

# MoleARlert - An Augmented Reality Game Based On Lemmings

Sandy Engelhardt, Annabell Langs, Gerrit Lochmann, Irini Schmidt  
Prof. Dr. Stefan Müller

Institut für Computervisualistik  
Universitätsstr. 1  
56072 Koblenz  
Tel.: +49 261 287-2727  
E-Mail: stefanm@uni-koblenz.de  
E-Mail: engelhardt@uni-koblenz.de  
E-Mail: allangs@uni-koblenz.de  
E-Mail: lochmann@uni-koblenz.de  
E-Mail: ischmidt@uni-koblenz.de

**Abstract:** In this paper we present our outdoor AR game MoleARlert. The idea behind this research prototype was to exploit the enormous potential of AR in combination with a classical game play on the basis of the well-known game *Lemmings* by *Psygnosis* [Hoc]. Real players interact with virtual creatures on a real playing field to guide the moles through the hazardous environment to their final goal. The multiplayer game is observed through a stationary video see-through monitor. The team leader directs the real players around the field in order to steer the game with special markers and human gestures. Thus, the game requires a lot of actions from players and is very entertaining.

**Keywords:** Augmented Reality, Game, Tracking, Interaction

## 1 Introduction

Numerous Augmented Reality (AR) games have already been developed. Popular indoor games such as *human packman* [CG04], *Levelhead* [Oli09] or *ARf* [Mac08] have been attractive prototypes to demonstrate the benefits of AR for the users.

Other approaches tried to track people inside a predefined area. *Kids Room* [TMML99] was one of the first demonstrations of how to track and classify people inside a room. *The eXperience Induction Machine* [UB07] added more interactive elements, for example a sensitive floor.

In the same way multiplayer games like *Invisible Train* [WPS09] or *SHEEP* [CS02] have influenced the genre by using handheld devices. Being free in movement while playing the game is a remarkable solution provided by such portable equipment. First AR games are available even on mobile phones (*Apple's iPhone*) and game consoles with integrated cameras (*Nintendo's DSi*). *Human PacMan* as well as *AR Quake* [PT02], a remake of the original *Quake* game by *id Software* [Car], were promising attempts to bring this concept outdoors.

The idea behind our game *MoleARlert* was to combine the potential and the added-value of AR

with a well-known classical game. A fast game flow aims for vivid action of the players on the field. This is one of the most important successes of our game concept.

Especially easy usability for players in terms of the opportunity to play immediately without any preparation and additional devices is an essential feature. The course of the game can be influenced in a very elementary way by human gestures and markers.

To our knowledge, the size of 10.8 x 10.8 meters of the playing field, which is tracked frame by frame, is unique as well. This approach implies a large scale AR game with models of appropriate size supporting the player's immersion. Suitable camera positions had to be found, in addition to realizing a fast preprocessing of the livestream.

## **2 Game concept**

Our augmented reality game is based on the original 2D-Game *Lemmings* by *Psygnosis*, 1991. Little creatures, the lemmings, have to be directed from a starting point to a finish mark. Their indetermined and blind behaviour characterizes the game flow. They walk helplessly in a row, unable to make up their minds. During their journey they have to be saved from various hazards. In order to accomplish this, the player has different options at his disposal, e.g. he can make the lemming dig through a hill or build a bridge over a gap.

We have adapted this game concept and adjusted it according to our needs and environment.

### **2.1 Adapted game concept**

As the title *MoleARlert* implies, we designed and animated moles in analogy to the game figures lemmings. Our moles, blind and awkward, walk equally straight in one direction unless the player interacts with them or a game element changes their direction.

In this approach, real people should become game figures themselves. So the idea of two-dimensionality has been extended to the real world. Due to this, a small part of our university campus has become our playing field. This field stretches 12 tiles in orthogonal directions (Figure 1). Each tile in the real world corresponds to one tile in the game. Consequently, we numbered the playing field like a chessboard for easy orientation and inner representation.

On each tile can be different game elements, like start- and endfields, arrows, hills, water, deadly hazards, or a combination of those. With the second floor on top of the hills the third dimension is added to the game concept. Figure 1 depicts a common game scene with different game elements as shown in the AR View (Chapter 4).

