Eurographics Conference on Visualization (EuroVis) 2016
E. Bertini, N. Elmqvist, and T. Wischgoll
(Guest Editors)

Short Paper

Crystal Glyph: Visualization of Directional Distributions Based on the Cube Map

X. Tong†1, H. Zhang‡2, C. Jacobsen§1, H.-W. Shen¶1 and P. McCormick‖3

[1]The Ohio State University, Columbus, Ohio, United States
[2]Northeast Normal University, Changchun, China
[3]Los Alamos National Laboratory, New Mexico, United States

Abstract

High resolution simulations are capable of generating very large vector fields that are expensive to store and analyze. In addition, the velocity fields generated from some particle simulations are not stored on spatial grids, which become difficult to visualize using some traditional vector field visualization methods such as streamlines. Furthermore, the noise and/or uncertainty contained in the data often affects the quality of visualization by producing visual clutter that interferes with both the interpretation and identification of important features. An alternative approach is to store the distribution of many vector orientations and visualize the distribution with 3D glyphs. This paper presents the cube map histogram, a new data structure for storing the distribution of three-dimensional vector directions. We also present a glyph called the crystal glyph that effectively visualizes the directional distribution using OpenGL cube map textures. By placing crystal glyphs in the 3D data space, users can identify the directional distribution of the regional vector field from the shape and color of the glyph without visual clutter.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications—

1. Introduction

With increases in spatial resolution and the emergence of ensemble simulations, massive amounts of data with uncertainties are commonly generated by simulations. In addition, storing and analyzing datasets at the full spatial or ensemble resolution has become unrealistic due to the required storage space and computation. On the other hand, down-sampling the data to a lower resolution loses the details of data. Using data aggregation, such as a histogram, becomes a trade-off between data size and details. Histograms can be generated from aggregations of spatial partitions, ensemble members, or an analyzed distribution models. The resulting histogram can be used to describe and detect features.

Visualizing the movement of a group of objects can assist the understanding of the motion of a large number of objects in the particle simulations of cosmology simulation [WFH*14] and fluid dynamics [Kuh14]. Plotting arrows to visualize the velocities of significant numbers of particles, however, is unrealistic due to the resulting visual clutter. Integrating streamlines or stream surfaces based on the particle velocities does not make sense if the data only describe local motion. To address the issue, we can use histograms of the velocities to represent the velocity distribution for a group of vectors and visualize the histograms. Neuroth *et al.* proposed two-dimensional (2D) velocity histogram to interactively visualize the large-scale velocity field. However, visualizing local 3D vector distributions remains unsolved.

Glyph-based visualization is an ideal approach to visualize multivariate data. It allows users to quickly perceive the pattern of multivariate data items within the context of a spatial relationship. Polar histograms [BK91] visualize the distribution of a group of 2D vectors based on their angles. Placing many polar histograms as glyphs in the data space can visualize the distributions of groups of 2D vectors in local regions. Jarema *et al.* designed similar glyphs to visualize 2D directional distributions in 2D vector field ensemble datasets [JDKW15]. The spherical histograms proposed in [GJL*09, WGJL12, PGN*14] can visualize distributions of three-dimensional vector fields, but their sphere partition methods are based on spherical coordinate system that suffers from polar effects and inaccurately represent the distributions. To better visualize the distribution of 3D velocities, we need an accurate 3D visual representation of directional histograms.

In this paper, we first introduce the concept of cube map histogram, a 3D directional histogram inspired by the environment cube map algorithm in computer graphics [Gre86]. The cube map

† tong@cse.ohio-state.edu
‡ zhanghj167@nenu.edu.cn
§ jacobsen.44@buckeyemail.osu.edu
¶ hwshen@cse.ohio-state.edu
‖ pat@lanl.gov

DOI: 10.2312/eurovisshort.20161154

histogram can be efficiently and accurately computed, stored, interpolated and visualized. We introduce a 3D directional distribution glyph called crystal glyph. To create the crystal glyph, an OpenGL cube map texture is used to efficiently map the cube map histogram onto a sphere and deform its shape through OpenGL shading language (GLSL). We place the glyphs on a slicing plane and interactively change the plane's orientation and position as well as the glyph's density to look at the velocity distributions in different spatial locations and with different levels of details. Users can use mouse to pick one glyph and see the unfolded view of its cube map histogram without occlusion. To provide a dynamic and intuitive visualization of global vector directions, we animate a texture on the glyph's surface that follows the velocity directions.

