Automatic Analysis of Players Behavior in Real Dyadic Chess Situations

Thomas Küchelmann, Paola Torche, Manjunath Prasad, Kai Essig, Thomas Schack

Automatic Analysis of Players Behavior in Real Dyadic Chess Situations

Thomas Küchelmann[1]
thomas.kuechelmann@uni-bielefeld.de

Paola Carolina Torche[1]
pa.torche@uni-bielefeld.de

Manjunath Prasad[1]
m.prasad@uni-bielefeld.de

Kai Essig[1]
kai.essig@uni-bielefeld.de

Thomas Schack[1]
thomas.schack@uni-bielefeld.de

[1] Center of Excellence Cognitive Interaction Technology
Bielefeld University
Inspiration 1, 33619 Bielefeld, Germany

ABSTRACT

The present paper brings together solutions for full-automatic analysis of chess moves and eye tracking data for real dyadic chess games. We combine an electronic chessboard with marker detection in order to automatically protocol relevant and situation-related chess moves and visual attention patterns on pieces and squares. Very first results from initial evaluation studies indicate that the presented solution allows for a reliable automatic offline analysis of the underlying strategic and attentional processes of chess behaviour in real game situations without any need for an error prone and time consuming manual data annotation. The proposed solution allows to run chess studies in shorter time and to analyse the differences between chess experts and novices with respect to visual attention and strategies in more detail. Finally, gained insights can be used to develop interactive and intuitive electronic chess assistant systems to analyse the weaknesses of both experts and novices and to recommend optimal moves in given chess situations, as well as to enhance players' expertise with individualized training methods and explanations.

AUTHOR KEYWORDS
Human-machine-interaction; eye tracking; adaptive artificial assistive systems; virtual agents; attentive learning processes; human perception; multi-modal data; visual behavior; mental memory representations

ACM CLASSIFICATION KEYWORDS
Algorithms; documentation; experimentation; human factors; measurement

1. INTRODUCTION

The techniques and experimental settings described in this paper have chess as a research object as it is a highly competitive expert domain enjoying worldwide popularity. Moreover, it is characterized by exactly differentiable skill levels and, although embedded into an easily controllable setting, limited in space, material and time, it is extremely complex in event space constellations. Therefore, chess involves great complexity in decision making, strategy and memory. This makes chess an outstanding research object in the fields of visual attention, artificial intelligence, acquisition of expertise, as well as the interplay between mental and emotional representation.

Performance and cognitive abilities of master chess players and the differences in the strategies and visual attention processes between chess experts and novices have been investigated in many studies of cognitive sciences. Pioneering work are the studies by de Groot [1,2] and Chase and Simon [3]. Both show that expert players could reproduce previously shown chess constellations much more accurately than less skilled ones. In the upshot, chess grand-masters use efficient perceptual encoding of chess configurations to generate the most promising candidate moves [4]. Therefore the encoding of chess configurations is a key determinant of chess skill. This leads to the theory of chunking [3] and templates. Chunks in chess are defined as structural units of up to five pieces which remain intact through encoding into a long-term memory if at least two thirds of its pieces remain together upon recall. Gobet and Simon [5] have proposed "templates" as an alternative to chunks. Templates can be understood as bigger chunks of up to 15 pieces [6]. The divergent predictions of the chunking and template theory have been tested by Gobbet and Clarkson [7] who found that expert players recall more pieces than predicted by the chunking theory in its original form. Thus, chunking and template emergence seems to be the key to understanding why humans are superior in long-range planning and static evaluation despite their small capacity of working memory. Also reaction times in chess detection tasks give deeper insight into experts' perceptional superiority. E.g. in [8] players had to judge the presence or absence of a check in certain chess constellations with and without distracting pieces. Significantly higher reaction times for novices but not for experts when verifying non-checking constellations with additional distractors suggest automatic and parallel encoding procedures for chess relations in experts. Further Kiesel and Kunde [9] have shown in a masked prime study with either a checking or non-checking chess configuration that expertise improves perceptual processing to an extent that allows complex visual stimuli to bias behaviour unconsciously.

With the advent of modern eye trackers gaze information has recently been used as a supplementing measure for investigating the perceptual aspects of chess skills [10]. If chess experts encode chunks rather than individual pieces then this should result in fewer fixations and the visual span should increase as a matter of expertise, allowing players to make greater use of peripheral and parafoveal processing and therefore to extract information from a larger area of the chessboard during a fixation [4]. This was reflected by more recent research with modern eye trackers. Charness et al. [11] found that experts produced more fixations on empty squares than intermediates. When fixating pieces, experts produced a greater proportion of fixations on relevant pieces than intermediates. Sheridan et al [12] found that experts show faster recognition of complex visual patterns than novices. A survey of the contribution of eye movement research to the study of human expertise in chess can be found in [4].

