2015 International Conference on 3D Vision

Graph Edit Distance based on Triangle-Stars Decomposition for Deformable 3D Objects Recognition

Kamel MADI
Université de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205, F-69622, France
kamel.madi@liris.cnrs.fr

Eric PAQUET
National Research Council Canada, Ottawa, Canada
eric.paquet@nrc-cnrc.gc.ca

Hamida SEBA , Hamamache KHEDDOUCI
Université de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205, F-69622, France
{hamida.seba , hamamache.kheddouci}@liris.cnrs.fr

Abstract

We consider the problem of comparing deformable 3D objects represented by graphs, i.e., triangular tessellations. We propose a new algorithm to measure the distance between triangular tessellations using a new decomposition of triangular tessellations into triangle-Stars. The proposed algorithm assures a minimum number of disjoint triangle-Stars, offers a better measure by covering a larger neighborhood and uses a set of descriptors which are invariant or at least oblivious under most common deformations. We prove that the proposed distance is a pseudo-metric. We analyse its time complexity and we present a set of experimental results which confirm the high performance and accuracy of our algorithm.

1. Introduction

In the field of 3D objects recognition, it is often required to compare different 3D objects represented by graphs. Using triangular tessellations, 3D objects may be compared with graph matching techniques. Graph matching is the process of finding a correspondence between nodes and edges of two graphs that satisfies a certain number of constraints ensuring that similar substructures in one graph are mapped to similar substructures in the other. Several approaches have been proposed to solve the graph matching problem [11, 5, 8, 10]. Graph edit distance is one of the most celebrated measures to determine the distance between graphs [6, 17, 15, 13]. It is defined as the minimum-cost sequence of edit operations that transform one graph into another. The tolerance to noise and distortion is one of the advantages of edit distance. Unfortunately, graph edit distance has a high computational complexity which grows exponentially with the number of nodes [7]. In this paper, we address the problem of comparing deformable or non-rigid shapes (such as human and animal bodies). The shapes considered are represented by graphs, *i.e.*, triangular tessellations. We propose a new distance for comparing deformable 3D objects. This distance is based on the decomposition of triangular tessellations into triangle-Stars. A triangle-Star is a connected component formed by the union of a triangle and its neighborhood. The number of triangle-Stars obtained is much smaller than the number of nodes and the number of classic stars [14, 21] and, as a result, the computational complexity is reduced. The proposed graph edit distance is based on triangle-Stars which is a local structure that covers a larger neighborhood than a classic star decomposition [14, 21]. Consequently, the correctness of the proposed dissimilarity measure is improved. This is justified by the fact that optimal methods are based on graphs global structures and, consequently, a larger local structure allows to be closer to the global one. The distance that we propose uses a set of parameters which are either invariant or at least oblivious under most common deformations. The remainder of the paper is organized as follows. Firstly, some useful definitions are introduced in Section 2. In Section 3, the proposed algorithm is described and its complexity analyzed. Then, experimental results are reported in Section 4. Finally, Section 5 concludes the paper.

2. Definitions

In this section, we introduce some useful definitions related to the problem of $graph\ edit\ distance$.

978-1-4673-8332-5/15 $31.00 © 2015 IEEE
DOI 10.1109/3DV.2015.14

Definition 1 (Graph) A graph g is a set of nodes connected by a set of edges. Formally, a graph g is a four tuple $g = (N, E, \alpha, \beta)$, where N is a finite not empty set of nodes. $E \subseteq N \times N$ is the set of edges. $\alpha : N \rightarrow L_N$ is the node labelling function, $\beta : E \rightarrow L_E$ is the edge labelling function while L_N and L_E are the labels associated with the nodes and edges respectively [2].

Definition 2 (Triangular tessellations) A triangular tessellation g_{Tr} is a graph defined by a set of nodes, edges and triangles. Formally, g_{Tr} is a graph defined by a six tuple $g_{Tr} = (N, E, T, \alpha, \beta, \theta)$, where T is a set of triangles and $\theta : T \rightarrow L_T$ is a labelling function and, L_T is the set of triangle labels.

Graph Edit Distance (GED) The Graph Edit Distance [14] between two graphs g_1 and g_2 is the minimum number of edit operations (minimum cost) to transform g_1 into g_2. A set of edit operations is given by insertions, deletions and substitutions (or relabeling) of graph elements (nodes and edges). We denote the substitution of two elements u and v by $(u \rightarrow v)$, the deletion of element u by $(u \rightarrow \varepsilon)$, and the insertion of element v by $(\varepsilon \rightarrow v)$. A cost is associated to each edit operation. A sequence of edit operations $e_1, , e_k$ transforming g_1 into g_2 is called an edit path between g_1 and g_2. However, for every pair of graphs (g_1 , g_2), several edit paths transforming g_1 into g_2 exist with different total costs. The edit distance of two graphs is then defined as the minimum cost edit path between the two graphs (g_1 , g_2).

3. Algorithm overview

In this section, we present a new decomposition of triangular tessellations into connected components that we call triangle-Stars. This decomposition aims to reduce the number of components while covering a larger number of neighbors. In addition, the proposed decomposition allows obtaining a representation which is invariant or at least oblivious under most common deformations. Prior to the decomposition, a strict total order on the triangles must be established. This order aims to reduce the number of triangle-Stars that is generated and guarantees the uniqueness of the resulting decomposition. Finally, we propose a distance (dissimilarity measure) between the triangle-Stars of the two triangular tessellations and address their matching. We also prove that the proposed distance is a pseudo-metric. We present the computational complexity of the proposed algorithm.

