FURY: SDF-based uncertainty representation for dMRI glyphs

About Me

Tania Valentina Castillo Delgado
Universidad Nacional de Colombia, B.S. Computer Science, EGD: September 2023
Timezone: UTC-5
Github username: tvcastillod
LinkedIn: https://www.linkedin.com/in/tania-valentina-castillo-delgado/

Code contribution

Tutorial on making a primitive using polygons and SDF

https://github.com/fury-gl/fury/pull/620

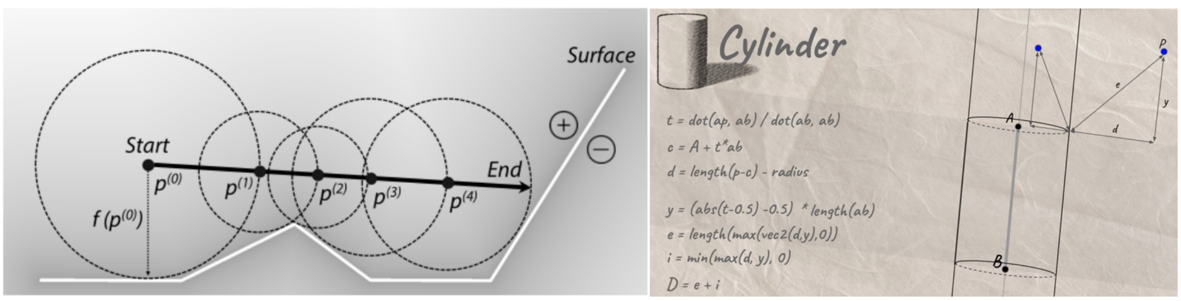

Figure 1: Visualization of ray marching algorithm and cylinder SDF definition.

Ray marching is a rendering technique where you step along a ray to find intersections with solid geometry defined by SDFs. A ray is defined by the parametric equation $r(t) = o + td$ where o is the ray origin d the ray direction d and t is the distance traversed along the ray. Then, with the SDF function F, we have $d = F(r(t))$, which gives the next step along the ray $(t = t + d)$, and the condition evaluated at each step of the algorithm to check if the distance from the current point is close enough to the surface, i.e. if $d < \varepsilon$ a ray intersection with the implicit surface is detected. In the case of the cylinder with radius r and height h, the SDF is defined by analyzing the point-to-set distance from a point located in different regions as shown in figure 1. The quality improvement using this technique is can be seen in figure 2.

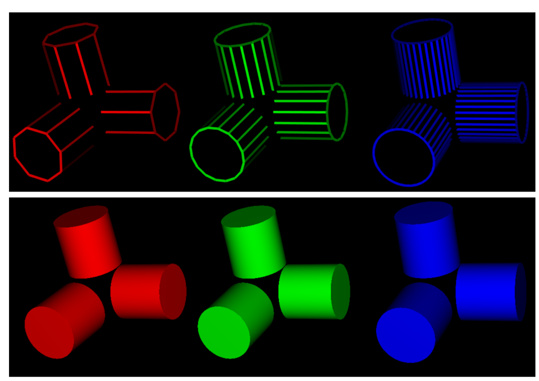

Figure 2: glyph cylinders rendered using traditional polygon discretization and SDFs

Cylindrical billboard implementation

https://github.com/fury-gl/fury/pull/677

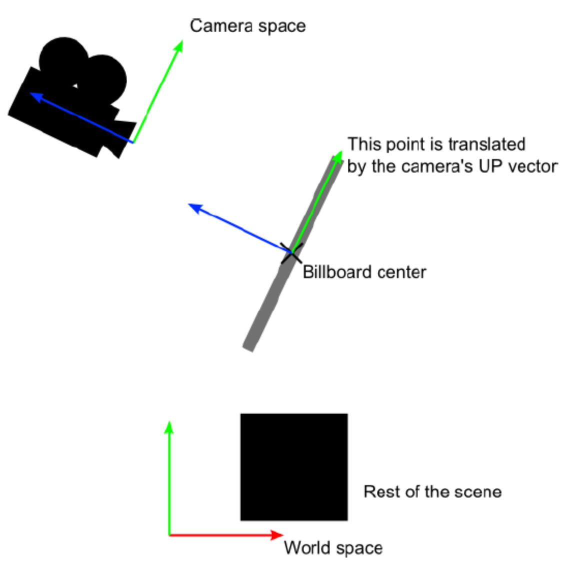

Figure 3: Billboard squeme

The vertices of the billboard are defined by getting cameras up and right vectors in world space. These vectors are obtained from the inverse of M_1, the top 3×3 submatrix of modelview matrix, that contains scaling and rotation operations. Since M_1 should be orthogonal, the inverse is equal to the transpose.

