

# Augmented Reality Graph Visualizations

## Investigation of Visual Styles in 3D Node-Link Diagrams

**Wolfgang Büschel**  
Interactive Media Lab,  
Technische Universität  
Dresden

**Stefan Vogt**  
Interactive Media Lab,  
Technische Universität  
Dresden

**Raimund Dachsel**  
Interactive Media Lab,  
Technische Universität  
Dresden

3D Node-link diagrams are an important class of visualization for immersive analysis. Yet, there is little knowledge on how to visualize edges to support efficient analysis. We present an exploration of the design space for edge styles and discuss the results of a user study comparing six different edge variants.

Augmented Reality (AR) can provide immersive, stereoscopic experiences in 3D without losing the connection to the environment or other people. Thus, it has a strong potential for collaboration among experts while integrating

real world environments and virtual content. With new, complex AR systems for use cases such as maintenance assistance or cyber physical production systems, the need to visualize relations between both real objects and virtual parts of the AR scene will increase. Examples for such relations are computer networks integrated into the environment, information flows in a smart home, and many more. Typically, they are visualized as 3D node-link diagrams. In addition to these in-situ use cases, there are also domains where graph data, e.g., flight paths, is analyzed in 3D space without necessarily being coupled to physical objects. These use cases can still benefit from AR in terms of presence, natural interaction, and physical navigation. Together, these examples show that 3D graph representations are an important class of visualizations in *Immersive Analytics*.<sup>1</sup> However, there has been little research on how to visualize 3D graphs to facilitate efficient and effective analysis in AR.

As a first step into this complex problem, we are interested in the design of *edge visualizations* in particular. They will most likely be the visually dominant part in an AR graph visualization, especially in complex scenarios or at room size. Which basic visual attributes lend themselves to visualize edges? How can users distinguish between edges and perceive directed edges? How can we map data to edges? In this article, we aim to lay the foundation to answering these questions. We contribute an analysis of design considerations for edge visualizations in 3D AR graphs. Based on these and our own experiences in the field, we selected and designed six visual styles for undirected and directed edges. We report on the results of a quantitative user study

with 18 participants, comparing the performance of these edge styles for typical graph exploration tasks.

## BACKGROUND

Our work builds on existing research in the fields of graph visualization, specifically edge visualization and mixed reality 3D graph visualization.

### Edge Visualization of Graphs

There has been some work on visualization of edges in 2D graphs. For example, Holten et al.<sup>2</sup> presented a comprehensive evaluation of different directed edge representations for 2D node-link diagrams. Recently, Romat et al.<sup>3</sup> investigated animation for edge attribute encoding. A different approach for edge visualizations are partial edge drawings as presented by Bruckdorfer et al.<sup>4</sup>, where only the start and end of an edge are drawn to minimize visual clutter. However, there has been very little research specifically on 3D graph edges, let alone for AR. In previous work, we examined different visualizations of graph edges in mobile, hand-held AR.<sup>5</sup> Our current work differs in that we consider head-mounted displays (HMD) instead of hand-held AR. In addition, we previously only evaluated different edge styles in a small, semi-formal study.

### Mixed Reality 3D Node-Link Diagrams

The visualization of 3D graphs in general and for VR/AR specifically has already been researched since at least the early 1990s. Early examples include the presentation of 3D networks on stereoscopic displays by Crutcher et al. in 1995.<sup>6</sup> Work so far has concentrated on examining the advantages, if any, of immersive visualization in comparison to less-immersive or traditional means. Already in 1996, Ware and Franck examined the effect of different stereo cues on path finding performance in 3D graphs.<sup>7</sup> Similarly, Belcher et al.<sup>8</sup> compared the effect of using AR for graph exploration to a desktop interface, showing the advantage of AR. Bacim et al.<sup>9</sup> studied how display fidelity affects graph analysis performance. They found evidence that more immersive displays positively affect task completion times. Kwon et al.<sup>10</sup> studied HMDs for graph visualization, comparing a 2D representation and two 3D representations. However, they did not consider AR systems and their study did not focus on different edge styles. Cordeil et al.<sup>11</sup> have presented an in-depth analysis of the difference between CAVE (Cave Automatic Virtual Environment) and HMD setups for the collaborative analysis of networks. They found HMDs lead to faster collaboration, making them a suitable alternative to the more expensive CAVE setups. Besides network analysis, other use cases for graphs in VR have been examined, for example networks of concepts to support creative design thinking by Georgiev et al.<sup>12</sup>

## EDGE VISUALIZATION IN MIXED REALITY 3D NODE-LINK DIAGRAMS

Our review of the related work shows a lack of structured research into the particulars of visualizing 3D graphs in general and for Augmented Reality. This is a rich space worthy of extensive exploration. As an important step towards a better understanding of this design space, we present a set of design considerations for edge visualizations in 3D node-link diagrams, specifically for usage in AR settings. The following recommendations are based on the review of the (limited) literature as well as our own experiments.

