Drawing Outside the Lines: Tracking-based Gesture Interaction in Mobile Augmented Entertainment

Wolfgang Hürst
Utrecht University, The Netherlands
Information and Computing Sciences
huerst@uu.nl

Ronald Poppe
Utrecht University, The Netherlands
Information and Computing Sciences
R.W.Poppe@uu.nl

Jerry van Angeren
Utrecht University, The Netherlands
Information and Computing Sciences
j.vanangeren1@students.uu.nl

***Abstract*—We present a proof-of-concept study for tracking-based gesture interaction in an augmented reality setting using tablets. By tracking a pen in front of a tablet using it's integrated camera, we are able to map certain motions to gestures, which in turn are used to interact with the application. A comparative user study investigates the feasibility and usefulness of our approach with a simple augmented reality board game allowing translation and drawing gestures to move and create virtual board pieces, respectively. In particular, we demonstrate that users can handle it (and to what degree) and that they enjoy it (and what they potentially dislike). The results from the 25 participants of our experiment provide both subjective and objective evidence of the potential of tracking-based gesture interaction for augmented reality gaming.**

Keywords*—*Augmented reality, handheld AR, tracking-based interaction, gesture-based interaction, AR interaction.

I. Introduction

The advent of augmented reality (AR) on mobile devices such as tablets and smart phones presents various opportunities for entertainment. For example, combining virtual and real game elements in an AR board game enables richer interactions, and allows for the design of engaging game experiences. One essential aspect of the experience is the way we interact with the game. When tablets and smart phones are used, interaction with the displayed media is generally done either on the touch screen or by tracking objects in front of or behind the screen using built-in cameras [1]. Touch screen interactions can be measured accurately and introduce no noticeable delay. On the other hand, they do not allow for interaction with the physical environment, and require the user to focus on the screen for both the display as well as the control of the game. Especially for interaction in AR, this limits the immersion in the created environment.

Tracking-based interaction can potentially address this issue. A hand or, for example, a pen can be tracked behind the screen showing the AR. This enables interactions with both the physical world (e.g., picking up actual game pieces) as well as the virtual parts (e.g., by moving around virtual pieces). As both physical and virtual world are visualized together on the device's screen, interactions in the AR are combined seamlessly. The fact that the tracked target (e.g., hand or pen) moves in the game's 3D space, instead of on a 2D screen, further adds to the immersion. Yet, there are also potential drawbacks to such a tracking-based interaction approach. Normally, it requires computer vision. It is thus computationally more demanding than touch screen interaction and considered less robust and accurate. This limits the functional potential. Moreover, it might negatively affect a user's experience of the interaction.

Considering this user experience, it may seem best to aim at a most "natural" interaction where manipulating virtual objects resembles interacting with physical ones as good as possible. Yet, above mentioned performance issues as well as the lack of tactile feedback, for example, when trying to grab virtual objects, can hinder the experience. We claim that gestures may provide an alternative, potentially even better interaction design. Gestures done by the user in the AR can be tracked and interpreted as different game actions depending on their location and context. Although such gesture-based interaction has its own set of problems (including again accuracy and robustness issues), devices such as Microsoft's Kinect have proven its potential to create engaging game play experiences.

The goal of this paper is therefore, to investigate the potential of tracking-based gesture interaction in an AR game played on mobile devices. Aside from general feasibility and performance, we are mostly interested in the created experience. While this type of interaction is not limited to entertainment applications, we particularly want to know whether people enjoy such gesture-based interaction. Focusing on such casual gaming, we designed a simple AR game that can be played using free-hand drawn gestures. The game is played on a tablet by manipulating and drawing shapes using a common pen, see Fig. 1. It acts as a proof-of-concept, used to evaluate whether gesture interaction is technologically feasible, as well as whether players enjoy this novel mode of interaction.

The remainder of the paper is structured as follows. We first discuss related work on interaction in augmented reality. Our proof-of-concept game is described in Section III. We then outline our user study and present the results. Finally, we discuss opportunities and limitations of the approach, and present avenues for future work.

II. Interaction in Mobile Augmented Reality

Gesture-based interaction has gained increasing popularity in recent years, partly due to the success of mobile devices and related touch screen gestures. The same accounts for tracking-based interaction due to the success of gaming devices such as Microsoft's Kinect. In the following, we discuss several related approaches in context of mobile computing, focusing on their potential for mobile AR gaming and justifying resulting design decisions for our experiments.

INTETAIN 2015, June 10-12, Torino, Italy

DOI 10.4108/icst.intetain.2015.259589

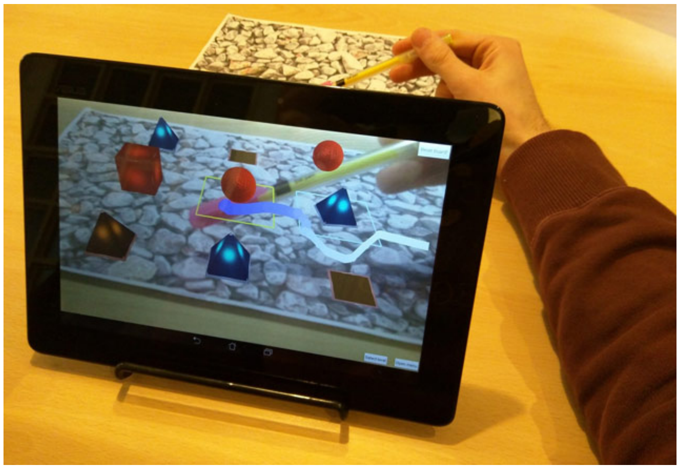


Fig. 1: The augmented reality game with virtual shapes and gesture-based interaction using a colored pen.

