

Article

# Isometry Invariant Shape Recognition of Projectively Perturbed Point Clouds by the Mergegram Extending 0D Persistence <sup>†</sup>

Yury Elkin and Vitaliy Kurlin \* 

Materials Innovation Factory, Computer Science Department, University of Liverpool, Liverpool L69 3BX, UK; yura.elkin@gmail.com or Yury.Elkin@liverpool.ac.uk

\* Correspondence: vitaliy.kurlin@gmail.com or vitaliy.kurlin@liverpool.ac.uk

<sup>†</sup> This paper is an extended version of the shorter paper published in the 45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020), Prague, Czech Republic, 24–28 August 2020.

**Abstract:** Rigid shapes should be naturally compared up to rigid motion or isometry, which preserves all inter-point distances. The same rigid shape can be often represented by noisy point clouds of different sizes. Hence, the isometry shape recognition problem requires methods that are independent of a cloud size. This paper studies stable-under-noise isometry invariants for the recognition problem stated in the harder form when given clouds can be related by affine or projective transformations. The first contribution is the stability proof for the invariant mergegram, which completely determines a single-linkage dendrogram in general position. The second contribution is the experimental demonstration that the mergegram outperforms other invariants in recognizing isometry classes of point clouds extracted from perturbed shapes in images.



**Citation:** Elkin, Y.; Kurlin, V. Isometry Invariant Shape Recognition of Projectively Perturbed Point Clouds by the Mergegram Extending 0D Persistence. *Mathematics* **2021**, *9*, 2121. <https://doi.org/10.3390/math9172121>

Academic Editors: Rocio Gonzalez Diaz and Matthias Zeppelzauer

Received: 6 July 2021

Accepted: 25 August 2021

Published: 1 September 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** shape recognition; Topological Data Analysis; machine learning; computer vision

## 1. Introduction: Motivations, Shape Recognition Problem, and Overview of Results

Real-life objects are often represented by unstructured point clouds obtained by laser range scanning or by selecting salient or feature points in images [1]. Point clouds are easy to store and can be used as primitives for visualization [2]. The above advantages strongly motivate the problem of comparing and classifying unstructured point clouds.

Rigid objects are naturally studied up to rigid motion or isometry (including reflections), which is any map that preserves inter-point distances. The recognition of point clouds of the same number of points is practically solved by the histogram of all pairwise distances, which is a complete isometry invariant in general position [3].

Real shapes are often given in a distorted form because of noisy measurements, when points are perturbed, missed or accidentally added. One of the first approaches to recognize nearly identical point clouds  $A, B$  of different sizes in the same metric space, for example, in  $\mathbb{R}^m$ , is to use the *Hausdorff* distance [4]  $\text{HD}(A, B) = \min \epsilon \geq 0$  such that the first cloud  $A$  is covered by  $\epsilon$ -balls centered at all points of  $B$ , and vice versa.

However, we also need to take into account infinitely many potential isometries of the ambient space  $\mathbb{R}^m$ . The exact computation of  $\inf_f \text{HD}(f(A), B)$  minimized over isometries  $f$  of  $\mathbb{R}^m$  has a high polynomial complexity already for dimension  $m = 2$  [5]. An approximate algorithm is cubic in the number of points for  $m = 3$  [6].

This paper extends the 12-page conference version [7], which introduced the new invariant mergegram but did not prove its continuity under perturbations. In addition to the proof of continuity, another contribution is Theorem 2 showing how to reconstruct a dendrogram of single-linkage clustering from a mergegram in general position.

The practical novelty is the harder recognition problem including perturbations of isometries within wider classes of affine and projective maps motivated by computer vision applications. Indeed, different positions of cameras produce projectively equivalent images

of the same rigid shape. The new experiments in Section 6 extensively compared several approaches on 15,000 clouds obtained from real images; see examples in Figure 1.



**Figure 1.** Images from the dataset of mythical creatures at <http://tosca.cs.technion.ac.il> (accessed on 25 August 2021) [8,9].

**Problem 1** (isometry shape recognition under noise). *Find an isometry invariant of point clouds in  $\mathbb{R}^m$  that is (a) independent of a cloud size, (b) provably continuous under perturbations of a cloud, (c) computable in a near linear time in the size of a cloud, and (d) more efficient for recognizing isometry classes of clouds than past invariants.*

The key contributions are Theorems 2 and 3 and the experiments in Section 6 showing that the mergegram achieves a state-of-the-art recognition on substantially distorted images.

## 2. Related Work on Isometry Shape Recognition and Topological Data Analysis

For the isometry classification of clouds consisting of the same number of points, the easiest invariant is the distribution of all pairwise distances, whose completeness (or injectivity) was proved for all point clouds in general (non-singular) position in  $\mathbb{R}^m$  [3].

Fixed point clouds  $A, B \subset \mathbb{R}^m$  of different sizes can be pairwise compared by the Hausdorff distance [4]  $\text{HD}(A, B) = \max\{\sup_{p \in A} d_B(p), \sup_{q \in B} d_A(q)\}$ , where  $d_B(p) = \inf_{q \in B} d(p, q)$  is the (Euclidean or another) distance from a point  $p \in A$  to the cloud  $B$ .

The rigid shape recognition problem for non-fixed clouds  $A, B$  is harder because of infinitely many potential isometries that can match  $A, B$  exactly or approximately. Partial cases of this problem were studied for clouds representing surfaces [10] and when two clouds have a given isometric matching of one pair of points [11]. Shape Google by Ovsjanikov et al. practically extends these ideas to non-rigid shape recognition [12].

The most general framework for the isometry shape recognition of point cloud data was proposed by Mémoi and Sapiro [13]. They studied the Gromov-Hausdorff distance  $d_{GH}(A, B) = \inf_{f, g, M} \text{HD}(f(A), g(B))$  minimized over all isometric embeddings  $f : A \rightarrow M$  and  $g : B \rightarrow M$  of given point clouds into a metric space  $M$ . Since the above definitions involve even more minimizations over infinitely many maps and spaces, GH can be only approximated. The Farthest Point Sampling (FPS) has a quadratic complexity in the number of points (Reference [13], Section 3.6) and was successfully tested on small clouds.

The proposed invariant mergegram extends the 0-dimensional persistence in the area of Topological Data Analysis (TDA), which grew from the theory of size functions [14]. TDA views a point cloud  $A \subset \mathbb{R}^m$  not by fixing any distance threshold but across all scales  $s$ , for example, by blurring given points of  $A$  to balls of a variable radius  $s$ . The evolution of this growing union of balls is summarized by a persistence diagram, which is invariant under isometries of  $\mathbb{R}^m$ . TDA can be combined with machine learning and statistical tools due to stability under noise, which was first proved by Cohen-Steiner et al. [15] and then extended to a very general form by Chazal et al. [16].

In dimension 0 the persistence diagram  $\text{PD}(A)$  for distance-based filtrations of a point cloud  $A$  consists of the pairs  $(0, s) \in \mathbb{R}^2$ , where values of  $s$  are distance scales at which subsets of  $A$  merge by the single-linkage clustering. These scales equal half-lengths of edges in a Minimum Spanning Tree  $\text{MST}(A)$ . If distances between all points of  $A$  are known,  $\text{MST}(A)$  is a connected graph with the vertex set  $A$  and a minimal total length.

Representing a point cloud  $A$  by  $\text{PD}(A)$  loses a lot of geometry of  $A$ , but gains stability under perturbations, which can be expressed in the case of point clouds as  $\text{BD}(\text{PD}(A), \text{PD}(B)) \leq \text{HD}(A, B)$ . Here, the bottleneck distance  $\text{BD}$  between diagrams is defined as a minimum  $\epsilon \geq 0$  such that all pairs of  $\text{PD}(A)$  can be bijectively matched to

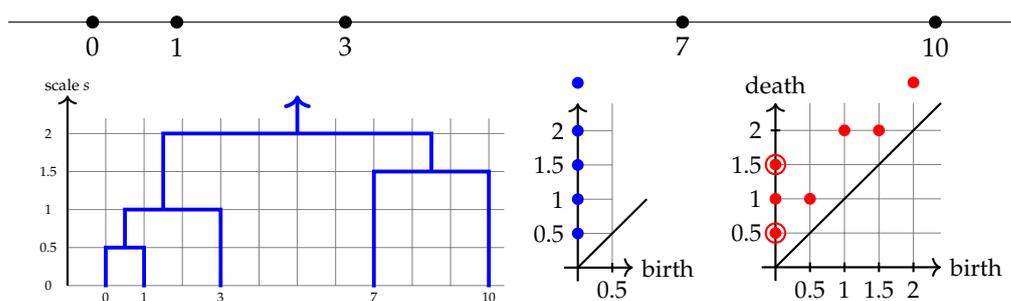
$\epsilon$ -close points of  $PD(B)$  or to diagonal pairs  $(s, s)$ , and vice versa. Here,  $\epsilon$ -closeness of pairs  $(a, c)$  and  $(b, d)$  in  $\mathbb{R}^2$  is measured in the distance  $L_\infty = \max\{|a - b|, |c - d|\}$ .

The *mergegram* extends  $PD(A)$  to a stronger invariant, whose stability under perturbations in the above sense is proved in Section 5 for the first time. The idea of a mergegram is related to the Reeb graph [17] or the merge tree [18] for the sublevel set filtration of a scalar function. The mergegram MG is defined at a more abstract level for any clustering dendrogram, which opens a possibility to extend a Homologically Persistent Skeleton (HoPeS) visualizing most persistent cycles in point clouds [19].

Since any persistence diagram and a mergegram are unordered collections of pairs, the experiments in Section 6 use the neural network PersLay [20], whose output is invariant under permutations of input points by design. PersLay extends the neural network DeepSet [21] and introduces new layers to accept as an input any diagram of unordered points. In other related work, deep learning was recently applied to outputs of hierarchical clustering [22–24] and to 0-dimensional persistence [25,26].