### Visual variables and their suitability for Augmented Reality

Carefully mapping data attributes to visual variables is an important step in creating an effective visualization. Examples for visual variables include basic properties such as shape, size, and color but also more complex aspects such as transparency, focus, and texture. Among others,

Carpendale<sup>13</sup> gives an overview of the general suitability of many of these visual variables for aspects such as conveying an order or a numerical difference. AR, however, poses several additional challenges that designers of visualization systems have to consider when choosing appropriate visual variables:

1. *Perspective*: 3D perspective projections lead to objects appearing smaller when further away from the viewer. This foreshortening also means that shapes appear distorted, i.e., parallel lines do not appear parallel. In addition, and in contrast to classic 3D visualization, AR typically limits the camera to those points of view a user can reach. For example, a top-down view, helpful to get an overview of the data, is often not feasible.
2. *Occlusion*: Free navigation in an AR environment leads to unpredictable occlusion between arbitrary objects in the scene – including real, physical objects. Thus, objects can be hidden from view or overlap each other. In combination with perspective, this can happen even independently of the size of the objects.
3. *Visual clutter*: AR often does not give any control over the background; in fact, the background will typically be the very thing one aims to augment. Therefore, in many cases an additional potential source of clutter is added to the overall scene. This applies both to the background color and motion (e.g., of passers-by.)
4. *Color reproduction and transparency*: Depending on the type of AR display, color reproduction can be limited. A typical HMD has an additive display, adding light to the environment as seen by the user, instead of blocking it. Most importantly, this means that black becomes transparent. It follows that no truly opaque appearance can be reached under normal circumstances. In addition, the meaning of transparency is not immediately clear to the user.

Given these additional challenges, we can revisit important visual variables and assess their suitability for designing edge visualizations:

*Color*: As mentioned above, color reproduction is problematic in optical see-through AR due to the display technology. In addition, a colorful background can lead to unpredictable results for color legibility. Still, color is a very powerful visual variable and our previous experiments<sup>5</sup> showed that it was suitable at least in video see-through AR.

*Size/stroke width*: Perspective means that size can be problematic. This is true especially for using absolute sizes. We believe that relative sizes would fare better, as differences between close objects would still be visible independent of distance. For edges specifically, thickness can be used to encode edge weights. Note that the comparably low resolution of current AR hardware further limits the use of size as a visual variable.

*Shape*: For edge visualizations, this includes stippling patterns, different glyphs, or the use of curvature. While some aspects of shapes might be influenced by perspective, we still believe shape to be one of the most important factors for AR graph visualization. Shape is not affected by bad color reproduction and may help to address problems of occlusion. However, it might negatively affect visual clutter.

*Transparency*: Transparency can easily be added but conflicts with lightness/blackness due to the display technology. Thus, special care needs to be taken when color and transparency are both used to encode attributes. Additionally, while transparency is useful to hide unimportant data, e.g., because of a filter operation, visibility is again highly dependent on the background.

*Focus*: Depth of field is a strong depth cue and may lead to a more realistic presentation. On the other hand, blur effects can be used to steer the user's attention or hide data that has been filtered out. Blurring far-away parts of the visualization may decrease visual clutter but comes at the expense of reduced legibility.

*Texture*: Texture, including the use of patterns or varying grain, can be used to differentiate categories of edges but usually does not support quantitative interpretations. In an AR application, designers need to choose textures that do not interfere with materials existing in the physical world. In addition, very detailed or fine patterns may suffer from aliasing due to low-resolution hardware or sharp viewing angles.

## Directed edges

In a directed graph, the individual edge directions need to be visualized. Approaches shown in the related work<sup>2</sup> include the use of tapered edges, curvature, and animations. Not all of these approaches are applicable to 3D. For example, curvature is dependent on the view direction and arrows can easily be occluded or too far away to be recognizable. Based on the challenges of 3D AR representations discussed above, we suggest that suitable techniques should fulfill the following requirements:

1. The technique should encode the edge direction over the whole length of the edge or repeatedly in sufficiently small intervals. Following this guideline, partial occlusion of an edge should not prevent users from perceiving the edge direction. In addition, the edge direction is still visible if the end of the edge is far away, e.g., in room-scale visualizations.
2. The encoding has to be independent or at least very robust against changes in the view direction. In a 3D environment, viewing an edge from a sharp angle may otherwise impair perception.
3. Direction is often not the only attribute of an edge. Thus, the encoding of the direction should not interfere with other visual variables used to encode additional attributes. Consequently, e.g., color gradients may not be suitable for directed edges as color is one of the most important visual variables.

## Encoding multivariate or continuous data on edges

In many cases, there are multiple attributes associated to edges in a graph. For example, in a computer network each link could have attributes like latency, bandwidth, and connection type.

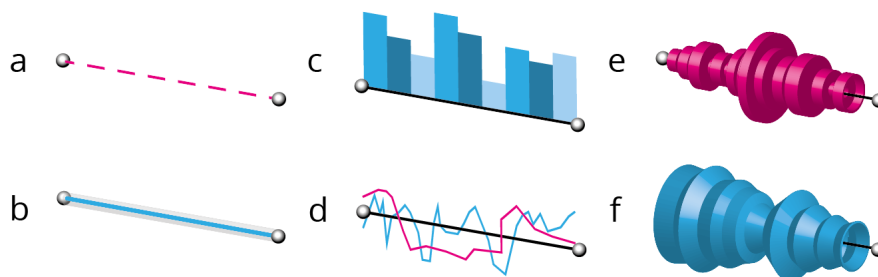


Figure 1: Examples for multivariate or continuous data encoding on edges: a) stippling & color b) color and relative stroke width c) bars d) line plot e) revolved line plot f) revolved plot of discrete data over time.

