

Critical Points to Determine Persistence Homology

Charmin Asirimath, Jayampathy Ratnayake, Chathuranga Weeraddana, and Nandana Rajatheva

Abstract—Computation of simplicial complexes of a large point cloud often relies on extracting a sample, to reduce the associated computational burden. The sampling of the point cloud should minimally mutilate the features of the underlying object to enable effective “feature extraction” that lies at the center of modern data analysis techniques, e.g., machine learning. The study considers sampling critical points of a Morse function associated with a point cloud, to approximate the Vietoris-Rips complex and to compute persistence homology. The effectiveness of the approach is compared with the farthest point sampling (FPS), in the context of two classification problems. The empirical results suggest that sampling critical points of the Morse function can be more effective than FPS when determining the persistence homology for the cases where the critical points play a decisive role.

Index Terms—Critical points, Face classification, Morse function, Persistence homology

I. INTRODUCTION

In recent years data sets have grown in size and dimension with the proliferation of advanced data acquisition techniques. We have been able to use such data meaningfully not only because the computation power has increased to match the size, but also due to the paradigm shift in data analysis techniques that handle such data. A prime example is machine learning (ML). As a result new applications and techniques are emerging more frequently than ever before. Examples include object classification with applications in medicine (e.g., brain image analysis) and security (e.g., face classification) [1].

In many applications, items in data sets are higher dimensional “point clouds” on which the computations are done to represent them in a suitably identified feature space. Even though, in certain cases the feature space is easily identifiable, in many other cases identifying a feature space could be a less obvious task. Making the matters less trivial, even when there is an obvious set of features that realize a data set as point cloud, “Whether those features enable an accurate and practical representation of the data for the given task?” may not be fully answered.

For example, the number of dimensions required to represent an object in the data set in a straightforward way may be in the order of thousands or even millions (e.g., DNA sequence). Not only the computational and storage cost of such high dimensional data is a burden, but also many fundamental assumptions in ML algorithms could fail in high dimensions.

Charmin Asirimath (Charmin.PingamageDon@oulu.fi) is with the Center for Machine Vision and Signal Analysis, University of Oulu, Finland.

Jayampathy Ratnayake (jratnaya@maths.cmb.ac.lk) is with the Department of Mathematics, University of Colombo, Sri Lanka.

Chathuranga Weeraddana (chathurangaw@uom.lk) is with the Department of Electronic and Telecommunication Engineering, University of Moratuwa, Sri Lanka.

Nandana Rajatheva (nandana.rajatheva@oulu.fi) is with the Centre for Wireless Communications, University of Oulu, Finland.

Thus effective ways of *down-sampling* an original point cloud or representing data in a low dimensional point cloud is necessary [2]–[4].

However, fast, computable, and meaningful feature extraction methods, if they exist and could be identified, provide an efficient comparison of data. Thus, the problem of “feature extraction” is an important step, as it may be less trivial. As a result, algorithms that enable extracting meaningful features lie at the center of modern data analysis techniques and research (e.g., ML). For example, the vector of grayscale values of each pixel represents each image in a grayscale digital image data set as a point in a high dimensional Euclidean space. However, different features of the image may represent the images in a space that is more meaningful for a specified purpose [5]–[9].

The size of the data could be a burden for the task of feature extraction itself, making sampling inevitable. However, the sampling should be done in such a way that the resulting point cloud is minimally mutilating the features of the underlying object.

To summarize, the following key challenges in the algorithms aforementioned could be identified:

- 1) Feature extraction for efficient representation of data,
- 2) Feature extraction for efficient comparison,
- 3) Efficient sampling techniques which could facilitate feature extraction.

This paper is centered around the idea of extracting persistence diagrams [5] as topological features, originated from the critical points of a point cloud. The importance of topological methods in image and signal processing applications have been emphasized extensively in literature [10]–[19].

The main motivation for this work is to answer the question “How one can devise a sampling method that can support and improve such feature extraction process?”. This paper assumes that each data point in the data set itself can be understood as a point cloud equipped with a real valued function defined on it. For example, in a database of grayscale images, each image itself can be thought of as a point cloud in 2-dimensional space, with the function as the grayscale value of each pixel.