Markers are 2D or 3D objects that can easily be detected from a camera image. Typically, markers rely on sharp contrast. When a 2D marker is visible in an image, its position, orientation and scale can be determined [2]. Moreover, different markers can be identified. A marker can be linked to a virtual object. Moving the marker then can affect the object's position or orientation. Additionally, a marker can be attached to a physical object that is used to manipulate virtual objects. For example, the Magic Paddle is a piece of cardboard with a marker [3]. Moving the paddle around allows for the positioning and simple manipulation of virtual objects. While the detection of markers is typically robust, it requires that these markers are present in the environment. Moreover, more complex interactions such as changing the shape of a virtual object require additional input modes.

Using the hand as a means to position and manipulate virtual objects seems therefore appealing. The Tether prototype [4], for example, tracks a user's hand. The hand's position and orientation can be measured both in front of a tablet, as well as behind it, using the front and back cameras. Functionality is linked to finger movement. Additionally, the user's face is tracked to allow for more intelligent display of graphics. As such, the user can have less constrained interactions with the augmented space. This system strongly relies on the use of 3D markers attached to the hand and face. This limits the ease of use, which reduces its potential for casual gaming. The same goes for FingARtips [5], a prototype that detects finger movement using markers attached to the fingertips.

Fast movement of the fingers, in combination with movement of the camera, can cause marker detections to become less accurate, especially when it comes to detecting a marker's orientation. This issue can be mitigated when, for example, only color markers on the fingertips are used. Lee et al. present an AR approach in which the hand is tracked without any markers [6]. This makes their system applicable in a range of settings. However, the detection of the hand is based on skin color and the additional processing required made the approach slow and introduced inaccuracies which reduced the quality of the interaction.

For our user study, we decided to use a pen, making a compromise between flexibility and accuracy. Not only does a pen allow for relatively easy and accurate tracking, it is also a "natural" object that people are familiar with and commonly use. While it does not seem intuitive to use a pen for direct interaction with an object, using it for gestures (which in turn activate a related object interaction) makes sense. Furthermore, pens are the natural choice for drawing – an action that can be used in AR to create new virtual objects.

Free drawing requires tracking over time. While drawing in AR can also be achieved by using a touch screen and considering the orientation of the tablet or smart phone [7], we focus on free drawing in the 3D space directly. That is, once a pen (or any other object, e.g., one's index finger) is detected, it can be tracked over time as a means to draw freely in 2D or 3D. 2D drawings can be used to create and manipulate 3D objects, for example, by drawing their contours [8]. Yet, drawing them in mid-air in a mobile AR setting turned out to be too difficult for most users, especially untrained ones [9]. The authors therefore introduced a virtual grid to aid the drawing. While this improved the quality of the drawing, it introduces limitations on the types of drawing that could be made; for example, round shapes are not supported by the grid.

Instead of free drawing 3D shapes, one could of course build these out of 3D primitives. These primitives can then be moved, extruded or combined. Also, 3D primitives need to be selected from a list. This requires require additional input modalities. Using on-screen menus is one option. Alternatively, these menus can be presented in the 3D environment. However, both approaches require that the user switches between drawing and selecting mode, which might be cumbersome especially when switching occurs often.

As an alternative, different functionality can be provided directly from a hand. Different hand poses can be recognized [10], but the accurate detection of these poses typically requires the use of markers (e.g., [5]). Alternatively, hand gestures could be used. These are movements made using the hand, fingers or object that is directly manipulated by the hand. Instead of recognizing a specific hand pose, this approach only requires that a single point (e.g. fingertip, pen, colored dot) is detected. Our approach utilizes this idea by tracking the tip of the used pen. Motions of the pen are interpreted as gestures, which in turn enable certain interactions depending on the type of gesture, its location, and context.

An overview of possible techniques to recognize user's gestures is discussed in a survey focusing on hand, arm, body, face and head gestures [11]. Better recognition results are obtained if the gesture recognition is invariant to rotation, scale and position. While gestures are commonly used, performing them in mid-air introduces discomfort to the user, based on the duration and location where a certain gesture has to be performed [12]. Therefore, it is important that the gestures are intuitive and can be detected quickly and reliably.

A common problem with gestures is that users are often uncertain about a system's state or the actual input performed by them (e.g., do my motions done in mid-air really match the intended shapes?). Feedback could help, but there is typically a lag between the performance and the detection of a gesture. Yet, in context of touch screen interactions, the Ripples system [13] demonstrated that one can overcome this issue by converting this lag into a design element. A contact visualization frameworks provides feedback related to possible interactions and whether interactions are performed successfully. The system showed promising results with fewer errors and quicker interactions.

Based on these previous works, we set out to investigate whether free-drawing hand gestures can be a useful and entertaining way for interaction in mobile AR applications. For this, we developed a game in which users can manipulate objects of certain shapes, as well as create new ones by making gestures. Instead of relying on markers, we track the tip of a colored pen held by the user. In order to deal with potential lag and usability issues, we further evaluate if the visualization of traces, which were successfully introduced for touch screen interaction [13], has potential benefits in a tracking-based AR context as well.

III. Tracking-based Augmented Reality Game

In this section, we present our augmented reality game that uses gestures recognized from tracking a pen. We first discuss how the pen tracking is realized, followed by a description of the game and the interaction with the user. Note that this pen tracking-based interaction concept can be the input for a wide range of applications and games. In this light, the implemented game should be seen as a proof-of-concept that is simple enough to be evaluated systematically, while it has sufficient entertainment value.

A. Pen Tracking

Because it seems a natural tool for “drawing-like” gestures, and to avoid difficulties with detecting fingers with or without markers, we have opted to track the tip of a pen instead. We require that the color of the pen is different from that of the background. The specific color does not matter. In the remainder, we have used a pen with a pink tip.

