

# Multi-scale approximation of the matching distance for shape retrieval

Andrea Cerri<sup>1</sup>, Barbara Di Fabio<sup>1</sup>, and Filippo Medri<sup>2</sup>

<sup>1</sup> ARCES, Università di Bologna, Italia  
{andrea.cerri2,barbara.difabio}@unibo.it

<sup>2</sup> Dipartimento di Scienze dell'Informazione, Università di Bologna, Italia  
filippo.medri@gmail.com

**Abstract.** This paper deals with the concepts of persistence diagrams and matching distance. They are two of the main ingredients of *Topological Persistence*, which has proven to be a promising framework for shape comparison. Persistence diagrams are descriptors providing a signature of the shapes under study, while the matching distance is a metric to compare them. One drawback in the application of these tools is the computational costs for the evaluation of the matching distance. The aim of the present paper is to introduce a new framework for the approximation of the matching distance, which does not affect the reliability of the entire approach in comparing shapes, and extremely reduces computational costs. This is shown through experiments on 3D-models.

**Keywords:** Persistence diagram, shape analysis, dissimilarity criterion

## 1 Introduction

Interpreting and comparing shapes are challenging issues in computer vision, computer graphics and pattern recognition [11, 12]. Topological Persistence – including Persistent Homology [9] and Size Theory [1, 10] – has proven to be a successful comparison/retrieval/classification (hereafter CRC) scheme.

In a nutshell, the basic idea for dealing with the CRC task is to define a measure of the (dis)similarity between the shapes in a given database. This can be done by extracting a battery of shape descriptors – the so-called *persistence diagrams* – from each element in the database, capturing meaningful shape properties. Thus, the problem of assessing the (dis)similarity between two shapes can be recast into the one of comparing the associated persistence diagrams according to the *matching* (or *bottleneck*) *distance*, a proven stable distance between these descriptors. This process defines a metric over the database, that can be used for CRC purposes. In general, a given persistence diagram may come from different shapes: This can be interpreted as an equivalence with respect to the properties captured by that descriptor.

Such an approach has been successfully used in a number of concrete problems concerning shape comparison and retrieval [4, 5, 8]. However, defining a (dis)similarity metric in the case of large databases can lead to considerable

computational costs. The bottleneck in this procedure can be identified in the evaluation of the matching distance.

**The contribution of the paper.** Reducing the computational costs in defining a (dis)similarity metric within a database of shapes is definitely a desirable target: This would enable us to further improve the persistence CRC framework and apply it to a wider class of concrete problems. The present paper aims to illustrate an idea to achieve this goal, ranging from a theoretical formalization of the proposed strategy to its validation through an experimental study. We introduce a multi-scale construction of our matching distance-based (dis)similarity metric. Our procedure is based on a “dissimilarity criterion” which is formalized in Theorem 1. Experiments on 3D-models show that, using our idea, it is possible not to affect the reliability of the entire approach in comparing shapes, extremely reducing the computational costs.

## 2 Preliminaries

In persistence, the shape of an object is usually studied by choosing a topological space  $X$  to represent it, and a function  $\varphi : X \rightarrow \mathbb{R}$ , called a *filtering (or measuring) function*, to define a family of subspaces  $X_u = \varphi^{-1}((-\infty, u])$ ,  $u \in \mathbb{R}$ , nested by inclusion, i.e. a filtration of  $X$ . Applying homology to the filtration allows us to study how topological features vary in passing from a set of the filtration into a larger one, and to rank topological features with bounded lifetime by importance, according to the length of their life. The basic assumption is that the longer a feature survives, the more meaningful or coarse the feature is for shape description. Vice-versa, noise and shape details are characterized by a shorter life. For further details we refer to [1, 9].

The filtration  $\{X_u\}_{u \in \mathbb{R}}$  is used to define persistent homology groups as follows. Given  $u \leq v \in \mathbb{R}$ , we consider the inclusion of  $X_u$  into  $X_v$ . This inclusion induces a homomorphism of homology groups  $H_k(X_u) \rightarrow H_k(X_v)$  for every  $k \in \mathbb{Z}$ . Its image consists of the  $k$ -homology classes that live at least from  $H_k(X_u)$  to  $H_k(X_v)$  and is called the  *$k$ th persistent homology group of  $(X, \varphi)$  at  $(u, v)$* . When this group is finitely generated, we denote by  $\beta_k^{u,v}(X, \varphi)$  its rank.

A simple and compact description of persistent homology groups of  $(X, \varphi)$  is provided by the so-called persistence diagrams, i.e. multisets of points whose abscissa and ordinate are, respectively, the level at which a new  $k$ -homology class is created and the level at which it is annihilated through the filtration.