Contrary to the above mentioned studies, our techniques and experimental settings described in the following sections employ chess situations with a real chessboard and real chess pieces, and this scenario is much more complicated in terms of data recording and analysis [4]. This is in stark contrast to all previous studies mentioned as they were screen-based and have employed artificial chess stimuli only (i.e., images of artificial chessboards with different chess constellations). In studies where an eye tracker is used, participants look at artificial chessboards (protocolling chess movements in a temporal order) presented on the screen while their eye movements are analysed with regard to fixations on occupied and empty chess fields, as well as jumps between them. Because of this highly controlled experimental design, the recorded data can mainly be analysed automatically. One crucial disadvantage of such a controlled screen-based design is that it eliminates several factors of a real game situation, such as human opponents, the influences of gestures, deceptive movements and the audience. We are interested in the question in what way the results will be different as soon as they go into the real chess game setting and into task-oriented scenarios which call for solving chess tasks on real chessboards and with real chess pieces. How far will outcomes of previous studies resemble the results from real dyadic situations? What will be the influences of the other factors on players' behaviour, strategies, as well as visual attention processes? And how will this depend on players' expertise? In the real settings, manual data annotation is always possible since it can be done based on video material. But this is an ineffective alternative as it is tedious and error prone - especially in the context of rapid chess matches in which each player's time control is strictly limited to less than 10 minutes and in which pieces can be moved very fast (e.g. "blitz" matches with a total match time of 5 minutes only for each player). Additionally, participants can freely move their heads which makes the correlation of the gaze position to the respective chess field or chess piece much more difficult. Therefore, due to the best of our knowledge, there are so far no multi-modal studies on real dyadic chess games. The real chess situation requires the analysis of gaze videos and an automatic detection of the single chess pieces and fields, which goes along with a lot of environmental challenges (such as uncontrolled head movements and lighting conditions), not yet adequately solved.

Therefore, in this paper an offline full-automatic annotation and analysis solution for real dyadic chess games and task-oriented scenarios with real chessboards and pieces is presented and described in the following section.

2. Hardware

For the mapping of each piece's identity onto its actual position and a transcript of all moves, a DGT (Digital Game Technology) electronic chessboard (e-board; see *Fig. 1* and https://www.webwiki.com/digitalgametechnology.com) is used that transmits game data via Bluetooth or USB to a computer [13]. It does so in the common algebraic chess notation (short or long notation can be chosen) in accordance with the FIDE [14] (Fédération Internationale des Échecs, the World Chess Federation) rules, e.g. 1.Nf3 when the first move in a match is knight from square g1 to f3.

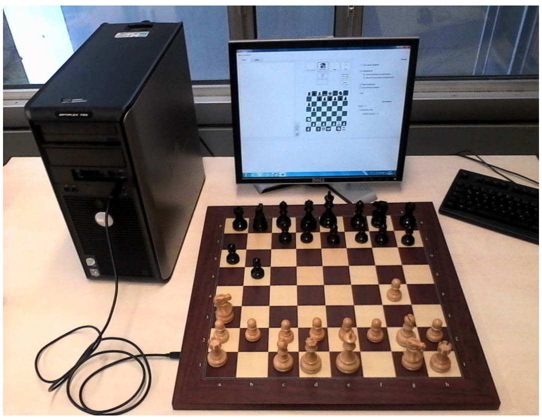


Figure 1. DGT electronic chessboard connected via USB to a computer.

The e-board can identify both type and colour of every chess piece and its location with a sample rate of 6 times per second which is sufficiently high for blitz chess matches. The internal memory of the e-board for recording chess games can store up to 500 moves. Different chess programs can be employed for recording and analysis. The board design is classic (standardized tournament regulations: 55 x 55 mm tournament-size squares, 8 mm thickness, chessboard classical wooden piece design and size).

For the recording of eye movements we have used the non-invasive mobile and binocular Eye Tracking Glasses (ETG-2) from Sensomotoric Instruments (SMI) [15] (*Fig. 2*). This is a binocular system with a sampling rate of up to 120 Hertz and a gaze tracking accuracy of 0.5° over all distances. For our recordings we have used the scene camera resolution of 960x720 pixels, 30 fps.

Figure 2. Chess player wearing the SMI mobile Eye tracking Glasses during game play.

3. ELECTRONIC CHESSBOARD - PROGRAMMING AND HANDLING

Automatically tracking of players' movements, positions of all chess pieces, and particularly the position of the chunks at all time is necessary in order to analyse correlations between visual attention and individual chess behaviour. In order to do this, we use the DGT e-board (See Section 2, and *Fig. 1*) connected to the computer, and a software to handle the communication between the board and the computer. The original software from DGT allows to track the movements of the game but lacks the feature for chunk tracking and needs an external clock to annotate times. To include these two important features, we developed our own software, based on an API (Application Programming Interface) provided by DGT. By using the API we can send commands via the serial port directly to the e-board to get the corresponding data, such as chess movements and state of each cell (i.e., which pieces are on each cell at a given time). The two main additions to our software with respect to the official one are the time annotation functionality without using an external clock -we use the computer internal clock instead-, as well as the tracking of chunks. Our software allows to mark chess pieces of chunks at the beginning of the game and to track them during the chess match (see *Fig. 3*). In this way, we are able to annotate the position of all pieces from the chunk at any time, which is useful to finally compare it with the visual attention patterns recorded by the eye-tracker.

The messages from the electronic board to the computer can be of two kinds. We call them 'state' message, and 'update' message. A state message is a 64-byte long message, where each byte refers to one cell from the board. The value of the byte indicates if the cell is empty -value zero-, or if it is occupied by a piece -value equal to the piece code. There are 12 pieces codes, one for each type of figure and colour: white pawn 1, white rook 2, white knight 3, white bishop 4, white queen 5, white king 6, black pawn 7, black rook 8, etc. The sensors do not distinguish between pieces of a same colour and figure, i.e. all white pawns have the same code 1, both black knights are code 9, etc. An update message is sent by the board every time a sensor notices a change, i.e. when a piece is removed or added. The update message contains two important bytes; one is the sensor cell code, and the other the new state of the cell (empty, or occupied by piece). For a complete simple movement (with no capture involved) two update messages are sent: one when the piece is removed, and one when the piece is placed on the new cell. For movements where capture is involved, the number of update messages may vary from two to five, depending on the capturing style of the player. When two pieces are on the same cell, it may also happen that no message is sent when one of them is removed. Finally, the actual movement can be calculated by analysis of all the related update messages.