### **2.2 Player interaction**

The player can interact with the moles in two different ways:

1. Redirect the moles by setting an arrow with a marker
2. Give the moles a special ability by a human gesture, namely to dig through a hill or to build a bridge



Figure 1: A scene out of *MoleARlert*: a player is placing a marker on the field. The AR View is shown, including moles, game elements and GUI.

The first interaction is realized by the use of markers, whose encoded direction can easily be recognized by the attached arrows. Therefore, the player can place them on the field intuitively. The number of markers is limited to four, so the usage of each marker needs to be considered carefully.

A straightforward way to change the moles' behaviour is the second interaction mentioned above. For better distinction this is not realized with additional equipment. The player himself grants the ability to the first mole which enters the corresponding field. More precisely, the player has to perform a predefined gesture. Standing on a tile for a certain time means the next mole will build a bridge. Covering will make the next mole dig through a hill.

Two constraints have to be considered here:

- The mole will only build a bridge if there is a gap or water in front of him. The bridge is automatically built in the appropriate length.
- The mole will only remove the ground if it hits a diggable hill. The mole will dig straight ahead until no hill is nearby or an undiggable hill is hit.

We differentiated between a diggable and an undiggable hill, because otherwise many levels would have become too easy or moles would be able to destroy important game elements which lie on top of the hill. Important game elements may be a startfield or an endfield, which lie on undiggable hills per definition.

### 2.3 Gameflow

First of all, a level — in the following called map — is selected in the menu. These maps hold the initial game elements, which exist at the beginning of the game. An underlying XML file encodes the map structure, which can easily be created and edited. Further game information besides game

elements are also stored there, e.g. the number of moles to be saved, the author of the map or the mapname.

After the map is selected, the actual game starts. At this point no markers or players should be positioned on the field. The first moles appear on the map. Now the players need to use the markers or gestures to show the moles the correct way to the endfields. Doing so will make the moles carry out different actions, e.g. dig through a hill or turn around.

When all moles are saved or killed due to hazards which could not be overcome, the game ends. The elapsed time and the number of moles saved is recorded and used in the next game state. The next game state is the highscore state in which players can enter their teamname. This is possible only if their score, calculated by the elapsed time and number of saved moles, is better than the first 10 scores from previous teams already in the highscorelist. Each map has an independent highscore and each highscore is saved in an XML file.

### 3 Architecture

The system consists of three separate components, connected via network; a video see-through Terminal (server) and two Tracking Modules (client). The AR Terminal, combined with camera (1) (Figure 2), is located directly in front of the field where players can see the virtual overlay on the livestream. This stream is captured by an ordinary Logitech Quickcam Pro 9000 webcam. The Tracking Module runs on two separate computers, each having its own camera (2) and (3).

This arrangement allows for us to have two different views of the scene to separate detection of markers and tracking of persons from each other. Thus, using two computers allows for greater speed.

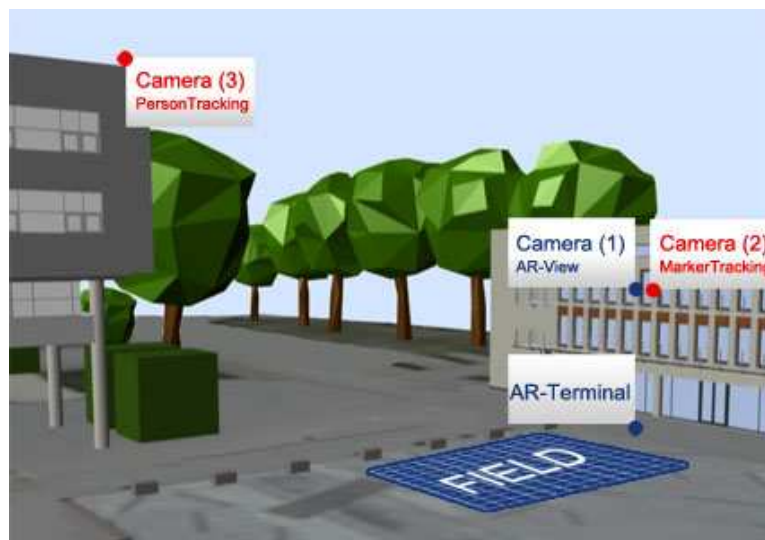


Figure 2: Architecture of *MoleARlert*. Location of the video see-through component (blue) and Tracking Module (red) on the campus.[Pri]

