Surface Glyphs for Visualizing Multimodal Volume Data

Timo Ropinski[1], Michael Specht[1,2], Jennis Meyer-Spradow[1], Klaus Hinrichs[1], Bernhard Preim[2]

[1]Visualization and Computer Graphics Working Group (VisCG), University of Münster
[2]Department of Simulation and Graphics (ISG), University of Magdeburg
Email: {ropinski, spech_m01, spradow, khh}@math.uni-muenster.de, preim@isg.cs.uni-magdeburg.de

Abstract

In this paper we present concepts for integrating glyphs into volumetric data sets. These concepts have been developed with the goal to make glyph-based visualization of multimodal volumetric data sets more flexible and intuitive. We propose a surface-based glyph placement strategy reducing visual clutter as well as image-space glyph aggregation. Thus the user is not distracted by unwanted clustering, and his focus of attention can rather be guided using appropriate visual appearances. Furthermore, we present techniques to make the setup of glyph-based visualizations more intuitive. These concepts have been integrated into a user interface which supports easy configuration and comparison of different glyph setups. Based on the chosen setup a visual legend is generated automatically to make a step towards quantitative visual analysis. We will discuss the placement strategy as well as the glyph setup process, explain the used rendering techniques and provide application examples of multimodal visualizations using the proposed concepts.

1 Introduction

Multimodal volume visualization has to deal with the proper integration of data obtained from different sources. In the medical domain acquisition of different modalities is about to become a daily routine since modern scanners such as PET/CT, PET/MRT or SPECT/CT can be used to capture multiple registered volume data sets. PET data sets which usually have a lower resolution than CT data sets represent a functional image, e.g., metabolism activity, while CT data sets provide a detailed morphological image. To benefit from both modalities, they have to be visualized simultaneously in an integrated manner. In contrast to these modalities additional data can be derived from a given volume data set and visualized with our technique. For instance, cardiac wall thickness or wall motion can be calculated from time-varying medical data sets. The use of multimodal volumetric data sets is manifold not only in medicine but also in areas like meteorology and seismology. Furthermore, many physical simulations, e.g., fluid dynamics or quality insurance simulations, produce multimodal volumetric data sets.

A common approach to visualize volumetric data sets containing two modalities is to generate a fusion image by blending between these. For instance, modern medical workstations support such a fusion imaging to explore registered PET/CT data sets and allow the user to control the degree of blending by using a slider. This fusion imaging has two drawbacks. First, this form of user control introduces an additional degree of freedom which makes it difficult to find adequate visualization parameters. Second, quantification becomes more difficult since the blending has a major influence on the colors used to represent intensities stored within the data sets, and the colors may interfere with the surface shading. While these issues may be manageable when dealing with only two modalities, the blended fusion image is insufficient if more than two modalities have to be taken into account.

Using glyphs in volume visualization is not new [6, 7, 8]. However, most of the proposed work has rather focussed on ways glyphs can be used to visualize information than on how to flexibly setup this information and make it quantifiable. The contribution of this paper is to make glyphs more usable as a tool for multimodal volume visualization. Therefore we propose a surface-based glyph placing strategy which decreases unwanted glyph clustering and reduces clutter by minimizing glyph occlusion. Furthermore, we will introduce a user interface, which allows an easy configuration of a glyph setup, i.e., the definition how properties are represented by glyphs and how the glyphs are placed. Based on this setup we describe how to generate a visual legend which is a necessary step towards a quantitative visual analysis of glyph visualizations.

The paper is structured as follows. In the next section related work is discussed. The so called surface glyphs and details regarding their placement are discussed in Section 3. In order to support a flexible and intuitive setup of a glyph-based visualization we propose some techniques accompanied with appropriate user interface concepts in Section 4. The used rendering technique is briefly described in Section 5, while application examples are given in Section 6. The paper concludes in Section 7.

2 Related Work

This section briefly discusses some work related to the topic of this paper, but does not intend to give a complete overview of glyph visualization techniques used in scientific as well as information visualization. Moreover, we focus on those techniques which are related to our approach for visualizing volumetric multimodal data sets.

Ward [18] states that glyph-based visualization is a powerful method for providing multimodal information, by adding iconic glyphs to a scene in order to display various variables through various properties such as shape or color. In his work he describes and classifies different glyph placing strategies and proposes rules for their usage in the context of information visualization.

