Robust Trajectory-based Density Estimation for Geometric Structure Recovery

Turner Richmond, Namita Lokare and Edgar Lobaton
Department of Electrical and Computer Engineering
North Carolina State University, Raleigh, NC 27606
Email: trichmo@ncsu.edu, ndlokare@ncsu.edu, edgar.lobaton@ncsu.edu

***Abstract*—We propose a method to both quickly and robustly extract geometric information from trajectory data. While point density may be of interest in some applications, trajectories provide different guarantees about our data such as path densities as opposed to location densities provided by points. We aim to utilize the concise nature of quadtrees in two dimensions to reduce run time complexity of counting trajectories in a neighborhood. We compare the accuracy of our methodology to a common current practice for subsampling a structure. Our results show that the proposed method is able to capture the geometric structure. We find an improvement in performance over the current practice in that our method is able to extract only the salient data and ignore trajectory outliers.**

***Index Terms*—Trajectory counting, density estimation, landmark selection, quadtree.**

I. Introduction

The increasing availability of large data sets containing spatiotemporal or geometric data is driving the need for data analysis methodologies. Obtaining counts of trajectories for regions in the data provides a means for analysis.

Researchers have explored applying the principles of clustering trajectories to determine animal movement patterns [1]–[3], mapping roads from vehicle GPS data [3]–[5], and describing environmental characteristics such as fault lines [5] and hurricane forecasts [1]. The trajectory clustering techniques are often used to supply a set of descriptive trajectories, yet these descriptive trajectories may also be provided from a region's trajectory density. Trajectory counts over regions also relate to subsampling techniques such as maxmin subsampling utilized in [6]. Density based methods have been used in the context of clustering as in DBSCAN [7], where density of points has been taken into account for purpose of clustering. The density based method TRACLUS [1] is also used in clustering, yet it attempts to cluster trajectory segments. We note TRACLUS avoids the resolution guarantees presented as part of the proposed method. Similarly in [8], the authors use the density of edges obtained by aggregating segmentation from different algorithms.

The strategy presented in the following paper aims to generate a representative subsampling of points that captures the structure of a trajectory-based density function in the space. In order to validate the correctness of our approach, we present a comparison of topological features with point-density based subsampling. As an example application, the topological features of sensor data on a person's joint can be used for activity recognition as shown by the authors of [9]. Yet we note the running time of density based subsampling does not allow for real time activity recognition. We assert the proposed method contributes to the work required to allow sensor data to be utilized in real time activity recognition through calculation of topological features.

This paper is organized as follows: section II gives an overview of the tools we used in our method, followed by our approach and implementation details in sections III and IV respectively. In section V, we compare our approach with the existing density based subsampling method. In section VI, we discuss applications to activity recognition; and finally, conclusion and future work is presented in section VII.

II. Background

The structure of the data provides important geometric information that can be used for inference. Often, such data can be represented as a point cloud with each point corresponding to a noisy observation. Topological data analysis provides us with tools to analyze such datasets. *Persistence Homology* is one such tool that allows us to analyze point clouds over a scale parameter ϵ, and extracts topological features of the dataset at specific scales [10]. A *filtration* $X(\epsilon)$ can be obtained by increasing ϵ over a range of interest, with the property that if $t < s$, then $X(t) \subset X(s)$ meaning that all simplices in $X(t)$ are included in $X(s)$ (see Figure 1). As an illustration, we can create a list of simplices $X(\epsilon)$ where each simplex represents a clique with the property that every pair of points is within ϵ distance from each other.

Persistent homology computes the values of ϵ for which the classes of topological features appear (b^i_n) and disappear (d^i_n) during filtration, referred to as the *birth* and *death* values of the i-th class in dimension n. This information is encoded into persistence intervals $l^i_n = [b^i_n, d^i_n]$, or a multi-set of

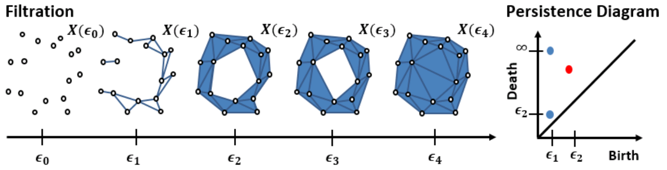

Fig. 1: Illustration of filtration sequence and the corresponding persistence diagram