A. Our Contribution

We propose a sampling mechanism based on extracting critical points of a “Morse function” [20] defined on a point cloud. The effectiveness of the sampling is empirically tested by using two classification problems.

The proposed method is based on the idea that *by knowing the critical points (up to some persistent level) of a point cloud with a Morse function f , we should be able to efficiently extract topological features, such as persistent homology, that are closely associated with f* . Thus, we consider *sampling critical points* from an underlying point cloud equipped with

a Morse function, enabled by the computation of the *Morse-Smale (MS) Complex* [21], [22], which is used to compute topological features; specifically, persistence diagrams. Henceforth, we refer to the proposed sampling as *MS sampling*.

The approach is compared with classic farthest point sampling (FPS). When compared with FPS, in addition to the potential advantage of considering critical points, there are several other advantages of the proposed method: 1) MS complex can be computed in a parallel framework and can attain acceptable time complexities. 2) The proposed method considers an additional structure, a Morse function, for the sampling. 3) Witness complex [23] computes persistent homology efficiently by building a simplified complex; however, relies on providing a dependable set of “landmark points”. Critical points can be an effective representation of landmark points, as opposed to manually or randomly selected set of points.

We apply the proposed method for two classification problems: classifying 3-D objects consisting of 7 categories and classifying human faces according to ethnicity. Numerical experiments suggest that the proposed mechanism can achieve efficient sampling as compared to FPS for the purpose of computing persistence homology.

B. Organization

The rest of this paper is organized as follows. Section II skims through previous work and relevant concepts in computational topology. Section III presents the methodology we propose. Applications of the proposed algorithm are presented in Section V, followed by conclusions in Section VI.

II. BACKGROUND

In this section, we outline the topological concepts and tools that are used in this paper. For more details, we refer the reader to [24].

Given a point cloud S , a data structure designated as a *complex* is constructed. There are many ways of generating complexes; however, this work focuses on Vietoris-Rips (VR) complex [24, § III]. We denote the VR complex associated with *persistence* level r by $\text{VR}_S(r)$.

Such a complex gives rise to *homology groups*, one for each non-negative integer p , denoted by $H_p(S)$. For an increasing sequence of persistence levels, $r_0 < r_1 < \dots < r_n$ there exists a sequence of complexes. For example, in the case of VR complex, we have $\text{VR}_S(r_0) \subseteq \text{VR}_S(r_1) \subseteq \dots \text{VR}_S(r_n)$. Such a sequence is designated as a *filtration*.

A filtration determines a sequence of homology groups. Moreover, inclusions in the filtration give rise to maps on the homology. The *persistent homology groups* are computed based on these maps. Under these maps new elements in the homology groups can be “born” and existing ones can “die”. These births and the deaths of homology classes are then encoded into *persistence diagrams*.

Persistence diagrams [24, p. 181] are multi-sets of points in the extended plane $\mathbb{R} \cup \{\infty, -\infty\}$, one for each dimension p . In particular, we draw points at (r_i, r_j) with multiplicity μ_p^{ij} , where μ_p^{ij} is the the number of p -dimensional homology

classes born at persistence level r_i and dying at level r_j . Persistence diagrams are the topological signatures used throughout this paper to compare point clouds.

One can measure the dissimilarity between two persistence diagrams by using appropriately defined metrics [24, § VIII.2]. For example, this paper, as in [9], uses *Wasserstein* distance with parameter q , defined as

$$W_q(D_1, D_2) = \left[\inf_{\eta: D_1 \rightarrow D_2} \sum_{x \in D_1} \|x - \eta(x)\|_\infty^q \right]^{1/q} \quad (1)$$

to compare two persistence diagrams D_1 and D_2 , where η ranges over all bijections from D_1 to D_2 [24].

Let \mathbb{M} be a manifold. A smooth function $f : \mathbb{M} \mapsto \mathbb{R}$ is a *Morse function*, if all critical points are non-degenerate. One can think of f as a height function on \mathbb{M} . We refer the reader to [20] for more details. This paper relies on such a function defined on the point cloud S . Computation of the MS complex up to a given a persistence level was considered in [25], on which we rely on, to extract critical points.

III. METHODOLOGY

The proposed methodology is outlined in Algorithm 1.