One challenge when analysing update messages is that we do not have clear information of when the movement is finished in order to start analysing the messages, and count the next messages as part of the next movement. For the case of a simple non-capture movement, it is very easy to know when the movement is finished. However, when a capture is involved, each new message has to be checked in order to know if the movement is complete or not. This may be problematic since each movement depends on how well the update messages belonging to the former movement are classified. For instance, if an error happened in the analysis of the messages from a given movement, some of the messages may be wrongly considered as part of the next movement, creating more and more errors to the following movements. To avoid possible 'chain-errors', we double check using also state messages. After each movement is considered finished (by checking each update message), we ask and receive a state message. By comparing the state message with the one before the movement happened, we can calculate the movement. We then compare this movement with the one resulting from analysing the movement's messages. If both movements (from the state messages and from the updates messages) coincide, then we consider that everything is running smoothly. In case of a mismatch, we execute a refreshing, which consists on deleting all possible update messages in the queue so that the next movement is not compromised, update the position of the pieces using the last state message, and restore the maximum number of pieces from the chunks as possible. Restoring the pieces from chunks is necessary to not lose track of them. Because of the ambivalence of piece codes, we cannot know for certain where all chunk pieces are placed after a wrongly interpreted movement. At the moment we implemented the simplest mechanism which is to restore all pieces belonging to chunks that have not changed position after the misinterpreted movement. However, this is far from ideal, and as future improvements, another method should be implemented. One possible simple improvement is to check if the piece code is unique, for instance, when only one non-pawn figure is remaining on the board. In any case, each update message is checked carefully, so that all the common possibilities are taken into account, and the movement annotation is practically free from error. Errors of the type explained in this paragraph may only occur when the player moves pieces incorrectly, such as removing pieces without capturing.

The output of the automatic game annotation can be seen in *Fig. 3a*. The game data is saved in text format, indicating the chunk positions, and the complete configuration of the board in FEN-like (Forsyth–Edwards Notation) [16] format at each time a movement happens. The FEN notation consists of two parts. The first part encodes the empty squares and the position of all pieces on the board from white's perspective line by line just like a matrix, starting with square a8 and ending with h1, two lines always separated via forward slash. The second part follows after a space character and gives information about who is to move next, who is still allowed to castle, the number of played moves since the last piece has been captured or the last move of a pawn, respectively. We only use the first part of the FEN code because we are interested in the complete configuration of the board at each step of the game play. E.g. *Fig. 3a* documents 7 steps within a task handling, each step represented by a block of 4 rows. In the last block the following information is given: At the time 650s and after the last move Rb7-d7, player 1 chunks the pieces on the squares b2, f1 and d7, player 2 chunks those on c4, d2, h8 and b5. The complete board configuration is denoted by 7R/3r1P1P/6P1/1P6/2Q5/5pp1/1q1R3p/1k1k1b2 which is the position displayed in *Fig. 3b* (line by line, “7R” means “8th row: a to g empty and black rook on h”; “3r1P1P” means “7th row: a to c empty, white rook on d, e empty, black pawn on f, g empty, black pawn on h”). The single chess pieces of the chunks can be selected manually before starting the game, or automatically using a database.

4. CHESSBOARD DETECTION AND STABILIZATION WITH MARKERS

For analyzing the eye fixations on the chessboard, we use computer vision algorithms to automatically detect the chess fields from the video stream (960x720 pixel, 30 fps) provided by the scene camera located in the center of the eye tracking glasses (see *Fig. 2, 5* and *6*). Here the stabilization of the chessboard is important because of chess players’ head, upper body and hand movements during gameplay. These movements cause changes in view angle and perspective towards the chessboard or they eclipse parts of the board. Furthermore, existing chessboard detection algorithms are not feasible with real chess pieces. The pieces can partly overlap due to perspective. In order to handle these challenges, we employ ArUco markers (see *Fig. 5* and *6*) to extract the chessboard region and to divide this region equally into 64 sub-regions to get the coordinates for the single chess fields in scene video frames provided by the eye tracker. These sub-regions are assigned with the corresponding chess labels. The calculated information allows then for a precisely timed assignment of the recorded eye fixations to the corresponding chess fields. ArUco markers [17,18] are synthetic square markers composed by a wide black border and an inner binary matrix which determines its identifier (id). The size of the internal matrix depends on the marker size, e.g. 16 bits for a 4x4 ArUco marker. The black border facilitates its fast detection in the image and the binary codification allows its identification and the application of error detection and correction techniques.

We use the OpenCV [19] library (version 3.2.0) for processing the scene video frames and ArUco marker(s) detection library for locating the markers in the 2D scene video recorded by the eye tracker (see *Fig. 5* and *6*). Both libraries were developed using the C++ programming language. Initially, we randomly created four different default ArUco markers [20] with a marker size of ~35 mm and a padding of ~10 mm and placed them at the corners of the chessboard (*Fig. 5* and *6*). Employing a large padding is an advantage to avoid the overlapping of chess pieces with the markers. After all markers are detected, the markers’ corners are sorted and their identifiers are allocated as left-top (LT), right-top (RT), left-bottom (LB) and right-bottom (RB). The LT and/or RT markers had fluctuations in detection because of noise to decode the marker dictionary pixels.

