VENTRICULAR BLOOD FLOW ANALYSIS USING TOPOLOGICAL METHODS

Scott Kulp[1], *Chao Chen*[1], *Dimitris Metaxas*[1], *Leon Axel*[2]

[1] Department of Computer Science, Rutgers University, Piscataway, NJ 08854
[2] Department of Radiology, New York University, New York, NY 10016

ABSTRACT

Thanks to the advances of data acquisition techniques, we can acquire ventricular blood flow data with very high quality. This extremely complex spatiotemporal data calls for novel visualization and analysis tools. In particular, the new tools need to assist domain experts in quick identification of critical patterns. In this paper, we present a method using topological data analysis tools with simulated ventricular blood flow, and automatically detect interesting topological features within the flow. We show that this completely unsupervised framework detects and extracts eddies formed from vortex shedding during late diastole, which normally requires highly specialized algorithms to capture.

Index Terms— Ventricular flow analysis, spatiotemporal data, topological methods, persistent homology.

1. INTRODUCTION

After a patient survives a heart attack or other cardiovascular disease, they may be at risk of thrombus due to the altered blood flow pattern within their left ventricle. In order to make an accurate diagnosis and prescribe appropriate treatment, doctors must analyze the blood flow patterns within the left ventricle (LV), often using Color Doppler echocardiography or MR phase-contrast flow imaging. In recent years, cardiac blood flow simulation has been shown to produce blood flow fields with much higher resolution and clarity [1, 2] and thus becomes a new alternative source of the ventricular flow analytics. However, as cardiac blood flow fields have become increasingly detailed and complex, more sophisticated tools of flow field visualization and analysis are required to allow doctors to interpret their results.

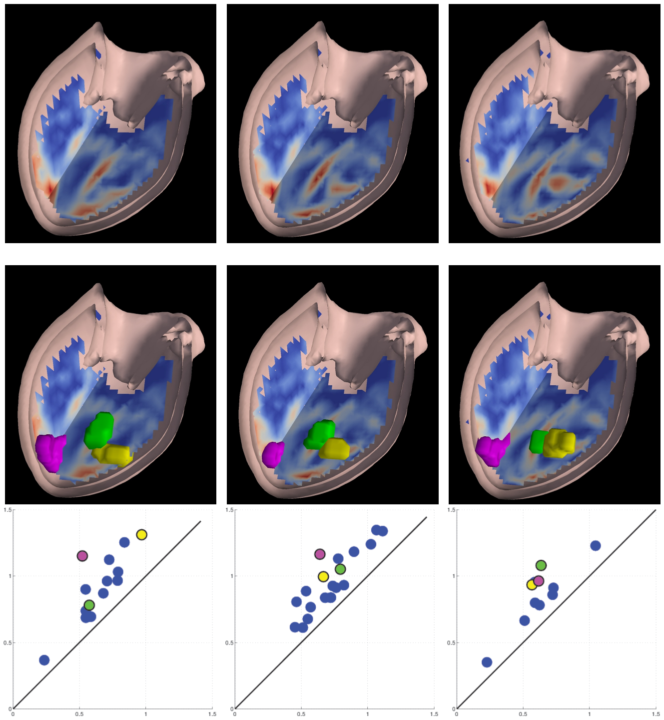


Fig. 1. Evolution of vorticity field (top), three high-persistence cycles (middle), and corresponding persistence diagrams (bottom). The x and y axes are the birth and death of the dots respectively.

Fluid field analysis has been a highly active field of research for over 20 years. Most previous studies either visualize the flow field directly, or are based on flow feature extraction. The former method often uses arrow glyphs, streamlines, or textures to visualize the flow, and a survey of such techniques in the context of cardiovascular blood flow can be found in [3]. Alternatively, the latter method uses classification algorithms to find specific patterns of vector field topology within the flow, such as vortices, critical points and shock waves [4]. [1] Regardless of the rich literature in flow field visualization, the flow features in previous works are insufficient for modern flow analytical tasks. The reason is threefold. First, previous features mostly depend on the singularities of the flow field, and thus are sparse and not discriminative enough. Second, these singularity-based features are well-known to be unstable and thus cannot be used as robust features of the flow data. Third, these features are very expensive to compute and track.

