On High-Precision Chessboard Detection on Static Scene Videos from Mobile Eye-Tracking Devices

Anastasia Afanaseva, Sergey Afonin
Lomonosov Moscow State University
Moscow, Russian Federation
serg@msu.ru, afanaseva.msu@gmail.com

***Abstract*—Eye-tracking analysis require annotation of a scene video by information on the target of the gaze. We develop a technique for automatic high-precision scene annotation for mobile head-mounted eye-trackers. The solution combines computer-vision techniques for scene recognition, 3-D modeling of the recognized objects, and head movement compensation using gyroscope and accelerometer information provided by the eye-tracking device. In this paper we address the problem of recognition of a chessboard, while the approach may be applied to other situations with static scenes.**

I. Introduction

It is widely accepted that eye movements indirectly reflect cognitive processes. Researchers use eye-tracking to study attention, memory, language, problem solving and decision making, e.g. [1], [2], [3]. The most important notions related to human vision are *gaze fixation* and *saccade*. Fixations are gaze maintaining on single object. Saccades are the rapid shifts of eye gaze between fixations. Typical questions related to eye-tracking analysis are: What parts of an image attract attention of the viewer?, Is there any differences between gaze patters of people with and without some brain disorders?

This work is motivated by the research of chess players behavior. It is known that unlike most of chess programs human players do not perform deep search. In a recent article [4] authors provide an example of a simple endgame position with large number of possible variations and show that a human player selects the best move without consideration of all possibilities. Our distant goal is to detect piece trajectories that the player considers while solving a given position. This information, compared with the expected solution to the problem, may shed a light on thinking processes.

Analysis of eye movement data requires accurate annotation of the recorded data. It is important to know the exact gaze position at any time during the experiment. For example, if a human watching a painting, then it is required to know what part of a painting he was looking for at a given moment in time. There are several eye-tracking techniques exist [5]. Some techniques assume that participants head does not move. In this case one can get precise gaze positions while the technique itself introduces inconveniences to the participant. Recently, a more natural mobile eye-tracking systems were designed. One such example is Tobii Pro Glasses 2 [**?**]. This is a wearable eye-tracking device that looks like usual glasses. It contains a *scene camera* that records the scene visible to a participant, or the field of vision, and a number of small infrared cameras located at the internal side of the glass for humans eyes movement detection. Head-mounted mobile eye-trackers can be used to record eyes movements in real settings, such as sporting activities, car driving, etc. On the other hand annotation procedure become more complicated as the scene changes from one shot to another because of head movement.

We use Tobii Pro Glasses for free reasons. First, eye-trackers of such type allow recordings in a most natural environments. Second, this specific model provides all data in an open machine-readable form suitable for subsequent processing. Finally, current versions of glasses record eye positions at frequency 50 or 100Hz. While such frequencies are not sufficient for doing some modern eye-tracking research, like research of reading where 500Hz is the minimal requirement for correct identification of gaze position on relatively small characters, in many real-life applications, including the analysis of chess player behaviour, objects of interest are relatively large, so 50Hz is enough for correct object identification. There is a balance between eye-tracker precision and size. In this research we use more mobile and convenient model.

The eye-tracker provides a video stream from the scene camera and various numerical values recorded with predefined frequency. To be specific, Tobii Pro Glasses we are using in our work generates at a predefined frequency the following information: 2D gaze position (a point on a unit square representing a frame of the scene video), 3D gaze position measured in millimeters, gyroscope and accelerometer data. A typical workflow of (semiautomatic) annotation of eye-tracking data includes the following stages. First, a sequence of *gaze fixations* are identified in the data stream. Fixation is the maintaining the gaze on single object, so it is a relatively stable gaze positions that can be identified automatically. Once fixations are known, the researcher manually process all fixations one by one and *annotate* video stream by descriptions of the gaze locations. For example, if a natural human-human conversation is investigated, then such annotation may include labels like “look at face”, “look at hands”, “look on environment”, etc. Annotated data may be used for further analysis, e.g. statistical analysis.

Video annotation is a time consuming and routine operation. In order to automate this task one should recognize the object of interest, such as hands or body, in a video stream, e.g. [6], [7]. Another issue related to mobile eye-trackers is possible eye movements compensating head or body movements. For example, if a participant rotates his head then his eyes compensate the rotation by movement in opposite direction. It is clear that compensation eye movement breaks fixation into parts. Special technique is needed for distinguish-

ISSN 2305-7254

ing compensation and other types of eyes movements [8].