In the field of scientific visualization diffusion tensor imaging (DTI) is probably the domain where the usage of glyphs has been most intensively investigated [10, 17, 8]. Most of the work published in this area focuses rather on choosing an appropriate glyph shape in order to transmit information [5] than on positioning of the glyphs. For instance, in [8] superquadrics are used to convey the principal eigenvectors of a diffusion tensor in order to depict the microstructure of white-matter tissue of the human brain. The distinct glyphs are placed in a regular grid and controlled by a fractional anisotropy threshold in order to minimize visual clutter. Jankun-Kelly and Mehta [6] have used superquadric ellipsoid glyphs to visualize traceless tensor data.

To achieve a beneficial glyph visualization, not only the shape of the used glyphs but also their placement is essential. In [10] a stochastically jittered placing of glyphs is described with the goal to eliminate the possibly distracting effects of a grid placement. In 2006 Kindlmann and Westin [7] have proposed a glyph packing strategy allowing a texture like appearance of a glyph aggregation. In contrast to the approach presented in this paper only one modality, i.e., DTI data, is used, while our glyph placing strategy is based on the fusion of multiple modalities. In [13] spherical glyphs are exploited to visualize cardiac MR data in order to allow an exploration of the structure as well as the function of the myocard.

Besides in the medical domain glyphs are also used in other areas dealing with multimodal data. Nayak et al. [12] have used glyphs in order to visualize seismic data representing the measured, time-dependent, 3D wave field of an earthquake recorded by a seismic network. Reina and Ertl [14] have proposed a technique for depicting molecular dynamics by means of hardware-accelerated glyph rendering, and Saunders et al. [15] have proposed the use of circular glyphs for visualizing nano particles in formation.

Besides using glyphs there are other strategies to visualize multi-field volume data. Recently Akiba et al. [1] have presented a novel user interface concept to support the identification of correlation. Their technique exploits linked views as well as parallel coordinates and is demonstrated by visualizing the hurricane Isabel data set. Similar to the concepts presented in [4] their approach supports brushing, to allow the user to formulate the current interest. Blaas et al. have introduced a technique which also exploits linked views [3]. Besides a physical view they use a feature-space view to visualize multi-field data.

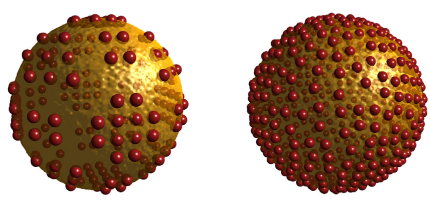
3 Surface Glyphs

Glyph placement is crucial in order to achieve meaningful glyph-enhanced visualizations. Glyphs that occlude each other or occlude big portions of the volume can easily counteract the information increase intended by the use of glyphs. Therefore it must be ensured that glyph occlusion is minimized, and that not too many glyphs are placed.

A basic approach for placing glyphs is to superimpose a regular three-dimensional grid on the volumetric data set and to place the glyphs at the grid points (see Figure 1(a)). However, this method has two drawbacks. First, the introduced grid structures do not necessarily match the structure of the underlying data set, e.g., in the example the underlying data set shows a round, smooth sphere. Second, placing glyphs in a regular grid may lead to the false impression of a *glyph aggregation* resulting from the glyph positions in image space. These aggregations attract the user's attention, although the user's focus could be better guided by the visual appearance of the glyphs, i.e., by the glyph properties and user-defined glyph property mapping functions (see Section 6).

To eliminate these drawbacks, we propose a surface-based glyph placement strategy. It achieves a feasible glyph distribution that fulfills both of the requirements stated above by placing glyphs on isosurfaces of a volumetric data set. A straightforward way of defining such an isosurface is to specify an isovalue as it is done in isosurface rendering. The effect of isosurface placement is that glyphs are placed at points that are exactly located on an isosurface (see Figure 1(b)). Isosurface placement leads to fewer glyph occlusions and also avoids misleading glyph aggregation in image space.

A possible approach to realize isosurface glyph placement is to calculate a polygonal mesh using the marching cubes algorithm [11], and then to randomly choose polygons and place glyphs on them.



(a) Regular grid placement. (b) Isosurface placement.

Figure 1: Comparison of the regular grid and isosurface glyph placement methods. The result of the isosurface placement is visually more pleasing because the glyphs are evenly distributed on the surface and occlusion is minimized.

