image. So, the recognition process is not complicated. On the other hand, the study presented in this paper deals with the extraction of various chess characters printed on a page of a chess book. We not only deal with extraction and recognition of chess characters but also extraction and recognition of printed chess moves on the pages of the chess book. Moreover, we deal with animating printed chess moves on the user's screen. The book we work on is called Chess Informant book.

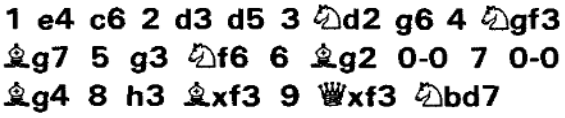
Baird and Tompson [3] proposed an empirical page reader system performed top-down layout analysis-divide a document image into smaller regions-for identification of columns, lines, and characters through skew-estimation technique. By analyzing the formal rules of chess, the error rate was minimized considerably. Baird and Thompson's [3] work guides us about how a page reader for Chess Informant will be constructed. In another work [8], a framework is proposed to recognize chess moves in FAN notation. To extract chess characters, feature vectors are formed. The feature vector contains area, perimeter, and shape properties such as thinness ratio, aspect ratio and compactness. The recognition of chess characters is achieved by using Multilayer Feed-Forward (MLF) neural network with back-propagation learning algorithm and the feature vector.

There are some differences between this study and the other previous works. The previous system works on extracted chess moves and extracts and recognizes characters of these moves. The previous work does not care about extraction of these parts from machine printed pages. The work presented in this paper proposed a set of techniques to extract chess moves printed on a double-column page of a Chess Informant book, and generate animation video from the extracted chess moves. The animation is played on a chess board. The proposed system not only performs extraction and animation but also checks if the moves are legal, in accordance with the chess rules.

The remainder of this paper is organized as follows. In section 2, chess notations are presented for encoding chess moves. Extraction of chess moves from a page, recognition of the extracted chess characters, and animating chess moves are explained in section 3. Moreover, performance evaluation for recognition of chess characters and complexity analysis of the proposed methodology are presented in same chapter. Section 4 draws a conclusion and future works.

2. Notation for Encoding Chess Moves

In a Chess Informant book, each game consists of series of moves, and each move is described by a move number and two plies (half-moves), and each play is described in three characters on average as shown in Figure 1.



1 e4 c6 2 d3 d5 3 ♘d2 g6 4 ♘gf3 ♗g7 5 g3 ♘f6 6 ♗g2 0-0 7 0-0 ♗g4 8 h3 ♗xf3 9 ♕xf3 ♘bd7

Figure 1. An example of chess moves from a page of Chess Informant book.

Different types of chess notations are used for symbolic representation of chess moves. There are several systems developed to encode and record chess moves (position of chess pieces). Each country has its own symbols for chess pieces. The algebraic chess notation is well-known and widely used notation standard. There are a number of different types of algebraic chess notations but popular types used today are Short Algebraic Notation (SAN), Long Algebraic Notation (LAN), and FAN. The Chess Informant books use FAN notation (see Figure 1). To better understand the proposed system, FAN is basically an algebraic notation in which chess figures are represented with symbols. Pawn is the only figure does not have piece identifier (symbol). Chessboards are divided into ranks and files in the algebraic notation. The ranks are known as the horizontal rows of squares labelled 1 through 8. The files are known as the vertical columns of squares labelled a through h. After labelling, each square on the chessboard can be described by a unique combination of file-rank tuples. In this notation, black movescome after white moves. There is a strict order among them. Each move is sequentially numbered. Many books and computer chess programs use FAN. This is as closely as normal algebraic notation except that identifiers of the piece are replaced with graphic symbols of the pieces [7].

There are also various other symbols found in chess informant books to present chess games, besides file-rank tuples and chess figures. Some of them are used by commentators to give evaluative comment on a move like "!" (a particularly good move), "!!" (an excellent move), "?" (a bad move). Some of them indicates the strategic balance of the game position like "∞" (unclear), "+-" (white has a clear advantage), "=/+" (black has slightly better chances), "=" (even position). Some other symbols used in multilingual publications like "□" (the player has counter play), "∇" (the opponent's plan this defends against), "∆" (the future plan this move supports). Characters, symbols, letters, numbers, and figures are concerned in recognition of chess text.