In this work, we present a new exploratory framework for blood flow analytics. Our framework generates a rich set of discriminative, multiscale and robust flow features in an auto-

---

[1] The word ‘topology’ in [5] means the combinatorial structure between singularities like critical points and separatrices. Whereas the word ‘topology’ in this paper means ‘algebraic topology’, a classic field in mathematics.

matic and parameter-free manner. These features, defined by the theory of persistent homology, are thickened one- or two-manifolds of arbitrary scales and shapes. These mid-level features, describing the intrinsic structure of the data, have never been explored in the cardiac flow analytic. We show that our system can automatically detect and track interesting features of the flow, such as vortex shedding during diastole. This preliminary result proves the potential of topological methods in the analytics of cardiac flow.

2. METHODS

There are three main steps to our flow analysis framework. First, we acquire 3D animated models of the left ventricle, to be used as boundary conditions. Next, we pass these boundary conditions to a Navier-Stokes fluid simulator. We compute the flow fields as well as various scalar functions, such as vorticity. Finally, we use these functions as input into our topological methods, which automatically compute features of arbitrary shapes and scales. These features encrypt the intrinsic structure of the flow data and will be used for flow analytics (Section 3).

Data Acquisition. Boundary conditions play a critical role in determining the accuracy of our simulation results. For this work, we use standard 3D model of a human left ventricle, which was reconstructed from the high-resolution Visible Human Project datasets, as discussed by Hurmusiadis et al [6]. Ventricular motion was derived from a fiber-based deformation model, also described in [6]. Models generated using this method were used in [7] to simulate flow through the left ventricle with high accuracy, so we are confident that boundary conditions that these models provide will produce acceptable results for our flow analysis. We then add 3D models of the aortic and mitral valves created from ultrasound data to each mesh in the sequence, and open and close the valves at the appropriate time steps. We then used this full, animated model as the solid boundary conditions in our fluid simulator.

2.1. Fluid Simulation

After we acquire the animated model of the left ventricle, we use it as boundary conditions in our fluid simulator. The motion of an incompressible fluid is governed by the laws of conservation of momentum and mass, modeled by the Navier-Stokes (NS) equations:

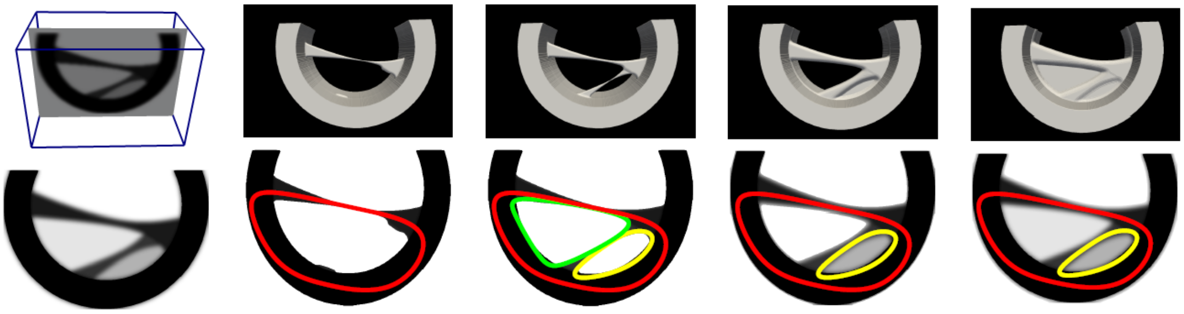
$$\rho(\tfrac{\partial \mathbf{u}}{\partial t} + u \cdot \nabla \mathbf{u}) = -\nabla P + \mu \nabla^2 \mathbf{u},$$
$$\nabla \cdot \mathbf{u} = 0.$$

Here, ρ is the fluid density, $\mathbf{u}$ is the 3D velocity vector field, P is the pressure field, and μ is the coefficient of viscosity. We seek to solve these equations for velocity and pressure. Further, we can compute vorticity through the equation $\omega = \nabla \times \mathbf{u}$.