The pen tracking software runs on the tablet or smart phone that is used for the game. The processing starts by capturing a frame from the camera. When the resolution of the image is larger than 640×480, it is resized to a size equal or smaller than 640×480. To avoid having to interpolate pixels, we only decrease the resolution with a power of two in either direction. The resulting image is then converted to HSV color space. In this space, the hue channel encodes the color and we apply a double threshold with pre-determined minimal and maximum values. We also apply a threshold on the saturation and value channels by setting minimal values. This ensures that only bright colors are retained. In the thresholded image, retained pixels should correspond to the tip of the pen.

To suppress potential noise, we apply erosion on the binary image. Finally, we search for the largest connected component and determine the 2D center using central moments. This center corresponds to the center of the tip of the pen and will be used as input in the game.

If an estimation is available from the previous frame, we limit the search space to a rectangle around the previously found center. This significantly reduced the processing time and results in an increase of 5 fps on average. The width and height of the rectangle are determined such that the pen tip will be inside the area even when fast movements are made.

With no previous estimate, the various steps combined take in the order of 12–50ms, depending on the device used. Given that other processes are also performed on the respective devices, the resulting frame rate is between 10 and 25 fps. Typically, a previous estimate is available, which increased the frame rate to 15–30 fps.

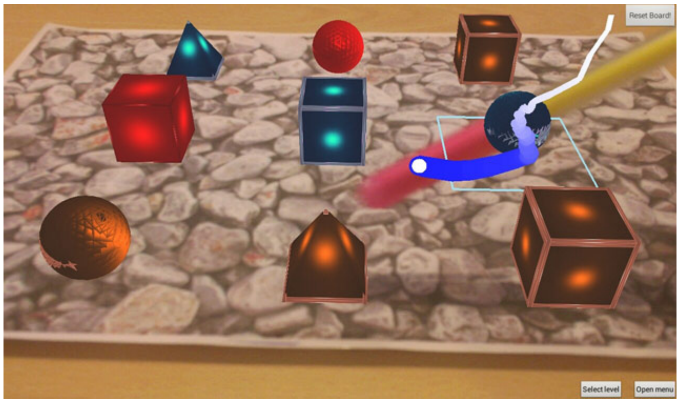


Fig. 2: Screenshot of the created game environment.

B. Augmented Reality Game

To test our interaction paradigm, we decided to use an AR board game as proof-of-concept implementation. Playing with any kind of board game requires interaction with game pieces. In traditional board games, physical pieces could be placed, moved and removed from the board. Virtual board pieces, in augmented or virtual board games, may also be placed on the scene (creation), removed from it (deletion), or modified, for example by moving them or by transforming them in a way which may be impossible in the real world, such as scaling or recoloring. There are various ways to realize such creation or modification actions. As said, in this research, we are investigating whether gestures are suitable for these basic interactions and in which way. In particular, we are looking at *drawing gestures* to create objects and *translation gestures* to move them. Gesture interaction is realised via pen tracking.

The created board game consists of a grid of tiles (3×3), see Fig. 2 (only the active tile currently passed by the pen is visualized). The grid is draw on top of a custom marker, which is typically placed on a flat surface. Each tile in the grid can contain one of three game pieces or shapes: a ball, a pyramid, or a cube. A tile can also be empty. Each tile, or the shape on it, can have one of three colors: red, blue, and orange. The goal of the game is to score points by lining up three of the same shapes, either horizontally or vertically, using a translation gesture (to switch two shapes) or a drawing gesture (to create a new one). Once a match of three shapes is found, they are removed from the board and new shapes are placed on the their tiles. For each match, the player is awarded 10 points, or 20 when they also have the same color.

The game starts with randomly placed shapes in randomly selected colors. It ends when there are no more possible moves, that is, when no shape can be drawn on any of the tiles, and

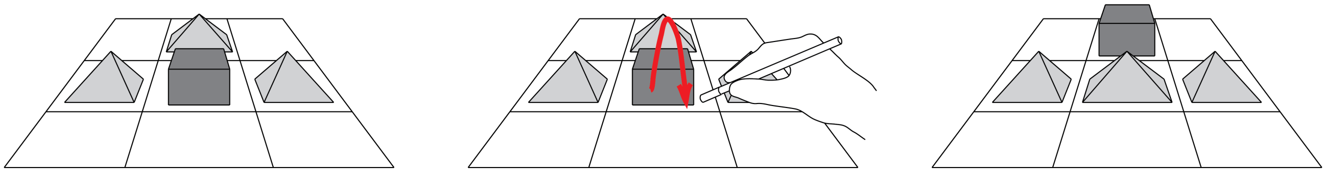


Fig. 3: Translation gesture (center) which swaps two objects to create a set of three or more objects with the same shape.

switching any two tiles does not result in the lining up of three matching shapes. In this case, the user can replay the game by resetting the board. This also resets the score, which is stored as a new highscore if it is higher than the previous highscore.

C. Implementation

The game builds on several well-known open-source libraries. Vuforia [14] is used to capture a frame. If the custom marker is detected, its position and orientation relative to the camera is determined. Based on this, a view matrix is created. The frame is passed on an image processing class, which contains the functionality described in Section III-A. OpenCV [15] is used to detect the pen tip. Based on the gesture input, the game logic is processed. Finally, the game board with all objects is drawn using the computer view matrix. We use OpenGL ES 1.1 [16] for this purpose.

D. Gesture Interaction

As said, there are two options for lining up shapes and thus scoring points. In "switching mode", the shapes on two adjacent tiles are switched. In "drawing mode", the player draws a shape on an empty tile. The game starts in switching mode. We discuss the selection, switching and drawing on the tiles and shapes subsequently.

