Monterey Bay Aquarium
Research Institute

Visualizations of Deep-Sea Flows Using Lightfield Imaging Data

Brooke Pauken, University of California, San Diego

Mentors: Dr. Kakani Katija

Summer 2020

Keywords: Visualization, 3D Flows, Lightfield, Plenoptic, ParaView, Streamlines

ABSTRACT

The quantitative evaluation of flows within the ocean can lead to a better understanding of how animals have adapted to best survive in their environment. The most effective way to capture this information varies based upon the process of interest, but lightfield imagaing, a three-dimensional imaging technique, is especially valuable for analyzing time-varying flows around surfaces. However, understanding complex small-scale fluid mechanics involved in animal-fluid interactions requires not only the extraction of quantitative information but also the ability to visualize the flow of interest. In this work, one method for visualizing the flow around and within a solmissus is shown using streamlines calculated at moving seed points along the solmissus's bell as it expands and contracts to produce motion. This method is intended to be refined and combined with other visualization techniques in future work to best display the direction and magnitude of flow as related to the surface of the solmissus and other animals of interest.

INTRODUCTION

Analyzing flows is integral to understanding a number of phenomena within the ocean. Animal locomotion, feeding, and reproduction as well as the distribution of nutrients and sediments often involve complex fluid mechanics at a large range of scales. Although one- or two-dimensional measurement systems are satisfactory for analyzing aspects of these processes as well as for understanding general trends, three-dimensional imaging techniques can provide quantitative information about an entire volume in space. Lightfield or plenoptic imaging in particular captures time-varying processes much more effectively than even other 3D methods and requires only a single camera, and is therefore a viable choice for analyzing unsteady, unsymmetric flows, such as those involved in animal-fluid interactions. An early form of plenoptic imaging was first introduced in the early 1900s by Professor Gabriel Lippman [1]. Lightfield imaging today generates a 3D image through the use of an array of micro lenses set between an image sensor and a single main camera lens, allowing for the reproduction of an imaging volume from any viewing angle and can be used to determine the location of individual particles in a 3D [5, 6]. When multiple lightfield images are taken to generate a video sequence, particle tracking software can be used to record the trajectories and velocities of these particles, meaning time-varying, three-dimensional processes can be analyzed successfully. The data used in generating the visualizations presented in the following paper were collected using a Raytrix R26 lightfield camera within EyeRIS, an instrument created by MBARI's Bioinspiration Lab to remotely image the deep sea.

There are a number of ways to visualize three-dimensional flows, each of which highlights a different aspect of the flow in question. Some examples of these methods include a series of two-dimensional contour plots colored by the magnitude of a property of interest, such as velocity, pressure, or vorticity [7, 9]. Similarly, iso-surfaces can be calculated based on vorticity or pressure to convey the effects of a body within a fluid volume in a more three-dimensional way, especially when analyzing drag or rotational flow [8]. Even further, streamlines, streak lines, and path lines can be used often in combination with a colormap indicating the magnitude of velocity to show the direction and speed of flow throughout the entire volume or a selected sub-volume [7, 10]. The final visualizations presented in this work consist mainly of streamlines and path lines, or particle trajectories, in combination with representations of the magnitude and direction of velocity. Although many of the other visualization methods mentioned here were not thoroughly explored, suggestions for which to attempt next are given in the conclusions section.

In this work, two main flows were analyzed using lightfield imaging. First, a set of vortex rings were generated in-lab to provide a known flow in order to validate particle tracking and visualization methods. Particle tracking was accomplished using Raytrix's RxLive and RxFlow, the output of which was then analyzed using Python and Paraview. Second, an in-situ flow around a solmissus was chosen and was processed and analyzed similarly. The solmissus was chosen as a target animal for EyeRIS in particular because of the potential to further study its locomotion. With frequently used two-dimensional imaging, it is rather straightforward to understand the general ways in which jellyfish like the solmissus are able to propel themselves through the water. However, 3D techniques like lightfield imaging allow for the quantitative analysis of how the animal is able to move, and what effect its bodily movement has on nearby particles. In particular, when a jellyfish manipulates its body in such a way that the symmetry in its movements is broken, the temporal resolution available through lightfield imaging make the solmissus a particularly interesting focus for EyeRIS.