Algorithm 1

Input:

- $X = \{X_1, X_2, \dots, X_n\}$, where each $X_i \in X$ itself is a point cloud with a distance metric d_i .
- $f_i : X_i \rightarrow \mathbb{R}$, a Morse function whose critical points are to be sampled.
- Persistence level r .
- q , parameter to Wasserstein distance.
- D , the maximum dimension of the persistence homology computed.

Steps:

- 1) For each $X_i \in X$
 - a) Extract the set X_i^r of critical points of f_i from X_i using MS algorithm with persistence level r .
 - b) For each $k \in \{0, 1, \dots, D\}$, compute the k -dimensional persistence diagrams $D_k(X_i^r)$.
 - 2) For each pair of X_i^r, X_j^r
 - a) Compute the distance $c(X_i, X_j)$ given by
$$c(X_i, X_j) = \max_k \{W_q(D_k(X_i^r), D_k(X_j^r))\}. \quad (2)$$
 - b) Form the distance matrix \mathbf{M} whose (i, j) entry is $c(X_i, X_j)$.
 - 3) Classify X based on \mathbf{M} .
-

In the sequel, we briefly discuss the steps of Algorithm 1. We assume that the data set $X = \{X_1, X_2, \dots, X_n\}$ consists of n points, where each X_i itself is a point cloud. We also assume each X_i to be a subset of a metric space with metric d_i and equipped with a function $f_i : X_i \rightarrow \mathbb{R}$.

A. Sampling: Step (1a)

Main goal of the sampling is to extract a well represented set of points X_i^r which can efficiently approximate the underlying space of X_i . The MS algorithm [25, Algorithm 1] is used with persistence level r to obtain X_i^r . For an illustration, see Fig. 3-(a) and (b).

B. Persistence Diagrams: Step (1b)

Persistence homology in dimensions $k = 0, 1, \dots, D$ is computed and the associated persistence diagrams are designated by $D_k(X_i^r)$. The value of K may be chosen based on the application.

C. Metricizing X for Classification: Step (2) and (3)

In this step, Wasserstein distances W_q between persistence diagrams obtained in Step (1b) are computed. These distances are then used to metricize X based on (2). From this a distance matrix M is computed to be used in a preferred classification algorithm.

IV. COMPUTATIONAL COMPLEXITY

The time complexity of FPS is $O(n \log n)$, where n is the number of points in the point cloud [2]–[4]. In the case of MS sampling, the critical points of the Morse function defined on a point cloud can be obtained by using an approximate discrete Morse-Smale (DMS) complex in $O(n \log n)$ [26]. If the point cloud is in 3-D, the computations can be further efficient. More specifically, for 3-D point clouds, the critical points of the Morse function can be obtained in $O(cn)$, where c is an upper bound on the number of critical points [27]. Note that the results of [26] and [27] relies on that the number of critical points of the Morse function is bounded. In general, authors of [25] show that the critical points of the Morse function can be computed in $O(n^2)$. Developments in parallel frameworks for computing DMS complexes with better time complexities are discussed in [28]–[31].

V. APPLICATIONS

In this section, we test the proposed Algorithm 1 in a shape classification problem and a face classification problem. The goal is essentially to compare and contrast the potentials of MS sampling [§ III-A] and commonly used FPS. To overcome the limitations in the numerosity of chosen data sets, we have used 6-fold cross-validation¹ in both applications to estimate the accuracy of the classifiers. The cross-validation estimate of the classification error is defined by [32, sec. 7.10]

$$CV = \frac{1}{n} \sum_{i=1}^n I_i, \quad (3)$$

where I_i the indicator function of miss-classifying the i data cloud and n is the number of objects. We compute the accuracy of the classifier by $1 - CV$.

¹In general, 5 or 10-fold cross-validation are recommended as a good compromise [32, sec. 7.10]; however, we use 6 as it divides both data sets in to equal size bins.

Training and Testing Phase: In both classification problems, the corresponding data set X is divided randomly into 6 considered partitions. We denote the partitions by B_j and the complement of B_j by A_j . For each $j \in \{1, \dots, 6\}$, we run Algorithm 1 with A_j as the input X to obtain the corresponding distance matrix M_j . In the training phase, we consider k -nearest neighbors (k -NN) classification method with $k = 1, 2, 3$, and test the resulting classifier on B_j . The accuracy $1 - CV$ is computed as discussed above.