1) Selecting a tile: The 2D screen location of the tip of the pen is first converted to coordinates used in the 3D augmented world. By projecting the location onto the grid, it can be determined whether the pen hovers over one of the tiles. If this is the case for at least five frames (corresponding to roughly 0.15–0.3s), the tile is selected. Once a tile is selected, a border is shown around the tile (see Fig. 2 for an example).

2) Switching two tiles: A player can switch two tiles by first selecting one tile and then performing a translation gesture. Once the first tile is selected, the user moves the pen to a tile that is horizontally or vertically adjacent, and subsequently moves back to the first tile. See Fig. 3 for a schematic overview of the process. If the second tile is not adjacent, the first tile is deselected and, when the pen hovers on it for at least five frames, the current tile is selected.

Upon performing the gesture, the game switches the two tiles. If this results in one or both of the two shapes being in a matched group, all shapes from that group are removed. If this is not the case, the two shapes are immediately switched back, thus returning to the starting state. A player can only switch tiles with shapes, not with empty tiles.

3) Drawing a new shape: When a player selects an empty tile, the game mode is set to drawing mode. A new shape can be created by making a drawing gesture. We included three different gestures that would create the sphere, pyramid, and cube, respectively: a circle, triangle, and square (see Fig. 4). Note that the gestures can be considered the 2D versions of the respective 3D board pieces.

Gestures are recognized from traces, that is, sequences of pen tip detections over time. We constrain the maximum length of a trace to 70 frames (corresponding to approximately 2-5s). If longer sequences would be allowed, there is an increased chance of incidental creation of shapes whereas the player intends to change to switching mode. On the other hand, if the limit of the trace length is too strict, gestures have to be drawn very quickly, which requires skill.

If the drawn path crosses itself within the trace, the sequence between the two crossed points is considered a gesture movement. This sequence is then normalized since gestures can be drawn at different scales. Based on a set of ten templates per gesture, the most probable class is selected using nearest neighbor matching. If the closest template is below a predetermined threshold, a gesture is recognized. If this is not the case, the game continues in switching mode.

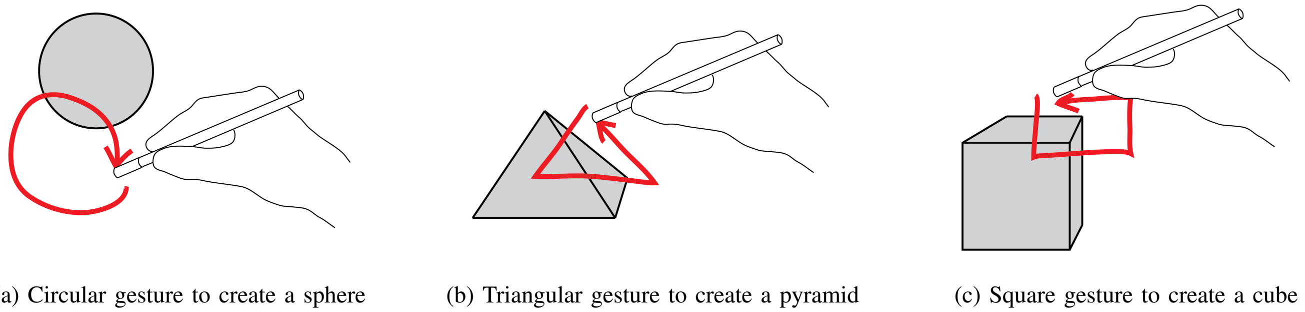
Once a gesture is recognized, the corresponding shape is created and placed on the empty tile in a random color. If the novel shape is part of a matching set, its shapes are removed and replaced by new shapes or empty tiles. If the novel shape is not part of a matching set, the game switches to switching mode and continues. The player can then continue to form matching sets by switching or drawing other shapes.

IV. User Study Setup

To test the feasibility of our proposed interaction method, we set up a controlled user study. The aim of the study was two-fold. First, we wanted to analyze the objective quality of the approach, and second the subjective experience of the players, because we are not only interested if this is a feasible but also an enjoyable way of playing AR games.

A. Experimental Setup

We split up the user study into two parts. In the first part, users completed a fixed number of trails in which they had to perform translation and drawing gestures. This controlled setup enabled us to measure how well they were able to perform these gestures. In the second part, they played the game for a couple of minutes, thus resulting in a more realistic gaming situation in order to verify gameplay experience. To quantify



(a) Circular gesture to create a sphere

(b) Triangular gesture to create a pyramid

(c) Square gesture to create a cube

Fig. 4: Drawing gestures which can be used to create new virtual objects after an empty cell is selected and the application switched towards a drawing state.

this, the second part was completed with a questionnaire aimed at subjective user experience and a closing informal interview.

Gestures are an essential part of the game, and the user should be able to make these gestures confidently. To this end, hypothesized that visual feedback on the detected pen movement could help the user in performing the gestures accurately. If, for example, they would see that the square that they intended to draw would not look as such, they could alter their movement of the pen. We choose a "trace" visualization, with the detected pen tip shown bright and a fading tail of previous detections, see Fig. 2. The detection and visualization of the pen tip lags behind a bit on the actual pen movement. Due to the relatively heavy processing steps of the computer vision algorithm, this is unavoidable. A user might over-compensate his movement because of this lag.

To investigate the influence of the trace visualization on the performance and experience of the user, each user performed the experiment both with and without trace visualization. Trace therefore is a within-subject variable. We further hypothesized that a learning effect might occur. As users play the game longer, they might get more skilled. As the trace helps to understand how the pen movement is detected, starting with visualization could improve the learning rate of the users. To be able to test this hypothesis, we alternated the order in which they performed the two trace conditions between test subjects.

