

Interactive Exploratory Visualization of 2D Vector Fields

Tobias Isenberg^{1,2} Maarten H. Everts¹ Jens Grubert² Sheelagh Cpendale²

¹University of Groningen, The Netherlands

²University of Calgary, Canada

Abstract

In this paper we present several techniques to interactively explore representations of 2D vector fields. Through a set of simple hand postures used on large, touch-sensitive displays, our approach allows individuals to custom-design glyphs (arrows, lines, etc.) that best reveal patterns of the underlying dataset. Interactive exploration of vector fields is facilitated through freedom of glyph placement, glyph density control, and animation. The custom glyphs can be applied individually to probe specific areas of the data but can also be applied in groups to explore larger regions of a vector field. Re-positionable sources from which glyphs—animated according to the local vector field—continue to emerge are used to examine the vector field dynamically. The combination of these techniques results in an engaging visualization with which the user can rapidly explore and analyze varying types of 2D vector fields, using a virtually infinite number of custom-designed glyphs.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques; I.3.m [Computer Graphics]: Miscellaneous—Visualization

1. Introduction

Vector field data arises in many scientific and technical applications. Thus, vector and flow field visualization has been an important research domain for visualization. Many successful techniques have been developed to help people understand the properties of such datasets. However, traditional vector field visualization typically relies on producing static images. For example, line integral convolution (LIC, [CL93]) or the extraction of topologic properties [LHZIP07] both produce one image per vector field (or possibly an animation or 3D shape) that can then be examined by the viewer. Using these techniques, exploration of a dataset is limited to setting parameters for the automatic image generation and then browsing through the final results. By providing techniques to interactively explore vector data in chosen regions using a set of custom-designed glyphs, we offer additional exploration possibilities that go beyond simple parameter changes of automated glyph placement algorithms.

Interactive exploratory visualization of vector fields allows people who need to analyze such information to probe vector data locally, to place multiple glyphs to show larger-scale properties, and to place glyph sources to explore the directional properties of the data. Our interface allows people to use hand postures to sketch custom glyphs that are

best able to reveal data properties and supports interactive distribution of these glyphs. This combination of custom-designed glyphs with direct-touch interaction through a minimalistic interface enables users to both explore the data in-depth as well as to annotate traditional vector field renditions. We emphasize with our approach the necessity of physically and intuitively interacting with visualizations, rather than just tweaking parameters and observing their effects in still images. While our system works best on touch-sensitive wall displays, it can also be used with desktop and mouse setups.

The paper is organized as follows. In Section 2 we review previous work related to our approach in the area of vector visualization. Next, Section 3 introduces our concept of interactive exploratory vector field visualization before we show possible application scenarios in Section 4. Section 5 reports on a preliminary evaluation of the presented techniques while Section 6 discusses technical aspects of the realization and points out some limitations. We conclude the paper in Section 7 and suggest directions for future work.

2. Related Work

Previous work in vector field or flow visualization can be classified in one of four categories [LHD*04]: direct visual-

ization, dense texture-based visualization, geometric visualization, and feature-based visualization. We discuss related approaches according to each of these categories.

Direct visualization techniques use a direct mapping from data to visualization to produce an overall picture of the flow. A common approach is to position glyphs at grid points to convey properties such as direction or velocity. A simple example is an arrow plot, but more complex glyphs have been used to convey, e. g., uncertainty [WPL96]. Alternatively, simulated painted brush strokes can be used as glyphs to obtain non-photorealistic visualizations of multi-dimensional data elements [HTER04, THE07]. Kirby et al. [KML99], inspired by layering in painting, use layers of glyphs and backgrounds to visualize different aspects of a vector dataset.

Texture-based techniques provide a dense view of the vector field by computing a texture that conveys both local and global properties. Similar to the direct techniques, these texture-based techniques also provide an overall picture of the flow. Line Integral Convolution (LIC) [CL93] is an early but commonly used texture-based technique. While many variations of the LIC technique exist, both in 2D and 3D, the basic idea is that a noise texture is smeared out in the direction of the vector field. More recent texture-based approaches are image-based flow visualization [vW02] and the level-set based dye advection approach presented in [Wei04]. Two-dimensional texture-based visualizations that have the same spatial domain as the original data can be used in our system as background visualizations.

The last two groups are geometric and feature-based methods. Geometric techniques integrate particle paths to form geometric objects such as streamlines, streaklines, or pathlines. These can show the long-term behavior of a field and are related to our approach as the glyphs we are using sometimes serve a similar purpose. Feature-based methods extract features from the vector field and visualize them, resulting in potentially less cluttered visualizations. Examples of such features include convergence, divergence, flow rotation, shear, and flow vorticity. A detailed account on feature-based flow visualization can be found in [PVH*02] or, more recently, for topology-based techniques in [LHZIP07].