3. Extraction, Recognition and Animating Chess Moves Printed on a Page

The proposed architecture is summarized in the following steps:

- Pre-processing of a page containing chess moves (converting grayscale, smoothing, thresholding) to improve the results of later processing steps (section 3.1).
- Extracting chess moves from a page (section 3.2)
- Horizontal projection profiling to detect printed lines of chess moves (section 3.3).
- Vertical projection profiling to segment characters of each line (section 3.3).
- Recognition of segmented chess characters (symbols, letters, numbers, and figures) (section 3.4).
- Animating digitized chess moves on a chessboard and recording it as a video for educational purposes (section 3.5).

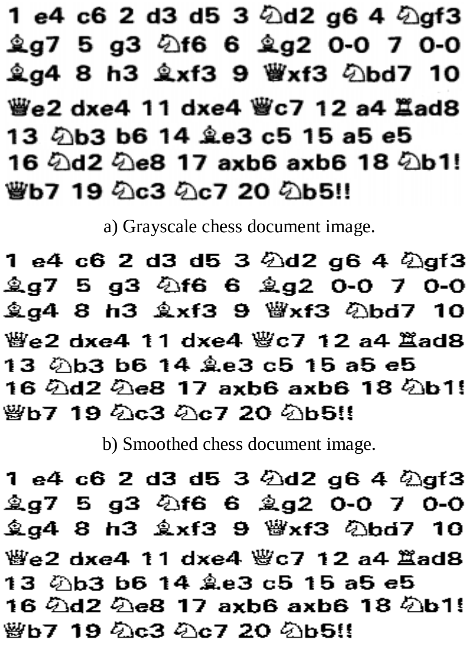
3.1. Pre-processing of an Image for Chess Document

Pre-processing has three main stages: converting chess image document to grayscale, smoothing and binarization. Raw chess image document is firstly converted into grayscale for simplifying the tasks of the subsequent step see Figure 2-a. Then, smoothing technique is applied. Most smoothing techniques remove noise, but they blur edges. It is important to maintain edges, because edges are critically important to the visual appearance of images. We utilize median filtering to reduce noise over the image for a chess document see Figure 2-b. The median filtering preserves certain edge shapes. Applying multiple median filters lessens noise and brightness. To avoid the computational expenses, the median filtering is used with a small size (3x3) window [20].

$$\begin{vmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{vmatrix}$$

In the filter windows, a_1 through a_9 represent grey levels of the pixels.

The aim of binarization is to reduce unwanted information to increase the visibility of the desired information. It helps with differentiating foreground regions characters for chess moves from background regions, which is a paper itself. Fixed threshold level often does not provide accurate distinction between foreground and background portions. This is because of the fact that many scanned chess documents have varying intensity values. So, the Otsu thresholding technique [17] helps here to distinguish foreground objects from the rest of the image see Figure 2-c. The Otsu binarization takes into consideration of gray level values distributions in image as well as the local characteristics of the pixels. After pre-processing steps, columns and text lines in chess image document can be extracted now. Figure 2 shows results of the pre-processing steps.



1 e4 c6 2 d3 d5 3 ♘d2 g6 4 ♘gf3 ♗g7 5 g3 ♘f6 6 ♗g2 0-0 7 0-0 ♗g4 8 h3 ♗xf3 9 ♕xf3 ♘bd7 10 ♕e2 dxe4 11 dxe4 ♕c7 12 a4 ♖ad8 13 ♘b3 b6 14 ♗e3 c5 15 a5 e5 16 ♘d2 ♘e8 17 axb6 axb6 18 ♘b1! ♕b7 19 ♘c3 ♘c7 20 ♘b5!!

a) Grayscale chess document image.

1 e4 c6 2 d3 d5 3 ♘d2 g6 4 ♘gf3 ♗g7 5 g3 ♘f6 6 ♗g2 0-0 7 0-0 ♗g4 8 h3 ♗xf3 9 ♕xf3 ♘bd7 10 ♕e2 dxe4 11 dxe4 ♕c7 12 a4 ♖ad8 13 ♘b3 b6 14 ♗e3 c5 15 a5 e5 16 ♘d2 ♘e8 17 axb6 axb6 18 ♘b1! ♕b7 19 ♘c3 ♘c7 20 ♘b5!!

b) Smoothed chess document image.