Foster and Metaxas [8] were the first to develop a fast method of solving the NS equations for graphics applications by applying a staggered grid across the domain and explicitly solving for velocity at the cell faces. Our fluid-solid interaction system uses the immersed-boundary formulation [9], allowing for an simple treatment of complex moving geometries embedded in a closed computational domain. The heart models used here are embedded in a computational mesh of 96^3 cells on which the full NS equations are solved using Finite Difference Method (FDM). Vorticity is then also computed using finite differencing. When computing vorticity, cells near the solid wall boundary (less than two cells distance) are ignored. The blood is modeled as a Newtonian fluid, with viscosity of 4mPa· s and density of 1050kg/m^3, which are physiologically accepted values for normal human blood[10]. The heart model is given to the solver as a set of meshes with point correspondences, which allows for easy interpolation and also obtaining the velocity of the heart mesh at every point in time. Our system represents the 3D meshes as a Marker Level Set (MLS) [7], where markers are placed on the boundary and are used to correct the level set at every time step. Since markers are only placed on the surface, MLS has been proven to be more efficient and more accurate for complex boundaries. The MLS and its velocity are rasterized onto the Eulerian grid and are used to impose Dirichlet boundary conditions for the fluid velocity in the solver. To ensure our analysis is based on a fully-developed flow, we run the simulation for 1.5 cardiac cycles, and remove the first half-cycle before analysis.

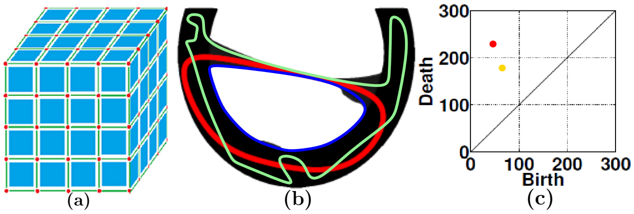
2.2. Persistent Homology

Persistent homology [11] is a topological data analysis tool which uses the language of algebraic topology to define a signature of a given function, e.g. the vorticity, velocity or pressure of the cardiac flow. The output, persistence diagram, is a set of dots on a 2D Euclidean plane. In Figure 2(a) and 3(c), we show a 3D function and its persistence diagram. Each dot corresponds to one topological structure that is salient relative to its surroundings. In this example, we clearly observe two handles, i.e. thickened loops. The reason is because the function values along these handles are relatively dark yet the function values inside these handles are relatively white. We call the maximal function value along a handle its birth and call the maximal value inside the handle its death. The difference between the death and birth of a handle, called its persistence, measures how salient this handle is, or say, how easily a human can observe it. Each handle corresponds to a dot in the diagram using the birth and death as the coordinates. The persistence is simply the distance of the dot from the diagonal. In general, a topological structure could be a connected component, a handle, or a void (a thickened sphere). The theory of persistent homology provides a principled definition of these topological structures and an efficient algorithm to detect all of them, regardless of their shapes and scales. Per-



(a) function (b) $\ell = b_1$ (c) $\ell = b_2$ (d) $\ell = d_2$ (e) $\ell = d_1$

Fig. 2. Illustration of persistent homology. From left to right: a 3D scalar function, the sublevel sets (see text for definition) at thresholds $b_1 < b_2 < d_2 < d_1$.



(a) (b) (c)

Fig. 3. (a) A cubical complex; (b) A set of cycles belonging to a same homology class, corresponding to the big handle in Figure 2; (c) the persistence diagram of the synthetic function in Figure 2(a).

sistent homology has many theoretical guarantees, making it appealing in both theory and application. Cohen-Steiner et al. [12] showed that the persistence diagram is stable with regard to changes of the function.

To detect and compute all topological structures in the persistence diagram, we need to go through all real values. We use each value as a threshold to cut the domain, namely, remove all points of the Euclidean domain whose function value is bigger than the threshold. In Figure 2(b-e), we show the cut results, called the *sublevel sets*, using four different thresholds. Notice that the sublevel set in Figure 2(b), corresponding to threshold b_1, has the larger topological handle. But the sublevel set using any smaller threshold would not have this handle. Therefore we know that b_1 is the birth threshold of the larger handle. Similarly, we know that b_2, d_2 and d_1 are the birth of the smaller handle, the death of the smaller handle and the death of the larger handle, respectively, based on the corresponding sublevel sets (Figure 2(c-e)). We use the algorithm by Edelsbrunner [13] which efficiently identifies all these critical thresholds, and more importantly, pair them correctly, e.g. pair b_1 with d_1 and b_2 with d_2. The algorithm, like most other algorithms computing persistent homology [14], bases on a discretization of the domain into complexes (Figure 3(a)) and linear algebra techniques, in particular, matrix reduction algorithms.