As studies have shown that visualizations that are custom-designed (e. g., by artists) with adequate tools can yield better results than standard methods [KKVL05], and that different visualization techniques for vector fields are suited for certain tasks to varying degrees [LKJ*05], we believe that interactive exploration of vector data is a promising path to investigate. Most existing literature on this subject focuses on the visualization of 3D vector fields and on generating those visualizations at interactive frame-rates, only few approaches explore the interaction with the data. In [dLvW93] a method is presented that allows users to interactively probe and annotate 3D flow fields using complex glyphs that show field properties such as velocity, shear, and rotation. In addition, the approaches presented in [RAEM94, vTRvdM97,

KE01] allow people to customize glyphs to use for data visualization by associating data values with glyph parameters. However, these customizations are restricted to simple geometric 3D glyph shapes. An interactive approach using dye advection is presented in [WBE05]. This work along with earlier work [MS92, Wei04] relate to our system in that they discuss the placement of sources and interactively compute which part of the flow is affected by them.

3. Interacting with Vector Fields

Vector field visualizations tend to focus on creating global representations of the data. If the data sets are either large or complex, the resulting visualizations can have overwhelming visual complexity. As a result it can be hard to pick out pertinent details and difficult to use these visualizations to communicate convincingly. We begin to address this visual complexity by providing personalization tools that enable:

- **the creation and personalization of the actual glyphs to be used:** Glyphs can be designed to be more understandable to the targeted group of people, and custom glyphs may better reveal particular patterns in the data.
- **the placement, and/or animation of glyphs in specifically chosen locations:** This enables us to control the visual complexity by annotating visualizations in specific location only, extending these local annotations at will to increase the visual complexity at a rate chosen to maintain comprehensibility, and thus allowing our interaction techniques to be used to create even large visualizations.
- **emphasis and annotation:** The capability for local vector visualization combined with the use of both personalized glyphs and color provides for a great variety of possibilities for creating emphasis and adding annotation.

In keeping with our goals to provide simple, interactive exploration tools for vector data, we look to develop a simple interface by avoiding, when possible, a complex parameterization of the program control by, instead, utilizing natural hand postures. In this section we describe how these goals are realized. Building from the smallest component, the glyph, we start by describing how hand postures can be used to create a huge variety of personalized glyphs. Next, we outline the underlying framework that provides the animation and simplifies the localization of visual effects. Then Section 3.3 explains how, with the same postures, one can create and adjust the vector visualizations.

3.1. Hand-Designed Glyphs

Glyphs are frequently used to illustrate properties of vector fields as they can show local data parameters by adapting their orientation, size, and other properties. Also glyphs, by their shape, can communicate properties of the data. For instance, arrows are often used to indicate the directionality of a field. Thus, glyphs are a natural choice as elements to work with to enable users to interactively explore vector data.



Figure 1: Examples for hand-sketched directional glyphs.

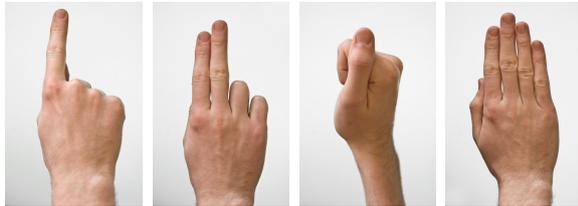


Figure 2: The four hand postures that can be recognized.

Traditionally, the design of glyphs used in vector field visualizations was finalized before run-time or even before compile-time. This can lead to sub-optimal visualizations as glyphs often depend on the type of data that is being used. Therefore, we allow individuals to personally sketch and design the glyphs that are used in visualizing the data (arrows, lines, etc.; see Figure 1). This provides the freedom to create glyphs that, for the intended usage, best reveal patterns in the current vector data. In addition, sketching custom glyphs can allow individuals to more rapidly test out and realize ideas with respect to given visualization tasks.

For the creation of personalized glyphs we support a number of tasks: people can indicate when they are creating glyphs and not interacting with the vector data, draw a variety of lines to design glyphs, and erase these lines or parts of them. The touch screen we are using, the SMART Technologies DViT, recognizes the center point of a touch interaction, the approximate size of the object that touches the screen, and detects when a pen has been picked up from a tray. We use this last function to signal a switch from vector data interaction to glyph drawing mode. This follows the notion that picking up a pen indicates the intention to sketch or draw.