2. Related Works

Two-dimensional vector field histograms have been widely used in computer vision to analyze the statistics of directions or orientations. The applications include real-time obstacle avoidance in mobile robots [BK91], crowd flow abnormality detection [ID08], and human detection using HOG [DT05].

Glyphs are very powerful in visualizing two-dimensional flow fields because there are less concerns of visual clutter or occlusion in 2D images [KML99, PL08]. In 3D flow visualization, the vector glyph can be used as a simple and direct rendering of the local vector field [CM92, Dov95] or to show characteristics of the flow such as velocity and velocity gradient tensor [dLvW93]. The avoidance of visual clutter and the occlusion of important details become important factors in determining the effectiveness of these glyph-based techniques [BP96, Lar03].

Our work is closely related to vector field uncertainty glyphs that show the uncertainty information of local regions within a vector field [LPSW96, WPL96, ZDG*08, HLNW11, PGL*12, JDKW15]. Most of the techniques [WPL96, ZDG*08, HLNW11, PGL*12, JDKW15] only visualize the measures of distribution (e.g. mean, range and variance) or the models of distribution (e.g. GMM). Our work is also related to the angular distribution in the high angular resolution diffusion imaging (HARDI). The angular distributions are described by certain models or functions, which can be visualized by 3D glyphs [PPvA*09, JPGJ12, SSSSW13].

3. 3D Directional Histogram

A 3D directional histogram records the distribution of 3D unit vectors. For 3D velocity data, we omit their velocity magnitude, and only record their directions in our histogram. A bin in this histogram represents a range of similar 3D vector directions. For 2D vector histogram, each bin is a fan shape and all bins have the same size. For 3D vector histogram, binning the 3D vectors is the same as partitioning the sphere surface into small patches. Each patch represents a bin, and its area describes the bin size. In a unit sphere, this patch area is equal to the solid angle, a two-dimensional angle in three-dimensional space, subtended from the sphere center. When bin sizes are equal, the probability density of a bin is proportional to the bin frequency (bin counts); when the bin sizes are not equal, a bin's probability density is equal to its frequency divided by the bin size. It is difficult to have small bins of the same shape and size for 3D directional histogram because partitioning a sphere into small same shape patches is non-trivial. In order to compute an accurate probability density of a bin, we have to compute the sizes of all bins.

To represent a 3D directional distribution, we need to design a histogram whose bins partition a unit sphere surface. This histogram should be easy to be computed and interpolated to a continuous distribution. Westerteiger *et al.* used HEALPix grid [WGH12] to decompose the sphere surface hierarchically and record the grid in the GPU memory for rendering the terrain on the sphere. Even though this hierarchical grid is too complex to record our simple histogram, it inspires us to use a grid that is easy to be recorded and looked up in the GPU memory. There are existing methods to partition the sphere into patches of equal area or similar areas. However, they are either not easy to determine the bin, such as the icosahedral-based grid (triangles or hexagons) and unstructured grid, or difficult to interpolate, such as the spiral grid [HS08], Fibonacci grids [SJP06], Leopardi [Leo06]'s grid, and spherical polar grid (latitude-longitude). On the other hand, the cubed grid-based sphere partitioning, as shown in the Figure 1, which produce our cube map histogram, satisfies our requirements.

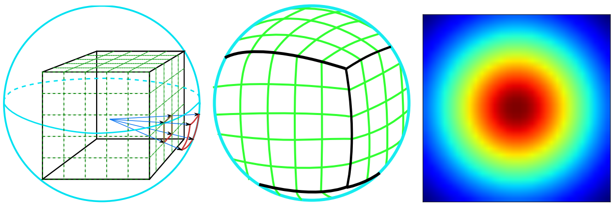


Figure 1: (a) The projection between a patch on the sphere surface and a grid cell on the inscribed cube. (b) Cube map grid on the sphere. (c) Solid angles for the cells on one cube face.