To encode these attributes in a 3D graph visualization, there are different approaches (Figure 1). The visual variables mentioned earlier can be combined, for example color and a stippling pattern (Figure 1a). Values relative to a maximum may be visualized by utilizing stroke width in combination with a transparent outline of the maximum (Figure 1b). More complex data can be shown by attaching plots such as bar charts (Figure 1c) or line plots (Figure 1d) directly to the edges. These plots can also visualize dynamic, time-dependent data. Several problems emerge. Depending on the viewing angle, these visualizations might be hard to read. 3D revolved plots of discrete (Figure 1e) or continuous (Figure 1f) data may address this partially. Although still subject to occlusion, these plots show general trends or outliers from a wider range of viewing angles. Another problem is that the length of the link affects the visualization. Finally, direction (e.g., “left to right”) would change when the edge is viewed from the back.

## STUDY OF EDGE VARIANTS IN 3D GRAPHS

Evidently, the design space described above is too big to address in one study. As a starting point, we wanted to compare different visual edge variants for undirected and directed 3D

graphs. We were interested to find out how these variants perform in typical graph analysis tasks set in an AR environment. Specific research questions that we aim to address are as follows: Can users adequately find paths between nodes in a 3D graph in AR? Can users distinguish directions of edges and thus find directed paths? Which edge variants are most suitable for these tasks? This forms the basis for further studies, e.g., with a focus on multivariate data. At the end of this article, we point out possible future work.

## Study Design

We designed our study to encompass two independent, controlled lab experiments. In the first experiment, we examined undirected edges, in the second we concentrated on directed edges. Both experiments followed a within-subjects design. Our independent variables were task difficulty (two levels) and edge variant (three variants), leading to 3x2 designs for both experiments. For the first experiment, task difficulty was the number of nodes (18 or 36). For the second experiment, the node count was fixed at 24 and the task difficulty was the path length (1 or 2). The dependent variables in both experiments were task completion time and error rate. We based the decisions for node count and path lengths on expert feedback during the prototype development and a small pre-study.

## Edge variants

We designed six edge variants based on the guidelines presented above, three undirected variants for the first and three directed variants for the second experiment. We wanted to focus on geometry as the dominant factor for each variant. This leaves other visual variables, such as color, to encode additional edge attributes if necessary. In addition, the geometry of each variant could be parametrized to encode further attributes, as described above and as already shown for animated edges by Romat et al.<sup>3</sup>

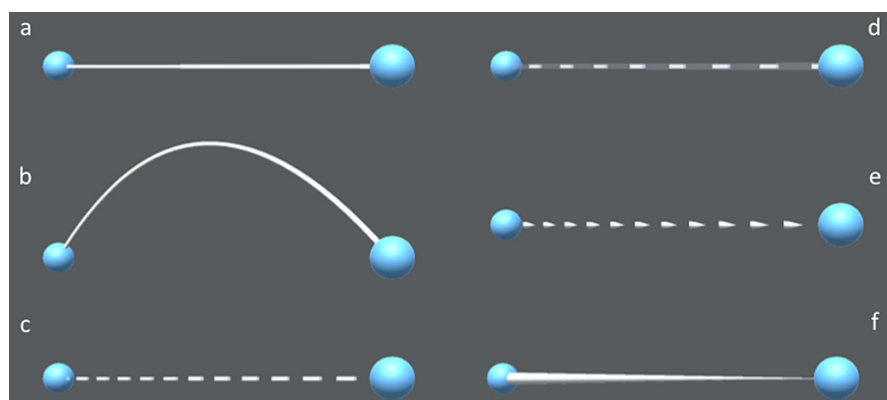


Figure 2. Illustration of the six edge variants selected for the study: a) straight edge b) curved edge c) dashed edge d) animated edge e) glyph-based edge f) tapered edge. We used the three undirected variants (a-c) in experiment one, the three directed variants (d-f) in experiment two.

The first undirected edge variant was a *straight edge*, build from a single segment (Figure 2a). The second technique was a *curved edge*, based on a Bézier curve with the arc's height set to be half of the distance between the two connected nodes (Figure 2b), leading to curvature being distance-dependent. We hypothesized that this might help to break up occlusion and allow users to more easily follow the edge. As the third edge variant, we chose *dashed edges* (Figure 2c). The idea behind this variant was to allow users to view what lies behind the edge, again helping with occlusion issues.

For the directed edges, we first chose *animated edges* built from segments moving in the direction of the edge. A transparent hull improves visibility of the edge's path (Figure 2d). The second variant was a *glyph-based edge* made up of small pyramids (Figure 2e). In comparison to

simple arrows, this edge type does not only encode the direction near the target node but repeatedly along the edge. For the last variant, we chose a form of *tapered edge* consisting of a single cone-shaped segment that encodes the direction over its whole length (Figure 2f). Figure 3 shows examples of the edge variants as seen in our prototype.

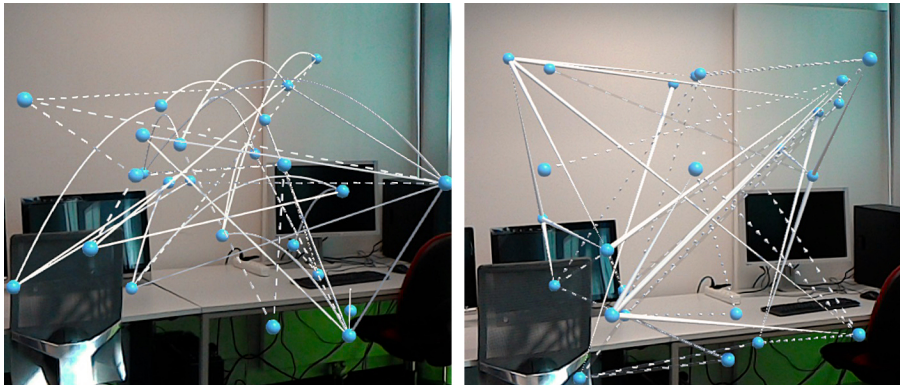


Figure 3. Example graphs for the three undirected edge variants used in the first experiment (left) and the three directed edge variants used in the second experiment (right). Note that during the study only one edge variant was presented at a time.