1 e4 c6 2 d3 d5 3 ♘d2 g6 4 ♘gf3 ♗g7 5 g3 ♘f6 6 ♗g2 0-0 7 0-0 ♗g4 8 h3 ♗xf3 9 ♕xf3 ♘bd7 10 ♕e2 dxe4 11 dxe4 ♕c7 12 a4 ♖ad8 13 ♘b3 b6 14 ♗e3 c5 15 a5 e5 16 ♘d2 ♘e8 17 axb6 axb6 18 ♘b1! ♕b7 19 ♘c3 ♘c7 20 ♘b5!!

c) Binary chess document image.

Figure 2. Pre-processing of image for a chess document.

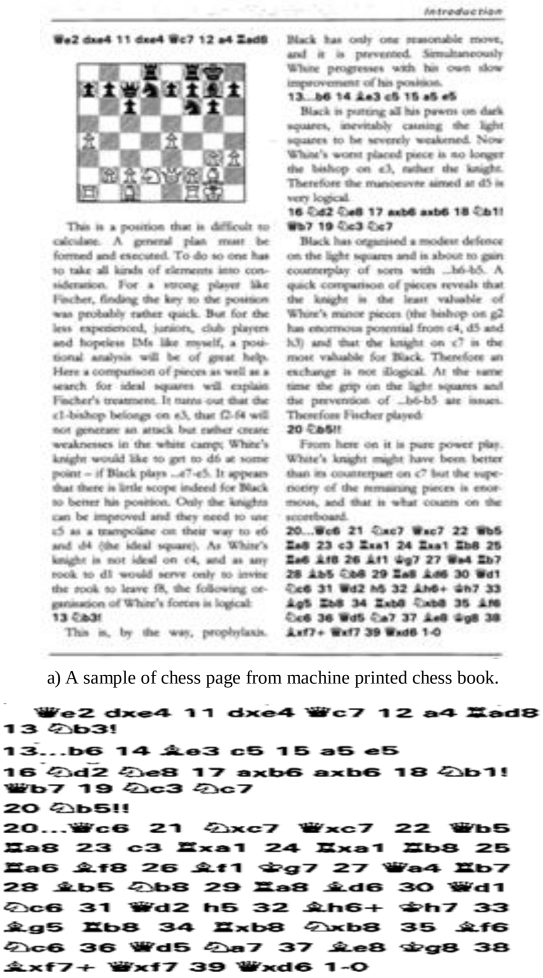
3.2. Extraction of Chess Moves

Throughout the paper, we assume text lines are horizontally and vertically parallel to the respective edges of the paper in a document. In other words, the skew angle of a page is zero, even if a page is manually scanned or photocopied.

Different approaches such as projection profiling, Hough transform, nearest-neighbor clustering, Fourier- transformation and moments have been used to detect text line position by authors in the literature [16, 10]. Chess moves in a page of Chess Informant book are written in bold font as shown in Figure 3-a. The other parts in the pages are explanations of chess moves and they are written in normal font. Using this information, chess moves are extracted from a page via the horizontal and vertical projection profiles. The Horizontal Projection Profile (HPP) is a sum of black pixels along every row of the image. Vertical Projection Profile (VPP) is the sum of black pixels along every column of the image. In Equations.1; n and m stand for column number and row number, respectively. I[x, y] represents intensity value of chess image.

$$HPP(x)=\sum_{y=1}^{n} I[x, y], \quad VPP(x)=\sum_{x=1}^{m} I[x, y] \tag{1}$$

The Chess Informant books are of two column pages. These columns are distinguished by taking vertical profiling. For multicolumn documents, the vertical projection profile will have a plateau for each column, separated by valleys for the between-column and margin spacing. The HPP will have peaks whose widths are equal to the character height and valleys whose widths are equal to the spaces between lines. Each column is distinguished by applying VPP, and each line is distinguished and separated by applying HPP. Moreover, since the chess moves are printed in bold format, the HPP values for the lines of chess moves are larger than the other parts' HPP values. This helps us to separate or extract the chess moves. After applying HPP and VPP processes, the chess moves are extracted from a page see Figure 3-a as shown in Figure 3-b. As shown in Figure 3-a, chess moves are scattered around the pages see bold texts. These are collected as one set of moves as shown in Figure 3-b.


