Topological Analysis of the 2D von Kármán Street

Adhitya Kamakshidasan, Vijay Natarajan

Topological Analysis of the 2D von Kármán Street

Adhitya Kamakshidasan*
Inria, Saclay

Vijay Natarajan†
Indian Institute of Science, Bangalore

ABSTRACT

Topology based analysis and feature tracking is a well studied area. In this work, we focus exclusively on a dataset called the von Kármán street, and apply topology-based methods to understand its vortices. For this analysis, we adapt the recently proposed edit distance between merge trees. We discern several interesting results. One, we observe spatial periodicity between the vortices, alternating every half-cycle. Two, we observe a distinct difference in spatial probability of vortex regions during a half-cycle. Further, we compare the accuracy of our spatial probability with an off-the-shelf machine learning approach.

Keywords: topological data analysis, feature tracking, von Kármán street, edit distance, machine learning

1 INTRODUCTION

The von Kármán street is formed, after the initial growth phase of a two-dimensional viscous flow around a simple solid cylinder, wherein a fluid is injected to the left of a channel bounded by solid walls. This is a time-dependent flow field and is often used as a test dataset for validating topological data analysis (TDA) methods for scalar and vector fields. In this poster, we present a few previously unreported findings on the vortex structure in this dataset.

2 MERGE TREES

Consider a scalar function $f \colon D \to \mathbb{R}$ defined on a manifold domain D. A value c in the range of f is called an *isovalue*. Given an isovalue, an *isocontour* or *level set* is defined as the collection of all points $x \in D$ such that $f(x) = c$. A merge tree captures the connectivity of sub-level sets $f^{-1}(-\infty, c]$ (*join tree*) or super-level sets $f^{-1}[c, \infty)$ (*split tree*) of f. The split tree consists of maxima M = $\{M_i\}$, split saddles S = $\{s_j\}$, and the global minimum. The join tree consists of minima m = $\{m_i\}$, join saddles S = $\{s_k\}$, and the global maximum. We use the split tree defined over the velocity magnitude field of the von Kármán street, since its maxima directly correspond to vortices. Figure 1 shows the split tree of a single timestep of the vortex street.

3 FEATURE TRACKING

Several techniques are available for comparing time-varying scalar fields. Since our focus is on vortices of the von Kármán street, it is sufficient to compare the split trees of the corresponding timesteps. In this regard, we extend the tree edit distance approach by Sridharamurthy et al. [4] to compare split trees. Similar to how two strings can be transformed into one another, via a sequence of relabel, delete and insert operations, two split trees can also be transformed, using a persistence based cost model.

While the work on tree edit distance, focused on providing a distance measure that is more discriminative than existing measures such as bottleneck and Wasserstein distances, we believe that it can be extended for feature tracking in time-varying data. Let us assume

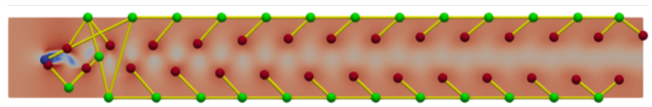
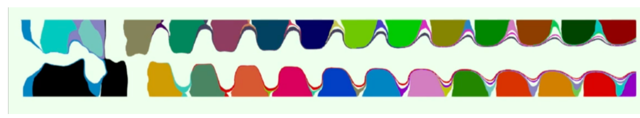
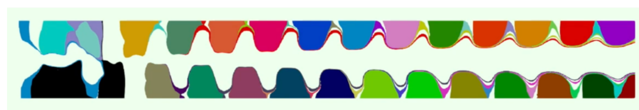


Figure 1: First time step of the von Kármán street rendered using a blue-red colormap ( ) and split tree of the velocity magnitude field.