In some applications it suffice to annotate the video by naming objects, such as "looking at participant 1", or "looking at face". The exact position of the gaze may not be important. The main complexity of this annotation is related to object detection. Special techniques such as [6] or general object detectors [9] may be used to detect objects of interest in the video stream. Other applications, in contrast, require precise location of the gaze. The general purpose detectors do not provide sufficient accuracy. In the problem addressed in this paper it is important to recognize the precise position of the gaze inside chessboard square. While detection of a chessboard pattern seems simple problem (this is a usual task of camera calibration procedure), the well known techniques are not as precise as required in real-life settings. Calibration procedures assumes that the "chessboard" is empty and high contrast [10], [11]. The actual chessboard does not satisfy these assumptions. Dark and clear squares are not of black and white color. In addition, some parts of the board are covered by pieces that are of the same colors as squares, i.e. color of white pieces are very similar to the color used for white squares. In such conditions it is more difficult to find regular patterns of corners using color properties. It is worth noticing that presentation form of a position affects humans perception. If typical pieces images are replaced by corresponding letters (K for king, N for knight, and so on) humans spend more time to recognize the position [12]. Thus, all experiments should be performed in an as much natural environments as possible.

The layout of the paper is the following. In the next section we provide technical description of available data, assumptions about scene properties and camera movements, and formulate the problem. Section III, which is the main part of the paper, contains description of chessboard recognition procedure, including standalone image processing and processing of a video stream. Experimental results on chessboard recognition are presented in Section IV. We conclude the paper by a brief discussion on possible directions for future work.

II. Problem statement

A. Description of available data

Tobii Pro Glasses is a mobile eye-tracking device. It contains one scene camera and writes down JSON file eye- and head-movement related data.

The scene camera is a Full HD (1920x1080) camera generating 25 fps video stream encoded using H.264 codec with keyframes at every 16 frames. The video stream is transmitted as MPEG-ts packets. Visual angles of the scene camera are $82°$ horizontal, and $52°$ vertical. Other parameters are not described in the specification of the glasses. An example of video frame is presented in Fig. 1.

All eye- and head-movement data are timestamped with an internal clock with a microsecond precision. Eye-tracking data are related to the glasses coordinate system. The origin is located at the scene camera, with x-, y-, and z- axes going left, up, and forward, respectively. The following data are recorded.

1) 2D gaze position. Two real values varying from 0 to 1 representing position of the gaze related to left-upper corner of the frame. Point $(0, 0)$ is the left upper corner, and $(1, 1)$ is the right-bottom one.
2) 3D gaze direction. A unit vector in the glasses coordinate system representing gaze direction.
3) Gaze position 3D. The 3D position, in mm, relative to the scene camera where the gaze is focused.

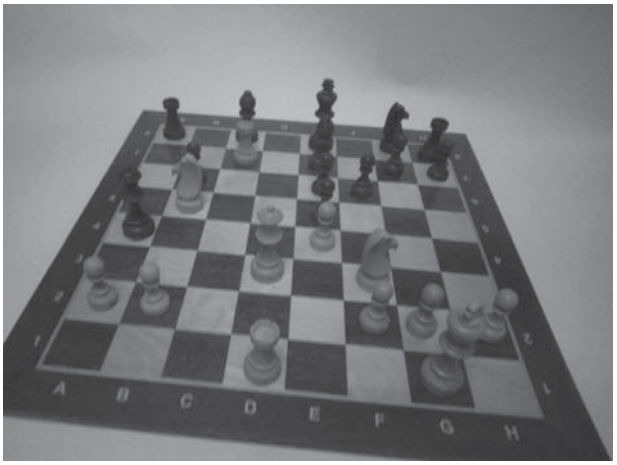

Fig. 1. Original image (cropped to chessboard area)

Gaze data are recorded at 50Hz frequency. Some measurement may be omitted in the data file due to impossibility of eyes recognition, e.g., closed eyes, or other issues. Such records are clearly marked in the JSON file by designated *status* property.

Head-movement data include information from glasses MEMS accelerometer and gyroscope.