3.1. Graph decomposition

We propose a decomposition of a triangular tessellation graph into a set of connected components that we call triangle-Stars (TS). We define the concept of triangle-Star as follows:

Definition 3 (neighborhood of a triangle): two triangles are neighbors, if they share, at least, a common node. Let t_1 and t_2 two triangles and $N(t_1)$ and $N(t_2)$ their respective nodes. Then, t_1 and t_2 are $neighbors \Leftrightarrow \|N(t_1) \cap N(t_2)\| > 0$. In other words, the neighbors of a triangle t are triangles sharing at least a common node with the triangle t.

Definition 4 (triangle-Star): A triangle-Star ts is a labelled sub-graph, defined by a triangle and a set of its neighbors. Formally, a triangle-Star ts is a three tuple $ts = (t_r, T', \theta)$, where: t_r is the root triangle, T' is the set of adjacent triangles and $\theta : T \rightarrow L_T$ is the triangle labelling function while L_T as a set of labels associated with the triangles.

Triangle-Star features: Each triangle t_j is defined with six-tuple $t_j = (n_1, n_2, n_3, e_1, e_2, e_3)$. The nodes n_i are labelled by their Cartesian coordinates. In our case, the nodes n_i are labelled with three coordinates $n_i = (x, y, z)$ corresponding to the three dimensions. The edges $e_k = (n_p, n_w)$ are labelled (weighted) with the Euclidian distance between their associated nodes (n_p, n_w). The triangles are labelled by a three-tuple $t_j = (id, Area, Perimeter)$, where id is a number. Each triangle-Star is characterised by a set of descriptors, allowing the evaluation of the dissimilarity between triangle-Stars. We consider the following descriptors: Area of triangle-Star, Perimeter of triangle-Star, Area of the triangles forming the triangle-Stars, their Perimeters, the Weights associated with their edges, and the Degrees of their nodes.

Triangle-Star Vector representation A vector representation of triangle-Star is given by (see Table (4) for symbols descriptions):

$$\{AG(ts),\ PG(ts),\ \{A(t_i),\ P(t_i),\ W(t_{i,\ j=1\ldots3}),\ deg(t_{i,\ j=1\ldots3})\}_{i=1}^{i=\|T(ts)\|}\}$$

where:

- The triangles of triangle-Star ts are ranked according to their areas (descending order).
- The weights of edges are ranked by descending order.
- The degrees of nodes are ranked by descending order.
- All triangle-Stars TS are represented by vectors which has the same size : $size = 2 + (\Gamma * 8)$.

- Γ is the maximum number of triangles in the two set of triangle-Stars in the two compared graphs.
- If a triangle-Star ts has a number of triangle less than Γ , the rest of the vector is completed with $zeros$.

Definition 5 (Disjoint triangle-Stars): Two triangle-Stars ts_i and ts_j are $disjoints$, if they do not share a common triangle. let $i \neq j$, ts_i and ts_j are $disjoints \Rightarrow T(ts_i) \cap T(ts_j) = \emptyset$.

3.2. Triangles ordering

The proposed method of decomposition, allows to have disjoint triangle-Stars (Definition 5), which significantly reduces the number of components ($\|TS\| \ll \|N\| < \|T\|$, see Figure 3) while reducing the number of comparisons in between the triangle-Stars associated with the two triangular tessellations. However, according to the order considered, the set of triangle-Stars obtained may be different (see Example 1). Indeed, the same triangular tessellation may generate different sets of triangle-Stars, both in terms of cardinality and in terms of triangle-Stars obtained, if the ordering of the triangles is not the same. In order to ensure the **uniqueness** of the decomposition and a further reduced number of triangle-Stars, a descending strict total order must be established on the set of triangles prior to their decomposition into triangle-Stars (see Example 2). In order to establish a descending strict total order on the triangles set, each triangle is represented by a 10-tuple$< \|neighbors\|$; x_1, y_1, z_1 ; x_2, y_2, z_2 ; $x_3, y_3, z_3 >$: the number of neighbors and the coordinates x, y, z, in the reference frame defined by the Eigen vectors of the tensor of inertia associated with the tessellation, of the three nodes associated with the triangle. The number of neighbors $\|neighbors\|$ is used in order to further reduce the number of triangle-Stars. If two triangles have the same number of neighbors, the node's coordinates are utilised in order to ensure the uniqueness of the decomposition. The nodes of the triangle in the 10-tuple are ordered according to their coordinates, starting by the first coordinate x. In case of equality, the next coordinates are compared until an inequality is obtained. The coordinates of the nodes are solely considered in order to ensure the uniqueness the decomposition.

Example 1 Let a graph-tessellation G_{tr} containing 5 triangles $t_{1...5}$, with $\|N(t_1)\| = 1$, $\|N(t_2)\| = 3$, $\|N(t_3)\| = 2$, $\|N(t_4) = 1\|$ and $\|N(t_5)\| = 1$, where N is a triangle neighborhood (see Figure 1). By applying a descending order we obtained 2 triangle-Stars (see Table 1). And by applying the ascending order we obtained 3 triangle-stars (see Table 2).