$$ModelView = \begin{pmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{pmatrix},$$

$$M_1^{-1} = \begin{pmatrix} {\color{red}m_0} & {\color{green}m_1} & {\color{blue}m_2} \\ {\color{red}m_4} & {\color{green}m_5} & {\color{blue}m_6} \\ {\color{red}m_8} & {\color{green}m_9} & {\color{blue}m_{10}} \end{pmatrix}$$

With the values of right and up vectors, each vertex of the billboard is defined with respect to the vectors associated with the camera as $vertex = center \pm (right \pm up) * size$

While a spherical billboard faces the camera no matter what axis is rotated around, cylindrical billboards are rotated only when the camera revolves around the y-axis, and can be achieved by setting the up vector to $(0,\ 1,\ 0)$ so the billboard won't change with the camera when moving up or down, i.e., $right = (m_0,\ m_4,\ m_8)$, $up = (0,\ 1,\ 0)$. Equivalently an implementation around the x-axis is defined with $right = (1,\ 0,\ 0)$, $up = (m_1,\ m_5,\ m_9)$.

Project Information

Sub-org name: FURY Free Unified Rendering in Python

Project Abstract

Diffusion Magnetic Resonance Imaging (dMRI) is a non-invasive imaging technique used by neuroscientists to measure the diffusion of water molecules in biological tissue. The directional information is reconstructed using either a Diffusion Tensor Imaging (DTI) or High Angular Resolution Diffusion Imaging (HARDI) based model, which is graphically represented as tensors and Orientation Distribution Functions (ODF). Traditional rendering engines discretize Tensor and ODF surfaces using triangles or quadrilateral polygons, making their visual quality depending on the number of polygons used to build the 3D mesh which might compromise real-time display performance. This project proposes a methodological approach to further improve the visualization of DTI tensors and HARDI ODFs glyphs by using well-established techniques in the field of computer graphics such as geometry amplification, billboarding, signed distance functions (SDFs), and ray marching.

Detailed description

The project consists of three components:

Geometry amplification for billboard creation

Although billboarding provides significant improvements by simplifying complex geometry, the creation of the billboards' quads can be accelerated using the inverse process known as geometry amplification. The current implementation of billboards uses `prim_square()` to create a quad(4 vertices) and then uses vertex shader to transform them using the camera's up and right vectors. In this first preliminary part, a parallelized version of computer-generated billboards using geometry shaders will be implemented, since geometry shaders handle the processing of primitives where new primitives can be created from a single one, hence we can create the billboards from a single point. This is the base on which the SDF glyphs will be displayed.

SDF ray marching for Tensor and ODF glyphs visualization

Ray marching is a robust technique for ray tracing implicit surfaces, that consists of stepping along a ray to find intersections with solid geometry defined implicitly with SDFs. The current implementation with `tensor_slicer`, and `odf_slicer` requires a sphere with a specific number of vertices to be deformed based on the model used, to get a higher resolution a sphere with more vertices is needed. Because the raymarching technique does not use polygonal meshes it is possible to define perfectly smooth surfaces and still obtain a fast rendering.

	repulsion10	repulsion200	repulsion724	SDF(Ours)
slice	65	64	62	63
ROI	41	26	*	38
whole brain	*	*	*	10

* Cannot be processed.

Table 1: average FPS from the display of tensor data of slice $30 \times 30(900)$, ROI $30 \times 30 \times 30(27000)$, and Whole Brain $71 \times 88 \times 62(387376)$.

