Highlighting Techniques for Real Entities in Augmented Reality

Sebastian Fuchs, Mario Sigel and Ralf Dörner
Hochschule RheinMain, Wiesbaden, Germany

Keywords: Highlighting, Augmented Reality, Particle Systems, AR User Interfaces.

Abstract: One fundamental task in Augmented Reality is to highlight specific real world entities or small areas in the user‘s view. For this purpose, a total of 9 novel highlighting techniques were designed and implemented. For comparison, they are complemented by two conventional techniques often used in practice. This paper presents the techniques and evaluates them based on a user study in which 23 participants were asked questions concerning the degree of attention as well as disturbances caused by the highlighting techniques. The user study revealed that particle systems are a successful tool to effectively draw attention to a highlighted entity without causing annoying distractions.

1 INTRODUCTION

In Augmented Reality (AR), it is a fundamental task to highlight an entity in the user's environment. For instance, highlighting can help a user to follow assembly instructions, highlighting can draw the user's attention to a specific object in a learning application, or highlighting can be used to reference entities in a collaborative setting. In principle, there are two different kinds of entities that can be highlighted in AR: virtual entities and real entities. Usually, virtual entities are easier to highlight as the AR system has not only the information available about their position and orientation, their geometry, and their appearance but can also manipulate them quite effortlessly (Milliron et al., 2002). Highlighting real entities is more challenging and the paper focuses on this task.

In practice, highlighting by changing the color of an entity or drawing a frame around it are commonplace; examples can be found in (Möller et al., 2012) and (Li et al., 2013). While these highlighting techniques are simple to implement, they have several drawbacks. First, a frame might occlude important information. Second, changing the appearance of an object might result in a loss of information. Third, some highlighting techniques might only be applicable to virtual objects resulting in inconsistencies as virtual and real objects are highlighted differently. For example, it might not be feasible to change the color of a real entity as convincingly as the color of a virtual entity. Moreover, it is desirable to have a larger variety of highlighting techniques available. This expands the design space and increases the chances that a designer can find a more suitable highlighting technique for a specific use case. For instance, it might be beneficial to use different highlighting techniques for highlighting humans or avatars as opposed to objects. Different highlighting techniques can also be used to group entities or to express different levels of urgency or importance. This is our motivation to conceive additional highlighting methods. The advances in hardware (e.g., increasing graphics processing power even on handheld devices used for AR or availability of high-resolution cameras and depth sensors) as well as in computer vision algorithms (e.g., for object segmentation and finding contours of real entities in a video image) provide a larger basis for exploring novel highlighting techniques.

The main contribution of this paper is the presentation of nine novel techniques for highlighting real entities in AR and the results of their evaluation. These methods do not only use color or enclosure as visual variables to cater to the user's pre-attentive perception (Ware, 2004). They rely on the use of particle systems (Reeves, 1983), distortion, and the segmentation of the object's contour. Comparing them to traditional techniques, a user study has been conducted with the highlighting techniques introduced in this paper. Criteria for the evaluation are the degree of the user's attention that is achievable as well as the extent of possible distractions that are

Fuchs, S., Sigel, M. and Dörner, R.
Highlighting Techniques for Real Entities in Augmented Reality.
DOI: 10.5220/0005674002570268
In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 1: GRAPP*, pages 259-270
ISBN: 978-989-758-175-5

caused by the highlighting techniques and can be perceived annoying by the users.

The paper is organized as follows. In section 2, we present related approaches regarding AR and highlighting. Section 3 describes the basic scenario including hardware set-up and software components used to implement and test several highlighting techniques. After that, in section 4, the highlighting techniques conceived are presented. A user study, that we conducted, including an evaluation is presented in section 5. Finally, a conclusion (section 6) is given to summarize the contribution of this paper.

2 RELATED WORK

Different research concerning recognition and marking of small areas or entities in the user's physical environment has been conducted in recent years. Guiding techniques are used to navigate the user towards the vicinity of the target. Both Buchmann et al. (Buchmann et al., 2008) and Li et al. (Li et al., 2014) tested and compared different directional AR-interfaces based on projected marks, e.g., an arrow or a compass, in the user's view. While Buchmann et al. and Li et al. conducted their tests in controlled indoor environments, Schwerdtfeger et al. (Schwerdtfeger et al., 2009) developed an application for the logistics system of a warehouse. The user, wearing an HMD, is guided to the targeted product by an overlaid curved tunnel. A square frame at the end of the tunnel highlights the label under the location of the product. One disadvantage of the described approaches so far may be that further elements were added to the user's view that possibly cause a distractive effect on the user's perception of the environment.

Instead of guiding the user to a specific location, the targeted entity can be highlighted itself. For instance, applications are conceivable that support users when searching for a specific item in a shelf. Li et al. (Li et al., 2013) developed an assistant tool for a supermarket scenario. Thereby, the user's visual search processes in front of a shelf is enhanced by highlighting targeted products when seen through the display of a tablet device. Red circles and spotlights (dark overlay in the surrounding of the product) were employed as highlighting techniques. A similar approach was pursued by Löchtefeld et al. (Löchtefeld et al., 2010) who used a mobile camera projector to highlight products in a supermarket shelf that fit the user's preferences. In their work, Li et al. as well as Löchtefeld et al. focus on highlighting as a tool for guidance in a special scenario. For highlighting, the authors chose commonplace techniques (circles and spotlights). The question arises if it is possible to improve the user's guiding process by choosing more advanced highlighting techniques. This is one key issue to be discussed in our work.

Most related work is particularly suitable to highlight virtual entities. Kim and Varshney (Kim and Varshney, 2008) created a new highlighting technique altering virtual models to direct the observer's attention to a targeted point. By altering the geometry of an entity beside the targeted region, attention can be attracted. Though it is an interesting approach for highlighting, the method is only applicable to virtual entities based on the fact that a 3D mesh is necessary to alter their shapes.

The research group Veas et al. (Veas et al., 2011) focused on research in the field of directing attention without distracting the observer. Furthermore, one objective of the research was to create a saliency technique that can be modulated in the rate of user's attention. For every frame of a video, the researchers changed the contrast, lightness, and color of different areas of the frame in a different way to have the highest attention salience inside a designated focus region. Using this technique for highlighting, it is possible to control the degree of the user's attention precisely. One drawback may be that it is necessary to alter most parts of the image presenting the user's view which can lead to a loss of quality in the user's perception of the environment.