ISBN 978-0-9928626-7-1 © EURASIP 2017

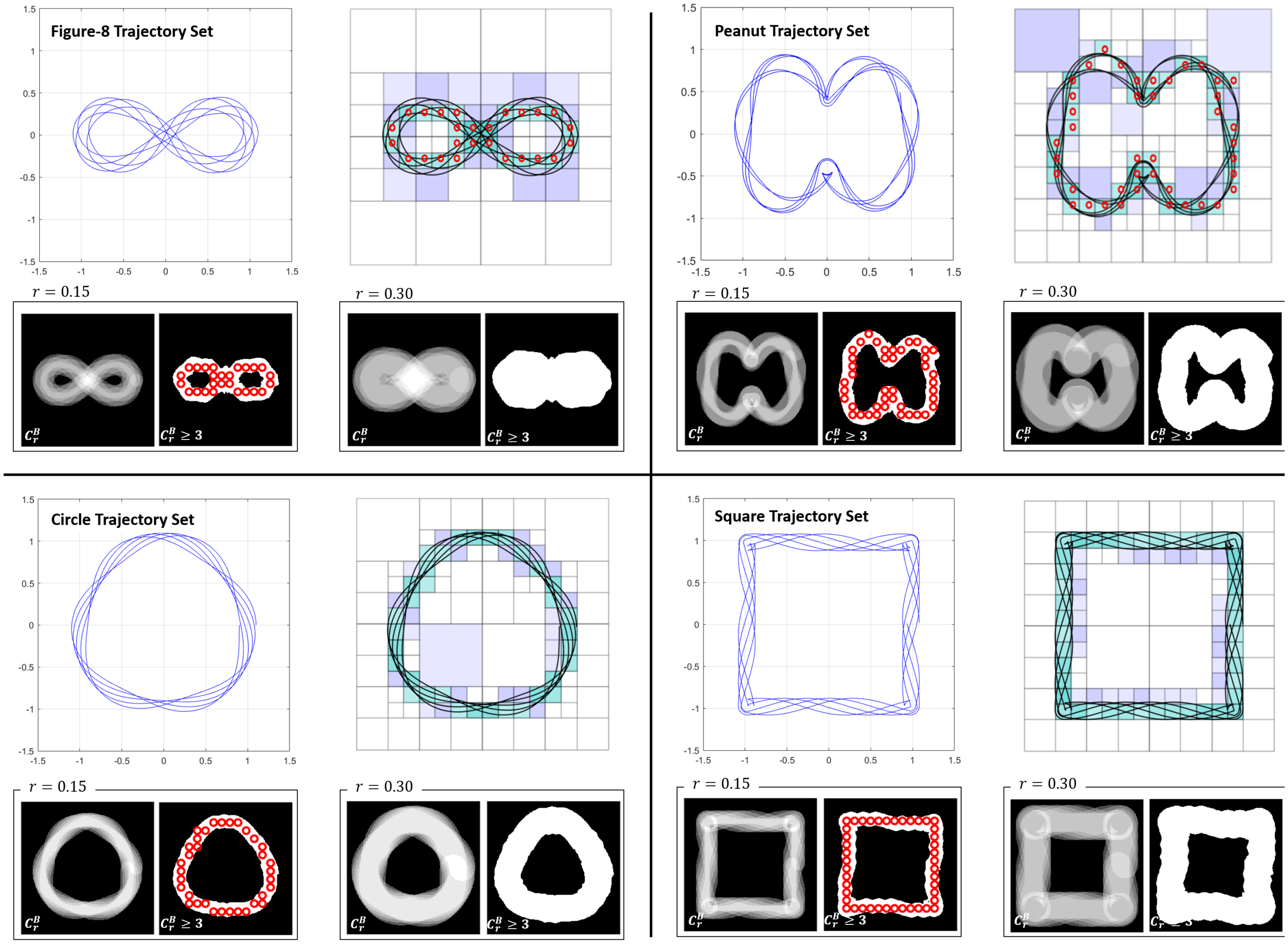

Fig. 2: Illustration for figure-8 (top-left panel), peanut (top-right panel), circle (bottom-left panel) and square (bottom-right panel) trajectory sets. In each panel, the trajectories (top-left), our TRED representation (top-right), and count functions for $r = 0.15$ and $r = 0.30$ (bottom) are shown. The landmarks selected by our approach are shown in red on the level sets for the corresponding count functions. This show how these landmarks capture the same geometric structure as C_r^B.