## Tasks

Lee et al.<sup>14</sup> presented a taxonomy of graph analysis tasks, classifying tasks into topology-based tasks, attribute-based tasks, browsing tasks, and overview tasks. Based on this taxonomy, we chose path-finding tasks between two given nodes, as they cover both following of paths (browsing) and detecting reachability (topology). For the first experiment, we asked the participants to find if there was a path of the length 2 between two highlighted nodes, i.e., a path from one highlighted node to a different node and a further path from that node to the second highlighted node. In the second experiment, participants had to decide if there was a directed path between a start and an end node. Depending on the task difficulty level, this path was either direct or again of length 2. We chose the path lengths for the two tasks after pre-trials showed that direct connections for the first task would have been too easy and that the difference in task difficulty between length 1 and 2 was big enough for the second task.

## Participants

We invited 18 unpaid participants (9 male, 9 female) from diverse educational backgrounds in fields such as computer science, psychology, and design, with ages ranging from 25 to 36. Their average age was 30. All participants reported normal or corrected-to-normal vision; there were no cases of stereo deficiency or color blindness.

## Setup & Procedure

As head-mounted display, we chose Microsoft's HoloLens. We implemented our prototype, including the different edge variants, in the Unity 3D engine. We used a standard Microsoft Xbox game controller for input during the study.

## Study environment

The study took place in a lab environment with controlled light and climate conditions. We explicitly decided against a neutral background as we were interested whether the real environment would have an impact on the participants' performance. The room had a size of 50 m<sup>2</sup>. The starting point for the participants was marked on the ground. We placed the virtual scene stationary

in the middle of the room, at eye level of the participant. An area of approximately  $6\text{ m} \times 4\text{ m}$  was available for the participants to move around the graph freely. The graph was contained in a volume with a width and depth of 1.25 m and a height of 1 m with the center at a distance of 3 m to the starting position (Figure 4). At this time, we were mainly interested in the different edge variants. Thus, we made sure that no physical objects would lead to occlusion. We also decided against even larger, room-size graphs as we felt that the limited field of view (FoV) of the HoloLens would have had too much of an influence on the study.

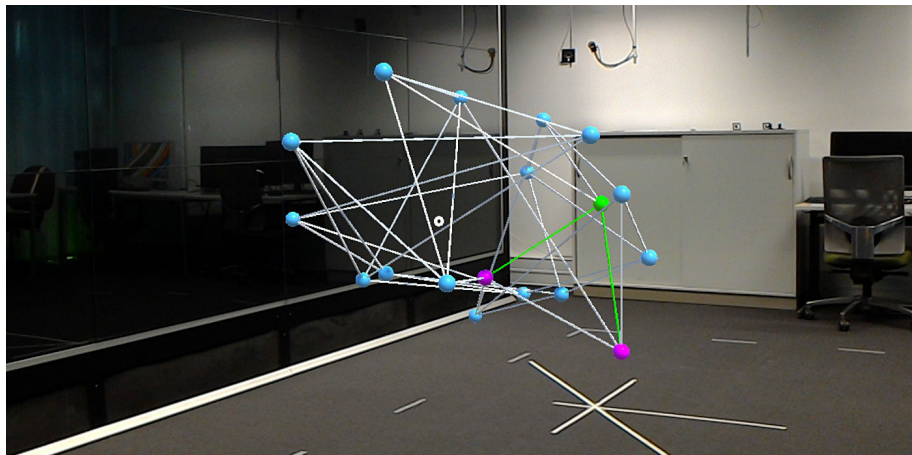


Figure 4. Example graph with straight edges and 18 nodes as used during the first experiment. The green path shows the correct answer. We only showed these during the training trials.

## Procedure

After an introduction and giving informed consent, each participant answered an entry questionnaire including demographic data and prior experience. We also measured their individual pupillary distance with a pupillometer and configured the HoloLens accordingly. Afterwards, we introduced the terminology used in the study, including the necessary concepts of nodes, edges, connectivity, and path length. We then explained the HoloLens and the interaction with the game controller.

Following this, each participant completed the two experiments explained above. For each, we first explained the task in detail. Then, each participant had six training trials, one for each condition (three edge variants, two task difficulties). Overall, training took around 3-4 minutes. After the training, 48 regular trials followed with block-wise counter-balanced conditions. We instructed the participants to complete the trials as correct as possible while being as fast as possible. For each trial, both training and measured, participants had to first return to the starting position. Then, they had to look at a virtual cube presented as a fixation object to control the initial view direction. After confirmation, we only showed the nodes at first. Here, the relevant nodes were already highlighted. After five seconds, we also showed the edges and time started. Participants could move freely in the room during the trials. After deciding if a path between the two highlighted nodes existed that fulfilled the conditions of the task (see above for a description of the tasks), the participants used the game controller to give their yes/no answer. Only during training, we then showed the correct solution. After the trials of an experiment were completed, each participant filled out a questionnaire rating the presented edge variants.