The approaches described above either deal with highlighting entities in a specific use case or with the development of a single novel highlighting technique. We did not find any research dealing with the comparison of different highlighting techniques for real entities. Gacem et al. created a design space for guidance techniques which, besides highlighting techniques, also contains tactile and auditory guidance techniques (Gacem et al., 2014). The approaches presented in this section used techniques based on static visualizations. Highlighting techniques that are partially based on movement, such as particle flow, have not been explored in depth yet. Here, the addition of a movement to the highlighting effect makes the entity itself as well as the background temporarily visible that might be a step towards providing a solution to the problem of occlusion.

3 REQUIREMENTS FOR HIGHLIGHTING

This section describes the basic requirements for our highlighting techniques that are presented in the next section. Moreover, we give an example how we met the requirements by giving some realization details on the prototype implementation of the highlighting techniques that was also used in the evaluation. In principle, AR applications use sensor data to align virtual objects in the real environment. This data will be also used for highlighting purposes. Our techniques require a video based sensor. Ideally, the video data is complemented by the data from a depth sensor. The data collected from the real environment needs to be suitable to estimate the contour of a targeted entity. Therefore, sensor data, such as the video stream of a camera, needs to be analyzed. Consequently, the area of a video stream, which belongs to a designated entity, has to be identified. For this, a segmentation algorithm that finds those pixels in a single frame of the video stream is required that contains the contour of the designated entity. The segmentation algorithm should provide the exact contour of an entity as a homogenous area. Moreover, the runtime of the algorithm should meet real-time constraints. A depth camera can be used as it already is able to identify 3D-entities as comparatively homogenous areas. Once the contour is determined, graphics libraries such as OpenGL can be utilized to implement the proposed highlighting effects. Concerning the hardware set-up, one can distinguish between video-see-through or direct-see-through AR systems (Tönnis, 2010). Not all highlighting techniques proposed are suitable for direct-see-through as image of the real object is distorted. This is not a problem when using video-see-through AR systems.

In our user study, we employ video-see-through using a notebook screen (with a screen size of 15.6”) to display the highlighted entities. A specific number of arbitrary entities is placed in front of a Microsoft Kinect-V1-sensor which provides a stream of depth images of the scene to a notebook. In order to get an RGB-video-stream with a resolution higher than the images delivered by the camera built in the Kinect, a HD-webcam was assembled on the Kinect. When conducting user studys, the coordinates of the targeted entities are determined by the administrators in an initial cycle of the experimental set-up. After that, the determined coordinates are delivered to a pipeline for highlighting a designated entity. Thereby, the depth image stream of the Kinect-sensor is analyzed, the contour of the entity is determined, and the entity is highlighted in the user’s view. Segmentation methods mostly are based on two principles (Adams and Bischof, 1994). The first principle contains methods that use edge detection, whereas the second one summarizes areas with similar features. We have implemented a segmentation algorithm that combines both principles. Moreover, for real-time usage we accelerated the segmentation process by limiting the segmentation to one entity instead of the segmentation of the whole image. Figure 1 shows an example of the depth image stream of the Kinect-sensor (1a) and the result of the segmentation process (1b).

(a) (b)

Figure 1: The contour (b) is determined using the depth image stream (a).

Based on the work of Mishra et al. (Mishra et al., 2009), we use the focal point of an entity to segment the image and estimate its contour around the focal point. The transformation into polar coordinates enables the fast estimation of the contour based on the fact that line-by-line iteration can be used to analyze the depth image.

4 HIGHLIGHTING TECHNIQUES

A total of 9 novel highlighting techniques were designed and implemented, with 4 of them using particle systems to highlight a certain entity. Particle systems are widely used, e.g., for visualizations of surfaces (Su and Hart, 2005) and phenomena like clouds, smoke, and fire (Reeves, 1983). Additionally, for comparison purposes, two more conventional techniques already described in related literature have been considered in the following. Although we are focusing on video-see-through, some of the techniques are suitable for direct-see-through as well. Table 1 shows an overview of all highlighting techniques where each of them is classified concerning the use of particle systems, the suitability for direct-see-through, and if one technique is considered conventional.

In the following, each of the highlighting techniques, that are shown in table 1, is presented in a separate subsection.

Table 1: Overview of all designed highlighting techniques assigned to a category. *Part-Sys* is an abbreviation for *using particle systems*, *Direct-ST* is an abbreviation for *suitable for direct-see-through*, and *Conv.* is an abbreviation for *conventional technique*.

Name	Part-Sys	Direct-ST	Conv.
Fire	•	•	
Emitter	•	•	
Garlands	•	•	
Hotplate	•	•	
Red Frame		•	•
Blinking Aura		•	
Black Aura		•	
Red Overlay			•
Color Contrast			
Distortion			
Magnifier			

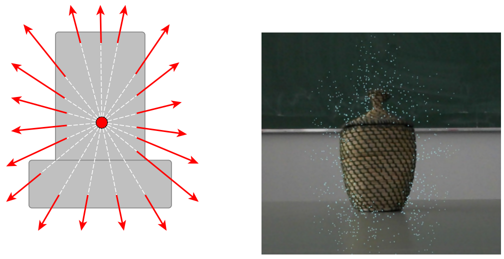
4.1 Fire

The idea of the highlighting technique described in this section – we call it *Fire* – is to place upwelling particles along the contour of the entity to be highlighted. These particles should be designed in a way that the highlighted entity looks like it burns at its contour. Therefore, the particles should have a fire-typical color (with a color range from red to light orange) and change their color as well as their size over time as observable in real fire (see Figure 5a).

At each edge point of the contour of the highlighted entity and at specified intervals between the edges, a particle system is placed which contains a certain number of particles at one point of time. For our user study, we experimented with different numbers of particles and choose the value of 50. A particle that is just emitted from a point on the contour is given a darker color at the beginning than at the end of its lifetime. As it moves upwards, its color is getting brighter. Furthermore, its size is getting smaller. After a certain time, each particle disappears and a new one is emitted in its place.