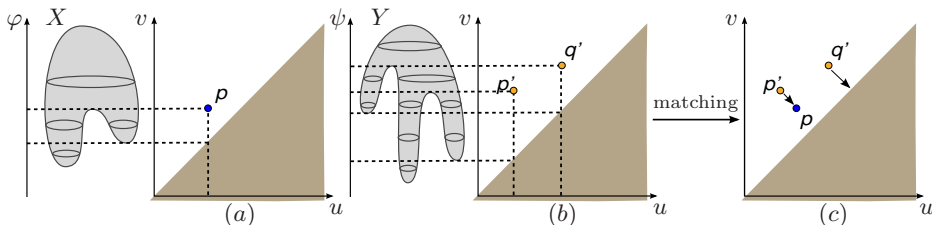
We use the following notation:  $\Delta^+ = \{(u, v) \in \mathbb{R}^2 : u < v\}$ ,  $\Delta = \{(u, v) \in \mathbb{R}^2 : u = v\}$ , and  $\overline{\Delta^+} = \Delta^+ \cup \Delta$ .

**Definition 1 (Multiplicity).** Let  $k \in \mathbb{Z}$  and  $(u, v) \in \Delta^+$ . The multiplicity  $\mu_k(u, v)$  of  $(u, v)$  is the finite non-negative number defined by

$$\lim_{\varepsilon \rightarrow 0^+} \left( \beta_k^{u+\varepsilon, v-\varepsilon}(X, \varphi) - \beta_k^{u-\varepsilon, v-\varepsilon}(X, \varphi) - \beta_k^{u+\varepsilon, v+\varepsilon}(X, \varphi) + \beta_k^{u-\varepsilon, v+\varepsilon}(X, \varphi) \right).$$

**Definition 2 (Persistence Diagram).** The persistence diagram  $D_k(X, \varphi)$  is the multiset of all points  $(u, v) \in \Delta^+$  such that  $\mu_k(u, v) > 0$ , counted with their multiplicity, union the points of  $\Delta$ , counted with infinite multiplicity.

We will call *proper points* the points of a persistence diagram lying on  $\Delta^+$ .



**Fig. 1.** (a) The height function  $\varphi$  on the space  $X$ , and the associated persistence diagram  $D_0(X, \varphi)$ . (b) The height function  $\psi$  on the space  $Y$ , and the associated persistence diagram  $D_0(Y, \psi)$ . (c) The matching between  $D_0(X, \varphi)$  and  $D_0(Y, \psi)$  realizing their matching distance.

Figures 1 (a) – (b) show two examples of persistence diagrams for  $k = 0$ . For instance, in Figure 1 (a) a surface  $X \subset \mathbb{R}^3$  is filtered by the height function  $\varphi$ . The sole proper point of  $D_0(X, \varphi)$  is  $p$ . Its abscissa corresponds to the level at which a new connected component is born, while its ordinate identifies the level at which this connected component merges with the existing one. To see, for instance, that  $\mu_0(p) = 1$ , letting  $p = (\bar{u}, \bar{v})$ , it is sufficient to observe that, for every  $\varepsilon > 0$  sufficiently small, it holds that  $\beta_0^{\bar{u}+\varepsilon, \bar{v}-\varepsilon}(X, \varphi) = 2$ ,  $\beta_0^{\bar{u}-\varepsilon, \bar{v}-\varepsilon}(X, \varphi) = \beta_0^{\bar{u}+\varepsilon, \bar{v}+\varepsilon}(X, \varphi) = \beta_0^{\bar{u}-\varepsilon, \bar{v}+\varepsilon}(X, \varphi) = 1$ , and apply Definition 1.

The matching distance between two persistence diagrams measures the cost of finding a correspondence between their points. In doing this, the cost of taking a point  $p$  to a point  $p'$  is measured as the minimum between the cost of moving one point onto the other and the cost of moving both points onto the diagonal, see Figure 1 (c) for an example. In particular, the matching of a proper point  $p$  with a point of  $\Delta$  can be interpreted as the destruction of the point  $p$ . Formally:

**Definition 3 (Matching Distance).** Let  $D_k^1, D_k^2$  be two persistence diagrams. The matching distance  $d_{\text{match}}(D_k^1, D_k^2)$  is defined as

$$d_{\text{match}}(D_k^1, D_k^2) = \min_{\sigma} \max_{p \in D_k^1} d(p, \sigma(p)),$$

where  $\sigma$  varies among all the bijections between  $D_k^1$  and  $D_k^2$  and

$$d((u, v), (u', v')) = \min \left\{ \max \{ |u - u'|, |v - v'| \}, \max \left\{ \frac{v - u}{2}, \frac{v' - u'}{2} \right\} \right\} \quad (1)$$

for every  $(u, v), (u', v') \in \overline{\Delta^+}$ .