### 3. Single-Linkage Clustering and the Invariant Mergegram of a Dendrogram

**Example 1.** Figure 2 illustrates the key concepts before formal Definitions 1, 3 and 4 for the point cloud  $A = \{0, 1, 3, 7, 10\}$  in the real line  $\mathbb{R}$ . Imagine that we gradually blur original data points by growing disks of the same radius  $s$  around the given points.



**Figure 2.** Top: the 5-point cloud  $A = \{0, 1, 3, 7, 10\} \subset \mathbb{R}$ . Bottom: from left to right: single-linkage dendrogram  $\Delta_{SL}(A)$  from Definition 1, the 0D persistence diagram  $PD$  in Definition 6, the novel mergegram  $MG$  from Definition 4, where double circles show pairs of multiplicity 2.

The disks of the closest points 0, 1 start overlapping at the scale  $s = 0.5$  when these points merge into one cluster  $\{0, 1\}$ . This merger is shown by blue arcs joined at the node at  $s = 0.5$  in the single-linkage dendrogram; see the left picture at the bottom of Figure 2. The persistence diagram  $PD$  in the middle picture at the bottom of Figure 2 represents this merger by the pair  $(0, 0.5)$ , meaning that a singleton cluster of (say) point 1 was born at the scale  $s = 0$  and then died later at  $s = 0.5$  by merging into another cluster of point 0.

When clusters  $\{0, 1, 3\}$  and  $\{7, 10\}$  merge at  $s = 2$ , this merger was earlier encoded in the persistence diagram by the single pair  $(0, 2)$ , meaning that one cluster inherited from (say) point 7 was born at  $s = 0$  and died at  $s = 2$ . The new mergegram in the bottom right picture of Figure 2 represents the above merger by the following two pairs. The pair  $(1, 2)$  means that the cluster  $\{0, 1, 3\}$  is merging at the current scale  $s = 2$  and was previously formed at the smaller scale  $s = 1$ . The pair  $(1.5, 2)$  means that another cluster  $\{7, 10\}$  is merging at the scale  $s = 2$  and was previously formed at  $s = 1.5$ .

The 0D persistence diagram represents the cluster of the whole cloud  $A$  by the pair  $(0, +\infty)$  because  $A$  was inherited from a singleton cluster starting from  $s = 0$ . The mergegram represents the same cluster  $A$  by the pair  $(2, +\infty)$  because  $A$  was formed during the last merger of  $\{0, 1, 3\}$  and  $\{7, 10\}$  at  $s = 2$  and continues to live as  $s \rightarrow +\infty$ .

In the above dendrogram, every vertical arc going up from a scale  $b$  to  $d$  contributes one pair  $(b, d)$  to the mergegram. So, both singleton clusters  $\{7\}$ ,  $\{10\}$  merging at  $s = 1.5$  contribute one pair  $(0, 1.5)$  of multiplicity two, shown by two red circles in Figure 2.



example,  $\Delta_0^1(\{0\}) = \{0, 1\} = \Delta_0^1(\{1\})$  and  $\Delta_0^1(\{2\}) = \{2\}$ , so  $\Delta_0^1$  is a well-defined map from the partition  $\Delta(A; 0)$  of 3 singleton sets to  $\Delta(A; 1)$  but is not an identity.

At the scale  $s_0 = 0$  the merge sets  $\{0\}, \{1\}$  have life =  $[0, 1)$ , the merge set  $\{2\}$  has life =  $[0, 2)$ . At the scale  $s_1 = 1$  the single merge set  $\{0, 1\}$  has life =  $[1, 2)$ . At the scale  $s_2 = 2$  the single merge set  $\{0, 1, 2\}$  has life =  $[2, +\infty)$ . The first (Greek) letter in the word ‘dendrogram’ and a  $\Delta$ -shape of a typical tree motivate the notation  $\Delta$ .

Condition (Definition 3 (a)) says that a partition of a set  $X$  is trivial for all large scales. Condition (Definition 3 (b)) means that, if the scale  $s$  is increasing, any sets from a partition  $\Delta(s)$  can merge but cannot split into smaller sets. Condition (Definition 3 (c)) implies that there are only finitely many mergers, when two or more subsets of  $X$  merge into a larger merge set.

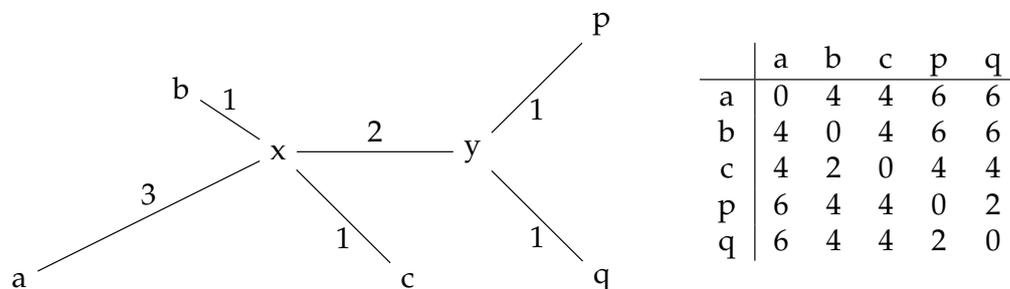
**Lemma 1** ([7], Lemma 3.3). *Given a metric space  $(X, d)$  and a finite set  $A \subset X$ , the single-linkage dendrogram  $\Delta_{SL}(X)$  from Definition 1 satisfies Definition 3.*

A mergegram represents life spans of merge sets by pairs (birth, death)  $\in \mathbb{R}^2$ .

**Definition 4** (mergegram  $MG(\Delta)$ ). *The mergegram of a dendrogram  $\Delta$  has the pair (birth, death)  $\in \mathbb{R}^2$  for each merge set  $B$  of  $\Delta$  with  $life(B) = [birth, death)$ . If any life interval appears  $k$  times, the pair (birth, death) has the multiplicity  $k$  in  $MG(\Delta)$ .*

If our input is a point cloud  $A$  in a metric space, then the mergegram  $MG(\Delta_{SL}(A))$  is an isometry invariant of  $A$  because  $\Delta_{SL}(A)$  depends only on inter-point distances. Though  $\Delta_{SL}(A)$  as any dendrogram is unstable under perturbations of points, the key advantage of  $MG(\Delta_{SL}(A))$  is its stability, which will be proved in Theorem 4.

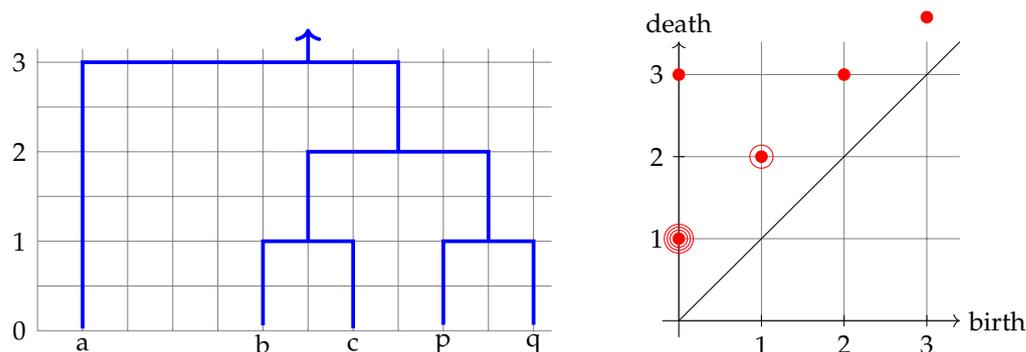
Figure 4 shows the metric space  $X = \{a, b, c, p, q\}$  with distances defined by the shortest path metric induced by the specified edge-lengths; see the distance matrix.



**Figure 4.** The set  $X = \{a, b, c, p, q\}$  has the distance matrix obtained by the shortest path metric.

The dendrogram  $\Delta$  in the first picture of Figure 5 leads to the mergegram as follows:

- each of the 1-point sets  $\{b\}, \{c\}, \{p\}, \{q\}$  has pair  $(0, 1)$ , so its multiplicity is 4;
- each of the merge sets  $\{b, c\}$  and  $\{p, q\}$  has the pair  $(1, 2)$ , so its multiplicity is 2;
- the singleton set  $\{a\}$  has the pair  $(0, 3)$ ; the merge set  $\{b, c, p, q\}$  has the pair  $(2, 3)$ ;
- the full set  $\{a, b, c, p, q\}$  continues to leave up to  $s = 3$ , hence having the pair  $(3, +\infty)$ .



**Figure 5.** Left: the dendrogram  $\Delta$  for the single linkage clustering of the set 5-point set  $X = \{a, b, c, p, q\}$  in Figure 4. Right: the mergegram  $MG(\Delta)$  with one pair  $(0,1)$  of multiplicity 4.

#### 4. Explicit Relations between 0-Dimensional Persistence and Mergegram

This section recalls the concept of persistence and then shows how any 0D persistence and dendrogram in general position can be reconstructed from a mergegram.

**Definition 5** (persistence module  $\mathbb{V}$ ). A persistence module  $\mathbb{V}$  over the real numbers  $\mathbb{R}$  is a collection of vector spaces  $V_t, t \in \mathbb{R}$  with linear maps  $v_s^t : V_s \rightarrow V_t, s \leq t$  such that  $v_t^t$  is the identity on  $V_t$ , and the composition is respected:  $v_s^t \circ v_r^s = v_r^t$  for any  $r \leq s \leq t$ .