## 4 AR Terminal

To visualize augmented reality, it is necessary to merge a live stream with computer generated images and animations. In this case, the AR Terminal runs an *OGRE*-application [Str] with a video see-through using a webcam live stream as background texture. The video frames for the background are captured asynchronous with Intel's *Open Source Computer Vision Library* [Int] and are written to *OGRE* dynamic textures. Different 3D-models are placed in front of the background plane, covering all game-relevant regions. The real field is now augmented with walls, gaps and bridges. Items like arrows, start- or endfields are allocated on virtual terrain as well as on real ground.

To enhance the effect of 3D-objects placed on the campus, shadows are cast on an invisible ground plane above the video plane. As the game starts, moles appear, start walking, react to items and cast shadows to where in fact is an empty place.

Without further computation, real objects or players would be covered by rendered models. To avoid this, we implemented a rudimental covering prevention by cutting out the concerned parts of the 3D scene. Cylinders are placed at the player's position and allow for the video to pass through. This passing is done by rendering cylinders before the video plane in the rendering queue. In addition, the video plane does not compute a depth check so it is unaffected by clipping at the cylinder's edges.

The players' positions are transmitted as an array of coordinates to the terminal via network. On each frame, cylinders are placed on the received field positions. If a new coordinate is nearby a position of the previous frame, the corresponding cylinder will be moved. If a position is not confirmed within the range of a predefined period, the cylinder will be discarded. Although the committing rate is not high enough to give the impression of a constant motion, it is useful to see stationary performed gestures of the players through the 3D scene. In this way covering prevention provides additional insight into the game to players and spectators.

## 5 Tracking

The following chapter describes the tasks of the Tracking Module using the methods of [Int].

### 5.1 Camera Calibration

In order to have the best suitable fitting of virtual objects into the real world, the position of the camera must be known (5.1). This is done with extrinsic calibration. Therefore, image coordinates from four corners of the field have to be selected in a frame. The 2D-image points are matched to their corresponding 3D world coordinates. As implemented in *OpenCV* ([Int], Chapter 5) the method of *Zhang* [Zha98] is applied.

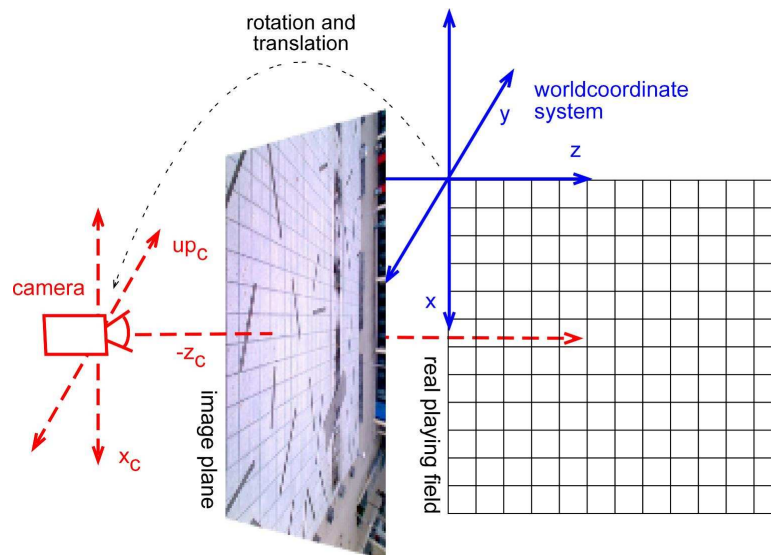


Figure 3: World coordinate system and camera coordinate system. The position and orientation of the camera in world coordinates is computed during calibration. Therefore rotation and translation have to be determined.

## 5.2 Markertracking

Our approach is to allow players to intuitively change the direction of the moles by placing a simple color marker on the field. This marker encodes four directions and can be placed on the field as shown in figure 4.

Considering the shape of the marker, a rectification ([Int], Chapter 1) was necessary in the frames from camera (2) in figure 2 to get a top view on the scene (figure 4, step 1). Not relevant regions of the image, e.g. parts that lie outside of the playing field, are set black.