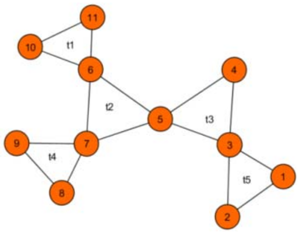


Figure 1. A graph-tessellation G_{tr}

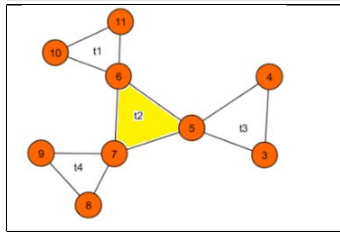
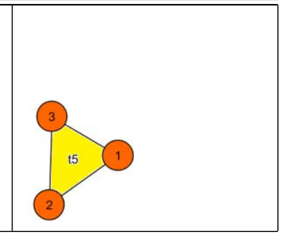
ST_1	ST_2
	

Table 1. The set of triangle-Stars using a Descending order.

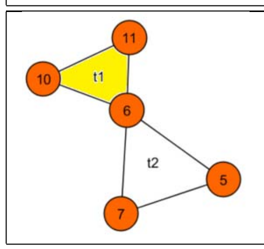
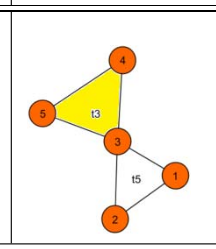
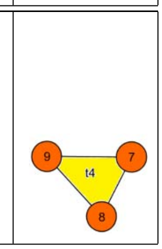
ST_1	ST_2	ST_3
		

Table 2. The set of triangle-Stars using a Ascending order.

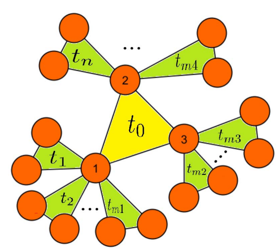


Figure 2. The triangle t_0 with n triangles neighbors.

Example 2 Let a triangle t_0 with n triangle neighbors $t_{1...n}$ (see Figure 2). If we consider a triangle order based on the number of neighbors with descending order, we obtained only one triangle-Star, otherwise, if the process of decomposition stars with any neighbors $t_{1...n}$ we obtained 3 triangle-Stars.

3.3. Triangle-Stars decomposition

Once the strict total order of the triangles is established, we evaluate the decomposition of the graph into triangle-

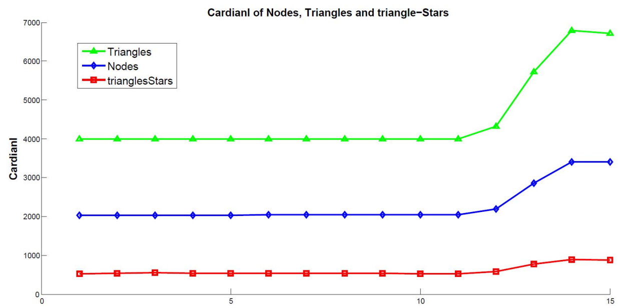


Figure 3. Comparison of the average number of nodes, triangles and triangle-Stars in the TOSCA database.

Stars. The process of decomposition is presented in the following algorithm (Algo. 1).

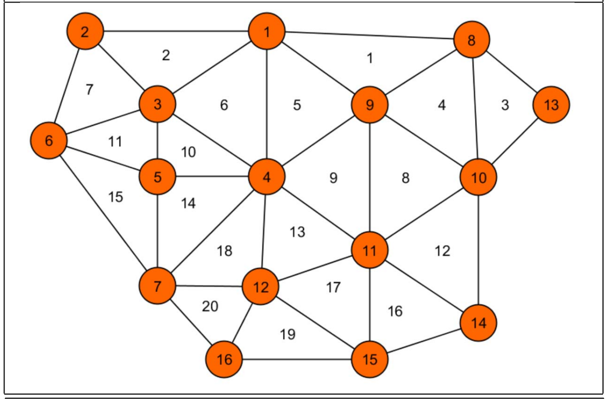
Algorithm 1 Graph decomposition into triangle-Stars.

Inputs: A graph g_{Tr}
Outputs: A set of triangle-Stars TS.
Begin
Apply a **descending** strict total order on the set of triangles of g_{Tr};
$TS = \emptyset$;
while $T(g_{Tr}) \neq \emptyset$ **do**
  $t_i = T(g_{Tr})[0]$
  $ts_i = t_i \cup$ **neighbors**(t_i) ;
  $TS = TS \cup ts_i$;
  $T(g_{Tr}) = T(g_{Tr}) - (t_i \cup$ **neighbors**$(t_i))$;
end while
return TS ;
End

We explore the list of triangles according to the defined order and we construct a triangle-Star which is defined by the current triangle and its neighbors (Definition 3). The process terminates when the list of triangles not yet explored is empty.