The importance of the matching distance in persistence is based on the fact that persistence diagrams are robust with respect to it. Roughly, small changing in a given filtering function (w.r.t. the max-norm) produces just a small changing in the associated persistence diagram w.r.t. the matching distance [7, 9].

*Remark 1.* From Definition 3 it follows that  $d_{match}(D_k^1, D_k^2) \leq (V - U)/2$ , with  $U = \min_{(u,v) \in L} u$ ,  $V = \max_{(u,v) \in L} v$  and  $L = D_k^1 \cup D_k^2$ . Indeed,  $(V - U)/2$  upper bounds the cost of the bijection between  $D_k^1$  and  $D_k^2$ , taking all the points of  $L$  onto  $\Delta$ . Since  $d_{match}$  is realized by the cheapest bijection between  $D_k^1$  and  $D_k^2$ , we have the claim.

This result will be useful later.

### 3 Theoretical setting and results

Computationally, evaluating the matching distance between two persistence diagrams takes  $O(h^{2.5})$  [6], being  $h$  the total amount of their proper points.

As stressed before, in CRC applications involving large databases, computing the matching distance for any possible shape comparison can imply a high computational cost. In fact, noisy or detailed shape models can produce persistence diagrams with a large number of proper points. Our goal is to reduce this computational complexity by contenting, at first, of a rough estimation of the metric induced by the matching distance over a database, to be possibly refined whenever it is not sufficient to distinguish between different shapes.

The key point here is the observation that, in most cases, realizing that two shapes are very dissimilar does not require to compute the *exact* matching distance between the associated persistence diagrams. Deciding, e.g., whether an elephant is different from an ant requires only a first glance at the two animals. In our framework, such a “first glance” could be equivalent to a rough estimation of the matching distance – and hence faster than its exact computation – between the persistence diagrams associated with the “elephant shape” and the “ant shape”, respectively. On the contrary, a different level of accuracy could be necessary to distinguish, e.g., the “wolf shape” from the “German shepherd shape”. This would lead to a sharper estimation of the matching distance between the associated persistence diagrams, possibly to its actual computation.

In light of these considerations, we propose a multi-scale construction of our matching distance-based (dis)similarity metric.

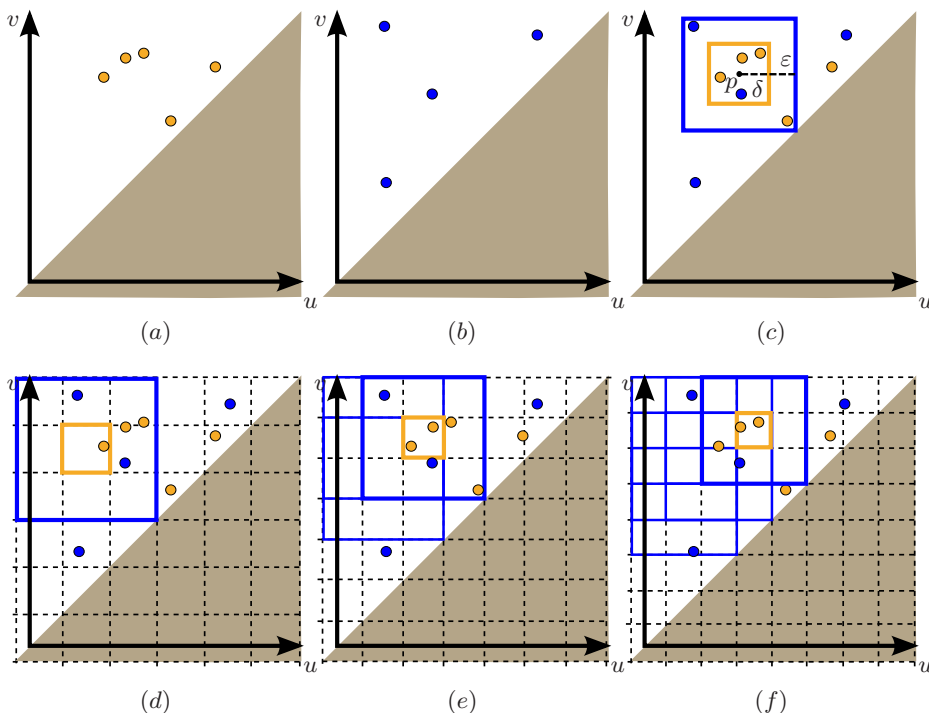
Let  $D_k$  be a persistence diagram. For every  $p = (u, v) \in \Delta^+$  and every  $\delta > 0$ , let  $\mathcal{Q}_\delta(p)$  be the open square centered at  $p$  of side equal to  $2\delta$ , and let us denote by  $\#(\mathcal{Q}_\delta(p), D_k)$  the number of points of  $D_k$  contained in  $\mathcal{Q}_\delta(p)$ .