Next, to actually draw the glyph, people can use the picked-up pen as well as hand postures to control the stroke width: using the pen provides the thinnest lines while using one finger, two fingers, and a fist creates increasingly thicker lines. The flat hand is used for erasing. These four simple hand postures are shown in Figure 2. For more details on the identification of possible hand postures see [GCI08]. In a nutshell, the postures can be distinguished by thresholding the touched surface area that the DViT provides. Factors that restrict the number of possible postures are the quality of the touch area data and that the hand postures need to be easy to form for human hands. Theoretically we would be able to track the lifting of more than one pen, increasing the possibilities for controlling aspects of the application. However, we

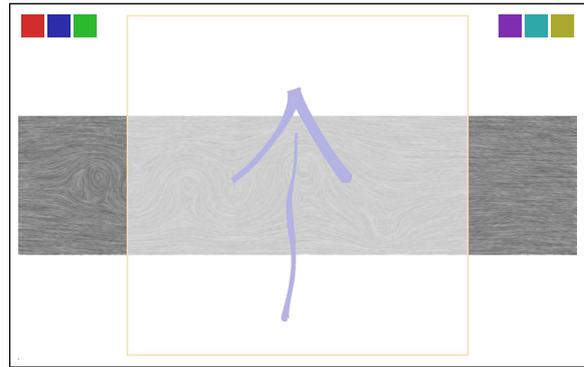


Figure 3: The area for sketching glyphs is displayed after a pen has been picked up from the tray. The stroke's width is controlled through postures as well as drawing speed, while erasing is possible using a flat hand.

decided against this as more options would be increasingly difficult to recall for users of the interface.

The described interactions allow people to sketch a wide variety of glyphs for use in vector field exploration and illustration (see example of sketching in Figure 3). As the vector field's direction is used to orient the glyphs, this consequently means that each glyph has an inherent directionality. We address this by employing the notion that the direction of a sketched glyph is straight up. In addition, the sampling point of the glyph is in the middle of the sketch area.

3.2. Interactive Glyph-Based Vector Fields

Using and interacting with a potentially high number of glyphs on a large display also poses a number of challenges. Interaction should offer a way to affect certain glyph properties. A system needs to be able to accommodate and render a high number of glyphs without losing interactive speed. For this purpose we make use of interface buffers (i-buffers, [IMC06]). These i-buffers store properties (e. g., size, orientation, or color) in a spatial manner so that primitives can access them locally to update their appearance accordingly. By interactively changing the buffer contents it is now possible to interact with the displayed primitives.

In order to visualize vector field data, we represent the vector directions in one of these i-buffers and the length of the vectors (i. e., the local strength of the field) in another one. These properties can now be used to control the appearance of glyphs whose location is projected to i-buffer positions. The vector field's direction is mapped to the orientation of the glyph while the field's strength is represented by the glyph size. As glyphs move across the buffers, they constantly check for new values with respect to their position and update their rendering accordingly. Alternatively, tools can modify the values of the i-buffers, to update the representation of glyphs reading from them. We make use of this latter technique, for instance, to control glyph transparency.

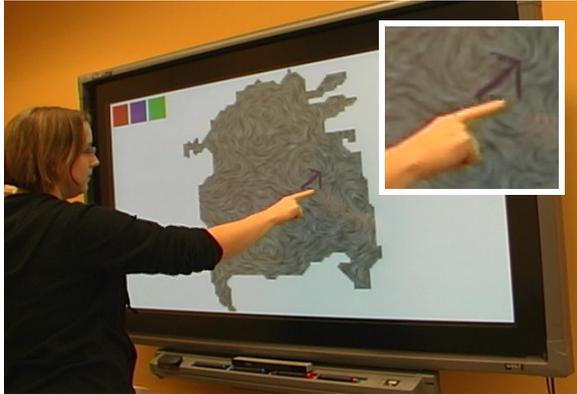


Figure 4: Probing the dataset with a single glyph and the one-finger posture. The glyph is off-set to ensure its visibility.



Figure 5: Exploring larger areas of the dataset using a fist.

3.3. Interactive Vector Field Exploration

Once custom glyphs have been designed, we use these to support the interactive exploration of vector field data. For this purpose we offer a second set of hand posture mappings for the phase when the pen is placed in its tray (one finger, two fingers, fist, and flat hand). These mappings need to support exploration tasks but also have to follow naturally from the postures to provide an intuitive interface. We selected four exploration actions for our visualizations: probing the vector field locally with a single glyph, placing a number of glyphs simultaneously at a larger scale, de-emphasizing and removing glyphs, as well as placing and re-locating sources that continuously emit glyphs which then start moving across the data, following the vector field's direction.