4.2 Emitter

An entity highlighted with the technique described in this section should look like it sends out particles in every direction creating the effect of an emitter. These particles have a certain lifetime, a certain color and are getting smaller over time to announce their end of lifetime. In our user study, we chose a light blue color which differentiates the particles of the background in an indoor environment. A particle system is set at each point of the contour of the highlighted entity. The starting point of the particles is moved a few pixels inside the shape of an entity so that the emitted particles overlap the entity at the beginning of their lifetime. Figure 2a shows an illustration of an entity (grey shape) highlighted by the *Emitter* technique.



(a) (b)

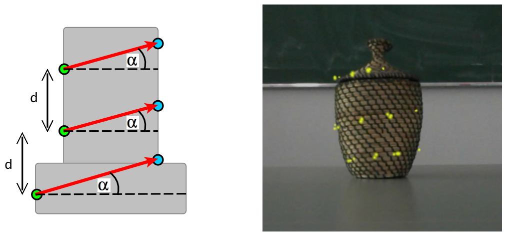
Figure 2: Illustration (a) and screenshot (b) of the *Emitter* technique.

The movement direction of each particle is defined by a vector between the focal point of the highlighted entity (red dot in the middle of the entity in Figure 2a) and one point of its contour. In Figure 2a the movement directions are symbolized as red arrows. The particles are emitted from the point where a red arrow starts. The maximum number of particles each system contains should be set in way that the highlighted entity is still visible and the highlighting effect is not too subtle. In our prototype, we set this value to 20.

4.3 Garlands

The highlighting techniques *Fire* and *Emitter* use particle systems that are set along the contour of the highlighted entity. Another idea is to place particles over the shape of an entity and in this way create a kind of garlands that wrap around the entity. A total of three particle systems is set along the left part of the contour (see the green dots in Figure 3a), each emitting a certain number of particles at one point of time with no diffusion. The maximum number of particles a system contains should be set to a value which on the one hand causes an eye-catching effect but on the other hand covers only a small part of the highlighted entity. For the prototype, we chose 20.

The sources of the particle systems have a continuous distance (*d* in Figure 3a) between them. The value for the distance depends on the height of the entity. The movement direction of the particles (red arrows) have a certain angle α to the horizontal line (dashed lines) which we set to a value of $\alpha = 10°$ in order to achieve a soft rise of the particles. The lifetime of the particles of one particle system ends when the particles reach the right part of the contour (blue



(a) (b)

Figure 3: Illustration (a) and screenshot (b) of the *Garlands* technique.

dots). The size of the the particles changes over time. After being emitted, one particle's size is increasing until it reaches the middle of its way to the right part of the contour (middle of its red arrow), then its size is decreasing again to its end of lifetime. In this way, the particles look like they follow an orbit around the highlighted entity.

4.4 Hotplate

With the intention to create a more subtle way to highlight a certain entity, we designed a highlighting technique that is applied only to parts of the entity. The idea of this technique is to place something like a hotplate at the bottom of the entity. This hotplate should emit particles that move upwards (y-direction in our case) and let the entity look like it is flowed around with blue flames comparably to standing in burning gas (see Figure 5h).

Other than *Fire* and *Emitter*, *Hotplate* is using particle systems emitting particles only along the bottom line of the highlighted entity. The bottom line is defined by a line through two points $p_1 = (x_{min}, y_{min} - offset_y)^T$ and $p_2 = (x_{max}, y_{min} - offset_y)^T$, where x_{min} is the minimum x coordinate of the contour, x_{max} the maximum x coordinate of the contour, and y_{min} the minimum y coordinate of the contour. The value $offset_y$ shifts the bottom line downwards to reduce the overlap of emitted particles with the bottom part of the shape of the entity. In our work, we focus on real entities that can usually be assumed to have a certain base area to stand on the ground. The width of the targeted entity defines the bottom line. At each point of the bottom line a particle system is set containing a certain number of particles at one point of time. We set this number to a value of 50 which led to satisfying results concerning the recognizability. We intended to create the effect that particles at the edges of the bottom reach a higher position than in the middle. Therefore, the speed of the particles along the bottom line is calculated with a parabola (see Equation 1). The parameter $speed_{base}$ was assigned empirically.

$$speed(x) = x^2 \cdot speed_{base} \tag{1}$$

The angular point of the parabola is set at the middle of the bottom line. The particles emitted by the particle system that is set at the point $(x, y_{min} - offset_y)^T$ on the bottom line move with a speed of the value $speed(x)$. A higher value for $speed_{base}$ results in a greater y-dimension for the *Hotplate* effect because the lifetime for the particles is set to a constant value. The emitted particles follow the shape of the given parabola. Furthermore, the size of each emitted particle is decreasing over lifetime.

4.5 Red Frame

Compared to the created highlighting techniques so far, a more conventional approach to highlight an entity is simply enframing it with a colored border. The technique that we call *Red Frame* highlights an entity by surrounding it with a red frame (see Figure 5d). This frame encloses the outmost parts of the entity's contour.

4.6 Blinking Aura

One approach to highlight entities is to create the visual impression of an aura surrounding a targeted entity (see Figure 5b). For this purpose, a glowing circle around an entity, that does not overlay the entity itself, was designed. Moreover, the color of the aura as well as the intensity changes over time to increase the highlighting effect.

The color video stream displayed in the user's view is altered for a pixel at position $p = (x, y)^T$ based on the radius $r = \|p - p_{focus}\|$, that is the Euclidean distance to the focal point p_{focus} of the targeted entity. The aura overlays the image at its pixels in the area between r_{min} and r_{max}. Thereby, the borders of the aura overlay the underlying color image with a low intensity value whereas the intensity of the overlay for a pixel is increasing when the radius is close to $r_{overlay}$, a value between r_{min} and r_{max}. So the parameter $r_{overlay}$ marks the line of points where the overlay covers the image completely. For one pixel, the coefficient c (see Equation 2) represents the ratio between the color value of the video stream image and the color of the overlay dependent on the current radius r.

$$c(r) = \begin{cases} \left(1 - \frac{r_{max} - r}{r_{max} - r_{overlay}}\right)^2 & \text{if } r_{max} > r \geq r_{overlay} \\ \frac{r_{overlay} - r}{r_{overlay} - r_{min}} & \text{if } r_{overlay} > r \geq r_{min} \\ 1 & \text{else} \end{cases} \tag{2}$$

The quadratic term in Equation 2 creates the effect of a smoother glow around the highlighted object. Based on the value c, for every pixel in an image the resulting RGB-values are computed including both the aura and the color image as well. Equation 3 shows the calculation of the color values for a pixel at position p. The resulting RGB-value is a linear combination of the RGB-value in the color image of the video stream and the color of the overlay.