At the end of the study, we asked the participants for general feedback on 3D graph visualization, the influence of the physical background, the prototype, and the study itself. In total, each participant completed the study in 75 to 90 minutes.

## Graph generation

For each trial during the tasks, we generated a random graph. To control for the number of edges between nodes and their effect on the task complexity, all graphs were generated to be 4-regular (i.e., all nodes had exactly four direct neighbors), in case of the second experiment with two incoming and two outgoing edges. If a correct solution for the task existed, it was unique. All nodes were placed randomly in the visualization's bounding volume. We chose the number of nodes (18 and 36 for the first experiment and 24 for the second experiment) after pre-trials. Our goal was to have difficult enough tasks without too much clutter, given the available space.

## Data collection

We measured task completion times and recorded the number of correct and wrong answers. We also tracked and logged the 3D position of the participants in the room, allowing us to generate movement trajectories and aggregate movement data (Figure 7). We decided against filming the study because screen recording with the HoloLens limits the framerate, while external recording would not show the AR scene. All questionnaires were done on paper and later digitized.

## RESULTS

In the following, we present the results of the two experiments of our study in detail. Error rates showed a strong floor effect – simply put, participants did not make many errors. This was in accordance with the focus on correctness over speed in the instructions. Thus, we concentrate on the task completion times and the user ratings. We briefly discuss error rates and other, additional results at the end of the section.

### Experiment 1 – Undirected Edges

In the first experiment, we compared three variants of undirected edges in two task complexity levels (graphs with 18 nodes and 36 nodes). The task completion times and user ratings for the first experiment are as follows:

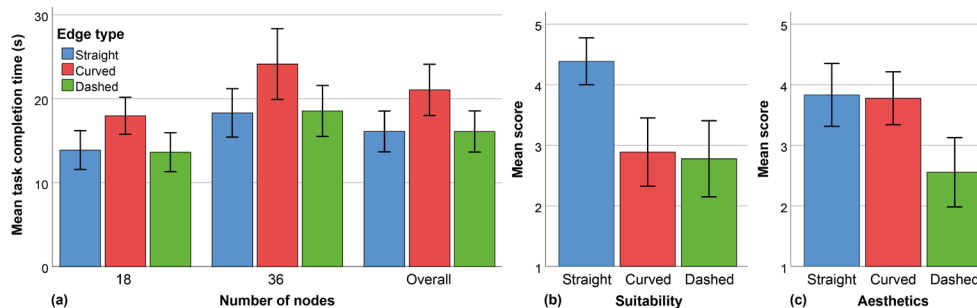


Figure 5. Results of the first experiment (undirected edges). a) Mean task completion times in seconds, b) user ratings for suitability, c) user ratings for aesthetics. Higher ratings are better. Error bars show 95% confidence intervals.

### Task completion times

To analyze the task completion times (Figure 5a), we did a two-factor repeated measures ANOVA with the within-factors of edge type and task difficulty. A Shapiro-Wilk test for normality and Mauchly's test on sphericity did not reveal any violations of the underlying assumptions. The ANOVA showed significant main effects for both edge type ( $F(2,34) = 27.83$ ,  $p < .001$ ,  $\eta_p^2 = .621$ ) and task difficulty ( $F(1,17) = 67.55$ ,  $p < .001$ ,  $\eta_p^2 = .799$ ) but no significant interaction. Bonferroni-corrected post-hoc tests showed significantly shorter task completion times for straight edges ( $M = 16.11$  s,  $SD = 4.90$  s) and dashed edges ( $M = 16.10$  s,  $SD = 4.94$  s)



in comparison to the curved edges ( $M = 21.06$  s,  $SD = 6.12$  s). The more difficult tasks with 36 nodes ( $M = 20.34$  s,  $SD = 5.93$  s) took significantly longer to complete than the tasks with 18 nodes ( $M = 15.17$  s,  $SD = 4.31$  s).

## User ratings

The analysis of the post-experiment questionnaires revealed a strong preference among our participants for the straight edges. We asked participants to rate both the suitability of the edge style for the task, as well as the aesthetics of the style (Figure 5b, c). All ratings were on a scale between 1 and 5, with 5 being the best rating. A non-parametric Friedman test showed significant differences between the edge variants, both for suitability ( $\chi^2(2) = 11.20$ ,  $p = .004$ ,  $N = 18$ ) and aesthetics ( $\chi^2(2) = 10.24$ ,  $p = .006$ ,  $N = 18$ ). Bonferroni-corrected post-hoc tests revealed that straight edges ( $M = 4.39$ ,  $SD = 0.76$ ) were rated significantly more suitable than both the curved ( $M = 2.88$ ,  $SD = 1.10$ ) and dashed ( $M = 2.78$ ,  $SD = 1.23$ ) variants. In terms of aesthetics, our participants rated dashed edges ( $M = 2.56$ ,  $SD = 1.12$ ) significantly worse than the straight edges ( $M = 3.83$ ,  $SD = 1.01$ ). The difference to curved edges ( $M = 3.78$ ,  $SD = 0.85$ ), however, was not significant due to the Bonferroni correction.