Uniform distribution could be achieved by calculating the surface area of each polygon and taking polygon surface area distribution into account when polygons are chosen randomly. Apparently, the contiguous surface resulting from the marching cubes algorithm is not necessary for isosurface glyph placement. Distinct glyphs should be placed at certain locations near an or even exactly on an isosurface, but it does not matter whether two locations of interest are directly connected by an isosurface. This makes the marching cubes approach inappropriate, especially if one considers the complexity and ambiguous special cases.

Therefore we propose a simpler method, which proceeds in two steps. The first step resembles closely the first step of the marching cubes algorithm. A three-dimensional boolean array having in each direction the size of the volumetric data source minus one is allocated. Each boolean array element corresponds to a cell which is determined by an octuple of adjacent voxels from the data set. In the first step, the *marking step*, the array is traversed, and for each cell the data source is evaluated at each of its eight vertices. A cell is marked if the values indicate that the isosurface passes through it, i.e., if the eight values are not all greater than or not all less than the specified isovalue. In the second step, the *placement step*, marked cells are repeatedly extracted from the array until none are left. A glyph is placed at every marked cell and adjusted inside that cell in such a way that it is located exactly on the specified isosurface while remaining as close as possible to the center of the cell. This is achieved by subdividing the cell into eight sub-cells and querying the data source at the eight vertices of each subcell. Among all sub-cells the isosurface is determined to run through, the sub-cell with the smallest distance to the root cell's center is chosen and recursively subdivided. Finally, all cells within a user-specified world space distance from the current cell are cleared in order to create a certain amount of empty space around each glyph.

Although the images resulting from the isosurface placement method are good for simple, generic data sets, a problem arises when real-world data is used. As depicted in Figure 2(a), it is possible that glyphs seem to be placed below the surface (this effect can be seen in the region of the ear) or placed not at all because isosurfaces may be very close and run parallel to each other, so that glyphs are distributed on both surfaces. Figure 2(b) reveals that the missing and misplaced glyphs have been placed on the inside surface of the skull.

In order to provide a solution for misplaced glyphs, the isosurface placement method has been restricted to visible isosurfaces only. Visible isosurface placement assumes that isosurfaces that are not visible to the viewer can not be of any interest and hence, they should be ignored in the cell marking step. This additional condition for the marking step is implemented by casting axis-aligned rays into the volume data set, starting at each outer cell and directed into the data set. When one cell has been marked by a ray, this ray is terminated so that in the end only cells visible from the outside have been marked. With this restriction we resolve ambiguities as those shown in Figure 2.

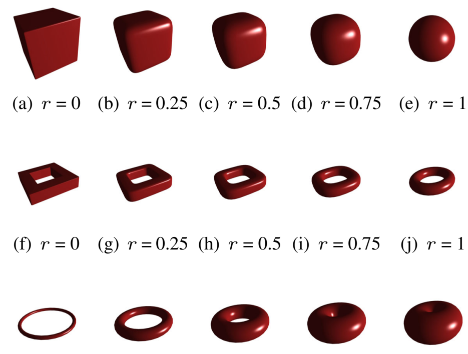
4 Glyph Visualization Setup

We have identified the following work flow for creating glyph-enhanced volume visualizations. Initially, after the user has loaded the contributing data sets the glyph modeling has to be carried out. Therefore the user selects a glyph prototype and specifies a mapping function which maps the information contained in the volume data sets to glyph properties, e.g., color or size. Finally the desired glyph placement method is chosen and the glyphs are arranged within a volume data set to represent information located at their position. Our proposed user interface assisting during the glyph setup consists of two parts, one window for glyph modeling and an overlayed graphical legend supporting evaluation of glyph-enhanced visualizations.

4.1 Glyph Shapes

Each glyph prototype is characterized by a set of properties through which information can be visualized. Depending on the desired visualization and the glyph properties, the suitability of glyph prototypes varies. Therefore the user has to choose a suitable glyph prototype for a certain task. The basic properties which are shared by all glyph prototypes are color, opacity and size, while more sophisticated glyph prototypes offer further possibilities to convey information by additional glyph properties.