Our cubed grid on the sphere is created by projecting a cubic uniform grid onto a sphere surface as shown in Figure 1a. We use the grid cells as the histogram bins of our cube map histogram. From another perspective, if we cast a ray from the sphere center following a 3D vector, the vector belongs to the bin of the cell that it intersects with. How to compute the cube map bin based on a given 3D vector coordinate is computationally simple and is explained in the original environment mapping paper [Gre86]. The bin of a vector can be determined by simple division among its x, y, z coordinates. Interpolating the cube map histogram is mostly a simple bilinear interpolation among its easily determined adjacent bins. When rendering the cube map histogram, the OpenGL cube map texture and the GLSL sampler can take care of all the adjacency lookups and value interpolation automatically and efficiently.

To normalize the histogram, we need to compute the histogram bin size or the solid angle of each partition that is the sum of the two constituent spherical triangles' solid angles. The spherical triangle's solid angle is equal to the spherical excess that can be computed by l'Huilier's theorem [Zwi95]. Figure 1c shows the solid angles of the cells (or bin sizes) on one face of a high resolution cube map. We notice that the bins around the center have larger solid angles than the bins near the boundaries.

4. Directional Distribution Visualization

In this section, we describe how to use crystal glyph to visualize the directional histogram in 3D space.

4.1. Glyph Design

Designing the glyph is essentially mapping two data attributes of each histogram bin, 3D vector direction and the probability density, to the visual channels of a glyph, such as size, color, shape and orientation. In our design, we follow Borgo *et al.*'s thirteen general considerations and guidelines of glyph design [BKC*13].

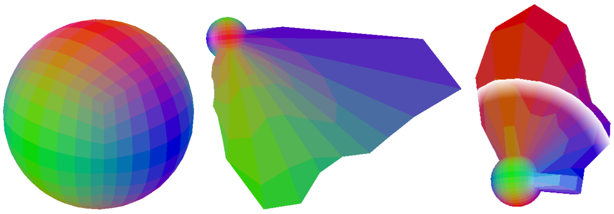


Figure 2: (a) The glyph is a sphere before drawing a histogram on it. (b) is the crystal glyphs showing the velocities distributed i y and z directions; (c) shows a white band moving from the spherical base outward as animation.

Our crystal glyph is generated from a sphere, whose grid is formed by projecting from an inscribed cube grid, as shown in Figure 2a. Note that the grid does not have to match the grid on the cube map histogram, because we can easily sample the cube map histogram and interpolate the values. Each face of the spherical mesh is a quadrilateral with a uniform color that is determined by the sphere's normal direction $\vec{v} = (x,y,z)$, a unit vector, at the center of the quadrilateral. The absolute values of the unit vector's components are used as the red, green and blue color components, i.e. $(r,g,b) = (abs(x), abs(y), abs(z))$. The XYZ-RGB color mapping scheme has been widely used in showing the 3D orientation in tensor data [PP99, Tuc04]. A very light Phong reflection is added to the color to provide depth cues without changing its original colors much. Using a uniform color on each face shows the mesh grid that helps viewers perceive the surface curvature [BS16]. We define p_v as the probability density of direction $\vec{v}$ in the directional distribution. Then as shown in Figure 2b, we extrude each sphere vertex along its normal direction $\vec{v}$ by an amount $H(p_v)$ that is a monotonically increasing function of p_v. This sphere surface extrusion form a terrain on the sphere. Higher terrain on the sphere surface corresponds to a higher probability density. Mapping the vector direction to the extrusion direction of the patch is intuitive and natural. As shown in Figure 2c and the accompanying video, a white band on the glyph moves from the spherical base outward by following the surface's extrusion directions, which animates the velocity directions. The animation is implemented in the OpenGL fragment shader with high performance.