Introduction

♕e2 dxe4 11 dxe4 ♕c7 12 a4 ♖ad8

This is a position that is difficult to calculate. A general plan must be formed and executed. To do so one has to take all kinds of elements into consideration. For a strong player like Fischer, finding the key to the position was probably rather quick. But for the less experienced, juniors, club players and hopeless IMs like myself, a positional analysis will be of great help. Here a comparison of pieces as well as a search for ideal squares will explain Fischer's treatment. It turns out that the c1-bishop belongs on e3, that f2-f4 will not generate an attack but rather create weaknesses in the white camp; White's knight would like to get to d6 at some point – if Black plays ...e7-e5. It appears that there is little scope indeed for Black to better his position. Only the knights can be improved and they need to use c5 as a trampoline on their way to e6 and d4 (the ideal square). As White's knight is not ideal on c4, and as any rook to d1 would serve only to invite the rook to leave f8, the following organisation of White's forces is logical:

13 ♘b3!

This is, by the way, prophylaxis. Black has only one reasonable move, and it is prevented. Simultaneously White progresses with his own slow improvement of his position.

13...b6 14 ♗e3 c5 15 a5 e5

Black is putting all his pawns on dark squares, inevitably causing the light squares to be severely weakened. Now White's worst placed piece is no longer the bishop on e3, rather the knight. Therefore the manoeuvre aimed at d5 is very logical.

16 ♘d2 ♘e8 17 axb6 axb6 18 ♘b1! ♕b7 19 ♘c3 ♘c7

Black has organised a modest defence on the light squares and is about to gain counterplay of sorts with ...b6-b5. A quick comparison of pieces reveals that the knight is the least valuable of White's minor pieces (the bishop on g2 has enormous potential from c4, d5 and h3) and that the knight on c7 is the most valuable for Black. Therefore an exchange is not illogical. At the same time the grip on the light squares and the prevention of ...b6-b5 are issues. Therefore Fischer played:

20 ♘b5!!

From here on it is pure power play. White's knight might have been better than its counterpart on c7 but the superiority of the remaining pieces is enormous, and that is what counts on the scoreboard.

20...♕c6 21 ♘xc7 ♕xc7 22 ♕b5 ♖a8 23 c3 ♖xa1 24 ♖xa1 ♖b8 25 ♖a6 ♗f8 26 ♗f1 ♔g7 27 ♕a4 ♖b7 28 ♗b5 ♘b8 29 ♖a8 ♗d6 30 ♕d1 ♘c6 31 ♕d2 h5 32 ♗h6+ ♔h7 33 ♗g5 ♖b8 34 ♖xb8 ♘xb8 35 ♗f6 ♘c6 36 ♕d5 ♘a7 37 ♗e8 ♔g8 38 ♗xf7+ ♕xf7 39 ♕xd6 1-0

a) A sample of chess page from machine printed chess book.

♕e2 dxe4 11 dxe4 ♕c7 12 a4 ♖ad8
13 ♘b3!
13...b6 14 ♗e3 c5 15 a5 e5
16 ♘d2 ♘e8 17 axb6 axb6 18 ♘b1!
♕b7 19 ♘c3 ♘c7
20 ♘b5!!
20...♕c6 21 ♘xc7 ♕xc7 22 ♕b5
♖a8 23 c3 ♖xa1 24 ♖xa1 ♖b8 25
♖a6 ♗f8 26 ♗f1 ♔g7 27 ♕a4 ♖b7
28 ♗b5 ♘b8 29 ♖a8 ♗d6 30 ♕d1
♘c6 31 ♕d2 h5 32 ♗h6+ ♔h7 33
♗g5 ♖b8 34 ♖xb8 ♘xb8 35 ♗f6
♘c6 36 ♕d5 ♘a7 37 ♗e8 ♔g8 38
♗xf7+ ♕xf7 39 ♕xd6 1-0

b) Extraction of chess moves portions from printed page.

Figure 3. Obtaining of chess moves from printed page.

3.3. Segmentation of Chess Characters

After having extracted scripts of chess moves from the page, segmentation is performed. To separate text lines of extracted chess moves, the HPP is performed. White spaces between the lines of chess moves are used to segment the lines. HPP has zero height valleys between the lines. Line segmentation is done at this zero height points. To improve result of the HPP, smoothing function (S(n)) is applied on HPP. Smoothing is used to remove spurious peaks and valleys of the histogram.

A moving average filter is applied for smoothing.

$$S(n)=\frac{1}{k}\sum_{i=-\frac{k}{2}}^{\frac{k}{2}}H(n+i) \qquad (2)$$

In Equation (2), H(n) is histogram function and k is a smoothing factor. Figure 4 illustrates HPP of the extracted chess moves and Figure 5 illustrates the line segmentation point after applying smoothing function.