The glyph prototypes described in this paper are based on the superquadric shapes presented



(a) $r = 0$ (b) $r = 0.25$ (c) $r = 0.5$ (d) $r = 0.75$ (e) $r = 1$

(f) $r = 0$ (g) $r = 0.25$ (h) $r = 0.5$ (i) $r = 0.75$ (j) $r = 1$

(k) $t = 0.2$ (l) $t = 0.4$ (m) $t = 0.6$ (n) $t = 0.8$ (o) $t = 1$

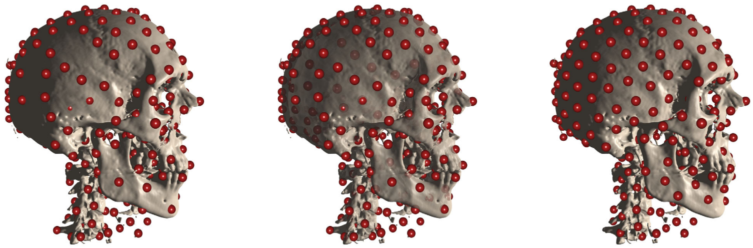
Figure 3: The supersphere and supertorus prototypes with varying parameters for roundness r (upper two rows), and the supertorus prototype with varying parameters for thickness t (lower row).

in [2]. In contrast to cuboid or ellipsoid glyphs, superquadrics have the advantage that a multitude of parameters can be mapped unambiguously. For instance, when using ellipsoids or cuboids an awkward viewing direction may result in visual ambiguity, i.e., different shapes are not distinguishable after projected onto the view plane [8]. For the sake of simplicity, some parameters of the original superquadric shapes have been made constant or combined in order to obtain intuitive shapes that can be interpreted easily: the supersphere and the supertorus. In contrast to the original superquadric ellipsoid which takes three radii as parameters, the supersphere has a fixed radius of 1. Glyph properties defined by the supersphere prototype are the scalar parameters α and β, which are derivations from the original superquadric ellipsoid α' and β'. The purpose of this simplification is a more easy setup as well as interpretation process. Thus it is more intuitive to define the roundness of the surface because their default value is 0 and the perceived change in roundness and edge sharpness resulting from adjusting α and β is linear. The conversion from supersphere α and β to the original α' and β' is given by:

$$\alpha' = 2^{\alpha} \quad (1)$$

$$\beta' = 2^{\beta} \quad (2)$$

In the original equations, the base shape (plain



(a) (b) (c)

Figure 2: Glyphs are distributed on front- and back-facing surfaces (a), semi-transparent isosurfaces reveal the hidden glyphs (b). Application of visible isosurface placement (c).

sphere or torus) results from $\alpha' = \beta' = 1$. Less roundness can be achieved with $0 > \alpha' < 1$ and $0 > \beta' < 1$, sharper edges can be achieved with $\alpha' > 1$ and $\beta' > 1$. With the conversion applied, the base shape results from $\alpha = \beta = 0$, less roundness is achieved with $\alpha < 0$ and $\beta < 0$, sharper edges are achieved with $\alpha > 0$ and $\beta > 0$, and the change in sharpness or roundness is linear and thus more suitable for user interaction. However, these prototypes are still rather complex because the interpretation of the α and β values is not very intuitive. For example, two supertori perceived as 'not round' can have different parameters. In order to resolve such ambiguities, simplified versions of both supersphere and supertorus prototypes have been created in which the α and β parameters are combined into a parameter r that represents the roundness of the object expressed by a value in the range from 0 (angular) to 1 (round). The conversion from r to α and β is defined by the equation

$$\alpha = \beta = 5 \cdot r - 5. \quad (3)$$

This results in a value between 0 and 5 serving as good values for round resp. non-round superquadrics (see Figure 3).

The supertorus prototype is defined by an additional scalar t that represents the thickness of the torus. In comparison to the original superquadric shapes, the supersphere and supertorus prototypes are easier to interpret because an estimation of the roundness r seems rather trivial when compared to an estimation of the parameters α and β used in the original equations. Even for users with no mathematical background it should be easy to interpret the roundness and thickness values.

The complex glyph modeling task can be done with the help of the user interface depicted in Figure 5, whereas the interface elements are arranged according to the work flow of the glyph modeling process. After all data sets have been loaded, a glyph prototype has to be chosen from a list of available prototypes. The currently chosen glyph prototype is shown in the prototype window (see upper overlay in Figure 5). To save screen space, glyph properties of interest can be expanded or hidden, e.g., in Figure 5, the roundness and thickness properties are currently expanded.

4.2 Glyph Property Mapping

For specifying the property mapping function, which maps input values from scalar data sources to scalar glyph property values, the user can control a set of mapping keys. Each mapping key defines a pair of source and destination values. The specification of the mapping function is similar to specifying a transfer function. An example is shown in Figure 4, where two keys are used to define the mapping behavior. In order to provide the possibility to put emphasis on certain glyphs, steps can be introduced in a mapping function by splitting keys and defining different destination values for points to the left and to the right of the key (see Figure 4). If the mapping function is evaluated for a source value other than those defined by the keys, linear interpolation between the keys to the left and the right of the queried location is used. If the mapping function is queried for locations that do not lie between two mapping keys, the value of the nearest mapping key is returned.