points (b_n^i, d_n^i), called *persistence diagram* [11], denoted as PD_n (see Figure 1). The persistent diagram introduced by Edelsbrunner et al. [11], is a multi-set of points (b_n^i, d_n^i) in the extended plane that summarizes how topological features of the point cloud vary over ϵ, which represent the prominence of such features. Note the points (b_n^i, d_n^i) live above the diagonal line which denotes $birthvalue = deathvalue$. The points close to the diagonal appear and dissappear very quickly, while the points that are father away from the diagonal are more persistent and are considered as significant features. Algorithms for computation of persistent homology can be found in [12].

Subsampling strategies based on density are used widely to get representative samples from a point cloud in order to handle outliers as well as reduce computational complexity. The maxmin density based subsampling [6] is a landmark selection strategy that maximizes the distance from the previous selected landmarks. The algorithm chooses a random point x_i from the point cloud X , and then the following landmarks are chosen inductively. Let $\mathcal{X}_{k-1} = \{x_1, x_2, .., x_{k-1}\}$, be the chosen landmarks. For the next landmark x_k, we want to pick a sample point that maximizes $min(D(x_k, \mathcal{X}_{k-1}))$, where D is the distance metric in the space.

III. Methodology

We begin by defining a dense version of density estimation procedure. We consider a set of trajectories $\{\gamma_k\}_{k=1}^N$ embedded in $\mathbb{R}^2$, where each $\gamma_k \subset \mathbb{R}^2$ is the trace of the trajectory in the space. We interpret a trajectory as the realization of a spatio-temporal process containing stochastic components centered about the true mean representative trajectory. We assume that these trajectories have a bounded maximum curvature notated as κ_{max}. In order to recover some information about the density of these trajectories over the space, we aim to determine the number of trajectories around a particular point.

ISBN 978-0-9928626-7-1 © EURASIP 2017

However, picking a neighborhood that is too large would not capture the appropriate geometric structure that we are after. Hence, we only consider neighborhoods that have a radius of curvature at most $1/\kappa_{max}$. This ensures that curves are unable to self-intersect within a given neighborhood. Formally, we define the following trajectory count function $C^B_r : \mathbb{R}^2 \to \mathbb{N}$ for $r \leq \frac{1}{\kappa_{max}}$:

$$C^B_r(x) = \sum_k cc[\gamma_k \cap B_r(x)], \tag{1}$$

where $cc[E]$ is the number of connected components in the set E, $B_r(x)$ is an open ball of radius r around $x \in \mathbb{R}^2$. Figure 2 (left) illustrates this count function for a set of trajectories for two values of the support radius r. Note that for $r > 1/\kappa_{max}$ the count function does not capture all of the structure present in the set since curves can actually wrap around inside B_r.

We also defined a similar count for an open square $S_r(x)$ with half side-length equal to r and centered at x. In this case,

$$C^S_r(x) = \sum_k cc[\gamma_k \cap S_r(x)]. \tag{2}$$

We definite the size of this square such that B_r is inscribed in S_r. This count can be computed more efficiently using a hierarchical quadtree approach (as discussed later). However, this count has some unavoidable instability. If we perturb a trajectory that moves along one of the straight edges of the square, we can move it back and forth into the square (increasing the count) but without violating our curvature constraint. Hence, we consider a modified count for the case of the square

$$C^S_{r_1,r_2}(x) = \sum_k cc[\gamma_k \cap S_{r_1}(x) | \gamma_k \cap S_{r_2}(x)], \tag{3}$$

where $r_1 < r_2$ and $cc[E_1|E_2]$ counts the number of connected components in E_1 modulus the connected components in E_2 (i.e., if there are multiple segments in S_{r_1} that are connected in S_{r_2} then they count as one).

