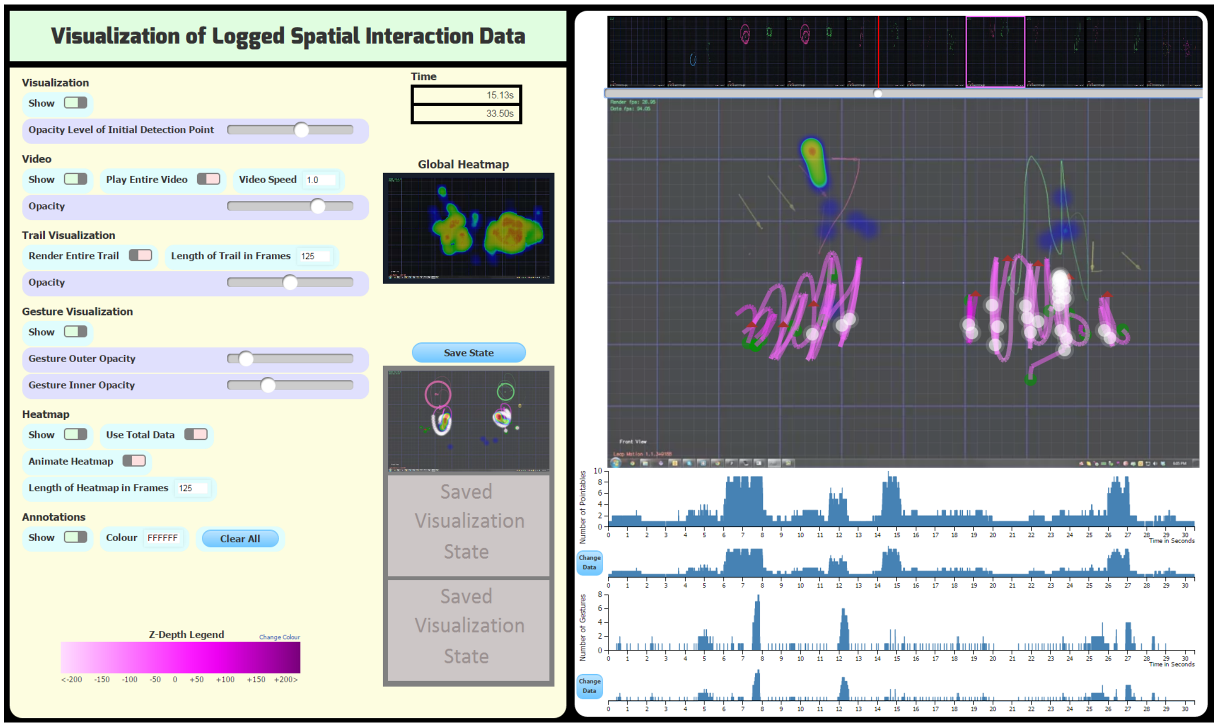
SpatialVis: Visualization of Spatial Gesture Interaction Logs

Erik Paluka and Christopher Collins


Visualization of Logged Spatial Interaction Data

Time: 15.13s / 33.50s

Video Speed 1.0

Length of Trail in Frames 125

Length of Heatmap in Frames 125

Colour FFFFFF

Figure 1: SpatialVis: Web application for visualizing logged spatial interaction data.

Abstract

This paper presents SpatialVis, a system for logging spatial gesture interactions and a visualization interface to analyze those logs. SpatialVis overlays a gesture log visualization atop screen recordings of interaction sessions to allow for replay of experimental trials. We discuss the challenges of logging spatial interactions and recommendations for best practices.

Keywords: interaction design, visualization, spatial gestures

1 Introduction

Spatial interaction devices, which enable the control of traditional pointers as well as the performance of single and multi-hand mid-air gestures to interact with computers, are becoming commonplace. Precise spatial gesture hardware such as the Leap Motion device allow for the design of gestures for interacting with visualizations, such as selection, zoom, filter, and other basic information visualization interactions [5]. They offer new capabilities to create visualization systems for use in environments where touch screens and mouse interaction are inappropriate, for example in sterile environments.