1) Gyroscope data indicate rotation of the glasses, measured in degrees per second ($°/s$).
2) Accelerometer data indicates movement of the glasses. The accelerometer data has the unit meter per second squared.

Gyroscope and accelerometer data are recorded at higher frequency.

Synchronization between video (90kHz pts labels of MPEG-ts) and data streams is managed by pts-synchronization records emitted by the glasses into the data stream.

B. Problem statement

Given a sequence of chessboard images (Fig. 1) and the data described in the previous section one should generate a sequence of chessboard squares the participant gaze was focused. At the moment we do not distinguish gazes focused on a square from the ones focused on pieces.

We assume that:

- the chessboard and pieces on it are static;
- participants head and body move arbitrary;
- participant can not change his position, e.g. moves himself to opponents place;

- on some frames the board may not be presented entirely;
- geometric properties of the board and pieces (such as height) are known.

III. Chessboard recognition

Our recognition procedure consists of two components. First, we use still image processing technique to identify area on the frame with an image of the chessboard. Then, using data about 3D-gaze positions we reconstruct chessboard location. Subsequent frames are processed with hints. All algorithms are implemented using Open CV2 library for Python.

A. Processing of standalone image

Processing of standalone images, similar to one presented in Fig 1, is performed in three stages:

- detection of edges on the image using Canny Edge Detector [13];
- detection of representative lines, that presumably contains the lines constituting the chessboard;
- evaluate every possible transformation from a "unit square chessboard" onto the image of the chessboard.

Edge detection is performed using Canny Edge Detector followed by Hough Transform for lines detection. CV2 function for edge detector depends on two threshold parameters for the hysteresis procedure. We use adaptive parameters

$$\begin{aligned} p_1 &= \max\{0, (1-\sigma)\cdot m\} \\ p_2 &= \min\{255, (1+\sigma)\cdot m\} \end{aligned} \quad (1)$$

where $\sigma = 0.33$ and m is the median value of the image. Hough transform uses predefined parameters $n = 125$ as a number of required votes. This value is related to the dimensions of the image and the area covered by the chessboard. Larger values cause lines misses, lower values leads to messy detection.

The result of line detection is presented in Fig. 2.

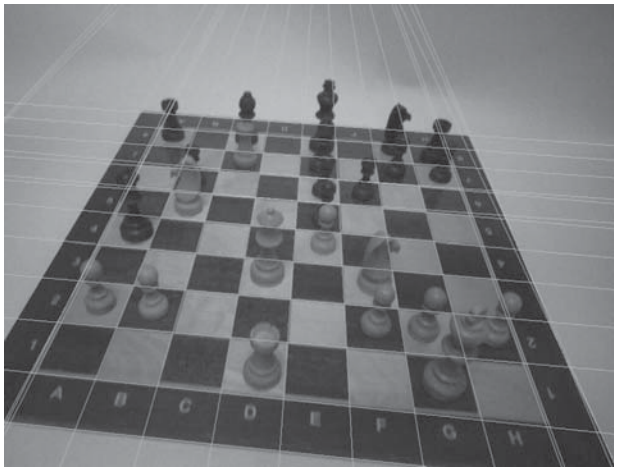

Fig. 2. Result of Canny Edge Detector followed by Hough transform

Three points should be mentioned. First, there are many lines going close to a vertical or horizontal border of the board. For example, there are several lines to the left of file "a". Second, board physical edges generate corresponding lines. Finally, some lines between squares may be missed.

To cope with multiple detection of the same line, like the lines between firth and sixth ranks on Fig. 2, we split all the lines into vertical and horizontal and use one-dimensional clustering approach to find a representative line in each group of "close enough" lines. As a result of this step we have a grid of intersecting lines. We shall call these lines *vertical* and *horizontal*. For two line l_1 and l_2 of the same type we write $l_1 \prec l_2$ to denote that l_1 precedes l_2. For vertical lines this relation means that l_1 is on the left, and for vertical lines it means that l_1 is higher than l_2.

As physical parameters of the actual chessboard are known it is possible to define a model of the board. Usually, the board has a square form, so the *ideal board* is just a unit square divided into 64 parts of equal size. The goal is to find a perspective transformation between the ideal board and an area of the frame that contains chessboard. This final stage of board recognition algorithm is a brute force search of the best possible mapping of recognized grid corners onto the unit square board.