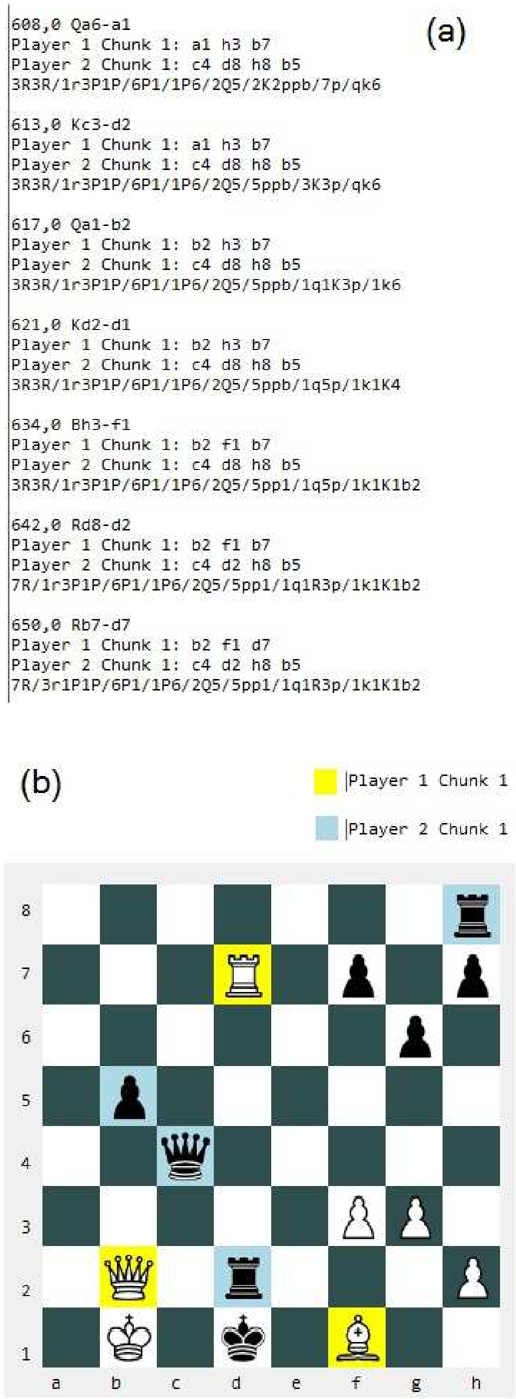


Figure 3. Acquisition of the game data (movements, chunks position and configuration of the board at each time) using the DGT e-board connected to the computer by USB and the software developed by us. In this example, we track two chunks, one for each player (yellow and blue pieces in (b)). (a) The game data is saved in a text file. Every time a player makes a movement, the time in ms with respect to the beginning of the game is written to the file, followed by the movement in algebraic long notation according to the FIDE rules [14], the position of all the chunks tracked (in this case two chunks, one for each player), and the complete configuration of the board in FEN-like notation (for the time range from 608-650s of the game play). The result allows to know the state of each cell (i.e., if it is occupied by a chess piece, by a piece from a chunk, or if it is empty) at all moments during the game. (b) shows the movements in real time as a 2D representation of the real 3D game. Illustrated match: Garry Kasparov vs. Veselin Topalov "Kasparov's Immortal" (1999) game, one of the latest moments before check-mate.

To overcome these instabilities in the marker detection, we estimated the correlation of markers' shift to determine the positions of the missing markers between the previous and the current frame. I.e., whenever the LT and/or the RT marker is/are missing in the current frame then the coordinate shift between the LB and/or RB marker positions in the previous and current frame are determined. This shift is then adapted to the LT and/or RT marker positions in the previous frame.

In the test cases where the eye tracker and four markers (see *Fig. 5 (i)*) are used, the bottom markers are often not seen in the frame due to the distance between the scene camera and the chessboard. Thereupon, two more markers have been added at the mid of left-side and right-side of the chessboard (see *Fig. 5 (ii)*) which alternatively act as LB and RB when the bottom markers could not be detected. This approach keeps the continuity of chessboard detection but only detects one half of the chessboard if bottom markers are missing. Therefore, two more markers have additionally been added (in total 8 markers, see *Fig. 5* lowest picture) between the mid-markers and the bottom markers that overlay most part of the chessboard.

From bottom to top, the LB and RB switch automatically according to the bottom markers' visibility. Therefore, at least four markers should be visible or estimated to extract the chessboard region (fully or partially). When any four markers are detected a 2D trapezium surface of the chessboard, spanned by the extracted marker position in the respective scene video frame, is created (see overlaid red drawing in *Fig. 5* and *6*). The nonlinear trapezoidal is then divided into 8 parts with 9 coordinate points on each side, using the 2D perspective transformation the correlation ratio with respect to the camera projection is achieved. These coordinates are connected to opposite sides by drawing vertical and horizontal lines which construct 8x8 sub-regions inside the trapezium representing the 64 chess fields. By finding the intersection of each vertical and horizontal line, the sorted boundaries of sub-regions (i.e. 81 coordinate points) are assigned to the chess field labels.

By increasing the number of markers, the error rates in chessboard detection significantly decreased which results in reliable automatic annotations of gaze positions on chess fields. In case of a partially detected chessboard due to the invisibility of bottom markers in the frame and/or gaze coordinates outside of the chessboard boundary, frames have to be manually post-annotated. In order to find these frames, we implemented a binary traffic light feature (see *Fig. 4*).