4.2. Visualization System Design

From a glyph's screen projection as in Figure 3a, the bins on the backside are not visible if viewing from a fixed view direction. In order to help users explore a specific glyph of interest and observe its represented cube map histogram without occlusion, we visualize the cube map histogram by unfolding it onto a 2D image. Figure 3 gives an example about how to unfold (or project) the cube map histogram. Two types of projections are provided: one (Figure 3b) is created by cutting along the cube edges and unfolding the 6 cube faces without distorting the grid; the other one (Figure 3c) is made by projecting to 2D spherical coordinates [Bou06]. In the 2D views, the bin densities are mapped to color brightness on the image with a gray-scale colormap. A colored wireframe of the histogram bins is overlapped on the image to discretize the image into bins. By using the same color as the bins on the crystal glyph, users can easily find out their correspondence. The six face view of the cube map histogram in Figure 3b shows the bins using the same size, which is good for visualizing the spread (or range) of the distribution. On the other hand, the spherical coordinate view in Figure 3c gives mostly continuous bins but distorted bin shapes, which is good for identifying the number of peaks in the distribution. Even though the 2D unfolded views of the cube map histogram do not have occlusion issues, they could not intuitively visualize the bins' corresponding 3D directions as the crystal glyph does.

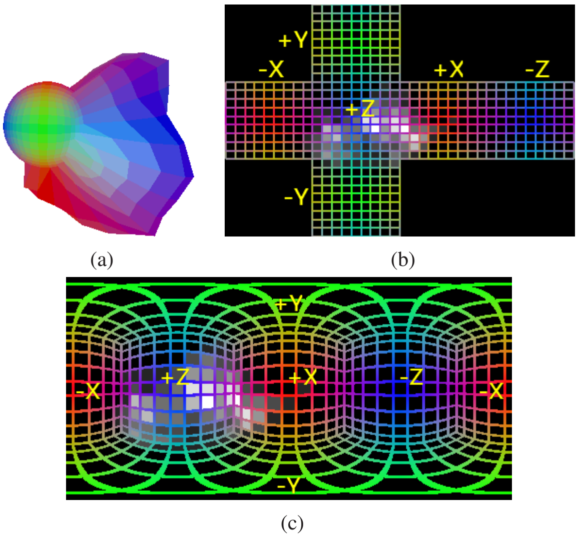


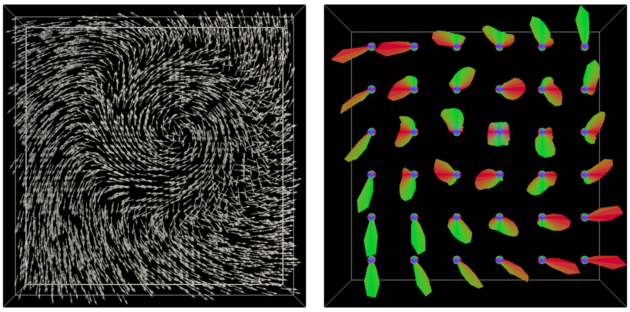
Figure 3: (b) and (c) are two different unfolded views of the cube map histogram on the user selected glyph (a).

5. Case Studies

To explore the effectiveness of our technique we applied our algorithm to two vector field datasets from different application domains: a synthetic tornado dataset (regular grid) and a cosmology dataset (particles). More details can be found in the accompanying video.

5.1. Tornado

The synthesized 48 × 48 × 48 tornado dataset is used as an example. Figure 4 shows the visualization of the 3D velocity field in the bottom 48 × 48 × 8 layer using both an arrow plot and our crystal



(a) 3D arrow plot. (b) Crystal glyph.

Figure 4: Visualizing the flow in the tornado dataset.

glyph. In the arrow plot (Figure 4a), 5000 arrows overlap with each other and result in severe visual clutter. On the other hand, in our crystal glyph visualization (Figure 4b), glyphs are uniformly put on a slicing plane to represent the vector field of a layer. Note that the glyph density can be interactively adjusted by users to visualize the distributions in different levels of details. They well separated without visual clutter and have different fan shapes indicating the vector distributions of different peaks and variances. The glyph at the 3rd row and 4th colume has velocity distributed in all directions in this slicing plane, which means the tornado center is nearby.