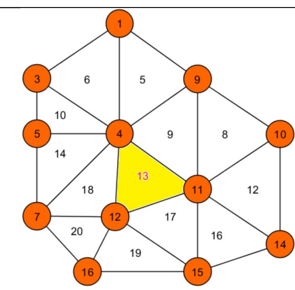
Example Let us consider a triangular tessellation defined as follow : $g_{Tr} = \{\, 16\, nodes \,,\, 20\, triangles\; t_{1\ldots20} \,\}$ (Table 3). The decomposition into triangles-Stars begins by constructing the first triangle-Star TS_1 using the triangle t_{13} with the set of its triangle neighbors. The triangle t_{13} is the first triangle chosen, since it is the one having the maximum number of neighbors, which is 12. In the remaining set of triangles not used in the construction of TS_1, the triangle t_1 that had 7 neighbors which is the maximum, is used to construct the second triangle-Star TS_2. TS_2 is constructed using t_1 and its 3 neighbours (not 7, because $T(TS_1) \cap T(TS_2) = \emptyset$). The third triangle-Star TS_3 is formed of t_{11} and its neighbors, t_{11} had 5 neighbors which is the maximum in the remaining set of triangles. TS_3 is

Graph G

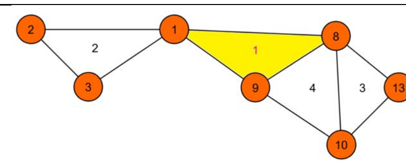


Triangle-Stars

TS_1



TS_2



TS_3

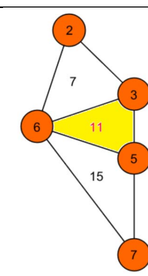


Table 3. Example, decomposition of a graph into a set of triangle-Stars.

constructed using t_{11} and its 2 neighbours (not 6 neighbours, because $\bigcup_{i=1}^{i=3} T(TS_i) = \emptyset$).

The proposed decomposition of triangular tessellations into triangle-Stars offers a reduced number of triangle-Stars ts as opposed to the number of nodes $\|TS\| \ll \|N\|$. The resulting triangle-Stars are disjoints, formally: let $i \neq j,\ \forall\, ts_i\,,\, ts_j \in TS(G) \Rightarrow T(ts_i) \cap T(ts_j) = \emptyset$.

The triangle-Star covers a larger local area than the classical star [14]. In addition, the proposed decomposition is unique.

3.4. Edit distance between triangle-Stars

In this section, we show how to compute the graph edit distance between triangle-Stars. The proposed similarity measure is intended for the comparison of deformable objects. Consequently, the set of parameters or descriptors must be invariant or at least oblivious under most common deformations. Indeed the proposed similarity measure is based on the following set of parameters: $Area$ of **triangle-Star**, $Perimeter$ of **triangle-Star**, $Area$ of triangles, $Perimeter$ of triangles, $Weights$ of edges and $Degrees$ of nodes. Formally a **triangle-Star** is represented as follows: $\{AG(ts),\ PG(ts),\ \{A(t_i),\ P(t_i),\ W(t_{i,\ j=1\ldots3}),\ deg(t_{i,\ j=1\ldots3})\}_{i=1}^{i=\|T(ts)\|}\}$. The similarity measure d between two triangle-Stars ts_i and ts_j is computed as:

$$d(ts_i, ts_j) = 1 - \frac{\sum_{k=1}^{k=6}\ sim_k\ (ts_i, ts_j)}{\sum_{k=1}^{k=6}\ \alpha_k} \quad (1)$$

The similarity measure d is a normalized value: $0 \leq d \leq 1$. The functions sim_k are defined as:

$$sim_1(ts_i, ts_j) = \alpha_1 * \frac{|AG(ts_i)\ -\ AG(ts_j)|}{AG_{MAX}} \quad (2)$$

$$sim_2(ts_i, ts_j) = \alpha_2 * \frac{|PG(ts_i)\ -\ PG(ts_j)|}{PG_{MAX}} \quad (3)$$

$$sim_3(ts_i, ts_j) = \alpha_3 * \frac{\sum_{l=1}^{l=\Gamma} |\ A(T(ts_i)_l)\ -\ A(T(ts_j)_l)\ |}{A_{MAX} * \Gamma} \quad (4)$$

$$sim_4(ts_i, ts_j) = \alpha_4 * \frac{\sum_{l=1}^{l=\Gamma} |\ P(t_{i,l})\ -\ P(t_{j,l})\ |}{P_{MAX} * \Gamma} \quad (5)$$

$$sim_5(ts_i, ts_j) = \alpha_5 * \frac{\sum_{l=1}^{l=\Gamma} \sum_{k=1}^{k=3} |\ W_{i,l,k}\ -\ W_{j,l,k}\ |}{3 * W_{MAX} * \Gamma} \quad (6)$$

$$sim_6(ts_i, ts_j) = \alpha_6 * \frac{\sum_{l=1}^{l=\Gamma} \sum_{k=1}^{k=3} |\ Deg_{i,l,k}\ -\ Deg_{j,l,k}\ |}{3 * Deg_{MAX} * \Gamma} \quad (7)$$

Where the symbols associated with the similarity measure are described in Table 4.

Symbol	Description
$t_{i,l}$	The triangle t_l in the triangle-Star ts_i : $t_l \in ts_i$
$W_{i,l,k}$	The weight (Euclidian distance) of the edge e_k of the triangle $t_l \in ts_i$
$Deg_{i,l,k}$	The degree of node n_k of the triangle $t_l \in ts_i$
Γ	Max number of triangles in the set triangle-Stars of the two graphs g_1 and g_2.
$\alpha_{k=1\ldots6}$	Parameters associated with the descriptors $\alpha_k \in \mathbb{N}$ and $\sum_{k=1}^{k=6}\ \alpha_k > 0$
$A(t_i)$	Area of the triangle i.
$P(t_i)$	Perimeter of the triangle i.
$AG(ts_i)$	Area of the triangle-Star i. $AG(ts_i) = \sum_{j=1}^{j=\|T(ts_j)\|} A(t_j)$
$PG(ts_i)$	Perimeter of the triangle-Star i. $DG(ts_i) = \sum_{j=1}^{j=\|T(ts_j)\|} D(t_j)$

Table 4. Symbols associated with the similarity measure and theirs description.