Computation of representing cycles. Formally, the topological structures we are interested in are called homology classes [11, 15]. Roughly speaking, a homology class is an equivalent class of cycles which can be continuously deformed into each other within the space.[2] A d-cycle is a d-manifold without boundary. For example, in our scenario, we are interested in one- and two-dimensional cycles, i.e. loops and bubbles. In many applications, we need to describe a homology class using one of its cycles. In such case, it would be useful to find the cycle with the most concise geometric measure, e.g. length or area, depending on the dimension of the cycle. Figure 3(b) shows three cycles of a same class, namely, the bigger handle of the function in Figure 2. The shortest one (blue) is the optimal one, best describing the class. In the computation of persistent homology, instead of computing the optimal one, we often use a convenient choice, namely, the one corresponding to the remaining column of the reduced matrix. This cycle is a reasonable approximation of the optimal cycle as we observed in practice. It also comes free computationally as the result of the algorithm for the computation of persistence diagram. In our experiments, we compute one cycle for each topological structure in the persistence diagram. In Figure 2, the cycles for the long and short handles are the red and yellow cycles respectively. We color their corresponding dots accordingly.

3. RESULTS AND DISCUSSION

Using the vorticity function of the flow field, we extracted three topological structures of high persistence during late diastole. In Figure 1, we visualize the vorticity function in three time steps near end diastole (top), topological structures with high persistence overlaid on the vorticity field (middle) and the corresponding persistent diagrams (bottom). In Figure 1 (top) and (middle), the color at a point is representative magnitude of vorticity at that location - blue regions represent 0 s^{-1} vorticity, while red regions represent 1.07 s^{-1}. As mentioned earlier, cells close to the heart wall are ignored in vorticity computation, causing the jagged border.

We select the three topological structures (green, magenta and yellow) with the highest persistence from the vorticity field of the end diastole (the right column in Figure 1). We only chose three for visualization purpose. In any learning algorithm, we would use all topological structures as features

---

[2] In fact, using "continuous deformed into each other" as the equivalence relationship is a very strong definition. A homology class is an approximation of this equivalent class, called *a homotopy class*.

of the data. Therefore, our method is a completely parameter-free feature generation algorithm. We find that these structures highlighted eddies created from vortex shedding, a phenomenon that occurs when viscous fluid flows across an obstruction, and causes vortices to periodically form behind the solid body and detach into the flow. Such shedding naturally occurs in a healthy flow across the mitral valve wall, and a lack of vortices may indicate pathology in LV or mitral valve function [16]. Normally, detection of vortex shedding is performed by specialized algorithms, such as in [17], but our method found these topological features completely automatically and without supervision. For completeness, we also track and show these three structures at other time frames (the left and middle columns of Figure 1). This provides a vivid illustration of the movements and shapes of these these eddies in the end-diastole stage.

Please note that this method is not simply detecting the global maxima within the flow, as the true global maximum for vorticity for this time frame is within the mitral valve. Furthermore, our algorithm is not simply finding local maxima. Within a same topological structure we show, there are many local maxima. Rather, our method is finding truly interesting flow features, which could be valuable analytical information in a clinical setting.

4. CONCLUSION

In this paper, we have described our new framework to automatically detect interesting features in flow topology through the left ventricle using persistent homology. In our experiment of blood flow through a healthy left ventricle, this unsupervised method was able to highlight vortices caused by vortex shedding, which is normally a very difficult problem. As the flow fields we can acquire become increasingly complex and high-resolution, such an automatic feature detection system could be highly valuable in a diagnostic setting.