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix}_{img,p} = c \cdot \begin{pmatrix} r \\ g \\ b \end{pmatrix}_{img,p} + (1-c) \cdot \begin{pmatrix} r \\ g \\ b \end{pmatrix}_{overlay} \quad (3)$$

The dimension of the aura is adapted to fit the size of the target entity. So the glow is scaled by adapting the parameter r_{min}, $r_{overlay}$, and r_{max} based on the dimension of the entity. The described highlighting technique creates an aura that surrounds the targeted entity in a comparably static way. Due to the fact that alteration over time is a way to draw someone's attention, we expanded the calculation to create a blinking effect with a periodic change between two colors as well as the intensity of the aura.

4.7 Black Aura

We created a highlighting technique using a black aura to help the observer to distinguish between the targeted entity and the background. Thereby, the entity itself is displayed unmodified. In contrast, the surrounding pixels are overlaid by an aura that decreases its intensity when the distance to the focal point of the targeted entity increases (see Figure 5f). While implementing a similar concept to the one described in section 4.6, a major difference is that the contour of the entity is highlighted with this technique. Due to the fact that the sensor data is not completely exact caused by noise, a distracting movement of the estimated contour of the entity is visible. Thus, we decided to use a Gaussian Blur Filter to stabilize the contour data.

4.8 Red Overlay

This highlighting technique overlays an entity with a red semi-transparent color. The overlay imitates the shape of the entity so that it does not transcend its contour (see Figure 5c). We classified this technique into the conventional ones. It is based on the technique presented in (Möller et al., 2012).

4.9 Color Contrast

According to the Gestalt principle of figure-ground articulation, one approach for a highlighting technique is to emphasize the contrast between an entity and the background. For this purpose, we implemented a technique which alters pixels in the image that do not belong to the shape of the targeted entity. More precisely, the background of a targeted entity is transformed to a gray-scale image while the targeted entity is put in the foreground by displaying its shape in color. Figure 5e illustrates the highlighting technique.

Although this technique showed satisfying results regarding the direction of the observer's attention to the targeted entity in early tests, the disadvantage in highlighting entities, whose shape contains only marginal color information, is obvious. In case the R-, G-, and B-values of the pixels inside the shape of the targeted entity are almost identical for each pixel, the described highlighting technique results not in an improvement regarding the direction of the observer's attention.

To create a highlighting technique working with any kinds of colored entities, we ensured a minimum color contrast between the background and the targeted entity by adding color information to almost gray entities. For this purpose, the RGB-color information is transformed into HSV-values. When the saturation s is going below a minimum threshold s_{thres} for a specific pixel, the color information is altered. Moreover, one goal is to avoid big leaps in the gradient of the color of the targeted entity. So the alteration of the color information of a pixel is not implemented as abrupt leap at the threshold. Instead, the alteration is applied relative to the distance to the threshold. Equation 4 shows the calculation of the coefficient c_s. Based on c_s, the saturation of a pixel is set to a linear combination of the actual saturation value and a maximum value s_{max} (see Equation 5).

$$c_s = \frac{s_{thres} - s}{s_{thres}} \quad (4)$$

$$s = s_{max} \cdot c_s + s \cdot (1 - c_s) \quad (5)$$

The increase of the saturation of a pixel is not sufficient to create a smooth highlighting because of the fact that small differences in the Hue-value h result in distracting artifacts of the targeted entity. So an adaption of the Hue-value is necessary as well. For this purpose, the average Hue-value h_{avg} of the entity's color is used as shown in Equation 6.

$$h = h_{avg} \cdot c_s + h \cdot (1 - c_s) \quad (6)$$

4.10 Distortion

Similar to the highlighting technique described in section 4.6, the technique described here displays a circle around a targeted entity. Instead of an overlay, this technique distorts the image of the entity at pixels near to the surrounding circle. More precisely, in the area of the highlighting effect, the surroundings of the entity are reflected (see Figure 5g). To implement the *Distortion* technique, the coefficient c is calculated as described in Equation 2 based on the parameters r_{min}, $r_{overlay}$, and r_{max}. Dependent on the value $1-c$, the distance between the actual pixel and the point, whose color information is reflected from, is calculated. Thus, a higher distortion for a specific pixel is equal to the reflection of a point in a higher distance.

Equation 7 shows the calculation of the reflected point for a specific pixel of the image. Thereby, s_{max} represents the maximum distance between the position of an actual pixel and the reflected point. The distance is weighted by the value $1-c$.

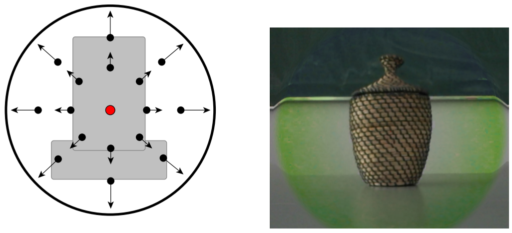
$$\begin{pmatrix} x \\ y \end{pmatrix}_{target} = \begin{pmatrix} x \\ y \end{pmatrix}_{focus} + \begin{pmatrix} x - x_{focus} \\ y - y_{focus} \end{pmatrix} \cdot (1+s) \quad (7)$$

with $s = s_{max} \cdot (1-c)$

Because the calculation for the coordinates of a reflected point in the image results in floating point numbers, the actual displayed RGB-values are taken by the four surrounding neighbors of the reflected point using bi-linear interpolation to generate a smooth highlighting effect without artifacts.

