

*Introduction to Computational and
Biological Vision*

Checkerboard Recognition
Final Project



Research Team:

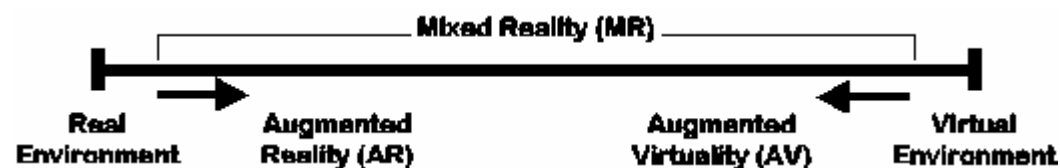
Fledel Yuval

Khait Vitaly

Introduction

Augmented Reality

Augmented Reality (AR) is a growing area in virtual reality research. The world environment around us provides a wealth of information that is difficult to duplicate in a computer. AR systems combine a real scene viewed with virtual objects and artificial intelligence. As a demand to these systems grows so will the need to allow them to coexist and interact with real humans who live in the real world. True interaction between virtual and real humans requires two-way communication. Real people are of course easily made aware of the virtual humans' actions. However, it is much more difficult to endow the virtual humans with perceptive capabilities. Direct transfer of data structures is not an option anymore. Recognition methods are required for virtual humans to perceive real humans' behaviors.



Checkers Example

One of the classical examples in this field is a checkers game between a real and a virtual human, which allows demonstrating the integration of techniques required to achieve a realistic-looking interaction.

The first step performed by the AR system is a perception of real world objects. In this paper we apply different computer vision techniques to perform a real world perception within a checkers game example.

Application environment

- The real human plays checkers using a real board while the virtual human observes him using a static single-color camera
- Since the AR system should be applicable in a real world the number of environment constraints should be decreased as much as possible. We make no assumptions regarding:
 - the colors of the units and the board.
 - the location of the board on the picture
 - the size of the board
 - the intensity and direction of light
- However we enforce some constraints that can be easily satisfied by the real world environment and simplify the perception process drastically:
 - The board is entirely contained in the image
 - The position of the camera is close enough to zenith such that units do not hide others.
 - The pieces are smaller than the cells
 - The background has no checkers board structure and has a relatively small number of lines.
 - Playable cells are the dark ones. (configurable)

Goals and Considerations

Our goal was to reconstruct the position of the units on the board within the environment described in the previous section.

To achieve these goals we have focused on the following computer vision techniques: edge detection, pattern recognition, Hough transform.

Course of Action

The entire processing was divided into a number of consecutive steps:

1. Detection of the board's orientation in the image
2. Detection of the board's location and size
3. Detection of the board parity (should a cells be black or white?)
4. Detection of cells containing white pieces (the pieces whose color differs from the color of cells they hold)
5. Detection of cells containing black pieces (the pieces whose color is same as the color of cells they hold)

To prevent misconception, in this document we'll use a convention that the pieces are placed on the dark cells.

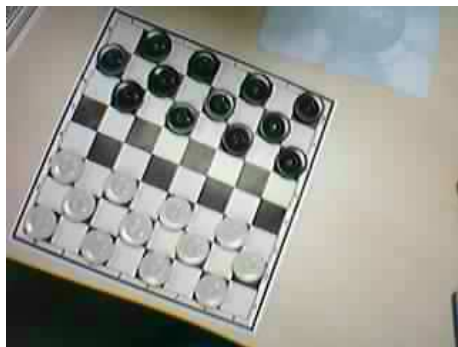
The following section elaborates each step and its implementation.