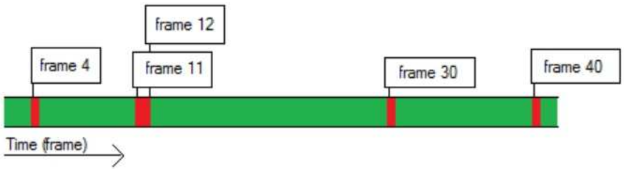
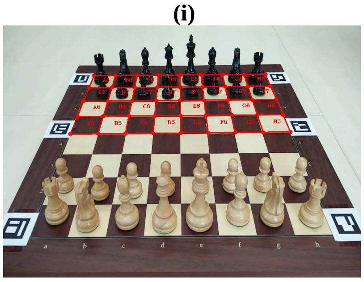


Figure 4. The traffic light shows the single frames of the scene video along the time axis. Each frame is represented as a small and vertical line. If the automatic annotation for a frame was successful, the line is drawn in green, otherwise in red (i.e. when the chessboard could not be automatically detected or if the fixation is not on the board). In this example, only frames 4, 11, 12, 30 and 40 could not be processed automatically and still have to be manually post-annotated.

On a timeline, all frames are represented by small and short vertical lines. If the automatic detection provides good results, the corresponding vertical line is drawn in green color. Frames, where a later manual annotation is necessary because the chess fields could not be detected or the player did not fixate the chessboard, are represented by red vertical lines in the traffic light.

(i)



(ii)

Figure 5. Usage of: (i) four markers; (ii) six markers – upper half of the chessboard extracted.

(i)

(ii)

(iii)

Figure 6. Usage of eight markers. (i) Full chessboard extracted; (ii) upper three-fourths of the chessboard extracted; (iii) upper half chessboard extracted.

5. PRELIMINARY RESULTS

In this section we present some examples of how we use automatic chunk tracking combined with eye tracking to generate plots of visual attention on chunks (*Fig. 6d* and *Fig. 9e, f*), as well as attentional landscapes over the whole board (*Fig. 8b, c* and *Fig. 9c, d*). These two types of data visualisations are useful to analyse the behaviour of different players for the same task (*Fig. 8*), or one single player during different tasks (*Fig. 9*).

The plots of visual attention on chunks (*Fig. 8d* and *Fig. 9c, d*) show the amount of time a player spent on 'fixating' pieces from the chunk over a given time interval. A 'fixation' is here defined as looking at a same position for a period of time equal or longer than a chosen threshold T_{min} (in seconds or frames). The plots can be adapted to show only the data for a selected time interval from the total duration of the game, by specifying a starting and end time value 't0' and 't1' (in seconds or frames), respectively (see *Fig. 8a*). For a set of different thresholds for T_{min} (e.g. from 2-5s on the x-axis in *Fig. 8d*), the percentage of attention on the chunk (shown on the y-axis) is calculated by adding all the fixations on pieces from the chunk during the selected time interval [t0, t1]. The result is the cumulative fixation time (in seconds or frames) which is then divided by the total time duration of the chosen interval. Note, that in this case, and just for the examples presented here, we do not differentiate between fixations on the single chess pieces of a selected chunk. Rather, we increase the cumulative fixation duration for a selected chunk as a whole whenever the player fixates on one of the chess pieces belonging to the respective chunk. In future work we will extend this feature to be able to calculate the cumulative fixation duration for each piece of a chunk separately.

A further type of analysis we present are the attentional landscapes [21] using a Gaussian distribution of standard deviation σ equal to the length of 1 cell of the chessboard, i.e. σ=5.5cm [13] *(Fig. 8b, c* and *Fig. 9e, f*). For the settings for the colour mapping in the attentional landscapes we follow the recommendations in [22, p. 238] and choose in a first estimate a value of around 4° for the visual angle (diameter), which gives an indication of what is looked at with the fovea, i.e. the area of the retina with the highest visual acuity, and also includes the closest area of the peripheral field. As described in [22, p.238] this chosen value is a first estimation, because items can still be perceived in the further periphery (therefore in the work of [23] the perceptual span is incorporated into the attention map settings). In our settings, the distance between the player's head and the chessboard plane is about 50cm, and the separation between the player and the position fixated on the board ranges from 20 to 64cm, depending on the fixated cell. From these three values and simple trigonometry, we can calculate the corresponding region on the board falling on the fovea and nearest periphery to 4.7cm for the closest cells on the board and 7.2cm for the furthest ones (see *Fig. 7*). For simplicity, and since the value is close to the length of one cell of the board (5.5cm), we approximate σ to the cell length 5.5cm for the examples herein. By using σ equal to one cell length, we can plot the attentional landscape on the 2D virtual board by simply discretizing the 2D representation of the chess board (see *Fig. 8a*) in 64 cells. *Fig. 7* illustrates the calculation of the field of view in cm on the board, with α equal to the visual angle of 4° and H equal to the distance of the player's head to the board plane (in cm). The attentional landscape can be twitched to show only the result during a defined range of time, and for fixations equal or longer than T_{min}.

After choosing values for t0, t1, and T_{min}, the attentional landscape is then calculated by adding all fixations in the selected time interval [t0, t1] of at least T_{min} length as a Gaussian of standard deviation equal to 1 cell, centred at the fixation location (provided by the eye tracker in x-y coordinates of the scene video). Finally, the total landscape for each cell is normalized so that the sum of all 64 cells equals to 1, or 100 for percentages.

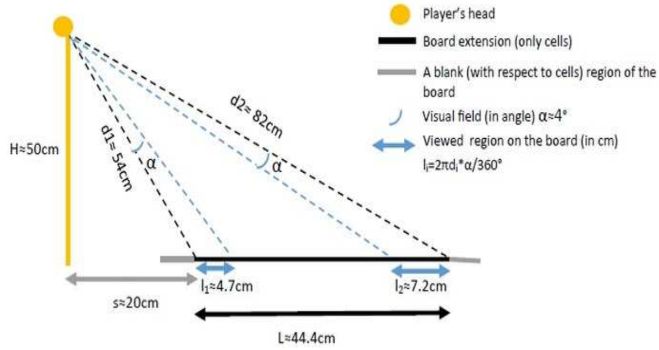


Figure 7. Sketch for the calculation of the field of view in cm on the board (l_1 and l_2) from the field of view in angle (α), the distance of the player's head to the board plane (H) and the distance between the player and the fixated cell on the board (variable from 10cm to 54.4cm).