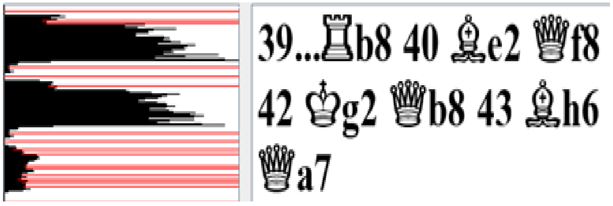


Figure 4. Text line extraction without smoothing function.

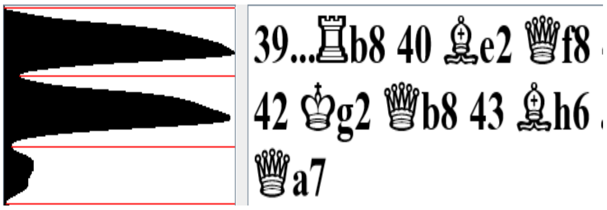


Figure 5. Text line extraction with smoothing function.

To extract each character from a segmented text line, VPP is performed. VPP has valleys whose widths are equal to the between-line spacing or low peaks between characterssee Figure 6.

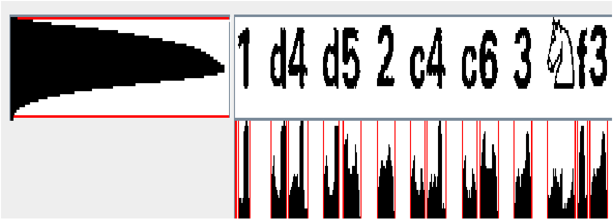


Figure 6. Running vertical projection profile on text lines.

Chess characters are segmented according to points found after VPP. To improve success of recognition step, all background pixels white pixels are trimmed. In other words, all characters are enclosed in full frame see Figure 7. Then, recognition process is performed on each candidate character.

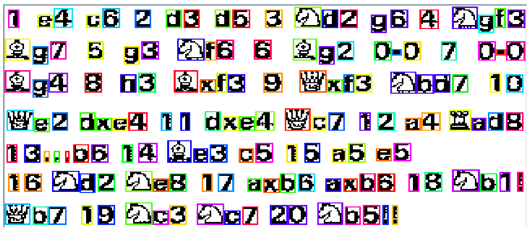

1 e4 c6 2 d3 d5 3 ♘d2 g6 4 ♘gf3
♗g7 5 g3 ♘f6 6 ♗g2 0-0 7 0-0
♗g4 8 h3 ♗xf3 9 ♕xf3 ♘bd7 10
♕e2 dxe4 11 dxe4 ♕c7 12 a4 ♖ad8
13...b6 14 ♗e3 c5 15 a5 e5
16 ♘d2 ♘e8 17 axb6 axb6 18 ♘b1!
♕b7 19 ♘c3 ♘c7 20 ♘b5!!

Figure 7. Segmented chess characters.

3.4. Recognition of Segmented Characters

To recognize chess characters used for representing chess moves, a data set is created.

3.4.1. Dataset Creation

To our knowledge no previous dataset for chess moves from machine printed pages has been created. We have created a dataset from the first thirty pages of Aagaard Chess book [1]. This is done by means of aforementioned character segmentation techniques. It is very important to know that all characters in dataset are gray level, not two-level. The dataset has 1674 number of character samples for each one of 30 different characters. Figure 8 shows a set of samples of images belonging to bishop. As shown in Figure 8, some of them are clear and some of them are noisy.The dataset is divided into mutually exclusive training and test sets. Training set contains 70% of the total samples and test set contains 30% of the total samples. The images in created dataset might be in varying sizes, even if for the same character. Therefore, characters need to be resized. All characters are enlarged to a fixed size (20x20) by using a bilinear interpolation algorithm [4].

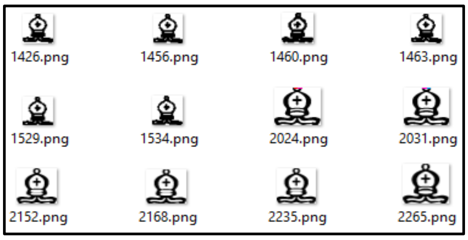


Figure 8. Sample bishop images from the dataset.