4.11 Magnifier

One approach to highlight a targeted entity and direct the observer's attention to it, is to implement a magnifier that is laid over the entity. An enlargement of the targeted entity as well as of the whole video stream would not result in any highlighting effect, when displaying just a region of the original video stream. Instead, only the area of the targeted entity is enlarged creating the effect of a magnifying lens aligned to the focal point of the targeted entity (see Figure 4b). As a result, the pixel at the focal point of the entity is not altered due to the fact that the – virtual – lens is aligned to it. A point of the targeted entity is projected in a farer distance to the focal point based on the current distance to the focal point. The higher the current distance to the focal point amounts, the higher results the difference between the current point and the projected point. Figure 4a illustrates this issue.



(a) (b)

Figure 4: Illustration (a) and screenshot (b) of the *Magnifier* technique.

The *Magnifier* effect is created by projecting points based on the distance to the focal point similar to the approach of section 4.10. Equation 7 shows the calculation for the projection of a point. In this case, the parameter s, that determines the degree of the magnification, is calculated by Equation 8. The value of $s = -0.3$ creates the effect of a magnified entity without occupying to much of the user's view. The transition between projection and unchanged display of a pixel (dependent on the radius r to the focal point) takes place at the distance r_{max} from the focal point. Due to this transition, there is a distortion of the image at the circle with radius r_{max} around the focal point.

$$s(r) = \begin{cases} -0.3 & \text{if } r < r_{max} \\ 0 & \text{else} \end{cases} \quad (8)$$

5 EVALUATION

In order to evaluate each of the highlighting techniques described in section 4 concerning recognizability, disturbances, and overall impression, a user study was conducted with 23 participants. The youngest participant was 16 years old, whereas the oldest participant was 72 years old. The average age over all participants was about 38.3 years. A total of 9 participants were female. Moreover, we were interested in the question if people, who are used to design and handle different applications including diverse kinds of interaction methods, assess the designed techniques differently than people with basic PC knowledge. So the user's knowledge concerning IT or PC handling was recorded as well. We classified the user's knowledge into two levels, that are "IT Knowledge" (in short "IT"), which means the user has profound skills in IT handling and programming, and "PC Knowledge" ("PC"), which means the user has basic skills to handle a computer.

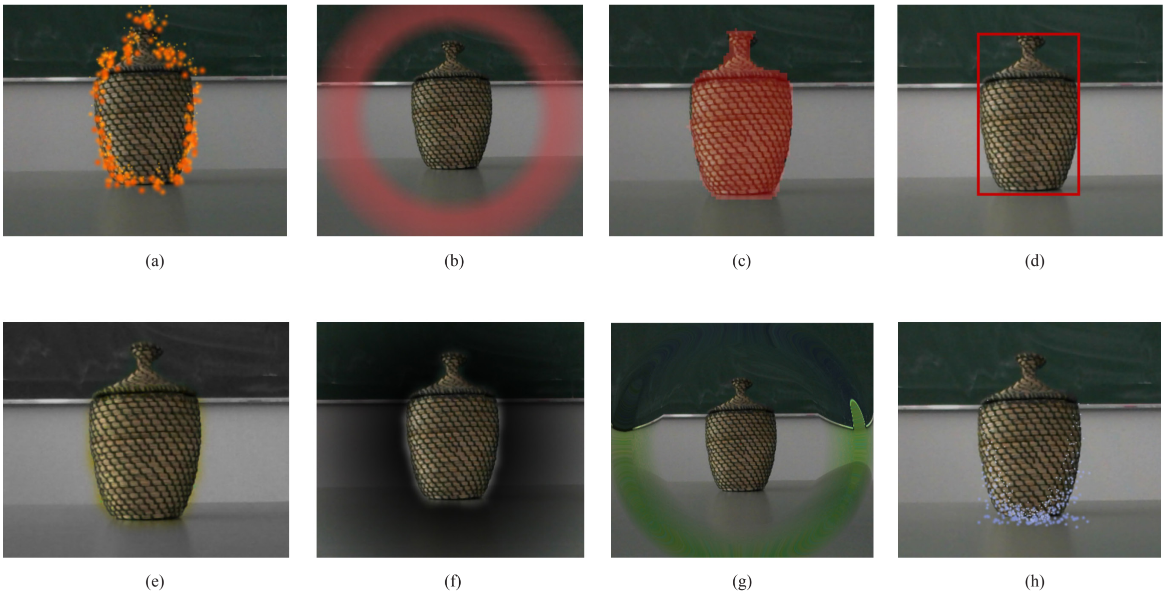


Figure 5: Overview of highlighting techniques: (a) *Fire*, (b) *Blinking Aura*, (c) *Red Overlay*, (d) *Red Frame*, (e) *Color Contrast*, (f) *Black Aura*, (g) *Distortion*, and (h) *Hotplate*.

Concerning the IT or basic PC knowledge, a total of 9 participants were computer science students who rated themselves "IT", whereas 14 of the participants rated themselves "PC". The average age in the "IT" group was about 24.3 years and the average age in the "PC" group was about 47.3 years. Except for one participant, a classification of all participants into "young" (that we defined with an age $\leq$ 30 years) and "old" (age $>$ 30 years) people is identical with the classification into "IT" and "PC".

For the user study, five real entities were arranged on a table. The test subject observed the scene using video-see-through. The application randomly picked out one entity of the scene and highlighted it randomly choosing one technique from the set of highlighting techniques described in section 4 that were not used in the test so far. Thus, the application chose each of the 11 highlighting techniques exactly once. The test users had to click the mouse on the entity that they think is highlighted. The time from showing a highlighted entity to the user's click was measured. After the click, the object remained highlighted on the screen and the test user was asked a total of five questions concerning the current highlighting technique. For each question, we asked the test user to rank the correspondent criterion on a scale from 1 to 5. The questions were:

1. How well or respectively how fast did you recognize the highlighted entity? (scale from 1 [very bad / slow] to 5 [very well / fast])
2. Imagine you wear some glasses for Augmented Reality in which certain entities are permanently highlighted in the way you see here. How strong would you feel disturbed? (scale from 1 [absolutely not disturbed] to 5 [very strongly disturbed])
3. Are there any disturbances in the (visual) perception of the background? (scale from 1 [no disturbances] to 5 [very strong disturbances])
4. Are there any disturbances in the (visual) perception of the entity itself? (scale from 1 [no disturbances] to 5 [very strong disturbances])
5. How well is the contour of the highlighted entity visible? (scale from 1 [not visible] to 5 [clearly visible])