3.5. Edit distance between two triangular tessellations

The dissimilarity between two graphs represented by **triangle-Stars** is addressed in the last part of the algorithm. We call this dissimilarity measure *Triangle-Star Measure* TSM which aims to determine the best matching between the **triangle-Stars** associated with two graphs. The dissimilarity between two sets of **triangle-Stars** is defined as follows:

Definition 6 **(TSM)** Let $g_{Tr\ 1}$ and $g_{Tr\ 2}$ be two triangular tessellations, TS_1 and TS_2 their corresponding sets of triangle-Stars and M the set of all possible matching between TS_1 and TS_2. The similarity $TSM(TS_1, TS_2)$ is formulated as follow (Eq. 8 and Eq. 9):

$$TSM(TS_1,\ TS_2) = \max_{m \in M} \sum_{ts_i\ \in\ TS_1,\ m(ts_i)\ \in\ TS_2} d(ts_i,\ m(ts_i)) \quad (8)$$

The normalised dissimilarity $TSM(TS_1,\ TS_2$ is given by:

$$TSM(TS_1,\ TS_2) = 1 - \frac{\max_{m \in M} \sum_{ts_i\ \in\ TS_1,\ m(ts_i)\ \in\ TS_2} d(ts_i,\ m(ts_i))}{\max(\|TS_1\|,\ \|TS_2\|)} \quad (9)$$

The computation of $TSM(TS_1, TS_2)$ is equivalent to solving the assignment problem which is one of the fundamental combinatorial optimization problems that aim to find the minimum/maximum weight matching in a weighted bipartite graph. To solve this assignment problem, we define a $n \times n$ matrix D, with $n = \max(\|TS_1\|,\ \|TS_2\|)$. Each element $D_{i,j}$ of the matrix represents the dissimilarity measure $d(ts_i,\ ts_j)$ between a triangle-Star ts_i in TS_1 and a triangle-Star ts_j in TS_2. We apply the Hungarian algorithm [9] on the matrix D in order to find the best assignment in $\mathcal{O}(n^3)$ time. The resulting distance (dissimilarity) is compared to a threshold $th \in [0, 1]$ defined by an expert or by experimentation (depending on the database), in order to decide if the compared triangular tessellations are similar or not.

Example Let TS_1 and TS_2 two set of triangle-stars, $\|TS_1\| = \|TS_2\| = 4$. Let D the matrix of similarities between TS_1 and TS_2.

$$\begin{array}{c} \\ ts_{1,0} \\ ts_{1,1} \\ ts_{1,2} \\ ts_{1,3} \end{array} \begin{array}{c} \begin{array}{cccc} ts_{2,0} & ts_{2,1} & ts_{2,2} & ts_{2,3} \end{array} \\ \begin{pmatrix} 0.11 & \mathbf{0.90} & 0.25 & 0.21 \\ 0.10 & 0.15 & \mathbf{0.65} & 0.89 \\ \mathbf{0.67} & 0.03 & 0.51 & 0.17 \\ 0.66 & 0.88 & 0.33 & \mathbf{0.99} \end{pmatrix} \end{array}$$

The max sum, $similarity = 3.21$. The normalised dissimilarity (edit distance) is $TSM(TS_1\ ,\ TS_2) = 1 - \frac{3.21}{4} = 0.1975$

3.6. The pseudo metric

In this section we prove that the proposed distance is a **pseudo-metric**.

Definition : Let X a set of objects and $x, y, z \in X$. Let f be a function defined as follow $f : X \times X \longrightarrow \mathbb{R}$.

Let the following set of properties:

1. **non-negativity**: $f(x, y) \geq 0$
2. **symmetry**: $f(x, y) = f(y, x)$
3. **triangle inequality**: $f(x, y) \leq f(x, z)\ +\ f(z, y)$
4. **uniqueness**: $f(x, y) = 0 \Rightarrow x = y$

The function f is a metric if f satisfies the four mentioned properties and f is a pseudo-metric if f satisfies only the first three properties (1, 2 and 3).

Since f is a pseudo metric, a distance function may be defined between each pair of graphs. As a result, the similarity of the objects associated with these graphs may be efficiently determined [19, 1].

Lemma The proposed similarity measure TSM (Eq. 9) between two sets of triangles-stars TS_1 and TS_2 is a **pseudo-metric**.

Proof: From (Eq. 9) it may be concluded that if TSM is a **pseudo-metric** then d (Eq. 1) is a **pseudo-metric** which implies that sim_k (Eq. 2) is a **pseudo-metric**. Consequently, we shall prove that sim_k (Eq. 2) is a **pseudo-metric**. Proving that sim_k (Eq. 2) is a **pseudo-metric** is equivalent to check the first three properties in sim_k (Eq. 2). The functions sim_k are defined as follows:

$sim_k = \alpha_k * \frac{|x_1 - x_2|}{\beta}$ with $x_1, x_2, \in R_{\geq 0},\ \alpha_k, \beta \in R_{>0}$.