All numerical simulations were carried out for three persistence levels, i.e., $r = 3$, with $q = 1$, and $D = 2$. The proposed algorithm is compared with an algorithm which replaces MS sampling by FPS, but with the same k in the nearest neighbour classifier. While the number of points returned by MS sampling can vary, FPS relies on specifying it. Therefore, for a fair comparison, in the case of FPS the average of the number of points returned by MS sampling over all images is used. Thus, both algorithms on average use the same number of points.

A. Shape Classification

1) *Preparing the Data Set:* We consider a publicly available database of 3-D shapes [33]. This database comprise of 66 shapes from seven different classes: camel, cat, elephant, face, flaming, head, and horse. Each class contains several different poses of the same shape.

The data set $X = \{X_1, X_2, \dots, X_{66}\}$, where $n = 66$, is such that each $X_i \subseteq \mathbb{R}^3$, corresponding some 3-D point cloud of a shape, see Fig. 1. For all $i \in \{1, \dots, 66\}$, the Morse function $f_i : X_i \rightarrow \mathbb{R}$ is taken to be $f_i(x, y, z) = z$, that simply assigns the height to each point $(x, y, z) \in X_i$.

2) *Numerical Results and Discussion:* Numerical simulations were carried out for the persistence levels: $r = 1.00, 0.99$, and 0.98 . For each r above, the corresponding average number of critical points from MS sampling are 66, 18, and 13, respectively, per shape. Fig. 1(a) and (b) show representations of MS sampling applied to a data point X_i , at persistent levels $r = 0.99$ and $r = 0.98$, yielding 16 points and 11 points, respectively. Similarly, Fig. 1(c) and (d) show the FPS sampling applied to the same data point. Fig. 1 shows that the critical points X_i^r from MS sampling is very different to those of FPS sampling.

Fig. 2 shows the performance of the k -NN classifiers for $k = 1, 2, 3$, with MS sampling and FPS sampling. In particular, the solid circles represent the accuracy level, $1 - CV$ [compare with (3)], of each classifier. The individual accuracy levels of the classifiers designed with MS sampling and FPS are also depicted by using associated box-plots to get an idea of the consistency of the classifiers within the 6-fold cross validation framework. The accuracy $1 - CV$ of the classifiers based on MS sampling outperform that of the FPS based classifiers in all considered scenarios except in the case of $r = 0.99$ with 2-NN. When the number of sampled points on average is 66, (i.e., $r = 1$ for MS sampling), the performance of MS sampling and FPS sampling is comparable, see Fig. 2(a). However, when the number of points on average is smaller [see Fig. 2(b)-(c)], the classifiers based on FPS sampling shows

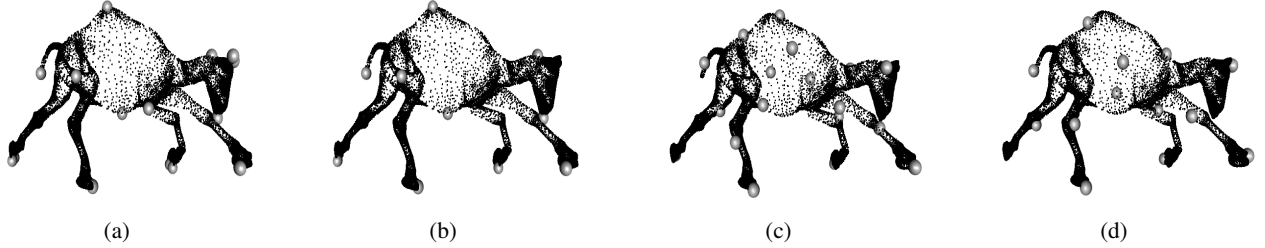


Fig. 1: MS Sampling with: (a) $r = 0.99$, (b) $r = 0.98$. FPS with (c) 18 points, (d) 13 points. Image courtesy of [33].