For question 2, we ensured that all participants knew about AR and according devices by giving them a clear explanation about the principle concept. If one test user ranked the questions 3 and 4 other than 1, we additionally asked the user to explain what kind of disturbances he or she perceived. After a test user ranked all of the 5 questions concerning the current highlighting technique the application had picked, he or she could hit the space key and continue with the next randomly chosen entity highlighted with another technique. To avoid that the user recognized a highlighted entity due to a change of the previous highlighted entity to the entity to be highlighted next, a black screen was replacing the video stream for about two seconds before the application chose and showed the next entity. At the end of the test, we asked the user to rank each of the highlighting techniques according to his or her overall impression on a scale from 1 ("I do not like it at all") to 5 ("I like it really much"). We assume that the rankings for the questions are normal distributed based on the fact that 100% of the values are in an interval of $3 \cdot \sigma$ (standard deviation) around the average for each question and each highlighting technique. For all 6 questions (5 during the test and one after it) a one-way ANOVA analysis between two groups of participants ("IT" and "PC") was conducted.

Figure 6 shows the average rankings of the first question. The test users stated they could recognize those entites best or respectively fastest that were highlighted with *Fire* (with an average value of 4.96), whereas entities highlighted with *Red Overlay* and *Blinking Aura* were ranked worst recognizable (3.43 and 3.52). There was a statistically significant difference between the "IT"- and "PC"-group of participants for the ranking of *Emitter* as determined by one-way ANOVA ($F(1,21) = 6.18$, $p = 0.021$), whereat the "PC"-group ranked higher (average of 4.79) than the "IT"-group (4.22). The p-values of the techniques *Garlands*, *Hotplate* and *Blinking Aura* are low as well (between 0.059 and 0.078), but not statistically significant low, due to the fact that the average ranking of the "PC"-group is about 0.6 higher than the ranking of the "IT"-group. The results show that the "PC"-group rates particle effects higher in question 1 than the second group.

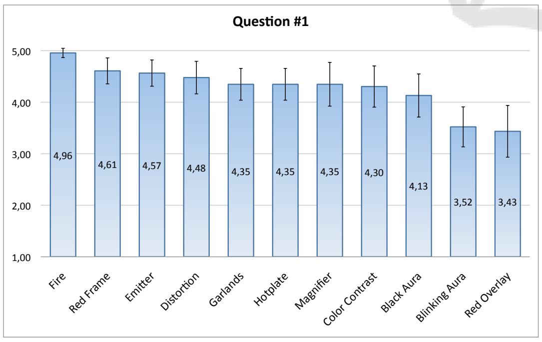


Figure 6: Average rankings of question #1 (including confidence intervals with confidence level $\gamma = 0.95$): Entities highlighted with *Fire* were recognized best.

The average rankings of the second question are illustrated in Figure 7. The chart shows that *Distortion* would be the most disturbing highlighting technique when permanently shown in some Augmented Reality glasses (with an average value of 3.91). The test users rated *Red Overlay* the least disturbing highlighting technique (2.35). The ranking of the two groups show no significant differences (p-values >= 0.364) except for the *Color Contrast* technique. Most members of the "IT"-group did not notice a gray scale background as a disturbance. Consequently, there is a low p- value ($F(1,21) = 10.99$, $p = 0.003$) based on an average value of 2.5 for the "PC"-group and an average value of 4.2 for the "IT"-group.

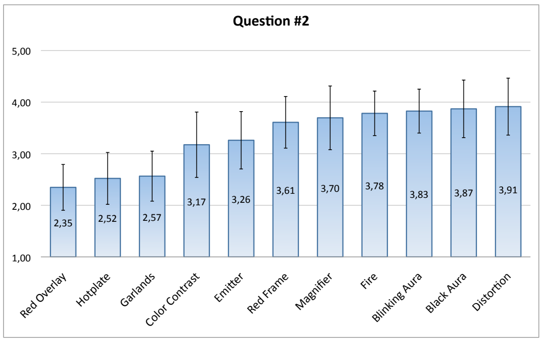


Figure 7: Average rankings of question #2 (including confidence intervals with confidence level $\gamma = 0.95$): The most disturbing highlighting technique was *Distortion*, the least disturbing one was *Red Overlay*.

Figure 8 illustrates the average rankings of the third question. The highlighting techniques with the strongest disturbances in the visual perception of the background were *Distortion* (3.70) and *Magnifier* (3.52). Most users, who ranked *Distortion* and *Magnifier* a high value in the third question, mentioned the distortion that both highlighting techniques provoke. They stated that the surroundings of the entity, especially the entities beside the targeted entity, were not clearly visible. Another disturbance was the change in color at the borders of the distortion ring or respectively the magnifier. Most of the participants ranked *Black Aura* high concerning distractions of the background because of the darkening aura that covers the surrounding area of an entity and so partially blocking the observer's view. Some of the participants stated that the aura near to the targeted entity was blurred. The *Color Contrast* technique is ranked high in this question based on the fact that a transformation of the background to a gray scale image is clearly a disturbance of the original image. As mentioned before, it is statistically significant that most members of the "PC"-group did not mention the gray image as a disturbance which results in different values for the average ("PC": 1.71, "IT": 3.44, $F(1,21) = 11.84$, $p = 0.002$). In addition to *Color Contrast*, *Red Frame* also shows statistically significant higher values for the "IT"-group ("PC": 1.0, "IT": 1.6, $F(1,21) = 8.52$, $p = 0.008$). This may be caused by the fact that "IT"-members stated a more critical evaluation mentioning the overlay of *Red Frame* as a distraction.