**Theorem 1 (Dissimilarity criterion).** *Let  $D_k^1, D_k^2$  be two persistence diagrams for which a point  $p = (u, v) \in \Delta^+$  and two real numbers  $\delta, \varepsilon > 0$  exist, such that  $\mathcal{Q}_{\delta+\varepsilon}(p) \subset \Delta^+$  and  $\#(\mathcal{Q}_\delta(p), D_k^1) - \#(\mathcal{Q}_{\delta+\varepsilon}(p), D_k^2) > 0$ . Then  $d_{match}(D_k^1, D_k^2) \geq \varepsilon$ .*

*Proof.* Since  $\#(\mathcal{Q}_\delta(p), D_k^1) > \#(\mathcal{Q}_{\delta+\varepsilon}(p), D_k^2)$ , for every bijection  $\sigma : D_k^1 \rightarrow D_k^2$  there exists at least one proper point  $\bar{q} = (\bar{u}, \bar{v}) \in D_k^1$  such that  $\bar{q} \in \mathcal{Q}_\delta(p)$  and  $\sigma(\bar{q}) = \bar{q}' = (\bar{u}', \bar{v}') \in D_k^2$ , with  $\bar{q}' \notin \mathcal{Q}_{\delta+\varepsilon}(p)$ . Then, from (1) it holds that

$$d(\bar{q}, \bar{q}') \geq \min \left\{ \varepsilon, \max \left\{ \frac{\bar{v} - \bar{u}}{2}, \frac{\bar{v}' - \bar{u}'}{2} \right\} \right\} \geq \min \left\{ \varepsilon, \frac{\bar{v} - \bar{u}}{2} \right\} = \varepsilon. \quad (2)$$

Indeed, in (2), the first inequality holds because both  $|\bar{u} - \bar{u}'|$  and  $|\bar{v} - \bar{v}'|$  are not smaller than the difference between the semi-sides of  $\mathcal{Q}_\delta(p)$  and  $\mathcal{Q}_{\delta+\varepsilon}(p)$ ; the second inequality is obvious; the equality follows from both the facts that  $\bar{v} - \bar{u} > (v - \delta) - (u + \delta)$ , being  $(\bar{u}, \bar{v}) \in \mathcal{Q}_\delta(p)$  and  $(u + \delta, v - \delta) \in \Delta^+$  the bottom right vertex of  $\mathcal{Q}_\delta(p)$ , and  $(v - \delta - \varepsilon) - (u + \delta + \varepsilon) \geq 0$ , i.e.  $(v - \delta) - (u + \delta) \geq 2\varepsilon$ , being  $(u + \delta + \varepsilon, v - \delta - \varepsilon) \in \Delta^+$  the bottom right vertex of  $\mathcal{Q}_{\delta+\varepsilon}(p)$ . Hence  $\max_{q \in D_k^1} d(q, \sigma(q)) \geq \varepsilon$  for every bijection  $\sigma$  and, by Definition 3, the claim is proved.



**Fig. 2.** (a) – (b) Two persistence diagrams  $D_k^1$  and  $D_k^2$ . (c) The overlapping of  $D_k^1$  and  $D_k^2$ , and the two squares  $\mathcal{Q}_\delta(p)$  and  $\mathcal{Q}_{\delta+\varepsilon}(p)$  for a certain  $p \in \Delta^+$ . (d) – (f) Algorithm 1 in action: three steps are necessary to find squares in which Theorem 1 holds.

Figures 2 (a)–(c) show an example of Theorem 1 in action. Figures 2 (a)–(b) represent two persistence diagrams, say  $D_k^1$  and  $D_k^2$ , respectively. In Figure 2 (c) the two multisets of points are overlapped, and the two squares  $\mathcal{Q}_\delta(p)$  and  $\mathcal{Q}_{\delta+\varepsilon}(p)$  are depicted. As can be seen, it holds that  $\#(\mathcal{Q}_\delta(p), D_k^1) - \#(\mathcal{Q}_{\delta+\varepsilon}(p), D_k^2) = 1$ . Hence, by Theorem 1 we get that surely  $d_{\text{match}}(D_k^1, D_k^2) \geq \varepsilon$ .