It is also possible to adapt existing systems for use with spatial gestures. An easy way to integrate spatial interaction on a desktop computer is to use spatial gestures to control the pointer. Although, this causes challenges since standard desktop graphical user interfaces (GUI) are designed for precise input devices such as the mouse. A typical virtual object's small display and interaction spaces reflect this, which can lead to problems selecting items as well as other fundamental tasks. To mitigate this problem, a designer can integrate concepts from techniques that facilitate target acquisition (e.g. Bubble Cursor [2]) into the mid-air selection gesture.

When an information space is larger than the display, it is typical for interfaces to only support interacting with content that is rendered within its viewport. To support interacting with off-screen content, our previous work [4] explored the design and evaluation of several spatial off-screen exploration techniques that make use of the interaction space around the display. These include *Paper Distortion*, *Dynamic Distortion*, *Dynamic Peephole Inset*, *Spatial Panning*, and *Point2Pan* (see Figure 2).

When implementing new interaction techniques for spatial interaction, as we did with Off Screen Desktop, it is beneficial to test the gestures with people who are not accustomed to them. To gather data for analysis, one can video record and/or observe people as they use the gesture in conjunction with a GUI, as well as administer post-questionnaires and interviews. Log data from the spatial gesture sensors can also be gathered. The problem with this is that, other than the video and observational data, logging techniques produce high frequency 3D data logs in text form. Long log files do not harness the full power of the human visual system; therefore making the analysis difficult.

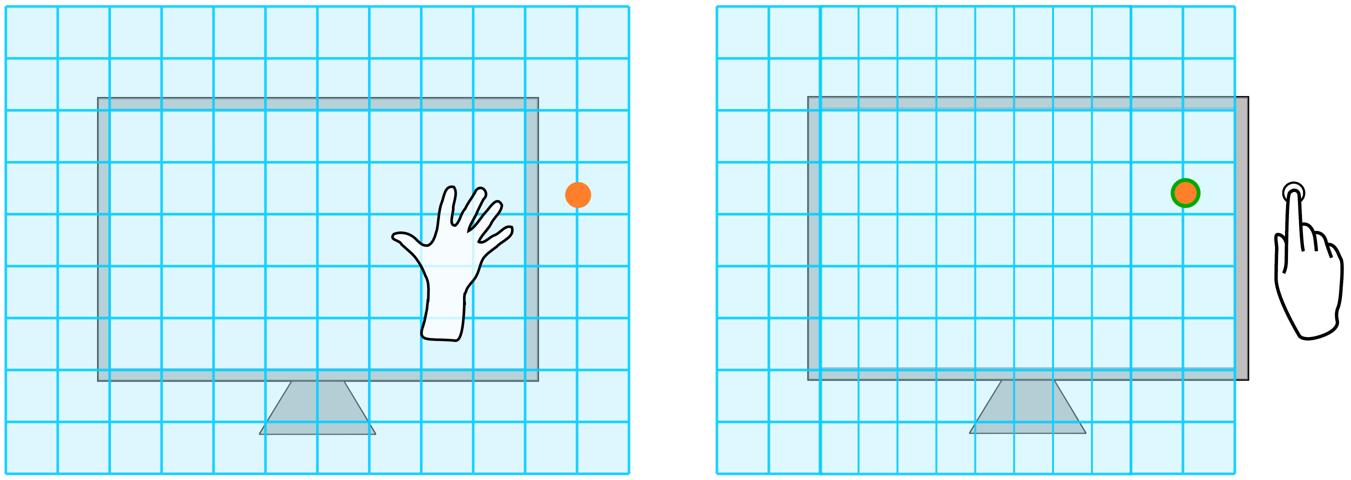


Figure 2: Off Screen Desktop geometrically transforms the visual presentation of the information space without affecting the interaction space to bring off-screen content into the viewport. Since the interaction space remains unchanged, users are able to directly manipulate off-screen content that has been brought onto the display by performing a spatial selection (e.g. tap or grab) in its original off-screen location beside the display. In the above figure, the Dynamic Distortion technique is being employed to transform the on-screen information to create room for off-screen content.

To mitigate these problems and help designers build better spatial user interfaces, as well as help us study our off-screen interaction system, we developed a web-based application that visualizes logged spatial interaction data. By first uploading a log file and an associated video screen capture of the display, an investigator can employ its features to analyze the 3D interactions and their effects on the graphical user interface. Our system is not meant to replace any other method, but to fit within the investigative process to gain further insight into the related phenomena.