The set of real numbers can be considered as a category  $\mathbb{R}$  in the following sense. The objects of  $\mathbb{R}$  are all real numbers. Any real numbers  $a \leq b$  define a single morphism  $a \rightarrow b$ . The composition of morphisms  $a \rightarrow b$  and  $b \rightarrow c$  is the morphism  $a \rightarrow c$ . In the language of category theory, a persistence module is a functor from  $\mathbb{R}$  to the category of vector spaces. A basic example of a persistence module  $\mathbb{V}$  is an interval module. An interval  $J$  between points  $p < q$  in  $\mathbb{R}$  can be one of the following types: closed  $[p, q]$ , open  $(p, q)$ , half-open or half-closed  $[p, q)$  and  $(p, q]$ , all encoded as follows:

$$[p^-, q^+] := [p, q], \quad [p^+, q^-] := (p, q), \quad [p^+, q^+] := (p, q], \quad [p^-, q^-] := [p, q).$$

The endpoints  $p, q$  can have the infinite values  $\pm\infty$ , but without superscripts.

**Example 2** (interval module  $\mathbb{I}(J)$ ). Let  $J \subset \mathbb{R}$  be an interval. The interval module  $\mathbb{I}(J)$  is the persistence module defined by the following vector spaces  $I_s$  and linear maps  $i_s^t : I_s \rightarrow I_t$

$$I_s = \begin{cases} \mathbb{Z}_2, & \text{for } s \in J, \\ 0, & \text{otherwise;} \end{cases} \quad i_s^t = \begin{cases} \text{id}, & \text{for } s, t \in J, \\ 0, & \text{otherwise} \end{cases} \quad \text{for any } s \leq t.$$

The direct sum  $\mathbb{W} = \mathbb{U} \oplus \mathbb{V}$  of persistence modules  $\mathbb{U}, \mathbb{V}$  is defined as the persistence module with the vector spaces  $W_s = U_s \oplus V_s$  and linear maps  $w_s^t = u_s^t \oplus v_s^t$ .

We illustrate the abstract concepts above by geometric constructions. Let  $f : X \rightarrow \mathbb{R}$  be a continuous function on a topological space. The sublevel sets  $X_s^f = f^{-1}((-\infty, s])$  form nested subspaces  $X_s^f \subset X_t^f$  for any  $s \leq t$ . The inclusions of the sublevel sets respect compositions similarly to a dendrogram  $\Delta$  in Definition 3. On a metric space  $X$  with a metric  $d : X \times X \rightarrow [0, +\infty)$ , a typical example of a function  $f : X \rightarrow \mathbb{R}$  is the distance  $d_A$  to a finite subset  $A \subset X$ . For any point  $p \in X$ , let  $d_A(p)$  be the distance from  $p$  to a closest point of  $A$ . For any  $r \geq 0$ , the preimage  $X_r^{d_A} = d_A^{-1}((-\infty, r]) = \{p \in X \mid d_A(p) \leq r\}$  is the union of closed balls with radius  $r$  and centers at all points  $q \in A$ . For example,  $X_0^{d_A} = d_A^{-1}((-\infty, 0]) = A$  and  $X_{+\infty}^{d_A} = d_A^{-1}(\mathbb{R}) = X$ .

Any continuous function  $f : X \rightarrow \mathbb{R}$  induces the inclusion  $X_s^f \subset X_r^f$  for any parameters  $s \leq r$ . Then, all sublevel sets  $X_s^f$  form a nested sequence of subspaces within  $X$ . The above

construction of a filtration  $\{X_s^f\}$  can be considered as a functor from  $\mathbb{R}$  to the category of topological spaces. Below, we discuss the simplest case of dimension 0.

**Example 3** (persistent homology). *For any topological space  $X$ , the 0-dimensional homology  $H_0(X)$  is the vector space (with coefficients  $\mathbb{Z}_2$ ) generated by all connected components of  $X$ . Let  $\{X_s\}$  be any filtration of nested spaces, e.g., sublevel sets  $X_s^f$  based on a continuous function  $f : X \rightarrow \mathbb{R}$ . The inclusions  $X_s \subset X_r$  for  $s \leq r$  induce the linear maps between homology groups  $H_0(X_s) \rightarrow H_0(X_r)$  and define the persistent homology  $\{H_0(X_s)\}$ , which satisfies the conditions of a persistence module from Definition 5.*

If  $X$  is a finite set of  $m$  points, then  $H_0(X)$  is the direct sum  $\mathbb{Z}_2^m$  of  $m$  copies of  $\mathbb{Z}_2$ .

The persistence modules that are decomposable as direct sums of interval modules can be described in a simple combinatorial way by persistence diagrams in  $\mathbb{R}^2$ .

**Definition 6** (persistence diagram  $\text{PD}(\mathbb{V})$ ). *Let a persistence module  $\mathbb{V}$  be decomposed as a direct sum of interval modules :  $\mathbb{V} \cong \bigoplus_{l \in L} \mathbb{I}(p_l^*, q_l^*)$ , where  $*$  is + or -. The persistence diagram  $\text{PD}(\mathbb{V})$  is the multiset  $\text{PD}(\mathbb{V}) = \{(p_l, q_l) \mid l \in L\} \setminus \{p = q\} \subset \mathbb{R}^2$ .*

The 0-dimensional persistence diagram of a topological space  $X$  with a continuous function  $f : X \rightarrow \mathbb{R}$  is denoted by  $\text{PD}\{H_0(X_s^f)\}$ . Lemma 2 will prove that the merge module  $M(\Delta)$  of any dendrogram  $\Delta$  is decomposable into interval modules. The mergegram  $\text{MG}(\Delta)$  will be interpreted as the persistence diagram of  $M(\Delta)$ .

The following result describes how the persistence diagram  $\text{PD}$  of the distance-based filtration of any point cloud  $A$  can be obtained from the mergegram  $\text{MG}(\Delta_{\text{SL}}(S))$ .

**Theorem 1** ([7], Theorem 5.3). *For any finite set  $A$  in a metric space  $(X, d)$ , let  $d_A : X \rightarrow \mathbb{R}$  be the distance to  $A$ . Let the mergegram  $\text{MG}(\Delta_{\text{SL}}(A))$  be a multiset  $\{(b_i, d_i)\}_{i=1}^k$ , where some pairs can be repeated. Then, the persistence diagram  $\text{PD}\{H_0(X_s^{d_A})\}$  is the difference of the multisets  $\{(0, d_i)\}_{i=1}^k - \{(0, b_i)\}_{i=1}^k$  containing each pair  $(0, s)$  exactly  $\#b - \#d$  times, where  $\#b$  is the number of births  $b_i = s$ , and  $\#d$  is the number of deaths  $d_i = s$ . All trivial pairs  $(0, 0)$  are ignored, and, alternatively, we take  $\{(0, d_i)\}_{i=1}^k$  only with  $d_i > 0$ .*

Theorem 1 is illustrated by Example 1, where  $A = \{0, 1, 3, 7, 10\}$  has the diagram  $\text{PD}(A) = \{(0, 0.5), (0, 1), (0, 1.5), (0, 2), (0, +\infty)\}$  obtained from the mergegram

$$\text{MG}(\Delta_{\text{SL}}(A)) = \{(0, 0.5), (0, 0.5), (0, 1), (0, 1.5), (0, 1.5), (0.5, 1), (1, 2), (1.5, 2), (2, +\infty)\}$$

as follows: one pair  $(0, 0.5) \in \text{PD}(A)$  comes from two deaths and one birth  $s = 0.5$  in  $\text{MG}(\Delta_{\text{SL}}(A))$ . Similarly, each of the pairs  $(0, 1), (0, 1.5), (0, 2) \in \text{PD}(A)$  comes from two deaths and one birth equal to the same scale  $s$ . The cloud  $B = \{0, 4, 6, 9, 10\} \subset \mathbb{R}$  in Reference [7] (Example 1.1) has exactly the same  $\text{PD}(B) = \text{PD}(A)$  but different  $\text{MG}(\Delta_{\text{SL}}(B)) \neq \text{MG}(\Delta_{\text{SL}}(A))$ . This example jointly with Theorem 1 justify that the mergegram is strictly stronger than 0D persistence as an isometry invariant of a point cloud.

The New Reconstruction Theorem 2 below can be contrasted with the weakness of 0D persistence  $\text{PD}\{H_0(X_s^{d_A})\}$  consisting of only pairs  $(0, s)$  whose finite deaths are half-lengths of edges in a Minimum Spanning Tree  $\text{MST}(A)$ . In Example 1, these scales  $s = 0.5, 1, 1.5, 2$  are insufficient to reconstruct the SL dendrogram in Figure 2. Such a unique reconstruction is possible by using the richer invariant mergegram as follows.

**Theorem 2** (from a mergegram to a dendrogram). *Let  $A$  be a finite point cloud in general position in the sense that all merge scales of  $A$  in a dendrogram  $\Delta$  from Definition 3 are different. Then, the dendrogram  $\Delta$  can be reconstructed from its mergegram  $\text{MG}(\Delta)$ , uniquely up to a permutation of nodes in  $\Delta$  at scale  $s = 0$ .*

**Proof.** Consider all merge scales one by one in the increasing order starting from the smallest. The general position implies that only two clusters merge at any merge scale. For any current scale  $s$ , the mergegram contains exactly two pairs  $(b_1, s)$  and  $(b_2, s)$ .

For a smallest merge scale  $s > 0$ , the births should be  $b_1 = b_2 = 0$ . We start drawing a dendrogram  $\Delta$  by merging any two points of  $A$  at this smallest scale  $s$ . To realize a merger at any larger  $s$ , we should select two clusters representing  $(b_1, s)$  and  $(b_2, s)$ .