The next processing step includes finding yellow and green colored regions on each frame. This is achieved by computing a histogram backprojection. A color histogram of the marker colors has to be provided as input for this technique.

The output is a grayscale image depicting yellow and green areas as white and light grey regions, as shown in figure 4, step 2.

The backprojection image is applied to compute bounding boxes of all white regions (figure 4, step 3). For that reason the white regions have to be closed first. This is done by morphological dilation and erosion operations.

Furthermore, the content of the computed bounding boxes have to be analyzed, as seen in figure 4, step 4. This is achieved by using the bounding boxes as regions of interest (ROI) on the original frame. The color ratio is analyzed by converting them into HSV color space and verifying the values in each channel to lie within specific thresholds.

If a region is classified as a marker, the position — the center point of the bounding box — and the direction of the marker will be sent to the AR Terminal and an arrow will appear in the AR View.



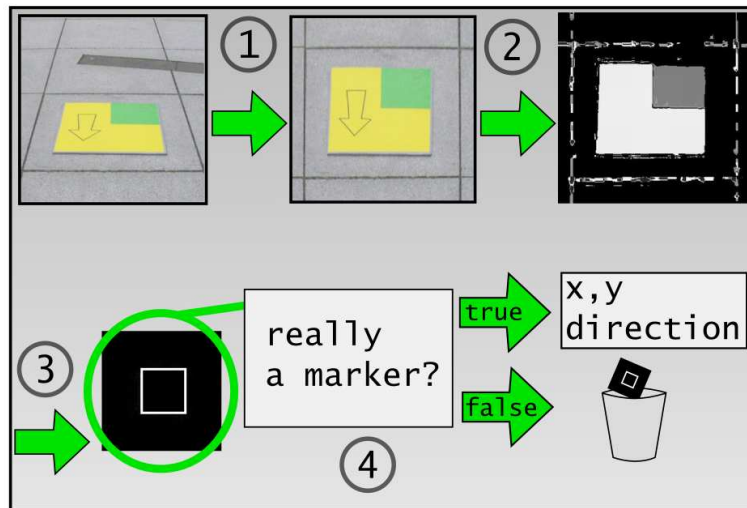


Figure 4: This figure illustrates the marker tracking exemplary with one marker. Depicted is a cutout of the field.

### 5.3 Person Tracking And Gesture Detection

To detect persons on the livestream we used the background subtraction algorithm. The basic function for calculating the difference between background and the actual frame of the camera can be found in ([Int], Chapter 3). More information about the technique can be found in [GB08], Chapter 9. By accumulating the first five frames, we create a background model which is needed to calculate the difference. As a result of this accumulation, a short forerun is needed after starting the program.

In order to reduce the effects of lighting changes, the background model is updated every 10th frame.

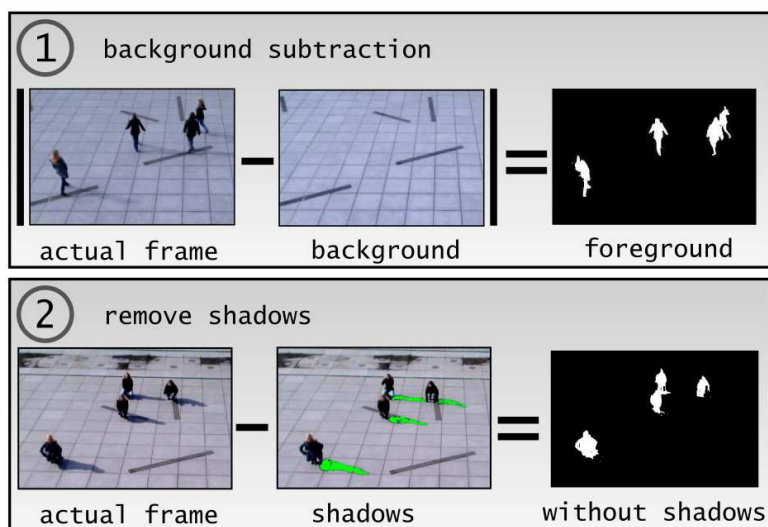


Figure 5: Person tracking. Step 1 shows the background subtraction, step 2 illustrates the removal of shadows. Additionally different gestures are depicted: standing (= bridge) and covering (= digging).