We implemented the application using JavaScript, HTML, and the D3 visualization toolkit [1]. Our prototype supports the spatial interaction data types provided by the Leap Motion controller (see **??**) and assumes that the controller's interaction space is in front and centred with the display. We also created a modified version of the application to be able to handle interaction spaces at the sides of the display. We did this to visualize data gathered from the study of our off-screen interaction system in order to gain further insight into participant usage patterns.

2 SPATIALVIS

To use our system to analyze a spatial GUI, the application being tested must automatically log all associated spatial interaction data. A video of this interface must also be recorded, using screen capture software, with a video length equal to the period of time spent interacting with the interface. This allows log data to be mapped to the user interface events that occur in the video. When complete, the designer or investigator can upload the video and log files to our web application, which will then display the video on the right side of the interface with a heatmap and a path visualization overlaid on top of it. The system also includes a timeline situated above the video, graphs of the interaction data underneath the video, and a global heatmap and different controls to the left (see Figure 1). When the video is played, the overlaid visualizations display spatial interaction data that is temporally related to the video's current frame.

Going back to the spatial target acquisition example from above, the analyst can use our system in conjunction with observational notes or a video recording of the person performing the gestures. For example, this would allow one to view what data the motion sensing hardware is producing and if that matches up with the gesture that the person is trying to perform. If this analysis was done with logged data that was not visualized, the investigator would have to look through hundreds or thousands of lines of text and would be very tedious.

2.1 Video Timeline

The timeline is created by dividing the video into ten equally sized sections and using images from the beginning of each video segment to represent each section (see F in Figure 3). When a user hovers over one of the sections, its border changes colour (see purple box at the top of Figure 1) and they are then able to select the section to seek the video to the start of it. If a section is selected, then the heatmap will update to show data only from this section's time range. If the video is then played, it will stop playing at the end of the selected section unless a widget on the left side of the interface is toggled. The timeline also contains a slider widget for seeking, while hovering over its handle will cause the play, pause and restart video controls to appear beside it.

2.2 Spatial Interaction Graphs

The graphs below the video show spatial interaction information over the length of the video. Their size match the width of the timeline to allow a person to match the current time of the video (slider's handle and vertical line above) with the graph data, as well as to provide awareness of video's current time value. The graphs are also enabled with the brushing and linking [3] techniques. Therefore, if one discovers a time range with an interesting data pattern, the visual complexity of the interface can be reduced to allow the analyst to concentrate on this subset of data. This is accomplished by selecting the data or time range of interest, which will then cause the rest of the data to be filtered out (see B in Figure 3). This brushing will then be reflected in the other graph, as well as in the heatmap and path visualization that are overlaid on top of the video. The video is also seeked to the beginning of the time range associated with the brushed data and if played, will stop at the end of this range. If the user is interested in analyzing other spatial interaction data types, they can change the data visualized in each graph from a range of options including pointables (tools and fingers), tools, fingers, hands, all gestures, as well as each individual gesture type.