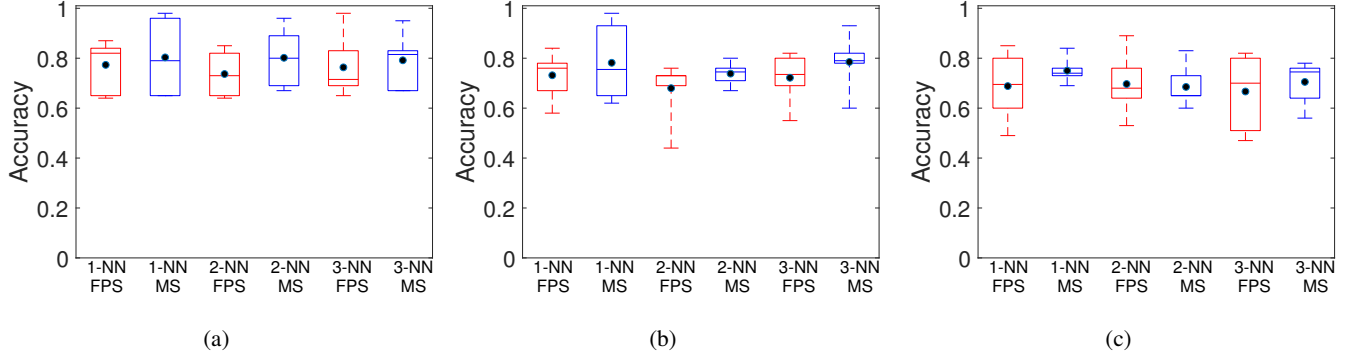


Fig. 2: Shape data set simulation results from 6-fold cross validations: (a) MS sampling with $r = 1$ and FPS with 66 points. (b) MS sampling with $r = 0.99$ and FPS with 18 points. (c) MS sampling with $r = 0.98$ and FPS with 13 points.

higher variance, whereas those based on MS sampling show low variance and consistency even with reduced sample size. For example, in these scenarios, at least 7 out of 36 (i.e., 19.4%) classifiers designed with FPS yield accuracy levels below 0.6. In contrast, only one out of 36 classifiers based on MS sampling yields an accuracy level below 0.6 that amounts to a 2.8%. Results further show that the worst-case accuracy level of the classifiers based on FPS can go even below 0.5 when the number of sampled points are smaller, see Fig. 2(b), 2-NN case and Fig. 2(c), 1 and 3-NN cases. In contrast, classifiers based on MS sampling, even with a smaller number of sampled points report no accuracy levels below 0.5, see Fig. 2(b)-(c). The above, together with the interquartile ranges of the box plots, show that the classifiers designed with MS sampling is more consistent than those with FPS.

B. Face Simulation Results

1) *Preparing the Data Set:* We consider a database of face images, which consist of 90 face images arbitrarily chosen from the on-line face database FERET [34]. As a preprocessing step, we resize each image into 100 pixel \times 150 pixel grayscale. The data set $X = \{X_1, X_2, \dots, X_{90}\}$ is such that each X_i corresponds to a 2-D grid $\mathbb{M} = \{1, 2, \dots, 100\} \times \{1, 2, \dots, 150\}$ and a function $f_i : \mathbb{M} \rightarrow \mathbb{G}$, where $\mathbb{G} = \{0, 1, \dots, 255\}$ and for each pixel (u, v) , $f_i(u, v)$ is the grayscale value. Thus;

$$X_i = \{(u, v) \in \mathbb{M} \mid f_i(u, v) < 255\}.$$

The Morse function on X_i is chosen to be f_i .

Note that the data set X is not labelled. Therefore, as a benchmark, we consider the following classifier which is experimentally devised. The goal is to first split the data set X into two groups, namely Asians and non-Asians, where the grouping was conducted by combining the results of N independent experiments². In particular, the outcome of j -th experiment is a labeling $y_i^{(j)} \in \{-1, 1\}$ associated with X_i , $i = 1, \dots, 90$, which is performed by a personnel based on their decision. The labels -1 and $+1$ denote Asians and non-Asians, respectively. The outcome of the N experiments are then combined by using the majority vote to get y_i , i.e., $y_i = \text{sign} \left\{ \sum_{n=1}^N y_i^{(n)} \right\}$. The resulting y_i , together with X_i , $i = 1, \dots, 90$ is used in the training and testing phase of our classifiers. In this particular case we use $N = 3$.

