Augmented Reality for Board Games

Eray Molla[*]
EPFL, CVLab

Vincent Lepetit[†]
EPFL, CVLab

ABSTRACT

We introduce a new type of Augmented Reality games: By using a simple webcam and Computer Vision techniques, we turn a standard real game board pawns into an AR game. We use these objects as a tangible interface, and augment them with visual effects. The game logic can be performed automatically by the computer. This results in a better immersion compared to the original board game alone and provides a different experience than a video game. We demonstrate our approach on Monopoly™ [1], but it is very generic and could easily be adapted to any other board game.

1 INTRODUCTION

Augmented Reality has most certainly an important potential for the game industry. However, it is still difficult to design AR games that would appeal to the large public, Sony's Eye of Judgement, developed for Sony's Playstation 3, being one notable exception.

Most of the AR games developed by the community involve visual markers [7, 17, 5, 4, 12] or Head Mounted Displays [13, 14, 1]. However, visual markers result in a less convincing illusion—Eye of Judgement developers hid them carefully—and restrict the creativity of the game designers. HMDs are still cumbersome, and far from being wide spread among the large public.

Thanks to recent developments [18], markers are not required any longer, and some games have been proposed for which the registration is performed using natural features on a mobile device [6]. For most of the proposed approaches, however, the game involves the players to move around a board with a handheld device. This is not necessarily comfortable, especially for a long time.

Our approach is therefore to adopt the same setup as Eye of Judgement, a camera pointing to the real scene, and the augmented scene visible on a computer screen, but with a different type of game. Figure 1 shows that we can enhance traditional board games with virtual elements, by combining existing Computer Vision techniques to locate the board and the pawns. These physical elements can be manipulated by the players as usual, making our approach natural to non-expert users.

In the remainder of the paper, we first describe the methods for detection of game board and object tracking, and present our results on the Monopoly™AR game.

2 RELATED WORK

Applying Augmented Reality to games is not new. Since [13] and [7], probably the earliest references on the topic, different types of AR games have been proposed.

Early works rely on Head Mounted Devices (HMDs) for the visualization of the virtual elements [13, 7, 14, 1, 5, 11]. While seducing, HMDs are still uncomfortable as the underlying technology is not mature yet. Another drawback of early works is the use of markers, as they are not elegant and constrain the game design.

[*]e-mail: eray.molla@epfl.ch
[†]e-mail: vincent.lepetit@epfl.ch

[1]Monopoly™is the Trademark of Hasbro Company, Rhode Island, United States.

With technological and algorithmic improvements, it is now possible to use natural features instead of markers, and handheld devices for the visualization instead of HMDs [18, 6]. But even with these new developments, playing AR games does not seem very comfortable. Most of the proposed game concepts require the player to remain stood up and to revolve around a table, with his arms lifted when using a handheld device.

That is why we chose to concentrate on board games, which use in AR was advocated in [10]. However [10] considers limited Computer Vision techniques—the camera must be on top of the board for example—or RFID transponders and magnetic field sensors. By contrast, we use only a simple webcam that can be positioned arbitrarily around the board. [5] proposes an AR version of the Chinese Checkers, but uses markers. The pawns are only virtual and are moved using a marker equipped with a physical button that must be pressed, thus loosing the advantage of a tangible interface.

One work closer to ours is [2], which uses recent Computer Vision techniques to track a textured planar object that can be augmented with a marble maze game. We rely on similar techniques to detect the board, but we also show how detect the pawns using the same camera to augment them. In our approach, the visualization is simply done on the computer screen, as was done in [15] and most of the current commercial AR applications.

3 TRACKING THE GAME ELEMENTS

In this section, we briefly describe the Computer Vision techniques we use to localize in 3D the physical support of the game, the board and the pawns, in the images captured by a webcam to implement the game logic and add virtual elements. Figure 1 gives an overview of our approach.