It is easy to see that $C^S_{r_1,r_2}(x) \leq C^S_{r_1}(x)$. Furthermore, assuming that $\sqrt{2} \cdot r_1 \leq r_2 \leq 1/\kappa_{max}$ then any trajectory that enters and leaves B_{r_1} cannot enter back into B_{r_1} without leaving S_{r_2}. So, $C^B_{r_1}(x) < C^S_{r_1,r_2}(x)$. A similar argument ensures that $C^S_{r_1,r_2}(x) < C^B_{r_2}(x)$. Hence, we have the following result:

Theorem 1. Given that $\sqrt{2} \cdot r_1 \leq r_2 \leq 1/\kappa_{max}$, then for all $x \in \mathbb{R}^2$ we have that

$$C^B_{r_1}(x) < C^S_{r_1,r_2}(x) < C^B_{r_2}(x). \tag{4}$$

IV. Hierarchical Implementation

We introduce our Trajectory based Representation for Estimation of Density (TRED) representation via a quadtree data structure to describe the geometric structure of the set of trajectories. For the implementation we will assume that there is only a single continuous trajectory. However, the approach can be extended without any difficulty. We define the bins of the quadtree to be square regions. A bin may be split into four equally sized and pairwise disjoint child bins. The ancestors of a bin are all bins with which its intersection is not null. We use the count of a bin's ancestors to describe the depth of a bin. Therefore given a bin with depth n, the length of its edges will be a factor $1/2^n$ shorter than the length of the edge of the bin with depth 0. The maximum depth of the quadtree is chosen such that the deepest bin satisfies the condition on the length of a bin's side as described in Theorem 1. We empirically set $\delta = 0.05$ and $r_2 = r_1 + \delta$ for all square sizes in our experiments. When describing the structure of a set of trajectories, the bins with maximum depth are of interest.

The quadtree is initialized with a square bounding box to ensure all bins are square. A top down approach is taken to divide larger bins into smaller bins of greater depth. A bin is split when the bin's depth is less than the maximum depth and the bin contains at least τ connected components. The value of $\tau = 3$ was set empirically for our experiments.

The count on the number of connected components $C^S_{r_1,r_2}(x)$ in a bin is maintained for a bin given it has no child bins. The trajectory count is incremented for a bin if the initial point is within the bin. By moving along a trajectory γ, we use S_{r_2} to determine if the trajectory leaves a bin before checking which new bin it has entered. The trajectory count of a bin is incremented once the trajectory enters the bin. It was observed that the value of τ helps in reducing the effects of noise along a bin boundary making the representation more stable.

Once the quadtree has been constructed, the set of points $\Pi = \{\pi_i\}$ is used to capture the geometric structure of the trajectory density. Each π_i is the center point of a bin which

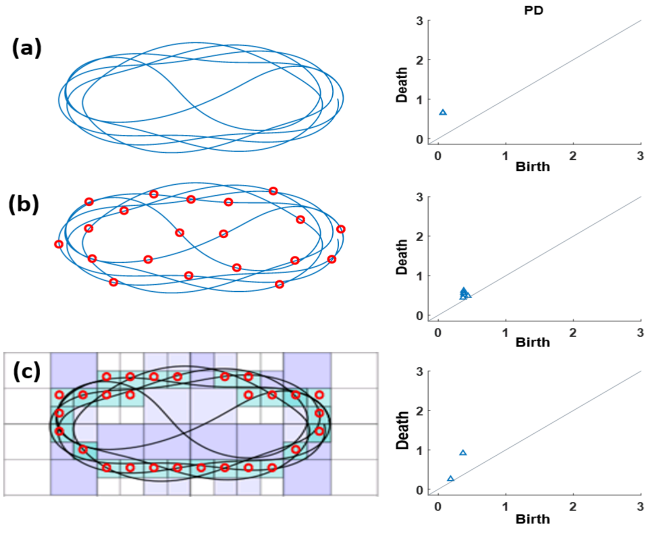

Fig. 3: Comparing Sampling Strategies. (a) Original datastream showing outlier trajectory, and persistence diagram PD_1 for groundtruth ellipse, (b) trajectories with landmarks (highlighted in red) from maxmin baseline method, (c) transition with landmarks (highlighted in red) from proposed TRED subsampling method.