In the mapping function window (see lower overlay in Figure 5), a mapping function can be defined by modifying mapping keys within the canvas spanning the complete range of possible input and output values. In addition to the mapping function, a scalar data source must be chosen that should be linked to the glyph property and that the mapping function should be applied to. The mapping canvas supports the user by providing the possibility to display the histogram of the current data source. After the glyph properties have been adjusted, a glyph placement method is chosen along with a data source the glyph placement is applied to. When the glyph prototype and the placement method have been set up, the glyph modeling is finished and the glyphs can be rendered.

4.3 Glyph Legend

To allow the interpretation of a glyph-enhanced visualization, it is necessary that the glyph mapping process is transparent, such that the user can derive the source information from a glyph's representation. Therefore we propose a glyph legend which represents the mapping function graphically. Since only those glyph properties that are dependent on data sources actually convey information, only these properties are of interest within a glyph legend. Thus a legend can be constructed from a number of rows each depicting how a property of interest relates to its underlying data source. If multiple properties depend on the same data source, these properties can be combined into a single row because for each of these properties, the data source is queried at exactly the same location and thus returns exactly the same value.

For generating the glyph legend, all data sources that are linked through a mapping function are taken into account. Thus, for each data source involved, a row is added to the legend that indicates how the

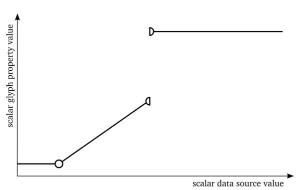


Figure 4: An example mapping function with two mapping keys. The second key is split.

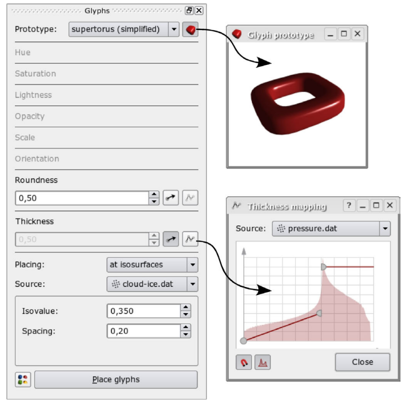


Figure 5: The glyph modeling user interface with the glyph prototype window and the graphical glyph mapping function editor displaying the histogram of the current scalar data source.

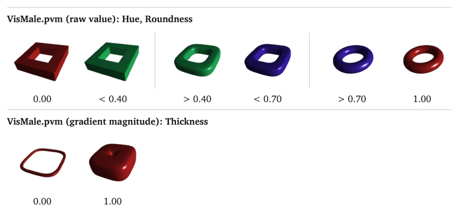


Figure 6: In the minimal glyph legend only key glyphs are displayed.

data source influences the visual appearance of the glyph. All glyph properties that are connected to the current row's data source through the mapping function are determined and printed in the row caption to clarify which properties are conveyed by the glyph icons. Furthermore, for all mapping keys an icon of the glyph prototype is added to the row, showing the glyph prototype with the resulting representation inherently defined by the mapping function. In the cases of split mapping keys, two glyph icons representing both left and right mapping key destination values are rendered and separated by a vertical line to emphasize the fact that a split mapping key is displayed. A legend created in such a way can be seen in Figure 6.

A more comprehensive legend can be created by inserting supplementary keys in cases where the differences between two consecutive mapping keys'

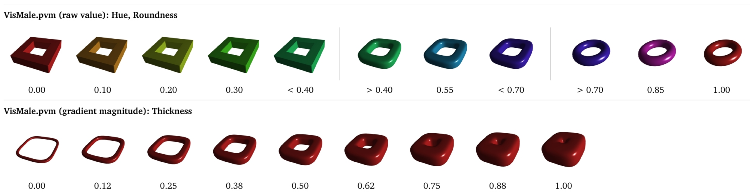


Figure 7: In the the extended glyph legend, glyphs are inserted at certain locations in order to bridge large differences.

destination values are large. A possible approach to extend the legend is to find pairs of consecutive mapping keys for which the change of at least one property exceeds a certain threshold and to add the appropriate glyph representations to the legend. Such an extended legend is shown in Figure 7.