2) *Numerical Results and Discussion:* Fig. 3(a) and (b) show representations of MS sampling applied to a data point X_i in X , at persistent levels $r = 0.87$ and $r = 0.6$, yielding 160 points and 95 points, respectively. Note that the points in X_i^r are not uniformly distributed in Fig. 3(a) and (b); however, the points are concentrated at “landmark points” of X_i , which are apparently crucial to build a rich representation of the original X_i with VR complexes. However, note that FPS does not discriminate special features of the face when sampling, see Fig. 3(c) and (d).

²Such a classifier is to be designed with careful attention. However, the focus of this paper is extraneous to the development of the aforementioned classifier. Therefore, here we consider a simple experimentally driven classifier as a benchmark based on what we compare MS sampling and FPS sampling.

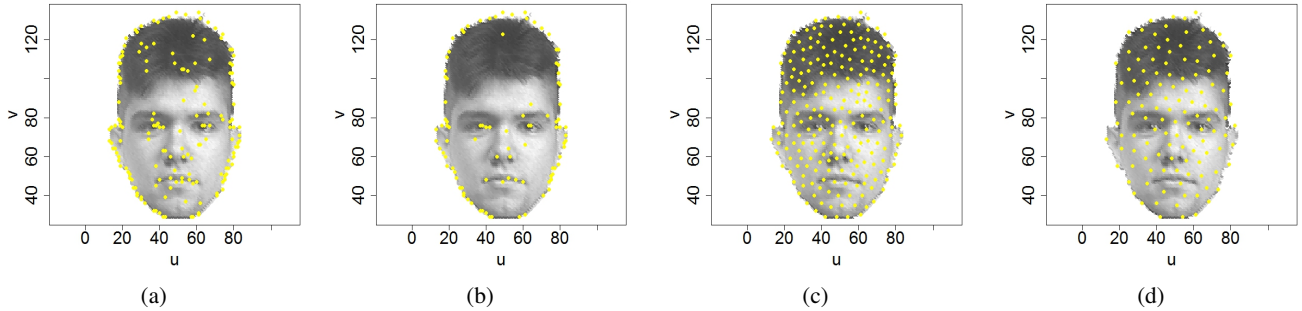


Fig. 3: MS Sampling with: (a) $r = 0.87$, (b) $r = 0.6$. FPS with (c) 220 points, (d) 119 points. Image courtesy of FERET [34].

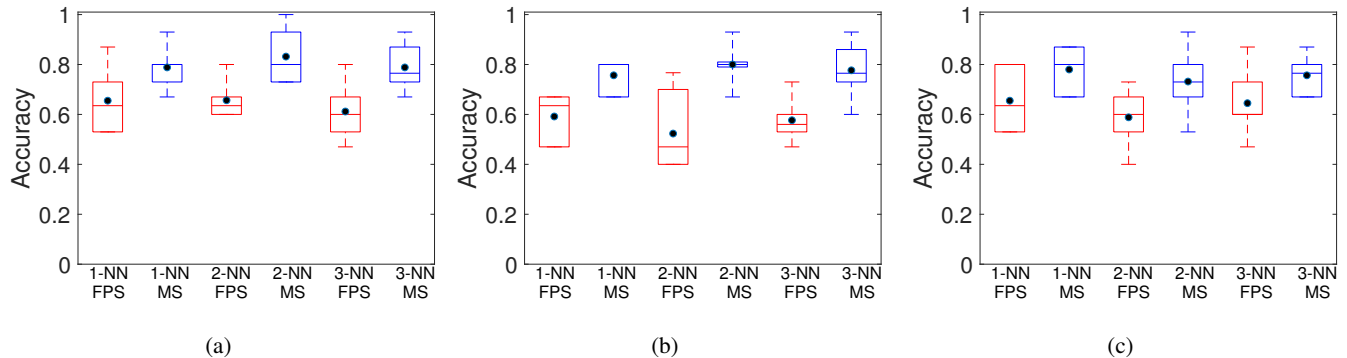


Fig. 4: Face data set simulation results from 6-fold cross validations: (a) MS sampling with $r = 0.87$ and FPS with 220 points. (b) MS sampling with $r = 0.76$ and FPS with 156 points. (c) MS sampling with $r = 0.6$ and FPS with 119 points.