1. **non-negativity**: $TSM(TS_1, TS_2) \geq 0$. We have $sim_k \geq 0 \Rightarrow TSM \geq 0$ Thus TSM is non-negative.
2. **symmetry**: $TSM(TS_1, TS_2) = TSM(TS_2, TS_1)$. The proposed decomposition is unique and the TSM is only based on symmetrical operations (addition, sum, subtraction in absolute value). Consequently TSM is symmetric.
3. **triangle inequality**: $TSM(TS_1, TS_2) \leq TSM(TS_1, TS_3) + TSM(TS_3, TS_2)$. We have the triangle inequality verified in: $|x_1 - x_2| \leq |x_1 - x_3| + |x_3 - x_2|$. Thus the triangle inequality is verified in sim_k therefore, we have: $TSM(TS_1, TS_2) \leq TSM(TS_1, TS_3) + TSM(TS_3, TS_2)$. Consequently the triangle inequality is verified in TSM.

3.7. Complexity of the proposed algorithm

The most important part, in term of complexity, is the one solving the assignment problem. We used the Hungarian algorithm [14, 9] to find the best assignment in $\mathcal{O}(n^3)$ time, where n is the $maximum$ number of components in the two graphs compared. Let $n = \max(\|N_1\|,\ \|N_2\|)$ and $n' = \max(\|TS_1\|,\ \|TS_2\|)$, where N_i is the set of nodes and TS_i is the set of **triangle-Stars** in $g_{tr\ i}$. In the proposed decomposition, any **triangle-Star** has at least one triangle. Consequently, in the worst case, we have $n' = \frac{n}{3} = 0.33 * n$, which means that the complexity is $\mathcal{O}(0.036\ \ n^3)$. However the number of **triangle-Stars** depends on the structure of the underlying graph. For the **TOSCA** Database [3, 4] , which is used in our experiments, we have on average $n' = \frac{n}{3.9828} = 0.2510 * n$ which means that the complexity is: $\mathcal{O}(0.01582\ \ n^3)$. Since $\frac{\|N\|}{\|TS\|} \cong 1.1411 * \log(\|N\|)$, the complexity is of the order of $\mathcal{O}(0.67 * [\frac{n}{\log(n)}]^3)$.

4. Experimental results

In order to evaluate the proposed method, we undertook a set of experimentations and we compare our approach with some state-of-the-art shape-matching algorithms on the TOSCA Database [3, 4] . The TOSCA Database [3, 4] consist of 148 three-dimensional objects. Each object is represented by a triangular tessellation. The

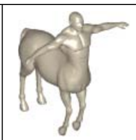
Class	$\|Class\|$	Pose 1	Pose 2
Centaur	6		
Gorilla	21		

Table 5. Some objects of the TOSCA Database [3, 4] .

Threshold	General Accuracy	Positive Accuracy	Negative Accuracy
0.06	83.24 %	82.29 %	83.35 %
0.07	78.3 %	91.1 %	76.79 %
0.08	74.6 %	95.46 %	72.14 %
0.05	88.03 %	64.92 %	90.75 %
0.04	90.54 %	40.75 %	96.40 %

Table 6. General, positive and negative accuracy according to the classification threshold.

database is categorized into twelve (12) classes. Each class contains an object with different deformations. The cardinality of the classes is not the same. On average, each triangular tessellation has 3154 nodes, 6220 triangles which result into 791 triangle-Stars. Table 5 shows some 3D objects of the TOSCA Database.

The proposed distance TSM is a parameterized distance having a set of parameters α_k allowing different configurations, the default value is: $\alpha_k = 1, \forall\, k$. In addition, we defined a threshold in order to improve the classification accuracy. Considering the set of parameters α_k and the threshold in our approach offer a error-tolerant distance and make the proposed approach invariant to different deformations. The parameters α_k and the threshold may be specified by inspection or by using machine learning techniques.

Table 6 shows some typical results with the following settings: $\forall\, k,\ \alpha_k = 1$ and $threshlod \in [0.04, 0.08]$. The different types of accuracy are defined later on.

For the TOSCA Database, we chose the following settings: $\forall\, k,\ \alpha_k = 1$ and $threshlod = 0.06$. We have computed the confusion matrix for each object belonging to TOSCA [3, 4] . Each element of the confusion matrix in Figure 4 is associated with the dissimilarity between objects i and j. **Dark** colours are associated to dissimilarity close or equal to zero, **Light** colours are associated to dissimilarity close or equal to one. Objects are similar if their dissimilarity is close or equal to zero. Using TOSCA, we generate a $n \times n$ matrix, with $n = 148$ (the number of 3D objects). Figure 4 show the confusion matrix associated

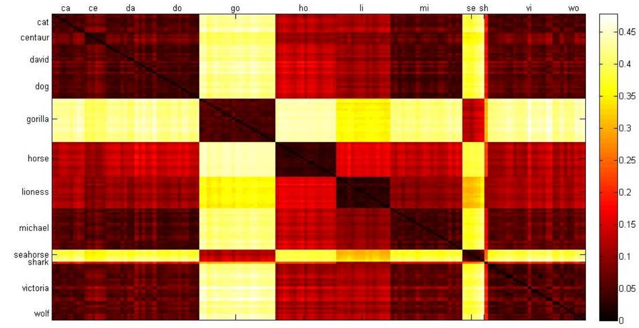


Figure 4. Confusion matrix associated with the TOSCA Database.