Probing the vector data is enabled with the one-finger posture. When just one finger is placed on the display, the system automatically creates a new instance of the most recently sketched glyph. While the finger is still touching the display, moving the finger also moves the glyph, whose rendition adapts to vector field's data values below its current position (Figure 4). As fingers are relatively blunt tools (compared to, e. g., a mouse pointer's tip), the glyph is off-set from the actual finger position to the top left (top right for left-handed people) to guarantee visibility [PWS88]. As soon as the finger is released from the screen, the glyph remains at its last location, visualizing the local conditions at this point.

While the above interaction allows users to explore the dataset locally, the fist posture is employed to explore the vector field at a larger scale. By placing the fist on and moving it across the screen, the system continuously emits new instances of the most recently sketched glyph at random locations [Dov95] within a radius around the interaction point (Figure 5). Using this technique, regions or the entire dataset can be covered with glyphs within a short time to explore the behavior of the field on a larger scale. We found relatively small and simple glyphs to be particularly useful for this type of interaction as they do not cause too much visual



Figure 6: Glyph de-emphasis & erasing using the flat hand.

clutter. The resulting images resemble those created by techniques which place short streamlines on vector data as well as traditional glyph-based visualizations.

The flat-hand posture provides de-emphasis or can completely erase glyphs (Figure 6), similar to its mapping in the sketching phase. De-emphasis of glyphs is realized by first increasing their transparency when the hand is first placed on the display and only deleting glyphs once their transparency has reached a given threshold. This de-emphasis can be used to provide an indication of the field's general direction and size properties. The probing interaction can be used on top of such a visualization to examine specific locations.

While static glyphs can be useful to explore a flow dataset, the display of animated glyphs following the streamlines can further aid the detection of patterns in the data. We offer exploration with animated glyphs through the use of the two-finger posture to create glyph sources that can be placed on and moved over the dataset. Each source continuously emits instances of the most recently sketched glyph. The instances are again generated at random positions within the radius of the glyph sources to allow users to explore local variations of

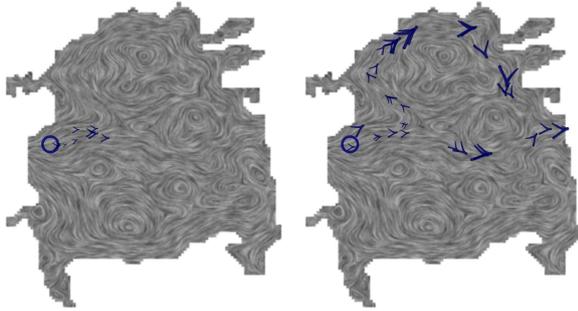


Figure 7: Using glyph sources to illustrate flow properties. Sources are placed with the two-finger posture.

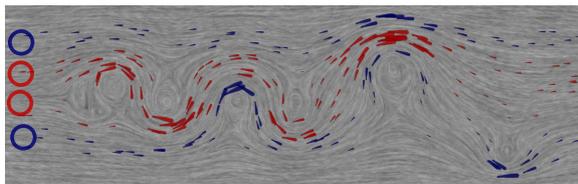


Figure 8: Using colored glyph sources to identify trends.

the data. These glyph instances, in contrast to the ones previously created with other postures, not only derive their size and orientation from the vector data but also move according to the streamline direction (Figure 7). This movement is realized by simple stepping along the local vector direction. Although it is not a physically correct integration of the vector field, it is sufficient to produce animations that help to illustrate the flow character and properties of a vector field.

Visual clutter is a problem in flow visualizations when too many glyphs are placed in close vicinity and start to overlap each other, as this makes it difficult to extract glyph properties from their representations. Visual clutter can become a problem with our interactive techniques when too many sources or individual glyphs have been placed. In order to disambiguate glyphs we provide the possibility to assign a color to individual glyphs and sources which then consequently color the glyphs that are generated by them. For this purpose the interface has “color pots” at the top of the screen into which sources or the probing finger can be “dipped,” assigning the respective color to them (Figure 8).

3.4. Interactive Visualization and Illustration

In the previous section we discussed how our techniques benefit the interactive exploration of vector field data. The same interaction techniques can also be used to interactively create static illustrations of a dataset. Illustrations can be used to emphasize interesting aspects in the vector field or to prove or disprove assumptions about a dataset. For these kinds of tasks we rely on representing some aspects of the vector data using glyphs and combining this with traditional

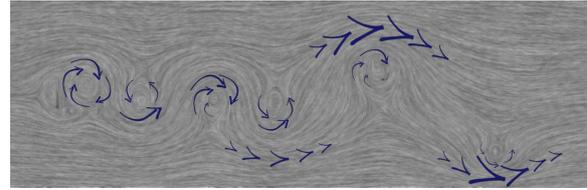


Figure 9: Annotation of a LIC image of a vortex simulation. Three hand-drawn glyphs are used to point out aspects that are otherwise not visible such as the field’s strength.

vector field visualization techniques. Examples for such traditional visualizations include LIC images, height images, vector field topology, vorticity images, or even a visual encoding of the vector data. Those visualizations are created in a pre-processing step and are rendered as a background. By using the same spatial domain as the original vector field and aligning the extra visualization layers with the dataset, people can match features in the traditional visualization with the properties that they see from the glyphs they place.