A final factor in the experiment was the type of device that was used. We decided to use three different devices with slightly different characteristics – most importantly their screen sizes: An Asus MeMO Pad Smart tablet (10.1 inch display, 1280×800 pixels resolution, 5MP camera), an HTC Nexus 9 (8.9 inch display, 2048×1536 pixels resolution, 8MP camera) and an Asus Nexus 7 (7 inch display, 1920×1200 pixels resolution, 5MP camera). We did not systematically assign users to devices, but instead let them choose which device they wanted to use.

B. Procedure

Upon starting the experiment, each user was seated at a table on which a highly textured marker was placed. In front of the marker, a tablet was placed on a stand such that it captured the marker and displayed the virtual objects so the user could clearly see them. Fig. 1 shows the setup of the experiment, in this case with trace visualization. Participants used a pen with a pink tip to make the gestures.

We conducted the experiments with either one, two or three users at the same time. When multiple users participated simultaneously, the different phases of the experiment were synchronized across the subjects.

An introduction talk briefly explained the game and the used interaction techniques to the participants. Then, they were assigned to start with either the trace visualization, or no visualization. When several users played simultaneously, all were assigned the same starting condition. To help the participants to get used to the way of interacting, they completed a tutorial: They were asked to perform two translation gestures, followed by each of the three drawing gestures (circle, triangle and square). At this point, no data was logged.

After that, users had to perform the translation gesture six times during which the objective data was recorded. We made use of predefined sets of layouts of shapes to ensure that comparable data is being captured for all participants. The next step concerned the creation of new objects in which each gesture is performed two times in a random order. Users were instructed to create the shape that formed a matching line. We presented the layout of shapes in a pre-defined manner so that only one gesture could be performed to create a matching set. When the wrong shape was detected, the user had to perform the drawing gesture again. We imposed a maximum of five attempts, after which the experiment continued with the next gesture.

At the end of the first condition, users were asked to play the game for five consecutive minutes with the aim to score a highscore. The score was not displayed, to prevent users playing simultaneously to communicate about it. We expected that this could influence how the users would play the game, and how they would rate it. Hence, their were told their scores only after the whole experiment was finished and they provided all their feedback. When no moves were possible, the user could reset the board to start with new random objects. The score would then also be reset. After these five minutes, the first part of the questionnaire was filled in.

After filling in the questionnaire, the users performed the translation and drawing gestures, and the five minutes free play, in the other condition (with or without trace). After completion, they filled in the questionnaire once more, followed by some questions regarding the differences between the two conditions. Finally, an informal interview was held with all participants. They were asked to motivate their ratings, and to indicate positive and negative aspects of the game and gesture interaction.

C. *Measurements*

For our objective measurements, we calculated the percentage of correctly performed translation and drawing gestures. As we knew in advance which gesture had to be made, these measurements could be derived directly from the log files. We use percentages rather than real counts as users could re-try drawing gestures up to five times. We also recorded the average duration spent on a drawing gesture. Quicker movement could be seen as a more natural way of input. Given that the three devices used had very different frame rates, we also report the average number of frames.

In addition to the objective performance measures, we asked the users to fill in two types of brief questionnaires. After each condition, they were asked to rate the game experience and how much they felt in control (overall, per interaction type, i.e., translation and drawing, and in the latter case also for each shape). Finally, after finishing the second condition and questionnaire, they were asked to make a comparison between the two versions. We asked them to rate whether or not they preferred the trace visualization and let them comparatively rate gameplay experience and level of control again. We also asked if they experienced any discomfort during the experiment.

Finally, we recorded the remarks made in the informal interview that concluded the experiment along with observations made during the tests. During the interview, we were mainly interested in the users' opinions regarding which aspects of the game they liked, and which could be improved.

D. *Participants*

A total of 25 users (20 male, 5 female), aged 21 to 27 years (average 23.16 years) participated in the user study. Ten users had experience with augmented reality in some form. The majority (23) of the user were right-handed. None of the participants were aware of the focus of this study or had any other pre-knowledge of it. We purposely decided to restrict the study to subjects of younger age; first to have a more homogeneous test group, second and most importantly, because we are interested in gameplay experience. All subjects had some affiliation with digital games (although not necessarily mobile ones) and are thus good representatives of the expected user target group.

V. Results and Discussion

In the following, we discuss the objective and subjective measurements, respectively. The objective measures are informative of the performance of the gesture recognition whereas the subjective evaluation should give us insight into the experience of the game. Since each participant did the experiment twice (with and without trace), we compare results from the two subsequent parts (i.e., Part 1 = averaged results over all first rounds, and Part 2 = averaged results over second one, independent of what trace visualization was used) to see if there are any learning or habituation effects. In order to investigate the provided feedback had any influence on performance and experience, we also compare the averaged results over all tests with trace visualization versus the ones without one. Finally, we briefly describe the outcome of the informal interview and discuss the game as a whole.

The experiment took around 30–45 minutes per group, depending on the present participants. The users could choose which device they wanted to use. Given its smaller screen and lower frame rate, the Nexus 7 was only used when three participants performed the experiment at the same time. In total, the Nexus 7 was used five times, the Nexus 9 seven times and the Asus MeMO Pad Smart 13 times. We discuss device differences in Section V-B.