We note that the proposed method addresses all the three issues of previous flow features as discussed in Section 1. First, persistent homology generates a rich set of features, each including a dot in the diagram and a generator describing its geometry. These features are discriminative for future learning tasks. Second, the computed features are partially stable. The persistence diagram has been proven to be stable. The stability of generators needs further investigation theoretically and experimentally. Third, the proposed algorithm is efficient and thus can be used for large scale data analysis. The computation of persistence diagram and generators is fast. It takes about half a minute to compute the feature for each time frame. Of course the simulation itself takes a long time (16 hours per cycle). But this is the prerequisite of any flow analysis method.

In the future, we plan to apply our method to more datasets. To validate the discriminating power of these topological features, we plan to apply our method to data from patients with various diseases like dyssynchrony. We would also hope the topological structures could reveal insight into physical symptoms of these diseases.

Acknowledgements

This paper is partially supported by the grants NSF IIS 1451292 and NSF CNS 1229628.

5. REFERENCES

[1] T. Jones, T. N. Jones, and D. N. Metaxas, “Patient-specific analysis of left ventricular blood flow,” in *MICCAI*, 1998, pp. 156–166.

[2] S. Kulp, M. Gao, S. Zhang, Z. Qian, S. Voros, D. N. Metaxas, and L. Axel, “Using high resolution cardiac ct data to model and visualize patient-specific interactions between trabeculae and blood flow.,” in *MICCAI 2011*, 2011, pp. 468–475.

[3] P. P. Sengupta, G. Pedrizzetti, P. J. Kilner, A. Kheradvar, T. Ebbers, G. Tonti, A. G. Fraser, and J. Narula, “Emerging trends in cv flow visualization,” *JACC: Cardiovascular Imaging*, vol. 5, no. 3, pp. 305–316, 2012.

[4] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch, “The state of the art in flow visualisation: Feature extraction and tracking,” in *Computer Graphics Forum*. Wiley Online Library, 2003, vol. 22, pp. 775–792.

[5] A. Pobitzer, R. Peikert, R. Fuchs, B. Schindler, A. Kuhn, H. Theisel, K. Matkovic, and H. Hauser, “On the way towards topology-based visualization of unsteady flow-the state of the art,” *H. und E. Reinhard (Hrsg.), Eurographics*, 2010.

[6] V. Hurmusiadis and C. Briscoe, “A functional heart model for medical education,” in *Functional Imaging and Modeling of the Heart*, vol. 3504 of *Lecture Notes in Computer Science*, pp. 85–91. Springer Berlin Heidelberg, 2005.

[7] V. Mihalef, D. Metaxas, and M. Sussman, “Textured liquids based on the marker level set.,” *Comput. Graph. Forum*, vol. 26, no. 3, pp. 457–466, 2007.

[8] N. Foster and D. Metaxas, “Realistic animation of liquids,” *Graph. Models Image Process.*, vol. 58, pp. 471–483, September 1996.

[9] M. Sussman, “A parallelized, adaptive algorithm for multiphase flows in general geometries,” *Comput. Struct.*, vol. 83, no. 6-7, pp. 435–444, 2005.

[10] V. Mihalef, R. Ionasec, Y. Wang, Y. Zheng, B. Georgescu, and D. Comaniciu, “Patient-specific modeling of left heart anatomy, dynamics and hemodynamics from high resolution 4d CT,” in *ISBI*, 2010, pp. 504–507.

[11] *Computational topology: an introduction*, American Mathematical Soc., 2010.

[12] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, “Stability of persistence diagrams,” *Discrete & Computational Geometry*, vol. 37, no. 1, pp. 103–120, 2007.

[13] H. Edelsbrunner, D. Letscher, and A. Zomorodian, “Topological persistence and simplification,” *Discrete and Computational Geometry*, vol. 28, no. 4, pp. 511–533, 2002.

[14] C. Chen and M. Kerber, “An output-sensitive algorithm for persistent homology,” *Computational Geometry*, vol. 46, no. 4, pp. 435–447, 2013.

[15] *Elements of algebraic topology*, Addison-Wesley Reading, 1984.

[16] *Recent Progress in Mitral Valve Disease*, Elsevier Science, 2013.

[17] M. Jiang, R. Machiraju, and D. Thompson, “Detection and visualization of vortices,” *The Visualization Handbook*, p. 295, 2005.