(a) Segmented regions of Timestep 11



(b) Segmented regions of Timestep 48

Figure 2: Tracked regions flip after each half-cycle.

we have time-varying data of k timesteps, along with the corresponding split tree computed for each timestep. When one split tree is compared to another using edit distance, each existing node of a tree is either relabeled or deleted. Now, by comparing the first two timesteps of a dataset, the set of all relabeled nodes of the first split tree $\{R_1\}$, will have a one-to-one correspondence with the set of all relabeled nodes of the second split tree $\{R_2\}$, denoted by $\{R_1\} \to \{R_2\}$. Similarly, by comparing the second and third timesteps, we can obtain another such relabel correspondence $\{R_2\} \to \{R_3\}$. Therefore, for a dataset with k timesteps, we can have a sequence of relabel correspondences $\{R_1\} \to \{R_2\} \to ... \to \{R_k\}$. Such a relabel sequence directly allows for tracking of critical points, and hence feature tracking in a time-varying dataset.

4 ANALYSIS

The von Kármán street street dataset used in this work is publicly available [5], the velocity magnitude over 1001 timesteps on a 400×50 grid. We chose to use an ordered tree edit distance based on APTED [2] with the persistence based cost model. For all our experiments, we compare each timestep to the first timestep and track the critical points using the method proposed above. We perform merge tree segmentation based on [3].

4.1 Spatial Periodicity

Prior work has found the flow to be periodic in nature, with a time period of ≈ 75 timesteps and a half-cycle of ≈ 37 timesteps [1,4]. Here, we would like to additionally visualize and track the vortices that exhibit the periodic behavior, see Figure 2. We observe that the tracked regions flip from above to below the cylinder (and vice-versa) at every half-cycle, when a new vortex is formed to the right of the cylinder and an old one is destroyed near the right boundary of the domain. A detailed video is available in the supplementary

*e-mail: adhitya.kamakshidasan@inria.fr
†e-mail: vijayn@iisc.ac.in

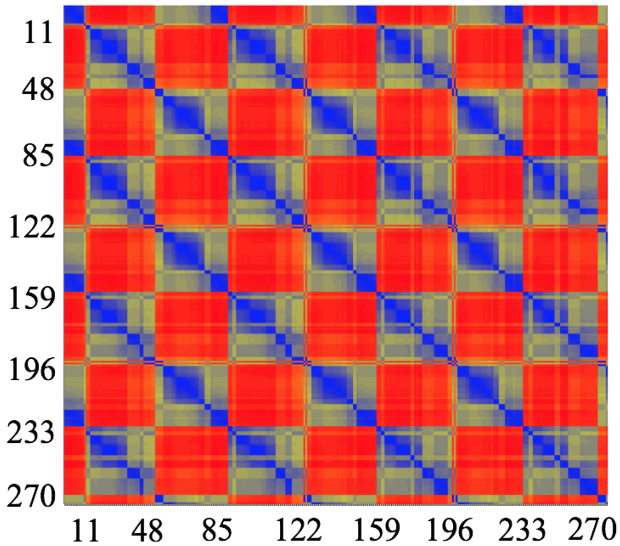


Figure 3: Spatial distance matrix between 275 timesteps, showing a half-period of 37 in addition to the periodicity of 75 using a blue-yellow-red colormap ( ).

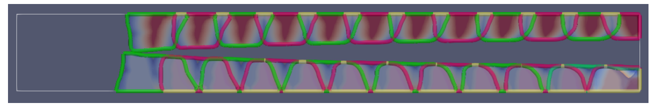
material. In order to check whether such flipping is general in nature, we created a pairwise spatial distance matrix between the first 275 timesteps, see Figure 3. For each pair of timesteps, we compute and plot the the sum of all Euclidean distances between tracked critical points in the spatial domain. This matrix clearly shows that the flipping is consistent across time. We call such a flipping in vortex tracks as a vertical instability.

4.2 Spatial Probability

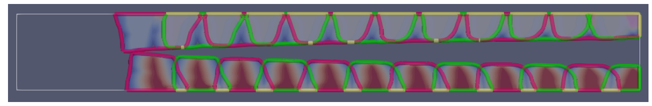
In Section 4.1, we saw that the tracked regions flip vertically, after each half-period. Now, we want to test for horizontal instability, as well. After feature tracking, we found that the vortices don't always match to the same spatial vortex in subsequent timesteps, but might also be matched to a vortex spatially horizontal to it. Let us call the set of all timesteps that do not vertically flip in comparison to the first timestep as Phase *I* and the set of all timesteps that vertically flip as Phase *II*. Since the domain remains the same in all timesteps, for each pixel in a single phase, we can assign a probability as, *Count*(pixel getting mapped to a particular vortex)/*Count*(timesteps). We can now understand the spatial probability of vortices, see Figure 4. In Phase *I*, we found that the set of vortices above the cylinder exhibit no horizontal instability, while the set of vortices below the cylinder exhibit horizontal instability. We found the converse of this observation, for vortices in Phase *II*.