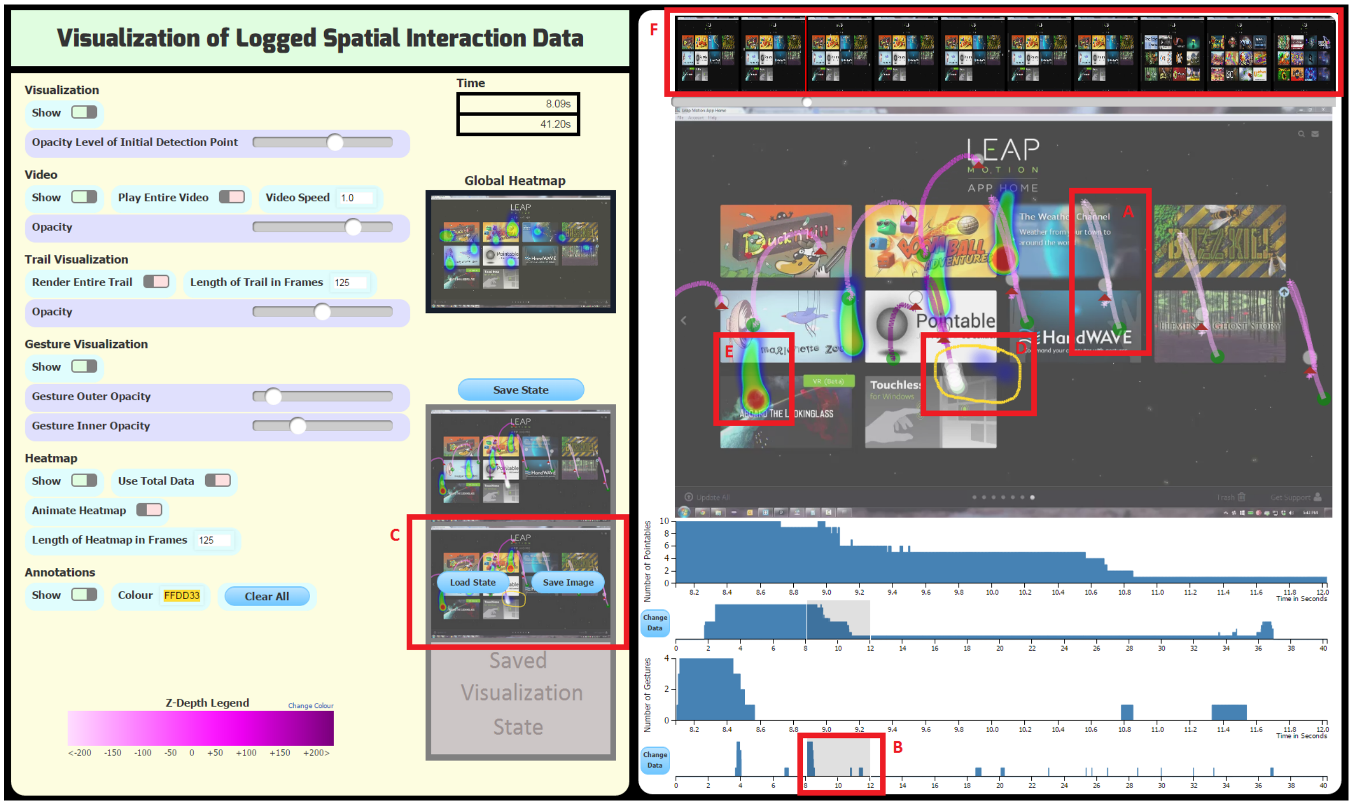


Figure 3: SpatialVis being used by an analyst. (A) Visualizing portion of spatial interaction data.(B) Brushing to show data only associated with 8 to 12 seconds into the video. (C) Saved visualization state. (D) User annotation. (E) Heatmap of data associated with 8 to 12 seconds into the video. (F) Video timeline.

2.3 Video Visualizations

We employed different visualization techniques to visualize each spatial interaction's location with respect to the user interface contained in the video. This was accomplished by overlaying them on top of the video using an orthographic projection mapping. We used a static heatmap to visualize the frequency of gestures that were performed at different locations. Data is selected to be visualized in the heatmap if its associated frame is within a non-sliding window. When the user first loads the required data into the application, the window is the size of the entire video; therefore all of the gesture data is initially visualized. If the video is played or seeked, then the window's starting frame is set to the seeked location or the beginning of the video segment being played. The window's ending frame is then calculated by adding a user-changeable value, contained in a widget, to the starting frame. Although, if the timeline sections or graphs are used to seek the video instead of the time slider, then the window's ending frame is set to either the timeline section's last frame or the last frame associated with the selected graph data. The interface also contains some other widgets that allow the user to set the window's ending frame to always be the last frame in the video, as well as to animate the heatmap over time using the data contained in its window.