5 Rendering

To allow simultaneous display of volumetric data and polygonal glyphs, we have extended the GPU-based ray casting technique [9], which is solely applicable to render volumetric data sets. The integration of opaque glyphs is quite easy. In order to combine opaque geometry with GPU-based ray casting, it is sufficient to modify the end points of each ray by drawing the additional polygons on top of the proxy geometry's back faces (see Figure 8). Thus each ray terminates as soon as it hits a polygon. Glyphs can be integrated by initially rendering them to the background and afterwards blending the ray-casting results using the modified exit parameters shown in Figure 8(c).

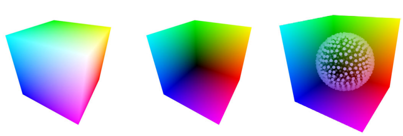


Figure 8: The entry and exit parameters for GPU-based volume ray-casting (left,middle). The exit parameters are modified in order to integrate opaque glyphs positioned on a sphere (right).

6 Application Examples

This section discusses some examples of the described glyph-enhanced volume visualization applied to the NCAT PET/CT data set [16] as well as the hurricane Isabel data set.

In contrast to Computed Tomography which reveals the inner structure of a subject, Positron Emission Tomography is used to get information about metabolic activity inside a living subject. Because PET images are of much lower resolution than CT data sets, it is difficult to tell where points in the image are located within the subject because the morphological context as provided by Computed Tomography is missing. To avoid such problems, a combined CT/PET scan can be performed. The result of such a scan are a PET data set and a CT data set. The latter provides contextual information and is registered with the PET data.

A method of displaying CT and PET data simultaneously is PET/CT fusion imaging, in which a color gradient is applied to the PET image and subsequently both images are combined into a single fusion image. However, this fusion makes quantization difficult, since the PET color mapping is influenced by the color of the blended CT data.

The visualizations presented in Figure 9(a) and in Figure 9(b) are alternatives to common PET/CT fusion imaging. A volume rendering of the CT data provides the context for the PET data which is conveyed by glyphs. In order to demonstrate that in glyph-enhanced volume visualization the glyphs are completely independent of the type of volume rendering applied to the context data, both direct volume rendering (see Figure 9(a)) and X-ray simulation (see Figure 9(b)) are shown. In both examples, different attempts to direct the viewer's attention to important regions of the volume are demonstrated. In Figure 9(a), the supertorus thickness is used to highlight the lesion on the left ventricle evident from the PET data. The thickness is mapped in a way that the user's attention is directed towards regions of unusually low metabolic activity, which are shown by using thicker supertori. Furthermore, the glyph orientation is adjusted to match the normal vectors of the heart surface, resulting in minimal occlusion and the effect that the glyphs look like glued to the surface. In Figure 9(b) regions of high PET activity are visualized in an unobtrusive manner in order to highlight regions of low PET activity.

The previous example could have been visualized to a certain extent by using non-glyph techniques. For instance a simple color coding could have been used in order to visualize the PET activity. However, in order to make this visualization quantifiable, the shading should not influence the shown color. Therefore the data set would have been rendered without shading, which would destroy the shape through shading cue. Another possibility would be using stippling. The CT data set could be rendered using regular phong shading and the PET intensity could be depicted by superimposing a stippling pattern, which intensity is altered based on the PET intensity. While this approach allows an integration of the PET information into a CT visualization without loosing the shape from shading, it is limited to the usage of two modalities.

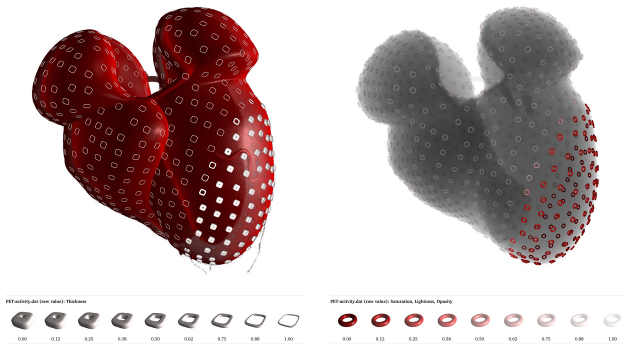
The hurricane Isabel data used for the visualization presented in Figure 10 consists of many modalities including cloud water, cloud ice, graupel, rain, snow, vapor, pressure, temperature and wind direction. The goal is to visualize some of these variables simultaneously in order to allow visual exploration of the value distribution and to identify correlations. Two of these variables plus the sum of three further variables are depicted by the glyphs in the image. Temperature is depicted by hue in a range from blue (cold) to red (warm). Air pressure is depicted by the supertorus thickness, a combination that allows for intuitive interpretation because of the analogy to a bicycle tire. The amounts of graupel, rain and snow are accumulated to the total amount of precipitation and depicted by the roundness of the glyphs.