3.4.2. K-NN Classification Based on Euclidian Distance

In k-NN classification, training patterns are plotted according to their observed feature values and are labeled according to their known classes. An unlabeled test pattern is predicted according to the most frequently occurring class among its k-most similar training patterns majority voting; its nearest neighbors. The most common used similarity measures for k-NN classification are Euclidean, Minkowski, Manhattan and Mahalanobis distances [12, 23] The Euclidian distance metric is used for similarity measure. k value is set to 1 to predict the same value, or class, as the nearest instance in the training set.

In the training phase, trained characters and their matrix forms are created. This is done for each character in the training set. In other words, a trained character is obtained by averaging all samples belonging to same character (e.g., pawns) using Equation (3).

$$T[i, j]=\frac{1}{N}\sum_{s=1}^{N} C_s[i, j] \qquad (3)$$

In Equation 3, T[I, J] represents matrix form of trained character and C_s[I, J]represents matrix of a sample belonging to the same character. Stands for number of samples. So, range value for elements of matrix T[I, J], is between 0 and 255 see Figure 9. In order to minimize effect of too large or too small values in the matrix, we use min-max normalization. So, elements of trained matrix are normalized between 0 and 1. Suppose that minA and maxA are the minimum and maximum values of an element of matrix T. Min-max normalization maps a value, *vi*, of T to v_i' in the range [new_minT, new_maxT] by computing

$$v_i'=\frac{v_i\text{-}min_T}{max_T\text{-}min_T}\left(new\_max_T\text{-}new\_min_T\right)+ new\_min_T \qquad (4)$$

Here, [min_T-max_T] range represents [0-255] and [new_min_T-new_max_T] range represents [0-1].

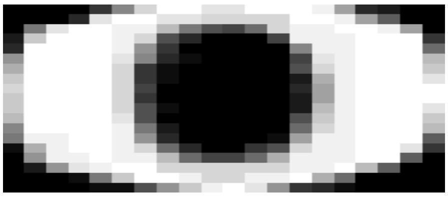


Figure 9. A trained image for character “0”.

In the test phase, the nearest instance in the training set is predicted by means of Euclidean distance metric. The smallest distance value (E) represents the greatest similarity to testing segmented character I [i, j].

$$E=\sum_i^n \sum_j^m \left|I[i, j]\text{-}T[i, j]\right| \qquad (5)$$

3.4.3. Performance Evaluation for Recognition Chess Characters

Results for chess character recognition are evaluated using sensitivity, specificity and accuracy as the performance measures. The measures are described below as:

$$\text{Sensitivity} = TP/(TP + FN) \qquad (6)$$

$$\text{Specificity} = TN/(TN + FP) \qquad (7)$$

$$\text{Accuracy} = (TP + TN)/(TP + TN + FP + FN) \qquad (8)$$

In Equations 6, 7, and 8, *TP* = True Positive, *TN*=True Negative, *FP* = False Positive and FN=False Negative. An event is said to be *TP* if a character is correctly classified and *TN* when a character is correctly non-classified. In related development, an event is said to be *FN* if a character is incorrectly non-classified and a *FP* when a character is incorrectly classified. Sensitivity measure indicates the ability of a classification technique to recognize the character while specificity measures the proportion of negatives which are correctly classified. The accuracy measure however indicates the degree of conformity of the correctly classified characters to the ground truth. Table 1 shows performance results for chess character recognition.

Table 1. Performance results for character recognition.

Chess Characters	Avg. Sensitivity	Avg. Specificity	Avg. Accuracy
0	100%	100%	100%
1	97%	100%	97%
2	100%	100%	100%
3	100%	100%	100%
4	100%	100%	100%
5	96%	100%	99%
6	100%	99%	99%
7	100%	100%	100%
8	100%	100%	100%
9	100%	99%	99%
a	100%	100%	100%
b	100%	100%	100%
c	100%	99%	99%
d	100%	100%	100%
e	95%	100%	99%
f	100%	100%	100%
g	97%	100%	99%
h	100%	100%	100%
!	100%	99%	99%
-	100%	100%	100%
.	100%	100%	100%
1/2	100%	100%	100%
+	100%	100%	100%
x	100%	99%	99%
Knight	100%	100%	100%
Pawn	100%	100%	100%
King	100%	100%	100%
Queen	100%	100%	100%
Rook	100%	100%	100%
Bishop	100%	100%	100%

3.5. Animating Recognized Chess Characters on Chessboard

Animation has been used in many real world applications to recognize the patterns in spatially and timely changing datasets [18, 19]. Here, each move results in one character to change its place on chessboard. When we play those moves successively, the game will be more easily understandable. If a wrong recognition happens during the game extraction phase, false recognitions are eliminated in accordance with the chess semantics, by applying knowledge of the rules of chess. Also, animated games are possibly recorded as videos for educational purposes see Figure 10.