## Experiment 2 – Directed Edges

In the second experiment, we compared three variants of directed edges in two task complexity levels (path length 1 and 2). The task completion times and user ratings for this experiment are as follows:

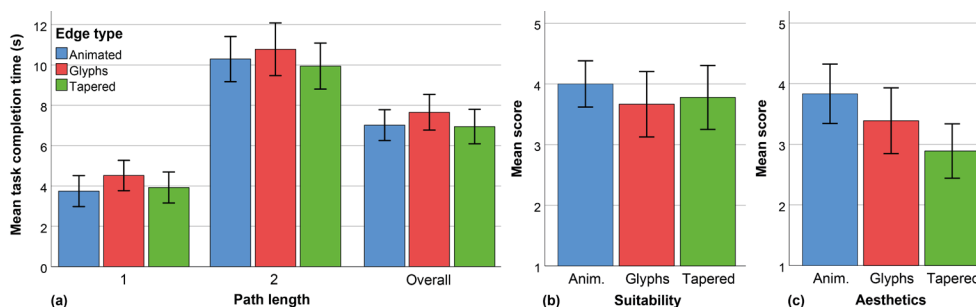


Figure 6. Results of the second experiment (directed edges). a) Mean task completion times in seconds, b) user ratings for suitability, c) user ratings for aesthetics. Higher ratings are better. Error bars show 95% confidence intervals.

## Task completion times

Just as in the first experiment, we analyzed the task completion times (Figure 6a) in a two-way repeated measures ANOVA. The Shapiro-Wilk test showed a violation of the normality assumption. However, after removing outliers (data points outside a range of three standard deviations, ca. 6% in this experiment), we no longer detected significant deviation from normality. Mauchly's test showed no violation of the sphericity assumption. Interestingly, the ANOVA revealed a significant main effect for both the edge variant ( $F(2,34) = 4.64$ ,  $p = .017$ ,  $\eta_p^2 = .214$ ) and the task complexity ( $F(1,17) = 258.94$ ,  $p < .001$ ,  $\eta_p^2 = .938$ ). There was no significant interaction of edge type and task complexity. Bonferroni-corrected post-hoc tests showed that the glyph-based edges led to significantly higher task completion times ( $M = 7.65$  s,  $SD = 1.77$  s) compared to the tapered edges ( $M = 6.94$  s,  $SD = 1.71$  s). There were no other significant differences between the edge variants, including the animated edges ( $M = 7.02$  s,  $SD = 1.533$  s). As expected, the participants were significantly slower in the more complex tasks with path length two ( $M = 10.34$  s,  $SD = 2.05$  s) compared to the simpler tasks with path length one ( $M = 4.06$  s,  $SD = 1.41$  s).

## User ratings

The user ratings for the directed edges experiment (Figure 6b, c) showed no significant differences for the suitability of the three edge variants for the task ( $\chi^2(2) = 0.68, p = 0.713, N = 18$ ). In fact, the means for animated edges ( $M = 4.00, SD = 0.75$ ), glyph-based edges ( $M = 3.67, SD = 1.05$ ), and tapered edges ( $M = 3.78, SD = 1.03$ ) are very similar. In contrast, the scores for aesthetics diverge more with animated edges being rated highest ( $M = 3.83, SD = 0.96$ ), followed by glyph-based edges ( $M = 3.39, SD = 1.06$ ), and tapered edges ( $M = 2.89, SD = 0.87$ ). However, analysis showed no significance for these differences.

## Additional Findings

As mentioned earlier, *error rates* were generally low in our study. For the undirected edges, the average error rates were 6.65% for the straight edges ( $SD = 5.86\%$ ), 7.75% for the dashed edges ( $SD = 7.89\%$ ), and 8.83% for the curved edges ( $SD = 8.66\%$ ). The more complex trials with 36 nodes showed higher error rates ( $M = 8.25\%, SD = 8.27\%$ ) than the smaller graphs ( $M = 7.24\%, SD = 6.21\%$ ). However, an ANOVA did not show any significant effects or interactions. For the directed edges, animated edges had the lowest error rate ( $M = 3.72\%, SD = 5.62\%$ ), followed by the tapered edges ( $M = 4.69\%, SD = 6.11\%$ ), and the glyph-based edges ( $M = 5.38\%, SD = 6.81\%$ ). This was not significant. The complex tasks ( $M = 8.09\%, SD = 6.47\%$ ), however, showed significantly higher error rates ( $F(1,17) = 20.251, p < .001, \eta_p^2 = .544$ ) than the simple tasks ( $M = 1.19\%, SD = 2.47\%$ ). Note that the underlying data violated the normality assumption of the ANOVA, which can lead to an increased danger of false positive results.

We also looked into *gender specific differences*. Half of our 18 participants were male, half were female. The order of the tasks per experiment was partially counterbalanced between the genders, with each of the six orders of the two independent variable levels appearing at least once in both groups. We found no significant differences for the task completion times between female ( $M = 18.66$  s,  $SD = 5.35$  s) and male ( $M = 16.85$  s,  $SD = 4.78$  s) participants in the first experiment. Likewise, there were no significant differences between female ( $M = 7.42$  s,  $SD = 3.91$  s) and male ( $M = 6.98$  s,  $SD = 3.57$  s) participants for the second experiment. Due to the insignificant differences between the groups, we did not consider gender for the rest of the analysis.