MATERIALS AND METHODS

Lab flows details, picture of lab set up from presentation. A short clip of 169 frames was chosen using RxLive in order to only capture the flow of interest and imported into RxFlow to preform particle tracking and velocimetry. Parameters were chosen and run through two to five frames and visually judged based upon the number and spread of particles as well as the number of velocities calculated per frame. After much iteration, some of the most influential paramaters were as follows: the background brightness average was manually estimated to be 7 on a scale of 0 to 255, the particle brightness minimum was similarly chosen to be 16 on the same scale, the particle radius maximum was set to 6 pixels, and the particles were matched from frame to frame based on distance [11]. An approximate background was found and removed before the software was run through all 169 frames. One file containing the positions of all identified particles in each frame and another containing the velocities and positions of all particles identified in two or more frames were then exported. The frames, id labels, positions, and velocity components were then extracted from these outputs in Python and written to a .vtu, a subset of the visualization toolkit dedicated for unstructured datasets, for each frame to be read in ParaView [4]..

EyeRIS details, picture?. The in-situ solmissus data was processed similarly. 115 frames were chosen using RxLive in which the solmissus expanded and contracted its bell to generate one cycle of motion. A background was again found and removed in RxFlow prior to running the software through all frames. The background average brightness was chosen to be 15 and the minimum particle brightness was set at 20 both on the same 0 to 255 scale. The particle radius maximum was again set at 6 pixels. A dataset was generated using the distance-based particle matching setting, but the visual-based matching resulted in more particle trajectories identified around the bell, so the latter was used [11]. The position and velocity outputs were then processed in Python using the same script to create two .vtu sequences.

RESULTS

To first visualize the lab-generated flows, the particle trajectories were found using the 3D positions output from RxFlow and the ParaView Temporal Particles to Pathlines filter. **Average and standard deviation of displacement between frames in each direction.** To filter out erroneous particles with an emphasis on the z-axis, a max step distance of 10 mm in the x and y direction and 5 mm in the z direction was set. Additionally, trajectories were terminated after 100 mm in order to minimize over-crowding of the visualizations [2]. Include animation

Because a particle must be present in two or more frames in order to have an associated velocity, the velocity data outputted by RxFlow was rather sparce in comparison to the positions output. **Av ratio of velocities to particles.** Therefore, in order to best visualize the flows of interest, it was necessary to generate a mesh from the particle velocity output. A 3D Delaunay Triangulation method was used to generate an unstructured grid [2]. Using this mesh, streamlines, which are curves tangent to the velocity field at all points and represent the direction a theoretical particle would be traveling at that point in space and are consequently very effective in displaying the direction of flow, were generated using ParaView's Stream Tracer filter [3]. Runge-Kutta 4-5 integrator in was used in both directions at 1000 seed points which were equally spaced throughout a sphere encompassing the entire field of view in order to generate streamlines throughout the volume [2]. The streamlines were then rendered as tubes to best show 3-dimensional flow and each line was colored distinctly as is shown in Figure 1. Angular velocity, vorticity, and rotation were also calculated through the use of this filter based on the cells resulting from the Delaunay triangulation, though they were not used for the resulting visualizations.

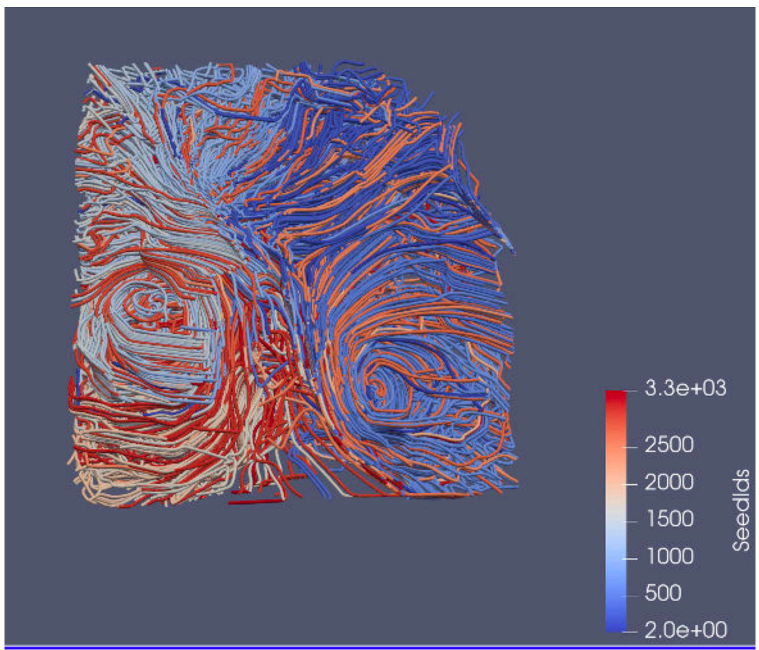

Figure 1: A single frame of streamlines calculated based upon a 3D Delaunay Triangulation of the velocity field of a lab-generated flow is shown, facing the back of the imaging volume. Each different color represents a different theoretical particle represented by a calculated streamline.