After choosing values for t0, t1, and T_{min}, the attentional landscape is then calculated by adding all fixations in the selected time interval [t0, t1] of at least T_{min} length as a Gaussian of standard deviation equal to 1 cell, centred at the fixation location (provided by the eye tracker in x-y coordinates of the scene video). Finally, the total landscape for each cell is normalized so that the sum of all 64 cells equals to 1, or 100 for percentages.

As a pre-test to evaluate whether the results on both type of plots described above reproduce the expected values, we created a well-defined dataset of eye tracking data to emulate two kinds of different chess behaviour for the same task (*Fig. 8*). The details about the task are not relevant, only how the emulated dataset for the scenario is created, which is as follows:

- For 154 seconds the configuration of the board is the one in *Fig. 8a*, the pieces in red are the chunk.
- The eye gaze dataset for player 1 is created using a uniform distribution between 1 and 64, so that each chess field is chosen with equal probability.
- On the other hand, the eye gaze dataset for player 2 is created as follows: at each frame the player chooses with a probability of ½ a cell occupied by the chunk, and with probability ½ to all 64 cells with equal probability.

Fig. 8b and *8c* show the *attentional landscapes* for each player using a Gaussian distribution of standard deviation equal to 1 cell, during 154 frames and for T_{min}=1 frame (*Fig. 8b*), as well as for T_{min} =3 frames (*Fig. 8c*). From the figures we can see that the attentional landscapes reflect the expectations and illustrate effectively the outcomes. Because the fixation data of player 1 was equally distributed over all chess fields, the corresponding landscape show no focused attention on any specific chess piece (*Fig. 8b*). Player 2, in contrast, for whom fixations on the chunk have a higher probability in the well-defined dataset than for the rest of the chess field, focuses on the chunk, illustrated by higher values in the upper left corner of the corresponding attentional landscape (see *Fig. 8b* right). *Fig. 8d* shows the percentage of the player's visual attention focussed on the chunk plotted versus T_{min} (from 2-5 frames). As expected, the proportion on attention on chunks for player 1 is close to 0, i.e. $(1/64)^n$ for T_{min}=n frames. On the contrary, for player 2, the proportion for T_{min}=n frames is $(1/2 + 2/64)^n$, as shown in *Fig. 8d* for T_{min}=2, 3, 4, and 5 frames.

As a second pre-test (with more realistic eye tracking data), we use the data from one participant (randomly chosen) from a series of real-time chess experiments we conducted. In this experiment (conducted with 8 participants) the chess player had to solve a set of fixed classical chess tasks such as "White has to move, mate in 3 moves." within a set of chess constellations. All players had to deal with the same set of tasks under time pressure (2 minutes time for each task). Their gaze behaviour was recorded, starting from the first look at each task and ending either as soon as they presented a correct solution or when the handling time expired.

For this example, we show the result for one task the player could solve ('task 1') *Fig. 9a, c, e*), or one he could not solve ('task 2') (*Fig. 9b, d, f*), within the time limit of two minutes. The configuration of the pieces for 'task 1' and 'task 2' are shown in *Fig. 9a* and *9b*, respectively. The pieces in yellow are the chunks for each of the tasks. In this experiment we used our software to automatically track the chunks' positions (see Section 3), and manually annotated the gaze video provided by the eye tracker. For the manual annotation of the gaze video we used common media player software (VLC player; www.vlc.de) and protocolled every ~100 ms at which cell the player was looking at. In this first approach we chose a threshold of ~100ms because the minimum processing duration during a fixation is 100-150ms [24]. In this way we can ensure that we get all fixations and could do a relatively fast manual annotation for a coarse data analysis. In future work we will look closer into the fixation duration distribution in the recorded data files and select the threshold accordingly, because the mean fixation durations in chess are usually longer and depend on expertise, task conditions and chess constellations. For example, in [8] mean fixation durations in a chess detection task are reported from 230-330ms depending on chess skill and condition. The conditions consisted of 'no-cued' trials with two attackers and two conditions in which a cued non-checking attacker appeared together with an attacker that was either congruent (i.e. non-checking) or incongruent (i.e. checking). From the results in *Fig. 9c* and 9*d*, it can be seen that the plots of attention on chunks show higher percentages for 'task 1' (*Fig. 9c*), the one the player could solve, than for 'task 2' (*Fig. 9d*), in agreement with previous studies of correlation of expertise and visual attention on chunks [4]. On the other hand, the attentional landscapes for T_{min}=1 frame (*Fig. 9e, f*) look quite similar and show some degree of focused attention on chunks for both cases.