If  $b_i = 0$ , then we take any of the unmerged points of  $A$ . If  $b_i > 0$ , then the already constructed dendrogram should contain a unique non-singleton cluster determined by the scale  $b_i \in (0, s)$ . Hence, at any merge scale  $s$ , we know how to select two clusters to merge. The only choice comes from choosing points of  $A$  or permuting notes of  $\Delta$ .  $\square$

Following the above proof of Theorem 2 for the cloud  $A = \{0, 1, 3, 7, 10\}$  in Example 1, the first two pairs  $(0, 0.5) \in \text{MG}(\Delta_{\text{SL}}(A))$  indicate that we should merge two points of  $A$  at  $s = 0.5$ . The scale  $s = 0.5$  uniquely determines this 2-point cluster.

The next two pairs  $(0, 1), (0.5, 1)$  mean that the above cluster born at  $s = 0.5$  should merge at  $s = 1$  with a singleton cluster (any free point of  $A$ ). The resulting 3-point cluster is uniquely determined by its merge scale  $s = 1$ . The further two pairs  $(0, 1.5), (0, 1.5)$  say that a new 2-point cluster is formed at  $s = 1.5$  by the two remaining points of  $A$ .

The final pairs  $(1, 2), (1.5, 2)$  tell us to merge at  $s = 2$  the two clusters formed earlier at  $s = 1$  and  $s = 1.5$ . The resulting dendrogram  $\Delta$  has the expected combinatorial structure as in Figure 2, though we can draw  $\Delta$  in another way by permuting points of  $A$ .

### 5. Stability of the Mergegram for Any Single-Linkage Dendrogram

This section fully proves the stability of a mergegram, which was stated in Reference [7] (Theorem 7.4), without proving key Lemmas 2 and 3. For simplicity, we consider vector spaces with coefficients only in  $\mathbb{Z}_2 = \{0, 1\}$ , which can be replaced by any field.

Definition 7 introduces homomorphisms between persistence modules, which are needed to state the stability of persistence diagrams  $\text{PD}\{H_0(X_s^f)\}$  under perturbations of a function  $f : X \rightarrow \mathbb{R}$ . This result will imply a stability for the mergegram  $\text{MG}(\Delta_{\text{SL}}(A))$  for the dendrogram  $\Delta_{\text{SL}}(A)$  of the single-linkage clustering of a set  $A \subset X$ .

**Definition 7** (a homomorphism between persistence modules). *Let  $\mathbb{U}$  and  $\mathbb{V}$  be persistent modules over  $\mathbb{R}$ . A homomorphism  $\mathbb{U} \rightarrow \mathbb{V}$  of a degree  $\delta \in \mathbb{R}$  consists of linear maps  $\phi_t : U_t \rightarrow V_{t+\delta}, t \in \mathbb{R}$ , such that the diagram below commutes for all  $s \leq t$ .*

$$\begin{array}{ccc}
 U_s & \xrightarrow{u_s^t} & U_t \\
 \phi_s \downarrow & & \downarrow \phi_t \\
 V_{s+\delta} & \xrightarrow{v_{s+\delta}^{t+\delta}} & V_{t+\delta}
 \end{array}$$

*Let  $\text{Hom}^\delta(\mathbb{U}, \mathbb{V})$  be all homomorphisms  $\mathbb{U} \rightarrow \mathbb{V}$  of degree  $\delta$ . Persistence modules  $\mathbb{U}, \mathbb{V}$  are isomorphic if there are inverse homomorphisms  $\mathbb{U} \rightarrow \mathbb{V} \rightarrow \mathbb{U}$  of degree 0.*

For a persistence module  $\mathbb{V}$  with maps  $v_s^t : V_s \rightarrow V_t$ , the simplest example of a homomorphism of a degree  $\delta \geq 0$  is  $1_{\mathbb{V}}^\delta : \mathbb{V} \rightarrow \mathbb{V}$  defined by the maps  $v_s^{s+\delta}, t \in \mathbb{R}$ . So,  $v_s^t$  defining the structure of  $\mathbb{V}$  shift all vector spaces  $V_s$  by the difference  $\delta = t - s$ .

Interleaved modules defined below algebraically generalize a geometric perturbation of a set  $X$  in terms of (the homology of) its sublevel sets  $X_s$ .

**Definition 8** (interleaving distance ID). *Persistence modules  $\mathbb{U}$  and  $\mathbb{V}$  are called  $\delta$ -interleaved if there are homomorphisms  $\phi \in \text{Hom}^\delta(\mathbb{U}, \mathbb{V})$  and  $\psi \in \text{Hom}^\delta(\mathbb{V}, \mathbb{U})$  such that  $\phi \circ \psi = 1_{\mathbb{V}}^{2\delta}$  and  $\psi \circ \phi = 1_{\mathbb{U}}^{2\delta}$ . The interleaving distance between the persistence modules  $\mathbb{U}$  and  $\mathbb{V}$  is  $\text{ID}(\mathbb{U}, \mathbb{V}) = \inf\{\delta \geq 0 \mid \mathbb{U} \text{ and } \mathbb{V} \text{ are } \delta\text{-interleaved}\}$ .*

If  $f, g : X \rightarrow \mathbb{R}$  are continuous functions such that  $\|f - g\|_\infty \leq \delta$  in the  $L_\infty$ -distance, the modules  $H_k\{f^{-1}(-\infty, s]\}, H_k\{g^{-1}(-\infty, s]\}$  are  $\delta$ -interleaved for any  $k$  [15]. The last conclusion extends to persistence diagrams for the bottleneck distance below.

**Definition 9** (bottleneck distance BD). Let  $C, D$  be multisets of finitely many points  $(p, q) \in \mathbb{R}^2, p < q$ , of finite multiplicity and all diagonal points  $(p, p) \in \mathbb{R}^2$  of infinite multiplicity. For  $\delta \geq 0$ , a  $\delta$ -matching is a bijection  $h : C \rightarrow D$  such that  $|h(a) - a|_\infty \leq \delta$  in the  $L_\infty$ -distance for any point  $a \in C$ . The bottleneck distance between modules  $\mathbb{U}, \mathbb{V}$  is defined as  $BD(\mathbb{U}, \mathbb{V}) = \inf\{\delta \mid \text{there is a } \delta\text{-matching between } PD(\mathbb{U}), PD(\mathbb{V})\}$ .

Historically, stability of persistence for sequences of sublevel sets was extended as Theorem 3 to  $q$ -tame persistence modules. A persistence module  $\mathbb{V}$  is  $q$ -tame if any non-diagonal square in the persistence diagram  $PD(\mathbb{V})$  contains only finitely many of points; see Reference [16] (Section 2.8). Any finitely decomposable persistence module is  $q$ -tame.

**Theorem 3** (stability of persistence modules [16], isometry Theorem 4.11). Let  $\mathbb{U}$  and  $\mathbb{V}$  be  $q$ -tame persistence modules. Then,  $ID(\mathbb{U}, \mathbb{V}) = BD(PD(\mathbb{U}), PD(\mathbb{V}))$ , where  $ID$  is the interleaving distance, and  $BD$  is the bottleneck distance between persistence modules.

**Definition 10** (merge module  $M(\Delta)$ ). For any dendrogram  $\Delta$  on a finite set  $X$ , the merge module  $M(\Delta)$  consists of the vector spaces  $M_s(\Delta), s \in \mathbb{R}$ , and linear maps  $m_s^t : M_s(\Delta) \rightarrow M_t(\Delta), s \leq t$ . For any  $s \in \mathbb{R}$  and  $A \in \Delta(s)$ , the space  $M_s(\Delta)$  has the generator or a basis vector  $[A] \in M_s(\Delta)$ . For  $s < t$  and any set  $A \in \Delta(s)$ , if the image of  $A$  under  $\Delta_s^t$  coincides with  $A \subset X$ , so  $\Delta_s^t(A) = A$ , then  $m_s^t([A]) = [A]$ , else  $m_s^t([A]) = 0$ , see Figure 6.

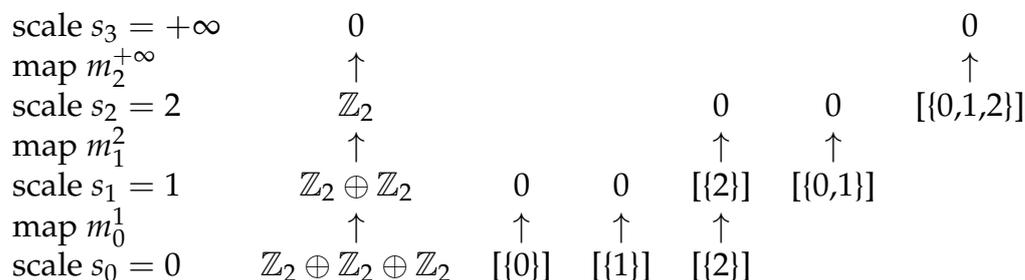


Figure 6. The merge module  $M(\Delta)$  of the dendrogram  $\Delta$  on the set  $X = \{0, 1, 2\}$  in Figure 3.

In a dendrogram  $\Delta$  from Definition 3, any merge set  $A$  of  $\Delta$  has  $\text{life}(A) = [b, d]$  from its birth scale  $b$  to its death scale  $d$ . Lemmas 2 and 3 are proved for the first time.

**Lemma 2** (merge module decomposition). For any dendrogram  $\Delta$  from Definition 3, the merge module  $M(\Delta) \cong \bigoplus_A \mathbb{I}(\text{life}(A))$  decomposes over all merge sets  $A$ .