Every four lines recognized on the frame, two horizontal and two vertical, defines four points. Mapping between corners of the unit board and four points at the frame defines a perspective transformation matrix M, provided that no three points lay on a line. Matrix M and the inverse matrix M^{-1} are used to compute coordinates of a point on a frame for a given point on the unit board, and vie versa. In order to describe searching procedure we have to define three sets of points. First set is the set of points found as intersection of detected vertical and horizontal lines. We shall refer to it as *detected corners* (of chessboard squares). These points are defined in coordinate system of the frame. The second set is the set of *ideal corners*, i.e. corners of the ideal board. There are 91 ideal corners. The last set consists of images of detected corners under frame-to-unit transformation. This set will be referred as *idealized corners*.

Given four detected corners there are three possibilities exist:

- these points constitutes the corners of the chessboard;
- these points constitutes the corners of some area on the board (in whole number of squares);
- none of the above.

For a 8×8 board there exist $8^2 \times 7^2$ variations for first two possibilities, starting from 8×8 area and ending with 64 variations of 1×1 areas. Searching algorithm sequentially guesses all variations of mapping and evaluates *mapping quality*. If four detected corners were selected properly, i.e. we have one of the two cases above, then images of intersection points will match intersections of unit board lines (Alg. 1).

Matching quality depends on coordinates of detected corners G and transformation matrix M. Let us note that grid G, which represents intersections of the *detected* lines, do not necessarily contains all the lines. For example, if border lines

Algorithm 1 Find for best transformation matrix

Require: V, H – sets of vertical and horizontal lines
1: {Let v-vars denotes vertical lines, h – horizontal lines}
2: **for** all v_1, v_2, h_1, h_2 such that $v_1 \prec v_2$ and $h_1 \prec h_2$ **do**
3: Construct grid G of intersection points of all lines h and v such that $h_1 \preccurlyeq h \preccurlyeq h_2$ and $v_1 \preccurlyeq v \preccurlyeq v_2$
4: Let (lt, rt, lb, rb) be corners of G
5: **for** all pairs $il, it \in \{0, \ldots, 8\}$ **do** {ideal left-top}
6: **for** all dx, dy such that $il + dx \leqslant 8, it + dy \leqslant 8$ **do**
7: Compute transformation matrix M induced by mapping $[lt, rt, lb, rb] \mapsto [(il, it), (il + dx, it), (il, it + dy), (il + dx, it + dy)]$
8: Evaluate matching quality $MQ(G, M)$
9: **end for**
10: **end for**
11: **end for**
12: **return** $M, MQ(G, M)$ for matrix M maximizing MQ value computed in line 8

v_1, v_2, h_1, h_2 selected in line 2 of Alg. 1 correspond to outer borders of the chessboard, the size of grid G may vary from 2×2 (only border lines are recognized on the frame) to 9×9 (all lines are recognized), or even larger, if some extra lines were erroneously emitted by Hough Transformation detector. Any intermediate shapes, e.g., 5×7, are possible as well, because vertical and horizontal lines may be omitted independently. Thus, lines of grid G should be *aligned* with lines on the ideal board. A row in G is a sequence of 2-d points on frame. After applying perspective transformation M these points lay on a horizontal line, provided that this grid line corresponds to a line on the chessboard. Hence, to make an alignment it suffice to calculate average value of y- coordinates of all points on idealized grid row and find the closest line of the ideal board. If more than one grid line correspond to a single line on the ideal board, then there are exist more than one possible alignment that should be evaluated independently.

The alignment problem may be formulated as follows. Given two increasing sequences $A = a_1 \ldots a_n$ and $B = b_1 \ldots b_m$ of real numbers and a threshold parameter γ find the largest l such that there exist subsequences $a_{i_1}, \ldots a_{i_l}$ and $b_{j_1}, \ldots b_{j_l}$ of length l satisfying

$$\sum_{k=0}^{l} |a_{i_k} - b_{j_k}|^2 < \gamma.$$

As in our specific problem all sequences are limited in length by a small number this optimization problem may be solved using a straight forward brute force algorithm in reasonable time.