Implementation

Step 0 – Acquiring an Image

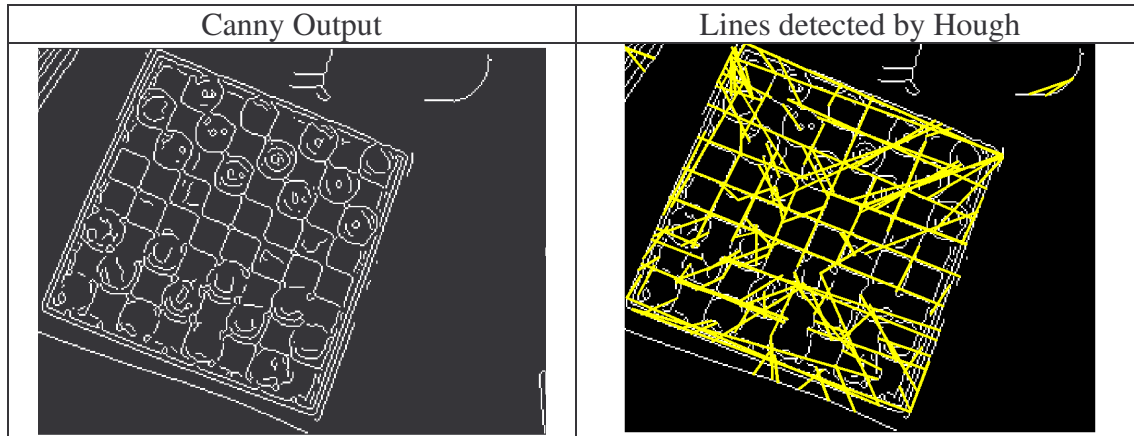
We used a cheap webcam capable of capturing images at 320x240 which does not always focus correctly to capture images with a additive noise.

The images were acquired from the video input in different environments, and with different boards:

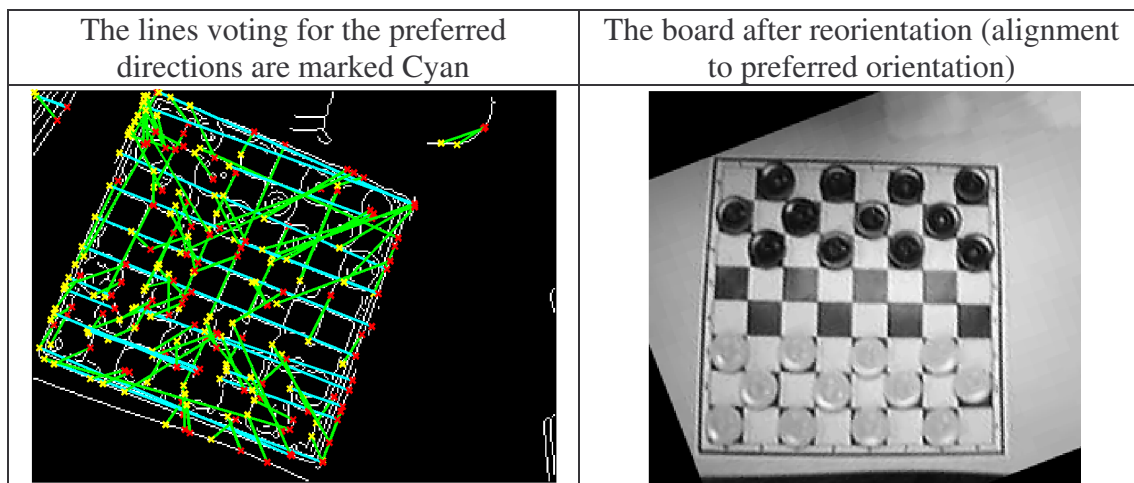


Step 1 – Detection of the Board's Orientation

To detect the orientation of lines we have first detected a set of lines in the image using Canny edge-detector followed by the application of the SHT (Standard Hough Transform - lines):



Now we are interested to compute the preferred orientation of the board in the image. For this purpose we iterate the lines returned by the Hough transform and perform a voting for the angles of the lines (the voting has the same concept as in Hough transform). The accumulator bin having the maximum vote reflects the preferred orientation of the board. This heuristic is incorrect when the background is heavily lined – however this case is prevented by our assumptions regarding the environment.