TABLE I: Average Percentage Of Correctly Performed Gestures Over Participants, Tasks & Trials

	Translation gestures	**Drawing gestures**
Part 1	60.08	67.12
Part 2	74.06	76.53
Average	**67.07**	**71.57**
With trace	66.40	72.12
Without trace	67.74	71.01

A. *Gesture Performance*

Every participant successfully performed the six translation gestures. Yet, sometimes it took more than one attempt to do it correctly. In Table I (first column), it can be observed that the total percentage of correct translation gestures for both parts is 67.07%. This number is rather low, and is mainly caused by the low performance in the first part of the study, where the participants use the interaction technique for the first time (cf. first two rows of the table). From the first to the second part, the performance increases with 23.27% relatively (13.98% overall). We expect that this difference is partly because participants gain experience. We also noticed that some participants tried to switch two tiles that would not lead to a matching group. A wrong gesture could also be detected when the user moved the pen due to gripping. These actions were mainly observed in the first part. The increase in performance seems to confirm that the approach is indeed feasible and manageable, but a certain learning time is required.

From Table I (bottom two rows) we further notice that differences with and without trace visualization are small. We hypothesized that there might be an interaction effect for trace and part as participants could learn quicker when the trace is shown in the first part. To investigate both the learning and the interaction effect, we conducted a repeated measures ANOVA with *part* as the repeated variable and *trace order* (trace in first part or in second part) as between-subjects variable. The average number of correct translation gestures was the dependent variable. We found a significant main effect of part on the performance of the translation gestures ($F(1, 23) = 23.554, p < 0.001$). However, we did not find a significant interaction effect, nor an effect of the trace order on the correct performance of the translation gestures.

Not all drawing gestures were completed successfully by the participants. One subject failed a triangle and two failed drawing a square gesture. All three started without trace visualization and succeeded performing the gestures while having a visual trace, suggesting that the visual feedback may have a positive learning effect. The right column of Table I shows the average results from the drawing gestures. We again performed a two-way independent ANOVA with part and trace as independent variables, and the percentage correctly performed gestures as the dependent variable. As before, gestures in part two were performed significantly better compared to those in part one, indicating a main effect for part ($F(1, 23) = 4.867, p < 0.05$). The increase in performance is 9.41% (from 67.12% to 76.53%). The difference with and without trace is less than one percent, and was not found to be significantly different, thus not confirming the above mentioned positive learning effect. Better performance in the second part seems mostly related to more experience rather than the provided visual feedback. Also, no interaction effect was found between trace and part ($F(1, 23) = 0.091, p = n.s.$).

TABLE II: AVERAGE PERCENTAGE OF CORRECT DRAWING GESTURES OVER PARTICIPANTS, TASKS & TRIALS

	Circle	**Triangle**	**Square**
Part 1	89.29	69.01	52.17
Part 2	83.33	89.29	62.50
Average	**86.21**	**77.95**	**56.98**
With trace	92.59	75.76	56.82
Without trace	80.65	80.33	57.14

Next, we turn our attention to the performance of individual shapes. Table II summarizes these results, for parts one and two (top two rows), both with and without trace visualization (bottom two rows). Overall, the circle gesture was performed best and the drawing of the square proved to be most difficult.

From the first to the second part, we see an improvement of the performance of the triangle and square gestures. This is especially noticeable for the triangle, with 20.28% increase. The circle gesture is performed somewhat worse in the second part. This is mainly due a difference in the number of incorrectly performed gestures in the group of participants that started with the trace visualization. They made seven incorrect circle gestures without trace visualization in part two, compared to only one mistake in the first part with trace visualization.

These numbers are also reflected when looking at difference in performance between the two trace conditions. With 80.65%, the performance without trace visualization was much lower than the 92.59% achieved with the trace shown. For the other two gestures, the difference is not that apparent, although the triangle is performed approximately 5% better without trace.

To analyze the effect of learning and trace visualization we conducted, for each shape, a repeated measures ANOVA with part as repeated measure and trace order as between-subjects variable, as before. We used the percentage of correctly performed circles, triangles and squares, respectively, as dependent variables. None of the main and interaction effects appeared to be significant. We found a marginal learning effect for the drawing of triangles though ($F(1, 23) = 3.762, p = 0.065$).

TABLE III: AVERAGE PERCENTAGE OF CORRECT GESTURES PER DEVICE OVER PARTICIPANTS, TASKS & TRIALS

	Translation gestures	**Drawing gestures**
Nexus 7	73.84	70.24
Nexus 9	52.11	65.35
Asus MeMO Pad Smart	69.47	77.11

B. Devices

The three devices used had different framerates. As a result, the amount of frame used to create the drawing gestures differs as well between devices. The Nexus 7 and Asus MeMO Pad Smart show rather similar results with an average of 145 and 169 frames per gesture, respectively. On the other hand, an average of 308 frames is required when using the Nexus 9. This is a lot higher, and can be explained by the better hardware and thus faster image processing. When the users move the pen around, the Nexus 9 processes more captured frames which leads to the higher value.

On the other hand, if we look at performance as shown in Table III, the Nexus 9 actually shows the lowest performance. Yet, this is especially caused by two participants who performed 12 and 15 incorrect translations, respectively, and thus biased the results. They also required more attempts during the drawing gestures. The best performances for the translation gestures are obtained when using the Nexus 7, while the Asus MeMO Pad Smart shows the best results for the drawing gestures. Overall, the performance of the three devices used in the experiment is not too different, suggesting that neither screen size nor device performance have a major impact on the interaction experience.

TABLE IV: AVERAGE GAME EXPERIENCE (ALL PARTICIP.)

	With trace	**Without trace**
Overall	4.52	4.76
Translation gestures	4.88	5.16
Drawing gestures	3.96	3.72

C. Questionnaire

With the questionnaires, we aimed at measuring how participants felt about the gesture interaction. We specifically asked them about their game experience and the level of control they experienced both when playing with and without trace visualization.