General Accuracy	Positive Accuracy	Negative Accuracy
83.24 %	82.29 %	83.35 %

Table 7. TSM Accuracy results for TOSCA.

with their dissimilarity. The darkest regions correspond to the block-diagonals of the confusion matrix which are associated with the intra-class dissimilarity. We observe that objects from the same classes are similar, for instance, the following classes: gorilla, centaur, horse ... etc. and we observe also that objects from different classes are dissimilar, for example: (cat, gorilla), (cat, seahorse), (gorilla, lioness) ... etc. In a few cases, there is some interclass similarity: the dog and the wolf, David and Victoria and, David and Michael. This is not surprising considering that their shape is relatively similar. All these observations demonstrate the efficiency of the proposed algorithm.

In order to measure the accuracy of the proposed distance TSM, we compute the distance between each pair of triangular tessellations in the database [3, 4] . Two triangular tessellations are considered similar if their distance is less than the chosen threshold (0.06). We use three (3) types of Accuracy: **Positive Accuracy** which is the percentage of elements well classified within their own class, **Negative Accuracy** which is the percentage of elements which are not attributed to classes in which they are not part, and **General Accuracy** which is the percentage of elements that satisfy both the Positive Accuracy and the Negative Accuracy. The accuracy results obtained by TSM are shown in Table 7.

Figure 5 shows the **precision-recall** curves, for the classes associated with the TOSCA Database [3, 4]. We used the following formulas to compute the precision and recall of an object from class i.

$$Recall = \frac{\|objects_found \in C_i\|}{\|C_i\|}$$

$$Precision = \frac{\|objects_found \in C_i\|}{\|objects_found\|}$$

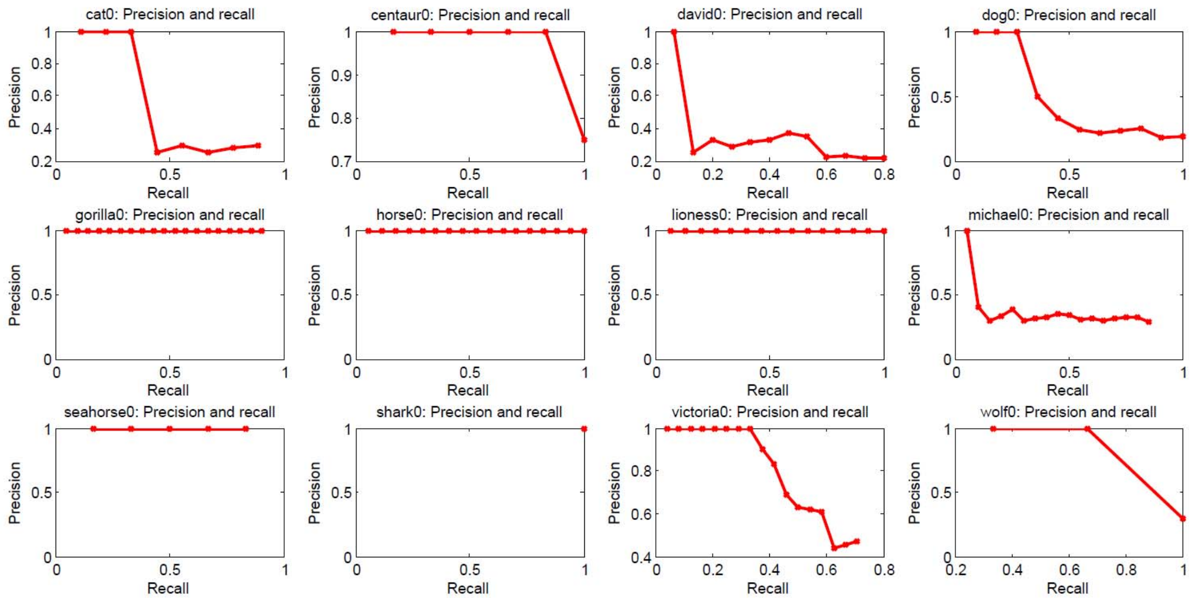


Figure 5. Precision-recall curves for nine distinct objects of the TOSCA Database.

Our method uses a threshold ($threshold = 0.06$), which means that only the objects which present a $dissimilarity \leq threshold$ are considering in the process of computing the recall and precision since, otherwise, they are automatically classified as dissimilar by our algorithm. As showed in Figure 5, we have obtained excellent precision-recall curves. For instance, $gorilla_0$, $horse_0$, $lioness_0$, $shark_0$ (only one element), and seahorse0 have a precision of 100% for a recall that goes from 86% to 100%.

In order to show the efficient of our approach, we compare it with a state-of-the-art set of shape-matching algorithms. The comparison is realized in term of precision and recall. The set of algorithms with which we compare are:

- **CAM**: 3D-Matching method using curve analysis [18].
- **GeodesicD2**: An extension of the Euclidean D2 [12], computed as a global distribution of geodesic distances in 3D shapes.
- **DSR**: The Hybrid Feature Vector, which is a combination of two view-based descriptors: the depth buffer and the silhouette and extent radialized function descriptor [20].
- **RSH**: The Ray-Based Approach with Spherical Harmonic Representation in which the authors of [16] align the models into the canonical position, extract the maximal extents and apply spherical harmonic.