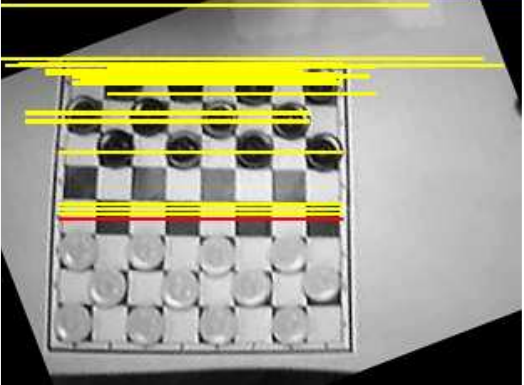
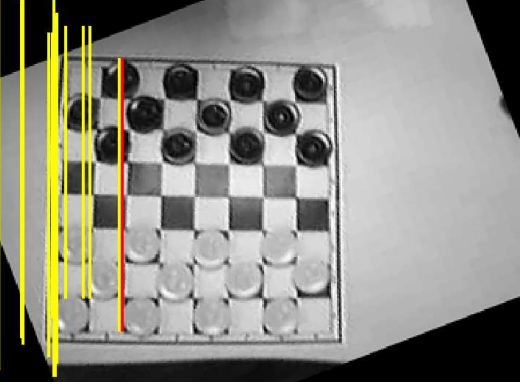
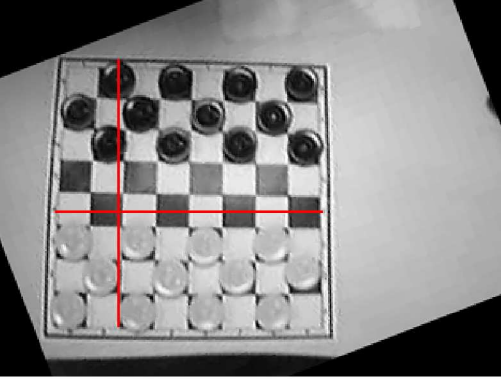
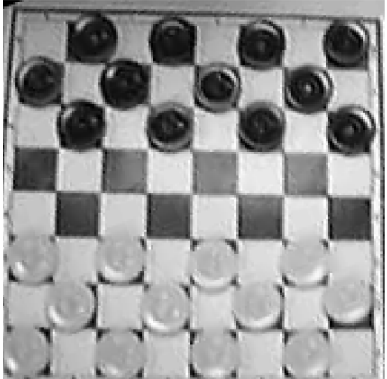
Step 2 – Detection of the Board's Location

The detection of the board's location was based on the color alternation property of the checkers board. Since the pieces are smaller than the cells they hold, this property is generally preserved. We have used segments of variable length having alternating black and white mask as a pattern.

The pattern matching is using the concept of the inner product. Dark pixels use a value of -1, while light pixels use the value of 1. Thus the inner product of the sampled line with an alternating line of -1 and 1 with 4x2 segments is the match

value. The maximum must be taken under absolute term, since the board may be reversed comparing to the alternating line. The inner product was computed by a convolution of the image with the pattern line.

The best match in our example board is depicted in the following image:

The candidate lines of the vertical scan	The candidate lines of the vertical scan
	
<p>The read lines are the best fitting placements of the pattern</p>	<p>The image with cropped background</p>
	

As an after affect of our assumption that the position of the camera is close enough to zenith, the board is relatively rectangular. If the image was taken from a different perspective, the board would be a non-rectangular trapezoidal, and we would have to detect the angles and stretch it accordingly. This is left for future research.

Now that the board location is known, we ignore the background. Therefore, we can crop the image.