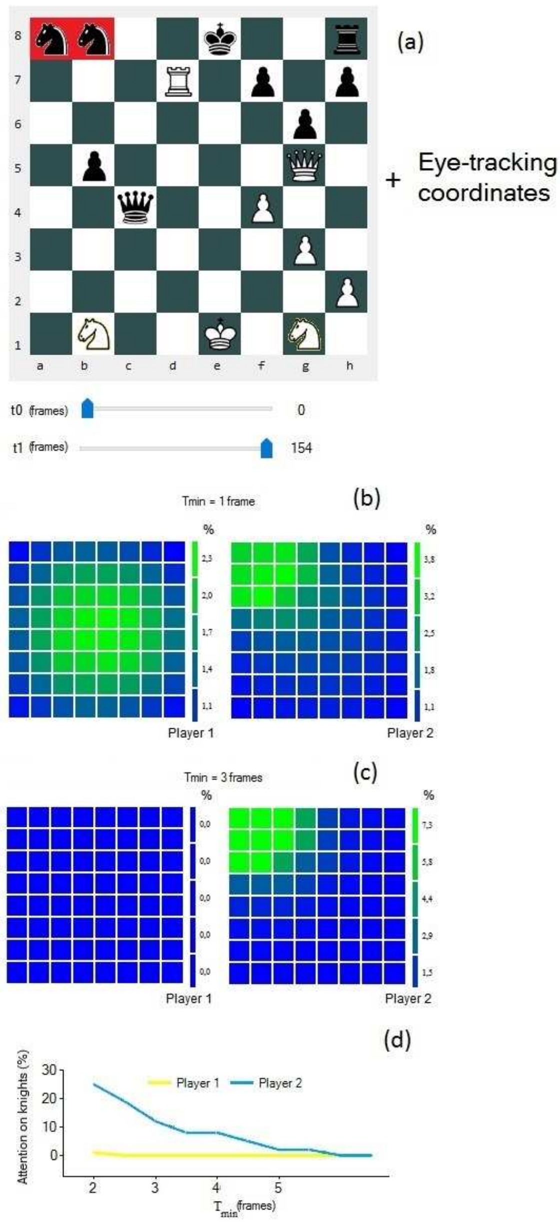


Figure 8. Screenshot of two types of plots for quantifying the visual attention from eye tracking measurements over the chessboard during the game.

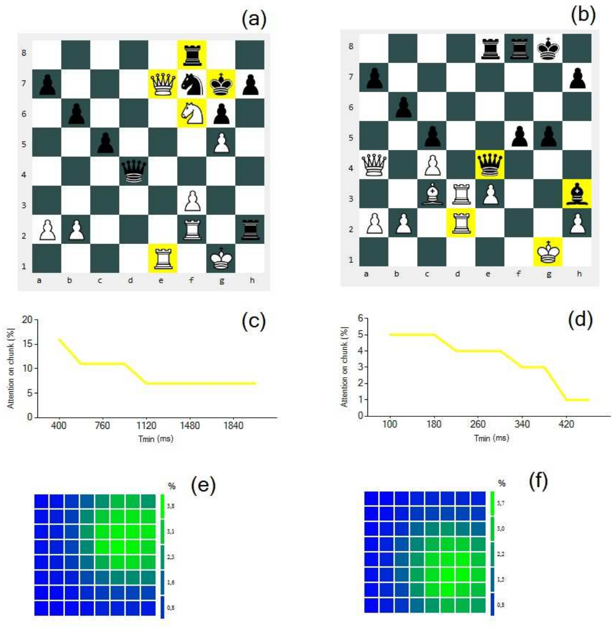


Figure 9. Screenshot of the attentional landscapes and plots of attention on chunks as a function of T_{min}, for a single participant and two different tasks in a real chess game. (a), (c) and (e) correspond to a mate in 3 moves task that the participant could solve after two minutes. On the other hand, (b), (d) and (f) correspond to another mate in 3 moves task that the participant could not solve in the two minutes given to solve it. The attentional landscapes are calculated for T_{min}=1 frame.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have introduced a novel and efficient approach for the automatic analysis of chess players' visual behavior in real-time chess scenarios (solving tasks or playing live blitz chess matches on a real chessboard). We additionally implemented the functionality to select chunks and to analyse players' attention on these chunks over the game play. Although, the system and its evaluation are still in a quite preliminary state, first results show that it works reliable and resembles expected outcomes. Our approach works with a real chessboard and real chess pieces and is able to handle influences by environmental challenges, such as uncontrolled head movements, overlays between chess pieces and chess fields, as well as changing lighting conditions. By preserving the detected chessboard grid size in each frame in a future step we can further enhance the accuracy of our solution and reduce its' error rate. This is in stark contrast compared to existing pure Computer Vision based approaches (such as described in [25]) that do not provide reliable automatic tracking and analysis results because they are not able to handle adequately the above mentioned environmental challenges occurring in real chess game play. Furthermore, due to the use of complicated Computer Vision algorithms they need much more processing time and computer power than our approach. With the proposed solution it will be possible to analyze the behavior of different players for the same chess task or of one single player for different tasks in real chess games without the need for a cumbersome and error-prone manual data annotation. By introducing the traffic light feature, the user is directly pointed to those frames, where the automatic annotation did not work sufficiently (e.g. when several markers cannot been detected in the scene video or when gaze coordinates are outside of the chessboard coordinates). Only for those frames the data still has to be manually post-annotated.

As mentioned in section 2, the gaze coordinates and the e-board data are post-processed and analysed after the game play. We intend to go from offline to online processing and therefore to refine our techniques described in sections 2 to 4 in order to perform the data analysis and the data acquisition part both in parallel. For this online automatic analysis of game play the algorithms for e-board handling (section 3), chessboard detection and eye gaze analysis (section 4) have to be combined into a single program in the future. From both eye and chunk tracking the correlations between visual attention and chunk positions can then be identified and extracted on the fly. In future work, we will evaluate and optimize our solution with more data from real chess studies and compare our results to the ones using Computer Vision algorithms for automatic chessboard detection (see [25] as an example). By deploying the 3D mapping and localization of markers with respect to the scene camera we might tackle the head and body movements (right and left) issues (see [26]). In fact, the synchronization of all-in-one software will provide an online feedback with less post-processing effort.
On the one hand, our refined techniques will have practical applications in the setting of natural and intuitive electronic chess assistant systems for an individualized training. These assistant systems will be able to identify the expertise and the weaknesses of chess players and guide them through individual training sessions. These sessions could also include virtual chess agents giving feedback about optimal moves in given chess situations. Both experts and novices will profit from these individual feedbacks. On the other hand, the techniques described can offer deeper insight into the chunking processes of chess players. In future work, this may contribute interesting data and findings to the controversy about chunking and templates in the theory of chess [7].