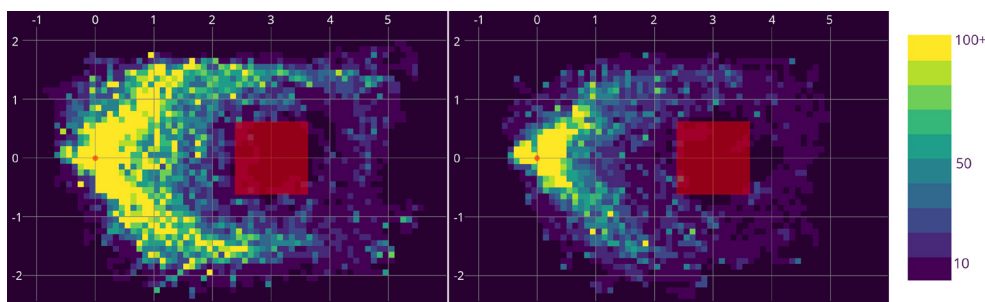


Figure 7. Position heatmap of all participants in the first experiment (left) and the second experiment (right). Colors represent the position count over all participants per 10 cm<sup>2</sup> cell (saturated at  $n \geq 100$ ). The red dot shows the starting position at (0,0); the red square shows the visualization boundaries. Scale in meters.

Based on the recorded position data and our observations during the study, we did notice some patterns regarding the *movement and exploration behavior* of our participants. Almost all participants tried to investigate from the outside. This shows in the position data (Figure 7) and corresponds with the advantages of an “outside-in” view.<sup>15</sup> Our users typically moved in circular motion from the start position until they found a position suitable to solve the task. Once there, they only moved very slightly for the duration of the trial. Several participants also initially stepped back from the starting position to get a better overview of the data. The range of movement typically increased during the study but only half of the participants used the full 360°

around the graph. In the second experiment (Figure 7, right), our participants moved less, probably because the graph complexity was lower and directed edges were easier to recognize. In their post study feedback, our participants stated that the presentation of a virtual graph in a physical environment did not confuse them. They also clearly stated that the background did not affect their performance.

## DISCUSSION

The results of our study show that all participants were able to solve the tasks, both for undirected and directed edges. Despite personal differences, there were no extreme outliers in terms of average task completion times or error rates. Overall, error rates were low. As expected, the more complex tasks (larger graphs or longer paths) proved to be more difficult.

For the undirected edges, we were interested how curved and dashed edges would compare to straight edges. Both variants address the challenge of occlusion. Compared to straight edges, the added distance-dependent curvature makes it less likely that large sections of curved edges overlap. Dashed edges, on the other hand, allow the user to look at edges behind them. However, in our study curved edges performed significantly worse, contrary to our hypothesis. According to our participants, they made it more difficult to follow the edge and did not indicate the direction of the edge as clearly as straight edges. For one participant they were even “confusing”. Our participants rated dashed edges worse than the other types. We believe that this is mainly due to the added visual clutter introduced by breaking up edges into multiple small objects. One participant mentioned specifically that this edge variant was “very busy” and crossings between edges were hard to perceive. However, this is not supported by the task completion times. Here, they performed as good as the straight edges. Based on these results, our recommendation is to use straight edges as the default. Other variants can be used to either encode additional data on the edge or, in the case of curved edges, to give a clearer view of the background or other visualizations in the environment.

For directed edges, we compared animated, glyph-based, and tapered edges, similar to Holten et al.<sup>2</sup> in their comparison for 2D graphs. In their work, they found that tapered edges had the shortest answer times and animated edges lead to the fewest errors. Our results partially confirm these findings for 3D graphs in Augmented Reality. Task completion times were significantly lower for tapered edges in comparison to glyph-based edges. Two participants described the glyph-based edges as “irritating” and “sometimes hard to differentiate”. Also, though not significant, the mean error rate for animated edges was the lowest among the tested alternatives. For the short path length condition using animated edges, our participants did not make any errors. However, all these differences are quite small. Based on the better aesthetics rating, we would recommend animated edges, which did not perform worse.

Overall, our results show that, despite some significant differences, all tested edge variants are in principal suitable for AR graph visualizations. This means we cannot give a single, clear recommendation for a variant. However, this also means that developers can freely choose from these (and possibly other, untested) alternatives when designing for a particular use case, which may add special requirements for the visualization.

## Limitations

As in all studies, there are some limitations and threats to validity in our experiments. We took great care to design a visualization that is generic enough to ensure the applicability of the results to different real-world use cases. However, this also means that for any particular use case, the graphs typically encountered might be different from our study’s graphs in structure and size. Specifically, we do not know if and how the results would change for much larger graphs. Another limitation are the edge styles themselves. For each of the variants tested, there are countless potential design alternatives. While we tried to refine each edge style in an iterative process including expert feedback, we cannot rule out that other parameter choices would have led to improved results. Finally, on the technical side, the participants reported no general problems with the rendering, e.g., regarding resolution. However, several participants criticized the limited FoV

of the HoloLens. Although this is a typical limitation of today's AR hardware, it could have a major influence on the users' exploration behavior and was one of the reasons we opted against larger, room-scale graphs for this study.

## CONCLUSION & FUTURE WORK

3D graph visualizations in AR will play a major role in future Immersive Analytics scenarios, be it for the visualization of datasets in a 3D analytics environment or even for graph data coupled to physical objects and locations themselves, visualizing connections, relations, and hierarchies. There has been a lack of research in regards to the basic questions of how to style edges and nodes or how to encode attributes with visual variables in such settings. The research that we presented in this article serves as one of the necessary steps to examine these questions in detail. Our results show that shape- or geometry-based approaches for edge variants in AR graphs are feasible and allow designers to choose techniques depending on their needs. In the future, we hope to examine multivariate data encodings on graph edges and nodes, as described in the design space. In addition, we also plan an in-depth look into specific use cases such as the analysis of networks and information flows in cyber physical production systems or other smart environments. There, real world aspects such as registration to physical objects and occlusion, both by and of physical objects, will also play an important role.