Similar to applying color mapping, the glyph-based visualization allows to get a quick overview of the value distribution. For instance it can be seen that the temperature is highest within the eye of the storm decreasing with increasing distance to it. Additionally we can see that the pressure is very low inside the eye and increasing with increasing distance to the eye. Since we are using supertorus glyphs, besides the thickness we can also use the roundness to represent a variable. With the precipitation mapped to roundness, it can be seen that it is highest in the vicinity of the eye of the storm and it becomes clear that it is not uniformly decreasing with the distance to the eye. Moreover the precipitation is highest behind the eye of the storm (the direction of movement is towards the upper left).

Although this example shows the potential of glyph-based visualizations it also demonstrates the limits when applying it to multimodal data having many variables. The remaining glyph properties that could be used to depict more scalar variables are saturation, lightness, opacity and scale. Saturation cannot be used to convey detailed information if hue is used to depict other data at the same time, because when the saturation value is approaching 0, correct interpretation of the hue value becomes increasingly difficult. When lightness is used, the interpretation of the hue value becomes difficult when the lightness value approaches 0 or 1. These effects lead to the conclusion that in most cases, only one data source can be depicted by the color properties, although simultaneous display of three completely unrelated variables seems possible. Furthermore, size is potentially inappropriate for conveying data as well because the data-determined size of a glyph conflicts with its size due to perspective projection. Additionally, shape perception becomes very difficult when glyphs are small.

Besides the glyph property mapping, the surface used for glyph orientation has to be defined. When having modalities which provide contextual information, e.g. a CT scan, this definition is quite easy. However, in the general case this task needs more attention. Sometimes it might even be desirable to have a dynamic surface, i.e., a 2D slice moving through the data set.

In both presented examples a non-continuous mapping can be used to further emphasize differences, e.g., when reaching a certain threshold. In the first example this can be used to show only glyphs representing abnormal PET intensities, while the one representing normal ones can be omitted by assigning transparency or minimal size.



(a) PET/CT DVR. (b) PET/CT x-ray.

Figure 9: Cardiac wall motion and activity, derived from PET/CT data, depicted with different glyph setups.

7 Conclusion and Future Work

In this paper we have presented concepts for easy setup and interpretation of glyph-based visualizations. We have introduced modifications of superquadric glyphs and have discussed a surface-based glyph placement strategy which binds glyphs to a surface of interest. To make a step towards a quantifiable glyph-based visualization, we have introduced a visual glyph legend.

In the future more improvements for generating glyph configurations could be considered. An important outcome of this development could be design guidelines specifying which glyph properties are best suited for depicting certain information. In some cases also non-symmetric glyphs could be helpful, e.g., for visualizing velocity or a direction of movement. In addition, during glyph modeling the user should be notified about glyph properties potentially shadowing each other.

Acknowledgements

This work was partly supported by grants from the Deutsche Forschungsgemeinschaft (DFG), SFB 656 MoBil Münster, Germany (project Z1).

References

[1] Hiroshi Akiba and Kwan-Liu Ma. A tri-space visualization interface for analyzing time-varying multivariate volume data. In *Proceedings of Eurographics/IEEE VGTC Symposium on Visualization*, May 2007.

[2] Alan H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, January 1981.

[3] Jorik Blaas, Charl P. Botha, and Frits H. Post. Interactive visualization of multi-field medical data using linked physical and feature-space views. In *Eurovis 2007, proceedings of the joint Eurographics / IEEE VGTC Symposium on Visualization*, pages 123–130, 2007.

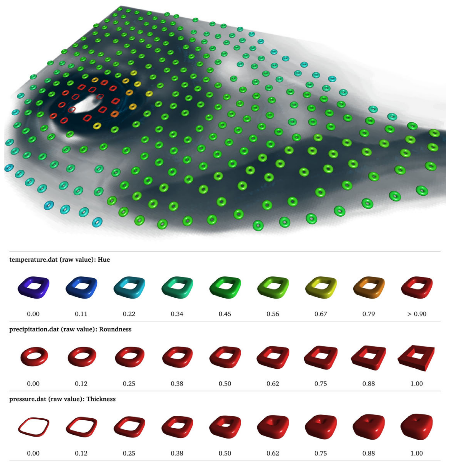
[4] Helmut Doleisch, Philipp Muigg, and Helwig Hauser. Interactive visual analysis of hurricane isabel with simvis. In *IEEE Visualization Contest*, 2004.