The same posture mappings are employed for the illustration of datasets as were used to enable exploration. Glyphs can be interactively placed using one finger or a fist as well as de-emphasized and removed using the flat hand. The probing interaction can now be used to place specially designed glyphs purposefully at locations that are worth pointing out. Glyph sources are typically not as useful in this case as they provide a different image at each frame, but may be used if desired. Generally, we found that more complex and elaborate glyphs are useful for illustrations since, when compared with exploration, fewer glyphs are typically placed to emphasize or annotate. This way users can create static images, using the interaction to show aspects of vector data that are otherwise not visible in a visualization. One example for such an illustration is highlighting the strength of the flow in a particular region in a LIC visualization (Figure 9).

4. Case Studies

To further illustrate the use of our methods we discuss them in the context of case studies of two simulated datasets: a water flow in a part of the Baltic sea and a fluid flow around a half-cylinder in 2D. Both datasets are time-dependent and we focus on visualizing individual slices. Users can cycle through the time slices, causing placed glyphs to adapt accordingly while additional glyphs can be added at any time. For both datasets we use additional visualizations (e. g., LIC and vorticity images) and annotate them.

4.1. Water Flow Simulation

To explore the larger scale water flow simulation dataset, one may start by sketching a small arrow glyph and then “flooding” the entire area with it using the fist posture (Figure 10). Once a large region of the field has been covered with glyphs,

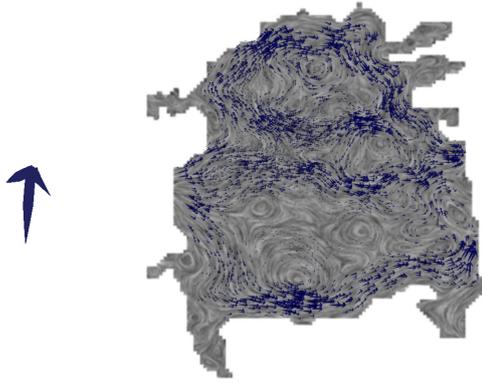


Figure 10: Exploring trends using the fist posture.

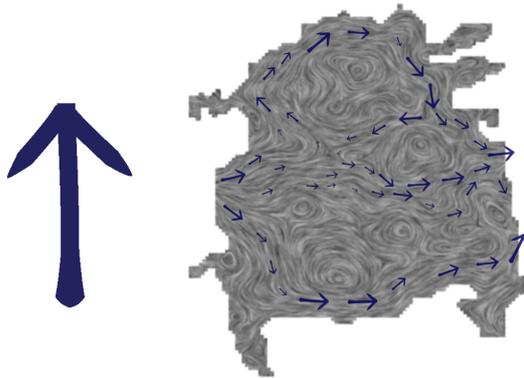


Figure 11: Illustrating the identified trends using the one-finger posture and a larger glyph.

the major streams in the flow and their strengths are revealed. After streams have been identified, one can delete all placed glyphs, draw a larger arrow as the next glyph, and use it to probe the field more precisely. By continuously leaving glyphs at interesting spots one can end up with an illustration of the major flows on top of the LIC background image (Figure 11). Finally, to confirm the flow characteristics with animated glyphs, one may draw a simple straight line glyph and place sources onto the dataset. By placing the sources at the inflow of the water body one can reveal how the glyphs proceed along the major streams (Figure 12, left). This flow pattern may change significantly when a different time step of the simulation is chosen which can reveal temporal relationships between the time slices (Figure 12, right).

4.2. Turbulence Simulation

For exploring the small-scale water flow simulation around a half-cylinder, one can start by drawing an arrow without the center bar and probing the dataset with it, using the LIC image in the background as a guide and leaving instances of the glyph as annotations (Figure 13, top). Next, using the same glyph, the fist posture is used to discover larger trends, in

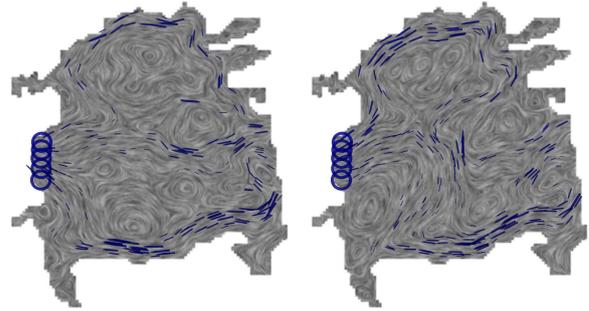


Figure 12: Using sources to show the flow at different times.