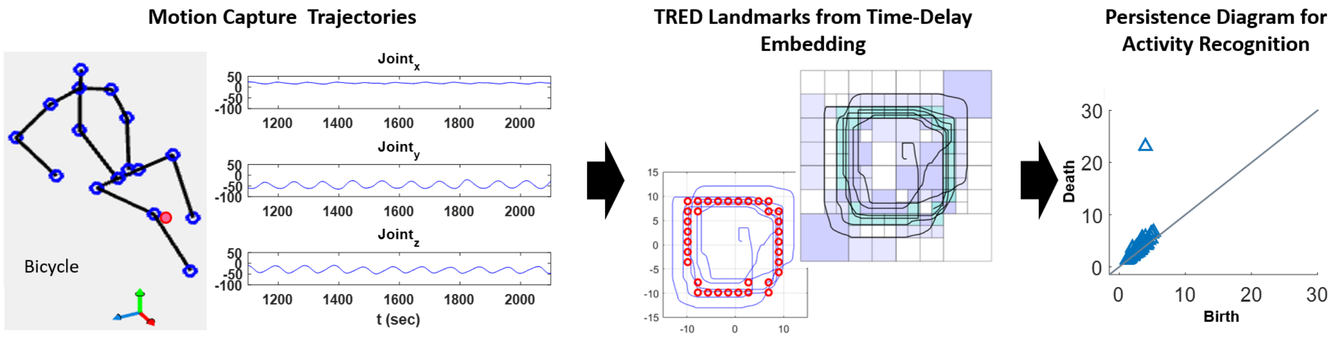

Fig. 4: TRED subsampling applied to activity recognition.

has a depth equal to the maximum depth of the quadtree and contains at least τ connected components. Figure 2 illustrates some examples for two trajectory sets. We note that the selected landmarks are within the corresponding level of the count function C_r^B.

V. Comparison

In this section we show a comparison of maxmin density based subsampling method (mentioned in section II) and our TRED subsampling method. A synthetic dataset (circle, square, ellipse, figure-8 and peanut shapes), were created to test our approach. Harmonic noise was added perpendicularly to the tangent of the curve at each data point. The harmonic noise was bounded by a magnitude of 0.1. Figure 2 shows some examples for these trajectory sets and the outcome of TRED representation. For these examples, our approach selected similar landmarks as the maxmin approach, with the advantage of faster performance.

In Figure 3 (a), we show an example of a transition stream from figure-8 to an ellipse. The figure-8 has one loop and the ellipse has four loops, so we consider the first trajectory as an outlier. The maxmin method tends to select evenly spaced points but is sensitive to outlier outliers, hence a pre-processing step to remove outliers that uses a K-nearest neighbor density estimation [13], was proposed before the subsampling method. The maxmin density based subsampling method picks samples based on density of the points and hence it ends up picking up samples from the outlier trajectory as well (see 3 (b)). On the other hand our TRED subsampling method picks samples based on trajectory densities (see 3 (c)), hence samples are picked up only from the ellipse.

We use bottleneck distances [14] to compare the dimension 1 persistence diagrams (PD_1), between the groundtruth (ellipse shape in this case) and subsampled point clouds obtained after maxmin and TRED method. We are interested in dimension 1 which gives us information about holes. In the maxmin method, since the samples are chosen from the outlier, the structure of the ellipse is lost which is seen in PD_1, a lot of holes appear and disappear near the diagonal but there are no persistent features. Whereas, in the TRED method we see two holes: one disappears quickly and the other one captures the structure of the ellipse yielding a diagram similar to the groundtruth.

VI. Application to Activity Recognition

TRED can be applied to an activity recognition pipeline. The authors in [9], use a windowed time-delay embedded [15] signal as an input to the maxmin subsampling strategy. Features from the persistence diagrams (obtained from subsampled point clouds) are used for classifying activities. Applying maxmin over each window can be computationally expensive $O(n^2)$, where n is the number of points in the trajectory set. Our proposed approach has a complexity of $O(n)$. In Figure 4 we show an example of how TRED can be applied to motion capture trajectories. The trajectories corresponding to the right leg joint (x, y and z axis) for bicycling activity are shown in Figure 4 (left). The TRED landmarks and quadtree for the time-delay embedded signal are shown in Figure 4 (middle), and the corresponding persistence diagram on the Figure 4 (right). In [9], features from the persistent diagram are used for activity recognition.