Fig. 4, shows the performance of the k -NN classifiers for $k = 1, 2, 3$, with MS sampling and FPS sampling applied to the face data set X . Similar to Fig. 2, the accuracy levels of each classifiers is indicated by solid circles. Moreover, the individual accuracy levels of the classifiers pertaining to 6-fold cross validation are presented by using box-plots. In general, the observations are consistent with those of the shape classification [Fig. 2]. The accuracy levels of all the classifiers based on MS sampling outperform those based on FPS. Moreover, when the number of sampling points on average is smaller, the accuracy levels pertaining to the FPS based classifiers become more deteriorated. For example, in certain cases, the worst-case accuracy levels become on the order of 0.4, see Fig. 4(b), 2-NN case and Fig. 4(c), 2-NN case. However, the performance of the classifiers based on MS sampling remains nearly unchanged, irrespective of the number of sampling points used on average. Moreover, the worst-case accuracy levels of the classifiers based on MS sampling have never gone below 0.5. Thus, even in the case of face classification, the MS sampling based classifiers show more consistent properties than those based on FPS.

VI. CONCLUSION

This paper has considered using a sample of critical points of a Morse function f with respect to a given persistence level to compute the persistence homology of the underlying space of a point cloud. It has been empirically shown that,

compared to farthest point sampling (FPS), Morse Smale (MS) sampling can better aid the computation of the persistence homology. This idea was tested using two applications: a shape classification problem and a face classification problem. In both applications, MS sampling based computation of the persistence homology performed better than that with FPS. It has been observed that MS based computation has comparable, or even better, performance compared to the computation that uses FPS sampling with as twice as many points. In other words, MS sampling based computation of persistence homology is less sensitive to a decrease in the sample size and is still capable of yielding consistent results.

In particular, the results suggest that the proposed method is more effective than FPS for the cases where the critical points of f play a decisive role in determining persistence homology. However, the theoretical substantiation of the aforementioned statement is still an open problem.

Several advantages of the method, over other sampling methods such as FPS were observed for the purpose of computing persistence homology. For example, a different approach than using VR complex is the use of *witness complex*, which reduces the computational burden. Sampling of critical points can be used as an input of landmark points to the witness complex. Thus, the advantage of the proposed method in the computation of the witness complex using critical points as landmark points is worth investigating. Another future direction is to consider the adaptability of the proposed

approach to *discrete* MS complexes, which can be computed more efficiently.