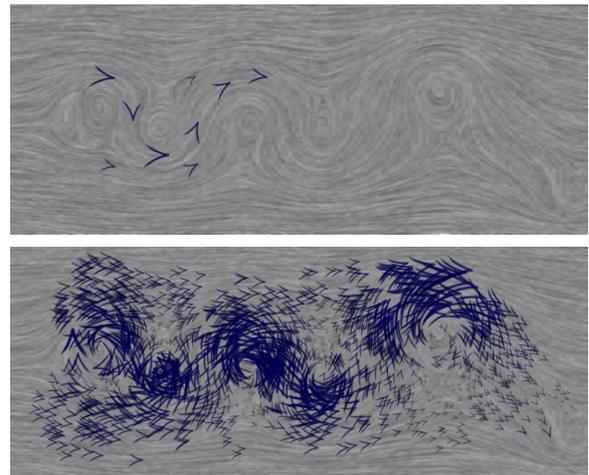


Figure 13: Probing the dataset with few glyphs vs. revealing the flow's strength and direction using many glyphs.

particular the strength of the flow (Figure 13, bottom). Now, to illustrate the different rotation directions of the vortices in the dataset, one can choose to draw an arrow that is bent clockwise, to show the vortices that rotate that way, and to repeat the same action with a mirrored arrow for vortices that rotate counter-clockwise (Figure 14, top). To further illustrate the differences, the second set of arrows can be colored in red to distinguish them from the first set which is blue. Finally, to confirm the findings of the previous step, the LIC image in the background is replaced by a visualization of the vorticity of the flow (Figure 14, bottom).

5. Preliminary Evaluation

In this section we report on some initial findings from domain scientists experimenting with our interaction techniques. We allowed them to work with both their own data as well as datasets from other sources (such as the wind data in Figure 15). Generally we received very positive feedback. In particular, people liked the possibility to explore aspects of new datasets to get an initial understanding: "This tool is very good when you do not yet understand your data."

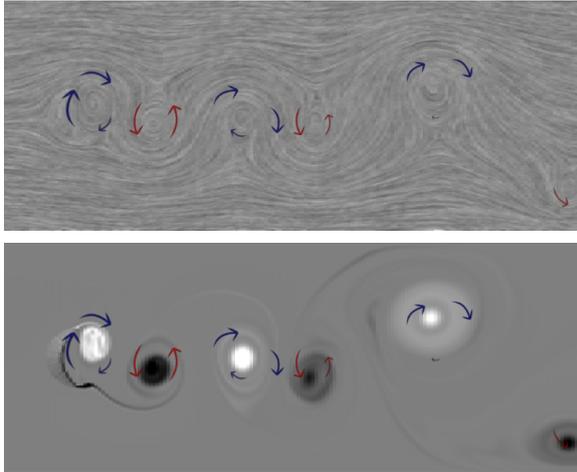


Figure 14: Illustrating the rotation direction using bent arrows and using a vorticity visualization for confirmation.

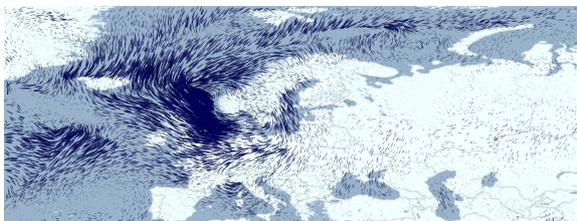


Figure 15: Visualization of storm Emma hitting Europe.

People said that it “is much easier and faster” to use than traditional tools (e. g., Gnuplot), in particular when specific glyphs are needed for highlighting or interpreting local features. Also, the option to place sources was liked very much. The interaction techniques were even seen as an interactive communication medium, for instance to discuss vector field properties of or to share insights about the dataset with colleagues. We also observed new ways to use the interaction techniques that we did not expect. For instance, one person drew a filled circle as glyph to only show the magnitude of a dataset, blending out its directional aspect. In addition, we received interesting suggestions for extensions such as an option to delete all generated glyphs at once, the use of entirely white backgrounds to only see the generated visualization, and the use of animation to provide a better overview.

6. Technical Realization Aspects and Limitations

The interactive vector field exploration system is implemented using OpenGL and relies on representing the vector data in i-buffers [IMC06]. Each glyph is represented by an OpenGL quad that is texture-mapped with the previously sketched glyph. During the glyph sketching phase, the strokes that form the glyphs are represented analytically in order to accommodate the easy erasing of parts of them.