3.1 Detecting the Board

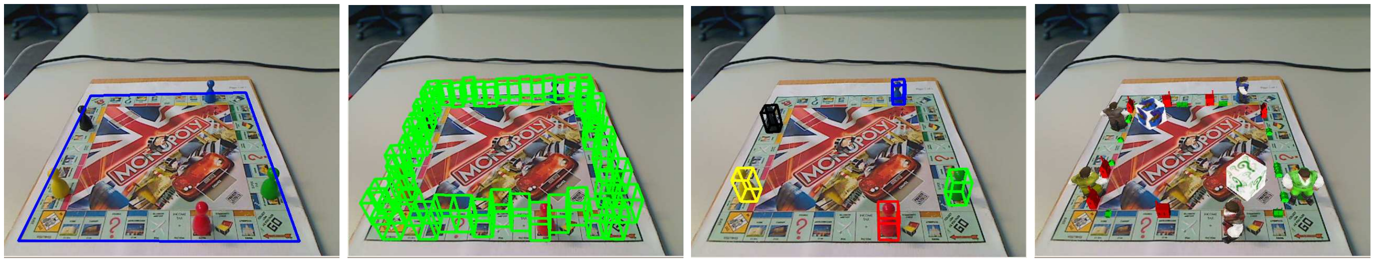
Detecting a planar object like a game board and estimating its pose in 3D using only natural features is now standard. In practice we use the BRIEF descriptor [3] to match feature points between a reference image of the board and the image captured by the camera. Then the rotation and translation of the board are estimated from these matches using EPnP [9] and RANSAC.

Knowing the board pose not only allows us to augment the board, but also to constrain the pawns detection, as described below.

3.2 Detecting the Pawns

The real pawns are moved by the players as a tangible according to the dice scores, and must be detected so the software can follow the game progress and transform the pawns into virtual characters.

We use pawns of all the same simple shape, distinguishable only by their color. We use the Viola & Jones [16] object detector implementation in OpenCV. It originally looks exhaustively over the whole input image and a range of scales. However, since we are interested only in detecting the pawns at the authorized places on the board, most of these 2D locations and scales do not correspond to a physically possible 3D location for a pawn. By using our knowledge of the board pose, computed as described in the previous section, we can constrain the detector to consider only these possible 3D locations. This approach was first suggested in [8]. It both speeds up the search and reduces the rate of false detections. Moreover, since the pawns move very rarely, we run the detector on only a random subset of the valid locations to save computation time.



(a) (b) (c) (d)

Figure 1: Overview of our approach. (a) We first locate the board using feature point matching. (b) By projecting 3D boxes located at valid places, we obtain the 2D locations the detector must try to locate the pawns on the board. Only a subset of the 3D bounding boxes is shown here. (c) The pawn detection and color recognition procedure gives us the location of the pawns on the board. (d) We can now add the virtual elements. Note that we can correctly orient the virtual characters even if the pawns are symmetric by relying on the board orientation.

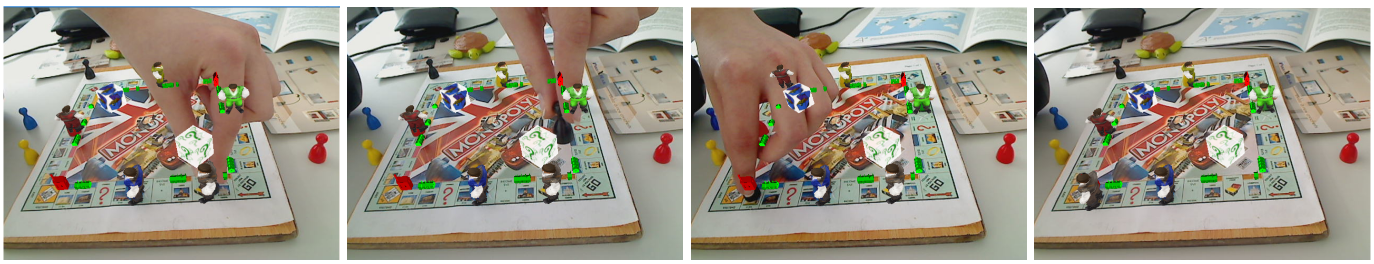


Figure 2: The pawns as tangible interface. The players can move their pawns according to the dice score. Note that the algorithm is not distracted by the other pawns on the table as we consider only valid locations.