Step 3 – Detection of the board parity

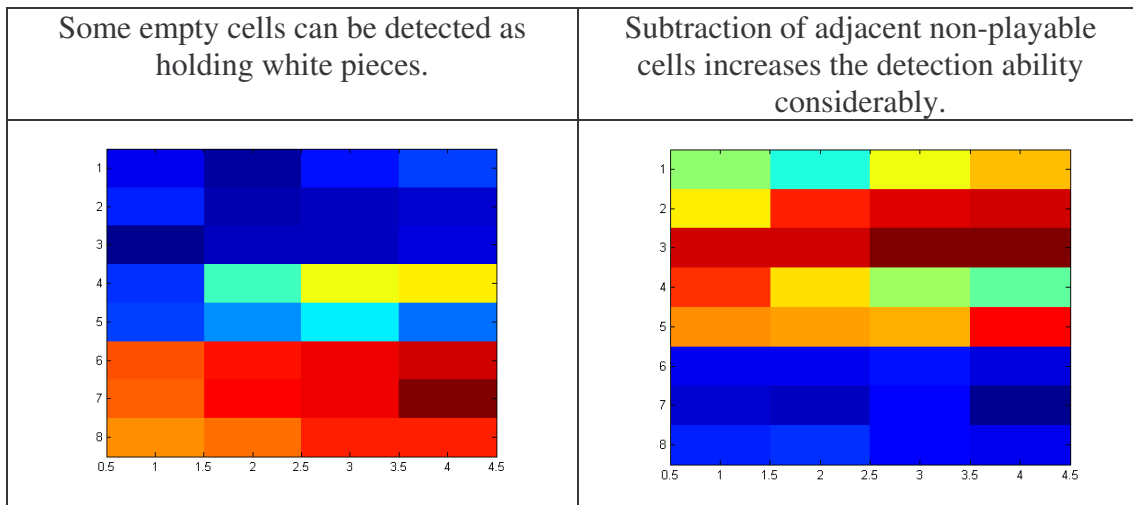
Before passing on to the next stages, we need to know what board cells may hold pieces and which may not. Since we know the board location and size, we can interpolate the cells locations. What we need to do is find out which of the even and odd sets are lighter.

We compute the average light intensity over the even and odd sets. And select the darker among them as playable. This is the convention of checker players. The program would still work if we chose and played otherwise.

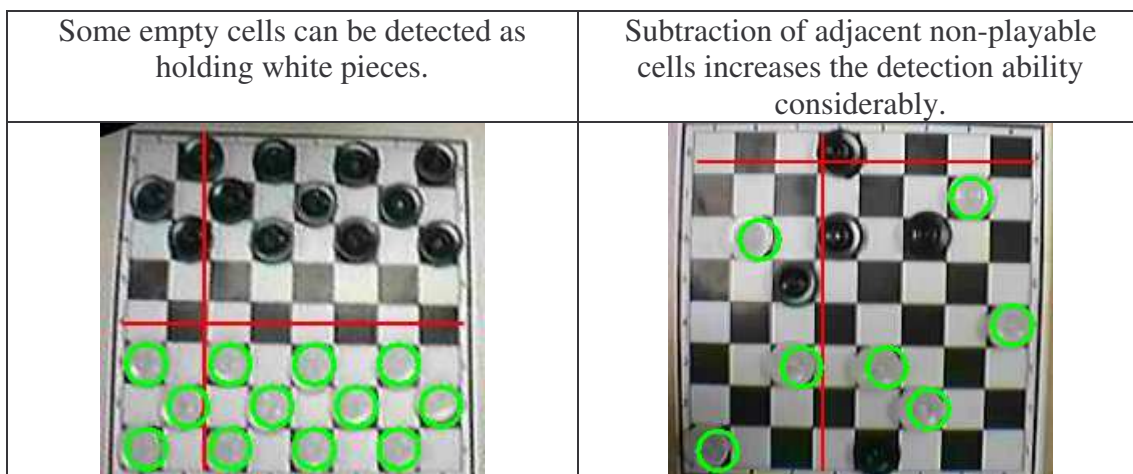
Step 4 – Detection of White Pieces

Light pieces are easy to detect when played on dark cells. (and vice versa, but we used one convention, and we plan to stick to it). This is because it changes the color of the cell. Therefore we need to detect cells with a color that don't match what we would expect.

We reuse the luminance intensities from the previous step. However, lighting artifacts prevent us from working on it directly. Some parts of the board are darker than others. A simple adjustment worked amazingly. Therefore, we did not pursue it any further. This adjustment is subtracting the luminance value of an adjacent cell from the playable one. This yielded in a 10 times value difference of the pieces from empty cells.