5.2. Cosmology

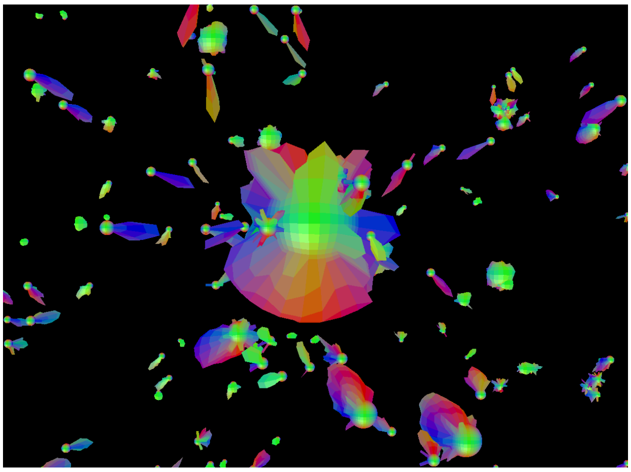


Figure 5: Crystal glyphs are used to visualize halos containing moving particles in the cosmology dataset.

Our goal in this case study is to visualize the particle velocities in the cosmology dataset from the Dark Sky Simulations [WFH*14]. The existing visualization techniques for cosmology datasets only visualize the particles' positions but not their velocities. When the vector directions are visualized using traditional techniques, the large number of particles and their noisy moving directions introduce severe visual clutter. The last time step of the dataset, which contains 3D velocities of 2,097,152 particles, is used. We visualize the velocity direction distributions of 7,383 halos, which are groups of gravitationally bounded particles. The directional distribution of the particle velocities in every halo is computed as cube map histogram and visualized as a crystal glyph. The glyph size is scaled using the radius of the represented halo. In the middle of Figure 5, there is a big halo whose velocity directions spread across many directions. Most of the surrounding halos are attracted by the gravity of the big halo and move towards the center. From the purple glyphs on the upper-left and upper-right of the image, we see that their velocities spread in a small range of directions. From the two reddish glyphs at the bottom of the image, we see that their directions spread in a big range and have outlier directions as the small green peaks.

6. Performance

We measured the performance of our technique on a machine running Windows 7 with an Intel Core i7-4770 CPU, 16 GB RAM and an NVIDIA GeForce GTX 980 Ti GPU with 6GB of frame buffer memory. Four resolutions (128^3, 256^3, 512^3 and 1024^3) of a regular grid vector field dataset are used. In the computation, we first divide the data into a fixed number of $128 \times 128 \times 128$ partitions and then compute a histogram for each partition. In this way, no matter what resolution the original vector field dataset is, we produce the same number of histograms. OpenMP is used to accelerate the computation with multithreading. Partitioning the space into $128 \times 128 \times 128$ partitions and computing histograms for all the partitions only took at most 5.3 seconds. The generated histograms can be either saved in disk or directly used as input to crystal glyph rendering. After glyphs' positions and sizes are decided, we aggregate the already computed histograms in each glyphs' represented region into a $9 \times 9 \times 6$ cube map texture. Computing cube map textures for all glyphs took less than 0.1 seconds. In the worst case when rendering 16,384 glyphs on a 1419×996 image, the frame rate is more than 30 frames per second (FPS). We have not tested more than 16,384 glyphs to avoid visual clutter.

7. Conclusion

We have presented a technique to efficiently compute and store distributions of three-dimensional vector directions using a cube map histogram. Besides, we designed the crystal glyph to visualize local 3D directional distributions with the OpenGL cube map texture. To allow users to freely explore the vector field, we designed an interactive visualization system. This paper is a work in progress, because we think the rendering of the crystal glyph can be further improved. A potential improvement can be visualizing the reference of the terrain heights, because the spherical base of the glyph is not always easy to see.

8. Acknowledgments

This work was supported in part by NSF grants IIS-1250752, program manager Almadena Chtchelkanova, IIS-1065025, and US Department of Energy grants DE-SC0007444, DE-DC0012495, program manager Lucy Nowell.