The ratings (on a 7-point Likert scale) are summarized in Table IV. Overall, the game experience was rated relatively high with an average of 4.64. A small difference between the two trace conditions was not found to be significant, as shown by a paired-sampled t-test ($t(24) = 1.238, p = n.s.$). There seems to be a slight preference for making translation gestures without trace visualization. But differences between the trace conditions were not found to be significant – neither for translation nor drawing gestures. Participants did however appreciate making translation gestures more. A paired-samples t-test between the averages for translation and drawing gestures over both trace conditions shows a significant effect ($t(24) = 3.952, p < 0.001$).

Ratings for level of control follow a similar trend, as can be seen in Table V. They are not significantly different for the two trace conditions. However, participants did rate their level of control when making translations significantly higher than when making drawing gestures ($t(24) = 7.462, p < 0.001$). With almost 2 points, the difference is twice as large as for the game experience. This shows that participants found it easier to make translation gestures. For drawing gestures, trace visualization was again not found to be a significant factor. We do see differences between the rating for individual shapes however. The level of control for circle gestures was rated best, for squares least. These numbers reflect the performance scores reported in Table II, and are thus to be expected as a lower level of control is likely to lead to more mistakes in performing the gesture, and the other way around.

TABLE V: Average Level Of Control (All Particip.)

	With trace	**Without trace**
Translation gestures	5.48	5.28
Drawing gestures	3.56	3.24
Circle	4.76	4.92
Triangle	4.04	4.28
Square	3.28	2.92

D. Discomfort and Informal Interview

From the 25 participants, eight experienced some discomfort while performing the user study. Most of them mentioned discomfort related to their arm, which became tired or felt heavy. These participants did not place their arm on the table but instead kept it unsupported in mid-air. The placement of the camera also caused these results when users were right-handed. This was most notable for the Nexus 9 and Nexus 7, as both have the camera on the top left, in contrast to the Asus MeMO Pad Smart, where it is placed in the top center. This causes the user to have some difficulty with interactions on the left side when the tablet is placed in the tablet stand. These observations are in line with other researchers [1] who also concluded that although feasible, this type of interaction can and should not be done intensively over a longer period of time. Given that in a “full” game, people will mix the interactions tested here with others (e.g., moving physical objects, waiting for other players in turn-based board games), we do not expect it to be a major issue, but it is an important factor to be considered in the game design.

During the informal interview, when asked about the general experience, most participants said they enjoyed the proposed interaction technique. Regarding the trace visualization, it was frequently remarked that the trace was rather distracting. This was mainly caused by the rather long trace with slow fade. This caused a large part of the board to be blocked, which was found to be distracting when performing translation gestures. While performing the drawing gestures, users tended to watch the trace and thus move slower instead of focusing on the creation of the gesture with faster movements. Another remark was that the users did not expect that the center of the pen tip will be used for tracking. Rather, they aimed at the extremity. Overall, the majority of the users seemed to prefer the implementation without trace visualization.

Some participants were frustrated by the false recognitions. Adding an undo button was suggested, as well as a pop-up dialog in which the recognized shaped needs to be confirmed. Also, they suggested to display a cursor on the last recognized location instead of having a visual trace. This would allow the player to see the detected point without being distracted and without the trace blocking the tiles.

E. Discussion

If we take a combined look at the achieved performance, the results of the questionnaire, and the feedback from the informal interview, the majority of the users seemed to have enjoyed the interaction technique. Users are able to perform the gestures properly to some extend and they improved during the short amount of time they interacted with the game. From observations, we understood that this learning effect also continued while the users were playing the game. As with all new technology, subjective comments have to be treated with care, as there is often a newness or ”coolness” factor biasing them. Yet, the fact that the members of the user group in our study all had an affiliation to gaming also suggests they are rather critical to both performance as well as gameplay experience characteristics. It is therefore safe to assume that our results give a realistic assessment of our approach.

We found some differences for the two types of gesture: translation and drawing. Overall, the drawing gestures are rather hard to make, which is especially true for the square gesture, whereas the simpler triangle and circle gestures could be performed reasonably well. Users frequently reported becoming frustrated by repeated misclassification. They also reported lower levels of control for the square gesture in particular. Even though the performance for these gestures improved over time, this suggests that it is preferable to choose gestures that are easier to perform. It should further be noted that gestures generally do not scale, that is, users can only remember a certain amount of them. Further research is therefore needed to verify what complexity of gestures is still feasible, and furthermore, how the resulting (likely small) set of simple gestures can be used to create a powerful and satisfying interaction experience. Context-dependent interpretation of simple gestures (as already done in our case when distinguishing between translation and drawing mode) plus appropriate and intuitive feedback may be a promising approach.

The fact that we did not find any significant difference between playing the game with and without trace visualization was a bit disappointing considering the good results of the Ripples system [13] with touch screens (although the fact that people were indeed able to draw gestures reliably even without feedback is positive and encouraging). There was generally no tendency of favoring one condition over the other. While the feedback on their movement was appreciated, the trace also blocked part of the users’ view, which they found distracting. Yet, although not statistically significant, observations on some mistakes still suggest some potential in this approach, especially for the more complex drawing gestures. In addition, several participants remarked that a different trace visualization could have helped them. It should be noted in this context, that our visualization was rather simplistic compared to the sophisticated and detailed representation used in [13].

Due to a limited amount of data available for some devices, we did not look into the game experience and rated level of control per device. This is left for future work. Based on our observation in this user study, we expect that a larger screen increases the overall enjoyment of the game. Also, the lag between performing a gesture and having it visualized on screen would be smaller when a faster device was used. We expect that this will add to the enjoyment of the game, and could potentially increase the level of control for making the gestures.