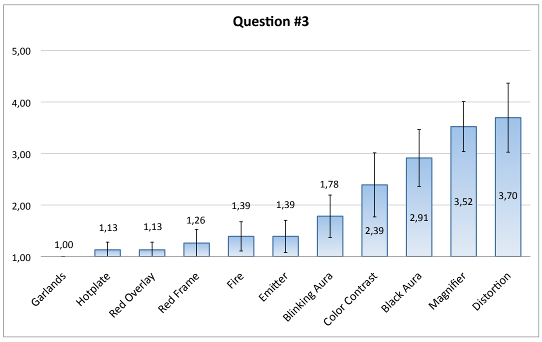


Figure 8: Average rankings of question #3 (including confidence intervals with confidence level $\gamma = 0.95$): A highlighting technique without any disturbances was *Garlands*, whereas *Distortion* was the one with the most disturbances.

As depicted in Figure 9, which shows the average rankings of the fourth question (concerning distractions of the targeted entity itself), *Black Aura* was identified as the technique that created the most distracting effects in a targeted entity itself based on the fact that irritating movement of the contour is visible mostly caused by unstable sensor data. The change of the shape of an entity and the partial darkening of the entity caused by the *Black Aura* effect was mentioned by most of the participants. On the contrary, *Blinking Aura* and *Distortion* caused no visible distortion for the entity itself. We observed high ratings in question 4 for the *Fire* effect. One explanation could be that the *Fire* effect makes it comparably difficult to clearly observe the contour of the targeted entity. The techniques *Red Overlay* and *Red Frame* show high values in this question as well, which could be attributed to the fact that *Red Overlay* alters the color of the entity, whereas *Red Frame* covers parts of the contour and shape. One result of question 4 and question 5 is that an observer's ability to recognize the highlighted entity is mainly depending on the visibility of the contour. Indeed, question 4 and question 5 show almost the same result except for the inversed order caused by the way of questioning. With reference to the "IT" and "PC" group, there is one statistically significant value for the *Magnifier* technique in question 4 as some of the "IT" members mentioned a reduced accuracy of the contour of an entity ($F(1,21) = 5.52$, $p = 0.029$). The contour is less clear when displaying the entity in a higher resolution.

The average rankings of the overall impressions of each highlighting technique are shown in Figure 10. The test users liked *Hotplate* most (3.70), followed by *Emitter* (3.48). The highlighting techniques *Red Frame* and *Distortion* obtained the lowest assessments (2.09 and 2.39). There are no statistically significant differences between the "IT" and the "PC" group.

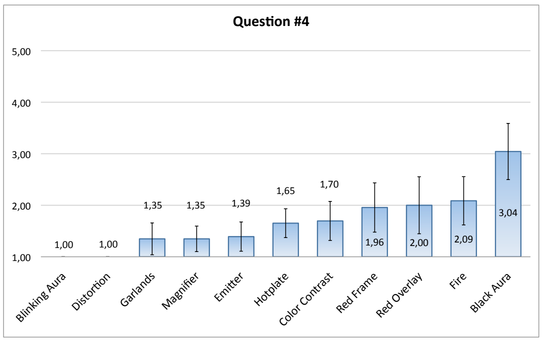


Figure 9: Average rankings of question #4 (including confidence intervals with confidence level $\gamma = 0.95$): The highlighting technique *Blinking Aura* did not provoke any disturbances, whereas entities highlighted with *Black Aura* led to the most disturbances.

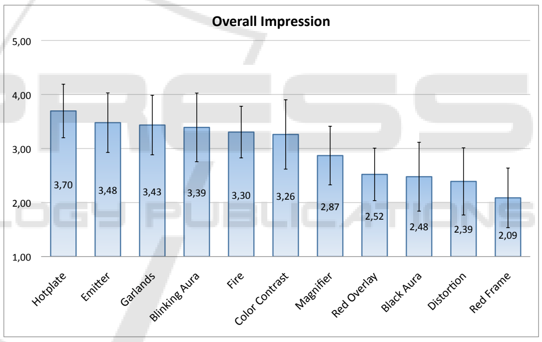


Figure 10: Average rankings according to the overall impression (including confidence intervals with confidence level $\gamma = 0.95$).

The chart in Figure 11 presents the measurements of the time needed between the display of a highlighting technique and the user's click response. Mostly, the median values according to the highlighting techniques are in a small range from 2.2 seconds up to 3.8 seconds. Particle systems are identified as a technique might achieve a higher degree of attention based on median values under 3 seconds. This observation conforms with the results of question 1 which evaluates a short recognition degree for particle systems. *Blinking Aura* (3.8 sec.) and *Color Contrast* (3.7 sec.) emerged as techniques that need a comparably higher amount of time to be recognized by a user. Furthermore, a test with a color-blind person, who stated this highlighting technique as almost invisible, indicates that *Color Contrast* is not barrier-free. In contrast, the other highlighting techniques could easily be recognized by this person.

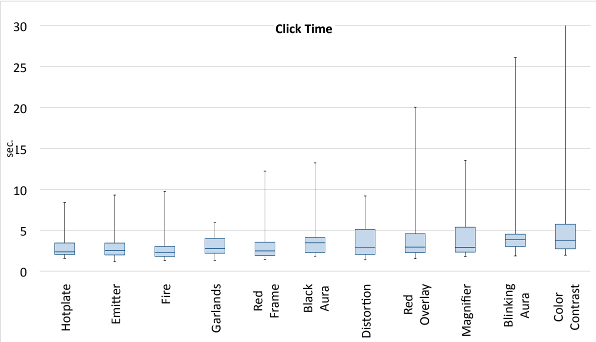


Figure 11: Measurements of recognition time. The recognition time for *Color Contrast* has a very high deviation due to the fact that one color-blind participant recognized the highlighted entity after more than 70 seconds.

To compare the conventional techniques *Red Frame* and *Red Overlay* with the designed techniques using novel approaches, we conducted paired t-tests (two-tailed, confidence level $\gamma = 0.95$) for the 6 questions where conventional and novel techniques are the related groups of the tests. We assume a normal distribution based on the fact that 100% of the values regarding the two groups are in an interval of $3 \cdot \sigma$ (standard deviation) around the average value for each question. As there are p-values ≤ 0.039 for all 6 tests, there are statistically significant differences between conventional and novel techniques.