REFERENCES

- [1] S. Z. Li and A. K. Jain, *Handbook of Face Recognition*. New York: Springer, 2011.
- [2] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE Trans. Image Processing*, vol. 6, no. 9, p. 13051315, 1997.
- [3] J. A. Sethian, "Fast marching methods," *SIAM Review*, vol. 41, no. 2, pp. 199–235, 1999.
- [4] C. Moenning and N. A. Dodgson, "Fast marching farthest point sampling for implicit surfaces and point clouds," University of Cambridge, Computer Laboratory, Tech. Rep. UCAM-CL-TR-565, 2003.
- [5] G. Carlsson, "Topology and data," *Bulletin (New Series) of the American Mathematical Society*, vol. 46, no. 2, pp. 255–308, April 2009.
- [6] F. Mémoli and G. Sapiro, "Comparing point clouds," in *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*. New York, NY, USA: ACM, 2004, pp. 32–40.
- [7] F. Mémoli and G. Sapiro, "A theoretical and computational framework for isometry invariant recognition of point cloud data," *Found. Comput. Math.*, vol. 5, no. 3, p. 313347, 2005.
- [8] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Efficient computation of isometry-invariant distances between surfaces," *SIAM Journals on Scientific Computing*, vol. 28, no. 5, pp. 1812–1836, 2006.
- [9] F. Chazal, D. Cohen-Steiner, L. J. Guibas, F. Mémoli, and S. Y. Oudot, "Gromov-hausdorff stable signatures for shapes using persistence," *Computer Graphics Forum (proc. SGP 2009)*, 2009.
- [10] S. Emrani, T. Gentimis, and H. Krim, "Persistence homology of delay embeddings and its application to wheeze detection," *IEEE Signal Processing Letters*, vol. 21, pp. 459–463, April 2014.
- [11] M. Robinson and R. Ghrist, "Topological localization via signals of opportunity," *IEEE Transaction on Signal Processing*, vol. 60, no. 5, pp. 2362–2373, 2012.
- [12] H. Chintakunta and H. Krim, "Distributed localization of coverage holes using topological persistence," *IEEE Transaction on Signal Processing*, vol. 62, no. 10, pp. 2531–2541, May 2014.
- [13] M. Robinson, *Topological Signal Processing*. New York: Springer, 2014.
- [14] H. Krim, T. Gentimis, and H. Chintakunta, "Discovering the whole by the coarse: A topological paradigm for data analysis," *IEEE Signal Processing Magazine*, vol. 33, pp. 95–104, March 2016.
- [15] E. Lobaton, R. Vasudevan, R. Bajcsy, and S. S. Sastry, "A distributed topological camera network representation for tracking applications," *IEEE Transaction on Image Processing*, vol. 19, pp. 2516–2529, 2010.
- [16] S. Emrani, H. Chintakunta, and H. Krim, "Real time detection of harmonic structure: A case of topological signal analysis," in *Proc. of Int. Conf. Acoustic Speech and Signal Processing Techniques and Methods*, vol. 19, no. 3, Nov 2013, pp. 3445–3449.
- [17] J. Ernst, M. K. Singh, and V. Ramesh, "Discrete texture traces: Topological representation of geometric context," in *Proc. IEEE Conf. Computer Vision and Recognition*, vol. 19, no. 3, 2012, pp. 422–429.
- [18] A. C. Wilkerson, T. J. Moore, A. Swami, and H. Krim, "Simplifying the homology of networks via strong collapses," in *Proc. Conf. Acoustics, Speech, and Signal Processing*, 2013, pp. 5258–5262.
- [19] H. Wagner, P. Dlotko, and M. Mrozek, "Computational topology in text mining," in *Proc. Computational Topology in Image Context*, 2012, pp. 68–78.
- [20] J. Milnor, *Morse Theory*. Princeton University Press, 1963.
- [21] R. Forman, "Morse theory for cell complexes," *Advances in Mathematics*, vol. 134, p. 90145, 1998.
- [22] —, "A user's guide to discrete Morse theory," in *Seminaire Lotharingien de Combinatoire*, vol. B48c, pp. 1–35, 2002.
- [23] V. de Silva and G. Carlsson, "Topological estimation using witness complexes," *Eurographics Symposium on Point-Based Graphics*, 2004.
- [24] H. Edelsbrunner and J. Harer, *Computational Topology: An Introduction*, 3rd ed., ser. 10. New York: American Mathematical Society, 2010, vol. 4.
- [25] S. Gerber, O. Rbel, P.-T. Bremer, V. Pascucci, and R. T. Whitaker, "Morse smale regression," *Journal of Computational and Graphical Statistics*, vol. 22, no. 1, pp. 193–214, 2013.
- [26] A. Gyulassy, P. T. Bremer, B. Hamann, and V. Pascucci, "A practical approach to Morse-Smale complex computation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, p. 16191626, 2008.
- [27] D. Günther, J. Reininghaus, H. Wagner, and I. Hotz, "Efficient computation of 3D MorseSmale complexes and persistent homology using discrete Morse theory," *The Visual Computer*, vol. 28, no. 10, pp. 959–969, 2012.
- [28] V. Robins, P. J. Wood, and A. P. Sheppard, "Theory and algorithms for constructing discrete Morse complexes from grayscale digital images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1646–1658, 2011.
- [29] N. Shivashankar and V. Natarajan, "Parallel computation of 3D Morse-smale complexes," in *Eurographics Conference on Visualization*, vol. 31, no. 3, Vienna, Austria, June 2012.
- [30] A. Gyulassy, V. Pascucci, T. Peterka, and R. Ross, "The parallel computation of morse-smale complexes," in *IEEE International Parallel and Distributed Processing Symposium*, no. 484-495, Shanghai, China, May 2012.
- [31] A. Gyulassy, D. Günther, J. A. Levine, J. Tierny, and V. Pascucci, "Conforming Morse-Smale complexes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2595–2603, 2014.
- [32] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York: Springer, 2009.
- [33] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," *SIGGRAPH 04: ACM SIGGRAPH 2004 Papers*, pp. 399–405, 2004.
- [34] P. J. Phillips, P. J. Rauss, and S. Z. Der, "Ferret (face recognition technology) recognition algorithm development and test results," Army Res. Lab., Adelphi, MD, USA, Army Res., Tech. Rep. 995, 1996.