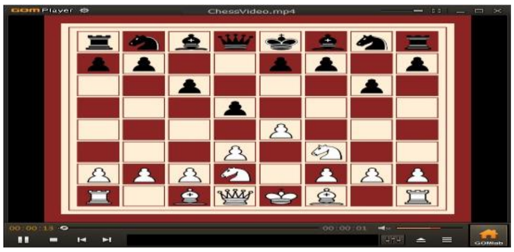


Figure 10. A sample screen for animating extracted and recognized chess moves.

3.6. Computational Complexity Analysis of Each Step

Computational complexity of each steps of proposed algorithm is analyzed as following:

- In median filtering step, the median of pixel values inside a filter window has to be calculated. In its straight-forward implementation the pixel values are sorted and the value located at $(k^2/2)^{th}$ position is taken, where k is the filter size, i.e. the size of one side of the square filter window. If quicksort is used, the computational complexity of filter is O(klgk) operations per pixel of result image. Total cost for entire image is $O(n^2 klgk)$, but we can eliminate klgk because k<<n. Finally, median filtering step costs $O(n^2)$.
- For binarization of chess image with L gray levels using M-1 thresholds, Otsu's exhaustive method searches $\binom{L}{M-1}$combinations of thresholds, which can be approximated to $O(L^{M-1})$ for (M-1) << L. Here, the pixels in the chess image document to two classes (black and white). So, total cost for segmenting is O(L)[11].
- HPP and VPP steps require operation of Equation (1). on entire image. So its cost is $O(n^2)$.
- We have n examples to be stored with d-dimension. O(d) is necessary to compute distance to each example and O(nd) to find one nearest neighbor. O(knd) costs to find k closest examples. In this study, k is chosen as 1. So, complexity is O(nd).

Total computational complexity of proposed steps to extract and recognize chess characters from machine printed pages is $O(n^2 + L + n^2 + nd) = O(n^2)$. Because, O(L) and O(nd) are negligible in the complexity analysis.

4. Conclusions and Future Works

It may be very hard to learn chess openings, endings for chess players from machine printed chess books. In this study, we propose some techniques to read in the Chess Informant series which contain descriptions of chess games selected by chess masters and published by the International Chess Federation (FIDE). Proposed techniques able to analyse the double-column pages, extract game descriptions from machine printed pages, recognize these chess characters and animate on chessboard. To animate chess moves, the system also check each move for its legality.

In the future, we aim at extracting and recognizing chess moves in other alternative notations such as SAN and LAN by using more robust classification techniques. For example, we plan to use shape analysis algorithms which are typical for such tasks and lead to muchbetter results even for more complicated tasks than simple recognition of several symbols instead of basic image analysis methods such as horizontal or vertical projection or area estimation. Also, a chess playing robotic arm system, capable of learning from extracted and recognized chess moves in Chess Informant series, can be developed.

References

[1] Aagaard J., *Excelling at Positional Chess*, Everyman Chess, 2003.

[2] Baird H., “Document Recognition without Strong Models,” *in Proceedings of the International Conference on Document Analysis and Recognition*, Beijing, pp. 414-423, 2011.

[3] Baird H. and Tompson K., “Reading Chess,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 6, pp. 552-559, 1990.

[4] Battiato S., Gallo G., and Stanco F., “A New Edge-Adaptive Algorithm for Zooming of Digital Images,” *in Proceedings of IASTED Signal Processing and Communications*, Catania, pp. 144-149, 2000.

[5] Chen W., “Chinese-Chess Image Recognition by using Feature Comparison Techniques,” *Applied Mathematics and Information Sciences*, vol. 8, no. 5, pp. 2443-2453, 2014.

[6] Chess Informant series, http://www.chessinformant.rs/, Last Visited 2015.

[7] Chesscrner, www.chesscorner.com, Last Visited 2015.

[8] Eken S. and Sayar A., “Animating Chess Moves Recorded on Chess Informant,” *in Proceedings of the ^{3}rd International Symposium on Computing in Science and Engineering*, Kuşadası, pp. 35-40, 2013.

[9] Hu P., Luo Y., and Li C., “Chinese Chess Recognition Based on Projection Histogram of Polar Coordinates Image and FFT,” *in Proceedings of the Chinese Conference on Pattern Recognition*, Nanjing, pp. 1-5, 2009.