---

## ACKNOWLEDGEMENTS

We would like to thank Georg Eckert for his contributions to the initial concepts. This work was supported in part by grant 03ZZ0514C of the German Federal Ministry of Education and Research (measure Twenty20 – Partnership for Innovation, project Fast) and DFG grant CollabWall (DA 1319/11-1).

---

## REFERENCES

1. T. Chandler, M. Cordeil, T. Czauderna, T. Dwyer, J. Glowacki, C. Goncu, M. Klapperstueck, K. Klein, K. Marriott, F. Schreiber, and E. Wilson, "Immersive Analytics," in *2015 Big Data Visual Analytics (BDVA)*, 2015, pp. 1–8.
2. D. Holten, P. Isenberg, J. J. van Wijk, and J.-D. Fekete, "An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs," 2011 IEEE Pacific Visualization Symposium, Hong Kong, 2011, pp. 195-202.
3. H. Romat, C. Appert, B. Bach, N. Henry-Riche, and E. Pietriga, "Animated Edge Textures in Node-Link Diagrams: A Design Space and Initial Evaluation," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2018, pp. 187:1–187:13.
4. T. Bruckdorfer, S. Cornelsen, C. Gutwenger, M. Kaufmann, F. Montecchiani, M. Nöllenburg, and A. Wolff, "Progress on Partial Edge Drawings," *Journal of Graph Algorithms and Applications*, vol. 21, no. 4, pp.757-786, 2017.
5. W. Büschel, S. Vogt, and R. Dachsel, "Investigating Link Attributes of Graph Visualizations in Mobile Augmented Reality," in *CHI 2018 Workshop on Data Visualization on Mobile Devices. MobileVis '18*, Montreal, QC, Canada, 2018.
6. L. A. Crutcher, A. A. Lazar, S. K. Feiner, and M. X. Zhou, "Managing Networks Through a Virtual World," *IEEE Parallel Distrib. Technol.*, vol. 3, no. 2, pp. 4–13, Jun. 1995.
7. C. Ware and G. Franck, "Evaluating Stereo and Motion Cues for Visualizing Information Nets in Three Dimensions," *ACM Trans. Graph.*, vol. 15, no. 2, pp. 121–140, Apr. 1996.

8. D. Belcher, M. Billinghamurst, S. Hayes, and R. Stiles, "Using Augmented Reality for Visualizing Complex Graphs in Three Dimensions," in *Proceedings of the 2Nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, Washington, DC, USA, 2003, pp. 84–93.
9. F. Bacim, E. Ragan, S. Scerbo, N. F. Polys, M. Setareh, and B. D. Jones, "The Effects of Display Fidelity, Visual Complexity, and Task Scope on Spatial Understanding of 3D Graphs," in *Proceedings of Graphics Interface 2013*, Toronto, Ont., Canada, 2013, pp. 25–32.
10. O. Kwon, C. Muelder, K. Lee, and K. Ma, "A Study of Layout, Rendering, and Interaction Methods for Immersive Graph Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 7, pp. 1802–1815, Jul. 2016.
11. M. Cordeil, T. Dwyer, K. Klein, B. Laha, K. Marriott, and B. H. Thomas, "Immersive Collaborative Analysis of Network Connectivity: CAVE-style or Head-Mounted Display?," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 441–450, Jan. 2017.
12. G. V. Georgiev, K. Yamada, T. Taura, V. Kostakos, M. Pouke, S. T. Yung, and T. Ojala, "Augmenting creative design thinking using networks of concepts," in *2017 IEEE Virtual Reality (VR)*, 2017, pp. 243–244.
13. M. S. T. Carpendale, "Considering Visual Variables as a Basis for Information Visualisation," University of Calgary, Calgary, AB, 2003.
14. B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry, "Task Taxonomy for Graph Visualization," in *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, New York, NY, USA, 2006, pp. 1–5.
15. W. Qi, R. M. Taylor II, C. G. Healey, and J.-B. Martens, "A Comparison of Immersive HMD, Fish Tank VR and Fish Tank with Haptics Displays for Volume Visualization," in *Proceedings of the 3rd Symposium on Applied Perception in Graphics and Visualization*, New York, NY, USA, 2006, pp. 51–58.

---

## ABOUT THE AUTHORS

**Wolfgang Büschel** is a researcher at the Interactive Media Lab of Technische Universität Dresden, Germany. He studied computational visualistics and graduated with distinction in 2012. His research interests include advanced interaction techniques for augmented reality and immersive analysis. Contact him at [bueschel@acm.org](mailto:bueschel@acm.org).

**Stefan Vogt** studied computer science at Technische Universität Dresden. His research interests include immersive interfaces, evaluation in AR/VR, and location based information. Contact him at [stefan.vogt@htw-dresden.de](mailto:stefan.vogt@htw-dresden.de).

**Raimund Dachzelt** is a university professor at Technische Universität Dresden, Germany, where he heads the Interactive Media Lab Dresden. His research interests are 3D user interfaces, natural human-computer interaction, and information visualization. He is a member of the IEEE Computer Society, ACM SIGCHI, and SIGMM. Contact him at [dachselt@acm.org](mailto:dachselt@acm.org).