VI. Conclusion

In this paper, we have presented a novel way of interaction with mobile augmented entertainment applications: tracking-based gesture interaction. By tracking a pen tip, users are able to perform actions in the augmented environment. We implemented a proof-of-concept game in which translation and drawing gestures were used to move and create shapes on a virtual board. In a user experiment, we evaluated both the performance of gesture recognition and the subjective rating of game experience of level of control.

Participants were able to correctly perform gestures in approximately 70% of the cases. However, we found a significant learning effect which caused an increase in recognition of translation and drawing gestures of 13.98% and 9.41%, respectively. Given the short amount of time available, we expect that this learning can further increase classification rates. We reported different performance scores for the three gestures used. Circles, triangles and squares could be recognized with 86.21%, 77.95% and 56.98% accuracy, respectively. Choosing gestures that users can reliably make is advisable. Visual feedback regarding the pen tracking did not show a significant effect on the performance.

Participants rated the overall game experience with a 4.64 on a 7-point Likert scale. In general, they seemed to enjoy the interaction, despite some frustrations when gestures were repeatedly misclassified. Translation gestures were more appreciated than drawing gestures, and also the level of control when making translation gestures was rated higher. For individual gestures, the rated level of control was in line with the performance scores. Again, this indicates that the choice of gestures is important.

We experimented with a trace visualization to give feedback to the user regarding the tracking of the pen tip. We did not find any significant differences in the performance with and without trace visualization. Also, participants did not appreciate either of the two settings. We expect that is partly due to some drawbacks of the implemented visualization style. Most notably, the trace occluded parts of the view, which distracted the user.

The overall positive responses from the participants of the experiment are motivating. Moreover, the analyses of the performance of the different gestures enables us to improve the game experience. We plan to address current shortcomings in several directions. First, the feedback to the users can be improved. By showing feedback, yet not distracting the user, we intend to increase the experienced level of control. Second, we plan to improve the pen tracking. Most notably, we will change the tracking point from the center of the pen tip to the extremity. Third, we plan to develop a novel game in which the advantages of the gesture-based interaction method can be exploited. Currently, we did not take advantage of the possibility of handling both virtual and physical objects in the same space. In future work, we tend to address this. We expect that these improvements will further add to feasibility and enjoyment of our interaction paradigm.

References

[1] W. Hürst and C. van Wezel, “Multimodal interaction concepts for mobile augmented reality applications,” in *Proceedings of the International Conference on Advances in Multimedia Modeling (MMM)*, 2011, pp. 157–167.

[2] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana, “Virtual object manipulation on a table-top AR environment,” in *Proceedings of the International Symposium on Augmented Reality (ISAR)*, 2000, pp. 111–119.

[3] T. Kawashima, K. Imamoto, H. Kato, K. Tachibana, and M. Billinghurst, “Magic Paddle: A tangible augmented reality interface for object manipulation,” in *Proceedings of the International Symposium on Mixed Reality (ISMR)*, 2001, pp. 194–195.

[4] D. Lakatos, M. Blackshaw, A. Olwal, Z. Barryte, K. Perlin, and H. Ishii, “T(ether): Spatially-aware handhelds, gestures and proprioception for multi-user 3D modeling and animation,” in *Proceedings of the ACM Symposium on Spatial User Interaction (SUI)*, 2014, pp. 90–93.

[5] V. Buchmann, S. Violich, M. Billinghurst, and A. Cockburn, “Fingartips: Gesture based direct manipulation in augmented reality,” in *Proceedings of the International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia (GRAPHITE)*, 2004, pp. 212–221.

[6] M. Lee, R. Green, and M. Billinghurst, “3d natural hand interaction for ar applications,” in *Proceedings of the International Conference Image and Vision Computing New Zealand (IVCNZ)*, 2008, pp. 1–6.

[7] T. Ha and W. Woo, “ARWand: Phone-based 3d object manipulation in augmented reality environment,” in *Proceedings of the International Symposium on Ubiquitous Virtual Reality (ISUVR)*, 2011, pp. 44–47.

[8] T. Igarashi, S. Matsuoka, and H. Tanaka, “Teddy: A sketching interface for 3D freeform design,” in *Proceedings of the Conference on Computer graphics and Interactive Techniques (SIGGRAPH)*, 2007, pp. 409–416.

[9] W. Hürst and J. Dekker, “Tracking-based interaction for object creation in mobile augmented reality,” in *Proceedings of the ACM International Conference on Multimedia (MM)*, 2013, pp. 93–102.

[10] X. Cao and R. Balakrishnan, “VisionWand: Interaction techniques for large displays using a passive wand tracked in 3D,” in *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*. ACM, 2003, pp. 173–182.

[11] S. Mitra and T. Acharya, “Gesture recognition: A survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 3, pp. 311–324, 2007.

[12] D. Ahlström, K. Hasan, and P. Irani, “Are you comfortable doing that? Acceptance studies of around-device gestures in and for public settings,” in *Proceedings of the International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI)*, 2014, pp. 193–202.

[13] D. Wigdor, S. Williams, M. Cronin, R. Levy, K. White, M. Mazeev, and H. Benko, “Ripples: Utilizing per-contact visualizations to improve user interaction with touch displays,” in *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, 2009, pp. 3–12.

[14] “Vuforia developer portal,” https://developer.vuforia.com.

[15] G. Bradski, “The OpenCV library,” *Doctor Dobbs Journal*, vol. 25, no. 11, pp. 120–126, 2000.

[16] A. Munshi and J. Leech, “OpenGL ES common/common-lite profile specification, version 1.1.12,” Khronos Group, Tech. Rep., 2008.