The background subtraction output is a binary image where persons appear as white regions (Figure 5). The white regions are computed with the technique described in [GB08], Chapter 9 (Connected Components). Applying this method implicates the appearance of shadows in the output image. In HSV color space shadows could be distinguished from the background like mentioned in [CGP<sup>+</sup>]. We used this technique to get rid of the shadows.

After preprocessing we compute the relative position of each person in the frame. This is necessary to render the cylinders at the same location described in chapter 4.

Therefore, all corresponding bounding boxes are determined. The center point of the lower border of the bounding box serves as position coordinates. This point is matched in the field and then sent to the AR Terminal.

As described in chapter 2.2, players are allowed to make two different gestures, which can be tracked and identified. These gestures are shown in figure 5 (step 1 shows the bridge gesture, step 2 the digger gesture). The gestures are differentiated by computing the side to side ratio of the bounding boxes.

## 6 Results

We developed a prototype of an interactive AR game entertaining for players as well as spectators. The enormous potential of AR was exploited to enhance and extend existing concepts. Traditional AR game approaches have been mixed up with realtime strategy games to a new genre. For the first time the player himself becomes an integrated piece of a 3rd person game. The player's physical involvement increases action and entertainment playing *MoleARlert*.

Already during the development phase, numerous people stopped by and became interested spectators of the action-packed scenario. It was a new experience to watch virtual characters among real persons interacting on the same screen.

The AR Terminal not only gives the audience the possibility to observe the course of the game, but the involved team also might plan their next move. Declaring a game leader to inspect the situation might also be an advisable solution to coordinate the players on the field. Good communication and teamwork is the key to reach a high score.

The attractiveness of the game is increased offering several maps of varying difficulty. Because of the underlying XML structure players are able to design their own dodgy game-scenarios combining obstacles like water, hills, valleys and traps.

Different map designs demand various means of interaction from the team. Focus can be set on action-loaded or strategical, easy or difficult gameplay. Markers or human gestures can be applied to control the mole's pathes to lead them to their goal savely. A wide variety of maps as well as a diverse and cheerful game experience have been achieved.

On the *Day of Computational Visualistics* [IoCV09], a day of the open door at the Institute of Computational Visualistics, University of Koblenz *MoleARlert* has won the audience award.





Figure 6: Players' interactions and coordinations on the field.

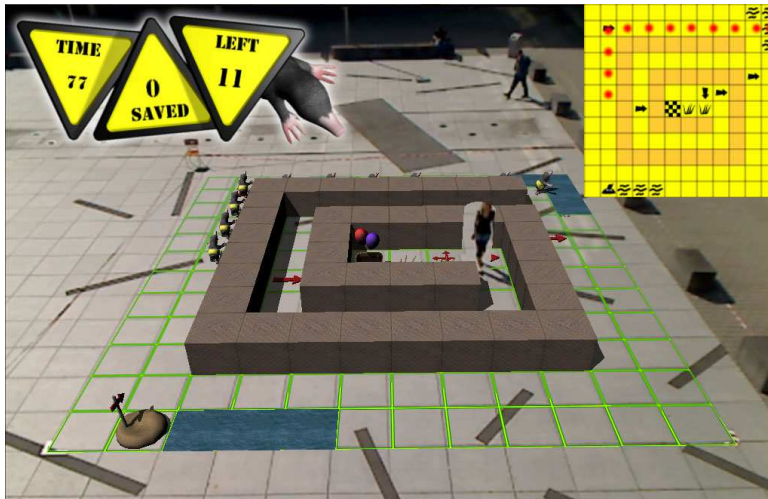


Figure 7: Playing scene with enabled grid depicting player coverage prevention using cylinders (during development).

## 7 Conclusion

To reflect on the current state of work and the possible tasks in the future, it is useful to consider the performance of the system. By using markers only as means of interaction, the application is very sufficient and real-time capable. However, using the gesture recognition and transparent cylinder rendering, the system delay is up to one second. Nevertheless, we found this delay quite acceptable and not really disturbing for this kind of gameplay.

The complex image-processing tasks have been distributed on two remote stations, joining their calculation results on the AR Terminal. Surprisingly we did not require any high-priced equipment for this. Three average webcams and computers basically fulfilled the technical specifications. Considering this point, there is still potential in increasing the performance of the system.