As shown in Figure 1(b), we project a large number of 3D bounding boxes the size of the pawns and located at valid places on the board. By simply taking the 2D bounding boxes of each of these projections, we get the locations and scales where the detector must be run.

Since the pawns have the same shape, we still have to estimate their color to find to which player they belong. Due to the shape of the pawns we are using, the upper part of the 2D boxes a pawn includes many pixels from the background, thus we consider only the bottom part of the box. We rely on a voting scheme: Each pixel votes for a pawn color—yellow, red, green, blue, or black—based on its hue and intensity. We finally assign the color that receives the majority of votes to the pawn.

Our implementation runs at about 20 Hz.

4 Conclusion

We have shown how to transform a board game into an Augmented Reality game using a simple webcam. Our current implementation does not include game logic, but given the output of the Computer Vision algorithms, it is straightforward to do. We demonstrated our approach on a simple game, but the topic is incredible rich: By animating the characters and the board, Augmented Reality can make traditional board games much more immersive. Doing so, we could reach a large audience outside of people attracted by technological novelties, to which current AR applications are still often limited.

References

[1] T. Andersen, S. Kristensen, B. Nielsen, and K. Gronbaek. Designing an Augmented Reality Board Game With Children: the Battleboard 3D Experience. In *Conference on Interaction Design and Children*, 2004.

[2] P. Azzari and L. Stefano. Vision-Based Markerless Gaming Interface. In *ICIAP'09*.

[3] M. Calonder, V. Lepetit, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *ECCV'10*.

[4] K. Cho, W. Kang, J. Soh, J. Lee, and H. Yang. Ghost Hunter: a Handheld Augmented Reality Game System With Dynamic Environment. In *ICEC'07*.

[5] N. Cooper, A. Keatley, M. Dahlquist, S. Mann, H. Slay, J. Zucco, R. Smith, and B. H. Thomas. Augmented Reality Chinese Checkers. In *International Conference on Advances in Computer Entertainment Technology*, 2004.

[6] ARhrrrr!, 2009. Georgia Tech Augmented Environments Lab.

[7] A. Govil, S. You, and U. Neumann. A Video-Based Augmented Reality Golf Simulator. In *International Conference on Multimedia*, 2000.

[8] D. Hoiem, A. Efros, and M. Hebert. Putting Objects in Perspective. In *CVPR'06*.

[9] V. Lepetit, F. Moreno-Noguer, and P. Fua. EP*n*P: an Accurate $O(n)$ Solution to the P*n*P Problem. *IJCV*, 2008.

[10] C. Magerkurth, T. Engelke, and M. Memisoglu. Augmenting the Virtual Domain with Physical and Social Elements. In *International Conference on Advances in Computer Entertainment Technology*, 2004.

[11] T. Nilsen and J. Looser. Tankwar - Tabletop War Gaming in Augmented Reality. In *International Workshop on Pervasive Gaming Applications*, 2005.

[12] O. Oda, L. Lister, S. White, and S. Feiner. Developing an Augmented Reality Racing Game. In *INTETAIN'08*.

[13] T. Ohshima, K. Satoh, H. Yamamoto, and H. Tamura. AR2 Hockey. In *SIGGRAPH'98*.

[14] W. Piekarski and B. Thomas. ARQuake: the Outdoor Augmented Reality Gaming System. *Communication of the ACM*, 45(1):36–38, 2002.

[15] C. Scherrer, J. Pilet, P. Fua, and V. Lepetit. The Haunted Book. In *ISMAR'08*.

[16] P. Viola and M. Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features. In *CVPR'01*.

[17] D. Wagner, T. Pintaric, and D. Schmalstieg. The Invisible Train: A Collaborative Handheld Augmented Reality Demonstrator. In *SIGGRAPH'04: Emerging technologies*.

[18] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Pose Tracking from Natural Features on Mobile Phones. In *ISMAR'08*.