[10] Kasturi R., and O’Gorman L., Govindaraju V., “Document Image Analysis: A Primer,” *Sadhana*, vol. 27, no. 1, pp. 3-22, 2002.

[11] Liu D. and Yu J., “Otsu Method and K-means,” *in Proceedings of 9th International Conference on Hybrid Intelligent Systems*, Washington, pp. 344-349, 2009.

[12] Majumdar A. and Bhattacharya A., “A Comparative Study in Wavelets, Curvelets and Contourlets as Feature Sets for Pattern Recognition,” *The International Arab Journal of Information Technology*, vol. 6, no. 1, pp. 47-51, 2009.

[13] Nabiyev V., *YapayZeka: Problemler Yöntemler, Algoritmalar*, Seçkin Publishing, 2005.

[14] Nabiyev V., “Providing Harmony among Different Notations through the Chess Readings,” *in Proceedings of the IEEE 19th Signal Processing and Communications Applications Conference*, Antalya, pp. 29-33, 2011.

[15] Nai-qiang Z., “Improved Chinese Chessboard Recognition Method,” *Journal of Computer Applications*, vol. 30, no. 4, pp. 980-981, 2010.

[16] O'Gorman L. and Kasturi R., *Document Image Analysis*, IEEE Computer Society Press, 1995.

[17] Otsu N., “A threshold Selection Method from Gray-Level Histograms,” *IEEE Transactions on Systems Man and Cybernetics*, vol. 9, no. 1, pp. 62-66, 1979.

[18] Sayar A., “A Distributed Map Animation Framework for Spatiotemporal Datasets,” *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 24, no. 2, pp. 683-694, 2016.

[19] Sayar A., “A Service Oriented Framework for Animating Big Spatiotemporal Datasets,” *in Proceedings of Web Information Systems Engineering-WISE 2011 and 2012 Workshops*, Sydney, pp. 238-250, 2013.

[20] Schalkoff R., *Digital Image Processing and Computer Vision*, Wiley, 1992.

[21] Tsujimoto S. and Asada H., “Major Components of a Complete Text Reading System,” *IEEE*, vol. 80, no. 7, pp. 1133-1149, 1992.

[22] Wang K., Zhang H., Ping Z., and Hai Y., “Chinese Chess Character Recognition with Radial Harmonic Fourier Moments,” *in Proceedings of the International Conference on Document Analysis and Recognition*, Beijing, pp. 1369-1373, 2011.

[23] Weinberger K. and Saul L., “Distance Metric Learning for Large Margin Nearest Neighbor Classification,” *Journal of Machine Learning Research*, vol. 10, pp. 207-244, 2009.

[24] Zhu H., Lei J., and Tian, X., “A Pattern Recognition System Based on Computer Vision The method of Chinese chess recognition,” *in Proceedings of the IEEE International Conference on Granular Computing*, Hangzhou, pp. 865-868, 2008.

Süleyman Eken has received his BEng. in Computer Engineering from Karadeniz Technical University in 2009 and ME. in Computer Engineering from Kocaeli University in 2012. He is currently working towards Ph.D. degree in Computer Engineering from Kocaeli University, Turkey. Also, he is currently a Research Assistant of Computer Engineering Department at Kocaeli University in Turkey. His current research interests include distributed file systems, data synchronization, Satellite Image Processing, Remote Sensing, WEB-GIS applications and Spatial Databases.

Abdülkadir Karabaş received the B.S. degree from the Computer Engineering Department, Kocaeli University, Kocaeli, Turkey, in 2015. His research interests include computer vision and pattern recognition.

Hayrunnisa Sarı received the B.S. degree from the Computer Engineering Department, Kocaeli University, Kocaeli, Turkey, in 2015. Her research interests include computer vision and pattern recognition.

Ahmet Sayar received M.S. in Computer Science (2001) from Syracuse University and Ph.D. in Computer Science (2009) from Indiana University, USA. He has worked at Los Alamos National Laboratory (New Mexico, USA) and Community Grids Laboratory (Indiana, USA) as a researcher. Since 2010, he is a professor at Computer Engineering Department, Kocaeli University, Turkey. His current research interests include Distributed Systems, Big Data, Data-Intensive Computing, Remote Sensing, GIS, and Spatial Data-Databases. He has co-authored 3 books and around 70 papers.