References

[BK91] Borenstein J., Koren Y.: The vector field histogram-fast obstacle avoidance for mobile robots. *Robotics and Automation, IEEE Transactions on 7*, 3 (Jun 1991), 278–288. 1, 2

[BKC*13] Borgo R., Kehrer J., Chung D. H. S., Maguire E., Laramee R. S., Hauser H., Ward M., Chen M.: Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications. In *Eurographics 2013 - State of the Art Reports* (2013), Sbert M., Szirmay-Kalos L., (Eds.), The Eurographics Association. 3

[Bou06] Bourke P.: *Converting to and from 6 cubic environment maps and a spherical map*. Tech. rep., May 2006. 3

[BP96] Boring E., Pang A.: Directional flow visualization of vector fields. In *Proceedings of the 7th Conference on Visualization '96* (Los Alamitos, CA, USA, 1996), VIS '96, IEEE Computer Society Press, pp. 389–ff. 2

[BS16] Butkiewicz T., Stevens A.: Effectiveness of structured textures on dynamically changing terrain-like surfaces. *Visualization and Computer Graphics, IEEE Transactions on 22*, 1 (Jan 2016), 926–934. 3

[CM92] Crawfis R., Max N.: Direct volume visualization of three-dimensional vector fields. In *Proceedings of the 1992 Workshop on Volume Visualization* (New York, NY, USA, 1992), VVS '92, ACM, pp. 55–60. 2

[dLvW93] de Leeuw W., van Wijk J.: A probe for local flow field visualization. In *Visualization, 1993. Visualization '93, Proceedings., IEEE Conference on* (Oct 1993), pp. 39–45. 2

[Dov95] Dovey D.: Vector plots for irregular grids. In *Visualization, 1995. Visualization '95. Proceedings., IEEE Conference on* (Oct 1995), pp. 248–253, 459. 2

[DT05] Dalal N., Triggs B.: Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (June 2005), vol. 1, pp. 886–893 vol. 1. 2

[GJL*09] Grundy E., Jones M. W., Laramee R. S., Wilson R. P., Shepard E. L.: Visualisation of sensor data from animal movement. *Computer Graphics Forum 28*, 3 (2009), 815–822. 1

[Gre86] Greene N.: Environment mapping and other applications of world projections. *Computer Graphics and Applications, IEEE 6*, 11 (Nov 1986), 21–29. 1, 2

[HLNW11] Hlawatsch M., Leube P., Nowak W., Weiskopf D.: Flow radar glyphs - static visualization of unsteady flow with uncertainty. *Visualization and Computer Graphics, IEEE Transactions on 17*, 12 (Dec 2011), 1949–1958. 2

[HS08] Hüttig C., Stemmer K.: The spiral grid: A new approach to discretize the sphere and its application to mantle convection. *Geochemistry, Geophysics, Geosystems 9*, 2 (2008). Q02018. 2

[ID08] Ihaddadene N., Djeraba C.: Real-time crowd motion analysis. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* (Dec 2008), pp. 1–4. 2

[JDKW15] Jarema M., Demir I., Kehrer J., Westermann R.: Comparative visual analysis of vector field ensembles. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology (IEEE VAST)* (2015). 1, 2

[JPGJ12] Jiao F., Phillips J., Gur Y., Johnson C.: Uncertainty visualization in hardi based on ensembles of odfs. In *Visualization Symposium (PacificVis), 2012 IEEE Pacific* (Feb 2012), pp. 193–200. 2

[KML99] Kirby R. M., Marmanis H., Laidlaw D. H.: Visualizing multivalued data from 2d incompressible flows using concepts from painting. In *Proceedings of the Conference on Visualization '99: Celebrating Ten Years* (Los Alamitos, CA, USA, 1999), VIS '99, IEEE Computer Society Press, pp. 333–340. 2

[Kuh14] Kuhnert J.: Meshfree numerical schemes for time dependent problems in fluid and continuum mechanics. *Sudarshan, S.: Advances in PDE modeling and computation New Delhi: Ane Books* (2014), 119–136. 1

[Lar03] Laramee R. S.: First: a flexible and interactive resampling tool for CFD simulation data. *Computers & Graphics 27*, 6 (2003), 905–916. 2