**Proof of Lemma 2.** The goal is to prove that  $M(\Delta) \cong \bigoplus_A \mathbb{I}(\text{life}(A))$ . Recall that the interval module  $\mathbb{I}(\text{life}(A))$  consists of only vector spaces 0 and  $\mathbb{Z}_2$ . For a scale  $r$ , let  $\mathbb{I}_r(\text{life}(A))$  be its vector space, whose generator is denoted by  $[\mathbb{I}_r(\text{life}(A))]$ . Define

$$\psi_r : M_r(\Delta) \rightarrow \bigoplus_A \mathbb{I}_r(\text{life}(A)) \text{ such that } [A] \rightarrow [\mathbb{I}_r(\text{life}(A))] \text{ for all } A \in \Delta(r),$$

$$\phi_r : \bigoplus_A \mathbb{I}_r(\text{life}(A)) \rightarrow M_r(\Delta) \text{ such that } [\mathbb{I}_r(\text{life}(A))] \rightarrow [A] \text{ for all } \text{life}(A) \text{ containing } r.$$

We will first prove that  $\phi_r$  is well-defined. If  $r \in \text{life}(A)$  then  $A \in M_r(\Delta)$ . We know that  $M_r(\Delta)$  is generated by elements  $A \in \Delta(r)$  for which  $r \in \text{life}(A)$ . Thus, the compositions satisfy  $\phi_r \circ \psi_r = \text{id}_r$  and  $\psi_r \circ \phi_r = \text{id}_r$ . It remains to prove that morphisms

correctly behave under the functors  $\psi, \phi$ . The proofs for both cases are essentially the same; thus, we will prove it only for  $\psi$ . The goal is to prove that the following diagram commutes:

$$\begin{array}{ccc} M_s(\Delta) & \xrightarrow{m_s^t} & M_t(\Delta) \\ \psi_s \downarrow & & \downarrow \psi_t \\ \bigoplus_A \mathbb{I}_s(\text{life}(A)) & \xrightarrow{i_s^t} & \bigoplus_A \mathbb{I}_t(\text{life}(A)) \end{array}$$

Here,  $i_s^t$  is the direct sum of the corresponding maps of interval modules  $\bigoplus_A (i_s^t)^A$ . Let  $[A]$  be an arbitrary generator of  $M_r(\Delta)$ . There are two possibilities how  $m_s^t$  can map  $[A]$ . If  $t \in \text{life}(A)$ , then  $m_s^t([A]) = [A] \in M_t(\Delta)$  and by definition

$$\phi_t \circ m_s^t([A]) = [\mathbb{I}_t(\text{life}(A))].$$

Since both  $s, t \in \text{life}(A)$ , we also have that

$$m_s^t \circ \phi_t([A]) = [\mathbb{I}_t(\text{life}(A))] = \phi_t \circ m_s^t([A]).$$

Assume now that  $t \notin \text{life}(A)$ . Then,  $m_s^t([A]) = 0$ ; thus,  $\phi_t(m_s^t([A])) = 0$ . On the other hand,  $i_s^t \circ \phi_s([A]) = [\mathbb{I}_t(\text{life}(A))] = \mathbb{Z}_2$ . Since  $t \notin \text{life}(A)$ , we get  $i_s^t \circ \phi_s([A]) = 0$ .  $\square$

**Lemma 3** (interleaving of merge modules). *If subsets  $A, B$  of a metric space  $(X, d)$  have  $\text{HD}(A, B) = \delta$ , then the merge modules  $M(\Delta_{SL}(A)), M(\Delta_{SL}(B))$  are  $\delta$ -interleaved.*

**Proof of Lemma 3.** The equality  $\text{HD}(A, B) = \delta$  means that  $A$  is covered by the union of closed balls that have the radius  $\delta$  and centers at all points  $b \in B$ . This union is the preimage is  $d_B^{-1}([0, \delta])$ , i.e.,  $A \subset d_B^{-1}([0, \delta])$ . Extending the distance values by  $s \geq 0$ , we get  $d_A^{-1}([0, s]) \subset d_B^{-1}([0, s + \delta])$  and similarly  $d_B^{-1}([0, s]) \subset d_A^{-1}([0, s + \delta])$ .

Let  $U$  be an arbitrary set in  $\Delta_{SL}(A)$ . Define map  $\phi_r : M(A; r) \rightarrow M(B; r + \delta)$

$$\phi_r([U]) = \begin{cases} [U], & \text{if } r + \delta \in \text{life}(\Delta_{SL}(B), U), \\ 0, & \text{otherwise.} \end{cases}$$

Symmetrically, for any  $V \in \Delta_{SL}(B)$ , we define  $\psi_r : M(B; r) \rightarrow M(A; r + \delta)$

$$\psi_r([V]) = \begin{cases} [V], & \text{if } r + \delta \in \text{life}(\Delta_{SL}(A), V), \\ 0, & \text{otherwise.} \end{cases}$$

In the notation above,  $\text{life}(\Delta_{SL}(B), U)$  is the  $\text{life}(U)$  in the dendrogram  $\Delta_{SL}(B)$ . If  $U \notin \Delta_{SL}(B)(t)$  for all values  $t$ , then  $\text{life}(U) = \emptyset$ . By symmetry, it is enough to prove that the following diagrams commute:

$$\begin{array}{ccc} M_s(\Delta_{SL}(A)) & \xrightarrow{m_s^t} & M_t(\Delta_{SL}(A)) \\ \phi_s \downarrow & & \downarrow \phi_t \\ M_{s+\delta}(\Delta_{SL}(B)) & \xrightarrow{m_{s+\delta}^{t+\delta}} & M_{t+\delta}(\Delta_{SL}(B)) \end{array}$$
  

$$\begin{array}{ccc} & M_s(\Delta_{SL}(B)) & \\ \phi_{s-\delta} \nearrow & & \searrow \psi_s \\ M_{s-\delta}(\Delta_{SL}(A)) & \xrightarrow{m_{s-\delta}^{s+\delta}} & M_{s+\delta}(\Delta_{SL}(A)) \end{array}$$

We note first that, if  $[a, b] = \text{life}(\Delta_{SL}(A), U)$ , then  $\text{life}(\Delta_{SL}(B), U) \subseteq [a - \epsilon, b + \epsilon]$ .

We begin by proving commutativity of the first diagram. Let  $U$  be arbitrary element of  $\Delta_{SL}(A)(s)$ . If  $t \notin \text{life}(\Delta_{SL}(A), U)$  then  $\phi_t \circ m_s^t = 0$ . If  $s + \delta \notin \text{life}(\Delta_{SL}(B), U)$  or  $t + \delta \notin \text{life}(\Delta_{SL}(B), U)$  then we are done. Since  $t \notin \text{life}(\Delta_{SL}(A), U)$ , it follows that  $t + \delta \notin \text{life}(\Delta_{SL}(B), U)$ . Thus, , with given assumptions, the diagram commutes.

Assume now that  $t + \delta \notin \text{life}(\Delta_{SL}(A), U)$ . Then, both  $\phi_t(m_s^t(U)) = 0 = m_{s+\delta}^{t+\delta}(\phi_s(U))$ . In the last case, we assume that  $t \in \text{life}(\Delta_{SL}(A), U)$  and  $t + \delta \in \text{life}(\Delta_{SL}(B), U)$ . In this case, obviously,  $s + \delta \in \text{life}(\Delta_{SL}(B), U)$ ; thus,  $\phi_t(m_s^t([U])) = [U] = m_{s+\delta}^{t+\delta}(\phi_s([U]))$ .

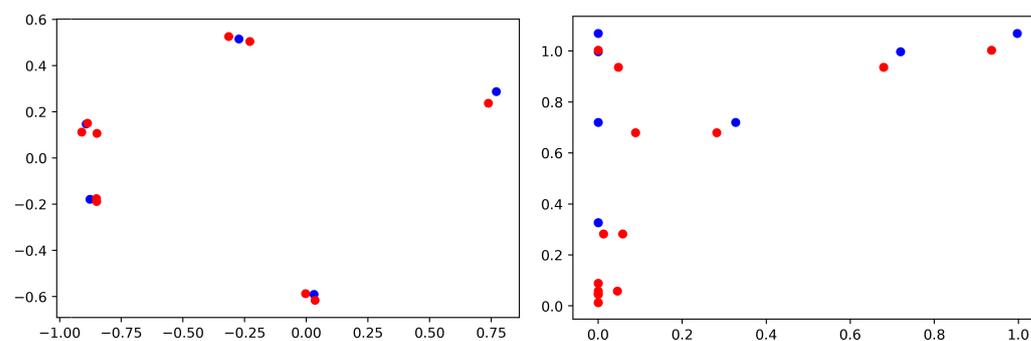
For the second diagram, assume now that  $U \in M(\Delta_{SL}(A))(s - \delta)$ . Assume first that  $s \notin \text{life}(\Delta_{SL}(B), U)$ , then  $s + \delta \notin \text{life}(\Delta_{SL}(B), U)$  and  $m_{s-\delta}^{s+\delta}([U]) = 0 = \psi_s(\phi_{s-\delta}[U])$ .