Once the glyph is completed, the strokes are rendered in an off-screen buffer and copied into an OpenGL texture.

Each glyph instance, at each animation step, looks up the i-buffer values for its position, i. e., its center. If this position does not happen to fall exactly on one of the buffer values, the value is linearly interpolated. Transparency is maintained by each glyph individually and is also modified through an i-buffer. For this purpose, the tool that is invoked by the flat hand posture writes transparency offsets into this separate i-buffer, while glyphs use this value to update their specific transparency values. Glyphs are deleted once this value drops below a given threshold. Glyphs are also deleted when their size drops below a small threshold. This causes glyphs that move out of the dataset to be deleted automatically as the values in the background part of the size i-buffer are initialized to zero. This ensures that no glyphs are maintained or rendered longer than necessary, keeping the interaction responsive and the animation running smoothly.

Our entire interactive system is realized on the CPU with i-buffers and data being maintained in main memory. While this provides much flexibility for the implementation of the animation, it also has its limits with respect to how much data can be accommodated. For example, not the entire datasets shown in Section 4 (which contain a stack of time steps) can be converted to screen-sized i-buffer data directly, as this would require more memory than is available on typical PCs. Instead, whenever we switch between time steps, the current i-buffer set as well as the texture for the current background visualization are created from the data. However, the smaller glyphs are all maintained as textures on the graphics card and we did not notice any problems even when using a large number of different glyphs simultaneously.

While the presented approach works nicely for many datasets, it may not be as powerful with very noisy vector data that does not have strong “streams.” We experimented, for example, with vector fields extracted from diffusion tensor imaging (DTI) datasets which arise, for instance, in biology. In such cases where there are no clear trends it is also difficult to place meaningful glyphs with our technique to illustrate aspects of the dataset.

7. Conclusion and Future Work

In summary, we have presented a system for *interactive exploratory visualization of vector fields* on large, touch-sensitive wall displays. By allowing users to sketch their own glyphs we can let them create ones that work best for the specific data. These custom glyphs can then be used to interactively explore vector fields as well as annotate traditional visualizations of a dataset. The interface is controlled through hand postures which allow us to largely abstain from using keyboard or menus. Certainly, different mappings as well as interactions are possible to control such a tool. Also, some parameters such as specific stroke widths, animation speeds,

or thresholds to detect hand postures are hidden and/or empirically determined. We think, however, that limiting the number of exposed parameters in a minimalistic approach provides an intuitive and easy-to-understand interface for flexible, human-guided visualization. Our technique complements existing visualizations by facilitating intuitive explorations of data and later the illustration of specific aspects.

For future work we are considering a number of paths. For example, we would like to investigate how to show a third dimension of a vector field using either glyph shape or color. We also want to include a more precise integration method for moving glyphs. Special datasets may also require a way to deal with sinks inside the vector field. In this case we would need to track the position of glyphs to eventually delete them. In addition, other types of interaction could be investigated such as the zooming and/or panning of large datasets or the interactive placement of chains of glyphs. Finally, we are pursuing the use of different types of vector datasets from various sources as well as a formal study of the interaction techniques with domain scientists.

Acknowledgments

We thank Kurt Frischmuth (Univ. of Rostock, Germany) for the simulated dataset of the Greifswalder Bodden, Michel Westenberg (Univ. of Groningen, Netherlands) for the turbulence dataset, and GRIB.US for the weather data. We also thank Petra Isenberg for her help with creating the video, Henk Bekker and Samuli Ollila for discussions on the technique, and the anonymous reviewers for their valuable comments. We also thank our funding providers Alberta Igenuity, NWO, NSERC, CFI, iCORE, and SMART Technologies.