For all highlighting techniques, the average frame rate in frames per second (fps) was measured, based on a 5 minute measurement for each highlighting technique. Figure 12 shows the results of the measurements. The highlighting techniques that use particle systems have the best frame rates (with values of 12.6 fps and 12.4 fps), followed by the aura effects (10.8 fps and 10.4 fps). *Color Contrast* shows the lowest frame rate (5.0 fps) due to the fact that a time-consuming color transformation to HSV color space was used. This shows that the proposed highlighting techniques might have a considerable impact on the frame rate.

6 CONCLUSION

We designed and implemented a total of 9 novel highlighting techniques that can not only be applied to virtual entities but to real entities as well. Thereby, we focused on techniques that are using moving particles (*Fire*, *Emitter*, *Garlands*, and *Hotplate*),

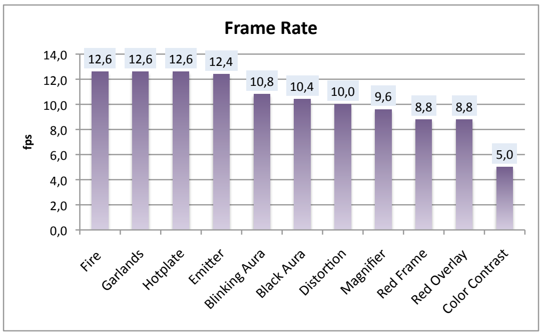


Figure 12: Measurements of the average frame rates in frames per second (fps) of all highlighting techniques.

auras (*Blinking Aura* and *Black Aura*), distortions (*Distortion* and *Magnifier*), and an alteration of the color (*Color Contrast*) to highlight a certain entity. Additionally, for comparison purposes, we implemented two more conventional techniques (*Red Frame* and *Red Overlay*). We conducted a user study with 23 participants evaluating the quality of each highlighting technique concerning recognizability, disturbances, and overall impression. Moreover, we examined the question if there are statistically significant differences in the perception of the highlighting techniques for users with much experience concerning IT handling and those with few.

The results of the user study show that *Hotplate* obtained positive assessments. It emerged as a technique combining useful attributes for highlighting an entity in the user's view. In fact, *Hotplate* was ranked highest in overall impression with an average value of 3.7 and identified as a technique recognized fast (proven by click time measurement). Furthermore, it caused only little disturbances of the background (question 3) as well as the entity itself (question 4/5). Moreover, based on the user study, particle systems in general can be named as a successful tool to effectively draw attention to a specific entity or an area in the user's view without creating annoying distractions. Though more conventional highlighting techniques like *Red Frame* and *Red Overlay* show good recognition times as well, they obtained low assessments concerning the overall impression. The differences between conventional and novel techniques proved to be statistically significant. Thus, the highlighting techniques presented in this paper can be considered a useful addition to the highlighting techniques already described in the related literature. A designer intending to use them needs to tweak several parameters associated with each technique (e.g. colors used or velocity of movement). Also, the impact of these highlighting techniques on performance have to be considered and optimizations that aim to increase performance need to be examined. The latter two issues are starting points for future work.

REFERENCES

Adams, R. and Bischof, L. (1994). Seeded region growing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16: pages 641–647.

Buchmann, V., Billinghurst, M., and Cockburn, A. (2008). Directional interfaces for wearable augmented reality. In *Proceedings of the 9th ACM SIGCHI New Zealand Chapter's International Conference on Human-Computer Interaction: Design Centered HCI*, CHINZ '08, pages 47–54, New York, NY, USA. ACM.

Gacem, H., Bailly, G., Eagan, J., and Lecolinet, E. (2014). A design space of guidance techniques for large and dense physical environments. In *Proceedings of the 26th Conference on L'Interaction Homme-Machine*, IHM '14, pages 9–17, New York, NY, USA. ACM.

Kim, Y. and Varshney, A. (2008). Persuading visual attention through geometry. *Visualization and Computer Graphics*, 14: pages 772–782.

Li, M., Arning, K., Bremen, L., Sack, O., Ziefle, M., and Kobbelt, L. (2013). Profi: Design and evaluation of a product finder in a supermarket scenario. In *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, UbiComp '13 Adjunct, pages 977–984, New York, NY, USA. ACM.

Li, M., Arning, K., Sack, O., Park, J., Kim, M.-H., Ziefle, M., and Kobbelt, L. (2014). Evaluation of a mobile projector-based indoor navigation interface. *Interacting with Computers*, 26: pages 595–514.

Löchtefeld, M., Gehring, S., Schöning, J., and Krüger, A. (2010). Shelftorchlight: Augmenting a shelf using a camera projector unit. In *Adjunct Proceedings of the Eighth International Conference on Pervasive Computing*. Springer Lecture Notes in Computer Science.

Milliron, T., Jensen, R. J., Barzel, R., and Finkelstein, A. (2002). A framework for geometric warps and deformations. *ACM Transactions on Graphics*, 21: pages 20–51.

Mishra, A., Aloimonos, Y., and Fah, C. L. (2009). Active segmentation with fixation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 468–475. IEEE.

Möller, A., Kranz, M., Huitl, R., Diewald, S., and Roalter, L. (2012). A mobile indoor navigation system interface adapted to vision-based localization. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, MUM '12, pages 4:1–4:10, New York, NY, USA. ACM.

Reeves, W. T. (1983). Particle systems – a technique for modeling a class of fuzzy objects. In *Proceedings of the 10th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '83, pages 359–375, New York, NY, USA. ACM.

Schwerdtfeger, B., Reif, R., Gunthner, W., Klinker, G., Hamacher, D., Schega, L., Böckelmann, I., Doil, F., and Tumler, J. (2009). Pick-by-vision: A first stress test. In *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium on*, pages 115–124. IEEE.

Su, W. Y. and Hart, J. C. (2005). A programmable particle system framework for shape modeling. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, New York, NY, USA. ACM.

Tönnis, M. (2010). *Augmented Reality – Einblicke in die Erweiterte Realität*. Springer. ISBN: 978-3-642-14178-2, DOI: 10.1007/978-3-642-14179-9.

Veas, E. E., Mendez, E., Feiner, S. K., and Schmalstieg, D. (2011). Directing attention and influencing memory with visual saliency modulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1471–1480, New York, NY, USA. ACM.

Ware, C. (2004). *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.