One problem with this method is with trapezoidal boards (under a perspective projection from a lower angle), the cells location is not interpolated correctly. This causes false detection on the board corners. However, this case is not valid under the specified assumptions. (this artifact is shown in the results section)



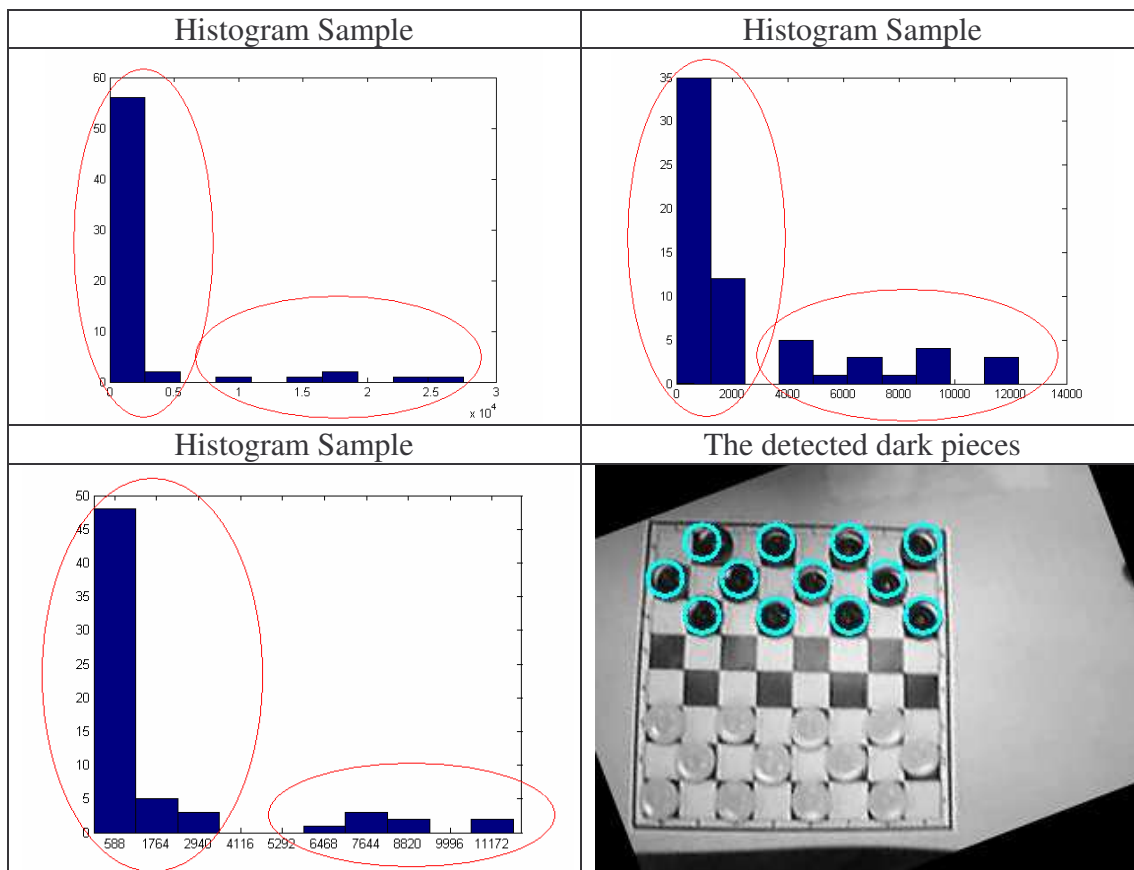
Step 5 – Detection of Black Pieces

As we have seen earlier the black pieces appear rather precisely in the output of the edge detector and thus we can apply a variation of Hough transform for circles.