The issue here is to find a suitable way to apply Theorem 1, so to improve our CRC framework. This is what the following Algorithm 1 is thought for.

Algorithm 1 takes as input the lists  $A$  and  $B$  of proper points of two persistence diagrams, and a parameter  $Exp$ . It runs a number of iterations equal to  $\lceil h^{Exp} \rceil$  where  $h = |A| + |B|$  is the sum of the number of points of  $A$  and  $B$ , and  $Exp$  is an arbitrary positive rational number. During each iteration, a finer grid is created on a triangular region  $T \subset \overline{\Delta^+}$  with vertices  $(U - \varepsilon, U - \varepsilon)$ ,  $(U - \varepsilon, V + \varepsilon)$ ,  $(V + \varepsilon, V + \varepsilon)$ , being  $U$  and  $V$  as in Remark 1, containing all the points belonging to  $A$  and  $B$ . In particular, at each iteration  $n$ , the algorithm produces  $n(n+1)/2$  small squares with side equal to  $(n+5)th$  part of the side of  $T$ . It then evaluates Theorem 1 on each small square compared with the square having its same center and side three times greater. The algorithm returns the maximum value for which Theorem 1 holds. Algorithm 1 makes use of two different subroutines:  $Matrix(i, j)$  which simply generates a two dimensional matrix  $0_{i \times j}$  and  $CountPoints(S, p, q)$  (Algorithm 2) whose output is the sum of the entries of the  $3 \times 3$  submatrix  $S[p-1, p, p+1; q-1, q, q+1]$  which is nothing more than the number of points of the largest square into which we are going to evaluate the theorem. Algorithm 3 gives as output the actual or the approximated distance between two persistence diagrams. An example of Algorithm 1 in action is shown in Figures 2 (d) – (f).

---

**Algorithm 1** MatchDistGridApprox( $A, B, Exp$ )

---

1: $N \leftarrow \lceil ( A  +  B )^{Exp} \rceil$ 2: $Res \leftarrow 0$ 3: $\varepsilon \leftarrow (V - U)/10$ 4: $Side \leftarrow V - U + 2\varepsilon$ 5: <b>for</b> $n = 1$ <b>to</b> $N$ <b>do</b> 6: $t \leftarrow 5 + n$ 7: $sSide \leftarrow Side/t$ 8: $qA \leftarrow Matrix(t, t)$ 9: $qB \leftarrow Matrix(t, t)$ 10: <b>for all</b> $a \in A$ <b>do</b> 11: $(i, j) \leftarrow \lceil (a - U + \varepsilon)/sSide \rceil$ 12: $qA(i, j) \leftarrow qA(i, j) + 1$ 13: <b>end for</b> 14: <b>for all</b> $b \in B$ <b>do</b> 15: $(i, j) \leftarrow \lceil (b - U + \varepsilon)/sSide \rceil$	16: $qB(i, j) \leftarrow qB(i, j) + 1$ 17: <b>end for</b> 18: <b>for</b> $p = 2$ <b>to</b> $(t - 1)$ <b>do</b> 19: <b>for</b> $q = (p + 3)$ <b>to</b> $(t - 1)$ <b>do</b> 20: $QA \leftarrow CountPoints(qA, p, q)$ 21: $QB \leftarrow CountPoints(qB, p, q)$ 22: $r1 \leftarrow (QA < QB(p, q))$ 23: $r2 \leftarrow (QB < qA(p, q))$ 24: <b>if</b> $(r1$ <b>or</b> $r2)$ <b>and</b> $(Res < sSide)$ <b>then</b> 25: $Res \leftarrow sSide$ 26: <b>end if</b> 27: <b>end for</b> 28: <b>end for</b> 29: <b>end for</b> 30: <b>return</b> $Res$
---	--

---

The computational complexity  $C$  of Algorithm 1 can be formalized as

$$C(h, Exp) = c_1 + \sum_{n=1}^{h^{Exp}} \left( c_2 + 2c_3(n+5)^2 + c_4 \cdot h + \sum_{p=2}^{n+4} \sum_{q=p+3}^{n+4} c_5 \right),$$

with  $c_4 \cdot h$  the cost of lines 10 – 17,  $c_3(n+5)^2$  the cost of lines 8 – 9,  $c_3$  and  $c_4$  being constants as well as  $c_1$  (lines 1 – 4),  $c_2$  (lines 6 – 7) and  $c_5$  (lines 20 – 30).