[5] D. S. Ebert, R. M. Rohrer, C. D. Shaw, P. Panda, J. M. Kukla, and D. A. Roberts. Procedural shape generation for multi-dimensional data visualization. In *Data Visualization '99*, pages 3–12. Springer-Verlag Wien, 1999.

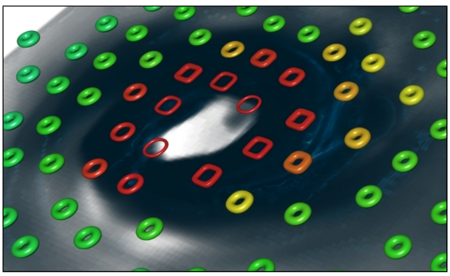
[6] T.J. Jankun-Kelly and Ketan Mehta. Superellipsoid-based, real symmetric traceless tensor glyphs motivated by nematic liquid crystal alignment visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1197–1204, 2006.

[7] Gordon Kindlmann and Carl-Fredrik Westin. Diffusion tensor visualization with glyph packing. *IEEE Visualization 2006)*, 12(5):1329–1135, September-October 2006.

[8] Gordon L. Kindlmann. Superquadric tensor



(a) Glyph visualization of the hurricane Isabel data set in conjunction with a legend.



(b) A detailed view of the eye of the hurricane.

Figure 10: Glyphs are used to visualize temperature, precipitation and air pressure. In this example the benefits of the supertorus glyph become present, since the three modalities are mapped to the color, thickness and roundness parameters.

glyphs. In *Joint Eurographics - IEEE TCVG Symposium on Visualization*, pages 147–154, 2004.

[9] J. Krüger and R. Westermann. Acceleration techniques for GPU-based volume rendering. In *IEEE Visualization 2003*, pages 38–43, 2003.

[10] David H. Laidlaw, Eric T. Ahrens, David Kremers, Matthew J. Avalos, Russell E. Jacobs, and Carol Readhead. Visualizing diffusion tensor images of the mouse spinal cord. In *VIS '98: Proceedings of the conference on Visualization '98*, pages 127–134. IEEE Computer Society Press, 1998.

[11] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, New York, NY, USA, 1987. ACM Press.

[12] A. M. Nayak, K. Lindquist, D. Kilb, R. Newman, F. Vernon, J. Leigh, A. Johnson, and L. Renambot. Using 3D Glyph Visualization to Explore Real-time Seismic Data on Immersive and High-resolution Display Systems. *AGU Fall Meeting Abstracts*, pages C1208+, December 2003.

[13] Lydia Paasche, Steffen Oeltze, Frank Grothues, Anja Hennemuth, Caroline Kühnel, and Bernhard Preim. Integrierte visualisierung kardialer mr-daten zur beurteilung von funktion, perfusion und vitalität des myokards. In *Bildverarbeitung fr die Medizin*, pages 212–216, 2007.

[14] G. Reina and T. Ertl. Hardware-Accelerated Glyphs for Mono- and Dipoles in Molecular Dynamics Visualization. In K. W. Brodlie and D. J. Duke and K. I. Joy, editor, *Joint Eurographics - IEEE TCVG Symposium on Visualization*, pages 177–182, 2005.

[15] Patrick Coleman Saunders, Victoria Interrante, and S.C. Garrick. Pointillist and glyph-based visualization of nanoparticles in formation. In *Joint Eurographics - IEEE TCVG Symposium on Visualization*, pages 169–176, 2005.

[16] W.P. Segars, B.M.W. Tsui, A.J. Da Silva, and L. Shao. Ct-pet image fusion using the 4d ncat phantom with the purpose of attenuation correction. In *IEEE Nuclear Science Symposium Conference Record 2002*, pages 1775–1779, 2003.

[17] A Sigfridsson, T Ebbers, Heiberg, and L Wigström. Tensor field visualization using adaptive filtering of noise fields combined with glyph rendering. In *Proceedings IEEE Visualization 2002*, pages 371–378, Boston, Massachusetts, 2002.

[18] Matthew O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3/4):194–210, 2002.