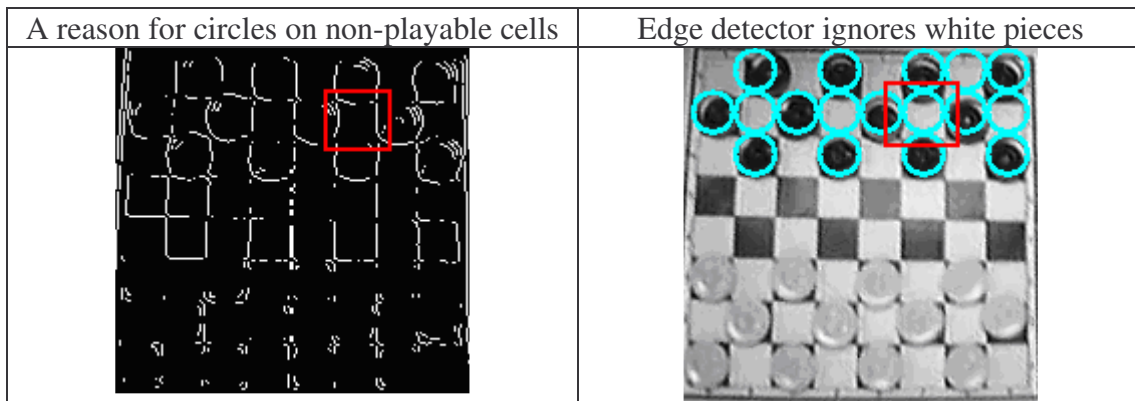
To improve the agility of this method we focused on the limited range of radiuses – close to the size of a single cell (by assumption the size of the piece can not be larger than the size of the cell).

When the voting is over it's impossible to perform thresholding on accumulator straightforwardly, since each piece can be represented by a large number of circles with different radiuses. To cope with this we have cleaned the accumulator with some basic thresholding to eliminate weak candidates. Then we have counted for each cell the number of circles whose center is contained within the cell. The histogram has shown that a count of circles in the cells accumulator has a small number (2-4) of clusters, with a noticeable distance between the first and the second clusters.

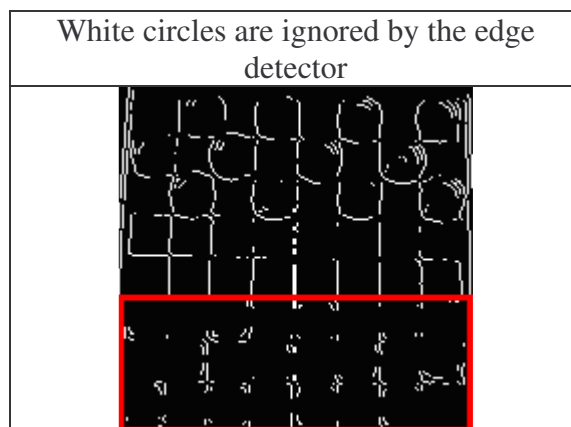
We ran weighted average distance (WPGMA) on the histogram vector in order to find 2 clusters, and used the border between them as a threshold. This allowed to us to define an adaptive global thresholding on a cells accumulator.



Since we do not have domain knowledge but only detect circles, and the image is noisy, this method also detects circles on non-playable (white) cells. We apply this knowledge by purging circles from these bins (we already identified these bins in Step 3 - Detecting the board parity).


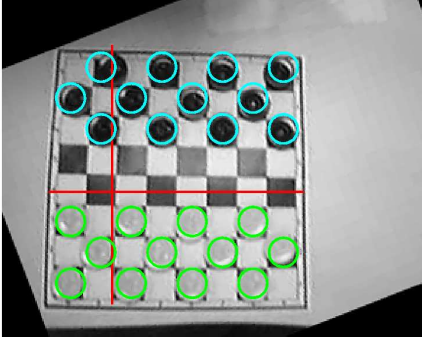
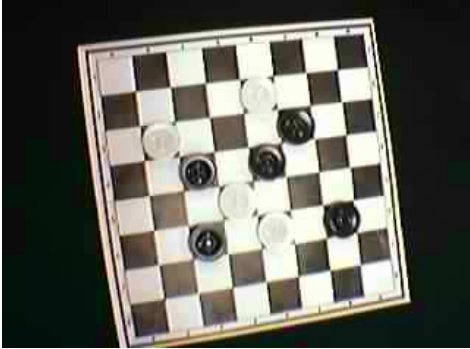
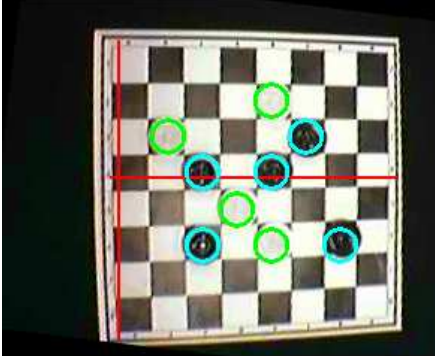