Assume then that  $s \in \text{life}(\Delta_{SL}(B), U)$ . Now, the outcome of both maps  $\psi_s$  and  $m_{s-\delta}^{s+\delta}$  depends on whether  $s + \delta \in \text{life}(\Delta_{SL}(A), U)$ ; thus,  $m_{s-\delta}^{s+\delta}([U]) = \psi_s(\phi_{s-\delta}[U])$ . Since all the diagrams commute, the required conclusion follows.  $\square$

**Theorem 4** (stability of a mergegram). *The mergegrams of any finite point clouds  $A, B$  in a metric space  $(X, d)$  satisfy  $\text{BD}(\text{MG}(\Delta_{SL}(A)), \text{MG}(\Delta_{SL}(B))) \leq \text{HD}(A, B)$ . Hence, any small perturbation of a cloud  $A$  in the Hausdorff distance leads to a similarly small perturbation in the bottleneck distance for its mergegram  $\text{MG}(\Delta_{SL}(A))$ .*

**Proof.** The given clouds  $A, B \subset X$  with  $\text{HD}(A, B) = \delta$  have  $\delta$ -interleaved merge modules by Lemma 3, so  $\text{ID}(\text{MG}(\Delta_{SL}(A)), \text{MG}(\Delta_{SL}(B))) \leq \delta$ . Since any merge module  $M(\Delta)$  is finitely decomposable, hence, it is  $q$ -tame by Lemma 2. The corresponding mergegram  $\text{MG}(M(\Delta))$  satisfies Theorem 3, so  $\text{BD}(\text{MG}(\Delta_{SL}(A)), \text{MG}(\Delta_{SL}(B))) \leq \delta$ .  $\square$

Figure 7 illustrates Theorem 4 on a cloud and its perturbation by showing their close mergegrams. The more extensive experiment on 100 clouds in Reference [7] (Figure 8) similarly confirms that the mergegram is perturbed within expected bounds. The computational complexity of the mergegram  $\text{MG}(\Delta_{SL}(A))$  was proved to be near linear in the number  $n$  of points in a cloud  $A \subset \mathbb{R}^m$ ; see Reference [7] (Theorem 8.2). The results above justify that the invariant mergegram satisfies conditions (a,b,c) of Isometry Recognition Problem 1.



**Figure 7.** Left: the cloud  $C$  of 5 blue points is close to the cloud  $C'$  of 10 red points in the Hausdorff distance. Right: the mergegrams are close in the bottleneck distance as predicted by Theorem 4.

### 6. New Experiments on Isometry Recognition of Substantially Distorted Real Shapes

This section fulfills final condition (d) of Problem 1 by experimentally comparing the mergegram with 0D persistence and distributions of distances to neighbors on 15,000 clouds. The earlier paper by Reference [7] did experiments only on randomly generated clouds.

We considered 15 classes of shapes represented by black and white images of mythical creatures [9]; see Figure 1. These shapes were chosen to make the shape recognition problem really challenging. Indeed, similar creatures from this dataset are represented by slightly different shapes, which can be hard to isometrically distinguish from each other. For example, several images of a horse include only minor differentiating features, such as a saddle or a different tails, which makes horses nearly identical.

**Shape generation.** For each image, we generated 1000 perturbed images by affine and projective transformations to get 15,000 distorted shapes split into 15 classes.

First, we rotated each image around its central point by an angle generated uniformly in the interval  $[0, 2\pi)$  using the function `cv::rotate` from the OpenCV library. If needed, we extended the resulting image to fit all black pixels of the rotated shape into a bounding box. Then, both affine and projective transformations distort each image by using a noise parameter  $\delta$  such that the value  $\delta = 0$  represents the identity transformation.

Figure 8 illustrates how an original image is randomly rotated, and then randomly distorted by affine or projective transformations, depending on the noise parameter  $\delta$ .

**Affine transformations** are implemented as compositions of the already applied rotations above and the function `cv::resize()` from the OpenCV library. This function scales an image of size  $w \times h$  by horizontal and vertical factors  $a, b$  sampled as follows.

- **Uniform noise:**  $a \in [1 - \delta w, 1 + \delta w]$ ,  $b \in [1 - \delta h, 1 + \delta h]$  have uniform distributions.
- **Gaussian noise:**  $a \in \mathcal{N}(1, \delta h) \cap \mathbb{R}_+$  and  $b \in \mathcal{N}(1, \delta w) \cap \mathbb{R}_+$  have Gaussian distributions with mean 1 and standard variance  $\delta h, \delta w$ , truncated to positive numbers.

**Projective transformation** are implemented as compositions of the already applied rotations above and the OpenCV function `cv::getPerspectiveTransform()` function, which is parametrized by 4-dimensional array  $v = (a_0, a_1, a_2, a_3)$  consisting of points  $a_i \in \mathbb{Z}^2$ ,  $i = 0, 1, 2, 3$ . This function maps the corners of the image as follows:

$$(0, 0) \mapsto a_0, (0, h) \mapsto a_1, (w, 0) \mapsto a_2 \text{ and } (w, h) \mapsto a_3.$$

The projective transformation of the rectangle  $w \times h$  is uniquely determined by the above corners. The above points  $a_i$  are randomly sampled by using a noise parameter  $\delta$ .

- **Uniform noise:** each coordinate has a uniform distribution with a noise parameter  $\delta$

$$a_0 \in [0, \delta w] \times [0, \delta h], \quad a_1 \in [0, \delta w] \times [h - \delta h, h],$$

$$a_2 \in [w - \delta w, w] \times [0, \delta h], \quad a_3 \in [w - \delta w, w] \times [h - \delta h, h].$$

- **Gaussian noise:** each coordinate has a Gaussian distribution truncated to the image

$$a_0 \in (\mathcal{N}(0, \delta w) \cap [0, w]) \times (\mathcal{N}(0, \delta h) \cap [0, h]),$$

$$a_1 \in (\mathcal{N}(0, \delta w) \cap [0, w]) \times (\mathcal{N}(h, \delta h) \cap [0, h]),$$

$$a_2 \in (\mathcal{N}(w, \delta w) \cap [0, w]) \times (\mathcal{N}(0, \delta h) \cap [0, h]),$$

$$a_3 \in (\mathcal{N}(w, \delta w) \cap [0, w]) \times (\mathcal{N}(h, \delta h) \cap [0, h]).$$

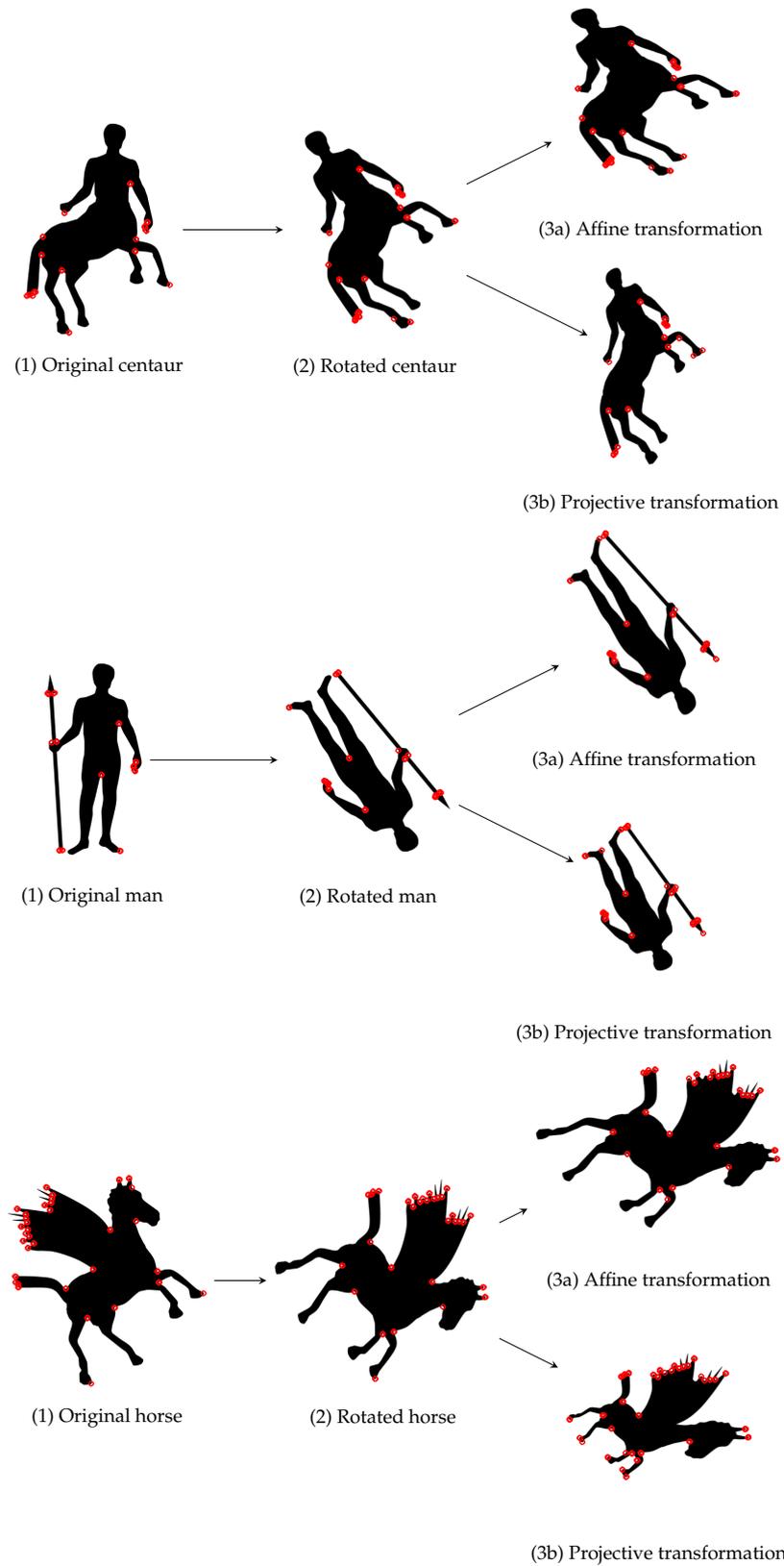
**Point cloud extraction.** For each distorted image, we extract classical Harris point corners [28] due to their simplicity; see the red points in Figure 8. For detecting corner points, the OpenCV function `cv::cornerHarris` was used with the parameters `blockSize = 3`, `apertureSize = 5`, `k = 0.04`, `thresh = 120`. However, one can use any reliable algorithms, such as FAST [29] or scale-invariant feature transform (SIFT) [30].

After describing the available point cloud data above, we specify condition (d) of Isometry Recognition Problem 1 in the context of supervised machine learning.