VII. Conclusion and Future Work

We propose a novel method for density based estimation of trajectories using TRED. It is not only fast but also robust in recovering the geometric structure as shown by the persistence diagram. We provide upper and lower bounds for describing the trajectory count in a region, and bounds on the minimum size of square region which correlates to the maximum curvature of a trajectory. Our method can be easily extended to higher dimensional spaces.

This work is able to be further extended to not only consider fixed trajectories of data, but consider instead the real-time processing of continuous stream using a fixed window size. This is particularly useful for applications such as the activity recognition framework discussed in section VI in which our representation can be computed and updated dynamically as new observations become available. The landmarks from our approach can then be used to extract geometric structure for activity classification. For this dynamic case, we will consider extensions to the presented algorithm as part of our future work. Adding a merge capability would maintain a concise data structure in that if data leaves a certain region, the bins may be merged to the single parent bin so that bare sections do not contain unneeded resolution. Merging would preserve the guarantees presented above and add the guarantee of a concise and dynamic data structure that can be easily updated.

REFERENCES

[1] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang, "Trajectory clustering: A partition-and-group framework," in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, 2007, SIGMOD '07, pp. 593–604, ACM.

[2] Y. Zhang and D. Pi, "A trajectory clustering algorithm based on symmetric neighborhood," in *2009 WRI World Congress on Computer Science and Information Engineering*, March 2009, vol. 3, pp. 640–645.

[3] Zhenhui Li, Jae-Gil Lee, Xiaolei Li, and Jiawei Han, "Incremental clustering for trajectories," in *Proceedings of the 15th International Conference on Database Systems for Advanced Applications - Volume Part II*, Berlin, Heidelberg, 2010, DASFAA'10, pp. 32–46, Springer-Verlag.

[4] Daniel Chen, Leonidas J. Guibas, John Hershberger, and Jian Sun, "Road network reconstruction for organizing paths," in *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, Philadelphia, PA, USA, 2010, SODA '10, pp. 1309–1320, Society for Industrial and Applied Mathematics.

[5] Mridul Aanjaneya, Frederic Chazal, Daniel Chen, Marc Glisse, Leonidas J. Guibas, and Dmitriy Morozov, "Metric graph reconstruction from noisy data," in *Proceedings of the Twenty-seventh Annual Symposium on Computational Geometry*, New York, NY, USA, 2011, SoCG '11, pp. 37–46, ACM.

[6] Vin De Silva and Gunnar E Carlsson, "Topological estimation using witness complexes.," *SPBG*, vol. 4, pp. 157–166, 2004.

[7] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. 1996, KDD'96, pp. 226–231, AAAI Press.

[8] Q. Ge and E. Lobaton, "Consensus-based image segmentation via topological persistence," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2016, pp. 1050–1057.

[9] A. Dirafzoon, N. Lokare, and E. Lobaton, "Action classification from motion capture data using topological data analysis," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec 2016, pp. 1260–1264.

[10] Herbert Edelsbrunner and John Harer, "Persistent homology-a survey," *Contemporary mathematics*, vol. 453, pp. 257–282, 2008.

[11] H. Edelsbrunner, D. Letscher, and A. Zomorodian, "Topological persistence and simplification," in *Annual Symposium on Foundations of Computer Science*, 2000.

[12] Afra Zomorodian and Gunnar Carlsson, "Computing persistent homology," *Proceedings of the twentieth annual symposium on Computational geometry - SoCG '04*, p. 347, 2004.

[13] Gunnar Carlsson, Tigran Ishkhanov, Vin de Silva, and Afra Zomorodian, "On the local behavior of spaces of natural images," *International Journal of Computer Vision*, vol. 76, no. 1, pp. 1–12, 2008.

[14] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer, "Stability of persistence diagrams," *Discrete & Computational Geometry*, vol. 37, no. 1, pp. 103–120, 2007.

[15] Floris Takens, "Detecting strange attractors in turbulence," in *Dynamical systems and turbulence, Warwick 1980*, pp. 366–381. Springer, 1981.