7. ACKNOWLEDGEMENTS

This research/work is funded by the German Research Foundation (DFG) in the CEEGE (Chess Expertise by Eye Gaze and Emotions) project [27]. Additionally, we thank the DGT (Digital Game Technology) for their advices concerning the handling of the DGT e-board and their technical support.

8. REFERENCES

[1] A. D. de Groot (1946). Het denken van den schaker. Amsterdam: Noord Hollandsche.

[2] A. D. de Groot (1965). Thought and choice in chess. The Hague: Mouton.

[3] W. G. Chase & H. A. Simon in W. G. Chase (Ed.) The mind's eye in chess. Visual Information Processing (pp. 215–281). New York: Academic Press. 1973a.

[4] E.-J. Reingold & H. Sheridan. Eye movements and visual expertise in chess and medicine. In S. P. Liversedge, I.D: Gilchrist, & S. Everling (Eds.), Oxford Handbook on Eye Movements (pp. 528-550). Oxford: Oxford University Press, 2011.

[5] F. Gobet & H. A. Simon. Templates in chess memory: A mechanism for recalling several boards. Cognitive Psychology, 31, 1-40.1996.

[6] W. G. Chase & K. A. Ericsson. Skill and Working Memory. In G. H. Bower (Ed.), The psychology of learning and motivation (Vol. 16, pp. 1-58). New York: Academic Press. 1982.

[7] F. Gobet & G. Clarkson. Chunks in Expert Memory: Evidence for the Magical Number Four ... or is it Two? Memory, Nov. 2004; 12(6): 732-47.

[8] Reingold, E. M., Charness, N., Schultetus, R. S. & Stampe, D. M. Perceptual automaticity in expert chess players: Parallel encoding of chess relations. Psychonomic Bulletin and Review, 8, 504–510, 2001.

[9] A. Kiesel, W. Kunde, C. Pohl, M. P. Berner & J. Hoffmann. Playing Chess Unconsciously. Journal of Experimental Psychology: Learning, Memory and Cognition, Vol. 35, No. 1, 292-298, 2009.

[10] D. H. Holding. The Evolution of Chess Positions. Simulation and Games, 10, 207-221. 1979.

[11] N. Charness, E. M. Reingold, M. Pomplun M & D. M. Stampe. The Perceptual Aspect of Skilled Performance in Chess: Evidence from eye movements Memory & Cognition, 2001, 29 (8), 1146-1152.

[12] H. Sheridan & E. M. Reingold. Chess players' eye movements reveal rapid recognition of complex visual patterns: Evidence from a chess-related visual search task. Journal of Vision March 2017, Vol. 13, 4.

[13] DCT e-board: http://www.digitalgametechnology.com /index.php/products/electronic-boards

[14] FIDE homepage: http://www.fide.com/fide/fide-world-chess-federation.html. In Particular: Handbook, Chapter E. Miscellaneous, 01. Laws of Chess.

[15] Sensomotoric Instruments: https://www.smivision.com/wp-content/uploads/2017/05/smi_prod_ETG_120Hz_asgm.pdf

[16] Forsyth-Edwards Notation: http://kirill-kryukov.com/chess/doc/fen.html

[17] OpenCV 3.1.0, Detection of ArUco markers: http://docs.opencv.org/3.1.0/d5/dae/tutorial_aruco_detectin.html

[18] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas & M. J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. 2014.

[19] G. Bradski & A. Kaehler. Learning OpenCV: Computer vision with the OpenCV library. 2008.

[20] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas & R. Medina-Carnicer. Generation of fiducial marker dictionaries using mixed integer linear programming. – 2016.

[21] M. Pomplun, H. Ritter & B. Velichkovky. Disambiguating complex visual information: Towards communication of personal views of a scene. In: Perception, vol. 25, no. 8, p. 931 – 948, 1996.

[22] K. Holmqvist et al.. Eye Tracking – A comprehensive guide to methods and measures. Oxford: Oxford University Press, 2011.

[23] P. Blignaut (2010). Visual span and other parameters for the generation of heatmaps In Proceedings of the 2010 Symposium on Eye Tracking Research and Applications (pp. 125-128). New York: ACM.

[24] Goldberg, J. H. & Kotval, X. P. Computer interface evaluation using eye movements: Methods and constructs, International Journal of Industrial Ergonomics, 24, 6, 631-645, 1999.

[25] Losing, V., Rottkamp, L., Zeunert, M., & Pfeiffer, T. (2014). Guiding Visual Search Tasks Using Gaze-CContingent Auditory Feedback. *UbiComp'14 Adjunct: The 2014 ACM Conference on Ubiquitous Computing Adjunct Publication*, 1093-1102

[26] R. Muñoz-Salinas, M. J. Marín-Jiménez, E. Yeguas-

Boliva & R. Medina-Carnicer. Mapping and localization from planar markers. CoRR, vol. abs/1606.0, pp. 1–14, 2016.

[27] CEEGE (Chess Expertise by Eye Gaze and Emotions): https://www.uni-bielefeld.de/sport/arbeitsbereiche/ab_ii/research/ceege.html