The in-situ solmissus data was treated similarly. Particle trajectories were calculated using the same filter and parameters as the lab-generated data. An animation of the particles and trajectories was generated and overlayed on the total focus clip of the solmissus. Using this animation, a number of particles were identified as being representative of one edge of the solmissus's bell. The frame of this animation with some of these points visible is shown in Figure 2. Although the Stream Tracer filter was originally used to generate streamlines throughout the entire imaging volume in the same

Figure 2: The final frame of the particle positions and trajectories overlayed onto the total focus video of the Solmissus clip of interest is shown. Note the grouping of particles within the top edge of the bell which were used to approximate the outer edge of the bell in creating the final streamline animation.

manner as with the lab-generated data, the flow of interest around the bell was obscured by the movement of surrounding particles.

For this reason, a high-resolution line source was instead created to be used as the seed points for calculating the streamlines. The line was manually adjusted to intersect the points of interest around the solmissus's bell by changing the location of the end points over the course of all 115 frames on interest using ParaView's animation view. This was done by separating the movement of the bell into four main segments: the initial expansion, a brief pause at the maximum point, the contraction, and then the final, faster re-expansion. The initial and final positions of the line's end points were adjusted manually, and a ramp interpolation was used for generating the intermediary locations. This process has the potential to be automated for increased accuracy and efficiency, though the initial manual identification of a subset of points of interest will likely still be necessary. Seven seed points were generated evenly spaced on the line source through which streamlines were calculated for each frame. Different numbers of seed points were tested, however, seven was chosen in order to provide the most information without overcrowding of the animation view. A Runge-Kutta 4-5 integration method in both directions was again used to generate the streamlines, which were then rendered as tubes. Finally, the ParaView Glyph filter was used to add glyph cones to the 3D streamlines to best indicate the direction of the velocity field along the streamlines. Both the streamlines and glyphs were colored by the magnitude of velocity. animation

DISCUSSION

Streamlines rendered as three-dimensional tubes in ParaView are presented here as being an effective way to create visualizations of 3D flows. The use of the Delaunay triangulation method to generate a mesh from the velocity field in combination with the choice of source type and number of seed points using the Stream Tracer filter allow for a more dense visualization of the direction and speed of flow than is easily available from the sparsely-defined velocity points in each frame alone. Initially, stream tubes were also considered as a visualization method. However, because the diameter of stream tubes is given by the divergence from the velocity field, the low density of points and high noise present in the lightfield datasets of interest result in misleading divergence values. This falsely highlights certain areas within the imaging volume and can again mask the true flows of interest.

Animations of particle positions and trajectories over time is also shown here as being useful for visualizing 3D flows. However, although the position output from particle tracking software such as RxFlow will always contain more datapoints than that of the velocity output, the additional particles present in the position output will only be present in one frame. Furthermore, a particle within a body of interest is less likely to be accurately identified over multiple frames by RxFlow because of the difference in contrast between particle and animal versus particle and general background, meaning few trajectories will be calculated within the area of interest. For these reasons, trajectories alone should not be used as a visualization of flow around or within an animal or surface. Streamlines, however, if generated based upon a mesh created from the velocity data, are able to be effectively up-sampled by increasing the number of seed points and can also be generated at any sub-location within the imaging volume.

CONCLUSIONS

Looking forward, calculating vorticities in a more accurate way than was possible with the Stream Tracer filter would be a good first step. Although I attempted to do so using other built-in filters such as Gradient of an Unstructured Grid, any useful results were largely obscured by a majority zero output, most likely a result of the sparce velocity field and subsequent mesh. Up-sampling the velocity field points before generating a grid may be the solution to this issue, though to my knowledge there is no built-in filter within ParaView to perform this task. The generation of a python script either within or outside of the ParaView interface would likely the best option. Once vorticities were calculated, the field of view could be rendered as a surface instead of a point cloud and sliced along the middle or edge of the solmissus's bell and colored by vorticity as another visualization technique. Similarly, surfaces could be generated based on different values of vorticity.

It would also be useful to look at energetics in order to analyze the role of background flow and the solmissus's own mechanical energy in generating its movement. This concept was not thoroughly explored by me during the course of this summer but seems possible within the ParaView/Python world.