We also visualized the path of each pointable (finger or tool) using a semi-transparent path. The pointable's Z-depth is encoded using colour with either a monochromatic or dichromatic divergent colour scheme. The path contains green semi-transparent circles to visualize the location of each pointable when it was first detected by the spatial interaction sensor. To visualize a pointer's spatial location in the current frame, a red semi-transparent triangle is attached to the end of the path. We also affixed a white semi-transparent circle to the path for visualizing the spatial location of different gestures, such as swipe, screen tap and key tap. For example, the white circles in Figure 1 show the location of discrete screen tap gestures. The visualization is dynamic since it displays data from a temporal sliding window that starts in the past and ends at the video's current frame. As the video plays, new data is visualized when it enters the sliding window and old data that is no longer inside the sliding window is removed. This aids the analysis process since interaction data would quickly disappear if only the current frame's data was visualized. The path visualization's sliding window is automatically set to a low value, but the user has the ability to change it, such as when one wants to visualize entire paths of all pointers.

In addition to the aforementioned visualizations, our system allows the analyst to create their own visual markings by providing video annotation abilities (see D in Figure 3), which can then be used to label the location of interesting events, for example.

2.4 Global Heatmap, Controls & Visualization States

Context is important for analysis, therefore we included a global context view of the gesture data with the use of a miniaturized image of the video that is overlaid with a heatmap that visualizes gesture data from the entire video. To further facilitate the analysis process, we also provide the ability to save and load different visualization states (see C in Figure 3). The video frame with the overlaid visualizations and user annotations that are associated with a saved visualization state can then be downloaded as an image for sharing and offline analysis. The interface also contains widgets on the left side to allow the investigator to show or hide the heatmap, path visualization, user annotations or the video itself. Opacity lev-