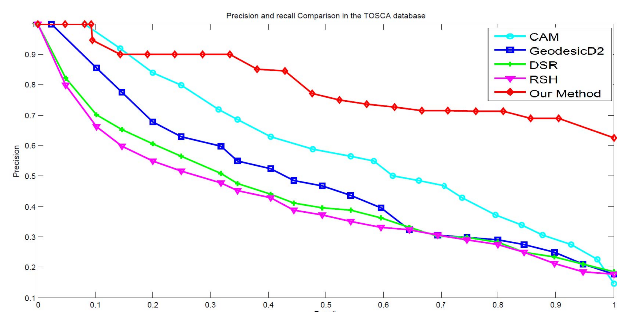


Figure 6. Precision and Recall plots comparing our approach to the CAM, GeodesicD2, DSR and RSH approaches on the TOSCA data set.

The Figure 6 shows the comparison of the Precision and Recall plot of our approach with these four methods (CAM, GeodesicD2, DSR and RSH). As the the curve of our approach is higher than the four approaches to which it was compared, we conclude that our method performs better than the others.(in [18], CAM was compared to GeodesicD2, DSR and RSH).

5. Conclusions

In this paper, we proposed a new matching algorithm for addressing the problem of comparing deformable 3D objects represented by graphs (triangular tessellations). The proposed approach is based on a new decomposition of triangular tessellations into triangle-Stars. The resulting triangle-Stars are used to determine the distance between triangular tessellation using the Hungarian algorithm. The proposed algorithm assures a minimum number of disjoints triangle-Stars, offers a better dissimilarity by covering a larger area of neighbors in triangle-Stars and used a set of descriptors which are invariant or at least oblivious under most common deformations. We proved that the proposed distance TSM is a pseudo-metric. The analysis of the time complexity and our experimental results confirm the high performance and accuracy of our algorithm. In future work, we project to extend our approach to partial shape matching.

Acknowledgment

We thank the anonymous reviewers for the valuable comments and suggestions.

This work was supported by the *LABEX IMU* (ANR-10-LABX-0088) of Université de Lyon, within the programs "*Investissements d'Avenir*" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR) and *"Avenir Lyon Saint-Etienne"* ("*PALSE*" Mobility Grant to National Research Council Canada).

References

[1] J. E. Barros, J. C. French, W. N. Martin, P. M. Kelly, and T. M. Cannon. Using the triangle inequality to reduce the number of comparisons required for similarity-based retrieval. pages 392–403, 1996. 6

[2] J. A. Bondy. Graph theory with applications. 1976. 2

[3] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Scientific Computing*, 28(5):1812–1836, 2006. 6, 7

[4] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Calculus of nonrigid surfaces for geometry and texture manipulation. *IEEE Trans. Vis. Comput. Graph.*, 13(5):902–913, 2007. 6, 7

[5] H. Bunke, A. Münger, and X. Jiang. Combinatorial search versus genetic algorithms: A case study based on the generalized median graph problem. *Pattern Recogn. Lett.*, 20(11-13):1271–1277, Nov. 1999. 1

[6] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recogn. Lett.*, 19(3-4):255–259, Mar. 1998. 1

[7] D. CONTE, P. FOGGIA, C. SANSONE, and M. VENTO. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):265–298, 2004. 1

[8] M. Gori, M. Maggini, and L. Sarti. Exact and approximate graph matching using random walks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(7):1100–1111, July 2005. 1

[9] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. 6

[10] S. Lagraa, H. Seba, R. Khennoufa, A. MBaya, and H. Kheddouci. A distance measure for large graphs based on prime graphs. *Pattern Recognition*, 47(9):2993 – 3005, 2014. 1

[11] R. Myers, R. C. Wilson, and E. R. Hancock. Bayesian graph edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:628–635, 2000. 1

[12] R. Osada, T. A. Funkhouser, B. Chazelle, and D. P. Dobkin. Shape distributions. *ACM Trans. Graph.*, 21(4):807–832, 2002. 8

[13] A. N. Papadopoulos and Y. Manolopoulos. Structure-based similarity search with graph histograms. pages 174–, 1999. 1

[14] K. Riesen and H. Bunke. Approximate graph edit distance computation by means of bipartite graph matching. *Image Vision Comput.*, 27(7):950–959, 2009. 1, 2, 5, 6

[15] A. Sanfeliu and K.-S. Fu. A distance measure between attributed relational graphs for pattern recognition. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-13(3):353–362, May 1983. 1

[16] D. Saupe and D. V. Vranic. 3d model retrieval with spherical harmonics and moments. pages 392–397, 2001. 8

[17] S. Sorlin, C. Solnon, and J.-M. Jolion. A generic graph distance measure based on multivalent matchings. 52:151–181, 2007. 1

[18] H. Tabia, M. Daoudi, J. Vandeborre, and O. Colot. A new 3d-matching method of nonrigid and partially similar models using curve analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(4):852–858, 2011. 8

[19] J. Vleugels and R. C. Veltkamp. Efficient image retrieval through vantage objects. *Pattern Recognition*, 35(1):69–80, 2002. 6

[20] D. Vranic. *3D Model Retrieval*. PhD thesis, University of of Leipzig, May 2004. 8

[21] Z. Zeng, A. K. H. Tung, J. Wang, J. Feng, and L. Zhou. Comparing stars: On approximating graph edit distance. *PVLDB*, 2(1):25–36, 2009. 1