Making some simple mathematical manipulations we obtain that

$$C(h, Exp) = c_1 + h^{Exp}(c_2 + c_4 \cdot h) + 2c_3 \cdot \sum_{n=1}^{h^{Exp}} (n+5)^2 + \sum_{n=1}^{h^{Exp}} \sum_{p=1}^{n+3} \sum_{q=1}^{n-p+1} c_5.$$

Now, by counting the total number of squares on which the theorem is evaluated on a run of the algorithm, which is

$$\sum_{n=1}^{h^{Exp}} \sum_{p=1}^{n+3} \sum_{q=1}^{n-p+1} 1 = \sum_{n=1}^{h^{Exp}} \sum_{p=1}^{n+3} (n-p+1) = \sum_{n=1}^{h^{Exp}} \frac{n(n+1)}{2} = \frac{h^{3Exp} + 3h^{2Exp} + 2h^{Exp}}{6},$$

we can conclude that the computational complexity of Algorithm 1 is  $O(h^{3Exp})$ . Hence, by choosing  $Exp \leq \frac{2.5}{3}$  we can ensure that Algorithm 1 has a computational complexity asymptotically lower than the one we would have by calculating the matching distance.

<b>Algorithm 2</b> CountPoints( $S, p, q$ )	<b>Algorithm 3</b> MetricApprox( $A, B, Exp, thresh$ )
1: <b>for</b> $i = (p - 1)$ <b>to</b> $(p + 1)$ <b>do</b>	1: $Res = \text{MatchDistGridApprox}(A, B, Exp)$
2: <b>for</b> $j = (q - 1)$ <b>to</b> $(q + 1)$ <b>do</b>	2: <b>if</b> $Res > thresh$ <b>then</b>
3: $Res \leftarrow Res + S(i, j)$	3: $Val = [(V - U)/2 + Res]/2$
4: <b>end for</b>	4: <b>else</b>
5: <b>end for</b>	5: $Val = d_{match}(A, B)$
6: <b>return</b> $Result$	6: <b>return</b> $Val$

## 4 Experimental results

Our goal is to validate the theoretical framework introduced in the previous section. Through some experiments on persistence diagrams for 0th homology degree (a.k.a. *formal series* [10]), associated with 3D-models represented by triangle meshes, we will prove that our algorithm allows us to reduce the computational complexity in defining a matching distance-based metric over a given database, without greatly affecting the goodness of results (in terms of database classification).

To test the proposed framework we considered a database of 228 3D-surface mesh models introduced in [2]. The database is divided into 12 classes, each containing 19 elements obtained as follows: A null model taken from the Non Rigid World Benchmark [3] is considered together with six non-rigid transformations applied to it at three different strength levels. An example of the transformations and their strength levels is given in Table 1. To define the considered filtering functions, we proceeded as follows: For each triangle mesh  $M$  of vertices  $\{P_1, \dots, P_n\}$ , the center of mass  $B$  is computed, and the model is normalized to be contained in a unit sphere. Further, a vector  $\mathbf{w}$  is defined as

$$\mathbf{w} = \frac{\sum_{i=1}^n (P_i - B) \|P_i - B\|}{\sum_{i=1}^n \|P_i - B\|^2}.$$

Three filtering functions  $\varphi_1, \varphi_2, \varphi_3$  are computed on the vertices of  $M$ :  $\varphi_1$  is the distance from the line parallel to  $\mathbf{w}$  and passing through  $B$ ,  $\varphi_2$  is the distance from the plane orthogonal to  $\mathbf{w}$  and passing through  $B$ , and  $\varphi_3$  is the distance from  $B$ . The values of  $\varphi_1, \varphi_2$  and  $\varphi_3$  are then normalized so that they range in the interval  $[0, 1]$ . These filtering functions are translation and rotation invariant, as

well as scale invariant because of a priori normalization of the models. Moreover, the considered models are sufficiently generic (no point-symmetries occur etc...) to ensure that the vector  $\mathbf{w}$  is well-defined over the all database, as well as its orientation is stable.

Taking a filtering function  $\varphi$ , we can now induce a metric over our database by computing the matching distances  $d_{ij}^\varphi = d_{match}(D_0(M_i, \varphi), D_0(M_j, \varphi))$  for every  $i, j = 1, \dots, 228$ . To approximate such a metric, we applied Algorithm 1 to get a lower bound for each  $d_{ij}^\varphi$ , say  $Res_{ij}^\varphi$ . This procedure is controlled by a threshold,  $thresh^\varphi$ , obtained as follows: For every class in the database, 4 elements are (randomly) selected, and an average of the matching distances on this small subset is evaluated. The final value of  $thresh^\varphi$  is then the average over all the classes in the database. In this perspective, the value  $thresh^\varphi$  represents the average matching distance between two elements of the same class.