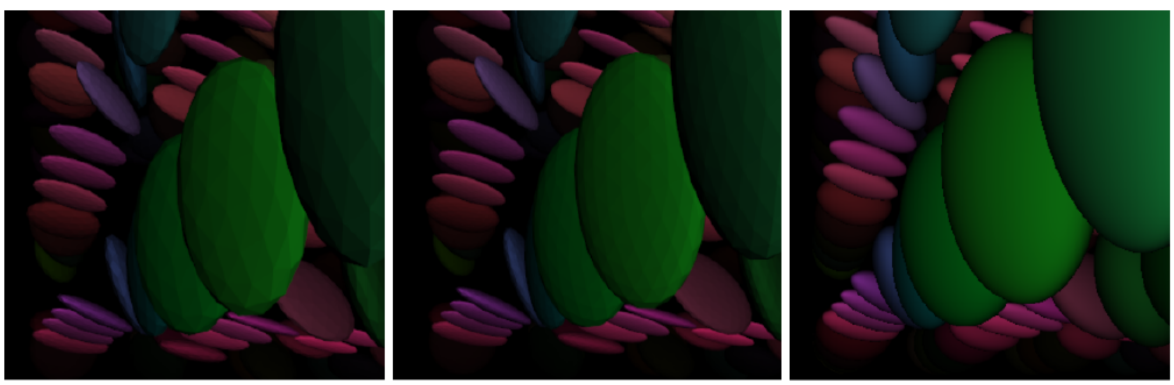

Figure 4: Visualization of tensor ellipsoid made with `tensor_slicer` with sphere `'repulsion100'`, `'repulsion200'` and SDF implementation applied to a box

A first approach for tensor glyph generation has been made, using raymarching and SDF applied to a cube instead of a billboard. Although a first result does not yet show an improvement in performance, it does show a better quality in the displayed glyphs, and support the display of a large amount of data. We expect to obtain an improvement in performance, i.e., higher frame rate and less memory usage on dMRI glyphs creation using billboards and looking for ways to optimize the naive raymarching algorithm and the definition of SDFs. If a good result is obtained, a better approach to DTI visualization is suggested by [Kin04] with superquadrics implementation, and since the implicit function is provided, a first implementation is feasible.

Uncertainty representation

The DTI visualization pipeline is fairly complex and because a considerable number of measurements, parameters, and model assumptions are involved, a level of uncertainty arises, which, if visualized, helps to assess the accuracy of the model, contributing to the analyst's work. This measure is not currently implemented, so given the proposed resource optimization, we expect to further expand the field by adding uncertainty visualization of the DTI model. Based on a preliminary search of Siddiqui's work in the visualization of uncertainty associated with DTI and ODF reconstruction models, Matrix Perturbation Analysis (MPA) proposed by Basser [Bas97] is considered for DTI uncertainty visualization. If ODF implementation is completed with good results, semitransparent glyphs representing the mean directional information proposed by Tournier [TCGC04] is considered for ODF uncertainty.

Weekly timeline

Pre GSOC

Familiarize myself fully with the code structure of FURY, documentation, and test system used. Start working on SDF ellipsoid implementation on boxes, as a preamble to the use of actors in fury and a better understanding of raymarching SDF implementation. Search for documentation about dMRI focused on DTI and ODF models to understand better the data I need to work with for SDF definition and ray marching implementation.

Community Bonding

Familiarize with the code base on which I will be working, and understand the current implementation of relevant actors such as billboard, `tensor_slicer`, and `odf_slicer`. Discuss with the team about technical details about the different parts of the project and discuss further ideas to contribute to the main project task.

Week 1 Implement a parallelized version of computer-generated billboards using geometry shaders for amplification and make the corresponding tests.

Week 2-3 Adapt the current way of raymarching SDFs in a box, to billboards for optimization in performance.

Week 4 Understand the maths of the proposed uncertainty model and begin with a basic implementation of the primitives used for its visual representation.

Week 5-6 Carry out the formal implementation of the uncertainty model as an independent entity, make the respective tests, and link it to the tensor ellipsoid.

Week 6 Midterm Make a review of the progress made in order to correct and, above all, improve the implementations made. Start reviewing spherical harmonics for the next glyph implementation.

Week 7-8 Develop a code implementation of spherical harmonics from scratch to understand how the function and parameters defined, affect the surface of a sphere.

Week 9-10 Comprehend current odf_slicer actor implementation to adjust the previous set-up to work with the same data.

Week 11 Focus on reviewing the overall progress by completing unfinished tasks, making requested changes, fixing bugs, prioritizing the progress, and completion of the most important contributions.

Week 12 Refine tests and documentation for the entire project.
Final week: project submission.

Other commitments

I am currently taking classes at the university, however, the academic period ends on June 6th, which means it will not interfere with the official coding period, and I will be able to work full-time on this project.

References

[Bas97] Peter J Basser. Quantifying errors in fiber direction and diffusion tensor field maps resulting from mr noise. In *5th Scientific Meeting of the ISMRM*, volume 1740, 1997.

[Kin04] Gordon Kindlmann. Superquadric tensor glyphs. In *Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization*, pages 147–154, 2004.

[TCGC04] J-Donald Tournier, Fernando Calamante, David G Gadian, and Alan Connelly. Direct estimation of the fiber orientation density function from diffusion-weighted mri data using spherical deconvolution. *Neuroimage*, 23(3):1176–1185, 2004.