**Problem 2** (experimental recognition). *Given a labeled dataset split into classes of similar but projectively distorted shapes, develop a learning tool to recognize a class of distorted shapes with a high recognition rate (percentage of correctly recognized classes).*

Since all isometry invariants are independent of point ordering, the most suitable neural network is PersLay [20], whose output is invariant under permutations by design. Each layer is a combination of a coefficient layer  $\omega(p) : \mathbb{R}^m \rightarrow \mathbb{R}$ , a transformation  $\phi(p) : \mathbb{R}^m \rightarrow \mathbb{R}^q$  and a permutation invariant layer  $\text{op}$  combined as follows

$\text{PersLay}(D) = \text{op}(\{w(p)\phi(p)\}_{p \in D})$ , where  $D$  is a diagram or multiset of points in  $\mathbb{R}^m$ .



**Figure 8.** Generating distorted shapes by applying random rotations, affine and projective transformations, which substantially affect the extracted clouds of Harris corner points [28] in red.

Coordinates of all input points are linearly normalized to  $[0, 1]$ . We have used the following parameters of the PersLay network for all experiments below.

**The max layer**  $\text{MAX}(q)$  consists of the following functions.

- The coefficient layer  $w : \mathbb{R}^m \rightarrow \mathbb{R}$  is the weight  $w(x_1, \dots, x_m) = k|x_1 - x_2|$ , where  $k$  is a trainable scalar, and the dimension is typically  $m = 2$ .
- The transformation layer  $\phi : \{\text{diagrams of points in } \mathbb{R}^m\} \rightarrow \mathbb{R}^q$  is the function  $\phi(D) = \sum_{p \in D} \lambda p + \gamma \text{maxpool}(D) + \beta$ , where  $\lambda, \gamma$  are  $\mathbb{R}^{m \times q}$  trainable matrices,  $\beta$  is a  $\mathbb{R}^q$  trainable vector, and maxpool returns a maximum value for every  $i = 1, \dots, m$ .
- The operational layer  $\text{op} : \mathbb{R}^q \rightarrow \mathbb{R}^t$  puts all coordinates in increasing order and composes the result with standard densely connected layer [31]  $\text{Dense} : \mathbb{R}^q \rightarrow \mathbb{R}^t$ .

An output is a vector in  $\mathbb{R}^t$  for  $t = 15$  of image classes. A final prediction is obtained by choosing a class with a largest coordinate in the output vector.

**The image layer**  $\text{Im}[x, y]$  for integer parameters  $x, y$  and a multiset of points in the unit square  $[0, 1]^2$  consists of the following functions.

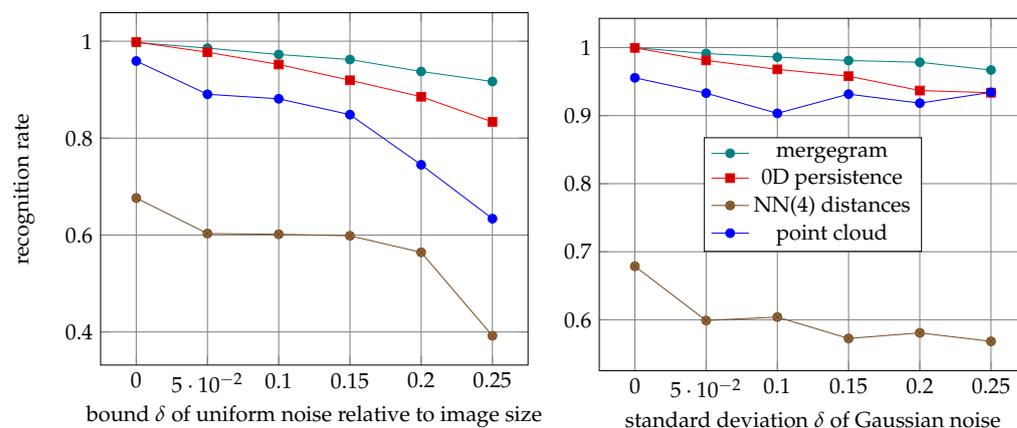
- The coefficient layer  $w : \mathbb{R}^2 \rightarrow \mathbb{R}$  is a piecewise constant function trained on  $x \cdot y$  parameters, defined on the unit square partition

$$\mathcal{P}(x, y) = \left\{ \left[ \frac{i}{x}, \frac{i+1}{x} \right] \times \left[ \frac{j}{y}, \frac{j+1}{y} \right] \mid i = 0, \dots, x-1 \text{ and } j = 0, \dots, y-1 \right\}.$$

- Let  $\phi_p : \mathbb{R}^2 \rightarrow \mathbb{R}$  be the Gaussian distribution centered at  $p \in D$  with a trainable standard deviation  $\sigma$ . The transformation layer  $\phi : \mathbb{R} \rightarrow \mathbb{R}^{xy}$  consists of  $xy$  functions  $\phi_p$ , where  $p$  runs over all centroids of the partition  $\mathcal{P}(x, y)$ .
- The operation layer  $\text{op}$  takes the sum over the given point cloud. A final prediction is made by composing the operation layer with the Dense layer.

Finally, the PersLay network used the optimizer `tf.keras.adam` with the standard learning rate 0.01 and 150 epochs, the loss function `SparseCategoricalCrossEntropy`, the 80:20 of training and testing, a 5-fold Monte Carlo cross validation for each run.

Figures 9–11 show that the mergegram MG consistently outperforms two other isometry invariants: 0D persistence and the multiset  $\text{NN}(4)$  consisting of 4-tuple distances to neighbors per given point. The simpler multiset  $\text{NN}(2)$  performed worse. A given cloud  $C \subset \mathbb{R}^2$  was considered as a baseline input. The noise factors  $\delta$  reached 25%, which means that original images were distorted up to a quarter of image sizes.



**Figure 9.** Recognition rates are obtained by training the max layer  $\text{MAX}(75)$  of PersLay on three isometry invariants and a cloud of corner points extracted from 15,000 affinely distorted images.

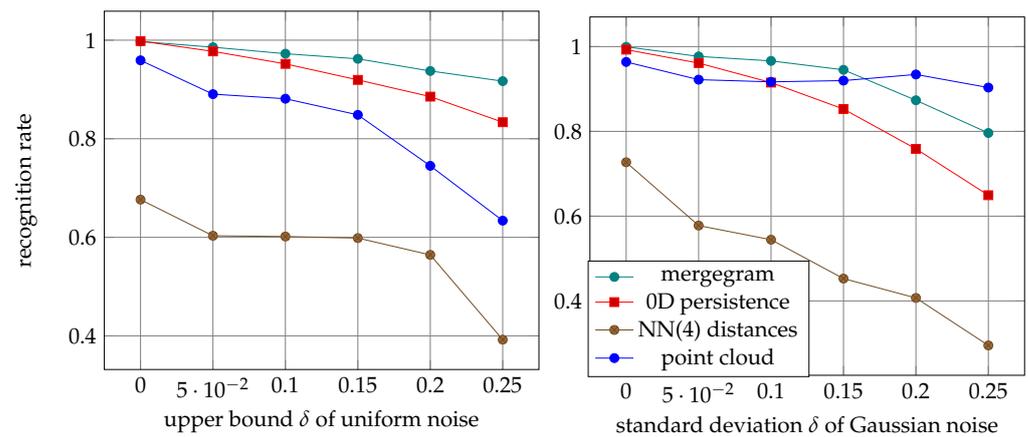


Figure 10. Recognition rates are obtained by training the max layer MAX(75) of PersLay on isometry invariants and corner points extracted from 15,000 projectively distorted images.

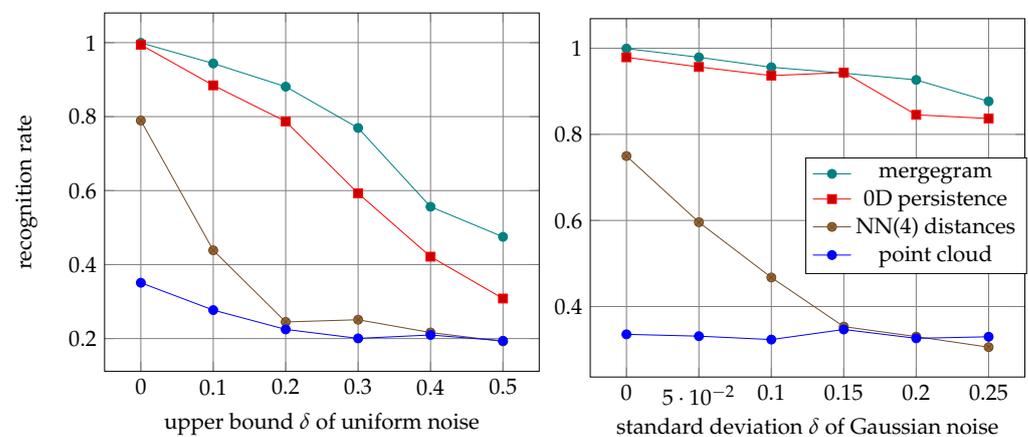


Figure 11. Recognition rates are obtained by training the image layer IM[20,20] of PersLay on isometry invariants and a cloud of corner points extracted from 15,000 affinely distorted images.

### 7. A Discussion of Novel Contributions and Further Open Problems

This paper has further demonstrated that the provably stable-under-noise invariant mergegram of a dendrogram is a fast and efficient tool in the challenging problem of isometry shape recognition, especially for substantially distorted images.

In comparison with the conference version [7], Section 4 proved new Theorem 2 describing how to reconstruct a single-linkage dendrogram in general position from its much simpler mergegram. It is hard to define a continuous metric between dendrograms, especially because they can be unstable under perturbations. Theorem 2 allows us to measure a continuous similarity between dendrograms in general position as the bottleneck distance between their unique mergegrams. This distance can be computed in time  $O(n^{1.5} \log n)$  [32] for diagrams consisting of at most  $n$  points.