[Leo06] Leopardi P.: A partition of the unit sphere into regions of equal area and small diameter. *Electronic Transactions on Numerical Analysis 25* (2006), 309–327. 2

[LPSW96] Lodha S., Pang A., Sheehan R., Wittenbrink C.: Uflow: visualizing uncertainty in fluid flow. In *Visualization '96. Proceedings.* (Oct 1996), pp. 249–254. 2

[PGL*12] Peng Z., Grundy E., Laramee R. S., Chen G., Croft N.: Mesh-driven vector field clustering and visualization: An image-based approach. *IEEE Transactions on Visualization and Computer Graphics 18*, 2 (Feb. 2012), 283–298. 2

[PGN*14] Peng Z., Geng Z., Nicholas M., Laramee R. S., Croft N., Malki R., Masters I., Hansen C.: Visualization of flow past a marine turbine: the information-assisted search for sustainable energy. *Computing and Visualization in Science 16*, 3 (2014), 89–103. 1

[PL08] Peng Z., Laramee R. S.: Vector glyphs for surfaces: A fast and simple glyph placement algorithm for adaptive resolution meshes. In *Proceedings of the Vision, Modeling, and Visualization Conference 2008, VMV 2008, Konstanz, Germany, October 8-10, 2008* (2008), pp. 61–70. 2

[PP99] Pajevic S., Pierpaoli C.: Color schemes to represent the orientation of anisotropic tissues from diffusion tensor data: Application to white matter fiber tract mapping in the human brain. *Magnetic Resonance in Medicine 42*, 3 (1999), 526–540. 3

[PPvA*09] Peeters T., Prckovska V., van Almsick M., Vilanova i Bartroli A., ter Haar Romeny B.: Fast and sleek glyph rendering for interactive hardi data exploration. In *Visualization Symposium, 2009. PacificVis '09. IEEE Pacific* (April 2009), pp. 153–160. 2

[SJP06] Swinbank R., James Purser R.: Fibonacci grids: A novel approach to global modelling. *Quarterly Journal of the Royal Meteorological Society 132*, 619 (2006), 1769–1793. 2

[SSSSW13] Schultz T., Schlaffke L., SchÃŰlkopf B., Schmidt-Wilcke T.: Hifive: A hilbert space embedding of fiber variability estimates for uncertainty modeling and visualization. *Computer Graphics Forum 32*, 3pt1 (2013), 121–130. 2

[Tuc04] Tuch D. S.: Q-ball imaging. *Magnetic Resonance in Medicine 52*, 6 (2004), 1358–1372. 3

[WFH*14] Warren M. S., Friedland A., Holz D. E., Skillman S. W., Sutter P. M., Turk M. J., Wechsler R. H.: Dark sky simulations collaboration, July 2014. 1, 4

[WGH12] Westerteiger R., Gerndt A., Hamann B.: Spherical Terrain Rendering using the hierarchical HEALPix grid. In *Visualization of Large and Unstructured Data Sets: Applications in Geospatial Planning, Modeling and Engineering - Proceedings of IRTG 1131 Workshop 2011* (Dagstuhl, Germany, 2012), vol. 27, pp. 13–23. 2

[WGJL12] Walker J., Geng Z., Jones M., Laramee R. S.: Visualization of Large, Time-Dependent, Abstract Data with Integrated Spherical and Parallel Coordinates. In *EuroVis - Short Papers* (2012), Meyer M., Weinkaufs T., (Eds.), The Eurographics Association. 1

[WPL96] Wittenbrink C., Pang A., Lodha S.: Glyphs for visualizing uncertainty in vector fields. *Visualization and Computer Graphics, IEEE Transactions on 2*, 3 (Sep 1996), 266–279. 2

[ZDG*08] Zuk T., Downton J., Gray D., Carpendale S., Liang J.: Exploration of uncertainty in bidirectional vector fields. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* (2008), vol. 6809. published online. 2

[Zwi95] Zwillinger D.: *CRC Standard Mathematical Tables and Formulae, 31st Edition*. Advances in Applied Mathematics. CRC Press, 1995. 2