4.3 Accuracy Comparison

From Section 4.1 and 4.2, we saw that there exists a notion of horizontal and vertical instabilities between the two phases of the dataset. We now test whether this behavior can be captured by a machine without providing the phase information. We model a node in the merge tree as a tuple: (*timestep*, *identifier*, *value*, *birth*, *death*) $\rightarrow$ *class*. A node from a given timestep can only get relabeled to a single node from the first timestep. So, we may consider each node of the first timestep as an individual class. Then, a node of any timestep should belong to a particular class. If the machine can predict using a tuple, which class the node belongs to, this it is equivalent to finding the relabel mapping. This turns out to be a simple classification problem. We split the dataset in the ratio 80:20, where 80% is used for training and the rest for testing. In our case,

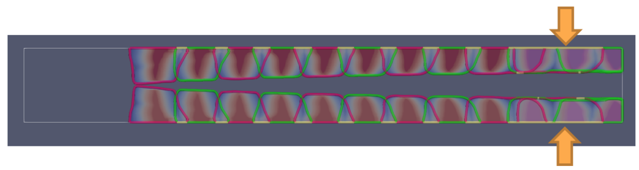


(a) Probability of vortices in Phase *I*

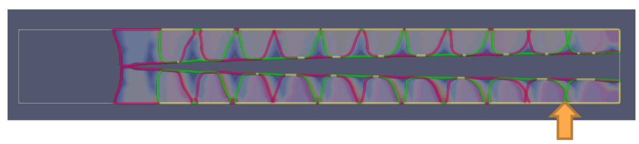


(b) Probability of vortices in Phase *II*

Figure 4: Spatial probability of vortices in both phases, rendered using a blue-red colormap ( ). Spatial probability boundaries of vortex regions are highlighted. Adjacent such regions are in pink and green for better perception, while overlapped boundaries in yellow.



(a) Predicted vortices above & below the cylinder (Phase *I* & *II* respectively)



(b) Predicted vortices above & below the cylinder (Phase *II* & *I* respectively)

Figure 5: Predicted regions of vortices rendered using a blue-red colormap ( ). Halo regions pointed by arrows show regions with lesser accuracy.

all tuples from the first 800 timesteps are used for training. The accuracy of classification algorithms are as follows: K-Neighbours (11.53%), LDA (42.14%), Naïve Bayes (81.84%), Decision Tree (92.7%) and Random Forest (95.85%). Then, using a Random Forest Classifier, we visualize our predictions, see Figure 5. We observe that our model has a good prediction overall. Prediction is not good close to the end of the street where the magnitude of velocity is small.

Acknowledgments

This work is partially supported by the Department of Science and Technology, India (DST/SJF/ETA-02/2015-16), a Mindtree Chair research grant, and the Robert Bosch Centre for Cyber Physical Systems, Indian Institute of Science, Bangalore.

References

[1] V. Narayanan, D. M. Thomas, and V. Natarajan. Distance between extremum graphs. In *PacificVis*, April 2015.

[2] M. Pawlik and N. Augsten. Efficient computation of the tree edit distance. *ACM Trans. Database Syst.*, Mar. 2015.

[3] M. Sharma and V. Natarajan. On-demand augmentation of contour trees. ICVGIP 2018.

[4] R. Sridharamurthy, T. B. Masood, A. Kamakshidasan, and V. Natarajan. Edit distance between merge trees. *IEEE TVCG*, 2020.

[5] T. Weinkauf and H. Theisel. Streak lines as tangent curves of a derived vector field. *IEEE TVCG*, 2010.