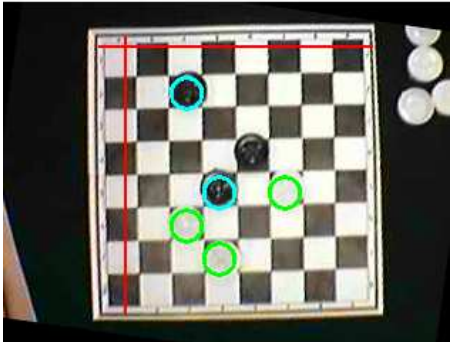


Note that although we used this method to detect the black pieces, we never addressed the light intensity. Therefore, this method may detect white pieces too. However, since edge detector commonly fails to detect white pieces, and almost always detects the black pieces with high precision, the probability of finding detecting white piece is insignificant. Anyway we have a better way to detect white pieces (furthermore we use it to identify and purge white pieces that were detected as black by the mentioned method).


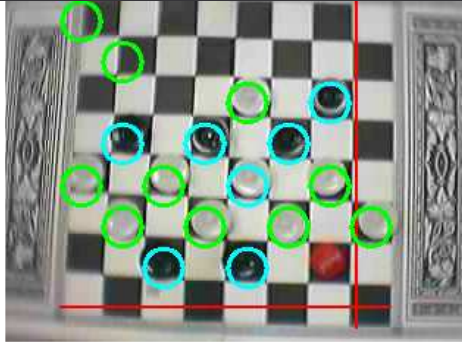



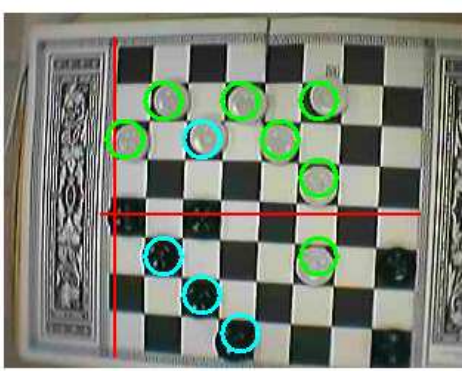


Step 6 – Enjoying the final result

Best result:

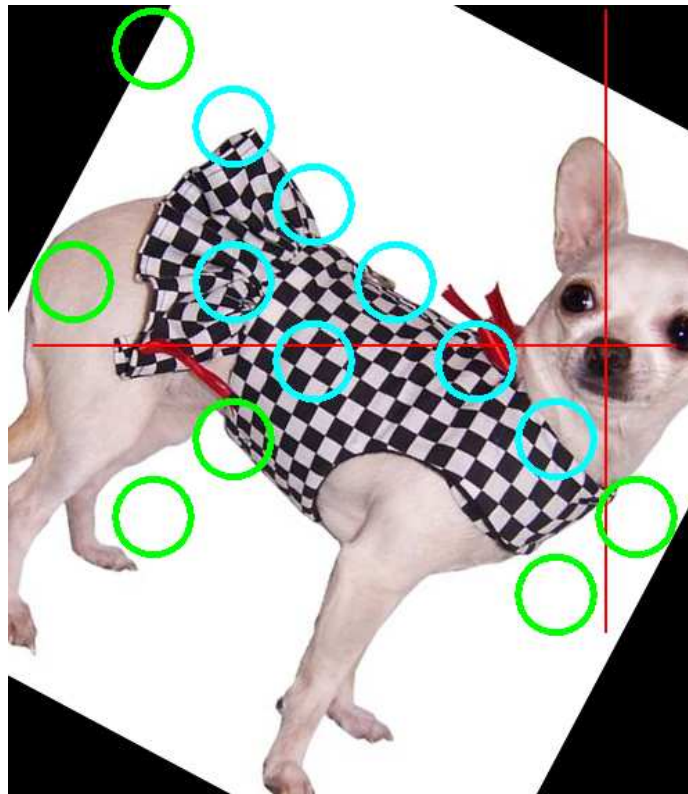
Before	After ☺
 A photograph of a 10x10 checkers board. The top half (rows 1-5) is filled with black pieces, and the bottom half (rows 6-10) is filled with white pieces. The board is tilted slightly to the right.	 The same board as in the 'Before' image, but with red lines drawn across it. A vertical red line is on the second column from the left, and a horizontal red line is on the sixth row from the top. The pieces on the board are highlighted with colored circles: black pieces are circled in cyan, and white pieces are circled in green.
 A photograph of the checkers board with several pieces moved from their starting positions. There are black pieces on squares that were previously white, and white pieces on squares that were previously black.	 The board from the previous image with the same red lines. The moved pieces are highlighted: black pieces are circled in cyan, and white pieces are circled in green.
 A photograph of the checkers board with more pieces moved. The black pieces are now clustered in the center and lower-middle of the board, while white pieces are also clustered in the center and lower-middle.	 The board from the previous image with the same red lines. The moved pieces are highlighted: black pieces are circled in cyan, and white pieces are circled in green.

Imperfect results:

Before	After ☺
 A photograph of a checkers board with pieces in their starting positions. A red piece is on the bottom row, and a black piece is on the second row from the bottom.	 The same checkers board as in the 'Before' image, but with several pieces highlighted by green and cyan circles. Two red lines are drawn on the board: a vertical line on the right side and a horizontal line near the bottom.
 A photograph of the checkers board from a different angle. The red piece has moved from the bottom row to the second row from the bottom.	 The same checkers board as in the 'Before' image, but with several pieces highlighted by green and cyan circles. Two red lines are drawn on the board: a vertical line on the right side and a horizontal line near the bottom.
 A photograph of the checkers board from a different angle. The red piece has moved from the bottom row to the second row from the bottom.	 The same checkers board as in the 'Before' image, but with several pieces highlighted by green and cyan circles. Two red lines are drawn on the board: a vertical line on the right side and a horizontal line near the bottom.

Conclusions

- While a human can detect the board easily on a low-res noisy picture from different perspectives instantly, it is far from trivial to detect it computationally. Even in almost ideal conditions, it is hard to detect it quickly, making it unrealistic in software in real time systems.
- Imperfect lighting conditions reduce the accuracy of the detection.
- Surprise: a straight line may not look like such when captured by a cheap webcam.
- We optimized for precision over recall, and showed that it is possible to have a low rate of false positives. The other side of the coin is that we have introduced a few false negatives (missed a few pieces).
- The assumptions made in this paper are realistic, yet restrictive which contradicts to the concept of AR systems. Further research may loosen up some of these restrictions. It was not done since it would increase the scope of this project considerably.
- Even if the environment is illegal, the system still succeeds to interpret the scene as it was designed:



References:

- Augmented Reality, <http://www.se.rit.edu/~jrv/research/ar/introduction.html>
- Interaction Between Real and Virtual Humans: Playing Checkers, R'emy Torre, Pascal Fua, Selim Balcisoy, Michal Ponder and Daniel Thalmann
- Introduction to Computational and Biological Vision, Course slides, Ohad Ben-Sachar, Computer Science, Ben-Gurion University of the Negev, Fall 2006.
- Clustering, http://en.wikipedia.org/wiki/Data_clustering