References

- [CL93] CABRAL B., LEEDOM L. C.: Imaging Vector Fields using Line Integral Convolution. In *Proc. SIGGRAPH* (New York, 1993), ACM Press, pp. 263–270.
- [dLvW93] DE LEEUW W., VAN WIJK J.: A Probe for Local Flow Field Visualization. In *Proc. IEEE VIS* (Los Alamitos, 1993), IEEE Computer Society, pp. 39–45.
- [Dov95] DOVEY D.: Vector Plots for Irregular Grids. In *Proc. IEEE VIS* (Los Alamitos, 1995), IEEE Computer Society, pp. 248–253.
- [GCI08] GRUBERT J., CARPENDALE S., ISENBERG T.: Interactive Stroke-Based NPR using Hand Postures on Large Displays. In *Eurographics 2008 – Short Papers* (Aire-la-Ville, Switzerland, 2008), Eurographics Assoc., pp. 279–282.
- [HTER04] HEALEY C. G., TATEOSIAN L., ENNS J. T., REMPLE M.: Perceptually-Based Brush Strokes for Nonphotorealistic Visualization. *ACM Transactions on Graphics* 23, 1 (Jan. 2004), 64–96.
- [IMC06] ISENBERG T., MIEDE A., CARPENDALE S.: A Buffer Framework for Supporting Responsive Interaction in Information Visualization Interfaces. In *Proc. C⁵* (Los Alamitos, 2006), IEEE Computer Society, pp. 262–269.
- [KE01] KRAUS M., ERTL T.: Interactive Data Exploration with Customized Glyphs. In *Proc. WSCG* (2001), pp. 20–23.
- [KKVL05] KEEFE D. F., KARELITZ D. B., VOTE E. L., LAIDLAW D. H.: Artistic Collaboration in Designing VR Visualizations. *Computer Graphics and Applications* 25, 2 (Mar./Apr. 2005), 18–23.
- [KML99] KIRBY R. M., MARMANIS H., LAIDLAW D. H.: Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting. In *Proc. IEEE VIS* (Los Alamitos, 1999), IEEE Computer Society, pp. 333–340.
- [LHD*04] LARAMEE R. S., HAUSER H., DOLEISCH H., VROLIJK B., POST F. H., WEISKOPF D.: The State of the Art in Flow Visualization: Dense and Texture-Based Techniques. *Computer Graphics Forum* 23, 2 (June 2004), 203–221.
- [LHZP07] LARAMEE R., HAUSER H., ZHAO L., POST F.: Topology-Based Flow Visualization, The State of the Art. In *Topology-based Methods in Visualization*, Hauser H., Hagen H., Theisel H., (Eds.). Springer-Verlag, Berlin, 2007, pp. 1–19.
- [LJK*05] LAIDLAW D. H., KIRBY R. M., JACKSON C. D., DAVIDSON J. S., MILLER T. S., DA SILVA M., WARREN W. H., TARR M. J.: Comparing 2D Vector Field Visualization Methods: A User Study. *IEEE Transactions on Visualization and Computer Graphics* 11, 1 (Jan./Feb. 2005), 59–70.
- [MS92] MA K.-L., SMITH P. J.: Virtual Smoke: An Interactive 3D Flow Visualization Technique. In *Proc. VIS* (Los Alamitos, 1992), IEEE Computer Society, pp. 46–53.
- [PVH*02] POST F. H., VROLIJK B., HAUSER H., LARAMEE R. S., DOLEISCH H.: Feature Extraction and Visualization of Flow Fields. In *Eurographics 2002 State of the Art Reports*. Eurographics Assoc., Aire-la-Ville, Switzerland, 2002, pp. 69–100.
- [PWS88] POTTER R. L., WELDON L. J., SHNEIDERMAN B.: Improving the Accuracy of Touch Screens: An Experimental Evaluation of Three Strategies. In *Proc. CHI* (New York, 1988), ACM Press, pp. 27–32.
- [RAEM94] RIBARSKY W., AYERS E., EBLE J., MUKHERJEA S.: Glyphmaker: Creating Customized Visualizations of Complex Data. *Computer* 27, 7 (July 1994), 57–64.
- [THE07] TATEOSIAN L. G., HEALEY C. G., ENNS J. T.: Engaging Viewers Through Nonphotorealistic Visualizations. In *Proc. NPAR* (New York, 2007), ACM Press, pp. 93–102.
- [vTRvdM97] VAN TEYLINGEN R., RIBARSKY W., VAN DER MAST C.: Virtual Data Visualizer. *IEEE Transactions on Visualization and Computer Graphics* 3, 1 (Jan.–Mar. 1997), 65–74.
- [vW02] VAN WIJK J.: Image Based Flow Visualization. *ACM Transactions on Graphics* 21, 3 (July 2002), 745–754.
- [WBE05] WEISKOPF D., BOTCHEN R. P., ERTL T.: Interactive Visualization of Divergence in Unsteady Flow by Level-Set Dye Advection. In *Proc. SimVis* (Erlangen, 2005), SCS European Publishing House, pp. 221–232.
- [Wei04] WEISKOPF D.: Dye Advection without the Blur: A Level-Set Approach for Texture-Based Visualization of Unsteady Flow. *Computer Graphics Forum* 23, 3 (Sept. 2004), 479–488.
- [WPL96] WITTENBRINK C. M., PANG A. T., LODHA S. K.: Glyphs for Visualizing Uncertainty in Vector Fields. *IEEE Transactions on Visualization and Computer Graphics* 2, 3 (Sept. 1996), 266–279.