Quality function MQ finds the best alignment between grid G (rows and columns) and the ideal board by applying sequence alignment to rows and columns, and computes numerical value of the alignment quality. This value takes into account distances between ideal and idealized corners and uses penalty for missing lines. Badness, the inverse to matching quality, is defined as

$$B = \alpha \operatorname{median}(dist) + \beta \left(1 - \log_2 \frac{81 + matched}{81}\right), \quad (2)$$

where $dist$ is vector of distances between matched idealized and ideal corners, and $matched$ is the number of matched corners. Parameters α and β allow to select a balance between closeness of detected lines and the number of missing lines. In our experiments we used $\alpha = 10$ and $\beta = 250$. With such parameters Alg. 1 tends to select areas with relatively large number of internal lines.

B. 3D-Modeling of a chessboard

Board recognition algorithm described in the previous section is not stable. For two consecutive frames it can generate quite different results. There are several reasons for such behavior.

- Parameters of Hough Transform may not be suitable for all situations. If position of scene camera changes some lines might become hidden in part by pieces. As a result, not all lines are recognized.
- Scene camera compresses video stream using 16-frames blocks. Reconstruction of frame images may affects quality of recognition.
- Due to perspective transformation lines of the 8-th rank are too close to the lines induced by boards edge. If the board is matched using an 8×8 pattern with correct top line, and with similar pattern with the top line replaced by a board edge line, the quality of these two matchings might be similar.

A high precision may be achieved by utilizing information computed on the previous frames. If the board was recognized with high quality value on one of previous frames, then one may suggests that the location of the board does not change drastically and transformation matrix of the previous frame will be a good starting point for the current frame as well. When participants head moved rapidly, suggested corners might be inaccurate (see Fig. 3).

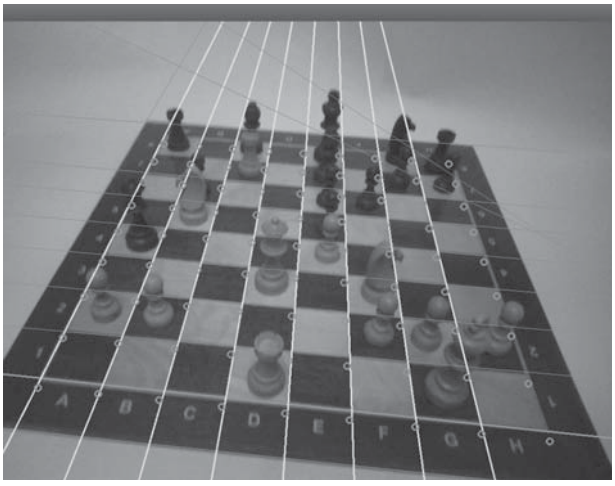

Fig. 3. Hough transform result with hinted corners positions

Accelerometer and gyroscope data provides necessarily information required for head movement tracking. Assume that frames F_1 and F_2 were recorded at time t_1 and t_2, respectively. Given gyroscope and accelerometer samples recorded between

t_1 and t_2 it is possible to compute total rotation and movement of the glasses occurred in that period of time. If 3D position of the chessboard, e.g., in glasses coordinate system, was known at t_1 then it may be computed for the t_2.

Given 3D coordinates of the chessboard one can compute its perspective projection on the image frame using formula:

$$\mathbf{x} = K \cdot [R\, T] \cdot \mathbf{W}, \tag{3}$$

where K is a 3×3 matrix of camera *intrinsic parameters*, R is a 3×3 *rotation matrix*, T is a 3×1 translation vector, $\mathbf{W}$ and $\mathbf{x}$ are homogeneous coordinates of a 3D (border corner) and 2D (its image) points, respectively.

Three dimensional position of the chessboard may be reconstructed using information on gaze positions. Eye-tracker provides both two- and three-dimensional gaze positions. For a single frame that was recognized by image processing detector with sufficiently high quality, two-dimensional gaze position gives the exact location on the chessboard where the gaze was focused. Three-dimensional gaze point gives the distance, in millimeters, between eye-tracker scene camera and gaze position. Taking three such frames one can determine parameters of the chessboard plane, as well as coordinates of chessboard outer corners.

C. Combined chessboard detector

The resulting detector essentially constructs a number of transformation matrices for the current frame and selects the best matrix. The list of candidates includes a matrix computed by stand-alone image processing, the matrix predicted by 3D modeling from the previous frame, and the best matrices from the previous and successive frames. In order to access information from the upcoming frames the processing is delayed for a given number N of frames.