Now, if  $Res_{ij}^\varphi > thresh^\varphi$ , then we can assume that the shapes of  $M_i$  and  $M_j$  are quite dissimilar (compared w.r.t.  $\varphi$ ) and therefore it is sufficient to have just an estimation of  $d_{ij}^\varphi$ : We opted for  $((V - U)/2 + Res_{ij}^\varphi)/2$ , with  $V$  and  $U$  taken according to Remark 1. If  $Res_{ij}^\varphi \leq thresh^\varphi$ , then the exact value of  $d_{ij}^\varphi$  is computed. The overall process is described in Algorithm 3.

Table 2 (first column) shows the average precision/recall (PR) graphs induced by  $\varphi_1$ ,  $\varphi_2$  and  $\varphi_3$ , respectively, when considering the computation of the matching distances on the whole database and on some subparts of it after running Algorithm 1, with  $Exp$  set at two different values. As can be seen, our approximation strategy does not affect so much the PR performances even in the displayed worst case (filtering function  $\varphi_2$ ).

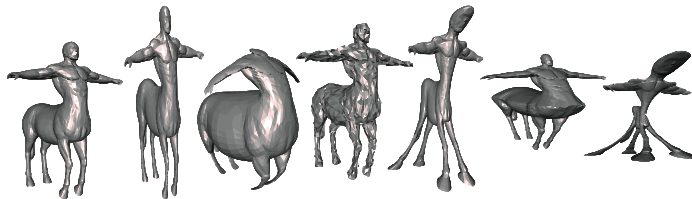
Table 2 (second column) gives a more general overview of the obtained results. From top to bottom, each graph shows the reduction in the computational costs – in terms of the percentage of computed matching distances used to build the metric approximations – and an evaluation of the PR performances according to the chosen values of  $Exp$ , for the filtering functions  $\varphi_1$ ,  $\varphi_2$  and  $\varphi_3$ , respectively. In particular, for a given value of  $Exp$  the evaluation of results is expressed as the average  $L_1$ -distance between the PR graph associated to that value  $Exp$  and the one obtained by computing all the matching distances between the elements in the database. The “critical Exp” depicted in all plots represents the value of  $Exp$  such that the cost of applying Algorithm 1 equals the one of computing the matching distance between two persistence diagrams.

As our plots show, it is possible to greatly reduce the computational costs by approximating the matching distance-based metric over the database, obtaining PR graphs which are quite close to the best possible.