Outdoor conditions, such as reflections on wet floor or changing lighting and shadow effects,

decreased the quality of the webcam signal. Nevertheless, we achieved very high error tolerance by applying a sophisticated set of filters to preprocess each frame.

Transparent cylinders were used to avoid occlusion of real characters by virtual objects. However the quality of the cutout regions in the images does not fulfill our conception so far. In the future we will focus on computing the contours of the dynamic persons in order to provide a shape based occlusion.

## 8 Acknowledgements

We would like to thank all developers of *MoleARlert*: Vera Müllenbach, Tim Oppermann, Dominik Ospelt, Elena Root, Nicole Thorn, Tobias Tropper, Hans-Christian Wollert, Elisa Ziegler and especially our supervisor Stefan Rilling. Without our team this game would not have been possible. Special thanks go to *Labor Bilderkennen* and the provided rendered image (Figure 2) out of their *Google Sketchup 3D Modell* of University of Koblenz

## References

- [Car] John Carmack. id software. <http://www.idsoftware.com/>.
- [CG04] Adrian David Cheok and Kok Hwee Goh. Human pacman: A mobile wide-area entertainment system based on physical, social and ubiquitous computing. <http://www.mixedreality.nus.edu.sg>, 2004.
- [CGP<sup>+</sup>] Rita Cucchiara, Costantino Grana, Massimo Piccardi, Andrea Prati, and Stefano Sirotti. Improving shadow suppression in moving object detection with hsv color information. Technical report.
- [CS02] Martin Wagner et al. Christian Sandor. Sheep: The shared environment entertainment pasture. <http://ar.in.tum.de/pub/dwarf2002sheep/dwarf2002sheep.pdf>, 2002.
- [GB08] Adrian Kaehler Gary Bradski. Learning opencv: Computer vision with the opencv library, 2008.
- [Hoc] Mick Hocking. Sce studio liverpool. <http://www.development.scee.net/>.
- [Int] Intel. Open source computer vision library. <http://opencv.willowgarage.com/wiki/CvReference>.
- [IoCV09] University of Koblenz Institute of Computational Visualistics. Cv-tag 2009. <http://www.uni-koblenz.de/~cvtag/2009/index.php>, 2009.

- [Mac08] Blair MacIntyre. A sneak peek at arf, an ar virtual pet game for the iphone. <http://www.augmentedenvironments.org/lab/>, 2008.
- [Oli09] Julian Oliver. levelhead. <http://julianoliver.com/levelhead>, 2009.
- [Pri] Lutz Priese. Pose. [http://www.uni-koblenz.de/~lb/lb\\_research/pose/](http://www.uni-koblenz.de/~lb/lb_research/pose/). Realtime Posetracking By The Help Of Distinctive Features.
- [PT02] Wayne Piekarski and Bruce Thomas. Arquake: The outdoor augmented reality gaming system. Technical report, 2002.
- [Str] Steve Streeting. Object-oriented graphics rendering engine 1.6.1. [www.ogre3d.org](http://www.ogre3d.org).
- [TMML99] Aaran F. Bobick et al. The MTI Media Laboratory. The kidsroom: A perceptually-based interactive and immersive story environment. <http://mf.media.mit.edu/pubs/journal/Kidsroom.pdf>, 1999.
- [UB07] Paul FMJ Verschure Ulysses Bernardet, Sergi Bermudez i Badia. The experience induction machine and its role in the research on presence. [http://www.temple.edu/ispr/prev\\_conferences/proceedings/2007/Bernardet,%20Badia,%20and%20Verschure.pdf](http://www.temple.edu/ispr/prev_conferences/proceedings/2007/Bernardet,%20Badia,%20and%20Verschure.pdf), 2007.
- [WPS09] Daniel Wagner, Thomas Pintaric, and Dieter Schmalstieg. the invisible train. [http://studierstube.icg.tu-graz.ac.at/invisible\\_train/](http://studierstube.icg.tu-graz.ac.at/invisible_train/), 2009.
- [Zha98] Zhengyou Zhang. A flexible new technique for camera calibration. Technical report, Microsoft Research, 1998.