In terms of improving the visualization methods presented here, a surface reconstruction created either in ParaView or elsewhere and later inputted to ParaView along with the RxFlow particle tracking software output would be the best course of action for generating a more accurate and useful animation of streamlines around the bell. The surface of the bell could be used as a custom source for creating seed points, resulting in a more full picture of the three-dimensional nature of the solmissus's movement and effect on the surrounding particles. Additionally, although attempts at filtering and smoothing the data were made throughout the process of creating these animations, more work certainly needs to be done in this area. For example, ParaView is unable to make computations between data in different timesteps if the number of points in each is different, which is unfortunately the case for the type of data being analyzed here. An error was generated even in using the built-in Python kernel for this purpose, but a Python script focused on filtering and smoothing could theoretically be generated and run outside of ParaView before creating the .vtu file.

With respect to lightfield imaging in general, a few limitations were made clear through working with a number of datasets. Beyond the need for heavy filtering and smoothing of the data, it is important to note that the particle tracking process can be rather iterative and necessitates the manual adjustment of parameters with each new video. The best use case in order to minimize these limitations would include a somewhat steady background especially in terms of average brightness. Similarly, a high contrast between particles and the background will result in a much easier particle tracking experience as well as a less noisy output. Finally, the particle density within the imaging volume is also important to consider. Although there needs to be enough particles spread throughout the volume to show even small-scale flow patterns, a high particle density will result in a decreased signal to noise ratio. A high density will be more valuable than low in that the flows of interest will more likely be captured, however increased processing time and a larger need for aggressive filtering would be necessary in order to extract any useful information.

Lightfield imaging in combination with the visualization methods explored and suggested here is not limited to the case of analyzing jellyfish. Some additional targets of interest include creatures with interesting locomotion, such as certain varieties of squids and siphonophores. It would also be valuable to image predator-prey flow interactions within the same imaging volume in order to analyze the relationship between background flows and feeding as well as to determine the circumstances in which prey could escape their predator.

ACKNOWLEDGEMENTS

I want to first thank the entire Bioinspiration Lab, especially Dr. Kakani Katija, Dr. Joost Daniels, and Dr. Paul Roberts, for enabling this project especially in a virtual format. I would also like to acknowledge Dr. George Matsumoto, Megan Bassett, and Tatjana Ellis for coordinating the MBARI summer internship program this year and for providing so much support throughout the summer, as well as the entire MBARI staff and my fellow interns for being exceptionally welcoming and interested in my project. Finally, I would like to thank the David and Lucile Packard Foundation, the Dean and Helen Witter Family Fund, and the Rentschler Family Fund for providing funding for the internship this summer and continuing to make programs like this one available.

References:

[1] No Credited Author, (1911). Integral Photography: A New Discovery by Professor Lippmann By the French Correspondent of the *Scientific American* : .
doi: 10.1038/scientificamerican08191911-164

[2] Utkarsh Ayachit, (2020). The ParaView Guide: Community Edition. *Kitware Inc.*.

[3] Adrain, R.J., Gogineni, S.P., Smits, A.J., (2020). Bicycle Aerodynamics: Streamlines and Streamtubes. *Efluids.*

[4] Herrera, P.A., (2010-2016). PyEVTK (Python Export VTK). *GitHub.*

[5] Perwab, C., Wietzke, L., (2012). Single Lens 3D-Camera with Extended Depth-of-Field. *Raytrix GmbH* : . doi:10.1117/12.909882

[6] Levoy, M., Hanrahan, P., (1996). Light Field Rendering: .
https://doi.org/10.1145/237170.237199

[7] van den Berg, J., Jux, C., Sciacchitano, A., van de Water, W., & Westerweel, J. (2019). Robotic Volumetric PIV measurements of a full-scale swimmer's hand. C. J. Kähler, R. Hain, S. Scharnowski, & T. Fuchs (Eds.), Proceedings of the 13th International Symposium on Particle Image Velocimetry, Universitat der Bundeswehr Munchen: . https://doi.org/10.18726/2019_3

[8] Jux, C.; Sciacchitano, A.; Scarano, F., (2018). Surface Pressure Visualization by 3D PTV. Conference Paper, ETH Zürich. : https://doi.org/10.3929/ethz-b-000279192

[9] Kitzhofer, J., Nonn, T., Brücker, C., (2011). Generation and Visualization of Volumetric PIV Data Fields. *Exp Fluids:* . DOI 10.1007/s00348-011-1176-1

[10] Tan, Z. P., Alarcon, R., Allen, J., Thurow, B. S., Moss, Anthony., (2020). Development of a High-Speed Plenoptic Imaging System and its Application to Marine Biology PIV. *Measuring Science Tehcnology:* **31** 054005.

[11] No Credited Author, (2014). HowTo for RxFlow Raytrix Particle Tracking and Velocimetry. *Raytrix GmbH.*