Section 5 provided a full proof of stability of the mergegram under perturbations of points, while the earlier paper by Reference [7] only announced this result without proving highly non-trivial Lemmas 2 and 3, which required a heavy algebraic machinery.

In addition to Example 1 and Theorem 1 showing that the mergegram is stronger than 0D persistence, its strength is confirmed by the new experiments on 15,000 point samples from substantially distorted real shapes. In Figures 9–11, the mergegram outperformed other isometry invariants. Since the distribution  $NN(2)$  of distances to two closest neighbors per point performed badly, we have tried distances to four nearest neighbors, but this  $NN(4)$  performed worse than the original cloud because of noise.

For very high levels of 20% and 25% distortions in projective transformations, the PersLay network trained on a point cloud achieved high recognition rates because we have

extensively tried many parameters in the layers MAX(75) and Im[20,20] for a best trade-off between accuracy and speed. The C++ code for the mergegram is at Reference [7].

The recent stronger invariants are Pointwise Distance Distributions [33]. Their generic completeness under isometry holds in a more challenging setting of periodic point sets [34–38].

**Author Contributions:** Conceptualization, V.K.; data curation, Y.E.; formal analysis, Y.E.; funding acquisition, V.K.; investigation, Y.E.; methodology, V.K.; project administration, V.K.; resources, Y.E.; software, Y.E.; writing—original draft preparation, Y.E.; writing—review and editing, V.K.; supervision, V.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the EPSRC grant ‘Application-driven Topological Data Analysis’, reference EP/R018472/1. The APC was funded by the University of Liverpool.

**Data Availability Statement:** The original image dataset is available at <http://tosca.cs.technion.ac.il> (accessed on 25 August 2021).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Pauly, M.; Gross, M.; Kobbelt, L.P. Efficient simplification of point-sampled surfaces. In Proceedings of the IEEE Visualization, Boston, MA, USA, 27 October–1 November 2002; pp. 163–170.
2. Zwicker, M.; Pauly, M.; Knoll, O.; Gross, M. Pointshop 3D: An interactive system for point-based surface editing. *ACM Trans. Graph. (TOG)* **2002**, *21*, 322–329. [[CrossRef](#)]
3. Boutin, M.; Kemper, G. On reconstructing n-point configurations from the distribution of distances or areas. *Adv. Appl. Math.* **2004**, *32*, 709–735. [[CrossRef](#)]
4. Huttenlocher, D.P.; Klanderma, G.A.; Rucklidge, W.J. Comparing images using the Hausdorff distance. *Trans. Pattern Anal. Mach. Intell.* **1993**, *15*, 850–863. [[CrossRef](#)]
5. Chew, P.; Goodrich, M.; Huttenlocher, D.; Kedem, K.; Kleinberg, J.; Kravets, D. Geometric pattern matching under Euclidean motion. *Comput. Geom.* **1997**, *7*, 113–124. [[CrossRef](#)]
6. Goodrich, M.T.; Mitchell, J.S.; Orletsky, M.W. Approximate geometric pattern matching under rigid motions. *Trans. Pattern Anal. Mach. Intell.* **1999**, *21*, 371–379. [[CrossRef](#)]
7. Elkin, Y.; Kurlin, V. The mergegram of a dendrogram and its stability. In Proceedings of the MFCS 2020, Prague, Czech Republic, 24–28 August 2020.
8. Bronstein, A.M.; Bronstein, M.M.; Kimmel, R. *Numerical Geometry of Non-Rigid Shapes*; Springer: Berlin/Heidelberg, Germany, 2008.
9. Bronstein, A.M.; Bronstein, M.M.; Bruckstein, A.M.; Kimmel, R. Analysis of two-dimensional non-rigid shapes. *Int. J. Comput. Vis.* **2008**, *78*, 67–88. [[CrossRef](#)]
10. Elad, A.; Kimmel, R. Bending invariant representations for surfaces. In Proceedings of the Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001.
11. Ovsjanikov, M.; Mérigot, Q.; Méholi, F.; Guibas, L. One point isometric matching with the heat kernel. In *Computer Graphics Forum*; Blackwell Publishing: Oxford, UK, 2010; Volume 29; pp. 1555–1564.
12. Ovsjanikov, M.; Bronstein, A.M.; Bronstein, M.M.; Guibas, L.J. Shape google: A computer vision approach to isometry invariant shape retrieval. In Proceedings of the ICCV Workshops, Kyoto, Japan, 27 September–4 October 2009; pp. 320–327.
13. Méholi, F.; Sapiro, G. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Found. Comput. Math.* **2005**, *5*, 313–347. [[CrossRef](#)]
14. Verri, A.; Uras, C.; Frosini, P.; Ferri, M. On the use of size functions for shape analysis. *Biol. Cybern.* **1993**, *70*, 99–107. [[CrossRef](#)]
15. Cohen-Steiner, D.; Edelsbrunner, H.; Harer, J. Stability of persistence diagrams. *Discret. Comput. Geom.* **2007**, *37*, 103–120. [[CrossRef](#)]
16. Chazal, F.; De Silva, V.; Glisse, M.; Oudot, S. *The Structure and Stability of Persistence Modules*; Springer: Berlin/Heidelberg, Germany, 2016.
17. Parsa, S. A Deterministic  $O(m \log m)$  Time Algorithm for the Reeb Graph. *Discret. Comput. Geom.* **2013**, *49*, 864–878. [[CrossRef](#)]
18. Morozov, D.; Beketayev, K.; Weber, G. Interleaving distance between merge trees. *Discret. Comput. Geom.* **2013**, *49*, 52.
19. Smith, P.; Kurlin, V. Skeletonisation algorithms with theoretical guarantees for unorganised point clouds with high levels of noise. *Pattern Recognit.* **2021**, *115*, 107902. [[CrossRef](#)]
20. Carriere, M.; Chazal, F.; Ike, Y.; Lacombe, T.; Royer, M.; Umeda, Y. PersLay: A Neural Network Layer for Persistence Diagrams and Graph Topological Signatures. In Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS), Palermo, Italy, 26–28 August 2020.
21. Zaheer, M.; Kottur, S.; Ravanbakhsh, S.; Poczos, B.; Salakhutdinov, R.R.; Smola, A.J. Deep sets. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 3391–3401.

22. Fu, G.; Hou, C.; Yao, X. Learning topological representation for networks via hierarchical sampling. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
23. Cirrincione, G.; Ciravegna, G.; Barbiero, P.; Randazzo, V.; Pasero, E. The GH-EXIN neural network for hierarchical clustering. *Neural Netw.* **2020**, *121*, 57–73. [[CrossRef](#)] [[PubMed](#)]
24. Karim, M.R.; Beyan, O.; Zappa, A.; Costa, I.G.; Rebholz-Schuhmann, D.; Cochez, M.; Decker, S. Deep learning-based clustering approaches for bioinformatics. *Briefings Bioinform.* **2021**, *22*, 393–415. [[CrossRef](#)] [[PubMed](#)]
25. Clough, J.; Byrne, N.; Oksuz, I.; Zimmer, V.A.; Schnabel, J.A.; King, A. A topological loss function for deep-learning based image segmentation. *Trans. PAMI* **2020**. [[CrossRef](#)]
26. Gabrielsson, R.B.; Nelson, B.J.; Dwaraknath, A.; Skraba, P. A topology layer for machine learning. In Proceedings of the International Conference Artificial Intelligence and Statistics, Virtually, 26–28 August 2020; pp. 1553–1563.
27. Carlsson, G.; Memoli, F. Characterization, stability and convergence of hierarchical clustering methods. *J. Mach. Learn. Res.* **2010**, *11*, 1425–1470.
28. Sánchez, J.; Monzón, N.; Salgado De La Nuez, A. An analysis and implementation of the harris corner detector. *Image Process. Line* **2018**, *8*, 305–328. [[CrossRef](#)]
29. Rosten, E.; Porter, R.; Drummond, T. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *32*, 105–119. [[CrossRef](#)] [[PubMed](#)]
30. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 2, pp. 1150–1157.
31. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv* **2016**, arXiv:1603.04467.
32. Kerber, M.; Morozov, D.; Nigmatov, A. Geometry Helps to Compare Persistence Diagrams. In Proceedings of the ALENEX, Arlington, VA, USA, 10–13 January 2016; pp. 103–112.
33. Widdowson, D.; Kurlin, V. Pointwise distance distributions of periodic sets. *arXiv* **2021**, arXiv:2108.04798.
34. Mosca, M.; Kurlin, V. Voronoi-Based Similarity Distances between Arbitrary Crystal Lattices. *Cryst. Res. Technol.* **2020**, *55*, 1900197. [[CrossRef](#)]
35. Anosova, O.; Kurlin, V. Introduction to Periodic Geometry and Topology. *arXiv* **2021**, arXiv:2103.02749.
36. Anosova, O.; Kurlin, V. An isometry classification of periodic point sets. In Proceedings of the Discrete Geometry and Mathematical Morphology, Uppsala, Sweden, 24–27 May 2021.
37. Edelsbrunner, H.; Heiss, T.; Kurlin, V.; Smith, P.; Wintraecken, M. The Density Fingerprint of a Periodic Point Set. In Proceedings of the SoCG, Virtually, 7–10 June 2021.
38. Widdowson, D.; Mosca, M.; Pulido, A.; Kurlin, V.; Cooper, A. Average Minimum Distances of Periodic Point Sets. *MATCH Communications in Mathematical and in Computer Chemistry*. Available online: <https://match.pmf.kg.ac.rs> (accessed on 6 July 2021).