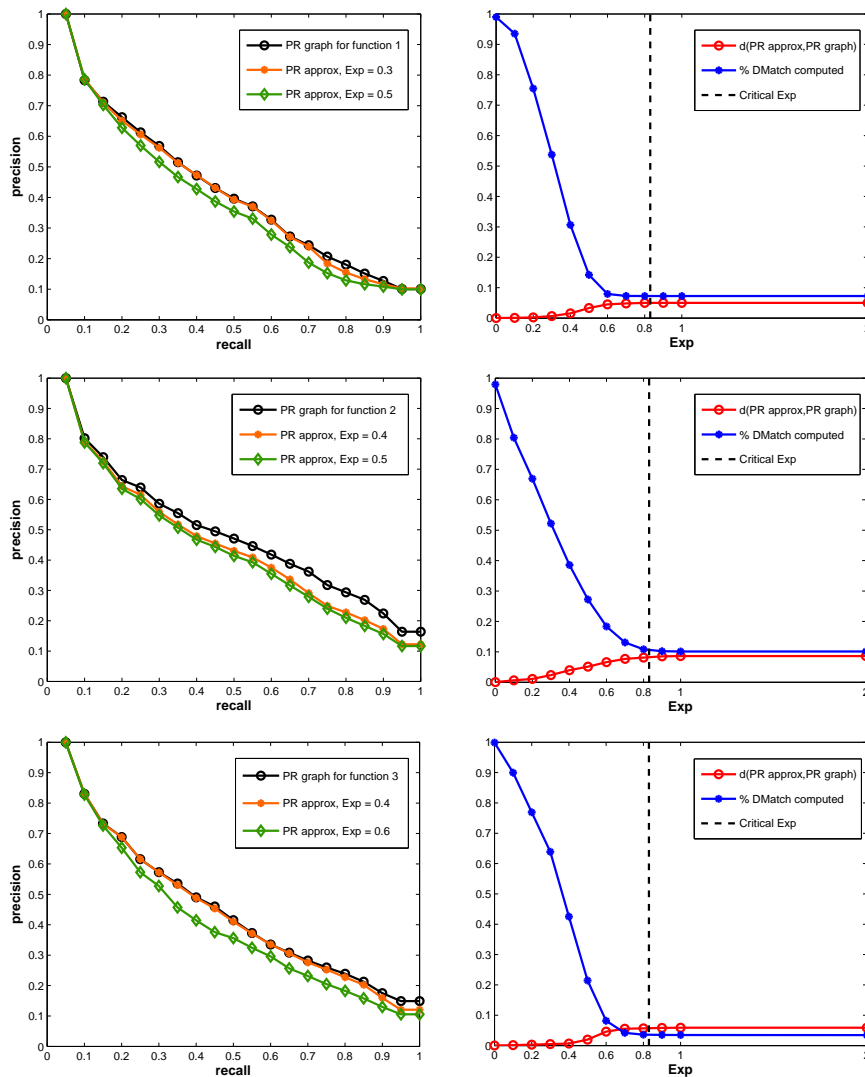
## 5 Conclusions

In this paper we introduced a multi-scale strategy to evaluate a (dis)similarity metric on a database of shapes – to be used for CRC purposes – using the concepts of persistence diagrams and matching distance. The proposed framework has been validated through experiments on 3D models represented by triangle





**Table 1.** The null model “Centaur0” and the 3<sup>rd</sup> strength level for each deformation.



**Table 2.** First column: PR graphs related to  $\varphi_1, \varphi_2, \varphi_3$  computing  $d_{match}$  on the whole database (black), and on subparts of it (PR approx) by virtue of Algorithm 1 for two different values of  $Exp$  (shaded); Second column: varying  $Exp$ , how the percentage of  $d_{match}$  computed and the distance between PR graph and PR approx vary.

meshes: The obtained results show that it is possible to provide an approximation of the metric induced by the matching distance between persistence diagrams without compromising the goodness of results – in terms of retrieval performance – and greatly reducing the computational costs coming from the exact evaluation of the matching distance.

For the next future we plan to generalize Algorithm 1 in such a way that the lower bound provided by Theorem 1 could be better exploited. We plan to do this by randomly generating the largest squares in the triangular area  $T$ , considered in Algorithm 1, allowing in this way partial covering of  $T$  and squares overlapping, and making that the smallest squares vary inside the wider ones. The expected result is to produce statistically better estimates of the matching distance lower bound through the use of a more flexible tool not stuck on a fixed tessellation like the one produced by Algorithm 1.

**Acknowledgments.** The authors wish to thank P. Frosini for suggesting the problem. However, the authors are solely responsible for any possible errors.

## References

1. Biasotti, S., De Floriani, L., Falcidieno, B., Frosini, P., Giorgi, D., Landi, C., Papaleo, L., Spagnuolo, M.: Describing shapes by geometrical-topological properties of real functions. *ACM Comput. Surv.* 40(4), 1–87 (2008)
2. Biasotti, S., Cerri, A., Frosini, P., Giorgi, D.: A new algorithm for computing the 2-dimensional matching distance between size functions. *Pattern Recognition Letters* 32(14), 1735 – 1746 (2011)
3. Bronstein, A., Bronstein, M., Kimmel, R.: *Numerical Geometry of Non-Rigid Shapes*. Springer Publishing Company, Incorporated, 1 edn. (2008)
4. Carlsson, G., Zomorodian, A., Collins, A., Guibas, L.J.: Persistence barcodes for shapes. *IJSM* 11(2), 149–187 (2005)
5. Chazal, F., Cohen-Steiner, D., Guibas, L.J., Mémoli, F., Oudot, S.: Gromov-hausdorff stable signatures for shapes using persistence. *Computer Graphics Forum* 28(5), 1393–1403 (2009)
6. d’Amico, M., Frosini, P., Landi, C.: Using matching distance in size theory: A survey. *Int. J. Imag. Syst. Tech.* 16(5), 154–161 (2006)
7. d’Amico, M., Frosini, P., Landi, C.: Natural pseudo-distance and optimal matching between reduced size functions. *Acta. Appl. Math.* 109, 527–554 (2010)
8. Di Fabio, B., Landi, C., Medri, F.: Recognition of occluded shapes using size functions. In: *Lecture Notes In Computer Science*. vol. 5716, pp. 642–651 (2009)
9. Edelsbrunner, H., Harer, J.: *Computational Topology: An Introduction*. American Mathematical Society (2009)
10. Frosini, P., Landi, C.: Size theory as a topological tool for computer vision. *Pattern Recogn. and Image Anal.* 9, 596–603 (1999)
11. Smeulders, A., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. *IEEE Trans. PAMI* 22(12) (2000)
12. Tangelder, J., Velkcamp, R.: A survey of content-based 3D shape retrieval methods. *Multimedia Tools and Applications* 39(3), 441–471 (2008)