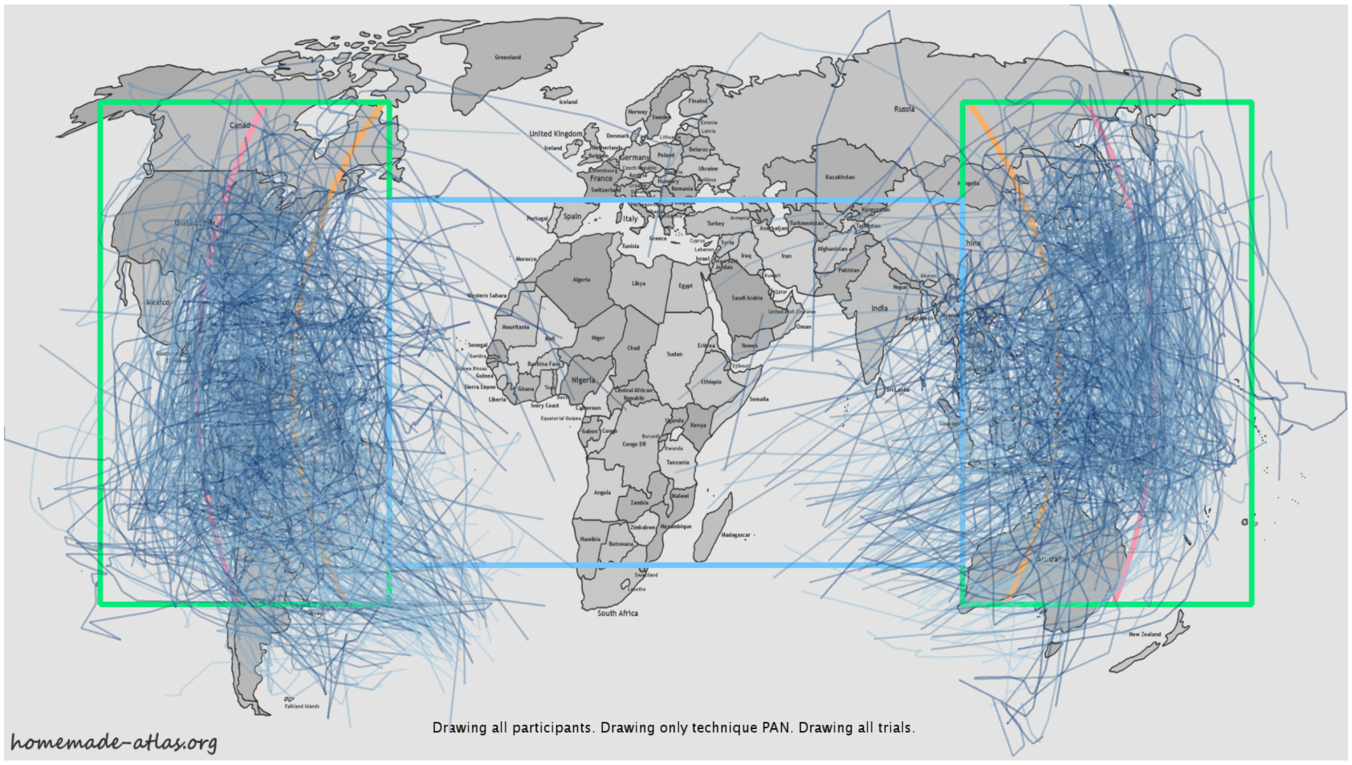

homemade-atlas.org
Drawing all participants. Drawing only technique PAN. Drawing all trials.

Figure 4: Path visualization showing the movement of all participants hands while searching the off-screen space with the Direct Spatial Panning technique.

els associated with the video and each component of the path visualization can be modified as well. The interface also contains widgets to allow the video's playback speed and the colour of the annotation brush to be changed.

3 Challenges

During the development and use of SpatialVis to support our research on the Off Screen Desktop project, we encountered several challenges which point to future research opportunities.

Color Scheme First of all, the color scheme of the overlay was difficult to design in a way that provided depth information as well as being discernible from the underlying application screen capture. Our solution to this is to allow the screen capture video to be toggled, but this is not ideal. In addition, in many cases the specific finger used in a gesture is important, and our visualization does not reflect this information. This could be encoded with color in the visualization, but additional colors would exacerbate the challenge of the visualization palette interacting with the screen capture.

2D projection The use of a 2D visualization (with depth of interaction encoded as color) to visualize a 3D interaction space results in some difficulty interpreting the resulting views. In particular, a lot of over-plotting can occur for even brief interaction logs. Our workaround for this is to limit the length of the spatial gesture interaction trails using the sliding window, but another alternative to explore would be the provision of a 3D reconstruction view of the spatial interactions.

Coupling Logging and Application

Interaction logging can be general (focused on the input device) or specific to an application (focused on high-level application events). In the case of a general application like SpatialVis, the interactions are logged by a process monitoring the input stream, and also recording the screen. The resulting log files from a variety of applications can be loaded into the same log analysis system. The advantage of this approach is that the logging and analysis system is generic and reusable. The disadvantage is that the logging is separated from the interaction events generated by the software which is being tested. For example, if a gesture is used to generate a selection event on a visualization application, or a specific data element receives a lot of interaction attention, SpatialVis has no knowledge of this. The analyst would have to derive this insight. If the spatial and interaction logging were both embedded in the application, potentially more useful details about how gestures trigger interface events would be available, at the cost of losing generality.

A potential compromise to this problem would be to create a standard logging format which application developers could use to output interaction event logs (including low level events such as button press and high level events such as filtering a view). These logs would be the same whether touch, mouse, or a spatial interface was used. Then these could be interleaved with the SpatialVis logs and screen recordings to create a unified analysis system without requiring integration of spatial interaction logging into the test application itself.

Scalability SpatialVis logs were useful in analyzing the behavior of individual experimental participants in detail. However, the visualization is not scalable to multiple participants as it quickly becomes too cluttered. Also, unless the screen capture is static, it does not make sense to overlay multiple screen captures. To see the trends for multiple participants in a repeated trial experiment, we had to create a simplified visualization we called PathVis (see Figure 4) which shows the spatial position of the pointer in the information space over time, overlaid for multiple participants. The spatial depth information as well as multiple fingers were removed from this view.

4 Conclusion

Spatial interaction is an emerging modality for many types of applications, including information visualization. We have presented an initial visualization application, available online at (URL to come for final version), for analyzing experimental trials of interactions with visualization applications using spatial gestures.

References

[1] M. Bostock, V. Ogievetsky, and J. Heer. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, Dec. 2011.

[2] T. Grossman and R. Balakrishnan. The Bubble Cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 281–290, New York, NY, USA, 2005. ACM.

[3] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, Jan. 2002.

[4] E. Paluka. Spatial peripheral interaction techniques for viewing and manipulating off-screen digital content. Master's thesis, University of Ontario Institute of Technology, Oshawa, Ontario, Canada, 2015.

[5] J. S. Yi, Y. a. Kang, J. Stasko, and J. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, Nov. 2007.