Algorithm 2 Processing of a video stream

```
i = 0 {frame number}
Initialize a circular queue framesQueue of size N > 1
while next frame exists do
  if framesQueue is full then
    yield framesQueue.pop()
  end if
  i = i + 1
  Detect V and H lines on frame using Hough Transform
  Compute grid G_i of V and H intersections
  Find transformation matrix M_i using Alg. 1
  if i > 1 and MQ(G, M_{i-1}) > MQ(G, M_i) then
    M_i = M_{i-1}
  else
    {Propagate "good" matrix backward to the left}
    pframe = last in framesQueue, k = i - 1
    repeat
      if MQ(G_k, M_{k+1}) > MQ(G_k, M_k) then
        M_k = M_{k+1}
      end if
      k = k - 1, pframe = previous in framesQueue
    until not pframe = first in framesQueue
  end if
end while
```

The detector generates sequence of transformation matrices, one per each frame. Two-dimensional positions of the gaze point, given in frame coordinate system, is transformed into ideal board coordinate system using this matrix. If the image point lies inside the unit square, i.e. the gaze was focused on the board, then corresponding square number is emitted into output stream. This sequence of time-labeled squares is a subject of further analysis that is not a part of current paper.

IV. Validation of chessboard detector

Statistical validation of the detector requires large amount of manually annotated data. As this data is not available we use indirect validation procedure. First level of validation is an "expert evaluation", i.e. manual evaluation of recognition quality. The second approach relies on the notion of heat maps. A participant was asked to find a solution to the given position. Video recording is processed by the described detection algorithm. The resulting sequence of squares are used for construction of a heat map – histogram of gaze attractions for each square on the board.

Fig. 4. Testing position for chessboard detection

The correct solution to every testing position is known and validation consists of comparing heat map with the solution. For example, correct solution (presented in a reduced form) to the position presented in Fig. 5 is as follows.

1. Qd3-g3 f6-e5 **2.** Qd3-g7 Rh8-f8 **3.** Rc1-c7 Qb6-c7 [3. ... Qb6-d6 4. Rc7-b7 d4-d3 5. Rb7-a7 Qd6-d8 6. Qg7-h7 d3-d2 7. Qh5] **4.** Qg7-c7 Bb7-d5 **5.** Qg7-e5 d4-d3 **6.** Qe5-e3

Heat map shows reasonable distribution of number of gazes, which indirectly supports the correctness of chessboard recognition.

Most of the gazes were attracted by squares f6 and f7 (red and blue colors of the heat map, representing the most and the least frequently considered squares appear the same on black and white figure).

V. Conclusion

In this paper we address the problem of automatic annotation of a video file recorded by a mobile head-mounted eye-tracking device. In our specific problem an object of interest is a chessboard with the given position. The approach relies on image recognition technique and three-dimensional modeling

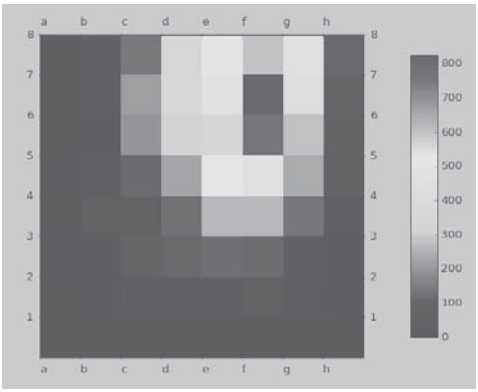

Fig. 5. Heatmap of attention attracting squares

for prediction of chessboard location on subsequent frames. This method is successful due to information about physical distance between eye-tracking glasses and gaze fixation point, and data from MEMS sensors, both provided by the eye-trackers firmware. Preliminary evaluation results indicates, rather indirectly, that recognition procedure gives reasonable results while processing real video recording lasting for several minutes.

Similar problems appear in other eye-tracking experiments. For example, in research on complexity of verbal description of complex objects, such as abstract tangram figures [14], a participant is asked to find a tangram card by its verbal description. During the search phase, the scene may be considered static. Moreover, there is an analogy between square tangram cards and chessboard squares.

Future work splits into two directions. The first one is related to chessboard recognition itself.

- Define more direct evaluation procedure to be used for automatic numerical measurement of entire video stream processing quality. One example of such measure is a "stability" of idealized corners between frames. In case of perfect detection idealized corners should be
- What conditions affects Hough Transformation quality? How recognition quality change in motion JPEG stream will be used instead of H.262?
- Image processing depends on several parameters. Is it possible to find optimal values for these parameters, e.g. using machine learning?

The second area for future research is connected to chess players eye-movement analysis and more accurate identification of squares. In particular, it will be interesting to answer the following questions.

- How to distinguish between gaze fixation on a square and on a piece that partially covers that square. For example, if a physically large piece stands on a square lying between point of view and gaze "target" square, then a small error in two-dimensional gaze position detection causes switching of object of interest between that piece and distant square. Usage of three-dimensional gaze position may helps to solve this problem.
- If a variation suggests that a square, which is occupied by a piece in the initial position, will become empty, will participant fixate his gaze on a piece, or on the square itself?

REFERENCES

[1] A. Glöckner and A.-K. Herbold, "An eye-tracking study on information processing in risky decisions: Evidence for compensatory strategies based on automatic processes," *Journal of Behavioral Decision Making*, vol. 24, no. 1, pp. 71–98, dec 2010. [Online]. Available: https://doi.org/10.1002/bdm.684

[2] B. Pan, H. A. Hembrooke, G. K. Gay, L. A. Granka, M. K. Feusner, and J. K. Newman, "The determinants of web page viewing behavior: An eye-tracking study," in *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*, ser. ETRA '04. New York, NY, USA: ACM, 2004, pp. 147–154. [Online]. Available: http://doi.acm.org/10.1145/968363.968391

[3] K. Rayner, "Eye movements in reading and information processing: 20 years of research." *Psychological bulletin*, vol. 124, no. 3, p. 372, 1998.

[4] I. Bratko, D. Hristova, and M. Guid, "Search versus knowledge in human problem solving: A case study in chess," in *Model-Based Reasoning in Science and Technology*. Springer International Publishing, 2016, pp. 569–583. [Online]. Available: https://doi.org/10.1007/978-3-319-38983-7_32

[5] A. Duchowski, *Eye Tracking Methodology*. Springer London, 2007. [Online]. Available: https://doi.org/10.1007/978-1-84628-609-4

[6] S. D. Beugher, G. Brône, and T. Goedemé, "Semi-automatic hand annotation of egocentric recordings," in *Communications in Computer and Information Science*. Springer International Publishing, 2016, pp. 338–355. [Online]. Available: https://doi.org/10.1007/978-3-319-29971-6_18

[7] K. Essig, D. Dornbusch, D. Prinzhorn, H. Ritter, J. Maycock, and T. Schack, "Automatic analysis of 3d gaze coordinates on scene objects using data from eye-tracking and motion-capture systems," in *Proceedings of the Symposium on Eye Tracking Research and Applications*. ACM, 2012, pp. 37–44.

[8] L. Larsson, A. Schwaller, M. Nystrm, and M. Stridh, "Head movement compensation and multi-modal event detection in eye-tracking data for unconstrained head movements," *Journal of Neuroscience Methods*, vol. 274, pp. 13 – 26, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165027016302023

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[10] A. de la Escalera and J. M. Armingol, "Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration," *Sensors*, vol. 10, no. 3, pp. 2027–2044, mar 2010. [Online]. Available: https://doi.org/10.3390/s100302027

[11] S. Bennett and J. Lasenby, "ChESS–quick and robust detection of chessboard features," *Computer Vision and Image Understanding*, vol. 118, pp. 197–210, 2014.

[12] E. M. Reingold, N. Charness, M. Pomplun, and D. M. Stampe, "Visual span in expert chess players: Evidence from eye movements," *Psychological Science*, vol. 12, no. 1, pp. 48–55, 2001.

[13] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, nov 1986. [Online]. Available: https://doi.org/10.1109/tpami.1986.4767851

[14] O. V. Fedorova, "The contribution of the kinetic component to the multimodal communication, or tangram description strategies revisited," in *Komp'juternaja Lingvistika i Intellektual'nye Tehnologii*, vol. 2. Rossiiskii Gosudarstvennyi Gumanitarnyi Universitet Moscow, 2017, pp. 118–133.